



**SIMULATION OPTIMIZATION SYSTEMS**  
Research Laboratory

**ELECTRICAL ENGINEERING 3K4**  
**SIMULATION AND OPTIMIZATION I**

**J. W. Bandler**

**January 1988**

McMASTER UNIVERSITY  
Hamilton, Canada L8S 4L7  
Department of Electrical and Computer Engineering

ELECTRIC ENGINEERING 301  
SIMULATION AND OPTIMIZATION I

J. W. Bandler

January 1988

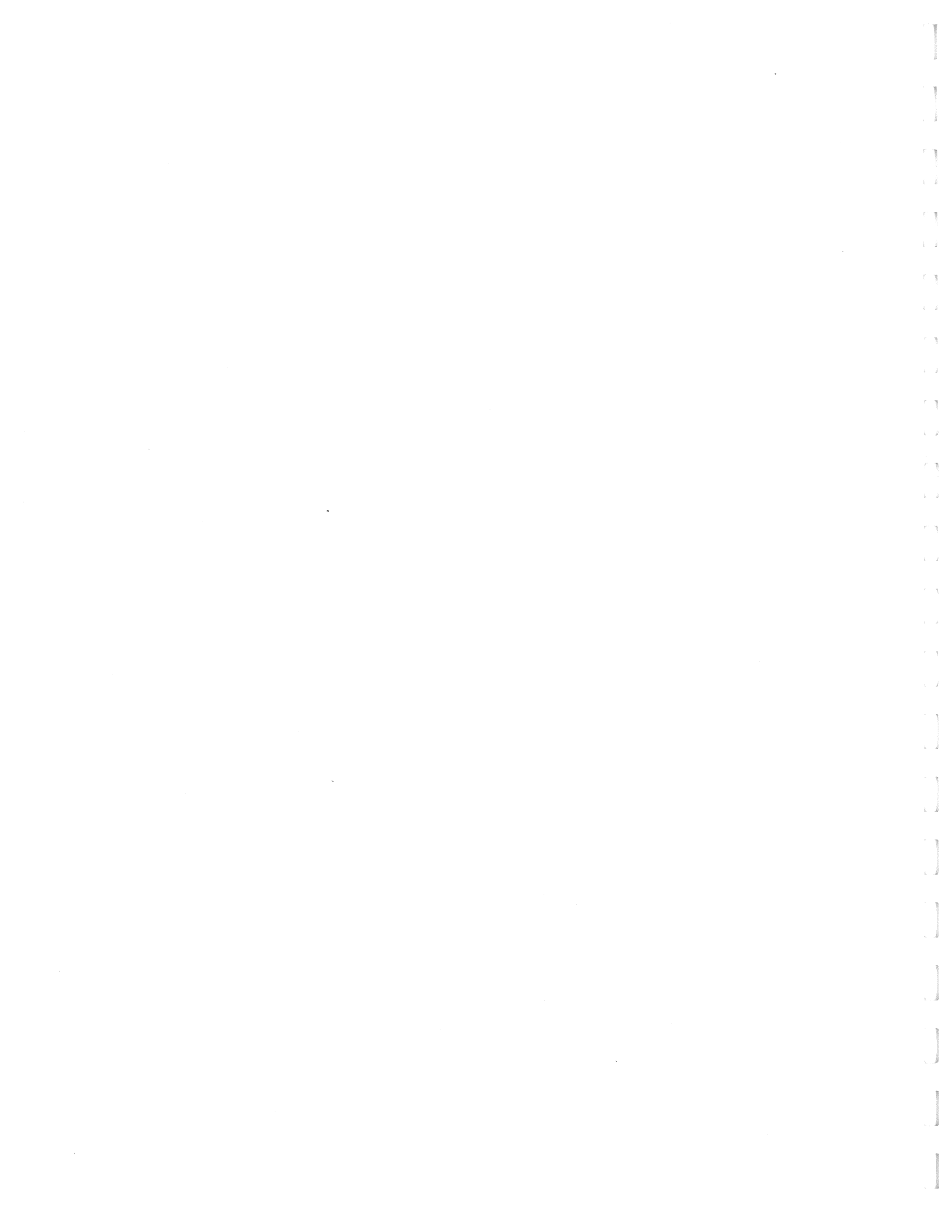
**ELECTRICAL ENGINEERING 3K4  
SIMULATION AND OPTIMIZATION I**

**J. W. Bandler**

**January 1988**

© J.W. Bandler 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler. This document may not be lent or circulated without this title page and its original cover.



**SECTION ONE**  
**INTRODUCTION TO SIMULATION AND OPTIMIZATION**

© J.W. Bandler 1984, 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



## INTRODUCTION

In order to set the stage for concepts and techniques to be considered later, some typical problems in the optimal computer-aided design of networks and systems will be discussed briefly. Only the essence of the problems is emphasized. As a first step in solving them, appropriate objective functions to be optimized are suggested and the variable parameters are identified. Some of the implications of the objective function formulations are also discussed. It is hoped that the electrical engineering student will be sufficiently motivated by this introduction to pursue the somewhat more mathematical material which will follow.

### The Optimization Problem

Minimize  $U$  where

$$U \triangleq U(\underset{\sim}{\phi})$$

and where

$$\underset{\sim}{\phi} \triangleq \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_k \end{bmatrix}.$$

$U$  is a scalar objective function of  $k$  independent variables or parameters

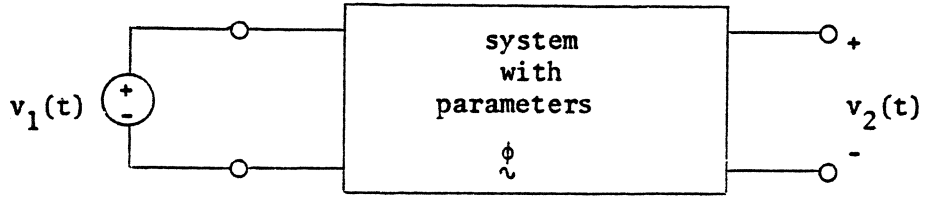
$\phi_1, \phi_2, \dots, \phi_k$ . Put another way, the objective is to adjust the variables to obtain an optimal set  $\check{\underset{\sim}{\phi}}$  such that

$$U(\check{\underset{\sim}{\phi}}) < U(\underset{\sim}{\phi})$$

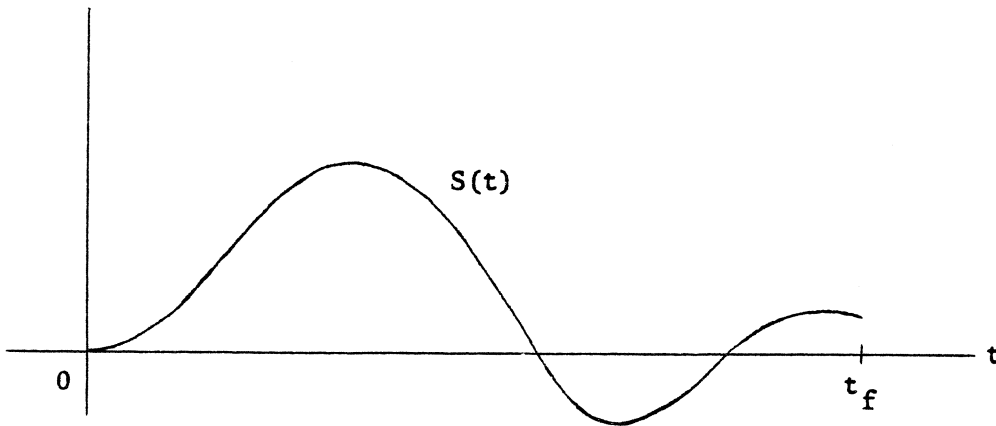
for all permissible sets of  $\underset{\sim}{\phi}$  in the neighbourhood of  $\check{\underset{\sim}{\phi}}$ . The point  $\check{\underset{\sim}{\phi}}$  defines a local minimum of  $U$  in the  $k$ -dimensional parameter space.

Approximation of a time response

A frequently occurring problem in system design is the approximation of a desired response in the time domain.



Let  $f(t) = v_2(t)$  be the response to the input  $v_1(t)$ . It is desired to approximate  $S(t)$  in a least squares sense.



We can set up the objective function

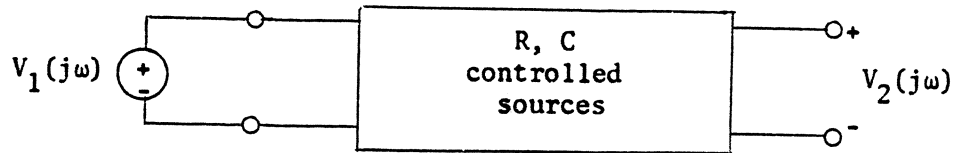
$$U = \int_0^{t_f} \{f(\phi, t) - S(t)\}^2 dt$$

to be minimized.



## Active filter design

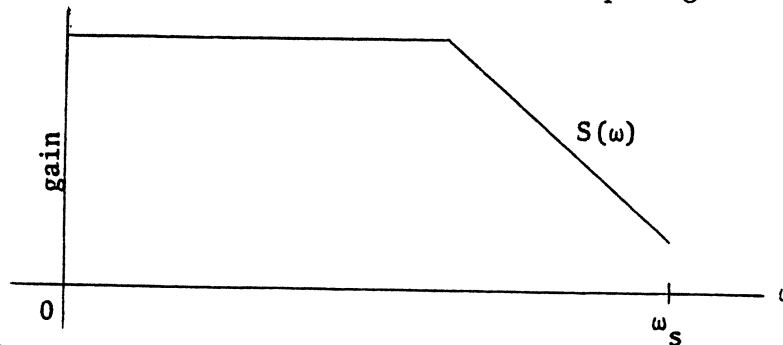
Consider the design of an active (inductorless) filter.



Suppose the problem is to obtain values for the R and C components which result in a gain  $G(\omega)$ , where

$$G(\omega) \triangleq 20 \log_{10} \left| \frac{V_2(j\omega)}{V_1(j\omega)} \right|$$

as close, in some sense, as possible to the desired low pass gain characteristic  $S(\omega)$  shown.



Let us form the objective function

$$U = \sum_{\omega_d \in \Omega_d} |G(\phi, \omega_d) - S(\omega_d)|^p$$

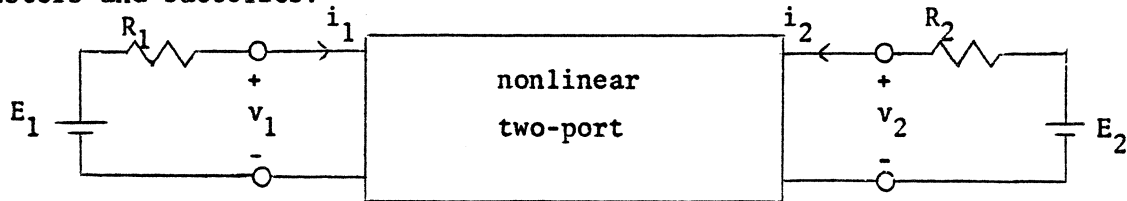
where  $\Omega_d$  is a discrete set of frequencies selected from the interval  $[0, \omega_s]$ , and  $p \geq 1$ . Here

$$\phi = \begin{bmatrix} R_1 \\ C_1 \\ R_2 \\ C_2 \\ \vdots \end{bmatrix}$$

Since  $G$  is usually a nonlinear function of  $\phi$  we can expect, for particular fixed values of  $p$ , several minima of  $U$  and hence several candidates for a "best" response. The question of which value of  $p$  to choose must also be settled.

### Nonlinear network d.c. analysis

Consider the nonlinear resistive two-port shown connected to linear resistors and batteries.



The problem is to determine the values of  $i_1$ ,  $v_1$ ,  $i_2$  and  $v_2$  which define the operating or equilibrium point.

Two mesh equations give

$$v_1 = E_1 - i_1 R_1$$

$$v_2 = E_2 - i_2 R_2$$

where  $E_1$ ,  $E_2$ ,  $R_1$  and  $R_2$  are constants. Let the network be current controlled.

Then

$$v_1 = r_1(i_1, i_2)$$

$$v_2 = r_2(i_1, i_2)$$

where  $r_1$  and  $r_2$  are specified functions. We have four equations, two of them nonlinear, to solve in four unknowns.

Our first thought might be to let

$$\underset{\sim}{\phi} = \begin{bmatrix} i_1 \\ v_1 \\ i_2 \\ v_2 \end{bmatrix} .$$

But  $v_1$  and  $v_2$  are dependent variables. So let

$$\underset{\sim}{\phi} = \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}$$

and solve

$$f_1(\phi) = E_1 - i_1 R_1 - r_1(i_1, i_2) = 0$$

$$f_2(\phi) = E_2 - i_2 R_2 - r_2(i_1, i_2) = 0$$

Consider

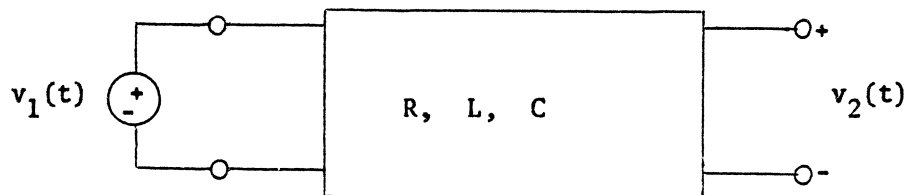
$$U = f_1^2 + f_2^2$$

Obviously, if  $U = 0$  then  $f_1^2 = 0$  and  $f_2^2 = 0$  so that  $f_1(\phi) = f_2(\phi) = 0$ .

The solution of equations problem has been reformulated as an optimization problem. In general, there may be many local minima. If  $\min U = 0$  we have a solution to the equations; if  $\min U \neq 0$  we have no solution. Conditions for the existence and uniqueness of the solution depend on the properties of the nonlinear functions, which in turn affect the form of the objective function  $U$ .

### Minimization of overshoot

Suppose it is desired to minimize the overshoot in the step response of a linear, time-invariant RLC network. Let  $v_1(t) = u(t)$ , the unit step.



The problem is to adjust the values of the R, L and C components, often within specific upper and lower bounds, so as to minimize

$$U = \max_{t \in [0, t_f]} v_2(\underset{\sim}{\phi}, t)$$

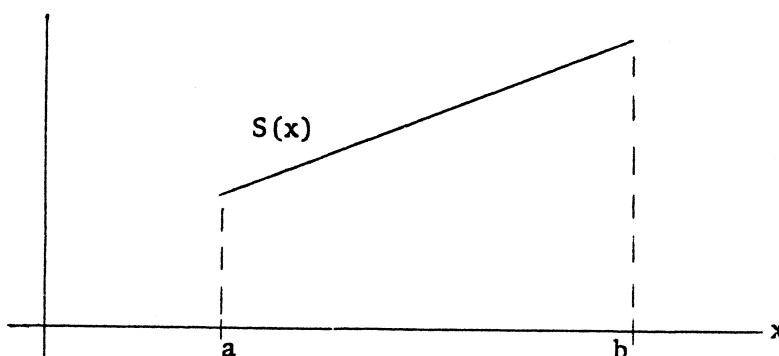
where  $t_f$  is some final value of time. In this case

$$\underset{\sim}{\phi} = \begin{bmatrix} R_1 \\ L_1 \\ C_1 \\ \vdots \end{bmatrix} .$$

### Approximation by a rational function

Consider the problem of approximating a specified continuous function  $S(x)$  by the rational approximating function

$$F(x) = \frac{P(x)}{Q(x)} = \frac{\sum_{i=0}^n a_i x^i}{1 + \sum_{i=1}^m b_i x^i}$$



To obtain an approximation in the Chebyshev or equal-ripple sense one needs to minimize

$$\max_{x \in [a,b]} |F(\phi, x) - S(x)|$$

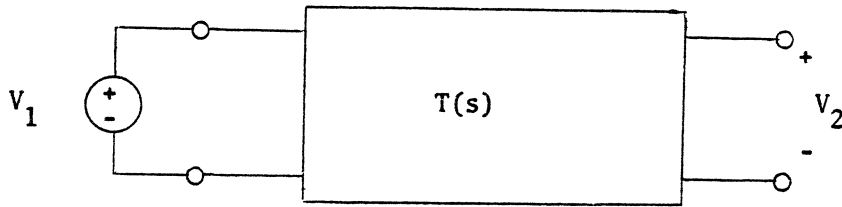
where

$$\phi = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

This problem falls into a class called minimax approximation, since we attempt to minimize the maximum deviation between the approximating function and the desired function.

The transfer function of a linear system

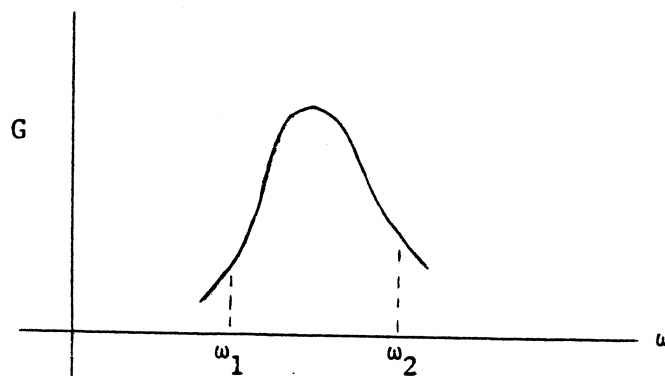
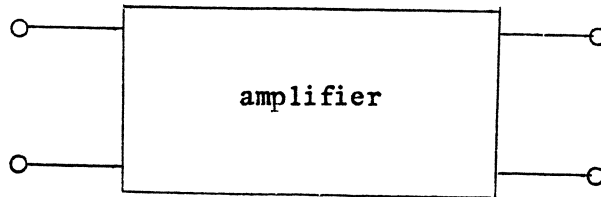
$$T(s) = \frac{V_2(s)}{V_1(s)}$$



can be optimized by such a formulation.

### Finding Response Maxima

A common problem is the determination of the peak in a response or the maximum error between a response and a specification. Consider, for example, the problem of finding the peak gain of an amplifier for a specified set of parameter values.



We require to maximize

$$U = G(\phi)$$

where

$$\phi = \omega$$

for

$$\omega_1 \leq \phi \leq \omega_2 \quad .$$

## Matching Coefficients of Rational Functions

Suppose we have a rational function

$$S(s) = \frac{\sum_{i=0}^n a_i s^i}{\sum_{i=0}^m b_i s^i}$$

with known (presumably, optimal) coefficients. Suppose we also have a network transfer function

$$T(\phi, s) = \frac{\sum_{i=0}^n a'_i(\phi) s^i}{\sum_{i=0}^m b'_i(\phi) s^i}$$

describing a network of the proper configuration. The coefficients are of course, in general, nonlinear functions of the network elements, as indicated. It is desired to adjust the element values so that, hopefully,  $T(s)$  can be made identical to  $S(s)$ .

Thus, we have to solve the nonlinear equations

$$\underset{\sim}{f} \stackrel{\Delta}{=} \begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ \cdot \\ f_{m+n+2} \end{bmatrix} = \begin{bmatrix} a'_0(\phi) \\ a'_1(\phi) \\ \vdots \\ a'_n(\phi) \\ b'_0(\phi) \\ b'_1(\phi) \\ \vdots \\ b'_m(\phi) \end{bmatrix} - \phi_{k+1} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \\ b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix} = \underset{\sim}{0}$$

where  $\underset{\sim}{f}$  is a vector of functions,  $\underset{\sim}{0}$  is a null vector and  $\phi_{k+1}$  is an unknown multiplicative constant.

One possible objective function to be minimized could be

$$U = \sum_{i=1}^{m+n+2} f_i^2(\phi, \phi_{k+1})$$

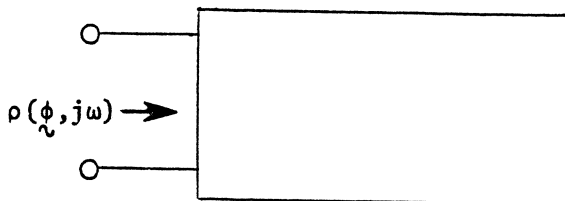
where  $U=0$  is the desired solution. Note that, in general,

$$m+n+2 \neq k+1 \quad .$$



## Sensitivity Analysis

A most important practical problem is the investigation of the effect on a nominally optimal response of the circuit or system parameters. A complete study of the subject would be rather involved. Let us illustrate the idea by means of the following example.



We have an optimal set of parameters, say  $\check{\phi}$  for the solution

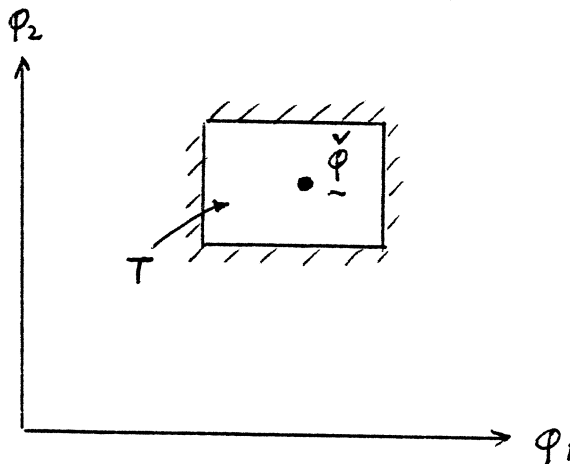
$$\check{U} = \min_{\check{\phi} \in R} \max_{\omega \in \Omega} |\rho(\check{\phi}, j\omega)|$$

that is to say, we have found the value of  $\check{\phi}$  within a feasible region  $R$  which minimizes the maximum magnitude of the input reflection coefficient over a region of frequencies  $\Omega$ .

One problem might be to find

$$\max_{\check{\phi} \in T, \omega \in \Omega} |\rho(\check{\phi}, j\omega)|$$

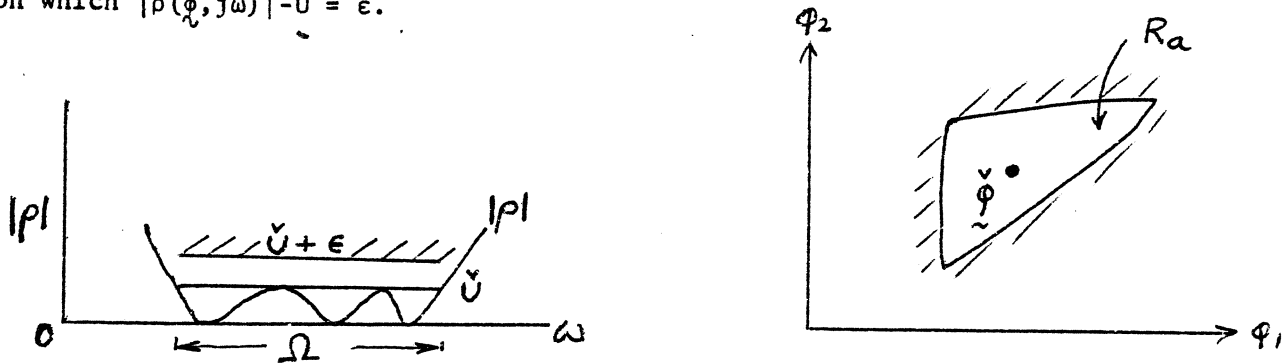
where  $T$  defines a region of tolerances on the components of  $\check{\phi}$  around  $\check{\phi}$ . In this case we would be trying to find the worst value of  $|\rho|$  in the frequency band of interest.



Suppose we can tolerate a deviation  $\epsilon$  from the optimum  $\check{U}$ . The problem of finding a region  $R_a$  of acceptable parameters, i.e., where

$$R_a \triangleq \{ \phi \mid | \rho(\phi, j\omega) | - \check{U} \leq \epsilon, \omega \in \Omega \}$$

is by no means a trivial one. The region would be described by its boundary, on which  $| \rho(\phi, j\omega) | - \check{U} = \epsilon$ .



A reason for wanting the region of acceptability is that one might be able to immediately predict the acceptability of any given design. Another that the nominal design may be more suitably located inside the region. Worst case tolerances can be evaluated.

The above considerations hold equally for large change sensitivities as for first-order sensitivities.

For first-order sensitivities, one could simplify the problem.

Suppose we are considering a function  $f(\phi)$ . We wish to obtain

$$\max_{\phi \in T} f(\phi)$$

using a linear approximation to  $f(\phi)$ . The problem becomes that of finding

$$\max_{\phi \in T} \{ f(\phi^0) + \nabla^T f(\phi^0) \Delta \phi \}$$

where

$$\Delta \phi = \phi - \phi^0$$

Note that  $f(\phi^0)$  and  $\nabla f(\phi^0)$  are constant. Suppose, as is usually the case,

$$T \triangleq \{ \phi \mid |\phi_i - \phi_i^0| \leq \epsilon_i, \quad i = 1, 2, \dots, k \}$$

where  $\epsilon_i$  is a (positive) prescribed tolerance limit on the parameter  $\phi_i$ .

Obviously the solution is given by

$$\sum_{i=1}^k \epsilon_i \left| \frac{\partial f}{\partial \phi_i} \right| .$$

A region of acceptability with respect to  $f(\phi)$  is given by

$$\{ \phi \mid \nabla^T f(\phi^0) (\phi - \phi^0) \leq \epsilon^0 \} .$$

If, as in the problem stated at the beginning of this section, we are dealing with  $|\rho(\phi, j\omega)|$  then

$$R_a \triangleq \{ \phi \mid |\rho(\check{\phi}, j\omega)| + \nabla^T |\rho(\check{\phi}, j\omega)| (\phi - \check{\phi}) - \check{U} \leq \epsilon, \quad \omega \in \Omega \} .$$

Thus, any set of parameters  $\phi$  which satisfies the above linear inequality at all values of  $\omega$  in the band of interest lies in a region of acceptability defined by first-order sensitivities.



**SECTION TWO**

**COLLECTED PROBLEMS IN  
COMPUTATIONAL METHODS, DESIGN  
AND OPTIMIZATION**

© J.W. Bandler 1983, 1988

This document originally appeared as Report SOS-83-16-N, September 1983. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.



Question 1 Develop an algorithm to efficiently calculate the value of

$$\frac{a_0 + a_2 s^2 + a_4 s^4 + \dots + a_n s^n}{b_1 s + b_3 s^3 + \dots + b_m s^m}$$

given  $m$ ,  $n$ , the coefficients and  $s$ . Test  $m$  and  $n$ . State the number of multiplications and divisions and the number of additions and subtractions.

Question 2 Develop an algorithm to efficiently calculate the value of

$$Z_0 \frac{Z_L + jZ_0 \tan\theta}{Z_0 + jZ_L \tan\theta}$$

given real  $Z_0$ ,  $0 \leq \theta \leq \pi$  and complex  $Z_L$ . Avoid  $\theta = \frac{\pi}{2}$ . State the number of multiplications and divisions, the number of additions and subtractions and the number of calls to a trigonometric function evaluation routine.

Question 3 Develop an algorithm to efficiently calculate the value of

$$a \sinh x + b \tanh x$$

given  $a$ ,  $b$  and  $e^x$ . State the number of multiplications and divisions, the number of additions and subtractions and the number of calls to function subprograms.

Question 4 State Horner's rule for polynomial evaluation. Explain its advantages compared with the direct method of evaluating a polynomial.

Question 5 Develop an algorithm to calculate as efficiently as possible the value of

$$a_1 \sin \theta + a_3 \sin 3\theta + a_5 \sin 5\theta$$

given  $a_1$ ,  $a_3$ ,  $a_5$  and  $\theta$ . State the number of multiplications and divisions, the number of additions and subtractions and the number of calls to a trigonometric function evaluation routine.

Question 6 Write an efficient algorithm for converting binary numbers to decimal numbers. Test it on the numbers 1101, 10111 and 1010101.

Question 7 Write and test on 44 an efficient algorithm for converting decimal numbers to binary numbers.

Question 8 Write an algorithm to efficiently evaluate  $\nabla F$  and  $\partial F/\partial s$  where

$$F(\underline{\phi}, s) = \sum_{i=0}^n a_i s^i$$

and

$$\underline{\phi} = \begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{bmatrix}, \quad \nabla F = \begin{bmatrix} \partial F/\partial a_0 \\ \partial F/\partial a_1 \\ \cdot \\ \cdot \\ \cdot \\ \partial F/\partial a_n \end{bmatrix}.$$



Question 9 Write an algorithm to efficiently calculate the value of the objective function

$$U(\underline{\phi}) = \sum_{i=1}^n (F(\underline{\phi}, t_i) - S(t_i))^2$$

and the gradient vector  $\nabla U(\underline{\phi})$   $m$  times for different  $\underline{\phi}$ , where

$$S(t) = \frac{3}{20} e^{-t} + \frac{1}{52} e^{-5t} - \frac{1}{65} e^{-2t} (3 \sin 2t + 11 \cos 2t)$$

is the specified function of time  $t$  (system response)

$$F(\underline{\phi}, t) = \frac{c}{\beta} e^{-\alpha t} \sin \beta t$$

is the approximating function of time (model response),

$$\underline{\phi} \triangleq \begin{bmatrix} \alpha \\ \beta \\ c \end{bmatrix} \text{ and } \underline{y} \triangleq \begin{bmatrix} \frac{\partial}{\partial \phi_1} \\ \frac{\partial}{\partial \phi_2} \\ \frac{\partial}{\partial \phi_3} \end{bmatrix} .$$

Question 10 Write an algorithm to efficiently calculate the frequency response  $V_2(j\omega)/V_1(j\omega)$  for the circuit of Fig. 1. Use the algorithm to calculate the response when  $L_1 = L_2 = 2H$ ,  $C_1 = C_2 = 0.5F$ , and  $\omega = 2$  rad/s.

Question 11 Write an algorithm to efficiently evaluate  $\nabla F$  where

$$F(\underline{\phi}, s) = \frac{\sum_{i=0}^n a_i s^i}{\sum_{i=0}^m b_i s^i}$$

and  $\underline{\phi} = [a_0 \ a_1 \ \dots \ a_n \ b_0 \ b_1 \ \dots \ b_m]^T$ .

Question 12 Write an algorithm to efficiently evaluate  $\tilde{V}T$  where

$$T(\phi, s) = \frac{1}{R_1 R_2 C_1 C_2 s^2 + (R_1 C_1 + R_1 C_2 + R_2 C_2) s + 1}$$

and

$$\phi = \begin{bmatrix} R_1 \\ C_1 \\ R_2 \\ C_2 \end{bmatrix}.$$

$T(s) = V_2(s)/V_1(s)$  for the circuit of Fig. 2.

Question 13 Show how the errors propagate in the calculation of

(a)  $\frac{a}{b - cd},$

(b)  $\frac{a}{b(c - d)},$

(c)  $\frac{xy}{u - v}.$

What is the relative error? Assuming all results are subject to the same roundoff errors, develop an expression yielding the maximum possible error.

Question 14 Derive an expression for the relative error in the computation of  $x/y$ . Neglect terms involving products of errors.

Question 15 Calculate and state the maximum number of multiplications and divisions in the efficient solution for  $\underline{x}$  of the linear system

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x \\ x \\ x \\ x \end{bmatrix},$$

where  $\underline{x} \triangleq [x_1 \ x_2 \ x_3 \ x_4]^T$ .

Question 16 Write an efficient Fortran program to calculate all the branch voltages and currents in the resistive ladder network of Fig. 3, allowing up to 100 resistors. Essential data:  $V_g, R_1, R_2, \dots, R_n$ .

Let  $n = 8, R_1 = R_3 = R_5 = R_7 = 3\Omega, R_2 = R_4 = R_6 = R_8 = 1\Omega$ . Calculate the voltages and currents for  $V_g = 1V$  using the program written.

Question 17 Write a program to calculate the input resistance of the circuit of Question 16. Use the program written to calculate the input resistance for the numerical example in Question 16.

Question 18 Write an efficient Fortran program using LU factorization to calculate and print out all the branch voltages and currents of the resistive ladder network of Fig. 4, allowing up to 99 resistors. Take account of symmetry and the tridiagonal nature of the admittance matrix.

Essential data:  $V_g, R_1, R_2, \dots, R_n$ .

Let  $n = 7, R_2 = R_4 = R_6 = 1/3\Omega, R_1 = R_3 = R_5 = R_7 = 1\Omega$ . Calculate the voltages and currents for  $V_g = 1V$  using the program.

Question 19 Write a program to calculate the input conductance of the circuit of Question 18. Use the program written to calculate the input conductance for the numerical example in Question 18.

Question 20 Consider the ladder network of Fig. 5.

- (a) Showing clearly all major steps, calculate the node voltages by
- (i) matrix inversion,
  - (ii) LU factorization.
- (b) What is the computational effort involved in (a)?
- (c) Set the right-hand source to zero and recalculate the node voltages. In general, what would the computational effort be for different excitations?

Question 21 Is the inverse of a tridiagonal matrix (in general) sparse, dense or tridiagonal? Justify your answer by a physically meaningful example.

Question 22 Define the term "relaxation method".

Question 23 State the Gauss-Seidel iterative formula for the solution of the linear system  $\underline{A} \underline{x} = \underline{b}$ , defining precisely any new symbols introduced.

Question 24 Factorize the following matrix into LU form utilizing available storage locations as much as possible:

$$\begin{bmatrix} 5 & -1 & 0 & 0 \\ -1 & 6 & -1 & 0 \\ 0 & -1 & 6 & -1 \\ 0 & 0 & -1 & 5 \end{bmatrix}$$

Question 25 Consider the resistive network in Fig. 6. Take  $G_1 = 2$  and  $G_3 = 1$  mho. Showing clearly all major steps, calculate the node voltages by LU factorization.

Question 26 Apply the Gauss-Seidel (relaxation) method to the circuit of Question 25. Take the initial node voltages as zero and use two iterations. Repeat with an overrelaxation factor of 1.5.

Question 27 Consider the resistive network shown in Fig. 7. Take  $G_1 = G_3 = G_5 = 1$  mho and  $R_2 = R_4 = 0.5$  ohm. Apply the Gauss-Seidel (relaxation) method to this network. Take the initial node voltages as zero and use two iterations. Repeat with an overrelaxation factor of 1.5.

Question 28 Consider the resistive network shown in Fig. 8. Let  $G_1 = 1$  and  $G_2 = 2$ . Showing clearly all major steps, apply two iterations of the Gauss-Seidel relaxation method starting with  $v_1 = 1$ ,  $v_2 = 0.5$ ,  $v_3 = 0$ . Continue the solution process with two iterations using an overrelaxation factor of 1.75. Expressing the nodal equations as error functions, calculate the Euclidean norm of the errors for each iteration.

Question 29 Consider the circuit shown in Fig. 9, which is operating in the sinusoidal steady state. Find  $V_3/V_1$  for this circuit at  $\omega = 2$  rad/s in the following ways, comparing the effort required. Take  $R_1 = R_2 = R_3 = 2\Omega$ ,  $C_1 = C_2 = C_3 = 1F$ . Show clearly all the steps in your calculations.

- (a) From an analytical expression of  $V_3(s)/V_1(s)$  derived by the Gauss elimination method.
- (b) By actual numerical inversion of the nodal admittance matrix.
- (c) By LU factorization of the nodal admittance matrix.
- (d) By assuming  $V_3$  and working backwards.
- (e) By ABCD or chain matrix analysis.

Question 30 Consider the circuit shown in Fig. 10, which is operating in the sinusoidal steady state. Find  $V_3/V_1$  for this network at  $\omega = 1$  rad/s in the following ways. Take  $R_1 = R_2 = R_3 = 1\Omega$ ,  $C_1 = C_2 = C_3 = 2F$ . Show clearly all the steps in your calculations.

- (a) From an analytical expression of  $V_3(s)/V_1(s)$ . Use the Gauss elimination method.
- (b) By actual numerical inversion of the nodal admittance matrix.
- (c) By LU factorization of the nodal admittance matrix.
- (d) By network reduction.
- (e) By assuming a value for  $V_3$  and working back through the ladder.

Question 31 Apply the Gauss-Seidel (relaxation) method to the circuit of Question 30. Take the initial node voltages to be zero and use two iterations. Repeat over with an overrelaxation factor of 1.5.

Question 32 Calculate and plot the reflection coefficient of the circuit shown in Fig. 11, where  $C_1 = 1.0F$ ,  $C_2 = 0.125F$ ,  $L = 2.0H$ ,  $0 \leq \omega \leq 4$  rad/s.

Question 33 Consider the iterative scheme

$$y^{i+1} = A^i y^i, \quad i = 1, 2, \dots, n$$

where the  $y$  vectors are of dimension 2 and the  $A$  matrices are  $2 \times 2$  with known values. Given the terminating conditions

$$y_1^{n+1} = 1,$$

$$y_1^1 = c^1 y_2^1,$$

where  $c^1$  is known, derive an analogous iterative scheme culminating in the evaluation of  $y^1$ .

Question 34 Consider the iterative scheme described in Question 33.

Given the terminating condition

$$y_1^1 = c^1 y_2^1,$$

where  $c^1$  is known, develop a computational scheme to evaluate

$$c^n = y_1^n / y_2^n.$$

Question 35 Assume that each matrix  $A^i$  in Question 33 is a function of a single variable  $x_i$ . Derive from first principles an approach to calculating  $\partial y_1^1 / \partial x$ , where  $x$  is a column vector containing the  $x_i$ ,  $i = 1, 2, \dots, n$ .

Question 36 Consider the system described by the iterative schemes

$$\tilde{y}^{i+1} = \tilde{A}^i \tilde{y}^i, \quad i = 1, 2, \dots, n, \quad i \neq j.$$

$$\tilde{z}^{i+1} = \tilde{B}^i \tilde{z}^i, \quad i = 1, 2, \dots, m,$$

the equation

$$\tilde{C} \begin{bmatrix} y_1^j \\ y_1^{j+1} \\ z_1^{m+1} \end{bmatrix} = \begin{bmatrix} -y_2^j \\ y_2^{j+1} \\ -z_2^{m+1} \end{bmatrix},$$

the terminating conditions

$$\begin{aligned} z_1^1 &= z_2^1, \\ y_1^1 &= y_2^1, \\ y_1^{n+1} &= 1, \end{aligned}$$

where the  $\tilde{y}$  and  $\tilde{z}$  vectors are of dimension 2 and the  $\tilde{A}$  and  $\tilde{B}$  matrices are 2 x 2 with known values and  $\tilde{C}$  is a given 3 x 3 matrix.

Carefully describe and explain an algorithm for evaluating  $y_2^{n+1}$  efficiently.

Question 37 Use the multi-dimensional Taylor series expansion to show that a turning point of a convex differentiable function is a global minimum. Justify all assumptions.

Question 38 Given a differentiable function  $f$  of many variables  $\tilde{x}$  and a corresponding direction vector  $\tilde{s}$ ,



$$\lim_{\lambda \rightarrow 0^+} \frac{f(\tilde{x} + \lambda \tilde{s}) - f(\tilde{x})}{\lambda} = \dots \dots \dots \text{(please state) ?}$$

Explain in a few words the meaning of the above expression.

Question 39 Use the method of Lagrange multipliers to prove that the greatest first-order change in a function of many variables occurs, for a given step size, in the direction of the gradient vector w.r.t. the variables.

Question 40 Use the method of Lagrange multipliers to minimize w.r.t.  $\phi_1$  and  $\phi_2$  the function

$$U = \phi_1^2 + \phi_2^2$$

subject to

$$\phi_1 + \phi_2 = 1$$

Sketch a diagram to illustrate the problem and its solution w.r.t.  $\phi_1$  and  $\phi_2$ . Verify your answer by substituting the constraint into the function.

Question 41 If  $g(\phi)$  is concave, verify that  $g(\phi) \geq 0$  describes a convex feasible region.

Question 42 Under what conditions could equality constraints be included in convex programming?

Question 43 Comment on each of the following concepts independently.

(a) The minimum of  $(\phi - a)^2$  and the maximum of  $b - (\phi - a)^2$ , where a and b are constants.

(b) The minimum of  $U$ , where

$$U = \begin{cases} -2\phi + 2, & \phi \leq 1 \\ \phi - 1, & \phi \geq 1 \end{cases}$$

and the minimum of  $U$  subject to  $0 \leq \phi \leq 3$ .

(c) The minimum of  $a\phi^2 + b$  contrasted with the minimum of  $a\phi^2 + b$  subject to  $\phi \geq 0$ , where  $a, b$  are constants.

(d) The number of equality constraints in a nonlinear program will generally be less than the number of independent variables.

Question 44 Find suitable transformations for the following constraints so that we can use an unconstrained optimization algorithm.

(a)  $0 \leq \phi_1 \leq \phi_2 \leq \dots \leq \phi_i \leq \dots \leq \phi_k$ .

(b)  $0 < \ell \leq \phi_2/\phi_1 \leq u, \phi_1 > 0, \phi_2 > 0$ .

Question 45 Write the following constraints in the form  $g_i(\phi) \geq 0, i = 1, 2, \dots, m$ .

(a)  $\ell_i \leq \phi_i \leq u_i, i = 1, 2, \dots, k$ .

(b)  $a \leq \phi_i/\phi_{i+1} \leq b, i = 1, 2, \dots, k-1$ .

(c)  $1 \leq \phi_1 \leq \phi_2 \leq \dots \leq \phi_k \leq 3$ .

(d)  $h_i(\phi) = 0, i = 1, 2, \dots, s$ .

Question 46 Discuss the scaling effects of the transformation  $\phi_i = \exp \phi_i'$ .

Question 47 Use an appropriate transformation to create the minimization of an unconstrained objective function for the problems

(a) minimize  $U = b\phi + c$  subject to  $\phi \geq 0$  with  $b > 0$ .

- (b) minimize  $U = a_1 \phi_1^2 + a_2 \phi_2^2$  subject to  $1 \leq \phi_i \leq 2$ ,  $i = 1, 2$  with  $a_1, a_2 > 0$ .

Question 48 Derive the gradient vector of  $U(\underline{\phi})$  w.r.t.  $\underline{\phi}$  for the objective functions

$$U = \int_{\psi_{\ell}}^{\psi_u} |e(\underline{\phi}, \psi)|^p d\psi$$

and

$$U = \sum_{i=1}^n |e_i(\underline{\phi})|^p,$$

where the appropriate error functions are complex.

Question 49 For the linear function (a polynomial is a special case)

$$F(\underline{\phi}, \psi) = \sum_{i=1}^k \phi_i f_i(\psi),$$

- (a) Formulate the discrete minimax approximation of  $S(\psi)$  by  $F(\underline{\phi}, \psi)$  as a linear programming problem, assuming  $\underline{\phi}$  to be unconstrained.
- (b) Assuming an objective function of the form of

$$U = \sum_{i=1}^n [e_i(\underline{\phi})]^p$$

derive the gradient vector of  $U$  and the Hessian matrix w.r.t.  $\underline{\phi}$ .

Question 50 Derive and compare the Newton methods for (a) minimization of a nonlinear differentiable objective function of many variables (as required in design), and (b) solving systems of nonlinear simultaneous

equations (as required in nonlinear d.c. network analysis). Sketch carefully each process for a single nonlinear function of a single variable indicating the various iterations. Under what conditions would you expect divergence from the solution?

Question 51 Derive from first principles Newton's method for function minimization w.r.t. many variables. Define all symbols introduced. Under what conditions would you expect convergence to a minimum? Prove that the direction of search is downhill if the Hessian matrix is positive definite. Sketch diagrams w.r.t. one variable showing

- (a) convergence to a minimum,
- (b) convergence to a maximum, and
- (c) oscillatory behaviour.

Describe and explain the "damped" Newton method.

Question 52 Derive carefully from first principles a numerical approach to finding the gradient vector  $\partial f / \partial \underline{x}$  subject to the system of equations  $\underline{h}(\underline{x}, \underline{y}) = \underline{0}$  given values for  $\underline{x}$ , where  $f \equiv f(\underline{y}(\underline{x}), \underline{x})$  is a scalar function and where the vector  $\underline{h}$  is nonlinear both in  $\underline{x}$  and in  $\underline{y}$ . Assume that  $\underline{h}$  and  $\underline{y}$  have the same dimensions and that the Jacobian of  $\underline{h}$  w.r.t.  $\underline{y}$  is nonsingular. Define all symbols used, and exhibit the structure of all matrices employed. Summarize the main steps of the computational procedure you would employ to solve a large problem.

Question 53 Define the term "positive definite" as it relates to a square symmetric matrix.

Question 54 Provide and discuss a link between the Hessian matrix of a differentiable function  $U(\phi)$ , where  $\phi$  is a  $k$ -vector, with the Jacobian matrix of  $\underline{f}(\phi)$ , where  $\underline{f}$  is a  $k$ -vector of functions of  $\phi$ .

Question 55 For the resistor-diode network shown in Fig. 12, illustrate with the aid of an  $i$ - $v$  diagram an iterative method of finding  $v$  at d.c. State Newton's method for solving this problem and derive the network model corresponding to the situation at the  $j$ th iteration. What is the significance of this model?

Question 56 We wish to calculate  $\partial f / \partial \underline{x}$  subject to  $\underline{h}(\underline{x}, \underline{y}) = \underline{0}$  where  $f \equiv f(\underline{y}(\underline{x}), \underline{x})$  given values for  $\underline{x}$ .

Explain fully the formula

$$\left. \frac{\partial f}{\partial \underline{x}} \right|_{\underline{h}=\underline{0}} = - \frac{\partial \underline{h}^T}{\partial \underline{x}} \hat{\underline{y}} + \frac{\partial f}{\partial \underline{x}},$$

where  $\hat{\underline{y}}$  is the solution to

$$\left( \frac{\partial \underline{h}^T}{\partial \underline{y}} \right) \hat{\underline{y}} = \frac{\partial f}{\partial \underline{y}}.$$

Describe the computational and analytical effort required in any given problem.

Let

$$\begin{aligned} 4x_1^2 y_1^2 - 3y_2 - 2 &= 0, \\ -x_1 y_1 + 2x_2^2 y_1 y_2 - 3 &= 0, \\ f &= y_1^2 + y_2^2 + 2x_2. \end{aligned}$$

Set up all the matrices and vectors required both for the solution of the nonlinear equations and also for the evaluation of  $\partial f / \partial \underline{x}$  s.t.  $\underline{h} = \underline{0}$ .

Question 57 Write down and define the first three terms of the multidimensional Taylor series expansion of a scalar function  $U$  of many variables  $\phi$ , defining any expressions used appropriately.

Question 58 Show that a step in the negative gradient direction reduces the function (neglecting second and higher-order terms) unless the gradient vector is zero.

Question 59 Derive a formula to approximately calculate all first partial derivatives of a function of  $k$  variables by perturbation, using  $2k$  function evaluations.

Question 60 What are the implication of a positive-semidefinite Hessian matrix in minimization problems?

Question 61 Derive Newton's method for function minimization. Explain under what conditions you would expect convergence. Sketch the algorithm for a function of one variable showing

- (i) a convergent process, and
- (ii) a divergent process.

Question 62 Write down a quadratic function of many variables and express its gradient vector and Hessian matrix in terms of constants involved in the function.

Question 63 Write down an objective function which can be minimized in an effort to solve the system of nonlinear equations  $\underline{f} = \underline{Q}$ . Differentiate it w.r.t. the variables and express the gradient vector in compact form.

Question 64 What is the implication of a negative first-order term in the multidimensional Taylor expansion of a differentiable function of many variables? Sketch your answer w.r.t. a function of two variables.

Question 65 State the principle behind the steepest descent approach to minimizing functions and sketch carefully the path taken on a contour diagram w.r.t. two variables.

Question 66 Write a simple Fortran program to implement steepest descent in the minimization of a scalar differentiable function of many variables and test it on suitable examples.

Question 67 Write a simple program to implement the one-at-a-time method of direct search for the minimization without derivatives of a function of many variables and test it on suitable examples.

Question 68 Describe the pattern search algorithm. Illustrate it on two-dimensional sketches of contours of a function to be minimized, noting exploratory moves, pattern moves and base points. Discuss any advantages enjoyed by this search method.

Question 69 Contrast the method of steepest descent with the method of changing one variable at a time to minimize an unconstrained function. Provide algorithms for both methods.

Question 70 Describe pitfalls in attempting the solution of constrained optimization problems using the algorithms of Question 69.

Question 71 Explain the concept norm. Give examples in (a) the continuous, (b) the discrete, approximation of a specified function of an independent variable by an appropriate function of many variables on a given interval of the independent variable. Use diagrams to illustrate your answer.

Question 72 For an electrical circuit design problem with upper and lower response specifications, explain the role of relative differences in the weighting factor(s) in the error functions. Distinguish the cases of specifications violated and specifications satisfied.

Question 73 Sketch contour and vector diagrams relating to constrained optimization problems illustrating the application of Kuhn-Tucker (KT) necessary conditions and showing

- (a) Points satisfying the KT conditions for minimization.
- (b) Points satisfying the KT conditions for maximization.
- (c) Points not satisfying the KT conditions for either maximization or minimization.



Question 74

- (a) What is a convex function?
- (b) What is a convex region?
- (c) How are these concepts related to a nonlinear optimization problem?

Question 75 Discuss the necessary conditions for an unconstrained optimum of a differentiable function. Derive them from

- (a) Conditions for a minimax optimum.
- (b) Conditions for a constrained minimum.

Question 76 Sketch contours and vector diagrams to illustrate the application of the Kuhn-Tucker (KT) conditions for a point satisfying the KT conditions for maximization of a constrained function.

Question 77 Sketch curves of  $|x - x^0|^p$  against  $x$  for  $p = 0.5, 1, 2, 4$  and  $\infty$ . Discuss the differentiability and convexity of these curves.

Question 78 Sketch in two dimensions the unit spheres centered at  $x^0$  defined by

$$\|x - x^0\|_p \leq 1$$

for  $p = 1, 2, 4$  and  $\infty$ . Comment on the convexity of these regions and the corresponding one for  $p = 0.5$ .

Question 79 Derive the necessary conditions (NC) for a minimax optimum for a set of nonlinear differentiable functions from the Kuhn-Tucker conditions (necessary conditions for a constrained minimum). Illustrate the results for the special cases of

- (a) a single function satisfying NC,
- (b) two active functions satisfying NC,
- (c) three active functions satisfying NC,
- (d) two active functions not satisfying NC.

Question 80 Draw a diagram for violated specifications that would illustrate the situation of multiple optimization of the frequency response and time response of an electrical circuit. Write down error functions in a form suitable for minimax optimization.

Question 81 Set up as a minimization problem the solution of the complex nodal equations of a linear analog circuit, required simultaneously for a number of frequencies. Identify clearly and compactly the objective function, the variables and any necessary gradient vectors required by the optimization program.

Question 82 Consider the problem of minimizing

$$U = \phi_3(\phi_1 + \phi_2)^2$$

subject to

$$g_1 = \phi_1 - \phi_2^2 \geq 0, \quad g_2 = \phi_2 \geq 0, \quad h = (\phi_1 + \phi_2)\phi_3 - 1 = 0.$$

Is this a convex programming problem? Formulate it for solution by the sequential unconstrained minimization method. Starting with a feasible point, show how the constrained minimum is approached as the parameter  $r \rightarrow 0$ . Draw a contour sketch to illustrate the process. Are the conditions for a constrained minimum satisfied?

Question 83 Apply the Fletcher-Powell-Davidon updating formula to the minimization of

$$\phi_1^2 + 2\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 1$$

w.r.t.  $\phi_1$  and  $\phi_2$  starting at  $\phi_1 = 0$ ,  $\phi_2 = 0$ , showing all steps explicitly and commenting on the results obtained.

Question 84 Apply the conjugate gradient algorithm for minimizing a differentiable function of many variables to the minimization of

$$\phi_1^2 + 2\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 1$$

w.r.t.  $\phi_1$  and  $\phi_2$  starting at  $\phi_1 = 0$ ,  $\phi_2 = 0$ , showing all steps explicitly and commenting on the results obtained.

Question 85 Apply the conjugate gradient algorithm for minimizing a differentiable function of many variables to the following data.

$$\text{Point: } \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 8 \\ 2 \end{bmatrix}, \begin{bmatrix} 8.4 \\ 2.45 \end{bmatrix}, \dots$$

$$\text{Gradient: } \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -2 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}, \dots$$

Sketch contours of a reasonable function that might have produced these numbers and plot the path taken by the algorithm.

Question 86 Consider the linear programming problem

$$\text{minimize } \phi_1 + 0.5 \phi_2 - 1$$

w.r.t.  $\phi_1, \phi_2$  subject to  $\phi_1 \geq 0$ ,  $\phi_2 \geq 0$ ,  $\phi_1 + \phi_2 \geq 1$ . Starting at  $\phi_1 = 2$ ,  $\phi_2 = 0$ , solve this analytically by steepest descent. Show how two one-dimensional searches yield the exact solution. Verify that the

Kuhn-Tucker relations (the necessary conditions for an optimum) are satisfied only at the solution.

Question 87 Minimize w.r.t.  $\phi$

$$U = \phi_1^2 + 4\phi_2^2$$

subject to

$$\phi_1 + 2\phi_2 - 1 = 0$$

The function has a minimum value of 0.5 at  $\phi_1 = 0.5$ ,  $\phi_2 = 0.25$ .

Suggested starting point:  $\phi_1 = \phi_2 = 1$ .

[Source: Fletcher (1970). See also Charalambous (1973).]

Question 88 Sketch contours of the function

$$V = \max[U, U + \alpha h, U - \alpha h]$$

w.r.t.  $\phi$  for  $U = \phi_1^2 + 4\phi_2^2$  and  $h = \phi_1 + 2\phi_2 - 1$  in the vicinity of the solution stated in Question 87 for  $\alpha = 0.1, 1.0$  and  $100$ , taking care to indicate points of discontinuous derivatives.

[Source: Bandler and Charalambous (1974).]

Question 89 Minimize w.r.t.  $\phi$

$$f = -\phi_1 \phi_2 \phi_3$$

subject to

$$\phi_i \geq 0, \quad i = 1, 2, 3,$$

$$20 - \phi_1 \geq 0, \quad 11 - \phi_2 \geq 0, \quad 42 - \phi_3 \geq 0,$$

$$72 - \phi_1 - 2\phi_2 - 2\phi_3 \geq 0.$$

The function has a minimum of  $-3300$  at  $\phi_1 = 20$ ,  $\phi_2 = 11$ ,  $\phi_3 = 15$ . This problem is referred to as the Post Office Parcel problem.

[Source: Rosenbrock (1960). See also Bandler and Charalambous (1974).]

Question 90 Minimize w.r.t.  $\phi$

$$f = \phi_1^2 + \phi_2^2 + 2\phi_3^2 + \phi_4^2 - 5\phi_1 - 5\phi_2 - 21\phi_3 + 7\phi_4$$

subject to

$$-\phi_1^2 - \phi_2^2 - \phi_3^2 - \phi_4^2 - \phi_1 + \phi_2 - \phi_3 + \phi_4 + 8 \geq 0,$$

$$-\phi_1^2 - 2\phi_2^2 - \phi_3^2 - 2\phi_4^2 + \phi_1 + \phi_4 + 10 \geq 0,$$

$$-2\phi_1^2 - \phi_2^2 - \phi_3^2 - 2\phi_1 + \phi_2 + \phi_4 + 5 \geq 0.$$

The function has a minimum of -44 at  $\phi_1 = 0$ ,  $\phi_2 = 1$ ,  $\phi_3 = 2$ ,  $\phi_4 = -1$ .

Suggested starting point:  $\phi_1 = 0$ ,  $\phi_2 = 0$ ,  $\phi_3 = 0$ ,  $\phi_4 = 0$ . This problem is referred to as the Rosen-Suzuki problem.

[Source: Rosen and Suzuki (1965). See also Kowalik and Osborne (1968).]

Question 91 Minimize w.r.t.  $\phi$

$$f = 9 - 8\phi_1 - 6\phi_2 - 4\phi_3 + 2\phi_1^2 + 2\phi_2^2 + \phi_3^2 + 2\phi_1\phi_2 + 2\phi_1\phi_3$$

subject to

$$\phi_i \geq 0, i = 1, 2, 3,$$

$$3 - \phi_1 - \phi_2 - 2\phi_3 \geq 0.$$

The function has a minimum of 1/9 at  $\phi_1 = 4/3$ ,  $\phi_2 = 7/9$ ,  $\phi_3 = 4/9$ .

Suggested starting points: (a)  $\phi_1 = 1$ ,  $\phi_2 = 2$ ,  $\phi_3 = 1$ ; (b)  $\phi_1 = \phi_2 = \phi_3 = 1$ ; (c)  $\phi_1 = \phi_2 = \phi_3 = 0.5$ ; (d)  $\phi_1 = \phi_2 = \phi_3 = 0.1$ . This problem is referred to as the Beale problem.

[Source: Beale (1967). See also Kowalik and Osborne (1968).]

Question 92 Minimize w.r.t.  $\phi$  the maximum of

$$f_1 = \phi_1^4 + \phi_2^2,$$

$$f_2 = (2-\phi_1)^2 + (2-\phi_2)^2,$$

$$f_3 = 2\exp(-\phi_1 + \phi_2).$$

The minimax solution occurs at  $\phi_1 = \phi_2 = 1$ , where  $f_1 = f_2 = f_3 = 2$ .

Suggested starting point:  $\phi_1 = \phi_2 = 2$ .

[Source: Charalambous (1973).]

Question 93 Minimize w.r.t.  $\phi$  the maximum of

$$\begin{aligned} f_1 &= \phi_1^2 + \phi_2^4, \\ f_2 &= (2-\phi_1)^2 + (2-\phi_2)^2, \\ f_3 &= 2\exp(-\phi_1 + \phi_2). \end{aligned}$$

The minimax solution occurs at

$$\phi_1 = 1.13904, \quad \phi_2 = 0.89956,$$

where

$$\begin{aligned} f_1 &= f_2 = 1.95222, \\ f_3 &= 1.57408. \end{aligned}$$

Suggested starting point:  $\phi_1 = \phi_2 = 2$ .

[Source: Charalambous (1973).]

Question 94

- (a) Formulate the design of a notch filter in terms of inequality constraints, given the following requirements. The attenuation should not exceed  $A_1$  dB over the range 0 to  $\omega_1$ , and  $A_2$  dB over the range  $\omega_2$  to  $\omega_3$ , with  $0 < \omega_1 < \omega_2 < \omega_3$ . At  $\omega_0$ , where  $\omega_1 < \omega_0 < \omega_2$ , the attenuation must exceed  $A_0$  dB.
- (b) Describe very briefly and illustrate the Sequential Unconstrained Minimization Technique (Fiacco-McCormick method) for constrained optimization.
- (c) Set up a suitable objective function for the optimization of the notch filter of (a).

Question 95 Write down explicitly the generalized least pth objective function comprising real functions  $f_i$  (not necessarily positive) of  $\phi$ , level  $\xi$ , maximum  $M$ , multipliers  $u_i$  and any other necessary symbols. Ensure that  $M > 0$ ,  $M = 0$  and  $M < 0$  are included in your description.

Question 96 Derive the gradient vector of the generalized least pth objective of Question 95 and discuss its features.

Question 97 Derive necessary conditions for a minimax optimum from the gradient vector of the least pth objective of Question 96, where the  $f_i$  are assumed differentiable functions of  $\phi$ .

Question 98 Fit  $f = \phi_1\psi + \phi_2$  to  $S(\psi)$ , where  $\psi_1 = 1$ ,  $\psi_2 = 2$ ,  $\psi_3 = 3$ ,  $\psi_4 = 4$ ,  $S(\psi_1) = 1$ ,  $S(\psi_2) = 1$ ,  $S(\psi_3) = 1.5$ ,  $S(\psi_4) = 1$ , using a program for least pth approximation. Consider  $p = 1, 2$  and  $\infty$  with uniform weighting to all errors.

Question 99 Solve analytically the problems described in Question 98 invoking optimality conditions.

Question 100 Consider the functions  $e_1$  and  $e_2$  of one variable  $\phi$  shown in Fig. 13. Explain the implications of least pth approximation with  $p = 1$  and  $2$ , minimax approximation and simultaneous minimization of  $|e_1|$  and  $|e_2|$  w.r.t.  $\phi$ .

Question 101 Consider the functions  $f_1$  and  $f_2$  of one variable  $\phi$  shown in Fig. 14. Explain the implications of generalized least  $p$ th optimization of  $f_1$  and  $f_2$  w.r.t.  $\phi$  for  $p > 0$ .

Question 102 Consider the two functions of one variable

$$e_1 = -\phi + 4$$

$$e_2 = \phi/3$$

Expose and explain the distinctive features and implications of

- (a) the least  $p$ th approximation with  $p = 1$  and  $p = 2$  of  $|e_1|$  and  $|e_2|$  w.r.t.  $\phi$ ,
- (b) the minimax optimization of  $|e_1|$  and  $|e_2|$  w.r.t.  $\phi$ ,
- (c) the simultaneous minimization of  $|e_1|$  and  $|e_2|$  w.r.t.  $\phi$ .

Question 103 Consider a transfer function of a filter as

$$H(j\omega) = \frac{1}{(j\omega - \alpha_1)(j\omega - \alpha_2)(j\omega - \alpha_3)}$$

All  $\alpha_i$  are real variables which are adjusted to satisfy given specifications for the filter gain and  $j = \sqrt{-1}$ . Filter gain  $G(\omega)$  is defined by

$$G(\omega) = -20 \log |H(j\omega)|$$

and specifications  $S(\omega)$  are

$$S(\omega) \leq 1 \text{ dB for } 0 \leq \omega \leq 1$$

$$S(\omega) \geq 40 \text{ dB for } \omega \geq 5$$

Formulate the optimization problem in a form suitable for programming with specific relevance to an available package you are familiar with.



Question 104 Suppose that the following table has been derived from impedance measurements at four frequencies.

frequency (rad/s)	real part ( $\Omega$ )	imaginary part ( $\Omega$ )
1	1.9	1.6
2	2.1	2.9
3	4.5	2.0
4	2.0	6.0

Obtain a uniformly weighted least pth approximation based on real approximating functions for (a)  $p = 1$ , (b)  $p = 2$ , and (c)  $p = \infty$ , for a proposed series RL circuit model with resistance  $R$  and inductance  $L$  as independent unknowns. Consider error functions of the form  $|R - S_R|$ ,  $|L - S_L|$ . Comment on the data in the table and on your solutions.

Question 105 Set up as a nonlinear program the problem of least pth optimization with  $p = 1$  given by

$$\min_{\phi} \sum_{i=1}^n |e_i(\phi)|,$$

where the  $e_i$  are real functions of  $\phi$ . State necessary conditions for optimality of the problem and discuss them. Apply these ideas to

(a)  $\min_{\phi} |\phi - 1| + |\phi|,$

(b)  $\min_{\phi_1, \phi_2} |\phi_1 + \phi_2 - 1| + |\phi_1| + |\phi_2|.$

Question 106 Optimize the LC lowpass filter shown in Fig. 15. Write all necessary subprograms to calculate the response and its sensitivities. Verify your results with an available analysis program.

Specifications	
Frequency Range (rad/s)	Insertion Loss (dB)
0 - 1	< 1.5
> 2.5	> 25

Question 107 Consider the following specification for a transient response of a linear system:

$$S(t) = \begin{cases} 5t, & 0 \leq t \leq 0.2 \\ -1.25t + 1.25, & 0.2 \leq t \leq 1 \\ 0, & t \geq 1 \end{cases}$$

Optimize the impulse response of the LC circuit of Question 106 to fit this specification in the least squares sense.

Question 108 Consider the linear circuit shown in Fig. 16, which is assumed to be in the sinusoidal steady state. Let  $R = 2\Omega$ ,  $C = 1F$ ,  $\omega = 2$  rad/s.

(a) Obtain by direct differentiation simplified formulas for  $\frac{\partial V_R}{\partial C}$ ,  $\frac{\partial V_R}{\partial R}$

and  $\frac{\partial V_R}{\partial \omega}$ .

(b) Obtain the formulas of (a) by the adjoint network method from first principles.

Question 109 Consider the linear circuit shown in Fig. 17, which is assumed to be in the sinusoidal steady state. Let  $V_g = 1V$ ,  $R_g = 0.5\Omega$ ,  $C = 2F$ ,  $R = 1\Omega$ ,  $\omega = 10$  rad/s.

Use the adjoint network approach to evaluate  $\partial V_R / \partial C$ ,  $\partial V_R / \partial R$  and  $\partial V_R / \partial \omega$ . Estimate the change in  $V_R$  when both  $C$  and  $R$  decrease by 5% using these partial derivatives and compare with the exact change. How would you conduct a worst-case tolerance analysis, in general?

Question 110 Consider the circuit shown in Fig. 18, which is assumed to be in the sinusoidal steady state.

Derive from first principles the adjoint network and sensitivity expressions for all the elements of the circuit. Derive the adjoint excitations appropriate for calculating the first-order sensitivities of  $V_{C_2}$  w.r.t. all the parameters.

Question 111 Derive the first-order sensitivity expression

$$-\underline{V}^T \Delta \underline{Y}^T \hat{\underline{V}}$$

for linear time-invariant networks in the frequency domain, where  $\underline{Y}$  is the s.c. admittance matrix of an element,  $\underline{V}$  the voltage vector in the original network and  $\hat{\underline{V}}$  the corresponding vector in the adjoint network of the element under consideration.

Question 112 Derive from first principles an approach to finding  $\partial y_i / \partial x_j$ , where  $\underline{A} \underline{y} = \underline{b}$  is a linear system in  $\underline{y}$ ,  $\underline{A}$  is a square matrix whose coefficients are nonlinear functions of  $\underline{x}$ , the term  $y_i$  is the  $i$ th component of the column vector  $\underline{y}$  and  $\partial y_i / \partial x_j$  represents a column vector containing partial derivatives of  $y_i$  w.r.t. corresponding elements of the column vector  $\underline{x}$ . Discuss the computational effort involved.

Question 113 Derive from first principles an approach to finding  $\partial V_i / \partial \omega$ , where  $\omega$  is frequency,  $V_i$  is an  $i$ th nodal voltage in the nodal equation of a linear, time-invariant circuit in the frequency domain, namely,

$$\underline{Y} \underline{V} = \underline{I},$$

assuming  $\underline{I}$  is independent of  $\omega$ .

Question 114 Consider the system of complex linear equations

$$\underline{Y} \underline{V} = \underline{I},$$

where  $\underline{Y}$  is a square nodal admittance matrix of constant, complex coefficients, and  $\underline{I}$  is a specified excitation vector. Set up the appropriate objective function for the least squares solution of this system of equations and derive the gradient vector w.r.t. the real and imaginary parts of the components of  $\underline{V}$ .

Question 115 Derive an approach to calculating  $\partial y / \partial x_i$ , where  $\underline{A} \underline{y} = \underline{b}$  is a linear system in  $\underline{y}$ ,  $\underline{A}$  is a square matrix whose coefficients are nonlinear functions of  $\underline{x}$  and  $x_i$  is the  $i$ th component of  $\underline{x}$ . Discuss the computational effort involved.

Question 116 Derive from first principles an approach to calculating

$$\frac{\partial^2 y_i}{\partial x_j \partial x_k}$$

for the system described in Question 112, where  $x_j$  and  $x_k$  are elements of the vector  $\underline{x}$ .

Question 117 Derive from first principles an approach to finding  $\partial\lambda/\partial\tilde{x}$ , where  $\lambda$  is an eigenvalue of the square matrix  $\tilde{A}$  whose coefficients are (in general) nonlinear functions of  $\tilde{x}$ , i.e.,

$$\tilde{A}\tilde{y} = \lambda\tilde{y}.$$

The expression  $\partial\lambda/\partial\tilde{x}$  is a column vector containing all first partial derivatives of  $\lambda$  w.r.t. corresponding elements of the column vector  $\tilde{x}$ . Discuss the computational effort involved. Give interpretations of any new symbols introduced. [Hint:  $\lambda$  is also an eigenvalue of  $\tilde{A}^T$ .]

Question 118 Derive an approach to calculating

$$\frac{\partial^2\lambda}{\partial x_j \partial x_k}$$

for the system described in Question 117, where  $x_j$  and  $x_k$  are elements of the vector  $\tilde{x}$ .

Question 119 Consider the quadratic approximation to a response function given by

$$f(\phi, \psi) = \frac{1}{2} [\phi^T \ \psi] \begin{bmatrix} \tilde{A} & \tilde{a} \\ \tilde{a}^T & \tilde{a} \end{bmatrix} \begin{bmatrix} \phi \\ \psi \end{bmatrix} + [\phi^T \ \psi] \begin{bmatrix} \tilde{b} \\ \tilde{b} \end{bmatrix} + c,$$

where  $\tilde{A}$  is a symmetric square matrix of the dimensions of the column vector  $\phi$ ;  $\tilde{a}$  and  $\tilde{b}$  are column vectors of constants of the same dimension as  $\phi$ ; and  $a$ ,  $b$  and  $c$  are constants. Develop a compact expression for  $f(\phi, \psi)$  subjected to the condition

$$\frac{\partial f}{\partial \psi} = 0.$$

Question 120 Develop from first principles a computationally attractive method of obtaining the Thevenin equivalent of an arbitrary linear,

time-invariant circuit in the frequency domain using only one analysis of a suitable circuit. [Hint: Show that this circuit is the adjoint of the given circuit and derive the appropriate terminations and all necessary formulas.]

Question 121 Derive from first principles the sensitivity expression and adjoint element corresponding to a voltage controlled current source. Draw circuit diagrams to fully illustrate your results.

Question 122 Derive from first principles the first-order sensitivity expressions relating to:

- (a) a voltage controlled voltage source,
- (b) a current controlled voltage source,
- (c) an open-circuited uniformly distributed line,
- (d) a uniform RC line.

Question 123 Derive from first principles the adjoint element equation and sensitivity expression for a two-port characterized by

$$\begin{bmatrix} V_p \\ I_p \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix}$$

Apply the result to the element shown in Fig. 19.

Question 124 Verify that the adjoint network may be characterized by the hybrid matrix description

$$\begin{bmatrix} \hat{I}_a \\ \hat{V}_b \end{bmatrix} = \begin{bmatrix} \tilde{Y}^T & -\tilde{M}^T \\ -\tilde{A}^T & \tilde{Z}^T \end{bmatrix} \begin{bmatrix} \hat{V}_a \\ \hat{I}_b \end{bmatrix},$$

where the corresponding description for the original network is

$$\begin{bmatrix} I_a \\ V_b \end{bmatrix} = \begin{bmatrix} \tilde{Y} & \tilde{A} \\ \tilde{M} & \tilde{Z} \end{bmatrix} \begin{bmatrix} V_a \\ I_b \end{bmatrix}.$$

Question 125 Verify that, for a network excited by a set of independent voltages  $J_V$  and a set of independent currents  $J_I$ ,

$$\tilde{G} = \sum_{i \in J_V} \hat{V}_i \tilde{V}_i - \sum_{i \in J_I} \hat{I}_i \tilde{V}_i,$$

where

$$\tilde{V} \triangleq \begin{bmatrix} \frac{\partial}{\partial \phi_1} \\ \frac{\partial}{\partial \phi_2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial}{\partial \phi_k} \end{bmatrix}$$

implies differentiation w.r.t.  $k$  parameters  $\phi_1, \phi_2, \dots, \phi_k$  and  $\tilde{G}$  is a vector of corresponding sensitivity expressions associated with elements of the network. The remaining variables  $V_i, I_i, \hat{V}_i$  and  $\hat{I}_i$  are associated with excitations and responses in the original network and adjoint network as implied by Fig. 20.

Question 126 Consider the linear circuit shown in Fig. 21 excited by a unit step  $u(t)$ . Obtain  $\partial v/\partial R$  and  $\partial v/\partial C$  using the adjoint network method and verify the resulting formulas by directly differentiating  $v(t)$ .

Question 127 Evaluate at 0.5 rad/s the partial derivatives of the input impedance (see Fig. 22) w.r.t. the inductors and capacitors of the filter of Question 32.

Question 128 Consider the circuit of Question 106 at  $\omega = 1$  rad/s. Let  $L_1 = L_2 = 2H$ ,  $C = 1F$ . Obtain the partial derivative values of the insertion loss in dB of the filter between the terminating resistors with respect to  $L_1$ ,  $C$  and  $L_2$  using the adjoint network method. If  $L_1$  changes by +5%,  $L_2$  by -5% and  $C$  by +10%, estimate the change in insertion loss at  $\omega = 1$  rad/s. Check your results by calculating the change in loss directly and explain any discrepancies.

Question 129 Derive from first principles an approach to finding the exact large change  $\Delta y_i$  due to the large change  $\Delta a_{jj}$ , where  $\underline{A} \underline{y} = \underline{b}$  is a linear system in  $\underline{y}$ ,  $\underline{A}$  is a square matrix, the term  $y_i$  is the  $i$ th component of the column vector  $\underline{y}$  and  $a_{jj}$  represents the  $j$ th diagonal element of  $\underline{A}$ . Discuss the computational effort involved. [Hint: First find  $\Delta y_j$ .]

Question 130 Consider the resistive network of Fig. 23.

- (a) Calculate the node voltages by LU factorization of the nodal admittance matrix showing all major steps. Verify that



$$\tilde{L} = \begin{bmatrix} 3 & 0 & 0 \\ -2 & 11/3 & 0 \\ 0 & -2 & 21/11 \end{bmatrix},$$

$$\tilde{U} = \begin{bmatrix} 1 & -2/3 & 0 \\ 0 & 1 & -6/11 \\ 0 & 0 & 1 \end{bmatrix}.$$

- (b) Draw the adjoint circuit appropriately excited with a unit current for finding the first-order sensitivities of the voltage  $V$  across  $G_3$ .
- (c) Calculate the node voltages of the adjoint circuit using the LU factors already obtained above.
- (d) Calculate  $\tilde{\nabla}V$ , where

$$\tilde{\nabla} = \begin{bmatrix} \partial/\partial G_1 \\ \partial/\partial R_2 \\ \partial/\partial G_3 \\ \partial/\partial R_4 \\ \partial/\partial G_5 \end{bmatrix}$$

using sensitivity formulas shown in the table.

Element	Branch Equation		Sensitivity	Parameters
	Original	Adjoint		
Resistor	$V = RI$	$\hat{V} = R\hat{I}$	$\hat{I}\hat{I}$	R
	$I = GV$	$\hat{I} = G\hat{V}$	$-\hat{V}\hat{V}$	G

Question 131 Consider the resistive network of Question 27.

- (a) Calculate the LU factors of the nodal admittance matrix.
- (b) Calculate using Tellegen's theorem (unperturbed) the Thevenin equivalent of the network as seen by the element  $G_3$ . Proceed as follows. You need
- (i) the open circuit voltage  $V_{TH}$  seen by  $G_3$ ,
  - (ii) the impedance  $Z_{TH}$  seen by  $G_3$  with  $I_g = 0$ .
- Prove that one adjoint network analysis can be used for both quantities, draw the appropriate excited adjoint network, and solve it using the LU factors of (a).
- (c) Calculate using your Thevenin equivalent the change in voltage across  $G_3$  when  $G_3$  increases from 1 mho to 2 mho. Now represent this change by an independent current source applied across  $G_3$ .
- (d) Hence, find the voltage across  $G_5$  due to the specified change in  $G_3$  using the LU factors obtained in (a).
- (e) Check by a direct method that your result in (d) is correct.

Question 132 Draw the adjoint network for the active circuit shown in Fig. 24, which is assumed to be in the sinusoidal steady state. Include excitations appropriate to calculating the sensitivities of  $V_2(j\omega)$  w.r.t. all parameters, clearly identifying zero and nonzero excitations. Develop an expression for the gradient vector of the following objective function to be minimized:

$$U = \sum_{i=1}^n (G(\omega_i) - S(\omega_i))^2,$$

where

$$G(\omega) = \left| \frac{V_2(j\omega)}{V_0(j\omega)} \right|^2$$

and  $S(\omega)$  is a given specification.

Element	Equation		Sensitivity Parameters	
	Original	Adjoint		
Resistor	$V = RI$	$\hat{V} = R\hat{I}$	$\hat{I}$	$R$
Capacitor	$I = j\omega CV$	$\hat{I} = j\omega C\hat{V}$	$-j\omega\hat{V}$	$C$
Voltage Controlled Source	$I_1 = 0 \quad 0 \quad V_1$ $V_2 = \mu \quad 0 \quad I_2$	$\hat{I}_1 = 0 \quad -\mu \quad \hat{V}_1$ $\hat{V}_2 = 0 \quad 0 \quad \hat{I}_2$	$V_1 \hat{I}_2$	$\mu$

**Question 133** Consider the circuit of Question 29, which is assumed to be in the sinusoidal steady state.

Let  $V_1 = 1V$ ,  $\omega = 2 \text{ rad/s}$ ,  $R_1 = R_2 = R_3 = 2\Omega$ ,  $C_1 = C_2 = C_3 = 1F$ .

- Write down the nodal equations for the circuit, using the component values and frequency indicated.
- Apply Gauss-Seidel (relaxation) method to find the node voltages, assuming the initial node voltages to be zero. Use two iterations. Repeat with an overrelaxation factor of 1.5.
- Factorize the nodal admittance matrix into upper and lower triangular form.
- Calculate  $\partial V_3 / \partial C_2$  and  $\partial V_3 / \partial R_1$  by the adjoint network method using the above LU factorization results in conjunction with the nodal admittance matrix of the adjoint circuit.
- Estimate  $\Delta V_3$  (the total change in  $V_3$ ) when  $C_2$  changes by +3% and  $R_1$

by -5%. Use  $\Delta V_3 \approx \frac{\partial V_3}{\partial C_2} \Delta C_2 + \frac{\partial V_3}{\partial R_1} \Delta R_1$ . Check the results by direct perturbation.

Question 134 Compare the computational effort in the ABCD or chain matrix analysis of a network and an efficient method based on a tridiagonal nodal admittance matrix.

Question 135 Discuss carefully the computational effort required in general for each approach used in Question 133.

Question 136 Write an efficient Fortran program using LU factorization in conjunction with Newton's method for solving nonlinear equations to find the node voltages of the resistor-diode network shown in Fig. 25 [Source: Chua and Lin (1975)], where

$$i_d = I_S (e^{\lambda v_d} - 1) ,$$

$$I_S = 10^{-12} \text{ mA} ,$$

$$\lambda = 1/V = 1/0.026 \text{ V}^{-1} ,$$

$$E = 10 \text{ V} ,$$

$$R_1 = R_2 = 1 \text{ k}\Omega .$$

Use the results to calculate

$$\begin{bmatrix} \frac{\partial V_3}{\partial R_1} \\ \frac{\partial V_3}{\partial R_2} \end{bmatrix}$$

subject to satisfying the nonlinear equations.

By running the program again with small perturbations in  $R_1$  and  $R_2$ , check these derivatives. Solve the equations for a number of starting points and comment on the results. Also use

$$v_1 = 5.75 \quad v_2 = 0.75 \quad v_3 = 5.0$$

as a test starting point.

Question 137 What is the companion network method of solving nonlinear networks? How does it take advantage of existing linear network simulation methods? Provide an illustrative example.

Question 138 Consider the resistor-diode network shown in Question 136. Draw the corresponding companion network at the  $j$ th iteration for its d.c. solution. Write down the nodal equations at this iteration.

Question 139 Consider the resistor-diode network shown in Question 136. Develop the system of linear equations derived from the nodal equations at the  $j$ th iteration for solution by the Newton method. Write down explicitly the Jacobian at the  $j$ th iteration.

Question 140 Consider the nonlinear circuit shown in Fig. 26, where  $i_a = 2v_a^3$ ,  $i_b = v_b^3 + 10v_b$ .

- (a) Express the nodal equations in the linearized form required at the  $j$ th iteration of the Newton algorithm.
- (b) Apply two iterations of the Newton method, starting at  $v_1 = 2$ ,  $v_2 = 1$ .

(c) Draw the companion network at the  $j$ th iteration and state the corresponding nodal equations.

(d) Continue with two iterations of the companion network method.

[Source: Chua and Lin (1975).]

Question 141 Consider least  $p$ th optimization with both upper and lower response specifications, where the specifications might be violated or satisfied. Discuss in as much detail as possible the role of the value of  $p$  and the effects of different weightings on the solution.

Question 142 Show, using the generalized least  $p$ th objective, that if specifications cannot be satisfied with a given value of  $p \geq 1$ , then they cannot be satisfied for any other value, e.g.,  $p = \infty$ .

Question 143 Set up and discuss a suitable least  $p$ th objective function for approximate minimization of

$$\max_{i \in I} f_i(\underline{\phi})$$

where  $\underline{\phi}$  contains the adjustable parameters and  $I$  denotes an index set relating to the differentiable nonlinear functions  $f_i$ , which are not necessarily positive.

Question 144 Relate the problem formulation of Question 143 to filter design, taking care to discuss upper and lower response specifications, errors and weighting functions.

Question 145 Derive the Golden Section search method for functions of one variable from first principles. Explain all the concepts involved.

Under what conditions would you expect a global solution?

Question 146 Apply 3 iterations of the Golden Section search method to the function of one variable given shown in Fig. 27. Show clearly all steps and label the diagrams appropriately. Fit a quadratic function to 3 points corresponding to the lowest function values observed and find its minimum. Estimate function values and points from the graph.

Question 147 Starting with the interval  $[0,6]$ , apply 4 iterations of the Golden Section search method to the minimization w.r.t.  $\phi$  of a function described by

$$\begin{aligned} U &= -\phi + 5 & \phi &\leq 1 \\ U &= 0.5(\phi - 3)^2 + 1 & 1 &< \phi \leq 4 \\ U &= 3 - (\phi - 6)^2/3 & \phi &> 4 \end{aligned}$$

What is the solution obtained? By how much has the interval of uncertainty been reduced?

Question 148 Devise an algorithm for finding the extrema of a well-behaved multimodal function of one variable.

Question 149 Discuss mathematically and physically the concept of steepest descent for  $\max_{1 \leq i \leq n} f_i(\phi)$ , where the  $f_i(\phi)$  are  $n$  real, nonlinear, differentiable functions of  $\phi$ .

Question 150 Suppose we have to minimize

$$(a) \quad U = \left( \sum_{\omega_i \in \Omega_d} |L(\omega_i) - S(\omega_i)|^p \right)^{1/p}, \quad p > 1.$$

$$(b) \quad U = \sum_{\omega_i \in \Omega_d} [L(\omega_i) - S(\omega_i)]^p, \quad p \text{ even} > 0.$$

where the  $L(\omega_i)$  is the insertion loss in dB of a filter between  $R_g$  and  $R_L$ ,  $S(\omega_i)$  is the desired insertion loss between  $R_g$  and  $R_L$  and  $\Omega_d$  is a set of discrete frequencies  $\omega_i$ . Obtain expressions relating  $\nabla U$  to  $\tilde{G}(j\omega_i)$ , where the elements of  $\tilde{G}$  are appropriate adjoint sensitivity expressions. Assume convenient values for the excitations of the original and adjoint networks.

Question 151 The complex impedance of a body has been measured at a set of frequencies. A linear circuit model to represent this impedance is proposed. Explain the steps you would take to optimize the model, assuming you were to use an available unconstrained optimization program requiring first derivatives.

Question 152 Describe the aims of the project you are carrying out for this course. Explain in detail the steps you are taking to meet these aims. What results have you obtained thus far and are they what you expected?

Question 153 Consider the circuit shown in Fig. 28, which is a linear time-invariant network with parameters  $\phi$ . It is desired to obtain the best impedance match between the complex, frequency-dependent load  $Z_L$  and the constant source resistance  $R_g$ .

Formulate a least squares objective function  $U$  of the parameter vector  $\phi$ , the optimum of which represents a good match over a band of frequencies  $\Omega$ . Explain carefully and in detail how the adjoint network



method may be used to calculate the gradient vector  $\nabla U(\phi)$ .

Question 154 Consider the voltage divider shown in Fig. 29. The specifications are as follows.

$$0.46 \leq \frac{R_2}{R_1 + R_2} \leq 0.53 ,$$

$$1.85 \leq R_1 + R_2 \leq 2.15 .$$

Assuming  $R_1 \geq 0$ ,  $R_2 \geq 0$ , derive the worst vertices of a tolerance region for independent tolerance assignment on these two components.

[Reference: Karafin, BSTJ, vol. 50, 1971, pp. 1225-1242.]

Question 155 Consider the problem defined in Question 154. Optimize the tolerances  $\varepsilon_1$  and  $\varepsilon_2$  on  $R_1$  and  $R_2$  given the cost function

$$C = \frac{R_1^0}{\varepsilon_1} + \frac{R_2^0}{\varepsilon_2}$$

assuming an environmental (uncontrollable) parameter  $T$  common to both resistors such that

$$R_1 = (R_1^0 + \mu_1 \varepsilon_1) (T^0 + \mu_t \varepsilon_t) ,$$

$$R_2 = (R_2^0 + \mu_2 \varepsilon_2) (T^0 + \mu_t \varepsilon_t) ,$$

where

$$-1 \leq \mu_1, \mu_2, \mu_t \leq 1 ,$$

$$T^0 = 1, \varepsilon_t = 0.05 .$$

[The independent designable variables include  $R_1^0$ ,  $R_2^0$ ,  $\varepsilon_1$  and  $\varepsilon_2$ .]

Question 156 Consider the problem defined in Question 154. Optimize the tolerance  $\epsilon_1$  on  $R_1$  given the cost function

$$C = \frac{R_1^0}{\epsilon_1}$$

assuming that  $R_2$  is tunable by  $\pm 10\%$  of its nominal value. [The independent designable variables include  $R_1^0$ ,  $\epsilon_1$  and  $R_2^0$ .]

Question 157 Consider the voltage divider shown in Fig. 30 with a nonideal source and load.

It is desired to maintain

$$0.47 \leq V \leq 0.53 ,$$

$$1.85 \leq R \leq 2.15 ,$$

for all possible

$$R_g \leq 0.01 ,$$

$$R_L \geq 100 ,$$

with

$$R_1^0 = R_2^0 ,$$

$$\epsilon_1 = \epsilon_2 ,$$

and maximum tolerances. Find the optimal values for  $R_1^0$ ,  $R_2^0$ ,  $\epsilon_1$  and  $\epsilon_2$ .

Question 158 Consider the voltage divider shown in Question 154. Formulate as precisely as possible the functions involved (objective and constraints) and their first partial derivatives required to optimize the tolerances on  $R_1$  and  $R_2$ , allowing the nominal point to move, subject to lower and upper limits on the transfer function and input resistance. Assume a worst-case solution is desired, and suggest cost functions.

Question 159 Consider the voltage divider shown in Question 154. Deriving all formulas from first principles, use the adjoint network method to calculate  $\partial T/\partial R_1$  and  $\partial T/\partial R_2$  given:

$$T \triangleq \frac{V_2}{V_1}, \quad R_1 = 1.1 \, \Omega, \quad R_2 = 0.9 \, \Omega.$$

Show both original and adjoint networks appropriately excited and verify your result by direct differentiation.

Question 160 Consider the voltage divider of Question 154 expressed as a minimax problem. Determine suitable active functions when

$$R_1 = 1.01$$

$$R_2 = 1.14$$

and calculate the steepest descent direction from first principles. Assume that if  $|M - f_i| < 0.01$  for any  $f_i$ , then the corresponding  $f_i$  is active, where  $M \triangleq \max_i f_i$ . Show all steps in your calculations.

Question 161 Consider an acceptable region given by

$$2 + 2\phi_1 - \phi_2 \geq 0,$$

$$143 - 11\phi_1 - 13\phi_2 \geq 0,$$

$$-60 + 4\phi_1 + 15\phi_2 \geq 0.$$

Determine optimally centered, optimally toleranced solutions using the following cost functions:

$$(a) \quad \frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2},$$

$$(b) \quad \log_e \frac{\phi_1^0}{\varepsilon_1} + \log_e \frac{\phi_2^0}{\varepsilon_2},$$

where  $\varepsilon_1$  and  $\varepsilon_2$  are tolerances and  $\phi_1^0$  and  $\phi_2^0$  are nominal values.

Formulate the problem as a nonlinear programming problem and give expressions for derivatives.

Question 162 Consider the voltage divider shown in Question 154 subject to the same specifications. Optimize the tolerances  $\epsilon_1$  and  $\epsilon_2$  on  $R_1$  and  $R_2$ , respectively, and find the best corresponding nominal values  $R_1^0$  and  $R_2^0$ , using the following cost functions:

$$(a) \quad C_1 = \frac{R_1^0}{\epsilon_1} + \frac{R_2^0}{\epsilon_2} ,$$

$$(b) \quad C_2 = \frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} .$$

[Source: Karafin, BSTJ, vol. 50, 1971, pp. 1225-1242.]

Question 163 Find the number of state variables and indicate a possible choice of these states for the circuit shown in Fig. 31.

Question 164 The circuit shown in Fig. 32 has the state equations

$$C_D \frac{dv_D}{dt} = -I_S (e^{\lambda v_D} - 1) + (E_1 - E_2 - v_D)/R_1 + (v_0 - E_2 - v_D)/R_2$$

$$C_0 \frac{dv_0}{dt} = (E_2 + v_D - v_0)/R_2$$

The parameters are

$$R_1 = R_2 = 1 \text{ k}\Omega$$

$$I_D = I_S (e^{\lambda v_D} - 1), \quad \lambda = 40 \text{ V}^{-1}, \quad I_S = 10^{-10} \text{ A}$$

$$C_1 = 1 \text{ }\mu\text{F}, \quad C_2 = 10 \text{ pF}$$

$$E_2 = 1 \text{ V}$$

Perform two steps of fourth-order Runge-Kutta integration starting at  $t=0$ ,  $v_D(0) = v_0(0) = 0$  and using a time step of 10 ns.

[Source: Chua and Lin (1975).]

Question 165 Describe briefly the principle behind the Runge-Kutta algorithms for solving a differential equation with a given initial value. Consider the following initial value problem

$$\dot{x} = (\cos x) + t \quad x_0 = 1, t \in [0, 0.3]$$

A solution is required for a step-size of 0.1.

- (a) Use Heun's algorithm.
- (b) Use the fourth-order Runge-Kutta method.

Question 166 Approximate in a uniformly weighted minimax sense

$$f(x) = x^2$$

by

$$F(x) = a_1 x + a_2 \exp(x)$$

on the interval  $[0, 2]$ .

[Source: Curtis and Powell (1965). See also Popovic, Bandler and Charalambous (1974).]

Question 167 Approximate in a uniformly weighted minimax sense

$$f(x) = \frac{[(8x - 1)^2 + 1]^{0.5} \tan^{-1}(8x)}{8x}$$

by

$$F(x) = \frac{a_0 + a_1 x + a_2 x^2}{1 + b_1 x + b_2 x^2}$$

on the interval  $[-1, 1]$ .

[Reference: Popovic, Bandler and Charalambous (1974).]

Question 168 Consider a lumped-element LC transformer (Fig. 33) to match a 1 ohm load to a 3 ohm generator over the range 0.5 - 1.179 rad/s. A minimax approximation should be carried out on the modulus of the reflection coefficient using all six reactive components as variables. The solution is

$$L_1 = 1.041,$$

$$C_2 = 0.979,$$

$$L_3 = 2.341,$$

$$C_4 = 0.781,$$

$$L_5 = 2.937,$$

$$C_6 = 0.347,$$

at which  $\max |p| = 0.075820$ . Use 21 uniformly spaced sample points in the band. Suggested starting point:

$$L_1 = C_2 = L_3 = C_4 = L_5 = C_6 = 1.$$

[Source: Hatley (1967). See also Srinivasan (1973). See Example 4 of Report SOS-78-14-U for hints in setting up the subprograms.]

Question 169 Consider the RC active equalizer shown in Fig. 34. The specified linear gain response in dB over the band 1 MHz to 2 MHz is given by  $G = 5 + 5f$ , where  $f$  is in MHz. Find optimal solutions using least pth approximation with  $p = 2, 4, 8, \dots, \infty$  taking as variables  $C_1, C_2, R_1$  and  $R_2$ . Twenty-one uniformly distributed sample points are suggested with starting values

$$C_1 = C_2 = R_1 = R_2 = 1$$

and

$$C_1 = C_2 = R_1 = R_2 = 0.5.$$

Comment on the results. *Take  $R=1$ .*

Reconsider the problem using only  $C_1$  and  $R_1$  as variables.

[Source: Temes and Zai (1969).]

Question 170 Consider the problem of finding a second-order model of a fourth-order system, when the input to the system is an impulse, in the minimax sense. The transfer function of the system is

$$G(s) = \frac{(s+4)}{(s+1)(s^2+4s+8)(s+5)}$$

and of the model is

$$H(s) = \frac{\phi_3}{(s+\phi_1)^2 + \phi_2^2} .$$

The problem is, therefore, equivalent to making the function

$$F(\phi, t) = \frac{\phi_3}{\phi_2} \exp(-\phi_1 t) \sin \phi_2 t$$

best approximate

$$S(t) = \frac{3}{20} \exp(-t) + \frac{1}{52} \exp(-5t) - \frac{\exp(-2t)}{65} (3\sin 2t + 11\cos 2t)$$

in the minimax sense. The problem may be discretized in the time interval 0 to 10 seconds and the function to be minimized is

$$\max_{i \in I} |e_i(\phi)| , \quad I = \{1, 2, \dots, 51\} ,$$

where

$$e_i(\phi) = F(\phi, t_i) - S(t_i) .$$

The solution is

$$\phi_1 = 0.68442,$$

$$\phi_2 = \pm 0.95409,$$

$$\phi_3 = 0.12286,$$

and the maximum error is  $7.9471 \times 10^{-3}$ . Suggested starting point:  $\phi_1 = \phi_2 = \phi_3 = 1$ .

[See, for example, Bandler (1977).]

Question 171 Develop a program to calculate and plot insertion loss of the circuit shown in Fig. 35 (elliptic low-pass filter).

Data for the circuit is

$$C_1 = 0.89318 \text{ F}$$

$$C_2 = 0.1022 \text{ F}$$

$$C_3 = 1.57677 \text{ F}$$

$$C_4 = 0.29139 \text{ F}$$

$$C_5 = 0.74177 \text{ F}$$

$$L_2 = 1.26033 \text{ H}$$

$$L_4 = 1.03950 \text{ H}$$

$$0 \leq \omega \leq 4 \text{ rad/s.}$$

What specifications does the circuit meet? Suggest ways of meeting these specifications by optimization assuming the solution was not known.

Question 172 Consider the LC filter of Question 106. The minimax solution corresponding to the specifications of Question 106, taking the passband sample points as 0.45, 0.5, 0.55, 1.0 and the stopband as 2.5, is

$$L_1 = L_2 = 1.6280, C = 1.0897.$$

Using appropriate optimization programs verify the worst-case tolerance solutions shown in the following table for the objective



$$\frac{L_1^0}{\varepsilon_1} + \frac{L_2^0}{\varepsilon_2} + \frac{C^0}{\varepsilon_C} .$$

Parameters	Continuous Solution		Discrete Solution		
	Fixed Nominal	Variable Nominal	from [1,2,5,10,15]%		
$\varepsilon_1/L_1^0$	3.5%	9.9%	5%	10%	10%
$\varepsilon_C/C^0$	3.2%	7.6%	10%	5%	10%
$\varepsilon_2/L_2^0$	3.5%	9.9%	10%	10%	5%
$L_1^0$	1.628	1.999	1.999		
$C^0$	1.090	0.906	0.906		
$L_2^0$	1.628	1.999	1.999		

[Source: Bandler, Liu and Chen (1975).]

Question 173 For the circuit of Question 172 verify numerically that the active worst-case vertices of the tolerance region are identified as in the table shown.

Vertex	Frequency
6	0.45, 0.50, 0.55
8	1.0
1	2.5

[Source: Bandler, Liu and Tromp (1976).]

Question 174 Consider the 10:1 impedance ratio, lossless two-section transmission-line transformer shown in Fig. 36. The lengths of the sections are  $l_1$  and  $l_2$ . The corresponding characteristic impedances are  $Z_1$  and  $Z_2$ . Minimize the maximum of the modulus of the reflection coefficient  $\rho$  over 100 percent relative bandwidth w.r.t. lengths and/or characteristic impedances. The known quarter-wave solution is given by

$$l_1 = l_2 = l_q \text{ (the quarter wavelength at centre frequency),}$$

$$Z_1 = 2.2361,$$

$$Z_2 = 4.4721,$$

where  $l_q = 7.49481$  cm for 1 GHz centre. The corresponding  $\max |\rho| = 0.42857$ .

Use 11 uniformly distributed (normalized frequency) sample points, namely 0.5, 0.6, ..., 1.5. Seven suggested starting points and problems are tabulated, namely, a, b, ..., g.

Parameters	Problem starting points						
	a	b	c	d	e	f	g
$l_1/l_q$		fixed (optimal)			0.8	1.2	1.2
$Z_1$	1.0	3.5	1.0	3.5	*	3.5	3.5
$l_2/l_q$		fixed (optimal)			1.2	*	0.8
$Z_2$	3.0	3.0	6.0	6.0	*	*	3.0

\* Parameter is fixed at optimal value.

A suggested specification, if appropriate to the method, is  $|\rho| \leq 0.5$ . A variation to the problem is to minimize the maximum of  $0.5 |\rho|^2$ . Suggested termination criterion:  $\max |\rho|$  within 0.01 percent of the

optimal value.

[Source: Bandler and Macdonald (1969).]

Question 175 Consider the problem described in Question 174. Using a computer plotting routine plot the contours

$$\{\max |\rho|\} = \{0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80\}$$

for the following situations:

- (a)  $1 \leq Z_1 \leq 3.5, 3 \leq Z_2 \leq 6,$
- (b)  $0.8 \leq \ell_1/\ell_q, \ell_2/\ell_q \leq 1.2,$
- (c)  $0.8 \leq \ell_1/\ell_q \leq 1.2, 1 \leq Z_1 \leq 3.5.$

Parameters not specified are held fixed at optimal values.

[Source: Bandler and Macdonald (1969).]

Question 176 Consider the problems described in Questions 174 and 175. Use a computer plotting routine to plot contours of a generalized least pth objective function for  $p = 1, 2, 10, \infty,$  taking  $|\rho|$  as the approximating function and 0.5 as the upper specification.

[Source: Bandler and Charalambous (1972).]

Question 177 Consider the same circuits, terminations and specifications as in Question 174. Let  $\epsilon_1$  and  $\epsilon_2$  be the tolerances on  $Z_1$  and  $Z_2,$  respectively. Starting at the known minimax solution with  $\epsilon_1 = 0.2$  and  $\epsilon_2 = 0.4$  minimize w.r.t.  $Z_1^0, Z_2^0, \epsilon_1$  and  $\epsilon_2$

$$(a) \quad C_1 = \frac{1}{\epsilon_1} + \frac{1}{\epsilon_2},$$

$$(b) \quad C_2 = \frac{Z_1^0}{\epsilon_1} + \frac{Z_2^0}{\epsilon_2},$$

for a worst-case design (yield = 100%).

[Source: Bandler, Liu and Chen (1975). See also Abdel-Malek (1977).]

Question 178 Consider the same circuit and terminations as in Question 174 but with three sections. The known quarter-wave solution is given by (see Question 174 for definition and value of  $l_q$ )

$$l_1 = l_2 = l_3 = l_q,$$

$$Z_1 = 1.63471,$$

$$Z_2 = 3.16228,$$

$$Z_3 = 6.11729.$$

The corresponding max  $|\rho| = 0.19729$ .

Use the 11 (normalized frequency) sample points 0.5, 0.6, 0.7, 0.77, 0.9, 1.0, 1.1, 1.23, 1.3, 1.4, 1.5. Three suggested starting points are tabulated, namely, a, b and c.

Parameters	Problem starting points		
	a	b	c
$l_1/l_q$	*	**	0.8
$Z_1$	1.0	1.0	1.5
$l_2/l_q$	*	**	1.2
$Z_2$	**	**	3.0
$l_3/l_q$	*	**	0.8
$Z_3$	10.0	10.0	6.0

\* Parameter is fixed at optimal value.

\*\* Parameter varies, starting at optimal value.

A variation to the problem is to minimize the maximum of  $0.5 |\rho|^2$ . Suggested termination criterion:  $\max |\rho|$  agrees with the optimal value to 5 significant figures.

[Source: Bandler and Macdonald (1969).]

Question 179 Design a recursive digital lowpass filter of the cascade form to best approximate a magnitude response of 1 in the passband, normalized frequency  $\psi$  of 0-0.09, and 0 in the stopband above  $\psi = 0.11$ . Take the transfer function as

$$H(z) = A \prod_{k=1}^K \frac{1 + a_k z^{-1} + b_k z^{-2}}{1 + c_k z^{-1} + d_k z^{-2}},$$

where  $K$  is the number of second-order sections,

$$z = \exp(j\psi\pi),$$

$$\psi = \frac{2f}{f_s},$$

$f$  is frequency and  $f_s$  is the sampling frequency.

Analytical derivatives w.r.t. the coefficients  $a_k$ ,  $b_k$ ,  $c_k$  and  $d_k$  are readily derived.

Suggested sample points  $\psi$  are

0.0 to 0.8 in steps of 0.01,

0.0801 to 0.09 in steps of 0.00045,

0.11 to 0.2 in steps of 0.01,

0.3 to 1.0 in steps of 0.1.

Use one section and a starting point of

$$\begin{aligned}a_1 &= 0, \\b_1 &= 0, \\c_1 &= 0, \\d_1 &= -0.25, \\A &= 0.1,\end{aligned}$$

for least pth approximation with  $p = 2, 10, 100, 1000, 10000$  and minimax approximation, each optimization starting at the solution to the previous one.

[See Bandler and Bardakjian (1973).]

Question 180 Grow a second section at the solution to Question 179 and reoptimize appropriately.

[See Bandler and Bardakjian (1973).]

Question 181 Optimize the coefficients of a recursive digital lowpass filter of the cascade form (see Question 179) to meet the following specifications:

$$0.9 \leq |H| \leq 1.1 \text{ in the passband,}$$

$$|H| \leq 0.1 \text{ in the stopband,}$$

where the passband sample points  $\psi$  are

$$0.0 \text{ to } 0.18 \text{ in steps of } 0.02,$$

and the stopband sample points  $\psi$  are

$$0.24,$$

$$0.3 \text{ to } 1.0 \text{ in steps of } 0.1.$$

Begin optimizing with one section starting at

$$\begin{aligned} a_1 &= 0, \\ b_1 &= 1, \\ c_1 &= -1, \\ d_1 &= 0.5, \\ A &= 0.1, \end{aligned}$$

for least pth approximation with  $p = 2, 10, 1000, 10000$  and minimax approximation, each optimization starting at the solution to the previous one.

[See Bandler and Bardakjian (1973).]

Question 182 Grow a second section at the solution to Question 181 and reoptimize appropriately.

[See Bandler and Bardakjian (1973).]

Question 183 For the five-section, lossless, transmission-line filter shown in Fig. 37, the following objectives provide two distinct problems, each of which is subjected to a passband insertion loss of no more than 0.01 dB over the band 0 - 1 GHz.

- (a) Maximize the stopband loss at 5 GHz.
- (b) Maximize the minimum stopband loss over the range 2.5 - 10 GHz.

The characteristic impedances are to be fixed at the values

$$\begin{aligned} Z_1 &= Z_3 = Z_5 = 0.2 \\ Z_2 &= Z_4 = 5 \end{aligned}$$

and the section lengths (normalized to  $\ell_q$  as the quarter-wavelength at 1 GHz) as variables. Suggested sample points are: 21 uniformly distributed in the passband, 16 for the stopband in problem (b). A suggested starting point is

$$\ell_1/\ell_q = \ell_5/\ell_q = 0.07,$$

$$\ell_3/\ell_q = 0.15,$$

$$\ell_2/\ell_q = \ell_4/\ell_q = 0.15.$$

[Source for Problem (a): Brancher, Maffioli and Premoli (1970). See also Bandler and Charalambous (1972).]

Question 184 Solve Question 183(a) with normalized lengths fixed at 0.2 and impedances variable.

[See Levy (1965).]

Question 185 Consider the circuit of Question 183. Let the passband be 0 - 1 GHz. Consider a single stopband frequency of 3 GHz. The attenuation in the passband should not exceed 0.4 dB, while the attenuation at 3 GHz should be as high as possible, subject to the following constraints:

$$\ell_i = \ell_q, 0.5 \leq Z_i \leq 2.0, i = 1, 2, \dots, 5,$$

where

$$\ell_q = 2.5 \text{ cm (quarterwave at 3 GHz).}$$

It is suggested that 21 uniformly spaced frequencies are chosen in the passband.

[See Srinivasan (1973) and Carlin (1971).]

Question 186 Reoptimize the example of Question 185 subject to the constraints

$$\begin{aligned} 0 \leq \ell_i/\ell_q \leq 2, \\ 0.4416 \leq Z_i \leq 4.419, \end{aligned} \quad i = 1, 2, \dots, 5$$



$$0 \leq \sum_{i=1}^5 \ell_i / \ell_q \leq 5,$$

where lengths  $\ell_i$  and impedances  $Z_i$  are allowed to vary.

[See Srinivasan and Bandler (1975).]

Question 187 Consider a third-order lumped-distributed-active lowpass filter as shown in Fig. 38. The passband is 0 - 0.7 rad/s, the stopband 1.415 -  $\infty$  rad/s. Three design problems are to be solved for minimax results.

- An attenuation and ripple in the passband of less than 1 dB, with the attenuation in the stopband at least 30 dB (second amplifier removed).
- An attenuation and ripple of 1 dB in the passband with the best stopband response.
- A minimum attenuation and ripple in the passband subject to at least 30 dB attenuation in the stopband.

The nodal equations for the circuit are

$$\begin{bmatrix} y_{22} + j\omega C_1 & -(y_{22} + y_{12}) & 0 \\ -(y_{22} + y_{12} + \frac{A}{R_0}) & y_{11} + y_{22} + y_{12} + y_{21} + \frac{1}{R_0} & 0 \\ -\frac{A}{R_1} & 0 & \frac{1}{R_1} + j\omega C_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} -y_{12} V_S \\ (y_{11} + y_{12}) V_S \\ 0 \end{bmatrix}$$

where  $y_{11}$ ,  $y_{12}$ ,  $y_{21}$  and  $y_{22}$  are the  $y$  parameters of the uniform distributed RC line given by

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = Y \begin{bmatrix} \coth \theta & -\operatorname{csch} \theta \\ -\operatorname{csch} \theta & \coth \theta \end{bmatrix}$$

where  $Y = \sqrt{\frac{sC}{R}}$  and  $\theta = \sqrt{sRC}$ .

Suggested passband sample points are

{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.65, 0.7} rad/s.

Suggested stopband sample points are

{1.415, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0} rad/s.

Let  $C_2 R_1$  be one variable with  $C_2$  fixed at 2.62. Variables to be used for problem (a) are  $A, R, C, R_0, R_1$  and  $C_1$ . For problems (b) and (c) the variables are  $A, C, R_1$  and  $C_1$  with  $R_0 = 1$  and  $R = 17.786$ . It is suggested that the transformation

$$\phi_i = \exp \phi_i'$$

is used so that the variables  $\phi_i'$  are unconstrained while the  $\phi_i$  are positive.

[Source: Charalambous (1974).]

Question 188 A seven-section, cascaded, lossless, transmission-line filter with frequency-dependent terminations is depicted in Fig. 39. The frequency dependence of the terminations is given by

$$R_g = R_L = 377 / \sqrt{1 - (f_c/f)^2},$$

where

$$f_c = 2.077 \text{ GHz.}$$

The section lengths are to be kept fixed at 1.5 cm. The problem is to optimize the 7 characteristic impedances such that a passband specification of 0.4 dB insertion loss is met in the range 2.16 to 3 GHz while the loss at 5 GHz is maximized. Suggested passband sample points

are 22 uniformly spaced frequencies including band edges.

[Reference: Bandler, Srinivasan and Charalambous (1972).]

Question 189 Consider the active filter shown in Fig. 40. Let  $R_g = 50 \Omega$ ,  $R = 75 \Omega$ . Take a model of the amplifier as

$$A(s) = \frac{A_0 \omega_a}{s + \omega_a},$$

where  $s$  is the complex frequency variable,  $A_0$  is the d.c. gain and  $\omega_a = 12\pi$  rad/s. Use the equivalent circuit shown in Fig. 41 for the purpose of nodal analysis.

The ideal transfer function, i.e., for  $A_0 \rightarrow \infty$  and  $R_3 \rightarrow \infty$  is

$$\frac{V_2}{V_g} = -G_1 \frac{sC_1}{s^2 C_1 C_2 + sG_2(C_1 + C_2) + G_2(G_4 + G_1)}$$

and the nodal equations for the nonideal filter are

$$\begin{bmatrix} G_1 + G_g & 0 & -G_1 & 0 \\ 0 & G_2 + G_3 + sC_2 + A_1 A_2 G_3 & -sC_2 & -G_2 + A_1 A_2 G_3 \\ -G_1 & -sC_2 & G_1 + G_4 + sC_1 + sC_2 & -sC_1 \\ 0 & -G_2 & -sC_1 & G_2 + sC_1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} G_g V_g \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Let  $F = |V_2/V_g|$ . The specifications are w.r.t. frequency  $f$ :

$$F \leq 1/\sqrt{2} \text{ for } f \leq 90 \text{ Hz,}$$

$$F \leq 1.1 \text{ for } 90 \leq f \leq 110 \text{ Hz,}$$

$$F \leq 1/\sqrt{2} \text{ for } f \geq 110 \text{ Hz,}$$

$$F \geq 1/\sqrt{2} \text{ for } 92 \leq f \leq 108 \text{ Hz,}$$

$$F \geq 1 \text{ for } f = 100 \text{ Hz.}$$

Find an optimum solution in the minimax sense for components  $R_1$ ,  $C_1$ ,  $C_2$  and  $R_4$ , given

$$\begin{aligned}A_0 &= 2 \times 10^5, \\R_2 &= 2.65 \times 10^4 \Omega, \\C_1 &= C_2 = C.\end{aligned}$$

Question 190 Describe in detail and explain all the information to be supplied by a user to run the optimization package you are currently using or are familiar with.

Question 191 Describe all necessary steps required to access the optimization package described in Question 190 to execute an optimization problem in conjunction with user-supplied programs.

Question 192 What is the effect on the number of function evaluations or iterations of changing starting points in the minimization problems you have tested using the package of Question 190.

Question 193 Each student should familiarize himself with the optimization package under study by running the examples in the user's manual. Run each example from starting points different to the ones given and compare the results with those in the manual.

Question 194 For the resistive network of Question 27, solve the nodal equations by an unconstrained minimization package. Take  $G_1 = G_3 = G_5 = 1$  mho,  $R_2 = R_4 = 0.5$  ohm. Write all necessary subprograms.

Question 195 For the voltage divider of Question 154, the specifications

$$0.46 < \frac{R_2}{R_1 + R_2} < 0.53$$

$$1.85 < R_1 + R_2 < 2.15$$

must be met in the minimax sense using an available package. Write all necessary subprograms.

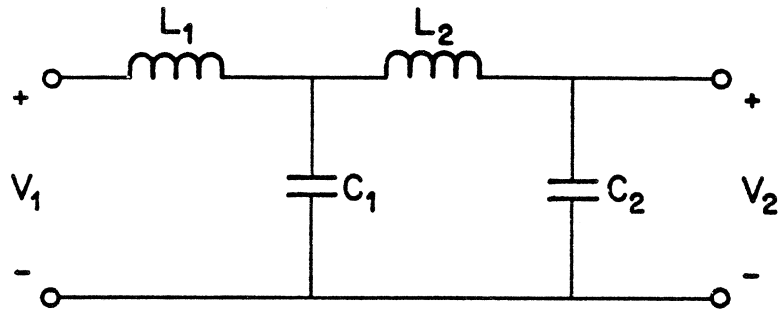


Fig. 1 LC ladder network (Question 10).

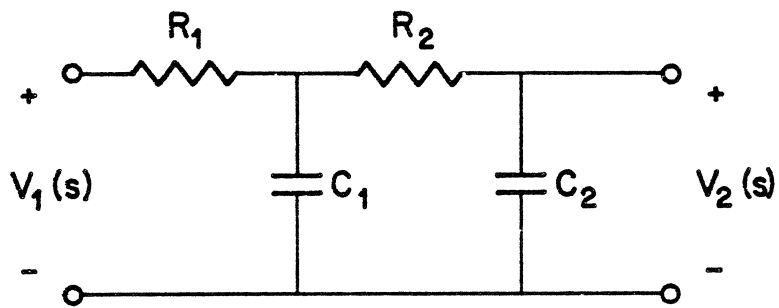


Fig. 2 RC ladder network (Question 12).

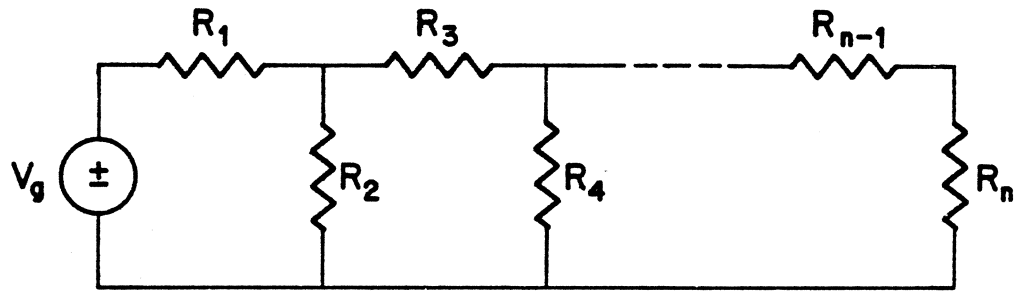


Fig. 3 Resistive ladder network (Question 16).

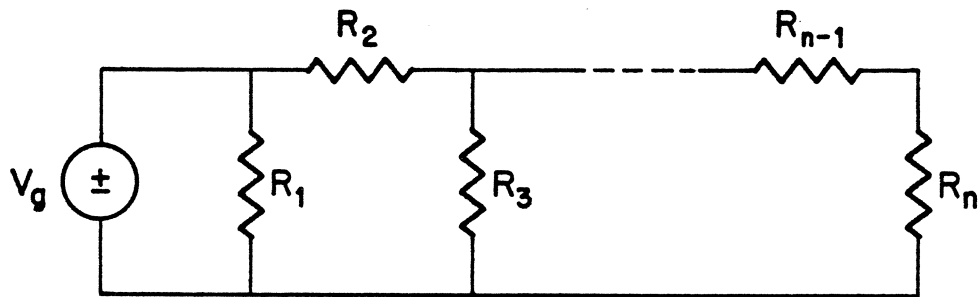


Fig. 4 Resistive ladder network (Question 18).

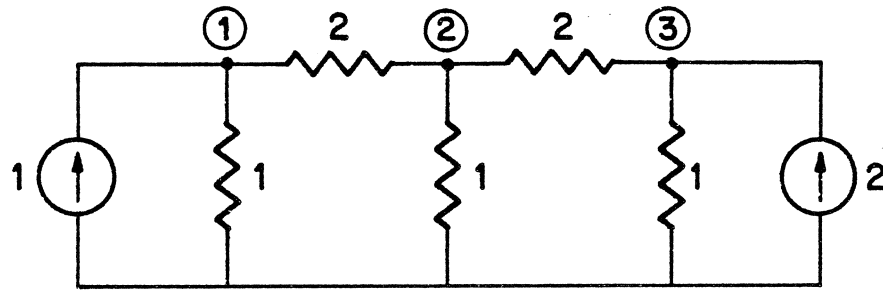


Fig. 5 Three-node resistive ladder network (Question 20).

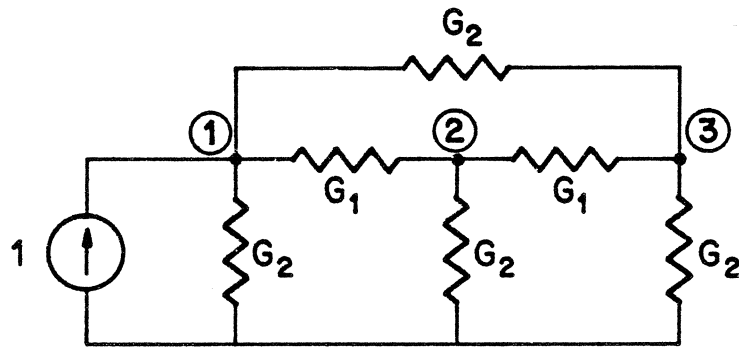


Fig. 6 Three-node resistive ladder network (Question 25).



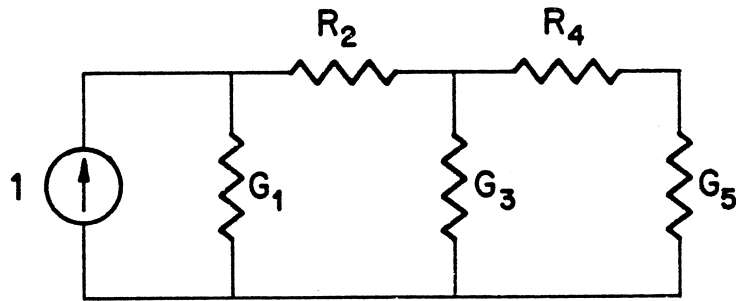


Fig. 7 Resistive ladder network (Question 27).

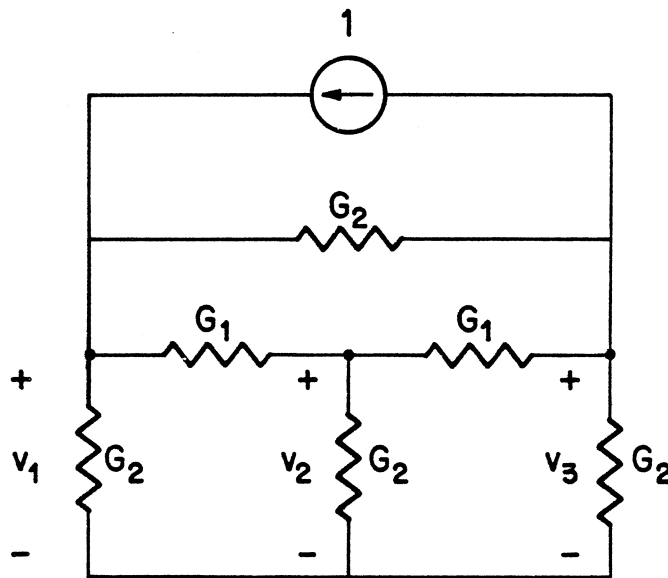


Fig. 8 Resistive network (Question 28).

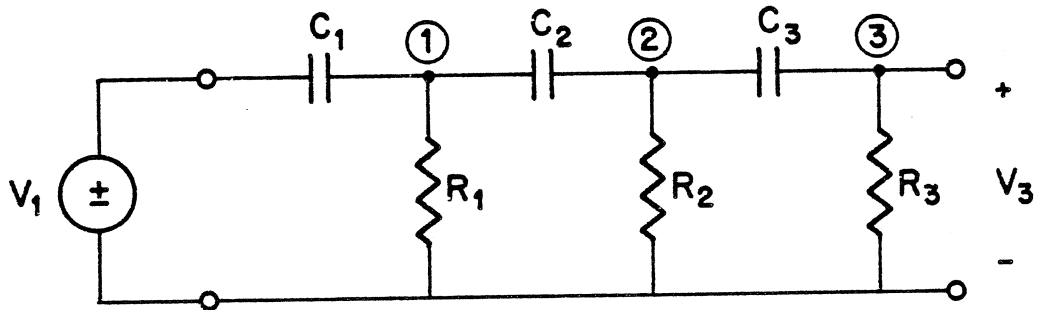


Fig. 9 CR ladder network (Question 29).

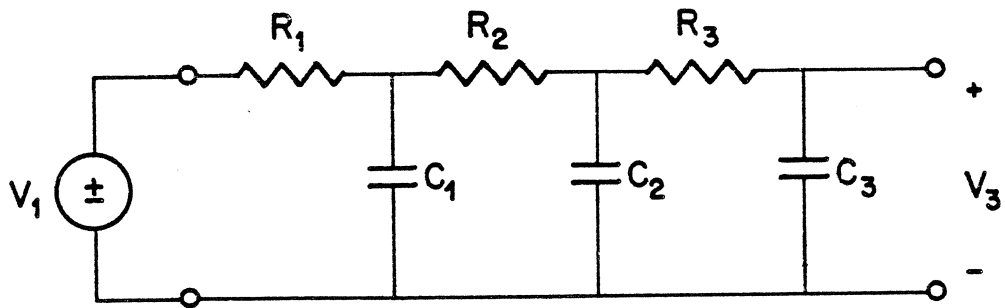


Fig. 10 RC ladder network (Question 30).

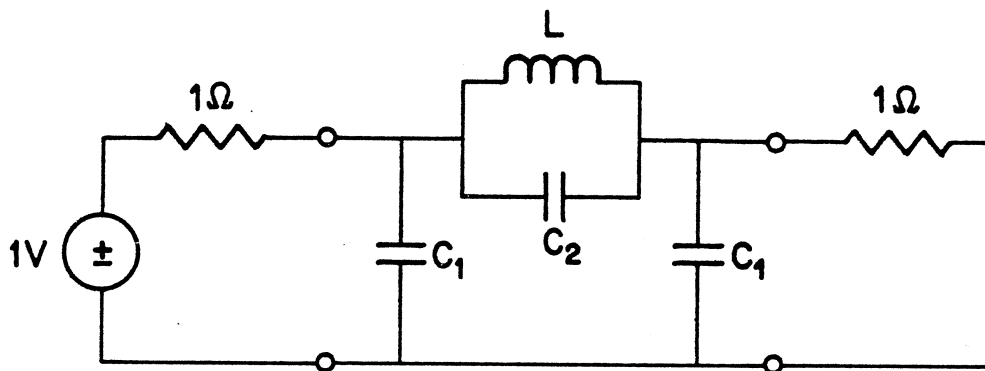


Fig. 11 LC filter network (Question 32).

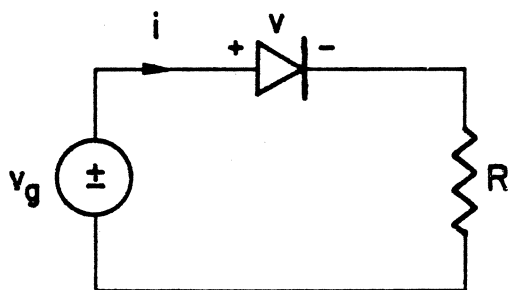


Fig. 12 Resistor-diode network (Question 55).

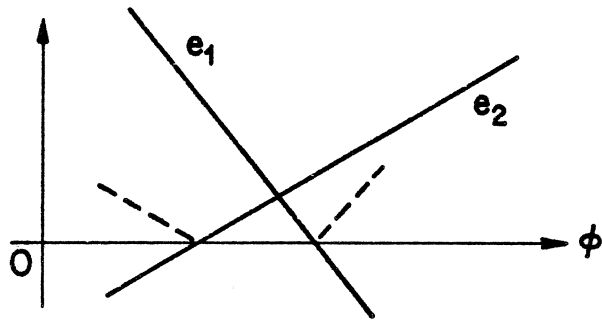


Fig. 13 Error functions (Question 100).

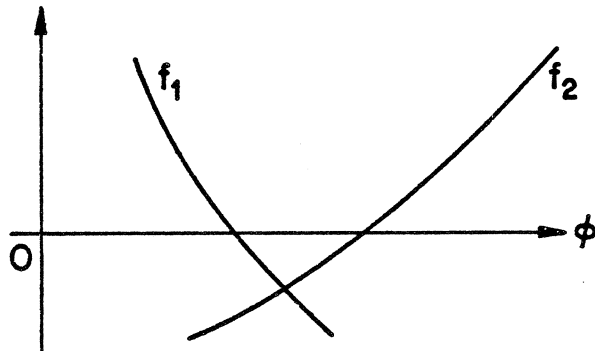


Fig. 14 Two functions of one variable (Question 101).

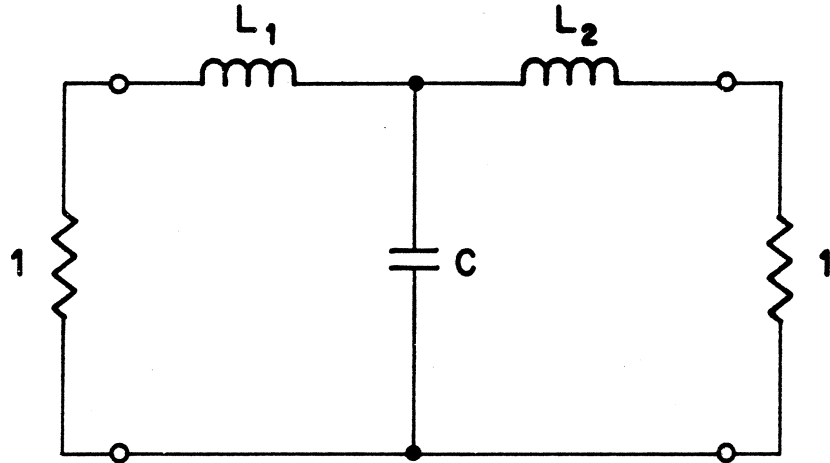


Fig. 15 LC lowpass filter (Question 106).

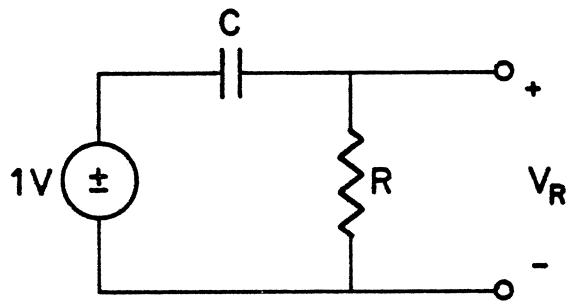


Fig. 16 RC circuit (Question 108).

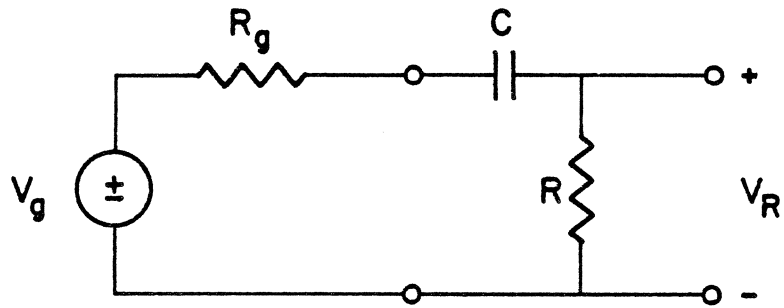


Fig. 17 RC circuit (Question 109).

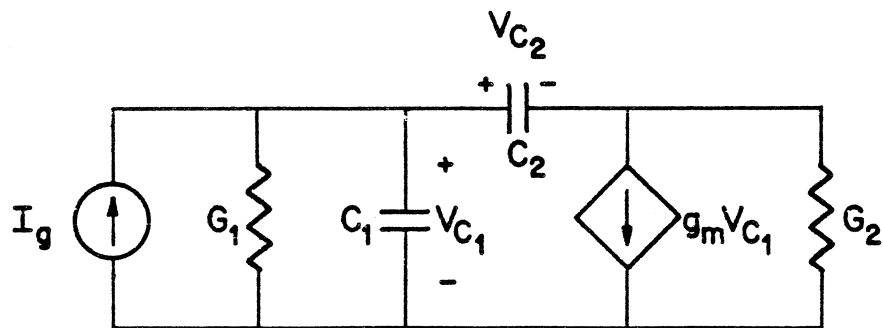


Fig. 18 Active circuit (Question 110).

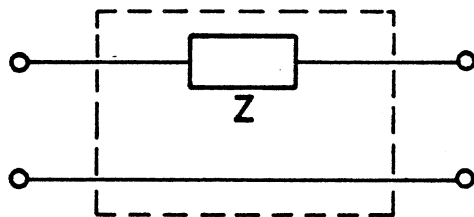
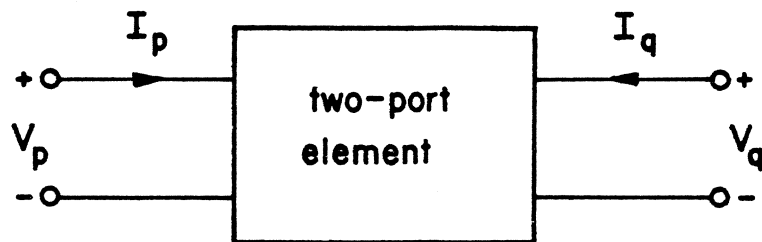


Fig. 19 Example of two-port (Question 123).

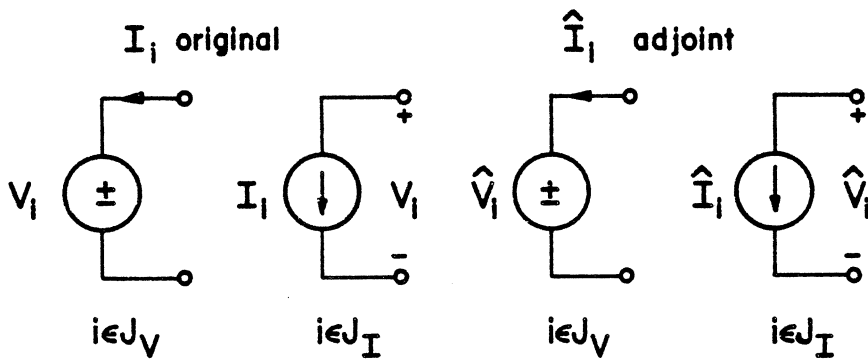


Fig. 20 Excitations and responses in the original and adjoint networks (Question 125).

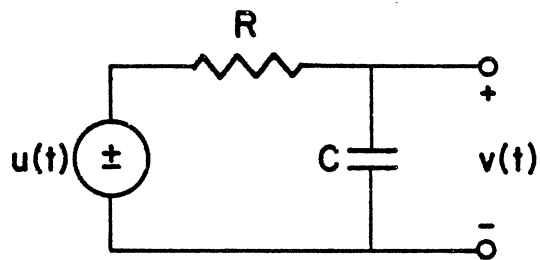


Fig. 21 RC circuit (Question 126).



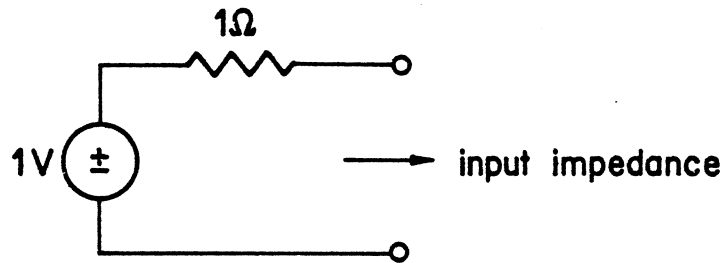


Fig. 22 Source for input impedance calculation (Question 127).

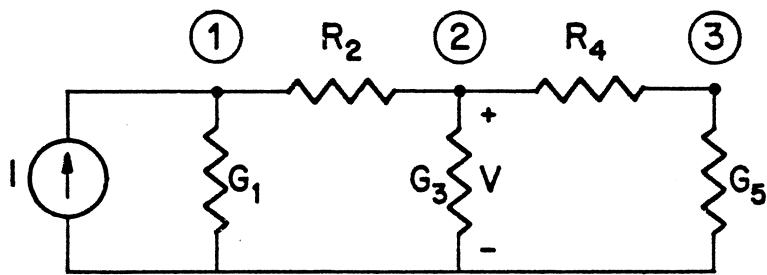


Fig. 23 Three-node resistive network (Question 130).

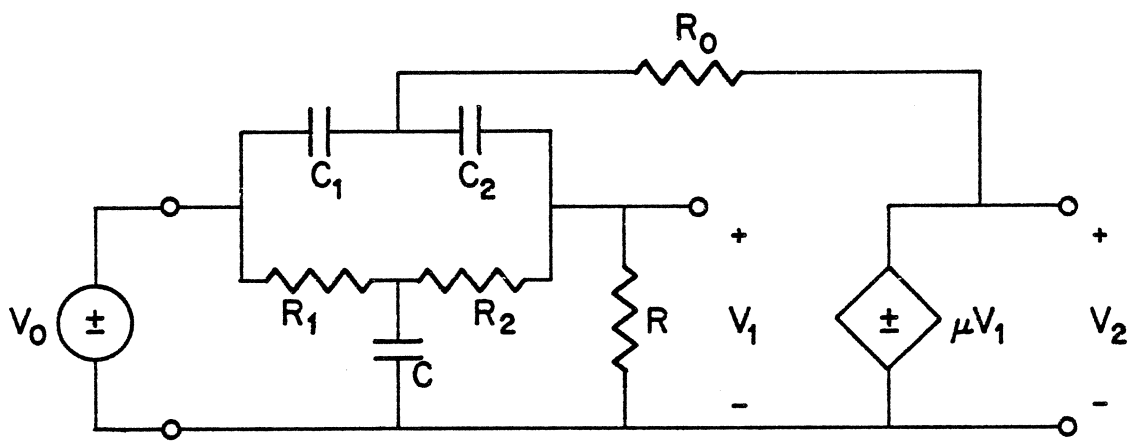


Fig. 24 Active circuit example (Question 132).

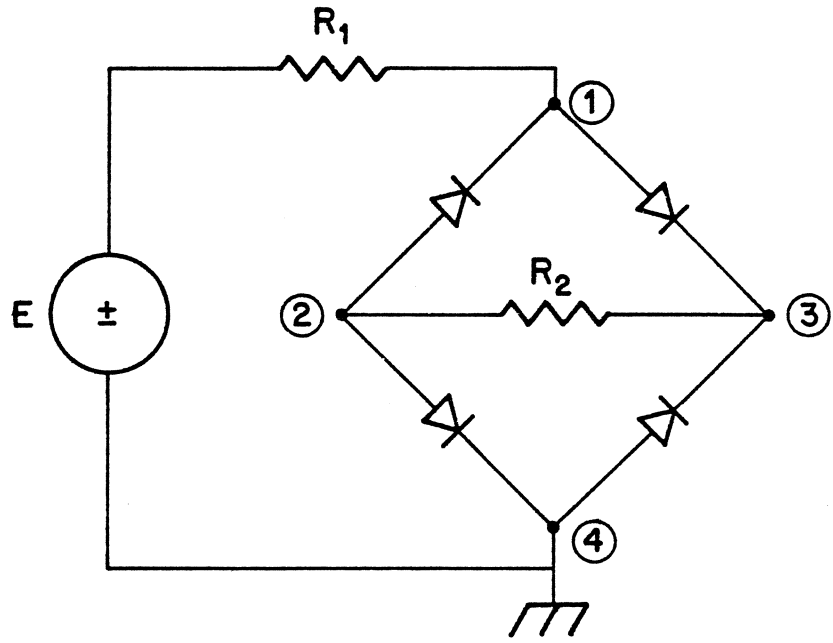


Fig. 25 Resistor-diode network (Question 136).

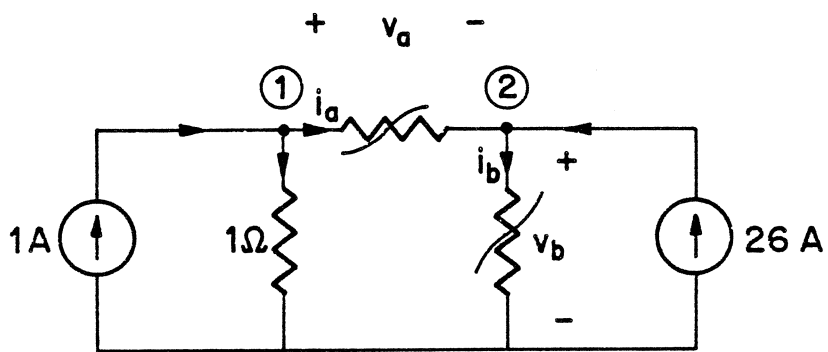


Fig. 26 Nonlinear circuit example (Question 140).

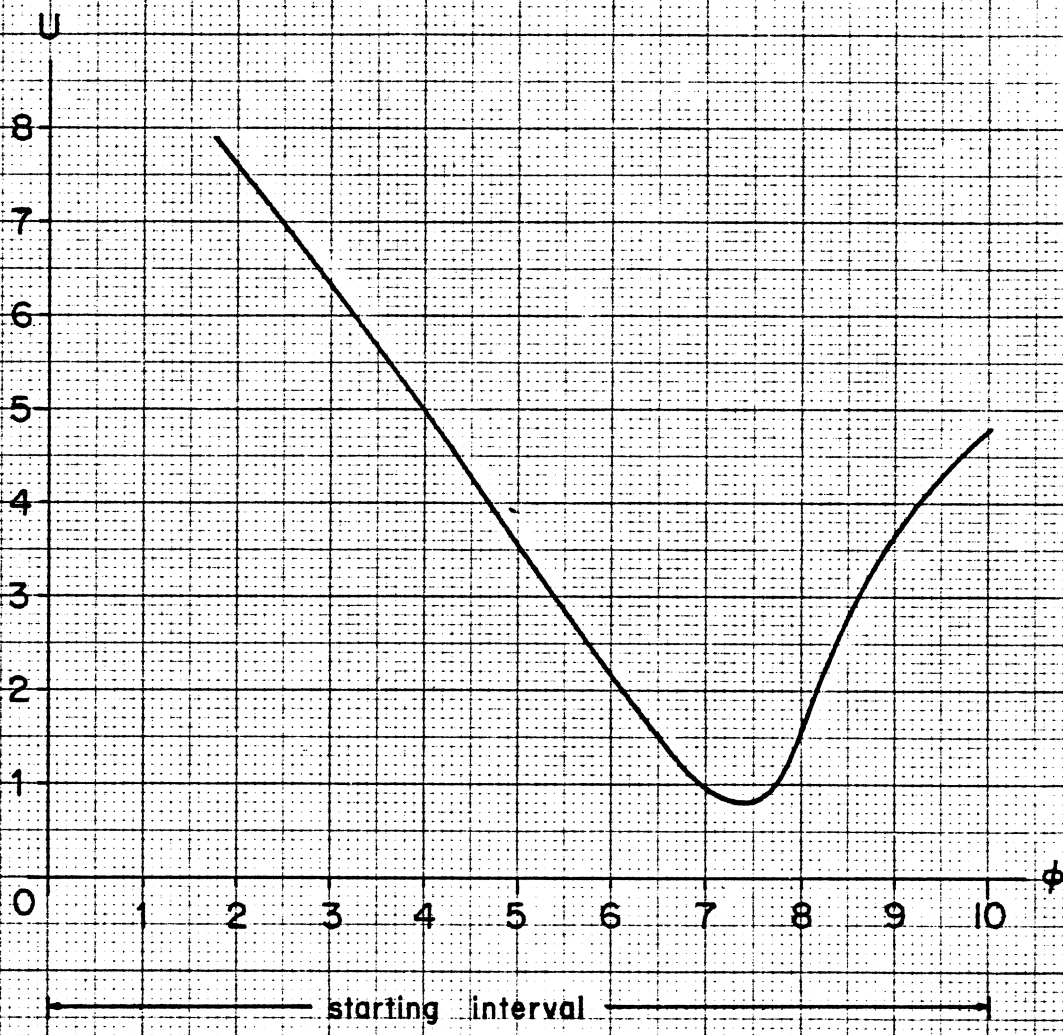


Fig. 27 Function of one variable (Question 146).

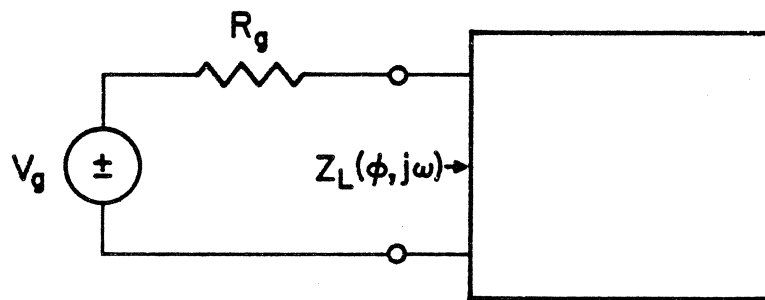


Fig. 28 Impedance matching example (Question 153).

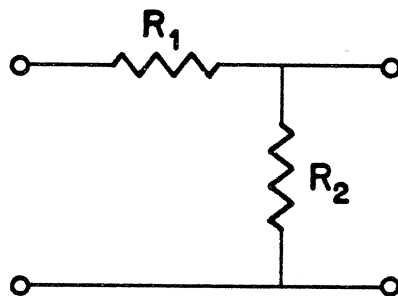


Fig. 29 Voltage divider circuit (Question 154).

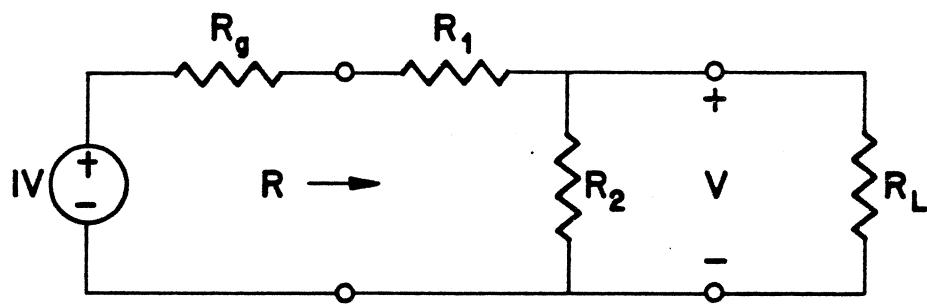


Fig. 30 Nonideal voltage divider circuit (Question 157).

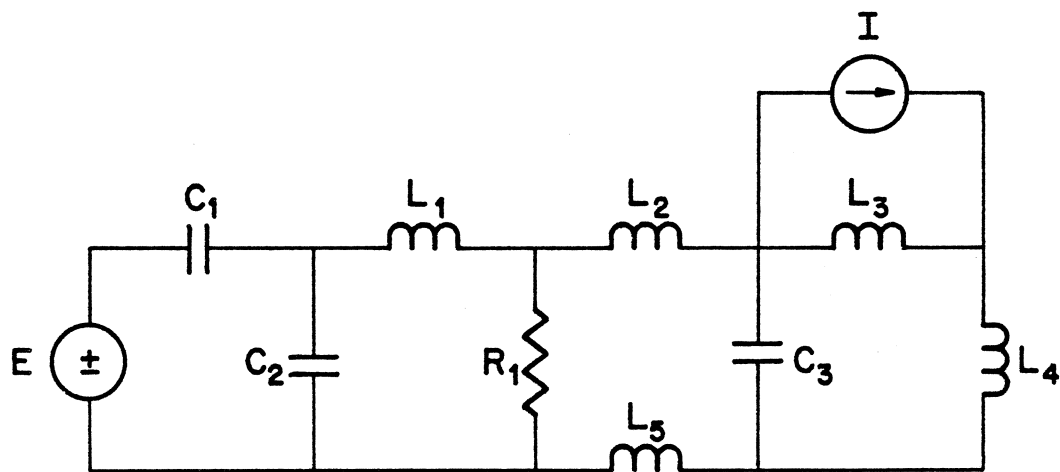


Fig. 31 Arbitrary network (Question 163).

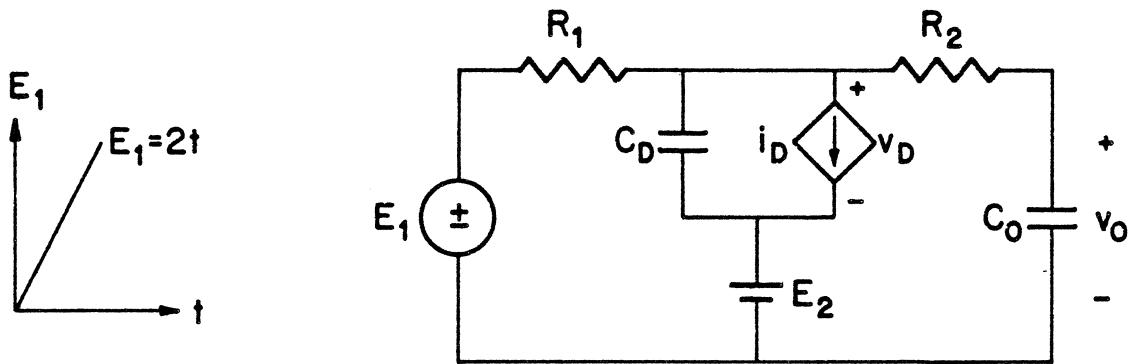


Fig. 32 Time domain circuit example (Question 164).

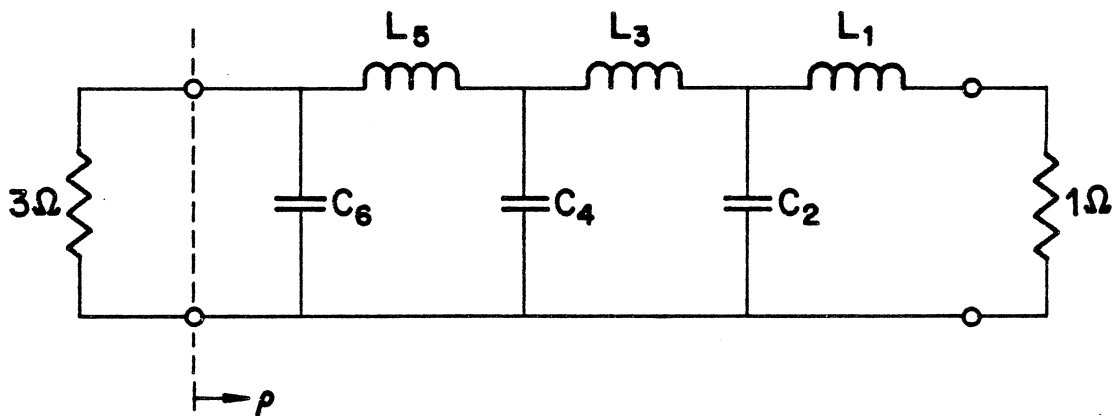


Fig. 33 Lumped element LC transformer (Question 168).



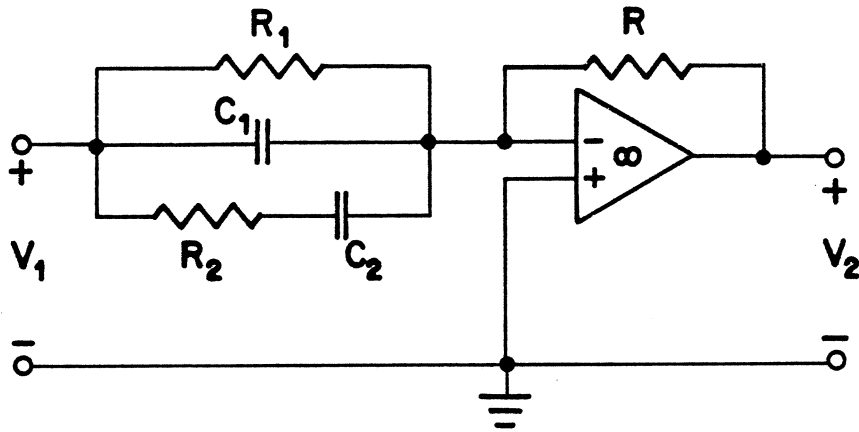


Fig. 34 RC active equalizer example (Question 169).

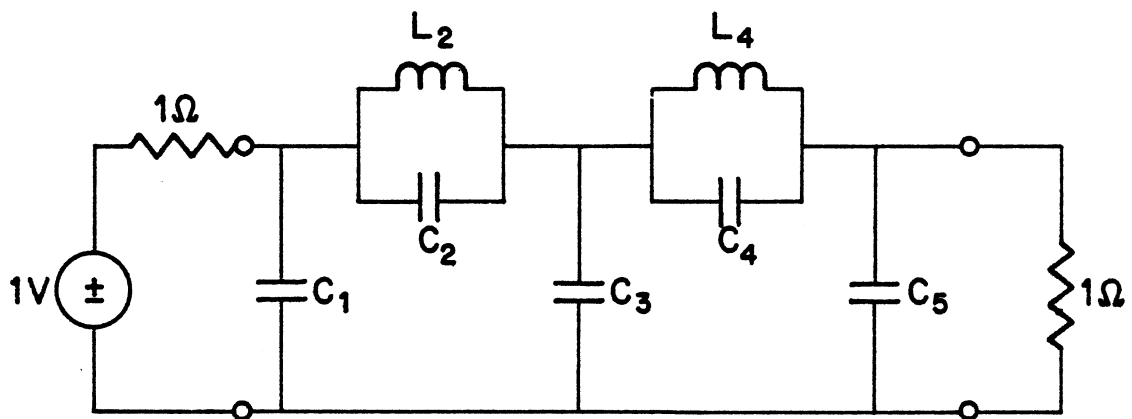


Fig. 35 Elliptic low-pass filter (Question 171).

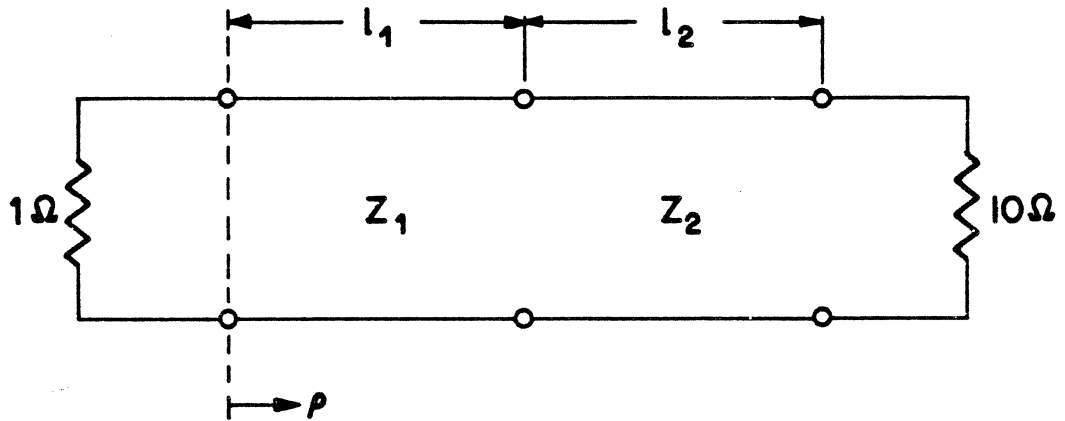


Fig. 36 Two-section transmission-line transformer example (Question 174).

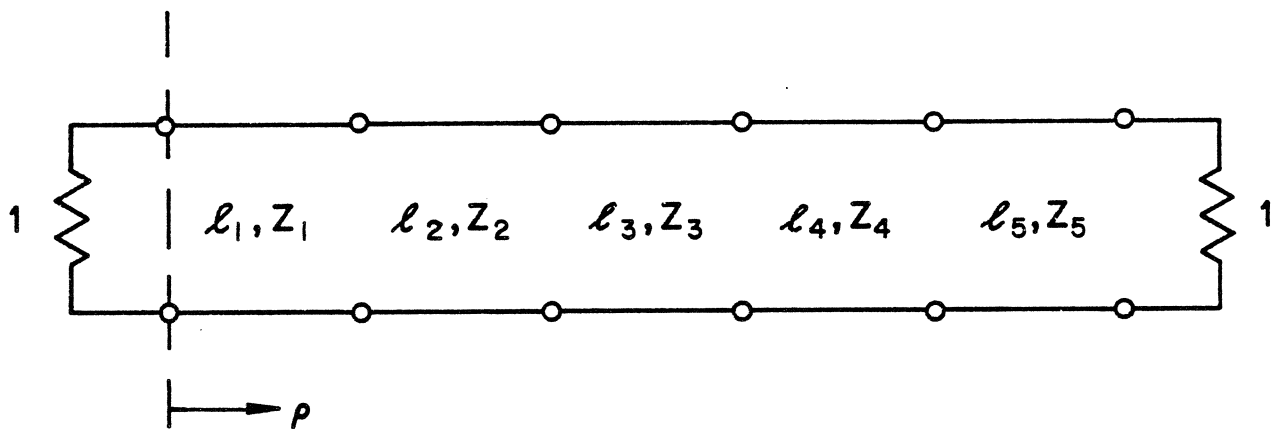


Fig. 37 Five-section transmission-line filter (Question 183).

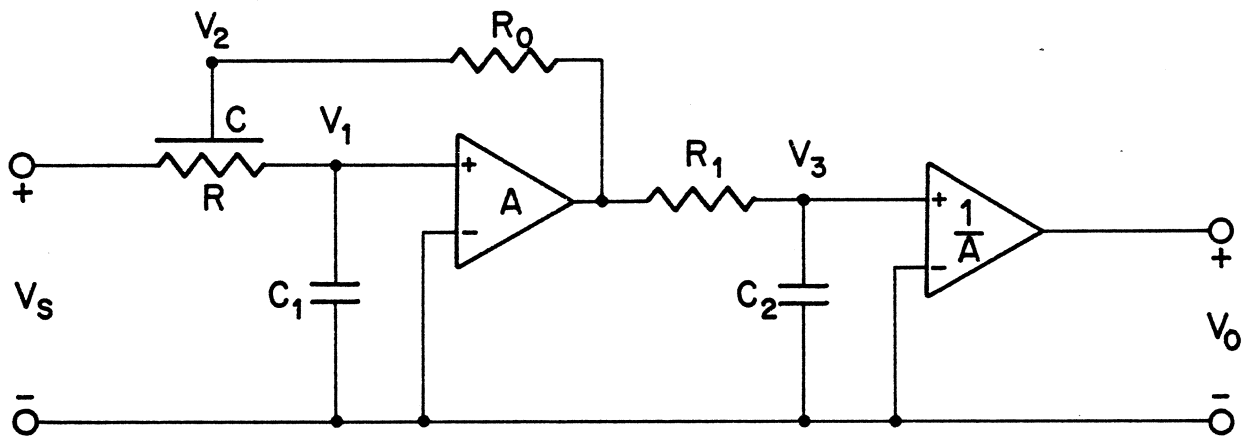


Fig. 38 Third-order lumped-distributed-active lowpass filter. (Question 187).

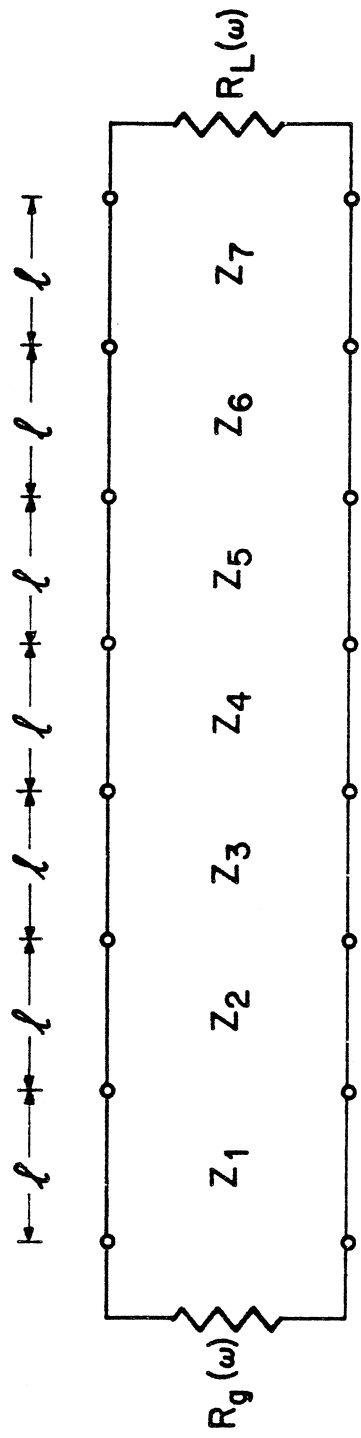


Fig. 39 Seven-section, cascaded transmission-line filter (Question 188).

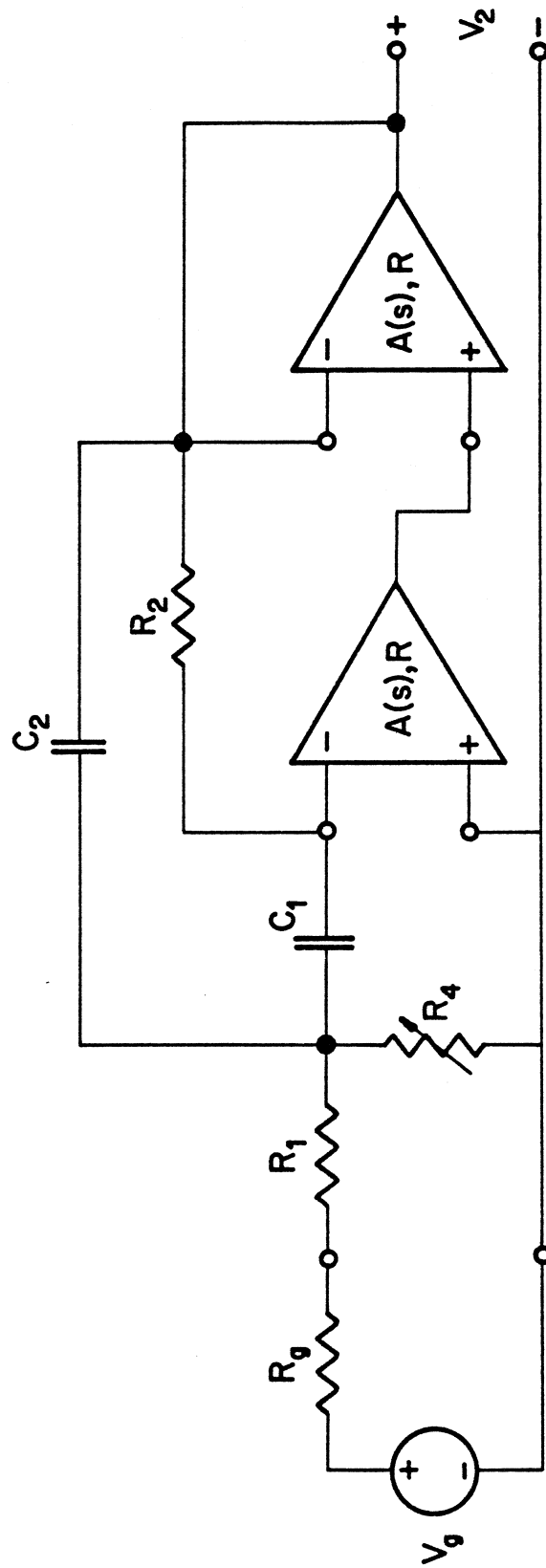


Fig. 40 Active filter (Question 189).

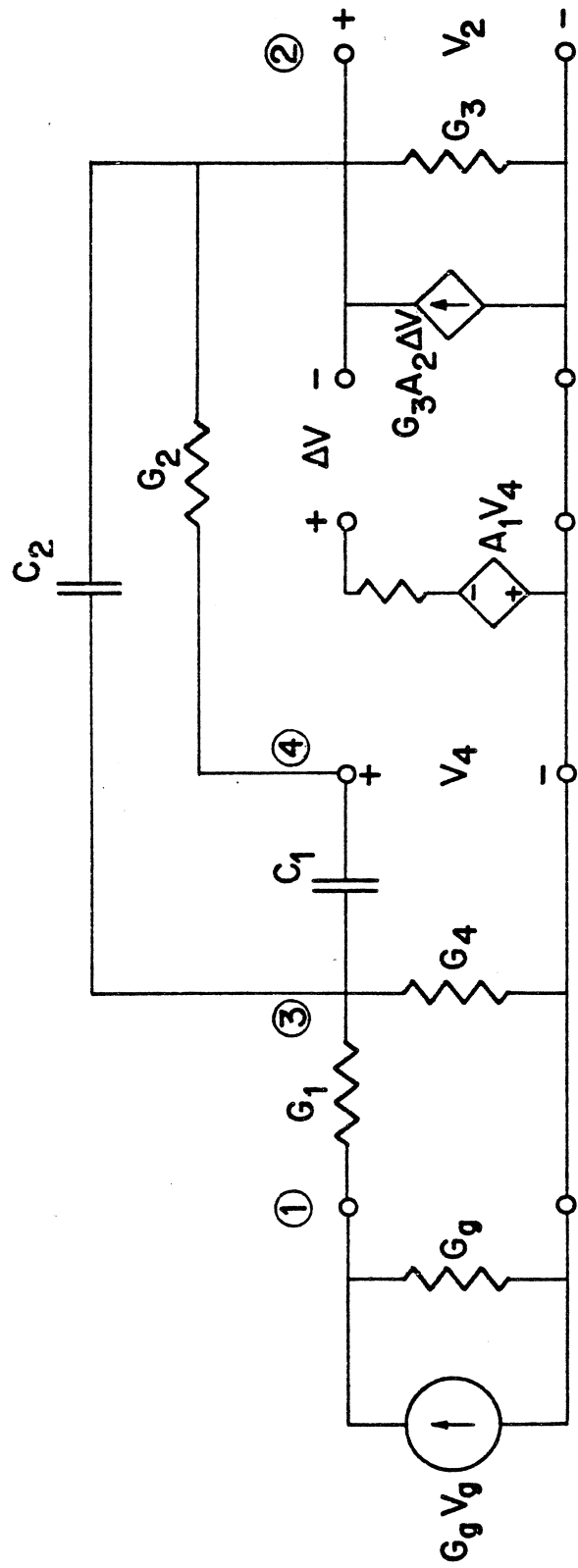


Fig. 41 Equivalent circuit for the active filter of Fig. 40 (Question 189).

**SECTION THREE**  
**NOTES ON VECTORS, MATRICES**  
**AND SENSITIVITIES**

© J.W. Bandler and Q.J. Zhang 1986

This document originally appeared as Report SOS-86-8-R, September 1986. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.





Matrix A

Let

$$\mathbf{A} \triangleq \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

This  $m \times n$  matrix has  $m$  rows and  $n$  columns.

Transpose of A

$$\mathbf{A}^T \triangleq \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

This  $n \times m$  matrix has  $n$  rows and  $m$  columns.

Symmetric Matrix A

A square matrix  $\mathbf{A}$  is said to be symmetric if

$$\mathbf{A}^T = \mathbf{A}$$

Vector a

Let

$$\mathbf{a} \triangleq \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_n \end{bmatrix}$$

This n-dimensional vector has n rows and 1 column.

Transpose of a

$$\mathbf{a}^T \triangleq [a_1 \quad a_2 \quad \dots \quad a_n]$$

This n-dimensional vector has 1 row and n columns.

Vector  $\mathbf{a}'$ 

Let

$$\mathbf{a}' \triangleq \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_m \end{bmatrix}$$

This m-dimensional vector has m rows and 1 column.

Transpose of  $\mathbf{a}'$ 

$$\mathbf{a}'^T \triangleq [a_1 \quad a_2 \quad \dots \quad a_m]$$

This m-dimensional vector has 1 row and m columns.

Forms of A and A<sup>T</sup>

Partitioned forms of A and A<sup>T</sup> can be written as

$$\mathbf{A}^T = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \dots \quad \mathbf{a}_m] = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{a}_n^T \end{bmatrix} = \mathbf{A}'$$

and

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{a}_m^T \end{bmatrix} = [\mathbf{a}'_1 \quad \mathbf{a}'_2 \quad \dots \quad \mathbf{a}'_n] = \mathbf{A}'^T$$

Here,  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$  are n-dimensional, whereas  $\mathbf{a}'_1, \mathbf{a}'_2, \dots, \mathbf{a}'_n$  are m-dimensional.

Rows of A

$$A = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \cdot \\ \cdot \\ \mathbf{a}_m^T \end{bmatrix}$$

This  $m \times n$  matrix has rows which are the transposes of the column vectors

$$\mathbf{a}_1 \triangleq \begin{bmatrix} a_{11} \\ a_{12} \\ \cdot \\ \cdot \\ a_{1n} \end{bmatrix}, \quad \mathbf{a}_2 \triangleq \begin{bmatrix} a_{21} \\ a_{22} \\ \cdot \\ \cdot \\ a_{2n} \end{bmatrix}, \quad \dots, \quad \mathbf{a}_m \triangleq \begin{bmatrix} a_{m1} \\ a_{m2} \\ \cdot \\ \cdot \\ a_{mn} \end{bmatrix}$$

Columns of A

$$A = [a'_1 \quad a'_2 \quad \dots \quad a'_n]$$

This  $m \times n$  matrix has columns

$$a'_1 \triangleq \begin{bmatrix} a_{11} \\ a_{21} \\ \cdot \\ \cdot \\ \cdot \\ a_{m1} \end{bmatrix}, \quad a'_2 \triangleq \begin{bmatrix} a_{12} \\ a_{22} \\ \cdot \\ \cdot \\ \cdot \\ a_{m2} \end{bmatrix}, \quad \dots, \quad a'_n \triangleq \begin{bmatrix} a_{1n} \\ a_{2n} \\ \cdot \\ \cdot \\ \cdot \\ a_{mn} \end{bmatrix}$$

Unit Vector  $u_i$ 

$$u_i \triangleq \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \leftarrow \text{ith row}$$

This vector is considered as having  $m$  elements, all of which are 0 except the  $i$ th, which is 1.

Unit Vector  $u_j$ 

$$u_j \triangleq \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \leftarrow \text{jth row}$$

This vector is considered as having  $n$  elements, all of which are 0 except the  $j$ th, which is 1.



The Identity Matrix

Let

$$\mathbf{u}_1 \triangleq \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \quad \mathbf{u}_2 \triangleq \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \quad \dots, \quad \mathbf{u}_n \triangleq \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}.$$

Then

$$\mathbf{I} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_n]$$

is called the identity matrix.

Scalar Product

Let

$$\mathbf{a} \triangleq \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_n \end{bmatrix}, \quad \mathbf{b} \triangleq \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_n \end{bmatrix}.$$

Then

$$\mathbf{a}^T \mathbf{b} \triangleq a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

The result is a scalar.

Element Selection from Vectors

$$\mathbf{u}_j^T \mathbf{a} \equiv \mathbf{a}^T \mathbf{u}_j = a_j$$

The result is the scalar element  $a_j$ .

$$\mathbf{u}_i^T \mathbf{a}' \equiv \mathbf{a}'^T \mathbf{u}_i = a'_i$$

The result is the scalar element  $a'_i$ .

Row Selection from Matrices

$$\mathbf{u}_i^T \mathbf{A} = \mathbf{u}_i^T \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{a}_m^T \end{bmatrix} = \mathbf{a}_i^T$$

This result is the row vector corresponding to the  $i$ th row of  $\mathbf{A}$ . In transposed form, we have

$$\mathbf{A}^T \mathbf{u}_i = \mathbf{a}_i$$

Column Selection from Matrices

$$\mathbf{A} \mathbf{u}_j = [\mathbf{a}'_1 \quad \mathbf{a}'_2 \quad \dots \quad \mathbf{a}'_n] \mathbf{u}_j = \mathbf{a}'_j$$

This result is the column vector corresponding to the  $j$ th column of  $\mathbf{A}$ . In transposed form, we have

$$\mathbf{u}_j^T \mathbf{A}^T = \mathbf{a}'_j{}^T$$

Element Selection from Matrices

$$\mathbf{u}_i^T \mathbf{A} \mathbf{u}_j = \begin{cases} \mathbf{u}_i^T (\mathbf{A} \mathbf{u}_j) = \mathbf{u}_i^T \mathbf{a}_j = a_{ij} \\ (\mathbf{u}_i^T \mathbf{A}) \mathbf{u}_j = \mathbf{a}_i^T \mathbf{u}_j = a_{ij} \end{cases}$$

This result is the scalar element corresponding to the coefficient of  $\mathbf{A}$  obtained from the intersection of row  $i$  and column  $j$ .

Element Representation

$$A = \begin{bmatrix} \mathbf{u}_1^T \mathbf{a}_1 & \mathbf{u}_2^T \mathbf{a}_1 & \dots & \mathbf{u}_n^T \mathbf{a}_1 \\ \mathbf{u}_1^T \mathbf{a}_2 & \mathbf{u}_2^T \mathbf{a}_2 & \dots & \mathbf{u}_n^T \mathbf{a}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_1^T \mathbf{a}_m & \mathbf{u}_2^T \mathbf{a}_m & \dots & \mathbf{u}_n^T \mathbf{a}_m \end{bmatrix} = A'^T,$$

where  $\mathbf{u}_j, j = 1, \dots, n$  is  $n$ -dimensional.

$$A'^T = \begin{bmatrix} \mathbf{u}_1^T \mathbf{a}'_1 & \mathbf{u}_1^T \mathbf{a}'_2 & \dots & \mathbf{u}_1^T \mathbf{a}'_n \\ \mathbf{u}_2^T \mathbf{a}'_1 & \mathbf{u}_2^T \mathbf{a}'_2 & \dots & \mathbf{u}_2^T \mathbf{a}'_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_m^T \mathbf{a}'_1 & \mathbf{u}_m^T \mathbf{a}'_2 & \dots & \mathbf{u}_m^T \mathbf{a}'_n \end{bmatrix} = A,$$

where  $\mathbf{u}_i, i = 1, \dots, m$  is  $m$ -dimensional.

Assembly of an Element into a Column Vector

To place the parameter  $\phi$  into the  $i$ th row of a column vector we write

$$u_i \phi = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ \phi \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \leftarrow \text{ith row}$$

This results in a column vector containing zeros everywhere except in the  $i$ th row, which contains  $\phi$ .



Assembly of an Element into a Row Vector

To place the parameter  $\phi$  into the  $j$ th column of a row vector we write

$$\phi \mathbf{u}_j^T = [0 \quad \dots \quad \phi \quad \dots \quad 0]$$

↑  
jth col

This results in a row vector containing zeros everywhere except in the  $j$ th column, which contains  $\phi$ .

Assembly of a Vector into a Row of a Matrix

To place the  $n$ -dimensional vector  $\mathbf{a}$  into the  $i$ th row of a matrix we write

$$\mathbf{u}_i \mathbf{a}^T = \begin{bmatrix} 0 & 0 & & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & & 0 \\ a_1 & a_2 & \dots & a_n \\ 0 & 0 & & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & & 0 \end{bmatrix} \leftarrow \text{ith row}$$

This results in a matrix containing zeros everywhere except in the  $i$ th row, which contains  $\mathbf{a}^T$ .

Assembly of a Vector into a Column of a Matrix

To place the  $m$ -dimensional vector  $\mathbf{a}'$  into the  $j$ th column of a matrix we write

$$\mathbf{a}' \mathbf{u}_j^T = \begin{bmatrix} 0 & \dots & 0 & a'_1 & 0 & \dots & 0 \\ 0 & \dots & 0 & a'_2 & 0 & \dots & 0 \\ & & & \vdots & & & \\ & & & \vdots & & & \\ 0 & \dots & 0 & a'_m & 0 & \dots & 0 \end{bmatrix}$$

↑  
jth col

This results in a matrix containing zeros everywhere except in the  $j$ th column, which contains  $\mathbf{a}'$ .

Assembly of an Element into a Matrix

To place the parameter  $\phi$  into the intersection of row  $i$  and column  $j$  we write

$$\mathbf{u}_i \phi \mathbf{u}_j^T = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \dots \phi \dots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \begin{array}{l} \text{ith row} \\ \text{jth col} \end{array}$$

This results in a matrix containing zeros everywhere except in the  $i,j$  location, which contains  $\phi$ .

Two other forms are convenient to represent. The first one is

$$\mathbf{u}_i (\phi \mathbf{u}_j^T) = \begin{bmatrix} \mathbf{0}^T \\ \mathbf{0}^T \\ \vdots \\ \phi \mathbf{u}_j^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} \leftarrow \text{ith row}$$

The other form is

$$(\mathbf{u}_i \phi) \mathbf{u}_j^T = [0 \quad 0 \quad \dots \quad \mathbf{u}_i \phi \quad \dots \quad 0]$$

$\uparrow$   
 jth col

Assembly of a Matrix

To assemble **A** we may write

$$\mathbf{A} = \sum_{i=1}^m \sum_{j=1}^n \mathbf{u}_i a_{ij} \mathbf{u}_j^T$$

Assembly of the ith Row of a Matrix

To assemble the  $i$ th row of  $\mathbf{A}$  we write

$$\mathbf{a}_i^T = \sum_{j=1}^n a_{ij} \mathbf{u}_j^T$$

Assembly of the jth Column of a Matrix

To assemble the jth column of  $A$  we write

$$\mathbf{a}_j = \sum_{i=1}^m \mathbf{u}_i a_{ij}$$

Dyadic Product of a and b

Let

$$\mathbf{a} \triangleq \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_n \end{bmatrix}, \quad \mathbf{b} \triangleq \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_m \end{bmatrix}$$

Then

$$\mathbf{a} \mathbf{b}^T \triangleq \begin{bmatrix} a_1 b_1 & a_1 b_2 & \dots & a_1 b_m \\ a_2 b_1 & a_2 b_2 & \dots & a_2 b_m \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_n b_1 & a_n b_2 & \dots & a_n b_m \end{bmatrix}$$

This result is the  $n$  by  $m$  matrix containing all possible products  $a_i b_j$ ,  $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, m$ .



Definitions of Derivative Operators

$$\frac{\partial}{\partial \mathbf{a}} \triangleq \begin{bmatrix} \frac{\partial}{\partial a_1} \\ \frac{\partial}{\partial a_2} \\ \cdot \\ \cdot \\ \frac{\partial}{\partial a_n} \end{bmatrix}$$

$$\frac{\partial}{\partial \mathbf{A}} \triangleq \begin{bmatrix} \frac{\partial}{\partial a_{11}} & \frac{\partial}{\partial a_{12}} & \dots & \frac{\partial}{\partial a_{1n}} \\ \frac{\partial}{\partial a_{21}} & \frac{\partial}{\partial a_{22}} & \dots & \frac{\partial}{\partial a_{2n}} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \frac{\partial}{\partial a_{m1}} & \frac{\partial}{\partial a_{m2}} & \dots & \frac{\partial}{\partial a_{mn}} \end{bmatrix}$$

Derivatives of Scalar Products

$$\frac{\partial(\mathbf{a}^T \mathbf{b})}{\partial \mathbf{a}} = \left( \frac{\partial \mathbf{a}^T}{\partial \mathbf{a}} \right) \mathbf{b} = \mathbf{b}$$

Also,

$$\frac{\partial(\mathbf{b}^T \mathbf{a})}{\partial \mathbf{a}} = \frac{\partial(\mathbf{a}^T \mathbf{b})}{\partial \mathbf{a}} = \mathbf{b}$$

Jacobian Matrix [ $\mathbf{b} \neq \mathbf{a}$ ]

Let

$$\mathbf{a} \triangleq \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{b} \triangleq \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Then

$$\begin{aligned} \frac{\partial \mathbf{b}^T}{\partial \mathbf{a}} &\triangleq \left( \frac{\partial}{\partial \mathbf{a}} \right) \mathbf{b}^T = \begin{bmatrix} \frac{\partial b_1}{\partial a_1} & \frac{\partial b_1}{\partial a_2} & \cdots & \frac{\partial b_1}{\partial a_n} \\ \frac{\partial b_2}{\partial a_1} & \frac{\partial b_2}{\partial a_2} & \cdots & \frac{\partial b_2}{\partial a_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial b_m}{\partial a_1} & \frac{\partial b_m}{\partial a_2} & \cdots & \frac{\partial b_m}{\partial a_n} \end{bmatrix}^T \\ &= \begin{bmatrix} \left( \frac{\partial b_1}{\partial \mathbf{a}} \right)^T \\ \left( \frac{\partial b_2}{\partial \mathbf{a}} \right)^T \\ \vdots \\ \left( \frac{\partial b_m}{\partial \mathbf{a}} \right)^T \end{bmatrix}^T = \begin{bmatrix} \frac{\partial b_1}{\partial \mathbf{a}} & \frac{\partial b_2}{\partial \mathbf{a}} & \cdots & \frac{\partial b_m}{\partial \mathbf{a}} \end{bmatrix} \\ &= \left[ \frac{\partial \mathbf{b}}{\partial a_1} \quad \frac{\partial \mathbf{b}}{\partial a_2} \quad \cdots \quad \frac{\partial \mathbf{b}}{\partial a_n} \right]^T \end{aligned}$$

Jacobian Matrix [ $\mathbf{b} = \mathbf{a}$ ]

$$\begin{aligned} \frac{\partial \mathbf{a}^T}{\partial \mathbf{a}} &= \begin{bmatrix} \frac{\partial a_1}{\partial \mathbf{a}} & \frac{\partial a_2}{\partial \mathbf{a}} & \cdots & \frac{\partial a_n}{\partial \mathbf{a}} \end{bmatrix} \\ &= [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_n] = \mathbf{1} \\ &= \begin{bmatrix} \frac{\partial \mathbf{a}}{\partial a_1} & \frac{\partial \mathbf{a}}{\partial a_2} & \cdots & \frac{\partial \mathbf{a}}{\partial a_n} \end{bmatrix}^T \\ &= [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_n]^T = \mathbf{1} \end{aligned}$$

where,

$$\mathbf{a} = \sum_{i=1}^n a_i \mathbf{u}_i$$

and

$$\frac{\partial}{\partial \mathbf{a}} = \sum_{i=1}^n \frac{\partial}{\partial a_i} \mathbf{u}_i$$

Hence,

$$\frac{\partial \mathbf{a}}{\partial a_i} \equiv \frac{\partial a_i}{\partial \mathbf{a}} \equiv \mathbf{u}_i$$

Jacobian Matrix [b = A a]

$$\frac{\partial \mathbf{b}^T}{\partial \mathbf{a}} = \frac{\partial (\mathbf{a}^T \mathbf{A}^T)}{\partial \mathbf{a}} = \mathbf{A}^T$$

Jacobian Matrix [b = A<sup>T</sup> a]

$$\frac{\partial \mathbf{b}^T}{\partial \mathbf{a}} = \frac{\partial (\mathbf{a}^T \mathbf{A})}{\partial \mathbf{a}} = \mathbf{A}$$

Jacobian Matrix [b = c<sup>T</sup> A a]

$$\frac{\partial \mathbf{b}}{\partial \mathbf{a}} = \mathbf{A}^T \mathbf{c}$$

Jacobian Matrix [b = a<sup>T</sup> A c]

$$\frac{\partial \mathbf{b}}{\partial \mathbf{a}} = \mathbf{A} \mathbf{c}$$

Jacobian Matrix [b = a<sup>T</sup> A a]

$$\frac{\partial \mathbf{b}}{\partial \mathbf{a}} = \mathbf{A} \mathbf{a} + \mathbf{A}^T \mathbf{a} = (\mathbf{A} + \mathbf{A}^T) \mathbf{a}$$

Derivative of an Element of a Matrix w.r.t. A

$$\frac{\partial a_{ij}}{\partial A} = \begin{bmatrix} \cdot & & & & & \\ & \cdot & & & & \\ & & \cdot & & & \\ \dots & & & 1 & & \dots \\ & & & & \cdot & \\ & & & & & \cdot \\ & & & & & \cdot \\ & & & & & \cdot \\ & & & & & \cdot \end{bmatrix} \begin{array}{l} \\ \\ \\ \text{ith row} \\ \\ \\ \\ \\ \end{array}$$

jth col

$$= \mathbf{u}_i \mathbf{u}_j^T$$

The result is a matrix with zeros everywhere except at the intersection of row  $i$  and column  $j$ , which has a 1.

Derivative of A w.r.t. a<sub>ij</sub>

$$\frac{\partial A}{\partial a_{ij}} = \mathbf{u}_i \mathbf{u}_j^T \equiv \frac{\partial a_{ij}}{\partial A}$$

Derivative of  $A^{-1}$  where  $m = n$ 

We have, by definition

$$A^{-1} A = I$$

from which

$$\frac{\partial A^{-1}}{\partial \phi} = -A^{-1} \frac{\partial A}{\partial \phi} A^{-1}$$

Derivative of  $A^{-1}$  w.r.t.  $a_{ij}$ 

$$\begin{aligned} \frac{\partial A^{-1}}{\partial a_{ij}} &= -A^{-1} u_i u_j^T A^{-1} \\ &= -p_i q_j^T \end{aligned}$$

where  $p_i$  and  $q_j$  are solutions to

$$A p_i = u_i$$

$$A^T q_j = u_j$$



Derivatives of the Solution of  $\mathbf{Ax} = \mathbf{b}$

Let  $\mathbf{A}$  be  $n \times n$ ,  $\mathbf{x}$  and  $\mathbf{b}$  be  $n$ -dimensional. Then,

$$\begin{aligned}\frac{\partial \mathbf{x}}{\partial a_{ij}} &= \frac{\partial (\mathbf{A}^{-1} \mathbf{b})}{\partial a_{ij}} \\ &= -\mathbf{A}^{-1} \mathbf{u}_i \mathbf{u}_j^T \mathbf{A}^{-1} \mathbf{b} = -\mathbf{p}_i x_j\end{aligned}$$

Derivative of an Element of the Solution of  $\mathbf{Ax} = \mathbf{b}$

Let

$$\begin{aligned}\frac{\partial x_k}{\partial a_{ij}} &= \mathbf{u}_k^T \frac{\partial \mathbf{x}}{\partial a_{ij}} = -\mathbf{u}_k^T \mathbf{A}^{-1} \mathbf{u}_i \mathbf{u}_j^T \mathbf{A}^{-1} \mathbf{b} \\ &= -\hat{\mathbf{x}}_k^T \mathbf{u}_i \mathbf{u}_j^T \mathbf{x} \\ &= -\hat{x}_{ki} x_j\end{aligned}$$

Hence,

$$\frac{\partial x_k}{\partial \mathbf{A}} = -\hat{\mathbf{x}}_k \mathbf{x}^T$$

where  $\hat{\mathbf{x}}_k$  is the solution of

$$\mathbf{A}^T \hat{\mathbf{x}}_k = \mathbf{u}_k$$

Derivative of a Linear Combination of the Elements of the Solution of  $\mathbf{A} \mathbf{x} = \mathbf{b}$

$$\text{Let } \bar{x} = \mathbf{u}^T \mathbf{x}$$

Then

$$\begin{aligned} \frac{\partial \bar{x}}{\partial a_{ij}} &= \frac{\partial(\mathbf{u}^T \mathbf{x})}{\partial a_{ij}} = -\mathbf{u}^T \mathbf{p}_i \mathbf{q}_j^T \mathbf{b} \\ &= -(\mathbf{u}^T \mathbf{p}_i)(\mathbf{b}^T \mathbf{q}_j) \end{aligned}$$

The result comes from the appropriate linear combinations of  $\mathbf{p}_i$  and  $\mathbf{q}_j$ .

Alternatively,

$$\begin{aligned} \frac{\partial \bar{x}}{\partial a_{ij}} &= -\mathbf{q}^T \mathbf{u}_i \mathbf{u}_j^T \mathbf{x} \\ &= -q_i x_j \end{aligned}$$

where  $\mathbf{q}$  is the solution of

$$\mathbf{A}^T \mathbf{q} = \mathbf{u}$$

Hence

$$\frac{\partial \bar{x}}{\partial \mathbf{A}} = -\mathbf{q} \mathbf{x}^T$$

Also,

$$\frac{\partial \bar{x}}{\partial a_{ij}} = \mathbf{u}_i^T \frac{\partial \bar{x}}{\partial \mathbf{A}} \mathbf{u}_j$$

Hence

$$\frac{\partial \bar{x}}{\partial \phi} = \sum_{i=1}^n \sum_{j=1}^n \frac{\partial \bar{x}}{\partial a_{ij}} \frac{\partial a_{ij}}{\partial \phi} = \sum_{i=1}^n \sum_{j=1}^n \mathbf{u}_i^T \frac{\partial \bar{x}}{\partial \mathbf{A}} \mathbf{u}_j \frac{\partial a_{ij}}{\partial \phi}$$

Special Case 1

Let

$$\frac{\partial \mathbf{A}}{\partial \phi} \equiv \mathbf{u} \mathbf{u}_i^T - \mathbf{u} \mathbf{u}_j^T$$

Then

$$\begin{aligned} \frac{\partial \bar{x}}{\partial \phi} &= -\mathbf{u}^T \frac{\partial \mathbf{x}}{\partial \phi} = -\mathbf{u}^T \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \phi} \mathbf{A}^{-1} \mathbf{b} \\ &= -\mathbf{u}^T \mathbf{A}^{-1} \mathbf{u} (\mathbf{u}_i - \mathbf{u}_j)^T \mathbf{A}^{-1} \mathbf{b} \\ &= -\mathbf{u}^T \mathbf{p} (\mathbf{x}_i - \mathbf{x}_j) \end{aligned}$$

Let

$$\mathbf{u} = \mathbf{u}_i - \mathbf{u}_j$$

Then

$$\frac{\partial \bar{x}}{\partial \phi} = -(\mathbf{p}_i - \mathbf{p}_j) (\mathbf{x}_i - \mathbf{x}_j)$$

and

$$\frac{\partial A}{\partial \phi} = (\mathbf{u}_i - \mathbf{u}_j)(\mathbf{u}_i - \mathbf{u}_j)^T$$

$$= \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \dots & 1 & \dots & -1 & \dots \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \dots & -1 & \dots & 1 & \dots \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{array}{l} \leftarrow \text{ith row} \\ \leftarrow \text{jth row} \end{array}$$

$\begin{array}{cc} \uparrow & \uparrow \\ \text{ith col} & \text{jth col} \end{array}$

Special Case 2

Let

$$\frac{\partial \mathbf{A}}{\partial \phi} \equiv (\mathbf{u}_k - \mathbf{u}_\ell)(\mathbf{u}_k - \mathbf{u}_\ell)^T$$

Then

$$\begin{aligned} \frac{\partial \bar{\mathbf{x}}}{\partial \phi} &= -\mathbf{u}^T \mathbf{A}^{-1} (\mathbf{u}_k - \mathbf{u}_\ell) (\mathbf{u}_k - \mathbf{u}_\ell)^T \mathbf{A}^{-1} \mathbf{b} \\ &= -(\mathbf{q}_k - \mathbf{q}_\ell)(\mathbf{x}_k - \mathbf{x}_\ell) \end{aligned}$$

where  $\mathbf{q}$  is the solution to

$$\mathbf{A}^T \mathbf{q} = \mathbf{u}$$

and  $\mathbf{u}$  is the vector leading to  $\bar{\mathbf{x}} = \mathbf{u}^T \mathbf{x}$ .

Placing Elements Arbitrarily

Let

$$\mathbf{u}_0 \triangleq \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

$$\mathbf{u}_1 \triangleq \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

$$\mathbf{u}_2 \triangleq \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

$$\mathbf{u}_n \triangleq \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$$

and denote

$$\mathbf{u}_{i,j} \triangleq \mathbf{u}_i - \mathbf{u}_j$$

$$\mathbf{u}_{rs,kl} \triangleq \mathbf{u}_r + \mathbf{u}_s - \mathbf{u}_k - \mathbf{u}_l$$

hence

$$\mathbf{u}_{i_1 i_2 \dots i_p, j_1 j_2 \dots j_q} \triangleq \sum_{k=1}^p \mathbf{u}_{i_k} - \sum_{\ell=1}^q \mathbf{u}_{j_\ell}$$

For example,

$$\mathbf{u}_{345,16} = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}$$

Then if

$$\mathbf{u} \equiv \mathbf{u}_{345,16}$$

we have

$$\mathbf{u}^T \mathbf{u} = \text{card} \{3, 4, 5, 1, 6\} = 5$$

and

$$\mathbf{u} \mathbf{u}^T = \begin{array}{cccccc} \begin{bmatrix} 1 & 0 & -1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 1 & -1 \\ -1 & 0 & 1 & 1 & 1 & -1 \\ -1 & 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & -1 & -1 & -1 & 1 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} \\ \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} & \end{array}$$

$$= (\mathbf{u}_3 + \mathbf{u}_4 + \mathbf{u}_5 - \mathbf{u}_1 - \mathbf{u}_6)(\mathbf{u}_3 + \mathbf{u}_4 + \mathbf{u}_5 - \mathbf{u}_1 - \mathbf{u}_6)^T$$



**SECTION FOUR**  
**EXAMPLES AND PROBLEMS**

© J.W. Bandler 1988

This material is taken from previous years' assignments and examples. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.



## SIMPLE ANALYSIS AND OPTIMIZATION OF AN LC TRANSFORMER

*(Assignment 1, January 1985)*

Consider Question 168 of SECTION TWO. We wish to perform a least squares optimization at the sample points indicated starting at

$$L_1 = C_2 = C_4 = C_6 = 1$$

$$L_3 = 2 \quad L_5 = 3$$

Thus the objective function should be of the form

$$\sum_{i=1}^{21} |\rho_i|^2$$

Write and execute a Fortran program to solve this problem using a method of searching one variable (inductance, capacitance) at a time. Express the input impedance as a rational function of  $s = j\omega$ . Derive and use an efficient method of evaluating this impedance. Use Horner's Rule for evaluating polynomials. State the CPU time required for the optimization process.

A solution to this problem follows.

Consider a lumped-element LC transformer (Fig. 1) to match a 1 ohm load to a 3 ohm generator over the range 0.5 - 1.179 rad/s. Using 21 uniformly spaced sample points in the band, Minimize the objective function  $U$ , where

$$U = \sum_{i=1}^{21} |P_i|^2 \tag{1}$$

and  $p$  is the reflection coefficient. The solution is

- $L_1 = 0.968$
- $C_2 = 0.966$
- $L_3 = 2.282$
- $C_4 = 0.761$
- $L_5 = 2.897$
- $C_6 = 0.323$

at which  $U = 0.04534$ .

Use the method of "one variable at a time" for minimization.

Suggested starting point:

$$L_1 = C_2 = C_4 = C_6 = 1$$

$$L_3 = 2 \quad L_5 = 3$$

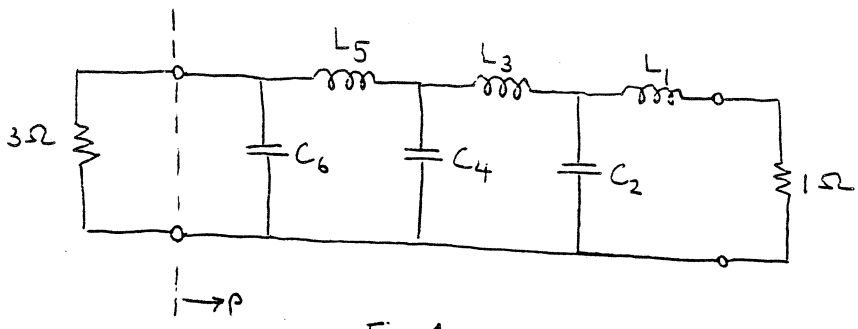


Fig. 1

Solution

The input impedance  $Z_{in}$  for the circuit of Fig. 1 can be obtained analytically by the appropriate combination of series and parallel impedances as:

$$Z_{in} = \frac{a_1 + a_2s + a_3s^2 + a_4s^3 + a_5s^4 + a_6s^5}{b_1 + b_2s + b_3s^2 + b_4s^3 + b_5s^4 + b_6s^5 + b_7s^6} \quad (2)$$

where

$$a_1 = 1$$

$$a_2 = L_1 + L_3 + L_5$$

$$a_3 = C_2(L_3 + L_5) + C_4L_5$$

$$a_4 = L_1 [C_2(L_3 + L_5) + C_4L_5] + L_3C_4L_5 \quad (3)$$

$$a_5 = C_2L_3C_4L_5$$

$$a_6 = L_1C_2L_3C_4L_5$$

$$b_1 = 1$$

$$b_2 = C_2 + C_4 + C_6$$

$$b_3 = L_1(C_2 + C_4 + C_6) + L_3(C_4 + C_6) + L_5C_6$$

$$b_4 = C_2L_3(C_4 + C_6) + L_5C_6(C_2 + C_4) \quad (4)$$

$$b_5 = L_1 [C_2L_3(C_4 + C_6) + L_5C_6(C_2 + C_4)] + L_3C_4L_5C_6$$

$$b_6 = C_2L_3C_4L_5C_6$$

$$b_7 = L_1C_2L_3C_4L_5C_6$$

and  $s = j\omega$ .

The reflection coefficient is then calculated as

$$\rho = \frac{Z_{in} - 3}{Z_{in} + 3} \quad (5)$$

In calculation of  $U$ , two factors should be taken into account for computational efficiency:

- 1) For a given set of parameter values, calculation of  $a_i$  and  $b_i$  coefficients, which is frequency independent, should be done with minimum operations. Subroutine COEFF is written based on the idea that common expressions, once calculated, should be saved to avoid recalculations. A more sophisticated subroutine may be written in a way that avoids recalculation of some  $a_i$  and  $b_i$  coefficients for a new set of parameter values. (In a new set, some parameters are unchanged.)
- 2) Evaluation of  $Z_{in}$  for different frequencies should be done efficiently. Horner's rule for polynomial evaluation is the most appropriate method for this purpose and it is used in Subroutine ZHORNER.

The method of one variable at a time for minimization is one of the most elementary optimization procedures. The method is based on the idea that, working with one variable at a time, we try to increase or decrease the value of the variable at hand, to get a decrease in the objective function. Once the right direction

has been found, we keep changing the variable in the right direction until we get an increase in the objective function. At this time, we switch to the next variable and repeat the procedure.

For physical realizability, the parameter values of the circuit should be +ve. To avoid -ve values in our minimization procedure, we can use transformation of variables in the following way:

$$X_E(I) = X(I) * X(I) \quad , \quad I=1, \dots, 6 \quad (6)$$

where  $\tilde{X}_E^T = [L_1 \ C_2 \ L_3 \ C_4 \ L_5 \ C_6]$  represents the actual parameter values and  $\tilde{X}$  is the vector on which optimization is performed.

```
0001      PROGRAM OVAAT
0002      REAL X(6),XE(6),DX(6),DXR(6)
0003      COMMON OMEG(21)
0004      C
0005      N=6
0006      C
0007      WRITE(6,*) ' SELECT STARTING VALUES FOR VARIABLES'
0008      READ(5,*) (XE(I),I=1,N)
0009      C
0010      DO 4 I=1,N
0011          X(I)=SQRT(XE(I))
0012      4 CONTINUE
0013      C
0014      WRITE(6,*) ' SELECT STARTING VALUES FOR STEP-LENGTHS'
0015      READ(5,*) (DXR(I),I=1,N)
0016      C
0017      WRITE(6,*) ' SELECT MAXIMUM NUMBER OF ITERATIONS'
0018      READ(5,*) MAXF
0019      C
0020      DO 5 I=1,21
0021          OMEG(I)=0.5+0.03395*FLOAT(I-1)
0022      5 CONTINUE
0023      C
0024      CALL FUN(N,X,U)
0025      C
0026      UPAST=U
0027      F2=UPAST
0028      C
```



```

0029      DO 20 J=1,MAXF
0030      C
0031          DO 15 I=1,N
0032              DX(I)=DXR(I)
0033          15 CONTINUE
0034      C
0035          DO 10 I=1,N
0036              INIT1=0
0037              INIT2=0
0038      C
0039          99      X(I)=X(I)+DX(I)
0040      C
0041          CALL FUN(N,X,U)
0042      C
0043          UPRES=U
0044      C
0045          IF (UPRES.LT.UPAST) THEN
0046              UPAST=UPRES
0047              INIT1=1
0048              GO TO 99
0049          ELSE
0050              IF (INIT1.EQ.0) THEN
0051                  X(I)=X(I)-DX(I)
0052                  IF (INIT2.EQ.0) THEN
0053                      DX(I)=-DX(I)
0054                      INIT2=1
0055                  ELSE
0056                      DX(I)=0.5*DX(I)
0057                      IF (ABS(DX(I)).LT.1.E-6) GO TO 10
0058                      INIT2=0
0059                  ENDIF
0060                  GO TO 99
0061              ELSE
0062                  X(I)=X(I)-DX(I)
0063                  GO TO 10
0064              ENDIF
0065          ENDIF
0066      10 CONTINUE
0067      C
0068          F1=F2
0069          F2=UPAST
0070      C
0071          IF ((F1-F2).LT.1.E-10) GO TO 98
0072      C
0073      20 CONTINUE
0074      C
0075          98      DO 25 I=1,N
0076              XE(I)=X(I)*X(I)
0077          25 CONTINUE
0078      C
0079          WRITE(6,55) J
0080          DO 30 I=1,N
0081              WRITE(6,50) I,XE(I)
0082          30 CONTINUE
0083          WRITE(6,51) UPAST
0084      C
0085          50      FORMAT(5X,'XE(',I1,')=',1PE19.12)
0086          51      FORMAT(//,5X,'THE OBJECTIVE FUNCTION=',1PE19.12,/)
0087          55      FORMAT(///,' THE SOLUTION REACHED AFTER ',I2,' ITERATIONS:',//)
0088          STOP
0089          END

```

```

0001      SUBROUTINE FUN(N,X,U)
0002      REAL X(N),XE(6),A(6),B(7)
0003      COMPLEX S,ZIN
0004      COMMON OMEG(21)
0005      C
0006      DO 5 I=1,N
0007          XE(I)=X(I)*X(I)
0008      5 CONTINUE
0009      C
0010      A(1)=1
0011      B(1)=1
0012      C
0013      CALL COEFF(XE,A,B)
0014      C
0015      U=0
0016      C
0017      DO 10 I=1,21
0018          S=CMLX(0.0,OMEG(I))
0019          CALL ZHORNER(S,A,B,ZIN)
0020          U=U+(CABS((ZIN-3)/(ZIN+3))**2)
0021      10 CONTINUE
0022      RETURN
0023      END

```

```

0001      SUBROUTINE COEFF(XE,A,B)
0002      REAL XE(6),A(6),B(7)
0003      C
0004      T1=XE(3)+XE(5)
0005      A(2)=XE(1)+T1
0006      T2=XE(4)*XE(5)
0007      A(3)=XE(2)*T1+T2
0008      T3=XE(3)*T2
0009      A(4)=XE(1)*A(3)+T3
0010      A(5)=XE(2)*T3
0011      A(6)=XE(1)*A(5)
0012      C
0013      T4=XE(4)+XE(6)
0014      B(2)=XE(2)+T4
0015      T5=XE(3)*T4
0016      T6=XE(5)*XE(6)
0017      B(3)=XE(1)*B(2)+T5+T6
0018      B(4)=XE(2)*T5+T6*(XE(2)+XE(4))
0019      B(5)=XE(1)*B(4)+XE(6)*T3
0020      B(6)=XE(6)*A(5)
0021      B(7)=XE(6)*A(6)
0022      C
0023      RETURN
0024      END

```

```

0001      SUBROUTINE ZHORNER(S,A,B,ZIN)
0002      REAL A(6),B(7)
0003      COMPLEX S,ZIN,ZN,ZD
0004      C
0005      ZN=CMPLX(A(6),0.0)
0006      ZD=CMPLX(B(7),0.0)
0007      C
0008      DO 10 I=6,1,-1
0009          ZD=S*ZD+B(I)
0010          IF(I.EQ.6)GO TO 10
0011          ZN=S*ZN+A(I)
0012      10  CONTINUE
0013      C
0014      ZIN=ZN/ZD
0015      RETURN
0016      END

```

```

SELECT STARTING VALUES FOR VARIABLES
Input : 1 1 2 1 3 1
SELECT STARTING VALUES FOR STEP-LENGTHS
Input : 0.1 0.1 0.1 0.1 0.1 0.1
SELECT MAXIMUM NUMBER OF ITERATIONS
Input : 50

```

THE SOLUTION REACHED AFTER 31 ITERATIONS:

```

XE(1)= 9.677668213844E-01
XE(2)= 9.659727215767E-01
XE(3)= 2.281047582626E+00
XE(4)= 7.611451745033E-01
XE(5)= 2.896585702896E+00
XE(6)= 3.230722248554E-01

```

THE OBJECTIVE FUNCTION= 4.534379392862E-02

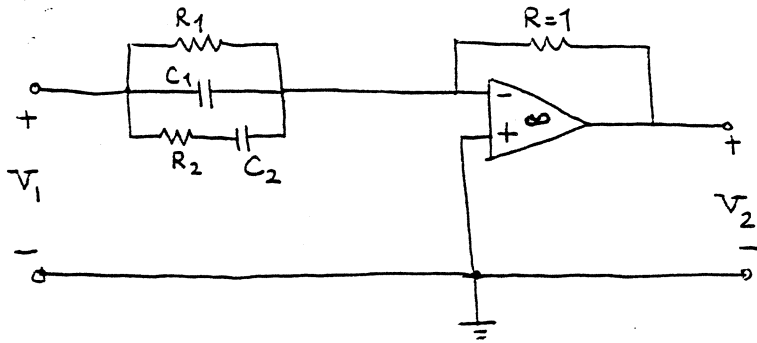
ONE AT A TIME OPTIMIZATION OF AN ACTIVE FILTER

*(Assignment 1, January 1986)*

Consider Question 169 of **SECTION TWO**.

Write and execute a Fortran program to solve this problem using a method of searching one variable at a time. State the CPU time required for the optimization process.

A solution to this problem follows.



$$\frac{V_2}{V_1} = - \left( \frac{1}{R_1} + sC_1 + \frac{1}{R_2 + \frac{1}{sC_2}} \right) \quad s = j2\pi f$$

$$\underline{\Phi} = [C_1 \ C_2 \ R_1 \ R_2]^T$$

$$G_{dB} = 20 \log_{10} \left| \frac{V_2}{V_1} \right|$$

Units: R in  $\Omega$

C in  $\mu F$

f in MHz

At frequency  $f_i$ :

Actual Gain:  $G_i \equiv G(f_i, \underline{\Phi})$

Specified Gain:  $SG_i = 5 + 5f_i$

$$\text{Minimize } U = \sum_{i=1}^{21} (G_i - SG_i)^p \quad p = 2, 4, 8, \dots$$

$$f_i = 1.0, 1.05, \dots, 2$$

If  $R_2$  and  $C_2$  not included

$$\frac{V_2}{V_1} = - \left( \frac{1}{R_1} + sC_1 \right)$$

$$\underline{\Phi} = [C_1 \ R_1]^T$$

```
PROGRAM EQUALIZ
REAL X(4),XE(4),DX(4),DXR(4),SPECG(21)
COMPLEX CFREQ(21)
LOGICAL FOUND_DIRECT,OPPOSIT_DIRECT_TRIED
CHARACTER*1 ANS
COMMON CFREQ,SPECG,NP
C
WRITE(*,*) ' Select the circuit: series RC included? Y/N'
READ(*, '(A)') ANS
IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
  N=4
ELSE
  N=2
ENDIF
C
WRITE(*,*) ' Select the value of p in least-pth optimization'
READ(*,*) NP
C
WRITE(*,*) ' Select starting values for variables'
READ(*,*) (XE(I),I=1,N)
C
DO 10 I=1,N
  X(I)=SQRT(XE(I))
10 CONTINUE
C
WRITE(*,*) ' Select starting values for step lengths'
READ(*,*) (DXR(I),I=1,N)
C
WRITE(*,*) ' Select maximum number of iterations'
READ(*,*) MAXF
C
PI2=8.0*ATAN(1.0)
DO 15 I=1,21
  FREQ=1.0+0.05*FLOAT(I-1)
  CFREQ(I)=CMPLX(0.0,PI2*FREQ)
  SPECG(I)=5.0+5.0*FREQ
15 CONTINUE
C
CALL FUN(N,X,U)
C
UPAST=U
F2=UPAST
C
DO 20 J=1,MAXF
C
  DO 25 I=1,N
    DX(I)=DXR(I)
25 CONTINUE
C
  DO 30 I=1,N
    FOUND_DIRECT=.FALSE.
    OPPOSIT_DIRECT_TRIED=.FALSE.
C
    X(I)=X(I)+DX(I)
C
```

```
CALL FUN(N,X,U)
C
UPRES=U
C
IF(UPRES.LT.UPAST)THEN
  UPAST=UPRES
  FOUND_DIRECT=.TRUE.
  GO TO 99
ELSE
  IF(.NOT.FOUND_DIRECT)THEN
    X(I)=X(I)-DX(I)
    IF(.NOT.OPPOSIT_DIRECT_TRIED)THEN
      DX(I)=-DX(I)
      OPPOSIT_DIRECT_TRIED=.TRUE.
    ELSE
      DX(I)=0.5*DX(I)
      IF(ABS(DX(I)).LT.1.E-6)GO TO 30
      OPPOSIT_DIRECT_TRIED=.FALSE.
    ENDIF
    GO TO 99
  ELSE
    X(I)=X(I)-DX(I)
    GO TO 30
  ENDIF
ENDIF
30 CONTINUE
C
F1=F2
F2=UPAST
C
IF((F1-F2).LT.1.E-10)GO TO 98
C
20 CONTINUE
C
98 DO 35 I=1,N
  XE(I)=X(I)*X(I)
35 CONTINUE
C
WRITE(*,50) J
DO 40 I=1,N
  WRITE(*,51) I,XE(I)
40 CONTINUE
WRITE(*,52) UPAST
C
50 FORMAT(///,' The solution reached after ',I2,' iterations:',///)
51 FORMAT(5X,'XE(',I1,')=',1PE19.12)
52 FORMAT(//,5X,'The objective function=',1PE19.12,///)
C
STOP
END
```

```

SUBROUTINE FUN(N,X,U)
REAL X(N),XE(4),SPECG(21)
COMPLEX CFREQ(21),G,G1
COMMON CFREQ,SPECG,NP
C
DO 5 I=1,N
  XE(I)=X(I)*X(I)
5 CONTINUE
C
U=0.0
DO 10 I=1,21
  IF(N.EQ.2)THEN
    G=CFREQ(I)*XE(1)+1.0/XE(2)
  ELSE
    G1=1.0/(XE(4)+1.0/(CFREQ(I)*XE(2)))
    G=G1+CFREQ(I)*XE(1)+1.0/XE(3)
  ENDIF
C
  U=U+(20.0*ALOG10(CABS(G))-SPECG(I))*NP
10 CONTINUE
RETURN
END
```



Select the circuit: series RC included? Y/N  
Input : N  
Select the value of p in least-pth optimization  
Input : 2  
Select starting values for variables  
Input : 1 1  
Select starting values for step lengths  
Input : 0.1 0.1  
Select maximum number of iterations  
Input : 50

The solution reached after 26 iterations:

XE(1)= 4.163925945759E-01  
XE(2)= 5.967150330544E-01

The objective function= 1.855218261480E-01

Select the circuit: series RC included? Y/N  
Input : N  
Select the value of p in least-pth optimization  
Input : 2  
Select starting values for variables  
Input : 10 10  
Select starting values for step lengths  
Input : 1 1  
Select maximum number of iterations  
Input : 50

The solution reached after 23 iterations:

XE(1)= 4.164692163467E-01  
XE(2)= 5.971335768700E-01

The objective function= 1.855216473341E-01

Select the circuit: series RC included? Y/N  
Input : Y  
Select the value of p in least-pth optimization  
Input : 2  
Select starting values for variables  
Input : 1 1 1 1  
Select starting values for step lengths

Input : 0.1 0.1 0.1 0.1  
Select maximum number of iterations  
Input : 50

The solution reached after 51 iterations:

XE(1)= 3.183872699738E-01  
XE(2)= 9.814125299454E-02  
XE(3)= 5.974499583244E-01  
XE(4)= 8.021529538382E-06

The objective function= 1.855301260948E-01

Select the circuit: series RC included? Y/N  
Input : Y  
Select the value of p in least-pth optimization  
Input : 2  
Select starting values for variables  
Input : 1 1 1 10000  
Select starting values for step lengths  
Input : 0.1 0.1 0.1 10  
Select maximum number of iterations  
Input : 80

The solution reached after 21 iterations:

XE(1)= 4.163728952408E-01  
XE(2)= 1.000000000000E+00  
XE(3)= 5.965453386307E-01  
XE(4)= 1.278064218750E+05

The objective function= 1.855238080025E-01

Select the circuit: series RC included? Y/N  
Input : Y  
Select the value of p in least-pth optimization  
Input : 2  
Select starting values for variables  
Input : 0.5 0.5 0.5 0.5  
Select starting values for step lengths  
Input : 0.1 0.1 0.1 0.1  
Select maximum number of iterations  
Input : 80

The solution reached after 81 iterations:

XE(1)= 2.428505420685E-01  
XE(2)= 1.729622483253E-01  
XE(3)= 5.966239571571E-01  
XE(4)= 4.713593982160E-03

The objective function= 1.856064051390E-01

Select the circuit: series RC included? Y/N  
Input : N  
Select the value of p in least-pth optimization  
Input : 4  
Select starting values for variables  
Input : 1 1  
Select starting values for step lengths  
Input : 0.1 0.1  
Select maximum number of iterations  
Input : 50

The solution reached after 17 iterations:

XE(1)= 4.182498455048E-01  
XE(2)= 6.021538972855E-01

The objective function= 3.043847624213E-03

Select the circuit: series RC included? Y/N  
Input : N  
Select the value of p in least-pth optimization  
Input : 4  
Select starting values for variables  
Input : 5 5  
Select starting values for step lengths  
Input : 0.1 0.1  
Select maximum number of iterations  
Input : 50

The solution reached after 15 iterations:

XE(1)= 4.182606935501E-01  
XE(2)= 6.021808385849E-01

The objective function= 3.043899079785E-03

Select the circuit: series RC included? Y/N  
Input : Y  
Select the value of p in least-pth optimization  
Input : 4  
Select starting values for variables  
Input : 1 1 1 1  
Select starting values for step lengths  
Input : 0.1 0.1 0.1 0.1  
Select maximum number of iterations  
Input : 80

The solution reached after 56 iterations:

XE(1)= 3.231257498264E-01  
XE(2)= 9.512175619602E-02  
XE(3)= 6.021496057510E-01  
XE(4)= 3.431117647779E-07

The objective function= 3.043843433261E-03

Select the circuit: series RC included? Y/N  
Input : Y  
Select the value of p in least-pth optimization  
Input : 4  
Select starting values for variables  
Input : 0.5 0.5 0.5 0.5  
Select starting values for step lengths  
Input : 0.1 0.1 0.1 0.1  
Select maximum number of iterations  
Input : 80

The solution reached after 81 iterations:

XE(1)= 2.059222012758E-01  
XE(2)= 2.107012420893E-01  
XE(3)= 6.021153330803E-01

XE(4)= 9.286314249039E-03

The objective function= 3.062061034143E-03

Select the circuit: series RC included? Y/N  
Input : N  
Select the value of p in least-pth optimization  
Input : 8  
Select starting values for variables  
Input : 1 1  
Select starting values for step lengths  
Input : 0.1 0.1  
Select maximum number of iterations  
Input : 50

The solution reached after 12 iterations:

XE(1)= 4.193002879620E-01  
XE(2)= 6.048710942268E-01

The objective function= 1.106618583435E-06

Select the circuit: series RC included? Y/N  
Input : N  
Select the value of p in least-pth optimization  
Input : 8  
Select starting values for variables  
Input : 0.5 0.5  
Select starting values for step lengths  
Input : 0.1 0.1  
Select maximum number of iterations  
Input : 50

The solution reached after 19 iterations:

XE(1)= 4.192835390568E-01  
XE(2)= 6.049123406410E-01

The objective function= 1.106388253902E-06

Select the circuit: series RC included? Y/N

Input : Y  
Select the value of p in least-pth optimization  
Input : 8  
Select starting values for variables  
Input : 1 1 1 1  
Select starting values for step lengths  
Input : 0.1 0.1 0.1 0.1  
Select maximum number of iterations  
Input : 90

The solution reached after 20 iterations:

XE(1)= 3.495447635651E-01  
XE(2)= 6.975876539946E-02  
XE(3)= 6.050227880478E-01  
XE(4)= 7.384940545307E-05

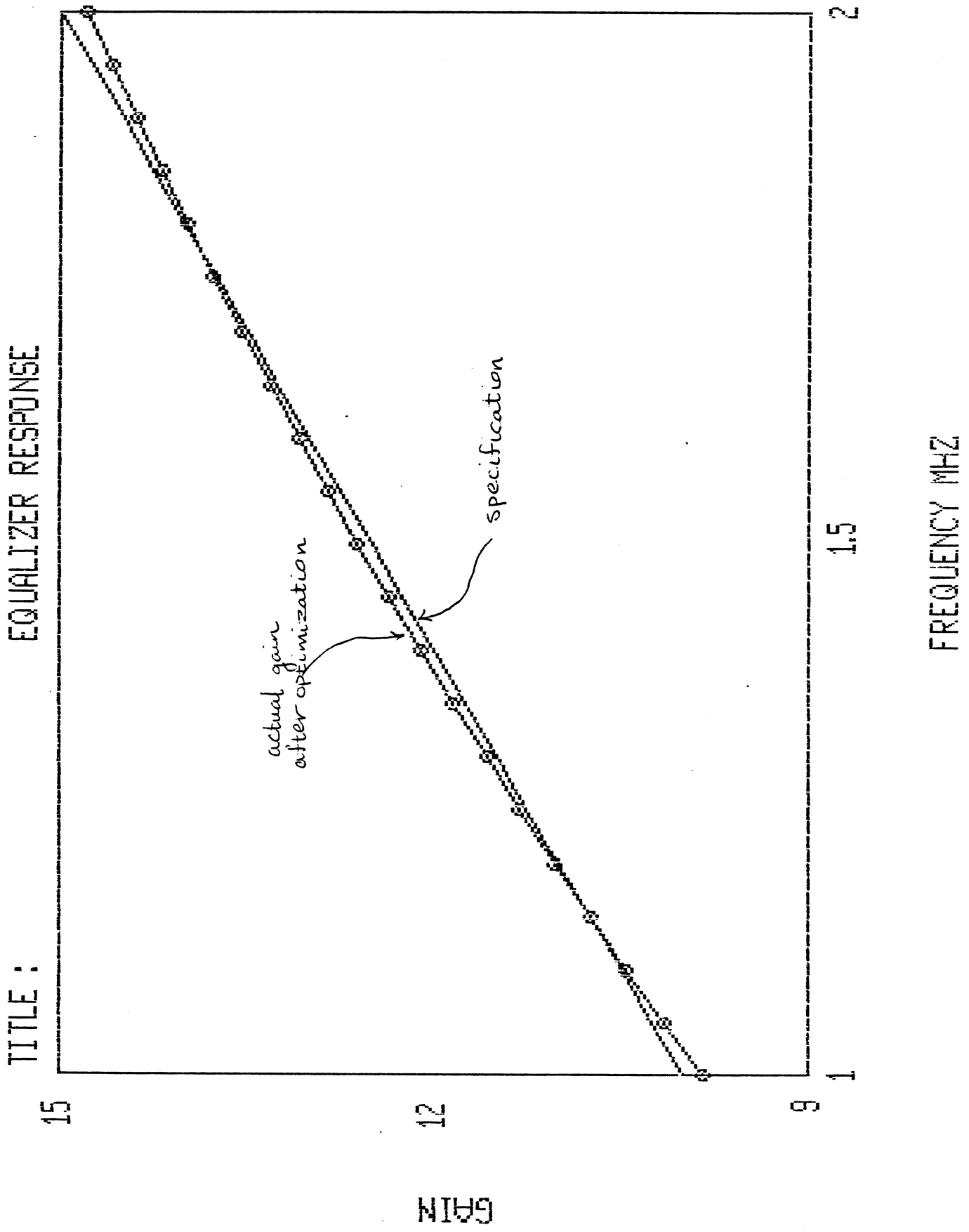
The objective function= 1.106466811507E-06

Select the circuit: series RC included? Y/N  
Input : Y  
Select the value of p in least-pth optimization  
Input : 8  
Select starting values for variables  
Input : 0.5 0.5 0.5 0.5  
Select starting values for step lengths  
Input : 0.1 0.1 0.1 0.1  
Select maximum number of iterations  
Input : 90

The solution reached after 91 iterations:

XE(1)= 1.801922470331E-01  
XE(2)= 2.358155697584E-01  
XE(3)= 6.066607236862E-01  
XE(4)= 1.690840721130E-02

The objective function= 1.166657852991E-06



TITLE :

EQUALIZER RESPONSE

GAIN

FREQUENCY MHZ

## RESPONSE CALCULATION AND CONTOUR PLOTTING

*(Assignment 2, January 1985)*

Consider Question 106 of **SECTION TWO**.

Derive an analytical formula for the insertion loss of the filter. Create error functions using the specifications of Question 106 at the following frequency points: 0.45, 0.50, 0.55, 1.0 and 2.5 rad/s.

Write an efficient Fortran program for drawing contours of the minimum objective function of the problem using the CNTOUR subroutine in the VPLOT library on the VAX system. Take  $L_1$  and  $C$  as variables varying from 1.00 to 3.00 for  $L_1$  and from 0.50 to 1.50 for  $C$ . Keep  $L_2$  at the optimal value 1.62410. Select values of the contours to be plotted to result in a meaningful diagram. Select a few sets of appropriate weighing factors to display the effects of varying them from 1.0.

A solution to this problem follows.



Consider the LC lowpass filter shown in Fig.1 and the corresponding specifications.

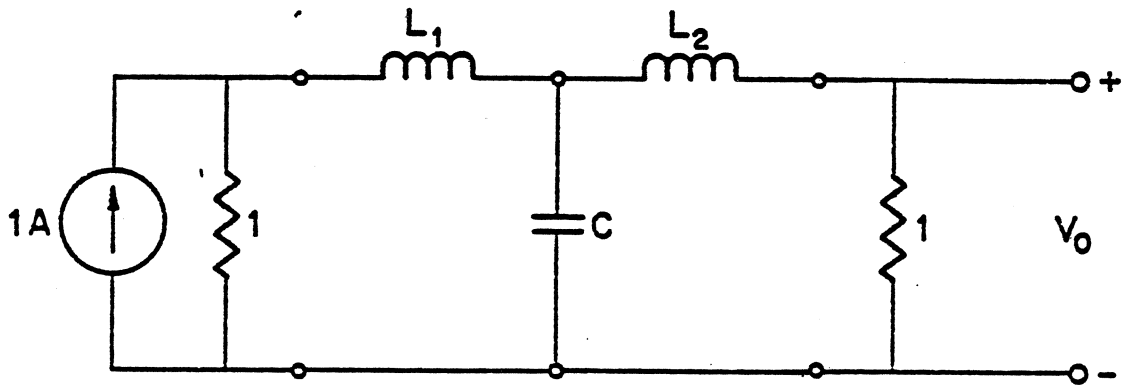


Fig.1 LC lowpass filter with excitation.

Specifications	
Frequency Range (rad/s)	Insertion Loss (dB)
0 - 1	< 1.5
> 2.5	> 25

The insertion loss between the source and the load is defined as

$$\text{insertion loss} \triangleq \frac{\text{power to the load when the circuit is removed}}{\text{power to the load when the circuit is in place}} \quad (1)$$

When the circuit is removed we have, for source and load resistances  $R_S$  and  $R_L$ ,

$$P_{L0} = |V_I / (R_S + R_L)|^2 R_L, \quad (2)$$

where  $V_I$  is the source voltage, and with the circuit in place, we get

$$P_0 = |V_0 / R_L|^2 R_L, \quad (3)$$

where  $V_0$  is the voltage across  $R_L$ . Substituting (2) and (3) into (1) gives

$$\begin{aligned} \text{insertion loss (in dB)} &= 10 \log_{10} \left\{ \frac{|V_I / (R_S + R_L)|^2 R_L}{|V_0 / R_L|^2 R_L} \right\} \\ &= 20 \log_{10} \frac{R_L}{R_L + R_S} \left| \frac{V_I}{V_0} \right|. \end{aligned} \quad (4)$$

For  $R_S = R_L = 1 \Omega$  and  $V_I = 1$  volt (corresponding to a 1 ampere excitation), (4) becomes

$$\begin{aligned} \text{insertion loss (in dB)} &= 20 \log_{10} (1/2) + 20 \log_{10} (1/|V_0|) = \\ &= -6.0206 - 8.6859 \ln |V_0|, \end{aligned} \quad (5)$$

where  $20 \log_{10} e = 8.6859$ .

The output voltage  $V_0$  for the circuit of Fig. 1 can be expressed as

$$V_0 = \frac{1}{CL_1L_2s^3 + C(L_1+L_2)s^2 + (L_1+L_2+C)s + 2} \quad (6)$$

$(s = j\omega)$

Fig. 2 illustrates the specifications for the LC filter.

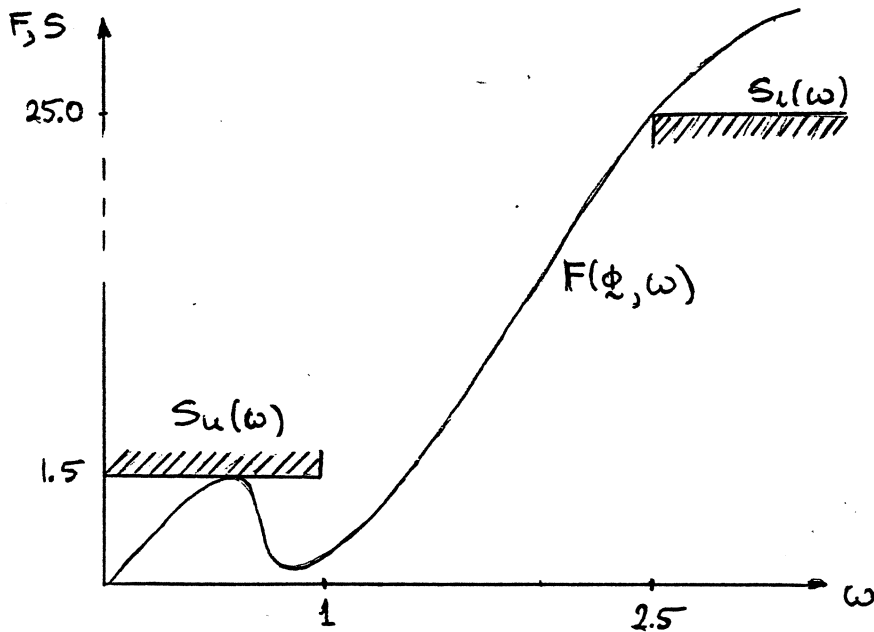


Fig. 2 Graphical representation of the specifications for the LC filter.

Assuming the approximating function and the specifications are real, let the error functions  $e_{u_i}$  and  $e_{L_i}$  be

$$e_{u_i} \triangleq w_u(\omega_i) [F(\phi, \omega_i) - S_u(\omega_i)], \quad i \in I_u \triangleq \{1, \dots, 10\},$$

$$e_{L_i} \triangleq w_L(\omega_i) [F(\phi, \omega_i) - S_L(\omega_i)], \quad i \in I_L \triangleq \{11\},$$

4

where  $\underline{\phi}^T = [L_1 \ L_2 \ C]$ ,  $\omega_1 = 0.1$ ,  $\omega_2 = 0.2$ , ...,  $\omega_{10} = 1.0$

and  $\omega_{11} = 2.5$  rad/s.

The minimax objective function of the problem is

$$M \triangleq \max_{j \in J} f_j, \quad J \triangleq \{1, 2, \dots, 11\}$$

where

$$f_j \triangleq \begin{cases} e_{ui}, & i \in I_u \\ -e_{li}, & i \in I_l. \end{cases}$$

Note: Calculation of derivatives is performed as follows.

From (5),

$$IL \text{ (in dB)} = C_1 + C_2 \ln |V_0|$$

$$C_1 = -6.0206 \quad C_2 = -8.6859$$

$$\frac{\partial(IL)}{\partial \phi} = C_2 \frac{1}{|V_0|} \frac{\partial |V_0|}{\partial \phi}$$

$$= C_2 \frac{1}{|V_0|} \frac{\partial}{\partial \phi} [(V_0 V_0^*)^{1/2}]$$

$$= C_2 \frac{1}{2|V_0|^2} \left[ \left( \frac{\partial V_0}{\partial \phi} \right) V_0^* + V_0 \frac{\partial V_0^*}{\partial \phi} \right]$$

$$= C_2 \frac{1}{2|V_0|^2} \left[ 2 \operatorname{Re} \left\{ V_0^* \frac{\partial V_0}{\partial \phi} \right\} \right]$$

$$= C_2 \operatorname{Re} \left\{ \frac{V_0^*}{|V_0|^2} \frac{\partial V_0}{\partial \phi} \right\}$$

$$= C_2 \operatorname{Re} \left\{ \frac{1}{V_0} \frac{\partial V_0}{\partial \phi} \right\}$$

where  $\frac{\partial V_0}{\partial \phi}$  can be calculated by direct differentiation of (6).

```
PROGRAM Q106
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 X(3),W(200),C(3),DC(3,3)
C
COMMON WL,WU,SPECL,SPECU
C
EXTERNAL FDF
C
N=3
M=11
L=3
LEQ=0
IC=3
C
DO 5 I=1,3
  C(I)=-0.01
  DO 10 J=1,3
    DC(I,J)=0.0
10  CONTINUE
    DC(I,I)=1.0
5  CONTINUE
C
WRITE(*,*) 'TYPE THE STARTING VALUES FOR L1,L2,C'
READ(*,*) (X(I),I=1,N)
C
WRITE(*,*) 'TYPE WEIGHTING FACTORS: PASSBAND AND STOPBAND'
READ(*,*) WU,WL
C
SPECU=1.5
SPECL=25.0
C
DX=0.1
EPS=1.E-6
MAXF=50
KEQS=3
IW=200
ICH=6
IPR=-10
C
CALL MMLC1A(FDF,N,M,L,LEQ,C,DC,IC,X,DX,EPS,MAXF,KEQS,W,IW,
+          ICH,IPR,IFALL)
C
STOP
END
```

```
SUBROUTINE FDF(N,M,X,DF,F)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 X(N),F(M),DF(M,N),IL,DILDX(3)
C
COMMON WL,WU,SPECL,SPECU
C
DO 10 I=1,10
  OMEGA=FLOAT(I)/10.0
  CALL FILTER(OMEGA,X,IL,DILDX)
  F(I)=WU*(IL-SPECU)
  DO 20 J=1,3
    DF(I,J)=WU*DILDX(J)
20  CONTINUE
10  CONTINUE
C
OMEGA=2.5
CALL FILTER(OMEGA,X,IL,DILDX)
F(M)=-WL*(IL-SPECL)
DO 30 J=1,3
  DF(M,J)=-WL*DILDX(J)
30  CONTINUE
C
RETURN
END
```

```
      SUBROUTINE FILTER(OMEGA,X,IL,DILDX)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      REAL*8 X(3),L1,L2,IL,DILDX(3)
      COMPLEX*16 S,VOUT,DVOUT(3),DENOM,CF
C
      S=DCMPLX(0.D0,OMEGA)
C
      L1=X(1)
      L2=X(2)
      C=X(3)
C
      T1=L1+L2+C
      T2=C*(L1+L2)
      T3=C*L1*L2
C
      DENOM=2.0+S*(T1+S*(T2+S*T3))
      VOUT=1.0/DENOM
C
      CF=-S/(DENOM*DENOM)
C
      DVOUT(1)=CF*(1.0+S*C*(1.0+S*L2))
      DVOUT(2)=CF*(1.0+S*C*(1.0+S*L1))
      DVOUT(3)=CF*(1.0+S*((L1+L2)+S*L1*L2))
C
      CONST1=-6.0206
      CONST2=-8.6859
      IL=CONST1+CONST2*DLOG(CDABS(VOUT))
C
      DO 10 I=1,3
         DILDX(I)=CONST2*DREAL(DVOUT(I)/VOUT)
10    CONTINUE
C
      RETURN
      END
```

TYPE THE STARTING VALUES FOR L1,L2,C

Input : 1 1 1

TYPE WEIGHTING FACTORS: PASSBAND AND STOPBAND

Input : 1 1

INPUT DATA

NUMBER OF VARIABLES (N) . . . . . 3  
 NUMBER OF FUNCTIONS (M) . . . . . 11  
 TOTAL NUMBER OF LINEAR CONSTRAINTS (L) . . . . . 3  
 NUMBER OF EQUALITY CONSTRAINTS (LEQ) . . . . . 0  
 STEP LENGTH (DX) . . . . . 1.000E-01  
 ACCURACY (EPS) . . . . . 1.000E-06  
 MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . 50  
 NUMBER OF SUCCESSIVE ITERATIONS (KEQS) . . . . . 3  
 WORKING SPACE (IW) . . . . . 200  
 PRINTOUT CONTROL (IPR) . . . . . -10

STARTING POINT :

VARIABLES		FUNCTION VALUES	
1	1.000000000000E+00	1	-1.489364606509E+00
2	1.000000000000E+00	2	-1.460151584189E+00
3	1.000000000000E+00	3	-1.419818585015E+00
		4	-1.379115391860E+00
		5	-1.349933639110E+00
		6	-1.342774758348E+00
		7	-1.363775493994E+00
		8	-1.410858484467E+00
		9	-1.468360322208E+00
		10	-1.499992904374E+00
		11	8.558896951669E+00

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.



SOLUTION

VARIABLES		FUNCTION VALUES	
1	1.627849368463E+00	1	-1.450689889784E+00
2	1.627849368463E+00	2	-1.321146099504E+00
3	1.089802755027E+00	3	-1.158772337170E+00
		4	-1.022873382979E+00
		5	-9.680223979380E-01
		6	-1.030506063607E+00
		7	-1.209965996010E+00
		8	-1.430432825282E+00
		9	-1.473669168301E+00
		10	-9.680223979373E-01
		11	-9.680223979374E-01

TYPE THE STARTING VALUES FOR L1,L2,C

Input : 1 1 1

TYPE WEIGHTING FACTORS: PASSBAND AND STOPBAND

Input : 100 1

SOLUTION

VARIABLES		FUNCTION VALUES	
1	2.340969647725E+00	1	-1.352634560969E+02
2	2.340969647725E+00	2	-9.804282942812E+01
3	9.162121046666E-01	3	-5.410327605641E+01
		4	-1.933544919235E+01
		5	-5.764892851446E+00
		6	-2.124407240361E+01
		7	-6.765726093819E+01
		8	-1.293037683637E+02
		9	-1.420902033997E+02
		10	-5.764892851446E+00
		11	-5.764892851446E+00

TYPE THE STARTING VALUES FOR L1,L2,C

Input : 1 1 1

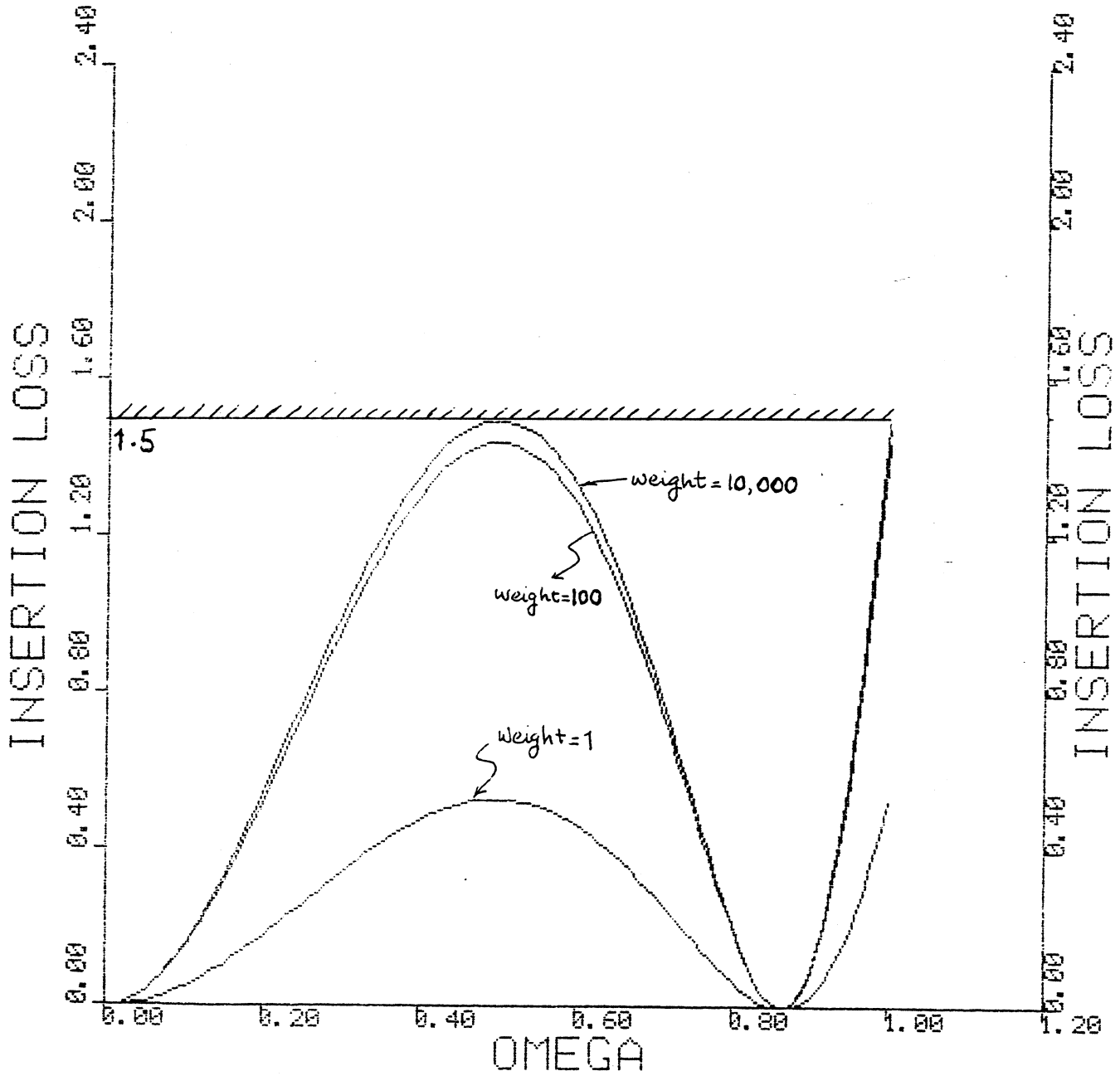
TYPE WEIGHTING FACTORS: PASSBAND AND STOPBAND

Input : 10000 1

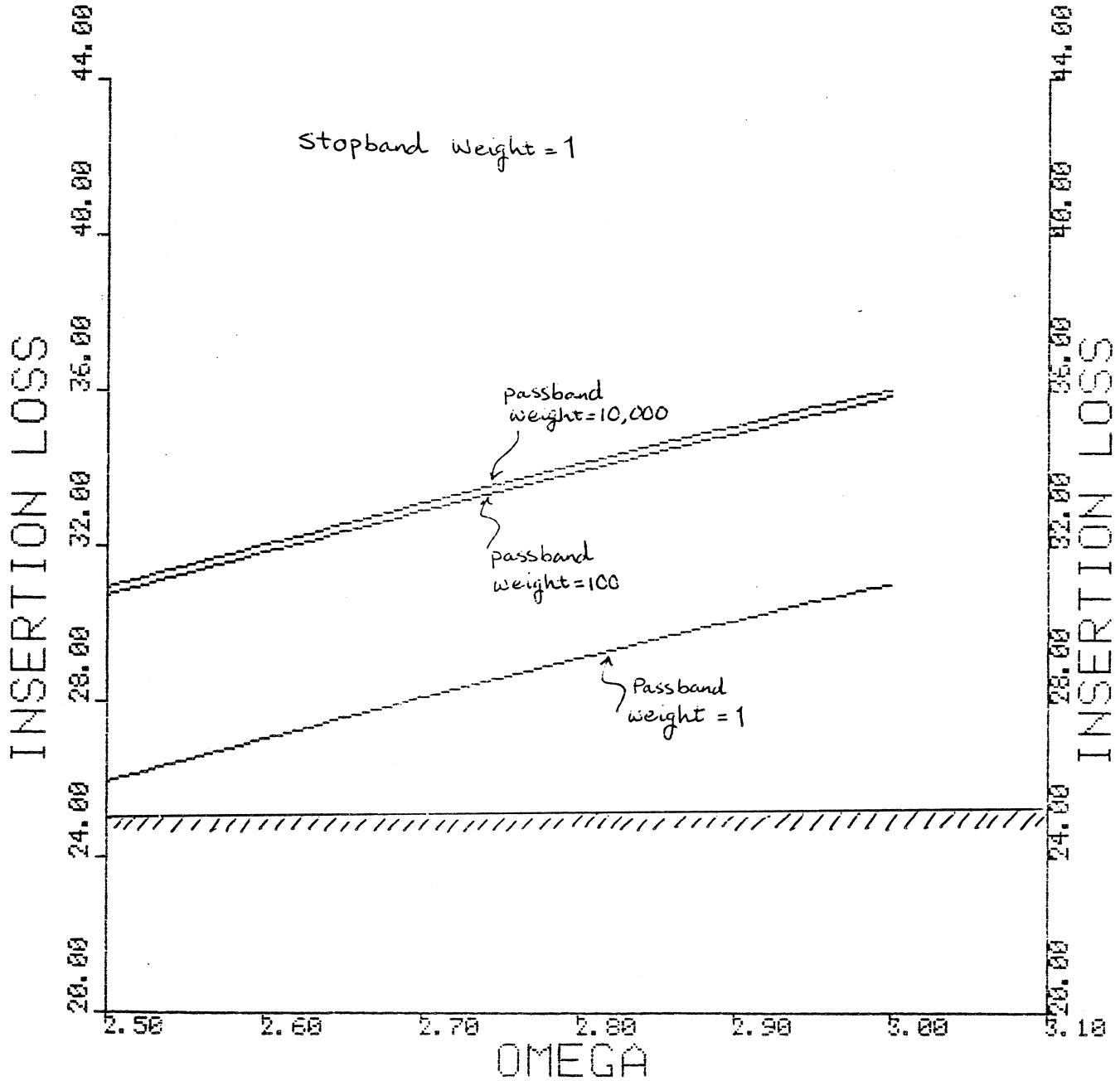
SOLUTION

VARIABLES		FUNCTION VALUES	
1	2.379904294816E+00	1	-1.345860108107E+04
2	2.379904294816E+00	2	-9.575742819672E+03
3	9.069829220882E-01	3	-5.009539905772E+03
		4	-1.408714629494E+03
		5	-5.963340382527E+00
		6	-1.606123025147E+03
		7	-6.416153635036E+03
		8	-1.283590928844E+04
		9	-1.417236140474E+04
		10	-5.963340382525E+00
		11	-5.963340382525E+00

# PASSBAND DETAIL



STOPBAND



**SECTION FIVE**  
**GAUSS ELIMINATION**

© J.W. Bandler 1984, 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



## GAUSS ELIMINATION

### Basic Problem

We have  $n$  linear simultaneous equations in  $n$  unknowns, namely,

$$a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n = b_1 \quad (1)$$

$$a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n = b_2 \quad (2)$$

$$\vdots \quad \vdots$$

$$a_{n1} x_1 + a_{n2} x_2 + \dots + a_{nn} x_n = b_n \quad (n)$$

In matrix notation

$$\underline{A} \underline{x} = \underline{b} \quad ,$$

where

$$\underline{A} \triangleq \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad ,$$

$$\underline{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad ,$$

$$\underline{b} \triangleq \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad .$$

Note that

$a_{ij}$ ,  $j = 1, 2, \dots, n$  generates the  $i$ th row,

$a_{ij}$ ,  $i = 1, 2, \dots, n$  generates the  $j$ th column,

$a_{ij}$ ,  $i, j = 1, 2, \dots, n$  generates the matrix  $\underline{A}$ .

The Gauss elimination algorithm is based on the following concepts.

Forward Reduction

Divide (1) by  $a_{11}$  to give

$$x_1 + \frac{a_{12}}{a_{11}} x_2 + \dots + \frac{a_{1n}}{a_{11}} x_n = \frac{b_1}{a_{11}} \quad .$$

Multiply this equation by  $a_{21}$  to give

$$a_{21} x_1 + a_{21} \frac{a_{12}}{a_{11}} x_2 + \dots + a_{21} \frac{a_{1n}}{a_{11}} x_n = a_{21} \frac{b_1}{a_{11}} \quad .$$

Subtract this equation from (2) to give

$$0 x_1 + \left[ a_{22} - a_{21} \frac{a_{12}}{a_{11}} \right] x_2 + \dots + \left[ a_{2n} - a_{21} \frac{a_{1n}}{a_{11}} \right] x_n = b_2 - a_{21} \frac{b_1}{a_{11}} \quad .$$

If we had multiplied the first equation by  $a_{i1}$  and subtracted the result from the  $i$ th equation we would have had

$$0 x_1 + \left[ a_{i2} - a_{i1} \frac{a_{12}}{a_{11}} \right] x_2 + \dots + \left[ a_{in} - a_{i1} \frac{a_{1n}}{a_{11}} \right] x_n = b_i - a_{i1} \frac{b_1}{a_{11}} \quad .$$

Writing out the new system we have

$$\begin{bmatrix} 1 & \frac{a_{12}}{a_{11}} & \dots & \frac{a_{1n}}{a_{11}} \\ 0 & a_{22} - a_{21} \frac{a_{12}}{a_{11}} & \dots & a_{2n} - a_{21} \frac{a_{1n}}{a_{11}} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2} - a_{n1} \frac{a_{12}}{a_{11}} & \dots & a_{nn} - a_{n1} \frac{a_{1n}}{a_{11}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \frac{b_1}{a_{11}} \\ b_2 - a_{21} \frac{b_1}{a_{11}} \\ \vdots \\ b_n - a_{n1} \frac{b_1}{a_{11}} \end{bmatrix}$$

The lower right partition involves  $n-1$  equations in  $n-1$  unknowns. Let the whole system be represented as originally but with superscript  $i$  to



distinguish the results of different iterations. We now have

$$a_{ij}^2 = a_{ij}^1 - a_{i1}^1 \frac{a_{1j}^1}{a_{11}^1} \quad , \quad i = 2, 3, \dots, n$$
$$b_i^2 = b_i^1 - a_{i1}^1 \frac{b_1^1}{a_{11}^1} \quad , \quad i = 2, 3, \dots, n$$
$$, \quad j = 1, 2, \dots, n$$

Check: for  $j = 1$  we have  $a_{i1}^2 = a_{i1}^1 - a_{i1}^1 = 0$ . Here, we have eliminated  $x_1$ .

The same approach is repeated until  $x_2$  is eliminated with  $n-2$  equations remaining. In this case we have

$$a_{ij}^3 = a_{ij}^2 - a_{i2}^2 \frac{a_{2j}^2}{a_{22}^2} \quad , \quad i = 3, 4, \dots, n$$
$$b_i^3 = b_i^2 - a_{i2}^2 \frac{b_2^2}{a_{22}^2} \quad , \quad i = 3, 4, \dots, n$$
$$, \quad j = 2, 3, \dots, n$$

When the  $k$ th variable is eliminated we have

$$a_{ij}^{k+1} = a_{ij}^k - a_{ik}^k \frac{a_{kj}^k}{a_{kk}^k} \quad , \quad i = k+1, k+2, \dots, n$$
$$b_i^{k+1} = b_i^k - a_{ik}^k \frac{b_k^k}{a_{kk}^k} \quad , \quad i = k+1, k+2, \dots, n$$
$$, \quad j = k, k+1, \dots, n$$

The final result is

$$\begin{bmatrix} 1 & \frac{a_{12}^1}{a_{11}^1} & \frac{a_{13}^1}{a_{11}^1} & \dots & \frac{a_{1n}^1}{a_{11}^1} \\ 0 & 1 & \frac{a_{23}^2}{a_{22}^2} & \dots & \frac{a_{2n}^2}{a_{22}^2} \\ 0 & 0 & 1 & \dots & \frac{a_{3n}^3}{a_{33}^3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \frac{b_1}{a_{11}^1} \\ \frac{b_2}{a_{22}^2} \\ \frac{b_3}{a_{33}^3} \\ \vdots \\ \frac{b_n}{a_{nn}^n} \end{bmatrix}$$

Thus, the original matrix has been transformed to upper-triangular form.

Back Substitution

We have obtained a system of the form

$$\begin{bmatrix} 1 & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & 1 & u_{23} & \dots & u_{2n} \\ 0 & 0 & 1 & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Now

$$x_n = b_n \quad ,$$

$$x_{n-1} = b_{n-1} - u_{n-1,n} x_n \quad ,$$

$$x_{n-2} = b_{n-2} - (u_{n-2,n-1} x_{n-1} + u_{n-2,n} x_n) \quad ,$$

$\vdots$

and, in general,

$$x_i = b_i - \sum_{j=i+1}^n u_{ij} x_j .$$

Multiple Right Hand Side Vectors

If  $\tilde{A}$  stays the same for different problems, the upper triangular form obtained by forward reduction is unchanged. We need, however, the terms (multipliers)

$$m_{ik} = \frac{a_{ik}^k}{a_{kk}^k} , i = k+1, k+2, \dots, n$$

i.e.,

$$m_{i1} = \frac{a_{i1}^1}{a_{11}^1} , i = 2, 3, \dots, n \text{ (hence: } m_{21}, m_{31}, \dots, m_{n1})$$

$$m_{i2} = \frac{a_{i2}^2}{a_{22}^2} , i = 3, 4, \dots, n \text{ (hence: } m_{32}, m_{42}, \dots, m_{n2})$$

⋮

To save storage we reuse the available matrix locations:

$$\begin{bmatrix} a_{11}^1 & a_{12}^1 & a_{13}^1 & \dots & a_{1n}^1 \\ m_{21} & a_{22}^2 & a_{23}^2 & \dots & a_{2n}^2 \\ m_{31} & m_{32} & a_{33}^3 & \dots & a_{3n}^3 \\ \vdots & \vdots & \vdots & & \vdots \\ m_{n1} & m_{n2} & m_{n3} & \dots & a_{nn}^n \end{bmatrix} , \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} , \begin{bmatrix} b_1^1 \\ b_2^2 \\ b_3^3 \\ \vdots \\ b_n^n \end{bmatrix} ,$$

since the number of multipliers is the same as the number of zero elements. Then we need only the calculations for  $b_i^{k+1}$ . Note: in practice only one variable  $m$  is needed at any time.

Matrix Inversion

By definition,

$$\underline{A} \underline{A}^{-1} = \underline{1} \quad ,$$

where

$$\underline{1} \triangleq \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} .$$

Suppose we want  $\underline{A}^{-1}$ . Let

$$\underline{X} \triangleq [\underline{x}_1 \quad \underline{x}_2 \quad \dots \quad \underline{x}_n] .$$

Let

$$\underline{b}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \underline{b}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \dots, \quad \underline{b}_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} ,$$

i.e.

$$\underline{1} = [\underline{b}_1 \quad \underline{b}_2 \quad \dots \quad \underline{b}_n] .$$

Solve the following:

$$\begin{aligned} \underline{A} \underline{x}_1 &= \underline{b}_1 \quad , \\ \underline{A} \underline{x}_2 &= \underline{b}_2 \quad , \\ &\vdots \\ \underline{A} \underline{x}_n &= \underline{b}_n \quad . \end{aligned}$$

Then, because

$$\underline{A} [\underline{x}_1 \quad \underline{x}_2 \quad \dots \quad \underline{x}_n] = [\underline{b}_1 \quad \underline{b}_2 \quad \dots \quad \underline{b}_n] \quad ,$$

we have

$$\underline{X} = \underline{A}^{-1} .$$

In a numerical example the Gauss elimination procedure may be applied simultaneously to the n problems defined above.

Evaluation of Determinants

Don't use the direct method! It requires  $n!$  multiplications.

After Gauss elimination take

$$\det \tilde{A} = a_{11}^1 a_{22}^2 \dots a_{nn}^n .$$

Pivoting

To avoid error accumulation which may destroy the significance of the numerical result we need the smallest possible multipliers:

$$m_{ik} = \frac{a_{ik}^k}{a_{kk}^k} , i = k+1, k+2, \dots, n .$$

(generates kth column)

Arrange or rearrange the equations so that

$$|a_{kk}^k| = \max_{k < i < n} |a_{ik}^k| .$$

In this case  $|m_{ik}| \leq 1$ . In practice, search down the column and interchange appropriate rows. This will, of course, handle the situation when  $a_{kk}^k = 0$ , and is essential for ill-conditioned systems. Note: interchanging rows does not affect the order of the variables. Complete pivoting (rarely worthwhile) involves searching the whole remaining submatrix for the largest element and interchanging rows and columns.

Ill-Conditioning

Small changes in the coefficients leads to large changes in solution, for example,

$$\begin{array}{ll} x_1 - x_2 = 1 & x_1 - x_2 = 1 \\ x_1 - 1.00001 x_2 = 0 & x_1 - .99999 x_2 = 0 \end{array}$$

yield, respectively,

$$\begin{matrix} x \\ \sim \end{matrix} = \begin{bmatrix} 100,001 \\ 100,000 \end{bmatrix} \qquad \begin{matrix} x \\ \sim \end{matrix} = \begin{bmatrix} -99,999 \\ -100,000 \end{bmatrix} .$$

Physically, this type of ill-conditioning can be thought of as being due to the intersection of nearly parallel lines.

Gauss elimination gives:

$$\begin{array}{rcl} \# & x_1 - x_2 = 1 & x_1 - x_2 = 1 \\ & - .00001x_2 = -1 & + .00001x_2 = -1 \end{array}$$

The Computational Effort

It can be shown that the number of multiplications and divisions is as follows.

Forward reduction:

$$\frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6}$$

Back substitution:

$$\frac{n(n-1)}{2}$$

Total:

$$\frac{n^3}{3} + n^2 - \frac{n}{3}$$

LU Factorization (Crout, Choleski, triangularization)

Assume

$$\begin{matrix} A & = & L & U & , \\ \sim & & \sim & \sim & \end{matrix}$$

where

L is lower triangular ,

~

U is upper triangular .

~

Consider:

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} .$$

Equate the coefficients as follows.

col. 1                     $l_{11} = a_{11}, \quad l_{21} = a_{21}, \quad l_{31} = a_{31},$

col. 2                     $l_{11} u_{12} = a_{12}, \quad l_{21} u_{12} + l_{22} = a_{22}, \quad l_{31} u_{12} + l_{32} = a_{32},$

col. 3                     $l_{11} u_{13} = a_{13},$

$$l_{21} u_{13} + l_{22} u_{23} = a_{23},$$

$$l_{31} u_{13} + l_{32} u_{23} + l_{33} = a_{33} .$$

The system is now given by

$$\underline{L} \underline{U} \underline{x} = \underline{b} .$$

Let

$$\underline{U} \underline{x} = \underline{y} .$$

Then solve

$$\underline{L} \underline{y} = \underline{b}$$

for  $\underline{y}$  after forward substitution followed by

$$\underline{U} \underline{x} = \underline{y}$$

for  $\underline{x}$  after backward substitution.

Note that factorization need be done only once for multiple  $\underline{b}$ .

Specifically,

$$l_{i1} = a_{i1} .$$

Solving for  $u_{12}$  from col. 2

$$u_{12} = a_{12}/l_{11} ,$$

$$l_{22} = a_{22} - l_{21} u_{12} ,$$

$$l_{32} = a_{32} - l_{31} u_{12} .$$

Solving for  $u_{13}$  from col. 3

$$u_{13} = a_{13}/l_{11} .$$

Then

$$u_{23} = (a_{23} - l_{21} u_{13})/l_{22} ,$$

$$l_{33} = a_{33} - l_{31} u_{13} - l_{32} u_{23} .$$

In general,

$$u_{ij} = (a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj})/l_{ii} , \quad i < j ,$$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} , \quad i \geq j .$$

After conversion to  $u_{ij}$  and  $l_{ij}$  simply replace  $a_{ij}$  appropriately since this is not needed again.

Step 1

$$\begin{bmatrix} l_{11} & u_{12} & u_{13} \\ l_{21} & a_{22} & a_{23} \\ l_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12}/l_{11} & a_{13}/l_{11} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Step 2

$$\begin{bmatrix} l_{22} & u_{23} \\ l_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{22} - l_{21} u_{12} & (a_{23} - l_{21} u_{13})/l_{22} \\ a_{32} - l_{31} u_{12} & a_{33} \end{bmatrix}$$



Step 3  $[l_{33}] = [a_{33} - l_{31} u_{13} - l_{32} u_{23}]$

Hence,

$$\begin{bmatrix} l_{11} & u_{12} & u_{13} \\ l_{21} & l_{22} & u_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} .$$

Choleski factorization:

If  $\underline{A}$  is symmetric,  $\underline{A}^T = \underline{A}$ . Make the diagonal elements of  $\underline{L}$  and  $\underline{U}$  equal and then  $\underline{U} = \underline{L}^T$ . Then we need only  $\underline{L}$ .

The Computational Effort

Calculation of  $u_{ij}, l_{ij}$ :

$$n^3/3 - n/3$$

Forward and back substitution:

$$n^2$$

Total:

$$n^3/3 + n^2 - n/3$$

Matrix Inversion and Superposition

Consider, for a resistive network,

$$\underline{G} \underline{v} = \underline{i}$$

and, hence,

$$\underline{G} [\underline{v}_1 \ \underline{v}_2 \ \dots \ \underline{v}_n] = [\underline{i}_1 \ \underline{i}_2 \ \dots \ \underline{i}_n] ,$$

where we place, on the right hand side, appropriate unit current excitations.

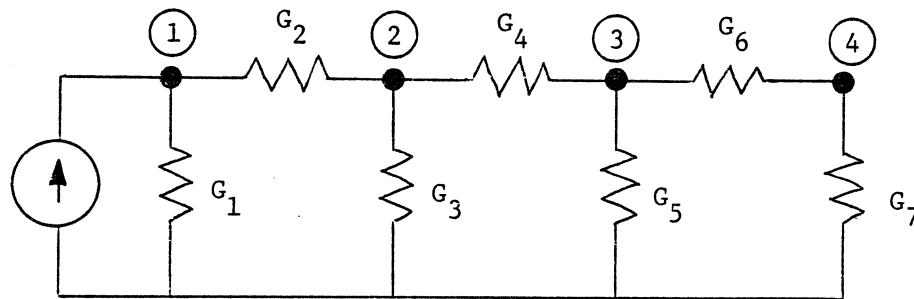
We can make up any vector  $\underline{i}$  as a linear combination of  $\underline{i}_i$ , i.e.,

$$\underline{i} = a_1 \underline{i}_1 + a_2 \underline{i}_2 + \dots + a_n \underline{i}_n .$$

Note that this demonstrates superposition, since

$$\begin{aligned} \underline{v} &= \underline{G}^{-1} \underline{i} = \underline{G}^{-1} (a_1 \underline{i}_1 + a_2 \underline{i}_2 + \dots + a_n \underline{i}_n) \\ &= a_1 \underline{v}_1 + a_2 \underline{v}_2 + \dots + a_n \underline{v}_n \end{aligned}$$

where the  $a_i$  are specific constants. Note:  $n^2$  multiplications are involved. This is the same as in the substitution steps in the LU factorization of  $\underline{G}$ .



For the circuit shown

$$\underline{G} = \begin{bmatrix} G_1+G_2 & -G_2 & 0 & 0 \\ -G_2 & G_2+G_3+G_4 & -G_4 & 0 \\ 0 & -G_4 & G_4+G_5+G_6 & -G_6 \\ 0 & 0 & -G_6 & G_6+G_7 \end{bmatrix}.$$

This is a tridiagonal matrix. The inverse is of the form

$$\underline{G}^{-1} = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}$$

whereas

$$\underline{L} \underline{U} = \begin{bmatrix} x & & & \\ x & x & & \\ & x & x & \\ & & x & x \end{bmatrix} \begin{bmatrix} x & & & \\ & x & & \\ & & x & x \\ & & & x \end{bmatrix}$$

Here, two substitution steps require only  $3n-2$  multiplications and divisions.

## Numerical Inversion of a Matrix

Start:

$$\begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Divide row 1 by 3:

$$\begin{array}{cccccc} 1 & -2/3 & 0 & 1/3 & 0 & 0 \\ -2 & 5 & -2 & 0 & 1 & 0 \\ 0 & -2 & 3 & 0 & 0 & 1 \end{array}$$

Multiply row 1 by -2 and subtract from row 2 to update row 2:

$$\begin{array}{cccccc} 1 & -2/3 & 0 & 1/3 & 0 & 0 \\ 0 & 11/3 & -2 & 2/3 & 1 & 0 \\ 0 & -2 & 3 & 0 & 0 & 1 \end{array}$$

Divide row 2 by 11/3:

$$\begin{array}{cccccc} 1 & -2/3 & 0 & 1/3 & 0 & 0 \\ 0 & 1 & -6/11 & 2/11 & 3/11 & 0 \\ 0 & -2 & 3 & 0 & 0 & 1 \end{array}$$

Multiply row 2 by -2 and subtract from row 3 to update row 3:

$$\begin{array}{cccccc} 1 & -2/3 & 0 & 1/3 & 0 & 0 \\ 0 & 1 & -6/11 & 2/11 & 3/11 & 0 \\ 0 & 0 & 21/11 & 4/11 & 6/11 & 1 \end{array}$$

Divide row 3 by 21/11:

$$\begin{array}{cccccc} 1 & -2/3 & 0 & 1/3 & 0 & 0 \\ 0 & 1 & -6/11 & 2/11 & 3/11 & 0 \\ 0 & 0 & 1 & 4/21 & 6/21 & 11/21 \end{array}$$

Multiply row 3 by  $-6/11$  and subtract from row 2 to update row 2:

$$\begin{array}{cccccc} 1 & -2/3 & 0 & 1/3 & 0 & 0 \\ 0 & 1 & 0 & 6/21 & 9/21 & 6/21 \\ 0 & 0 & 1 & 4/21 & 6/21 & 11/21 \end{array}$$

Multiply row 2 by  $-2/3$  and subtract from row 1 to update row 1:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 11/21 & 6/21 & 4/21 \\ 6/21 & 9/21 & 6/21 \\ 4/21 & 6/21 & 11/21 \end{bmatrix}$$

### Example of Matrix Inversion

Consider

$$\underline{G} \underline{G}^{-1} = \underline{1}$$

in the form

$$\begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \begin{bmatrix} a & d & e \\ d & b & f \\ e & f & c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We can write for column 1

$$3a - 2d = 1, \quad (1)$$

$$-2a + 5d - 2e = 0, \quad (2)$$

$$-2d + 3e = 0, \quad (3)$$

from which, using (1),

$$a = \frac{1 + 2d}{3}, \quad (4)$$

using (2) and substituting for this value of a

$$-\frac{2}{3}(1 + 2d) + 5d - 2e = 0,$$

or

$$-2 - 4d + 15d - 6e = 0,$$

or

$$11d - 6e = 2. \quad (5)$$

From (3)

$$-4d + 6e = 0,$$

which, added to (5), gives

$$7d = 2,$$

from which

$$d = 2/7 .$$

Substituting for this value of d in (4)

$$a = \frac{1 + 4/7}{3} = \frac{11}{21} ,$$

and in (3)

$$e = \frac{2 (2/7)}{3} = \frac{4}{21} .$$

We can now write for column 2, substituting for d,

$$3 \left( \frac{2}{7} \right) - 2b = 0 ,$$

from which

$$b = \frac{3}{7} ,$$

and, substituting for b,

$$-2 \left( \frac{3}{7} \right) + 3f = 0 ,$$

from which

$$f = \frac{2}{7} ,$$

and for column 3, substituting for f,

$$-2 \left( \frac{2}{7} \right) + 3c = 1 ,$$

from which

$$c = \frac{1 + 4/7}{3} = \frac{11}{21} .$$

In conclusion, the desired inverse is given by

$$\tilde{G}^{-1} = \frac{1}{21} \begin{bmatrix} 11 & 6 & 4 \\ 6 & 9 & 6 \\ 4 & 6 & 11 \end{bmatrix} .$$





**SECTION SIX**  
**EXAMPLES AND PROBLEMS**

© J.W. Bandler 1988

This material is taken from previous years' assignments and examples. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.



## EFFICIENT COMPUTATION OF CIRCUIT FUNCTIONS

### Algorithm for Rational Function Evaluation

Write an algorithm which evaluates efficiently the transfer function

$$F = \frac{a_0 + a_2s^2 + a_4s^4 + \dots + a_ns^n}{b_1s + b_3s^3 + b_5s^5 + \dots + b_ms^m},$$

where  $m$  is odd and  $n$  even.

### Solution

Horner's rule will be used for polynomial evaluation. Hence,  $F$  can be written as

$$F = \frac{a_0 + s^2 (a_2 + s^2 (a_4 + s^2 (a_6 + \dots)))}{s(b_1 + s^2 (b_3 + s^2 (b_5 + \dots)))},$$

### Algorithm

- (1) Insure that  $n$  is even and  $m$  is odd as follows  
IF(MOD( $n$ ,2).EQ.1.OR.MOD( $m$ ,2).EQ.0) stop
- (2)  $C \leftarrow s * s$   
Numerator evaluation
- (3)  $A \leftarrow a_n$
- (4)  $i \leftarrow n$
- (5) IF  $i = 0$  go to (9)
- (6)  $i \leftarrow i - 2$
- (7)  $A \leftarrow C * A + a_i$
- (8) GO to (5)
- Denominator evaluation
- (9)  $B \leftarrow b_m$
- (10)  $i \leftarrow m$
- (11) IF  $i = 1$  go to (15)

- (12)  $i \leftarrow i - 2$   
 (13)  $B \leftarrow C * B + b_i$   
 (14) Go to (11)  
 (15)  $F \leftarrow A / (s * B)$ .

Operation count for the computation of F only

\* Number of multiplications and divisions =  $\frac{n + m + 1}{2} + 2$ .

Number of additions =  $\frac{n + m - 1}{2}$ .

Algorithm for Evaluation of Transmission-Line Impedance

Write an algorithm to efficiently evaluate

$$F = Z_0 \frac{Z_L + j Z_0 \tan \theta}{Z_0 + j Z_L \tan \theta}$$

for any value of  $\theta$ .

Solution

It should be noticed that when  $\theta = \pi/2, 3\pi/2, \dots$ , F has the limiting value

$$F = Z_0^2 / Z_L.$$

Algorithm

- (1) Declare  $Z_0, Z_L, F$  and  $c$  to be complex
- (2) Ensure  $0 \leq \theta \leq \pi$   
 $\theta \leftarrow \theta - \pi * \text{INT}(\theta/\pi)$
- (3) IF  $|\theta - \pi/2| \geq \epsilon$ ,  
 where  $\epsilon$  is a small positive number ( $\epsilon = 10^{-14}$  say), go to (5)
- (4) Set  $F \leftarrow Z_0 * Z_0 / Z_L$  and stop
- (5)  $c \leftarrow j \tan \theta$
- (6) Set  $F \leftarrow Z_0 * (Z_L + c * Z_0) / (Z_0 + c * Z_L)$

Operation count in steps (5) and (6)

Number of multiplications and divisions = 4

Number of additions = 2

Number of trigonometric function evaluations = 1

### Efficient Evaluation of a Function

Show how to efficiently evaluate

$$F = a \sinh x + b \tanh x.$$

#### Solution

$$F = (e^x - e^{-x}) [a/2 + b/(e^x + e^{-x})].$$

#### Algorithm

- (1)  $c_1 \leftarrow e^x$
- (2)  $c_2 \leftarrow 1/c_1$
- (3)  $F \leftarrow (c_1 - c_2) * (a/2 + b/(c_1 + c_2))$ .

#### Operation count

Number of multiplications and divisions = 4.

Number of additions or subtractions = 3.

Number of exponential evaluations = 1.

### Efficient Evaluation of a Trigonometric Function

Show how to efficiently evaluate

$$F = a_1 \sin \theta + a_3 \sin 3\theta + a_5 \sin 5\theta.$$

#### Solution

$$\sin 3\theta = 3 \sin \theta - 4 \sin^3 \theta.$$

$$\sin 5\theta = \sin \theta - 20 \sin^3 \theta + 16 \sin^5 \theta.$$

Thus,

$$F = \sin \theta [a_1 + 2a_3 + (a_3 + 5a_5) - 4 \sin^2 \theta + ((a_3 + 5a_5) - 4a_5 \sin^2 \theta)].$$

#### Algorithm

- (1)  $b \leftarrow a_3 + 5 * a_5$
- (2)  $c \leftarrow \sin \theta$
- (3)  $d \leftarrow c + c$
- (4)  $e \leftarrow d * d$
- (5)  $F \leftarrow c * (a_1 + a_3 + a_3 + b - e * (b - a_5 * e))$ .

Operation count

Number of multiplications = 5.

Number of additions or subtractions = 7.

Number of trigonometric function evaluations = 1.







$$NM1 = NM-1 = 5$$

$$L = M + MM + MM = 7$$

READ R(I)

$$R(1) = 1$$

$$R(2) = 1/3$$

$$R(3) = 1$$

$$R(4) = 1/3$$

$$R(5) = 1$$

$$R(6) = 1/3$$

$$R(7) = 1$$

$$B = 1/R(2) = 3A$$

DO Loop 5

$$I = 1$$

$$ML = I+M = 1+3 = 4$$

$$J = I+I = 2$$

$$JJ = J+2 = 4$$

$$Y(1) = \frac{1}{R(2)} + \frac{1}{R(3)} + \frac{1}{R(4)} = 7 \text{ mho}$$

$$Y(4) = -\frac{1}{R(4)} = -3 \text{ mho}$$

$$I = 2$$

$$ML = 2+3 = 5$$

$$J = 4$$

$$JJ = 6$$

$$Y(2) = \frac{1}{R(4)} + \frac{1}{R(5)} + \frac{1}{R(6)} = 7 \text{ mho}$$

$$Y(5) = -\frac{1}{R(6)} = -3 \text{ mho}$$

$$Y(3) = \frac{1}{R(6)} + \frac{1}{R(7)} = 4 \text{ mho}$$

### LU Factorization

The matrix  $\underline{Y}$  is

$$\begin{bmatrix} 7 & & \\ -3 & 7 & \\ 0 & -3 & 4 \end{bmatrix} = \begin{bmatrix} l_{11} & u_{12} & u_{13} \\ l_{21} & l_{22} & u_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix}$$

$$Y(NM) = Y(MP)/Y(1)$$

$$Y(6) = Y(4)/Y(1) = -\frac{3}{7}$$

Y(6) is  $u_{12}$

### DO Loop 10

$$I = 2$$

$$ML = I+MM = 2+2 = 4$$

$$MU = I+NM2 = 2+4 = 6$$

$$Y(2) = Y(2) - Y(4) * Y(6) = 7 - (-3) * \left(\frac{-3}{7}\right) = \frac{40}{7}$$

$$MU = MU+1 = 7$$

$$ML = ML+1 = 5$$

$$Y(7) = \frac{Y(5)}{Y(2)} = \frac{-3}{40/7} = \frac{-21}{40}$$

Y(7) is  $u_{23}$

$$Y(3) = Y(3) - Y(5) * Y(7) = 4 - (-3) * \left(\frac{-21}{40}\right) = \frac{97}{40}$$

Forward Substitution

$$\begin{bmatrix} 7 & -\frac{3}{7} & 0 \\ -3 & \frac{40}{7} & -\frac{21}{40} \\ 0 & -3 & \frac{97}{40} \end{bmatrix}$$

$$\underline{\underline{L}} \underline{\underline{U}} \underline{\underline{V}} = \underline{\underline{I}}$$

Let

$$\underline{\underline{UV}} = \underline{\underline{z}}$$

Then

$$\underline{\underline{Lz}} = \underline{\underline{I}}$$

$$\begin{bmatrix} 7 & & \\ -3 & \frac{40}{7} & \\ 0 & -3 & \frac{97}{40} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$$

$$V(1) = \frac{B}{Y(1)} = \frac{3}{7}, \text{ i.e., } z_1 = \frac{3}{7} \quad *$$

DO Loop 20

$$I = 2$$

$$J = I-1 = 1$$

$$ML = I+MM = 2+2 = 4$$

$$V(2) = -\frac{Y(5) * V(1)}{Y(2)} = \frac{-3 * \frac{3}{7}}{40/7} = \frac{9}{40} \text{ i.e., } z_2 = \frac{9}{40} \quad *$$

$$I = 3$$

$$J = 2$$

$$ML = 5$$

$$V(3) = - \frac{Y(5) * V(2)}{Y(3)} = - \frac{-3 * \frac{9}{40}}{97/40} = \frac{27}{97}, \text{ i.e., } z_3 = \frac{27}{97} \quad *$$

Backward Substitution

$$\underline{U} \underline{V} = \underline{z}$$

$$\begin{bmatrix} 1 & -\frac{3}{7} & 0 \\ 0 & 1 & -\frac{21}{40} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} \frac{3}{7} \\ \frac{9}{40} \\ \frac{27}{97} \end{bmatrix}$$

$$C = \frac{V(3)}{R(7)} = \frac{27/97}{1} = \frac{27}{97}$$

$$J = MP = 4$$

$$JJ = N = 7$$

DO Loop 30

$$J = J-1 = 3$$

$$J1 = J-1 = 2$$

$$JJ = JJ-2 = 5$$

$$MU = J+NM2 = 3+4 = 7$$

$$V(2) = V(2) - Y(7) * V(3) = 0.371134 \text{ V}$$

$$C = \frac{V(J1)}{R(JJ)} = \frac{V(2)}{R(5)} = 0.371134 \text{ A}$$

$$I = 2$$

$$J = 2$$

$$J1 = 1$$

$$JJ = 3$$

$$MU = J + NM2 = 2 + 4 = 6$$

$$V(1) = V(1) - Y(6) * V(2)$$

$$= \frac{3}{7} - \left(\frac{-3}{7}\right) * 0.371134 = 0.5876289 \text{ V}$$

$$C = \frac{V(J1)}{V(JJ)} = \frac{0.5876289}{R(3)=1} = 0.5876289 \text{ A}$$

$$MB = 1$$

$$VB = 1 - V(1) = 1 - 0.5876288 = 0.4123712 \text{ V}$$

$$C = \frac{VB}{R(2)} = \frac{0.4123712}{1/3} = 1.2371136 \text{ A}$$

$$JJ = 2$$

DO Loop 40

$$I = 1$$

$$J = I + 1 = 2$$

$$JJ = 2 + 2 = 4$$

$$MB = MB + 1 = 2$$

$$UB = V(1) - V(2) = 0.21649 \text{ V}$$

$$C = \frac{(VB)}{R(JJ)} = 0.64949 \text{ V}$$

$$I = 2$$

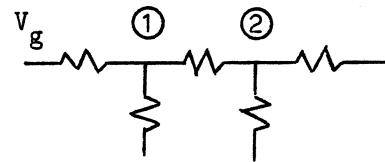
$$J = 3$$

$$JJ = 6$$

$$MB = 3$$

$$VB = V(2) - V(3) = .09278 \text{ V}$$

$$C = VB/R(JJ) = \frac{VB}{R(6)} = -\frac{0.09278}{1/3} = 0.27835 \text{ A}$$



```

PROGRAM EE3K4A1( INPUT, OUTPUT, TAPE3= INPUT, TAPE6=OUTPUT)          000001
DIMENSION Y(145), V(49), R(99)                                       000002
C                                                                           000003
PRINT(6,*) "TYPE TOTAL NUMBER OF RESISTORS"                          000004
READ(5,*) N                                                             000005
C                                                                           000006
M=(N-1)/2                                                               000007
MP=M+1                                                                   000008
MM=M-1                                                                    000009
NM=N-1                                                                    000010
NM2=NM-2                                                                  000011
NM1=NM-1                                                                  000012
L=M+MM+MM                                                                000013
C                                                                           000014
C SET UP THE Y MATRIX(NOTE Y MATRIX IS SYMMETRIC)                     000015
C MATRIX Y IS STORED SUCH THAT THE ENTRIES ON THE DIAGONAL           000016
C ARE NUMBERED FIRST ,THE LOWER DIAGONAL NEXT AND THEN THE          000017
C UPPER DIAGONAL                                                       000018
C                                                                           000019
PRINT(6,*) "TYPE RESISTOR VALUES FROM 1 TO N"                       000020
READ(5,*) (R(I), I=1, N)                                               000021
C                                                                           000022
B=1./R(2)                                                                000023
C                                                                           000024
DO 5 I=1, MM                                                            000025
ML=I+M                                                                    000026
J=I+I                                                                      000027
JJ=J+2                                                                    000028
Y(I)=1./R(J)+1./R(J+1)+1./R(JJ)                                         000029
Y(ML)=-1./R(JJ)                                                         000030
5 CONTINUE                                                                000031
Y(M)=1./R(JJ)+1./R(N)                                                  000032
C                                                                           000033
C LU FACTORIZATION                                                       000034
C                                                                           000035
Y(NM)=Y(MP)/Y(I)                                                         000036
DO 10 I=2, MM                                                            000037
ML=I+MM                                                                    000038
MU=I+NM2                                                                    000039
Y(I)=Y(I)-Y(ML)*Y(MU)                                                    000040
MU=MU+1                                                                    000041
ML=ML+1                                                                    000042
Y(MU)=Y(ML)/Y(I)                                                         000043
10 CONTINUE                                                                000044
Y(M)=Y(M)-Y(NM1)*Y(L)                                                  000045
C                                                                           000046
C FORWARD SUBSTITUTION                                                 000047
C                                                                           000048
V(1)=B/Y(1)                                                              000049
DO 20 I=2, M                                                             000050
J=I-1                                                                      000051
ML=I+MM                                                                    000052
V(I)=-Y(ML)*V(J)/Y(I)                                                    000053
20 CONTINUE                                                                000054
C                                                                           000055
C BACKWARD SUBSTITUTION                                                 000056
C                                                                           000057
C=V(M)/R(N)                                                              000058
PRINT(6,200)                                                            000059
PRINT(6,100) M, V(M), C                                                000060
J=MP                                                                      000061
JJ=N                                                                      000062
DO 30 I=1, MM                                                            000063
J=J-1                                                                      000064
J1=J-1                                                                    000065

```

	JJ=JJ-2	000066
	MU=J+NM2	000067
	V(J1)=V(J1)-Y(MU)*V(J)	000068
	C=V(J1)/R(JJ)	000069
	PRINT(6,100) J1,V(J1),C	000070
	30 CONTINUE	000071
C		000072
	MB=1	000073
	VB=1.-V(1)	000074
	C=VB/R(2)	000075
	PRINT(6,300)	000076
	PRINT(6,100) MB,VB,C	000077
	JJ=2	000078
	DO 40 I=1,MM	000079
	J=I+1	000080
	JJ=JJ+2	000081
	MB=MB+1	000082
	VB=V(I)-V(J)	000083
	C=VB/R(JJ)	000084
	PRINT(6,100) MB,VB,C	000085
	40 CONTINUE	000086
C		000087
C		000088
	100 FORMAT(5X,15,10X,F10.5,10X,F10.5,/)	000089
	200 FORMAT(1H1,2X,"NODAL VOLTAGES AND CURRENTS THROUGH SHUNT RESISTANC	000090
	+ES",/,3X,"NODE NUMBER",10X,"VOLTAGE",13X,"CURRENT",/)	000091
	300 FORMAT(2X,"BRANCH NUMBER",8X,"VOLTAGE",13X,"CURRENT",/)	000092
C		000093
	STOP	000094
	END	000095

TYPE TOTAL NUMBER OF RESISTORS  
 \*INPUT\* 7  
 TYPE RESISTOR VALUES FROM 1 TO N  
 \*INPUT\* 1.0,0.33333,1.0,0.33333,1.0,0.33333,1.0  
 1 NODAL VOLTAGES AND CURRENTS THROUGH SHUNT RESISTANCES

NODE NUMBER	VOLTAGE	CURRENT
3	.27835	.27835
2	.37114	.37114
1	.58763	.58763
BRANCH NUMBER	VOLTAGE	CURRENT
1	.41237	1.23712
2	.21649	.64949
3	.09278	.27835



**SECTION SEVEN**  
**ITERATIVE METHODS**

© J.W. Bandler 1984, 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



## ITERATIVE METHODS

### Jacobi's Method

Consider the  $n$  equations in  $n$  unknowns

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$\vdots$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

Rewrite this system in the form

$$x_1 = \frac{1}{a_{11}} (b_1 - a_{12}x_2 - \dots - a_{1n}x_n)$$

$$x_2 = \frac{1}{a_{22}} (b_2 - a_{21}x_1 - \dots - a_{2n}x_n)$$

$\vdots$

$$x_n = \frac{1}{a_{nn}} (b_n - a_{n1}x_1 - \dots - a_{n,n-1}x_{n-1})$$

Guess at  $x_1, x_2, \dots, x_n$  on the RHS find the LHS. Substitute in the RHS and repeat. Convergence is not guaranteed.

### Gauss-Seidel Method

This method is similar to the Jacobi method except that new values are substituted immediately they are obtained.

Let

$$\underline{A} \underline{x} = \underline{b} .$$

Partition  $\underline{A}$  so that

$$\underline{A} = \underline{L} + \underline{1} + \underline{U} ,$$

where we assume that each equation has already been divided by the corresponding diagonal element. The basic iteration is contrasted with the Jacobi iteration as follows.

Jacobi:

$$\tilde{x}^{j+1} = - (\tilde{L} + \tilde{U}) \tilde{x}^j + \tilde{b} .$$

Gauss-Seidel:

$$\tilde{x}^{j+1} = - \tilde{L} \tilde{x}^{j+1} - \tilde{U} \tilde{x}^j + \tilde{b} ,$$

or

$$(\tilde{1} + \tilde{L}) \tilde{x}^{j+1} = \tilde{b} - \tilde{U} \tilde{x}^j .$$

Relaxation method: any method in which a new approximation is obtained from the previous approximation and residuals, i.e., for Gauss-Seidel:

$$\tilde{x}^{j+1} = \tilde{x}^j + [\tilde{b} - \tilde{L} \tilde{x}^{j+1} - \tilde{U} \tilde{x}^j] ,$$

where the expression in brackets is the residual (known).

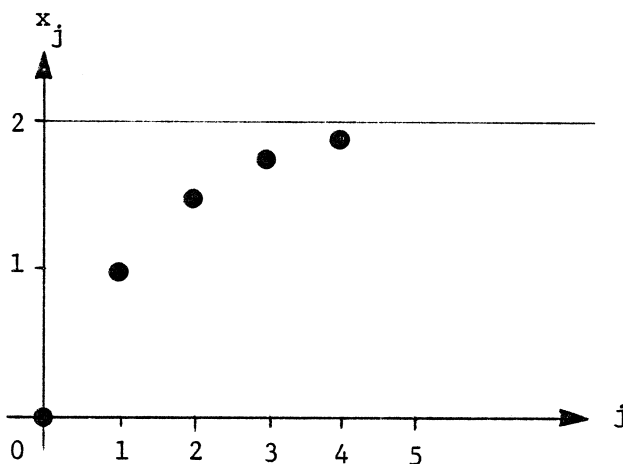
### Over-Relaxation

Generally used for the linear equations arising out of the solution of elliptic partial differential equations.

Consider, for example,

$$x^{j+1} = a x^j + b ,$$

where  $a = 1/2$  and  $b = 1$ . Let  $x^0 = 0$ .



$$\begin{aligned} x^0 &= 0 \\ x^1 &= 1 \\ x^2 &= 1.5 \\ x^3 &= 1.75 \\ x^4 &= 1.875 \\ &\vdots \\ x^\infty &= 2 \end{aligned}$$

Analytically,

$$x^1 = b$$

$$x^2 = ab + b$$

$$x^3 = a^2b + ab + b$$

$$x^4 = a^3b + a^2b + ab + b$$

⋮

$$x^n = b(1 + a + a^2 + a^3 + \dots + a^{n-1} \dots)$$

The term in brackets is a geometric series.

As  $n \rightarrow \infty$

$$x^\infty = \frac{b}{1-a} \quad \text{for } |a| < 1 .$$

For  $a = 1/2$ ,  $b = 1$ , we have  $x^\infty = 2$ . To speed up convergence consider

$$x^{j+1} = x^j + \omega(\bar{x}^{j+1} - x^j) , \quad \omega > 0 ,$$

where

$$\bar{x}^{j+1} = a x^j + b .$$

Then

$$\begin{aligned} x^{j+1} &= x^j + \omega(a - 1) x^j + \omega b \\ &= (\omega a + 1 - \omega) x^j + \omega b . \end{aligned}$$

Now when  $\omega = 1$  we recover the basic relaxation method. If  $\omega < 1$  we have under relaxation. If  $\omega > 1$  we have over relaxation. Usually we take  $1 < \omega < 2$ .

Let us consider  $\omega = 3/2$ . Then we obtain

$$\begin{aligned} x^0 &= 0 \\ x^1 &= ((1.5)(0.5) + 1 - 1.5)0 + 1.5 = 1.5 \\ x^2 &= 1.875 \\ x^3 &= 1.96875 \\ &\vdots \\ &\vdots \\ &\vdots \end{aligned}$$

Convergence, it should be noted, is twice as fast as before.

Let us formally consider over-relaxation, as follows. We take

$$\tilde{x}^{j+1} = \tilde{x}^j + \omega(\bar{x}^{j+1} - \tilde{x}^j) ,$$

where  $\bar{x}^{j+1}$  is calculated by Gauss-Seidel iteration. Therefore,

$$\begin{aligned} \tilde{x}^{j+1} &= \tilde{x}^j + \omega(\tilde{b} - \tilde{L} \tilde{x}^{j+1} - \tilde{x}^j - \tilde{U} \tilde{x}^j) \\ &= (-\omega \tilde{U} + (1-\omega) \tilde{I}) \tilde{x}^j - \omega \tilde{L} \tilde{x}^{j+1} + \omega \tilde{b} . \end{aligned}$$

It is hard to predict  $\omega$  in advance. Better usually to choose a high value in the range  $1 < \omega < 2$ .

### Convergence

Let us consider the iterative formula

$$\tilde{x}^{j+1} = \tilde{A} \tilde{x}^j + \tilde{B} ,$$

where  $\tilde{A}$  and  $\tilde{B}$  are square matrices. The solution is

$$\tilde{x}^\infty = \tilde{A} \tilde{x}^\infty + \tilde{B} \quad (2 = (0.5) 2 + 1 \text{ for the previous example})$$

Then we may express the error vector as

$$\tilde{e}^j \triangleq \tilde{x}^\infty - \tilde{x}^j$$

so that

$$\tilde{x}^\infty - \tilde{x}^{j+1} = \tilde{A}(\tilde{x}^\infty - \tilde{x}^j)$$

or

$$\tilde{e}^{j+1} = \tilde{A} \tilde{e}^j = \tilde{A}^{j+1} \tilde{e}^0$$

since

$$\begin{aligned}\tilde{e}^1 &= \tilde{A} \tilde{e}^0 \\ \tilde{e}^2 &= \tilde{A} \tilde{e}^1 = \tilde{A}^2 \tilde{e}^0 \\ \tilde{e}^3 &= \tilde{A} \tilde{e}^2 = \tilde{A}^3 \tilde{e}^0 \\ &\vdots\end{aligned}$$

where superscripts of  $\tilde{A}$  denote exponentiation.

Thus  $\tilde{A}$  must have the effect of reducing  $\tilde{e}^{j+1}$  for convergence in the Jacobi method.  $\tilde{A}$  takes the form

$$- \tilde{L} - \tilde{U} ,$$

whereas in the Gauss-Seidel method  $\tilde{A}$  takes the form

$$-(\tilde{1} + \tilde{L})^{-1} \tilde{U} .$$

Diagonal dominance is sufficient for convergence, i.e., if

$$\frac{\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|}{|a_{ii}|} < 1 \quad \text{for } j = 1, 2, \dots, n .$$

Positive definiteness also ensures convergence, i.e., for all  $\tilde{x}$

$$\tilde{x}^T \tilde{A} \tilde{x} \geq 0 ,$$

and

$$\tilde{x}^T \tilde{A} \tilde{x} = 0$$

implies that

$$\tilde{x} = \tilde{0} .$$

The diagonal elements are positive in a positive-definite matrix. The element of largest modulus lies on the diagonal.





**SECTION EIGHT**  
**EXAMPLES AND PROBLEMS**

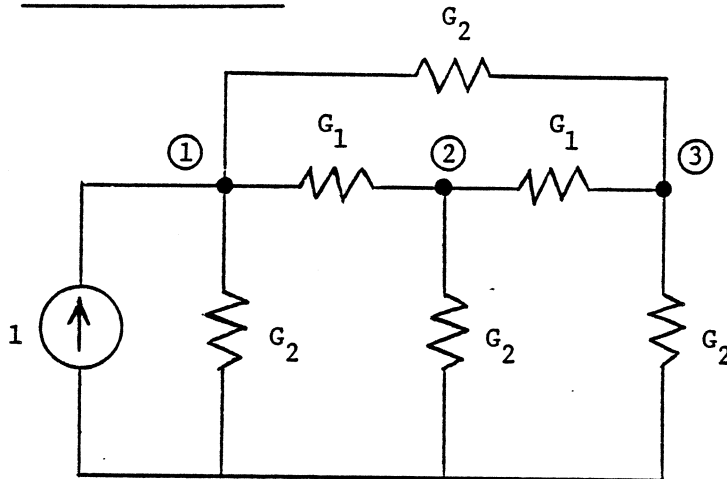
© J.W. Bandler 1988

This material is taken from previous years' assignments and examples. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.



## RELAXATION METHOD OF SOLVING EQUATIONS I

### Resistive Circuit



$$G_1 = 2$$

$$G_2 = 1$$

### Nodal Equations

$$\underline{G} \underline{V} = \underline{I}$$

$$\underline{G} = \begin{bmatrix} 4 & -2 & -1 \\ -2 & 5 & -2 \\ -1 & -2 & 4 \end{bmatrix} \quad \underline{I} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

The node equations are

$$4V_1 - 2V_2 - V_3 = 1$$

$$-2V_1 + 5V_2 - 2V_3 = 0$$

$$-V_1 - 2V_2 + 4V_3 = 0$$

### Gauss-Seidel Method

The superscript denotes the iteration number in the following steps.

$$\begin{bmatrix} v_1^0 \\ v_2^0 \\ v_3^0 \end{bmatrix} = \tilde{0}$$

$$v_1^1 = \frac{1 + 2v_2^0 + v_3^0}{4} = 0.25$$

$$v_2^1 = \frac{2v_1^1 + 2v_3^0}{5} = 0.1$$

$$v_3^1 = \frac{v_1^1 + 2v_2^1}{4} = 0.1125$$

$$v_1^2 = \frac{1 + 2v_2^1 + v_3^1}{4} = 0.328125$$

$$v_2^2 = \frac{2v_1^2 + 2v_3^1}{5} = 0.17625$$

$$v_3^2 = \frac{v_1^2 + 2v_2^2}{4} = 0.17015625$$

### Overrelaxation

$$\begin{aligned} \tilde{y}^n &= (1 - \omega)\tilde{y}^{n-1} + \omega \bar{y}^n \\ &= \tilde{y}^{n-1} - \omega \tilde{y}^{n-1} + \omega \bar{y}^n \\ &= \tilde{y}^{n-1} + \omega \delta, \end{aligned}$$

where

$$\delta = \bar{v}^n - v^{n-1}.$$

$\bar{v}^n$  is obtained by Gauss-Seidel. In the following,  $\omega = 1.5$ .

$$\begin{bmatrix} v_1^0 \\ v_2^0 \\ v_3^0 \end{bmatrix} = \underline{0} \quad \begin{array}{ll} \bar{v}_1^1 = 0.25 & v_1^1 = (1 - \omega)*0 + 1.5*0.25 = 0.375 \\ \bar{v}_2^1 = 3/20 & v_2^1 = (1 - \omega)*0 + 1.5*3/20 = 0.225 \\ \bar{v}_3^1 = 33/160 & v_3^1 = (1 - \omega)*0 + 1.5*33/160 = 0.309375 \end{array}$$

$$\bar{v}_1^2 = 0.43984375$$

$$v_1^2 = 0.472265625$$

$$\bar{v}_2^2 = 0.31265625$$

$$v_2^2 = 0.356484375$$

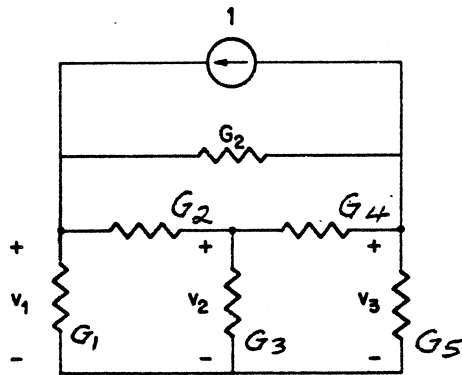
$$\bar{v}_3^2 = 0.2963085938$$

$$v_3^2 = 0.2897753907$$

## RELAXATION METHOD OF SOLVING EQUATIONS II

(Assignment 3, February 1985)

Write a general Fortran subroutine implementing the Gauss-Seidel method for solving a system of linear equations. Expressing the nodal equations as error functions calculate the Euclidean norm of the errors for each iteration. Use the Euclidean norm of the errors as a stopping criterion of the iterative algorithm. Test your subroutine on the resistive network shown starting with  $v_1 = 1.0$ ,  $v_2 = 0.5$ ,  $v_3 = 0$ . Assume that the solution has been found if the norm of the errors is less than  $10^{-4}$ .



$$G_1 = 1.0$$

$$G_2 = 2.0$$

$$G_3 = 1.5$$

$$G_4 = 2.0$$

$$G_5 = 2.0$$

$$G_6 = 1.0$$

A solution to this problem follows.

1. The Gauss-Seidel Method.

Consider the system of  $n$  simultaneous equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \vdots & \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (1)$$

The Gauss-Seidel iteration is of the form

$$\begin{aligned} x_1^{j+1} &= \frac{1}{a_{11}} (b_1 - a_{12}x_2^j - a_{13}x_3^j - \dots - a_{1n}x_n^j) \\ x_2^{j+1} &= \frac{1}{a_{22}} (b_2 - a_{21}x_1^{j+1} - a_{23}x_3^j - \dots - a_{2n}x_n^j) \\ \vdots & \\ x_n^{j+1} &= \frac{1}{a_{nn}} (b_n - a_{n1}x_1^{j+1} - a_{n2}x_2^{j+1} - \dots - a_{n,n-1}x_{n-1}^{j+1}) \end{aligned} \quad (2)$$

2. The Euclidean Norm of the Errors.

Nodal equations expressed as error functions take the form

$$\underline{e} = \underline{G} \underline{v} - \underline{i} \quad (3)$$

The Euclidean norm of the errors is defined as

$$\|\underline{e}\|_2 \triangleq \left\{ \sum_{i=1}^n |e_i|^2 \right\}^{1/2}$$

### 3. Test Problem

The conductance matrix for the test problem is

$$\underline{G} = \begin{bmatrix} 4 & -2 & -1 \\ -2 & 5.5 & -2 \\ -1 & -2 & 5 \end{bmatrix}$$

The excitation vector is

$$\underline{i} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}.$$

Starting values for the nodal voltages are taken as

$$\underline{v}^0 = \begin{bmatrix} 1 \\ 0.5 \\ 0 \end{bmatrix}.$$

The listing of the program together with numerical results of running the program are attached.



```
0001      PROGRAM TEST1
0002      DIMENSION G(3,3),AI(3),E(3),V(3)
0003      C
0004      G1=1.0
0005      G2=2.0
0006      G3=1.5
0007      G4=2.0
0008      G5=2.0
0009      G6=1.0
0010      N=3
0011      NCH=6
0012      G(1,1)=G1+G2+G6
0013      G(1,2)=-G2
0014      G(1,3)=-G6
0015      G(2,2)=G2+G3+G4
0016      G(2,3)=-G4
0017      G(3,3)=G4+G5+G6
0018      C
0019      DO 10 I=1,N
0020      DO 10 J=1,N
0021      10  G(J,I)=G(I,J)
0022      WRITE(6,100)((G(I,J),J=1,N),I=1,N)
0023      100  FORMAT(/' CONDUCTANCE MATRIX'/3(1X,F10.5)/)
0024      AI(1)=1.0
0025      AI(2)=0.0
0026      AI(3)=-1.0
0027      V(1)=1.0
0028      V(2)=0.5
0029      V(3)=0.0
0030      EPS=1.E-4
0031      CALL GAUSEI(G,AI,V,EPS,N,NCH,E)
0032      STOP
0033      END
```

```
0001      SUBROUTINE GAUSEI(G,AI,V,EPS,N,NCH,E)
0002      REAL G(N,N),AI(N),V(N),E(N)
0003      C
0004      IT=1
0005      C
0006      C      CALCULATE NEW VOLTAGES
0007      C
0008      50      DO 10 I=1,N
0009              SUM=0.0
0010              DO 20 J=1,N
0011                  IF(J.EQ.I)GO TO 20
0012                  SUM=SUM+G(I,J)*V(J)
0013      20      CONTINUE
0014              V(I)=(AI(I)-SUM)/G(I,I)
0015      10      CONTINUE
0016      C
0017      C      CALCULATE ERRORS
0018      C
0019              CALL MULTIP(G,V,E,N)
0020              EN=0.0
0021              DO 30 I=1,N
0022                  E(I)=E(I)-AI(I)
0023                  EN=EN+(ABS(E(I)))**2
0024      30      CONTINUE
0025              EN=SQRT(EN)
0026              WRITE(NCH,100)IT
0027      100      FORMAT(/' ITERATION NO. :',2X,I3/)
0028              WRITE(NCH,101)(I,V(I),I=1,N)
0029      101      FORMAT(/' V(',I3,')=' ,2X,F10.5)
0030              WRITE(NCH,102)EN
0031      102      FORMAT(/' EUCLIDEAN NORM OF THE ERRORS:',E13.5)
0032              IF(EN.LT.EPS)GO TO 40
0033              IT=IT+1
0034              GO TO 50
0035      40      RETURN
0036      END

0001      SUBROUTINE MULTIP(A,X,B,N)
0002      REAL A(N,N),X(N),B(N)
0003      C
0004              DO 10 I=1,N
0005                  B(I)=0.0
0006              DO 20 J=1,N
0007                  B(I)=B(I)+A(I,J)*X(J)
0008      20      CONTINUE
0009      10      CONTINUE
0010              RETURN
0011      END
```

CONDUCTANCE MATRIX

4.00000	-2.00000	-1.00000
-2.00000	5.50000	-2.00000
-1.00000	-2.00000	5.00000

ITERATION NO. : 1

V( 1)= 0.50000

V( 2)= 0.18182

V( 3)= -0.02727

EUCLIDEAN NORM OF THE ERRORS: 0.66587E+00

.  
.  
.

ITERATION NO. : 13

V( 1)= 0.23144

V( 2)= 0.03309

V( 3)= -0.14048

EUCLIDEAN NORM OF THE ERRORS: 0.91564E-04



**SECTION NINE**  
**NONLINEAR SYSTEM SIMULATION**

© J.W. Bandler 1984, 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



## SOME GRADIENT CONCEPTS

### Gradient

$$\lim_{\lambda \rightarrow 0^+} \frac{f(\underline{x} + \lambda \underline{s}) - f(\underline{x})}{\lambda} = \underline{\nabla} f^T \underline{s} .$$

### Consequence

Suppose

$$\underline{\nabla} f^T \underline{s} > 0 ,$$

then there exists  $\sigma > 0$  such that for all  $\lambda, \sigma \geq \lambda > 0$

$$f(\underline{x} + \lambda \underline{s}) > f(\underline{x}) .$$

### Mean Value Theorem

If  $f$  is differentiable in the open interval  $(a,b)$  and continuous at  $a$  and  $b$ , then there is a number  $c$  with  $a < c < b$  such that

$$\frac{f(b) - f(a)}{b-a} = f'(c) .$$

Therefore,

$$f(b) = f(a) + f'(c)(b-a)$$

This is often referred to as the Extended Mean Value Theorem (Taylor's Formula).

# NONLINEAR SYSTEM SIMULATION

## The Problem

Consider the function

$$f(\underline{y}(\underline{x}), \underline{x}),$$

where

$$\underline{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_k \end{bmatrix}, \quad \underline{y} \triangleq \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_n \end{bmatrix}$$

subject to

$$\underline{h}(\underline{x}, \underline{y}) = \underline{0},$$

where

$$\underline{h} \triangleq \begin{bmatrix} h_1 \\ h_2 \\ \cdot \\ \cdot \\ h_n \end{bmatrix}$$

given  $\underline{x}$ . We wish to calculate

$$\frac{\partial f}{\partial \underline{x}} \text{ s.t. } \underline{h}(\underline{x}, \underline{y}) = \underline{0},$$

where



$$\frac{\partial f}{\partial \underline{x}} \triangleq \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \cdot \\ \cdot \\ \frac{\partial f}{\partial x_k} \end{bmatrix} \cdot$$

Solution of Nonlinear Equations

Solution of the nonlinear equations can be carried by linearization (Newton's method) as follows. At the jth iteration,

$$\underline{h}^{j+1} = \underline{h}^j + \left( \frac{\partial \underline{h}^T}{\partial \underline{y}} \right)^T \Big|_j (\underline{y}^{j+1} - \underline{y}^j) + \dots$$

where

$$\underline{h}^j \triangleq \underline{h}(\underline{x}, \underline{y}^j)$$

and

$$\frac{\partial \underline{h}^T}{\partial \underline{y}} \triangleq \begin{bmatrix} \frac{\partial h_1}{\partial y_1} & \frac{\partial h_2}{\partial y_1} & \dots & \frac{\partial h_n}{\partial y_1} \\ \frac{\partial h_1}{\partial y_2} & \frac{\partial h_2}{\partial y_2} & \dots & \frac{\partial h_n}{\partial y_2} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial h_1}{\partial y_n} & \frac{\partial h_2}{\partial y_n} & \dots & \frac{\partial h_n}{\partial y_n} \end{bmatrix} =$$

Set  $\tilde{h}^{j+1} = \tilde{0}$  to obtain the linear system

$$\left( \frac{\partial \tilde{h}^T}{\partial \tilde{y}} \right) \Big|_j \tilde{y}^{j+1} = \left( \frac{\partial \tilde{h}^T}{\partial \tilde{y}} \right) \Big|_j \tilde{y}^j - \tilde{h}^j = \tilde{e}^j$$

to be solved for  $\tilde{y}^{j+1}$ . A byproduct of this process is the evaluation of the Jacobian

$$\tilde{J}^T \triangleq \frac{\partial \tilde{h}^T}{\partial \tilde{y}}$$

in addition to the solution, which will be denoted

$$\tilde{y} .$$

### The Adjoint System

Now

$$\left. \frac{\partial f}{\partial \tilde{x}} \right|_{\tilde{h}=\tilde{0}} = \frac{\partial \tilde{y}^T}{\partial \tilde{x}} \frac{\partial f}{\partial \tilde{y}} + \frac{\partial f}{\partial \tilde{x}} ,$$

where  $\partial \tilde{y}^T / \partial \tilde{x}$  is defined by analogy with  $\partial \tilde{h}^T / \partial \tilde{y}$ .

Furthermore,

$$\left. \frac{\partial \tilde{h}^T}{\partial \tilde{x}} \right|_{\tilde{h}=\tilde{0}} = \frac{\partial \tilde{y}^T}{\partial \tilde{x}} \frac{\partial \tilde{h}^T}{\partial \tilde{y}} + \frac{\partial \tilde{h}^T}{\partial \tilde{x}} = \tilde{0} .$$

Substituting for  $\partial \tilde{y}^T / \partial \tilde{x}$  we have

$$\left. \frac{\partial f}{\partial \tilde{x}} \right|_{\tilde{h}=\tilde{0}} = - \frac{\partial \tilde{h}^T}{\partial \tilde{x}} \left( \frac{\partial \tilde{h}^T}{\partial \tilde{y}} \right)^{-1} \frac{\partial f}{\partial \tilde{y}} + \frac{\partial f}{\partial \tilde{x}} = - \frac{\partial \tilde{h}^T}{\partial \tilde{x}} \hat{\tilde{y}} + \frac{\partial f}{\partial \tilde{x}} ,$$

where  $\hat{\underline{y}}$  is the solution to the adjoint linear system

$$\left(\frac{\partial \underline{h}^T}{\partial \underline{y}}\right) \hat{\underline{y}} = \frac{\partial f}{\partial \underline{y}}.$$

### Conclusion

The Newton method is summarized by

$$\underline{J}^j \underline{y}^{j+1} = \underline{J}^j \underline{y}^j - \underline{h}^j$$

and the adjoint system is described by

$$\underline{J}^{jT} \hat{\underline{y}} = \frac{\partial f}{\partial \underline{y}}.$$

where we assume that

$$\underline{y}^j \rightarrow \underline{y} \text{ and } \underline{J}^j \rightarrow \underline{J} \text{ as } \underline{h}^j \rightarrow 0.$$

The matrix of coefficients of the adjoint system is the transpose of the matrix used in the solution of the nonlinear system by linearization. Hence, LU factors can be reused for sensitivity evaluation.



**SECTION TEN**  
**EXAMPLES AND PROBLEMS**

© J.W. Bandler 1988

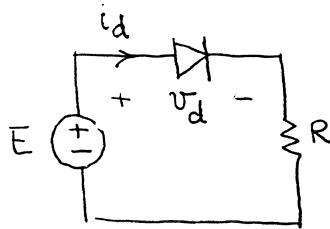
This material is taken from previous years' assignments and examples. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.



## SIMPLE RESISTOR DIODE CIRCUIT

For the circuit below, find the operating point using

- Techniques for solving nonlinear equations
- Optimization techniques



$$E = 10 \text{ V}$$

$$R = 1 \text{ k}\Omega$$

$$I_s = 10^{-12} \text{ mA}$$

$$\lambda = 38.7 \text{ V}^{-1}$$

$$i_d = I_s (e^{\lambda v_d} - 1)$$

Answer :  $v_d = 0.771404$

Newton - Raphson Method

$$f(x) = 0 \quad x^{j+1} = x^j - \frac{f^j}{\left(\frac{\partial f}{\partial x}\right)^j}$$

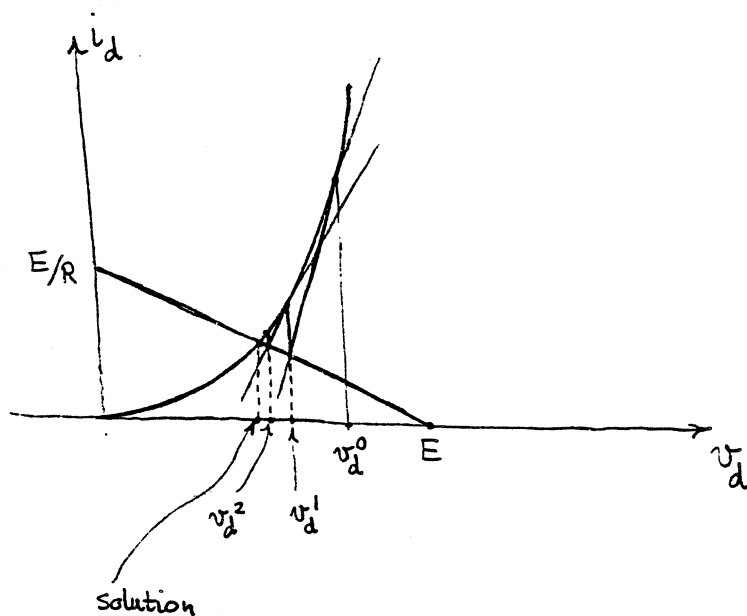
For our example:

$$f(v_d) = E - v_d - R i_d = 0$$

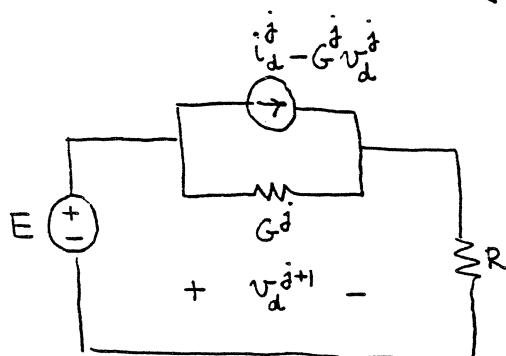
$$f(v_d) = E - v_d - R I_s (e^{\lambda v_d} - 1)$$

$$v_d^{j+1} = v_d^j + \frac{E - v_d^j - R I_s (e^{\lambda v_d^j} - 1)}{1 + R I_s \lambda e^{\lambda v_d^j}}$$

## Graphical Interpretation



## Circuit Interpretation (Companion Network)



$$i_d^j = I_s (e^{\lambda v_d^j} - 1)$$

$$G^j = \frac{\partial i_d^j}{\partial v_d^j} = \lambda I_s e^{\lambda v_d^j}$$

Using KVL

$$R(v_d^{j+1} G^j + i_d^j - G^j v_d^j) = E - v_d^{j+1}$$

Simplifies to

$$v_d^{j+1} = \frac{E - v_d^j - R i_d^j}{R G^j + 1} + v_d^j = v_d^j + \frac{E - v_d^j - R I_s (e^{\lambda v_d^j} - 1)}{1 + R I_s \lambda e^{\lambda v_d^j}}$$



Results for Newton-Raphson

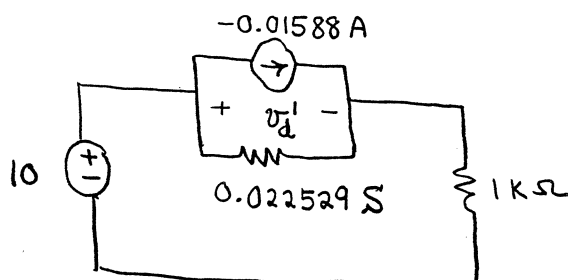
3

Using HP calculator

$$v_d^0 = 0.7$$

Iteration	$v_d$
1	1.070510
2	1.044670
3	1.018831
4	0.992993
5	0.967158
6	0.941331
7	0.915527
8	0.889783
9	0.864206
10	0.839073
11	0.815107
12	0.794018
13	0.778939
14	0.772401
15	0.771423
16	0.771404
17	0.771404

Companion Network :  $v_d^0 = 0.7$



$$v_d^1 = 1.070510$$

## Modified Newton method

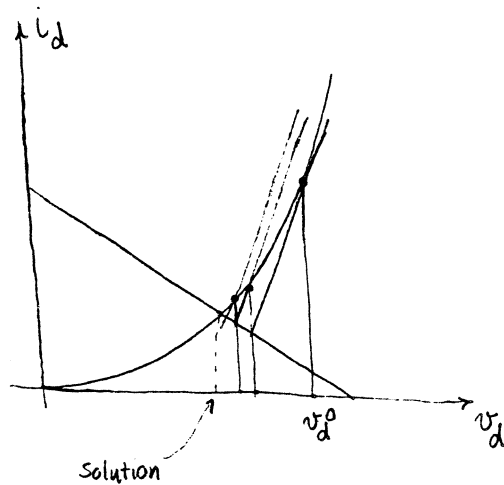
$$f(x) = 0 \quad x^{j+1} = x^j - \frac{f^j}{m} \quad m = \left(\frac{\partial f}{\partial x}\right)^0$$

For our example:

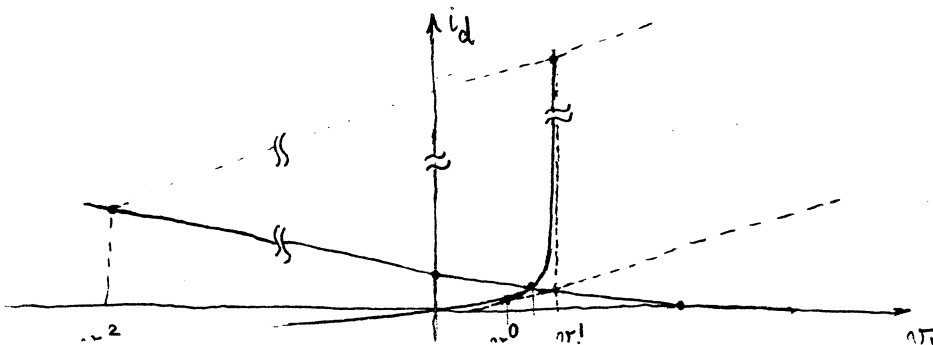
$$v_d^{j+1} = v_d^j + \frac{E - v_d^j - R I_s (e^{\lambda v_d^j} - 1)}{m}$$

$$m = 1 + R I_s \lambda e^{\lambda v_d^0}$$

Graphical Interpretation



It is easy to see that if  $v_d^0$  is smaller than the solution for our problem, we have a badly conditioned problem. This is what happens:



5  
Results for modified Newton method

$$v_d^0 = 0.7 \quad m = 23.53$$

Iteration	$v_d$
1	1.07051
2	-41748.44
⋮	⋮

Badly conditioned

---

$$v_d^0 = 0.85 \quad m = 7479.7$$

Iteration	$v_d$
1	0.825387
2	0.816647
⋮	⋮

More than 100 iterations

---

$$v_d^0 = 0.8 \quad m = 1081.1$$

Iteration	$v_d$
1	0.782694
2	0.778006
⋮	⋮
25	0.771404

## b) Optimization Methods

$$f(v_d) = E - v_d - R I_s (e^{\lambda v_d} - 1)$$

Minimize  $U = f^2(v_d)$   
w.r.t.  $v_d$

Newton's Method

$$U(\phi + \Delta\phi) = U(\phi) + \frac{\partial U}{\partial \phi} \Delta\phi + \frac{1}{2} \frac{\partial^2 U}{\partial \phi^2} (\Delta\phi)^2 + \dots$$

$$\left. \frac{\partial U}{\partial \phi} \right|_{\phi + \Delta\phi} = \left. \frac{\partial U}{\partial \phi} \right|_{\phi} + \left. \frac{\partial^2 U}{\partial \phi^2} \right|_{\phi} \Delta\phi + \dots$$

$$\Delta\phi = - \frac{\left. \frac{\partial U}{\partial \phi} \right|_{\phi}}{\left. \frac{\partial^2 U}{\partial \phi^2} \right|_{\phi}}$$

$$U = f^2 \quad \frac{\partial U}{\partial \phi} = 2f \frac{\partial f}{\partial \phi} \quad \frac{\partial^2 U}{\partial \phi^2} = 2 \left( \frac{\partial f}{\partial \phi} \right)^2 + 2f \frac{\partial^2 f}{\partial \phi^2}$$

$$\phi^{j+1} = \phi^j - \frac{\left( \left. \frac{\partial U}{\partial \phi} \right|_{\phi^j} \right)}{\left( \left. \frac{\partial^2 U}{\partial \phi^2} \right|_{\phi^j} \right)}$$

For our problem

$$v_d^{j+1} = v_d^j - \frac{f^j \left( \frac{\partial f}{\partial v_d} \right)^j}{f^j \left( \frac{\partial^2 f}{\partial v_d^2} \right)^j + \left( \frac{\partial f}{\partial v_d} \right)^j \left( \frac{\partial f}{\partial v_d} \right)^j}$$

$$v_d^{j+1} = v_d^j - \frac{[E - v_d^j - R I_s (e^{\lambda v_d^j} - 1)] [-1 - \lambda R I_s e^{\lambda v_d^j}]}{[E - v_d^j - R I_s (e^{\lambda v_d^j} - 1)] [-\lambda^2 R I_s e^{\lambda v_d^j}] + [-1 - \lambda R I_s e^{\lambda v_d^j}]^2}$$

## Results for optimization

$$v_d^0 = 0.85 \quad U = 33892.4$$

Iteration	$v_d$	U
1	0.837393	11985
2	0.824993	4128
3	0.812934	1358
4	0.801447	413
5	0.790913	109
6	0.781955	22
7	0.775462	2.5
8	0.772182	0.08
9	0.771437	0.0001
10	0.771404	$4 \times 10^{-9}$
11	0.771404	

## ANALYSIS AND SENSITIVITY EVALUATION FOR NONLINEAR SYSTEMS

### Problem

Given values for  $x_1$  and  $x_2$ , solve the nonlinear system

$$\begin{aligned}4x_1y_1 - 3y_2 &= 0 \\ -x_1y_1y_2 + 2x_2^2y_2 - 3 &= 0\end{aligned}$$

then find  $\partial f/\partial x_1$  and  $\partial f/\partial x_2$  for the function

$$f = y_1^2 + y_1x_1 .$$

### Notation

Let

$$\begin{aligned}h_1(x_1, x_2, y_1, y_2) &\triangleq 4x_1y_1 - 3y_2 \\ h_2(x_1, x_2, y_1, y_2) &\triangleq -x_1y_1y_2 + 2x_2^2y_2 - 3\end{aligned}$$

Taylor Series for  $h_1$  and  $h_2$  in Terms of  $y_1$  and  $y_2$

$$h_1^{j+1} = h_1^j + \partial h_1/\partial y_1 \Big|_j (y_1^{j+1} - y_1^j) + \partial h_1/\partial y_2 \Big|_j (y_2^{j+1} - y_2^j) + \dots$$

$$h_2^{j+1} = h_2^j + \partial h_2/\partial y_1 \Big|_j (y_1^{j+1} - y_1^j) + \partial h_2/\partial y_2 \Big|_j (y_2^{j+1} - y_2^j) + \dots$$

Given  $y_1^j, y_2^j$ , set  $h_1^{j+1} = h_2^{j+1} = 0$  and solve

$$\begin{bmatrix} \partial h_1 / \partial y_1 & \partial h_1 / \partial y_2 \\ \partial h_2 / \partial y_1 & \partial h_2 / \partial y_2 \end{bmatrix}_j \begin{bmatrix} y_1^{j+1} \\ y_2^{j+1} \end{bmatrix} = \begin{bmatrix} \partial h_1 / \partial y_1 & \partial h_1 / \partial y_2 \\ \partial h_2 / \partial y_1 & \partial h_2 / \partial y_2 \end{bmatrix}_j \begin{bmatrix} y_1^j \\ y_2^j \end{bmatrix} - \begin{bmatrix} h_1^j \\ h_2^j \end{bmatrix} \triangleq \begin{bmatrix} e_i^j \\ e_j^j \end{bmatrix}$$

for  $y_1^{j+1}$ ,  $y_2^{j+1}$ . Notice that a byproduct of this process is the evaluation of

$$\begin{bmatrix} \partial h_1 / \partial y_1 & \partial h_1 / \partial y_2 \\ \partial h_2 / \partial y_1 & \partial h_2 / \partial y_2 \end{bmatrix} = \begin{bmatrix} 4x_1 & -3 \\ -x_1 y_2^j & -x_1 y_1^j + 2x_2^2 \end{bmatrix}$$

Eventually, we expect to have for  $h_1=0$  and  $h_2=0$  the values of

$$\begin{bmatrix} 4x_1 & -3 \\ -x_1 y_2 & -x_1 y_1 + 2x_2^2 \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Partial Differentiation of  $f$ ,  $h_1$  and  $h_2$  w.r.t.  $x_1$  and  $x_2$

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}_{h=0} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial y_1} \\ \frac{\partial f}{\partial y_2} \end{bmatrix} + \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial h_1}{\partial x_1} \\ \frac{\partial h_1}{\partial x_2} \end{bmatrix}_{h=0} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial h_1}{\partial y_1} \\ \frac{\partial h_1}{\partial y_2} \end{bmatrix} + \begin{bmatrix} \frac{\partial h_1}{\partial x_1} \\ \frac{\partial h_1}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial h_2}{\partial x_1} \\ \frac{\partial h_2}{\partial x_2} \end{bmatrix}_{\tilde{h}=0} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial h_2}{\partial y_1} \\ \frac{\partial h_2}{\partial y_2} \end{bmatrix} + \begin{bmatrix} \frac{\partial h_2}{\partial x_1} \\ \frac{\partial h_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and, therefore,

$$\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = - \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_2}{\partial x_1} \\ \frac{\partial h_1}{\partial x_2} & \frac{\partial h_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial h_1}{\partial y_1} & \frac{\partial h_2}{\partial y_1} \\ \frac{\partial h_1}{\partial y_2} & \frac{\partial h_2}{\partial y_2} \end{bmatrix}^{-1}$$

so that

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}_{\tilde{h}=0} = - \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_2}{\partial x_1} \\ \frac{\partial h_1}{\partial x_2} & \frac{\partial h_2}{\partial x_2} \end{bmatrix} \hat{\tilde{y}} + \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

where  $\hat{\tilde{y}}$  is the solution to

$$\begin{bmatrix} \frac{\partial h_1}{\partial y_1} & \frac{\partial h_2}{\partial y_1} \\ \frac{\partial h_1}{\partial y_2} & \frac{\partial h_2}{\partial y_2} \end{bmatrix} \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial y_1} \\ \frac{\partial f}{\partial y_2} \end{bmatrix}$$

or

$$\begin{bmatrix} 4x_1 & -3 \\ -x_1y_2 & -x_1y_1 + 2x_2^2 \end{bmatrix}^T \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} 2y_1 + x_1 \\ 0 \end{bmatrix}$$



Finally,

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}_{h=0} = - \begin{bmatrix} 4y_1 & -y_1y_2 \\ 0 & 4x_2y_2 \end{bmatrix} \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} + \begin{bmatrix} y_1 \\ 0 \end{bmatrix}$$

NUMERICAL EXAMPLE OF ANALYSIS AND SENSITIVITY EVALUATION  
FOR NONLINEAR SYSTEMS

Solve the nonlinear system for  $\tilde{x} = [1 \quad 1.25]^T$

$$\begin{aligned} 4x_1y_1 - 3y_2 &= 0 \\ -x_1y_1y_2 + 2x_2^2y_2 - 3 &= 0 \end{aligned}$$

and then find  $\partial f/\partial x_1$  and  $\partial f/\partial x_2$  for the function

$$f = y_1^2 + y_1x_1.$$

Notation

Let

$$\begin{aligned} h_1(x_1, x_2, y_1, y_2) &\triangleq 4x_1y_1 - 3y_2 \\ h_2(x_1, x_2, y_1, y_2) &\triangleq -x_1y_1y_2 + 2x_2^2y_2 - 3 \end{aligned}$$

For the nonlinear system

$$\begin{aligned} h_1 &= 4y_1 - 3y_2 \\ h_2 &= -y_1y_2 + 3.125y_2 - 3 \end{aligned} \quad \text{or } \tilde{h} = \begin{bmatrix} 4y_1 - 3y_2 \\ -y_1y_2 + 3.125y_2 - 3 \end{bmatrix} \quad (1)$$

and the Jacobian  $\tilde{J}$  is given by

$$\tilde{J} = \begin{bmatrix} \frac{\partial h_1}{\partial y_1} & \frac{\partial h_1}{\partial y_2} \\ \frac{\partial h_2}{\partial y_1} & \frac{\partial h_2}{\partial y_2} \end{bmatrix} = \begin{bmatrix} 4 & -3 \\ -y_2 & -y_1 + 3.125 \end{bmatrix} \quad (2)$$

jth Iteration

$$\underline{J}^j \underline{y}^{j+1} = \underline{J}^j \underline{y}^j - \underline{h}^j \quad (3)$$

Starting Point

Taking  $\underline{y}^0 = [1 \quad 1]^T$  we have

$$\underline{J}^0 = \begin{bmatrix} 4 & -3 \\ -1 & 2.125 \end{bmatrix}$$

using equation (2),

$$\underline{h}^0 = \begin{bmatrix} 1 \\ -0.875 \end{bmatrix}$$

using equation (1).

First Iteration

From equation (3)

$$\underline{y}^1 = [1.090908 \quad 1.454544]^T$$

therefore, using equation (2)

$$\underline{J}^1 = \begin{bmatrix} 4 & -3 \\ 1.454544 & 2.034092 \end{bmatrix}$$

and using equation (1)

$$\underline{h}^1 = \begin{bmatrix} 0 \\ -0.0413237 \end{bmatrix}$$

Second Iteration

Substituting  $\underline{J}^1$ ,  $\underline{h}^1$  and  $\underline{y}^1$  in equation (3)

$$\underline{y}^2 = [1.120017 \quad 1.497336]^T$$

therefore, using equation (2)

$$\tilde{J}^2 = \begin{bmatrix} 4 & -3 \\ -1.497336 & 2.004983 \end{bmatrix}$$

and using equation (1)

$$\tilde{h}^2 = \begin{bmatrix} -0.0119388 \\ 0.00213245 \end{bmatrix}$$

### Third Iteration

Substituting  $\tilde{J}^2$ ,  $\tilde{h}^2$  and  $\tilde{y}^2$  in equation (3)

$$\tilde{y}^3 = \begin{bmatrix} 1.124989 \\ 1.499985 \end{bmatrix}$$

therefore, using equation (2)

$$\tilde{J}^3 = \begin{bmatrix} 4 & -3 \\ -1.499985 & 2.0000112 \end{bmatrix}$$

and using equation (1)

$$\tilde{h}^3 = \begin{bmatrix} -0.0000002 \\ 0.00001315 \end{bmatrix}$$

### Fourth Iteration

Substituting  $\tilde{J}^3$ ,  $\tilde{h}^3$  and  $\tilde{y}^3$  in equation (3)

$$\tilde{y}^4 = \begin{bmatrix} 1.125000 \\ 1.500000 \end{bmatrix}$$

therefore, using equation (2)

$$\tilde{J}^4 = \begin{bmatrix} 4 & -3 \\ -1.5 & 2 \end{bmatrix}$$

and using equation (1)

$$\tilde{h}^4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Check

Substituting  $\underline{J}^4$ ,  $\underline{h}^4$  and  $\underline{y}^4$  in equation (3)

$$\underline{y}^5 = \begin{bmatrix} 1.125 \\ 1.500 \end{bmatrix}$$

Sensitivity Evaluation

Now

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}_{\underline{h}=\underline{Q}} = - \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_2}{\partial x_1} \\ \frac{\partial h_1}{\partial x_2} & \frac{\partial h_2}{\partial x_2} \end{bmatrix} \hat{\underline{y}} + \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}, \quad (4)$$

where  $\hat{\underline{y}}$  is the solution to

$$\begin{bmatrix} \frac{\partial h_1}{\partial y_1} & \frac{\partial h_2}{\partial y_1} \\ \frac{\partial h_1}{\partial y_2} & \frac{\partial h_2}{\partial y_2} \end{bmatrix} \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial y_1} \\ \frac{\partial f}{\partial y_2} \end{bmatrix}, \quad (5)$$

or, substituting the solution  $\underline{y}^4$  into equation (5),

$$\begin{bmatrix} 4 & -1.5 \\ -3 & 2 \end{bmatrix} \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} 2y_1 + x_1 \\ 0 \end{bmatrix}.$$

By LU factorization, equation (5) gives the solution

$$\hat{\underline{y}} = \begin{bmatrix} 1.8571429 \\ 2.7857143 \end{bmatrix}$$

Hence, equation (4) yields

RESISTOR DIODE CIRCUIT SIMULATION  
INCLUDING SENSITIVITY ANALYSIS

(Assignment 3, February 1984)

See Question 136 of SECTION TWO for data, diagram and statement of objectives.

NOTES

1. Use only one LU factorization per iteration of Newton's method.
2. Use the results to find

$$\left[ \begin{array}{c} \frac{\partial}{\partial R_1} \\ \frac{\partial}{\partial R_2} \\ \frac{\partial}{\partial I_s} \\ \frac{\partial}{\partial \lambda} \\ \frac{\partial}{\partial E} \end{array} \right] \quad (i_2^2 R_2)$$

subject to satisfying the nonlinear equations.

3. Check derivatives by perturbation and comment on the results.
4. Use the test starting point, among others, suggested in Question 136.

A solution to this problem follows.

Resistor-diode Circuit

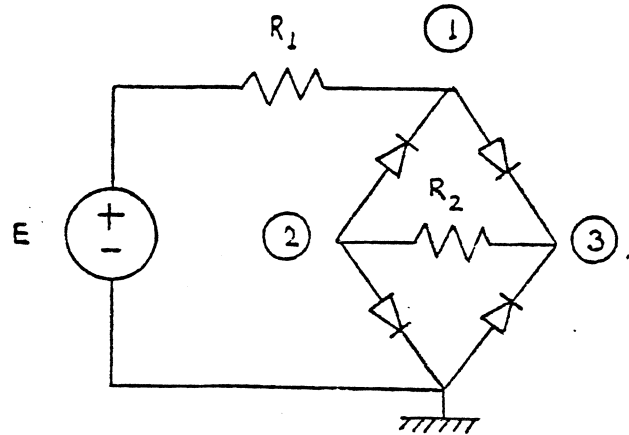
$$i_d = I_S (e^{\lambda v_d} - 1)$$

$$I_S = 10^{-12} \text{ mA}$$

$$\lambda = 1/V_T = 1/0.026 \text{ V}^{-1}$$

$$E = 10 \text{ V}$$

$$R_1 = R_2 = 1 \text{ k}\Omega$$



1. Solution by the Newton Method

The nodal equations of the given circuit are

$$f_1(v_1, v_2, v_3) = -\frac{E-v_1}{R_1} + I_S (e^{\lambda(v_1-v_3)} - 1) - I_S (e^{\lambda(v_2-v_1)} - 1) = 0$$

$$f_2(v_1, v_2, v_3) = \frac{v_2-v_3}{R_2} + I_S (e^{\lambda(v_2-v_1)} - 1) + I_S (e^{\lambda v_2} - 1) = 0 \quad (1)$$

$$f_3(v_1, v_2, v_3) = \frac{v_3-v_2}{R_2} - I_S (e^{\lambda(v_1-v_3)} - 1) - I_S (e^{-\lambda v_3} - 1) = 0$$

Define the vector

$$\underline{v} = [v_1 \quad v_2 \quad v_3]^T$$

and the vector

$$\underline{v}^j = [v_1^j \quad v_2^j \quad v_3^j]^T$$

as the value of  $\underline{v}$  obtained at the  $j$ th iteration.

Then, the Newton algorithm for solving the above system of nonlinear equations (equation (1)) is, for  $i = 1, 2, 3$ ,

$$f_i^j + \left(\frac{\partial f_i}{\partial v_1}\right)^j (v_1^{j+1} - v_1^j) + \left(\frac{\partial f_i}{\partial v_2}\right)^j (v_2^{j+1} - v_2^j) + \left(\frac{\partial f_i}{\partial v_3}\right)^j (v_3^{j+1} - v_3^j) = 0 \quad (2)$$

Equations (2) represent a system of 3 linear equations which can be written in the following matrix form

$$\underline{J}_v^j \underline{v}^{j+1} = \underline{J}_v^j \underline{v}^j - \underline{f}^j \quad (3)$$

where

$$\underline{J}_v^j = \begin{bmatrix} \left(\frac{\partial f_1}{\partial v_1}\right)^j & \left(\frac{\partial f_1}{\partial v_2}\right)^j & \left(\frac{\partial f_1}{\partial v_3}\right)^j \\ \left(\frac{\partial f_2}{\partial v_1}\right)^j & \left(\frac{\partial f_2}{\partial v_2}\right)^j & \left(\frac{\partial f_2}{\partial v_3}\right)^j \\ \left(\frac{\partial f_3}{\partial v_1}\right)^j & \left(\frac{\partial f_3}{\partial v_2}\right)^j & \left(\frac{\partial f_3}{\partial v_3}\right)^j \end{bmatrix} \quad \text{and} \quad \underline{f}^j = \begin{bmatrix} f_1^j \\ f_2^j \\ f_3^j \end{bmatrix}$$

The system of linear equations (equation (3)) can be solved by LU factorization.

The first-order derivatives of system (1) are

$$\frac{\partial f_1}{\partial v_1} = \frac{1}{R_1} + \lambda I_S [e^{\lambda(v_1 - v_3)} + e^{\lambda(v_2 - v_1)}]$$

$$\frac{\partial f_1}{\partial v_2} = -\lambda I_S e^{\lambda(v_2 - v_1)}$$

$$\frac{\partial f_1}{\partial v_3} = -\lambda I_S e^{\lambda(v_1 - v_3)}$$



$$\frac{\partial f_2}{\partial v_1} = -\lambda I_S e^{\lambda(v_2 - v_1)}$$

$$\frac{\partial f_2}{\partial v_2} = \frac{1}{R_2} + \lambda I_S e^{\lambda(v_2 - v_1)} + \lambda I_S e^{\lambda v_2}$$

$$\frac{\partial f_2}{\partial v_3} = -\frac{1}{R_2}$$

$$\frac{\partial f_3}{\partial v_1} = -\lambda I_S e^{\lambda(v_1 - v_3)}$$

$$\frac{\partial f_3}{\partial v_2} = -\frac{1}{R_2}$$

$$\frac{\partial f_3}{\partial v_3} = \frac{1}{R_2} + \lambda I_S [e^{\lambda(v_1 - v_3)} + e^{-\lambda v_3}]$$

Now, starting with  $\underline{v}^0$  as

$$\underline{v}^0 = [5.00000 \quad 1.00000 \quad 6.00000]^T$$

and using LU factorization the problem becomes very ill-conditioned.

A second initial guess  $\underline{v}^0$  as

$$\underline{v}^0 = [5.75000 \quad 0.75000 \quad 5.00000]^T$$

was used with LU factorization. Then the first 4 iterations obtained are

$$\underline{v}^1 = \begin{bmatrix} 5.75673 \\ 0.75673 \\ 5.00000 \end{bmatrix}, \underline{v}^2 = \begin{bmatrix} 5.75600 \\ 0.75600 \\ 5.00000 \end{bmatrix}, \underline{v}^3 = \begin{bmatrix} 5.75599 \\ 0.75599 \\ 5.00000 \end{bmatrix}, \underline{v}^4 = \begin{bmatrix} 5.75599 \\ 0.75599 \\ 5.00000 \end{bmatrix}$$

Comment

The Newton method for solving the set of nonlinear equations using LU factorization seems to be very sensitive to the initial guess  $\underline{v}^0$ .

## 2. Sensitivity Calculations

The sensitivity expressions are:

$$\begin{bmatrix} \frac{\partial F}{\partial R_1} \\ \frac{\partial F}{\partial R_2} \\ \frac{\partial F}{\partial I_s} \\ \frac{\partial F}{\partial \lambda} \\ \frac{\partial F}{\partial E} \end{bmatrix}_{\underline{x}=0} = - \begin{bmatrix} \frac{\partial f_1}{\partial R_1} & \frac{\partial f_2}{\partial R_1} & \frac{\partial f_3}{\partial R_1} \\ \frac{\partial f_1}{\partial R_2} & \frac{\partial f_2}{\partial R_2} & \frac{\partial f_3}{\partial R_2} \\ \frac{\partial f_1}{\partial I_s} & \frac{\partial f_2}{\partial I_s} & \frac{\partial f_3}{\partial I_s} \\ \frac{\partial f_1}{\partial \lambda} & \frac{\partial f_2}{\partial \lambda} & \frac{\partial f_3}{\partial \lambda} \\ \frac{\partial f_1}{\partial E} & \frac{\partial f_2}{\partial E} & \frac{\partial f_3}{\partial E} \end{bmatrix} \hat{\underline{v}} + \begin{bmatrix} \frac{\partial F}{\partial R_1} \\ \frac{\partial F}{\partial R_2} \\ \frac{\partial F}{\partial I_s} \\ \frac{\partial F}{\partial \lambda} \\ \frac{\partial F}{\partial E} \end{bmatrix}, \quad (4)$$

where  $F = i_2^2 R_2$  and  $\hat{\underline{v}}$  is the solution to the adjoint system:

$$\begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \frac{\partial f_2}{\partial v_1} & \frac{\partial f_3}{\partial v_1} \\ \frac{\partial f_1}{\partial v_2} & \frac{\partial f_2}{\partial v_2} & \frac{\partial f_3}{\partial v_2} \\ \frac{\partial f_1}{\partial v_3} & \frac{\partial f_2}{\partial v_3} & \frac{\partial f_3}{\partial v_3} \end{bmatrix} \begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \hat{v}_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial F}{\partial v_1} \\ \frac{\partial F}{\partial v_2} \\ \frac{\partial F}{\partial v_3} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{2}{R_2} (v_2 - v_3) \\ -\frac{2}{R_2} (v_2 - v_3) \end{bmatrix}. \quad (5)$$

The solution to the adjoint system obtained by reusing the LU factors of the Jacobian matrix at the last iteration of the Newton method is

$$\hat{\underline{v}} = \begin{bmatrix} 4.21816854 \\ -0.02584169 \\ 4.24401023 \end{bmatrix}. \quad (6)$$

The first-order derivatives in equation (4) are

$$\frac{\partial f_1}{\partial R_1} = \frac{E - v_1}{R_1^2}$$

$$\frac{\partial f_2}{\partial R_1} = 0$$

$$\frac{\partial f_3}{\partial R_1} = 0$$

$$\frac{\partial f_1}{\partial R_2} = 0$$

$$\frac{\partial f_2}{\partial R_2} = -\frac{v_2 - v_3}{R_2^2}$$

$$\frac{\partial f_3}{\partial R_2} = -\frac{v_3 - v_2}{R_2^2}$$

$$\frac{\partial f_1}{\partial I_s} = e^{\lambda(v_1 - v_3)} - e^{\lambda(v_2 - v_1)}$$

$$\frac{\partial f_2}{\partial I_s} = e^{\lambda(v_2 - v_1)} + e^{\lambda v_2} - 2$$

$$\frac{\partial f_3}{\partial I_s} = -e^{\lambda(v_1 - v_3)} - e^{-\lambda v_3} + 2$$

$$\frac{\partial f_1}{\partial \lambda} = I_s [e^{\lambda(v_1 - v_3)} (v_1 - v_3) - e^{\lambda(v_2 - v_1)} (v_2 - v_1)]$$

$$\frac{\partial f_2}{\partial \lambda} = I_s [e^{\lambda(v_2 - v_1)} (v_2 - v_1) + e^{\lambda v_2} v_2]$$

$$\frac{\partial f_3}{\partial \lambda} = I_s [e^{-\lambda v_3} v_3 - e^{\lambda(v_1 - v_3)} (v_1 - v_3)]$$

$$\frac{\partial f_1}{\partial E} = -\frac{1}{R_1}$$

$$\frac{\partial f_2}{\partial E} = 0$$

$$\frac{\partial f_3}{\partial E} = 0.$$

The numerical results obtained by running the program are:

$$\begin{bmatrix} \frac{\partial F}{\partial R_1} \\ \frac{\partial F}{\partial R_2} \\ \frac{\partial F}{\partial I_s} \\ \frac{\partial F}{\partial \lambda} \\ \frac{\partial F}{\partial E} \end{bmatrix} = \begin{bmatrix} - 0.179019504 \times 10^{-4} \\ 0.109672382 \times 10^{-6} \\ 0.219344764 \times 10^{12} \\ 0.165822398 \times 10^{-3} \\ 0.421816854 \times 10^{-2} \end{bmatrix} .$$

By running the program with small perturbations in  $R_1$  and  $R_2$ , the sensitivities of  $F$  w.r.t.  $R_1$  and  $R_2$  were checked and they agree well with the values obtained without perturbations.

```
PROGRAM TST(OUTPUT,TAPE6=OUTPUT)                                000001
C                                                                 000002
C THIS PROGRAM SOLVES THE SYSTEM OF NONLINEAR EQUATIONS        000003
C DESCRIBING THE RESISTOR-DIODE CIRCUIT USING LU FACTORIZATION  000004
C IN CONJUNCTION WITH THE NEWTON METHOD AND CALCULATES SENSITI- 000005
C VITIES OF F=I2**2*R2 W.R.T. R1,R2,IS,L AND E.                000006
C                                                                 000007
C REAL L                                                         000008
C REAL A(3,3),B(3),V(3)                                         000009
C                                                                 000010
C STARTING POINT FOR THE NEWTON METHOD                            000011
C                                                                 000012
C DATA V10,V20,V30/5.75,0.75,5.0/                              000013
C                                                                 000014
C INITIALIZE PARAMETERS                                         000015
C                                                                 000016
C LP=0                                                           000017
C IPAR=0                                                         000018
C                                                                 000019
C DEFINE THE NUMBER OF VARIABLES                                000020
C                                                                 000021
C N=3                                                            000022
C                                                                 000023
C ACCURACY OF THE SOLUTION                                     000024
C                                                                 000025
C EPS=1.0E-10                                                  000026
C                                                                 000027
C PERTURBATION IN R1 AND R2                                    000028
C                                                                 000029
C DR=0.2                                                        000030
C                                                                 000031
C VALUES OF RESISTORS                                         000032
C                                                                 000033
C R1=1000.0                                                     000034
C R2=R1                                                         000035
C 2 WRITE(6,130)R1,R2                                           000036
130 FORMAT(//1X," R1 = ",F6.1,5X," R2 = ",F6.1)                000037
C                                                                 000038
C OTHER GIVEN DATA FOR THE CIRCUIT                            000039
C                                                                 000040
C E=10.0                                                         000041
C L=1.0/0.026                                                   000042
C C=1.0E-15                                                     000043
C                                                                 000044
C MM IS THE ITERATION NUMBER                                    000045
C                                                                 000046
C MM=1                                                           000047
C                                                                 000048
C VARIABLES FOR FUNCTION AND DERIVATIVE CALCULATIONS          000049
C                                                                 000050
C V(1)=V10                                                       000051
C V(2)=V20                                                       000052
C V(3)=V30                                                       000053
C 10 V13=V(1)-V(3)                                              000054
C V21=V(2)-V(1)                                                000055
C V23=V(2)-V(3)                                                000056
C V32=-V23                                                       000057
C C1=EXP(L*V13)                                                 000058
C C2=EXP(L*V21)                                                 000059
C C3=EXP(L*V(2))                                                000060
C C4=EXP(-L*V(3))                                              000061
C C11=C1-1.0                                                    000062
C C21=C2-1.0                                                    000063
C                                                                 000064
C VECTOR OF NONLINEAR FUNCTIONS                                000065
```

```
C
V23R2=V23/R2
B(1)=- (E-V(1))/R1+C*(C11-C21)
B(2)=V23R2+C*(C21+C3-1.0)
B(3)=-V23R2-C*(C11+C4-1.0)
C
ELEMENTS OF THE JACOBIAN
C
R21=1.0/R2
ALC=L*C
A(1,1)=1.0/R1+ALC*(C1+C2)
A(1,2)=-ALC*C2
A(1,3)=-ALC*C1
A(2,1)=A(1,2)
A(2,2)=R21+ALC*(C2+C3)
A(2,3)=-R21
A(3,1)=A(1,3)
A(3,2)=A(2,3)
A(3,3)=R21+ALC*(C1+C4)
C
SOLUTION TO LINEAR SYSTEM IN THE NEWTON ITERATION
C
CALL SOLLU(N,A,B,IPAR)
V(1)=V(1)-B(1)
V(2)=V(2)-B(2)
V(3)=V(3)-B(3)
WRITE(6,50)(V(I),I=1,N)
50 FORMAT(//,5X,3(5X,F12.8))
IF(MM.GT.100)GO TO 90
C
CHECK THE CONVERGENCE CRITERION
C
Z1=ABS(V(1)-V10)
Z2=ABS(V(2)-V20)
Z3=ABS(V(3)-V30)
IF(Z1.LT.EPS.AND.Z2.LT.EPS.AND.Z3.LT.EPS)GO TO 70
C
HERE STARTS NEW ITERATION
C
MM=MM+1
V10=V(1)
V20=V(2)
V30=V(3)
V23=V(2)-V(3)
GO TO 10
70 WRITE(6,200)
200 FORMAT(//," SOLUTION TO NONLINEAR EQUATIONS : "//)
WRITE(6,300)MM,(V(I),I=1,N)
300 FORMAT(1X," NUMBER OF ITERATIONS = ",I3,/,5X,3(5X,F12.8))
IF(LP.GT.0)GO TO 13
C
SENSITIVITY CALCULATIONS
C
IPAR=1
C
RHS VECTOR OF THE ADJOINT SYSTEM
C
B(1)=0.0
B(2)=2.0*R21*V23
B(3)=-B(2)
C
SOLUTION TO THE ADJOINT SYSTEM
C
CALL SOLLU(N,A,B,IPAR)
WRITE(6,101)
```

```
101 FORMAT(/" SOLUTION TO THE ADJOINT SYSTEM : "/)
WRITE(6,102)(B(I),I=1,N)
102 FORMAT(//,1X,3(5X,E17.9),//)
C
C
C
      SENSITIVITY EXPRESSIONS
      R1R1=R1*R1
      R2R2=R2*R2
      DFDR1=((V(1)-E)/R1R1)*B(1)
      DFDR2=(V23/R2R2)*(B(2)-B(3))-(V23*V23)/R2R2
      V13=V(1)-V(3)
      V21=V(2)-V(1)
      ELV13=EXP(L*V13)
      ELV21=EXP(L*V21)
      ELV2=EXP(L*V(2))
      ELV3=EXP(-L*V(3))
      DFDIS=(ELV21-ELV13)*B(1)-(ELV21+ELV2-2.0)*B(2)+
1      (ELV13+ELV3-2.0)*B(3)
      D1=ELV13*V13
      D2=ELV21*V21
      DFDL=-C*(D1-D2)*B(1)-C*(D2+ELV2*V(2))*B(2)-C*(-D1+ELV3*V(3))*B(3)
      DFDE=1.0/R1*B(1)
      WRITE(6,201)
201 FORMAT(/" SENSITIVITIES : "/)
WRITE(6,85)DFDR1,DFDR2,DFDIS,DFDL,DFDE
85 FORMAT(/,6X," DFDR1 = ",E17.9,/
1      6X," DFDR2 = ",E17.9,/
2      6X," DFDIS = ",E17.9,/
3      6X," DFDL = ",E17.9,/
4      6X," DFDE = ",E17.9,/)
      IPAR=0
C
C
C
      SMALL PERTURBATIONS IN R1 AND R2
13 IF(LP.EQ.4)GO TO 89
   IF(LP.EQ.3)GO TO 88
   IF(LP.EQ.2)GO TO 87
   IF(LP.EQ.1)GO TO 86
      R1=1000.1
      LP=1
      GO TO 2
86 FP=V23*V23/R2
   R1=999.9
   LP=2
   GO TO 2
87 FN=V23*V23/R2
   FR1=(FP-FN)/DR
   WRITE(6,91)FR1
91 FORMAT(//,1X," CHECK OF SENSITIVITY DFDR1 : ",2X,E17.9)
   R1=1000.0
   R2=1000.1
   LP=3
   GO TO 2
88 FP=V23*V23/R2
   R2=999.9
   LP=4
   GO TO 2
89 FN=V23*V23/R2
   FR2=(FP-FN)/DR
   WRITE(6,92)FR2
92 FORMAT(//,1X," CHECK OF SENSITIVITY DFDR2 : ",2X,E17.9)
   GO TO 110
90 WRITE(6,100)
100 FORMAT(1X," NO CONVERGENCE")
110 STOP
```

**C**      **END**

**000196**  
**000197**





R1 = 1000.0      R2 = 1000.0

5.75673019	.75673019	5.00000000
5.75600015	.75600015	5.00000000
5.75598978	.75598978	5.00000000
5.75598977	.75598977	5.00000000
5.75598977	.75598977	5.00000000

SOLUTION TO NONLINEAR EQUATIONS :

NUMBER OF ITERATIONS = 5

5.75598977	.75598977	5.00000000
------------	-----------	------------

SOLUTION TO THE ADJOINT SYSTEM :

.421816854E+01	-.258416866E-01	.424401023E+01
----------------	-----------------	----------------

SENSITIVITIES :

DFDR1 =	-.179019504E-04
DFDR2 =	.109672382E-06
DFDIS =	.219344764E+12
DFDL =	.165822398E-03
DFDE =	.421816854E-02

R1 = 1000.1      R2 = 1000.0

5.75577629	.75598848	4.99978781
5.75577629	.75598848	4.99978781

SOLUTION TO NONLINEAR EQUATIONS :

NUMBER OF ITERATIONS = 2

5.75577629	.75598848	4.99978781
------------	-----------	------------

R1 = 999.9      R2 = 1000.0

5.75620328	.75599107	5.00021221
5.75620328	.75599107	5.00021221
5.75620328	.75599107	5.00021221

SOLUTION TO NONLINEAR EQUATIONS :

NUMBER OF ITERATIONS = 3

5.75620328	.75599107	5.00021221
------------	-----------	------------

CHECK OF SENSITIVITY DFDR1 : -.179019519E-04

R1 = 1000.0 R2 = 1000.1

5.75620067	.75598848	5.00021219
5.75620067	.75598848	5.00021219
5.75620067	.75598848	5.00021219

SOLUTION TO NONLINEAR EQUATIONS :

NUMBER OF ITERATIONS = 3

5.75620067	.75598848	5.00021219
------------	-----------	------------

R1 = 1000.0 R2 = 999.9

5.75577885	.75599107	4.99978779
5.75577885	.75599107	4.99978779
5.75577885	.75599107	4.99978779

SOLUTION TO NONLINEAR EQUATIONS :

NUMBER OF ITERATIONS = 3

5.75577885	.75599107	4.99978779
------------	-----------	------------

CHECK OF SENSITIVITY DFDR2 : .109672427E-06

## RESISTOR DIODE CIRCUIT SIMULATION, INCLUDING COMPANION NETWORK

### Resistor-diode Circuit

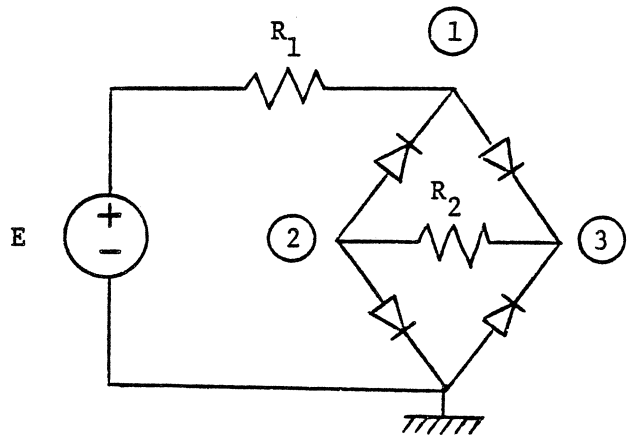
$$i_d = I_S (e^{\lambda v_d} - 1)$$

$$I_S = 10^{-12} \text{ mA}$$

$$\lambda = 1/V_T = 1/0.026 \text{ V}^{-1}$$

$$E = 10 \text{ V}$$

$$R_1 = R_2 = 1 \text{ k}\Omega$$



### 1. Solution by Newton Method

The nodal equations of the given circuit are

$$f_1(v_1, v_2, v_3) = -\frac{E-v_1}{R_1} + I_S (e^{\lambda(v_1-v_3)} - 1) - I_S (e^{\lambda(v_2-v_1)} - 1) = 0$$

$$f_2(v_1, v_2, v_3) = \frac{v_2-v_3}{R_2} + I_S (e^{\lambda(v_2-v_1)} - 1) + I_S (e^{\lambda v_2} - 1) = 0 \quad (1)$$

$$f_3(v_1, v_2, v_3) = \frac{v_3-v_2}{R_2} - I_S (e^{\lambda(v_1-v_3)} - 1) - I_S (e^{-\lambda v_3} - 1) = 0$$

Define the vector

$$\underline{v} = [v_1 \quad v_2 \quad v_3]^T$$

and the vector

$$\underline{v}^j = [v_1^j \quad v_2^j \quad v_3^j]^T$$

as the value of  $\underline{v}$  obtained at the  $j$ th iteration.

Then, the Newton algorithm for solving the above system of nonlinear equations (equation (1)) is, for  $i = 1, 2, 3$ ,

$$f_i^j + \left(\frac{\partial f_i}{\partial v_1}\right)^j (v_1^{j+1} - v_1^j) + \left(\frac{\partial f_i}{\partial v_2}\right)^j (v_2^{j+1} - v_2^j) + \left(\frac{\partial f_i}{\partial v_3}\right)^j (v_3^{j+1} - v_3^j) = 0 \quad (2)$$

Equations (2) represent a system of 3 linear equations which can be written in the following matrix form

$$\underset{\sim}{J}_v^{j,j+1} = \underset{\sim}{J}_v^{j,j} - \underset{\sim}{f}^j \quad (3)$$

where

$$\underset{\sim}{J}^j = \begin{bmatrix} \left(\frac{\partial f_1}{\partial v_1}\right)^j & \left(\frac{\partial f_1}{\partial v_2}\right)^j & \left(\frac{\partial f_1}{\partial v_3}\right)^j \\ \left(\frac{\partial f_2}{\partial v_1}\right)^j & \left(\frac{\partial f_2}{\partial v_2}\right)^j & \left(\frac{\partial f_2}{\partial v_3}\right)^j \\ \left(\frac{\partial f_3}{\partial v_1}\right)^j & \left(\frac{\partial f_3}{\partial v_2}\right)^j & \left(\frac{\partial f_3}{\partial v_3}\right)^j \end{bmatrix} \quad \text{and} \quad \underset{\sim}{f}^j = \begin{bmatrix} f_1^j \\ f_2^j \\ f_3^j \end{bmatrix}$$

The system of linear equations (equation (3)) can be solved either by the Gaussian elimination algorithm, LU factorization or matrix inversion.

The first-order derivatives of system (1) are

$$\frac{\partial f_1}{\partial v_1} = \frac{1}{R_1} + \lambda I_S [e^{\lambda(v_1 - v_3)} + e^{\lambda(v_2 - v_1)}]$$

$$\frac{\partial f_1}{\partial v_2} = -\lambda I_S e^{\lambda(v_2 - v_1)}$$

$$\frac{\partial f_1}{\partial v_3} = -\lambda I_S e^{\lambda(v_1 - v_3)}$$

$$\frac{\partial f_2}{\partial v_1} = -\lambda I_S e^{\lambda(v_2 - v_1)}$$

$$\frac{\partial f_2}{\partial v_2} = \frac{1}{R_2} + \lambda I_S e^{\lambda(v_2 - v_1)} + \lambda I_S e^{\lambda v_2}$$

$$\frac{\partial f_2}{\partial v_3} = -\frac{1}{R_2}$$

$$\frac{\partial f_3}{\partial v_1} = -\lambda I_S e^{\lambda(v_1 - v_3)}$$

$$\frac{\partial f_3}{\partial v_2} = -\frac{1}{R_2}$$

$$\frac{\partial f_3}{\partial v_3} = \frac{1}{R_2} + \lambda I_S [e^{\lambda(v_1 - v_3)} + e^{-\lambda v_3}]$$

Now, starting with  $\underline{v}^0$  as

$$\underline{v}^0 = [5.00000 \quad 1.00000 \quad 6.00000]^T$$

and using matrix inversion the problem becomes very ill-conditioned.

A second initial guess  $\underline{v}^0$  as

$$\underline{v}^0 = [5.75000 \quad 0.75000 \quad 5.00000]^T$$

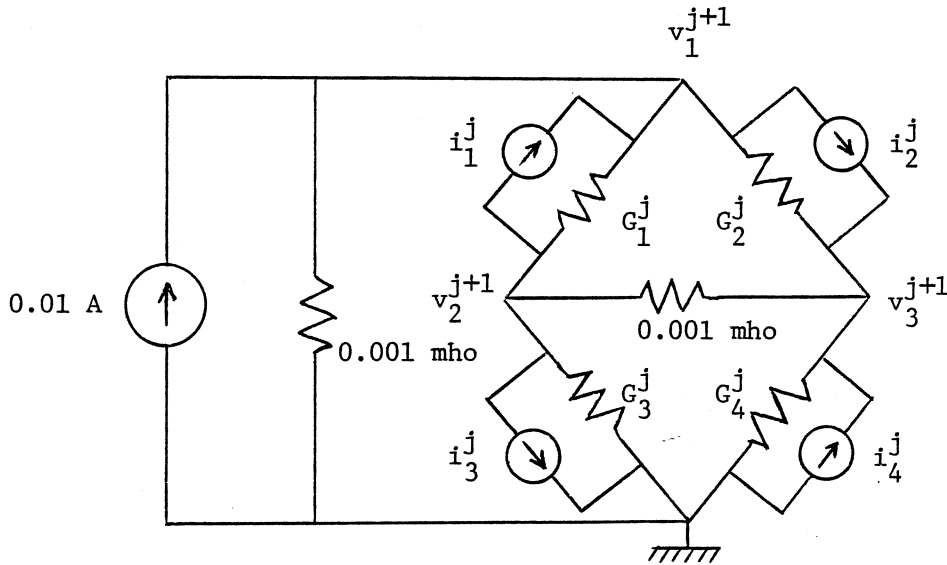
was used with matrix inversion. Then the first 4 iterations obtained are

$$\underline{v}^1 = \begin{bmatrix} 5.75673 \\ 0.75673 \\ 5.00000 \end{bmatrix}, \underline{v}^2 = \begin{bmatrix} 5.75600 \\ 0.75600 \\ 5.00000 \end{bmatrix}, \underline{v}^3 = \begin{bmatrix} 5.75599 \\ 0.75598 \\ 5.00000 \end{bmatrix}, \underline{v}^4 = \begin{bmatrix} 5.75598 \\ 0.75598 \\ 5.00000 \end{bmatrix}$$

#### Comment

The Newton method for solving the set of nonlinear equations using matrix inversion seems to be very sensitive to the initial guess  $\underline{v}^0$ .

2. Solution by Companion Network (Equivalent to Newton Method)



Companion Network at the  $j$ th Iteration

A subroutine for LU factorization was used to solve the following system of equations at each iteration

$$\begin{bmatrix} 0.001 + G_1^j + G_2^j & -G_1^j & -G_2^j \\ -G_1^j & 0.001 + G_1^j + G_3^j & -0.001 \\ -G_2^j & -0.001 & 0.001 + G_2^j + G_4^j \end{bmatrix} \begin{bmatrix} v_1^{j+1} \\ v_2^{j+1} \\ v_3^{j+1} \end{bmatrix} = \begin{bmatrix} 0.01 + (i_{d1}^j - G_1^j v_{d1}^j) - (i_{d2}^j - G_2^j v_{d2}^j) \\ -(i_{d1}^j - G_1^j v_{d1}^j) - (i_{d3}^j - G_3^j v_{d3}^j) \\ (i_{d2}^j - G_2^j v_{d2}^j) - (i_{d4}^j - G_4^j v_{d4}^j) \end{bmatrix}$$

where

$$v_{d1}^j = v_2^j - v_1^j, \quad v_{d2}^j = v_1^j - v_3^j, \quad v_{d3}^j = v_2^j, \quad v_{d4}^j = -v_3^j$$

and

$$\left. \begin{aligned} i_{dk}^j &= I_S (e^{\lambda v_{dk}^j} - 1) \\ G_k^j &= \lambda I_S e^{\lambda v_{dk}^j} \end{aligned} \right\} \quad k = 1, 2, 3, 4$$

Starting at

$$\tilde{v} = \begin{bmatrix} 6.0 \\ 1.0 \\ 5.0 \end{bmatrix}$$

the solution is

$$\tilde{v} = \begin{bmatrix} 5.75599 \\ 0.75599 \\ 5.00000 \end{bmatrix}$$

in 14 iterations and using a stopping criterion of  $10^{-6}$  for the change in the voltages.

### 3. Solution by Optimization

The objective function (or the error function)

$$U = \sum_{i=1}^3 f_i^2 = f_1^2 + f_2^2 + f_3^2 = \tilde{f}^T \tilde{f}$$

where  $f_1$ ,  $f_2$  and  $f_3$  are the network equations (equation (1)). We have to supply the gradients of the objective function with respect to the variables  $v_1$ ,  $v_2$  and  $v_3$  to the optimization routine, i.e.,

$$\frac{\partial U}{\partial v_1}, \quad \frac{\partial U}{\partial v_2}, \quad \frac{\partial U}{\partial v_3}$$

Thus,

$$\tilde{\nabla} U = \begin{bmatrix} \frac{\partial U}{\partial v_1} \\ \frac{\partial U}{\partial v_2} \\ \vdots \\ \vdots \end{bmatrix} = 2\tilde{J}^T \tilde{f}$$



$$\left[ \frac{\partial U}{\partial v_3} \right]$$

where  $J$  and  $f$  are as defined before.

Starting with

$$\tilde{v}^0 = \begin{bmatrix} 6.0 \\ 1.0 \\ 5.0 \end{bmatrix}$$

the objective function was  $7.6 \times 10^9$ .

After 61 iterations the values of  $v_1$ ,  $v_2$  and  $v_3$  which minimize  $U$  were reached and these are

$$\tilde{v} = \begin{bmatrix} 5.75599 \\ 0.75599 \\ 5.00000 \end{bmatrix}$$

and  $U$  is  $7.2 \times 10^{-20}$  in 0.229 seconds.

## NONLINEAR CIRCUIT SIMULATION AND SENSITIVITY ANALYSIS

*(Assignment 5, March 1986)*

Solve Question 140 of SECTION TWO generalized as follows. Let the 1 ohm resistor be R. Let the R = 1 and

$$i_a = Av_a^3$$

$$i_b = Bv_b^3 + Cv_b$$

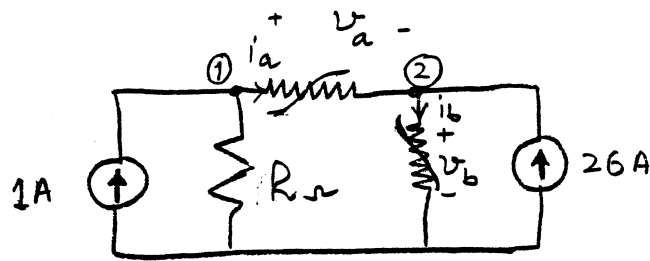
where A = 2, B = 1, C = 10. Let the starting point be  $v_1 = 1$ ,  $v_2 = 2$ .

Consider the function

$$F = v_a i_a$$

Calculate the partial derivatives, assuming you have a solution after the 4th iteration, of F w.r.t. R, A, B and C using an adjoint system. Check the results by small perturbations.

A solution to this problem follows.



$$v_a = v_1 - v_2$$

$$v_b = v_2$$

nodal equations

(K. current law at nodes 1 & 2)

$$i_a = A v_a^3$$

$$i_b = B v_b^3 + C v_b$$

$$R = 1$$

$$A = 2, B = 1, C = 10$$

$$f_1(A, B, C, v_1, v_2) = 1 - v_1 / R - A (v_1 - v_2)^3$$

$$f_2(A, B, C, v_1, v_2) = A (v_1 - v_2)^3 - B v_2^3 - C v_2 + 26$$

$$\tilde{J}^j = \begin{bmatrix} -\frac{1}{R} - 3A(v_1 - v_2)^2 & 3A(v_1 - v_2)^2 \\ 3A(v_1 - v_2)^2 & -3A(v_1 - v_2)^2 - 3Bv_2^2 - C \end{bmatrix}$$

at  $\{v_1^j, v_2^j\}$

Newton's iteration

$$\delta v^j = - (\tilde{J}^j)^{-1} f^j$$

$$v^{j+1} = v^j + \delta v^j$$

suggested starting point  
(in the class notes)

$$\tilde{v}^0 = \begin{bmatrix} 2.0 \\ 1.0 \end{bmatrix}$$

$$\epsilon = 10^{-10}$$

$j$	$v_1$	$v_2$
1	2.46391753	2.04123711
2	1.60352217	1.88474733
3	1.22796476	1.89585387
4	1.32462198	1.89116942
5	1.33756196	1.89053917
6	1.33777432	1.89052882
7	1.33777438	1.89052881
8	1.33777438	1.89052881

SOLUTION TO NONLINEAR EQUATIONS:

NUMBER OF ITERATIONS = 6

1.33777438      1.89052881

Alternative starting point  $\underline{v}^0 = \begin{bmatrix} 1.0 \\ 2.0 \end{bmatrix}$

$j$	$v_1$	$v_2$
1	1.20000000	1.90000000
2	1.31704384	1.89154854
3	1.33725109	1.89055434
4	1.33777404	1.89052883
5	1.33777438	1.89052381
6	1.33777438	1.89052881

SOLUTION TO NONLINEAR EQUATIONS:

NUMBER OF ITERATIONS = 6

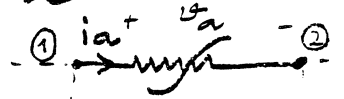
1.33777438	1.89052881
------------	------------

## Sensitivity Calculations

$$\frac{dF}{dy} = \begin{bmatrix} \frac{dF}{dR} \\ \frac{dF}{dA} \\ \frac{dF}{dB} \\ \frac{dF}{dC} \end{bmatrix} = - \begin{bmatrix} \frac{\partial f_1}{\partial R} & \frac{\partial f_2}{\partial R} \\ \frac{\partial f_1}{\partial A} & \frac{\partial f_2}{\partial A} \\ \frac{\partial f_1}{\partial B} & \frac{\partial f_2}{\partial B} \\ \frac{\partial f_1}{\partial C} & \frac{\partial f_2}{\partial C} \end{bmatrix} \hat{V} + \begin{bmatrix} \frac{\partial F}{\partial R} \\ \frac{\partial F}{\partial A} \\ \frac{\partial F}{\partial B} \\ \frac{\partial F}{\partial C} \end{bmatrix}$$

$\hat{V} = 0$

Let  $F = v_a i_a$   
 $= A v_a^4 = A (v_1 - v_2)^4$



$$i_a = A v_a^3$$

$$v_a = v_1 - v_2$$

Adjoint system equations

$$\begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \frac{\partial f_2}{\partial v_1} \\ \frac{\partial f_1}{\partial v_2} & \frac{\partial f_2}{\partial v_2} \end{bmatrix} \begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial F}{\partial v_1} \\ \frac{\partial F}{\partial v_2} \end{bmatrix} = \begin{bmatrix} 4A(v_1 - v_2)^3 \\ -4A(v_1 - v_2)^3 \end{bmatrix}$$

Adjoint solution

$$\hat{v}_1 = 0.462436813$$

$$\hat{v}_2 = -0.0223159044$$

The first-order partial derivatives :-

$$\frac{\partial f_1}{\partial R} = \frac{v_1}{R^2}, \quad \frac{\partial f_2}{\partial R} = 0$$

$$\frac{\partial f_1}{\partial A} = -(v_1 - v_2)^3, \quad \frac{\partial f_2}{\partial A} = (v_1 - v_2)^3$$

$$\frac{\partial f_1}{\partial B} = 0, \quad \frac{\partial f_2}{\partial B} = -v_2^3$$

$$\frac{\partial f_1}{\partial c} = 0, \quad \frac{\partial f_2}{\partial c} = -v_2$$

Also

$$\frac{\partial F}{\partial R} = 0, \quad \frac{\partial F}{\partial A} = (v_1 - v_2)^4, \quad \frac{\partial F}{\partial B} = \frac{\partial F}{\partial c} = 0$$

Answers

$$\frac{dF}{d\vec{u}} = \begin{bmatrix} -0.618636119 \\ 0.01484619 \\ 0.15078717 \\ -0.04218886 \end{bmatrix}$$

Verified by small perturbations





**SECTION ELEVEN**  
**LINEAR SYSTEM SIMULATION**

© J.W. Bandler 1984, 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



## LINEAR SYSTEM SIMULATION

### Problem

Consider the function

$$f(\underline{y}(\underline{x}), \underline{x})$$

subject to

$$\underline{A}(\underline{x}) \underline{y} = \underline{b}(\underline{x})$$

given  $\underline{x}$ . We wish to calculate

$$\frac{\partial f}{\partial \underline{x}} \text{ s.t. } \underline{A}(\underline{x}) \underline{y} = \underline{b}(\underline{x}) .$$

### Association with Nonlinear Equation Formulation

Here,

$$\underline{h} \triangleq \underline{A}(\underline{x}) \underline{y} - \underline{b}(\underline{x})$$

hence, for

$$\underline{A} \triangleq \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

we have

$$\frac{\partial h_1}{\partial \underline{y}} = [a_{11} \ a_{12} \ \cdots \ a_{1n}]^T ,$$

$$\frac{\partial h_2}{\partial \underline{y}} = [a_{21} \ a_{22} \ \cdots \ a_{2n}]^T ,$$

·  
·  
·

$$\frac{\partial h_n}{\partial \underline{y}} = [a_{n1} \ a_{n2} \ \dots \ a_{nn}]^T .$$

Note that all coefficients may be functions of  $\underline{x}$ .

Summarizing,

$$\begin{aligned} \frac{\partial \underline{h}^T}{\partial \underline{y}} &= \left[ \frac{\partial h_1}{\partial \underline{y}} \quad \frac{\partial h_2}{\partial \underline{y}} \quad \dots \quad \frac{\partial h_n}{\partial \underline{y}} \right] \\ &= \underline{A}^T . \end{aligned}$$

#### Adjoint System (Network)

We have the linear system in  $\hat{\underline{y}}$

$$\underline{A}^T \hat{\underline{y}} = \frac{\partial f}{\partial \underline{y}} ,$$

where  $\partial f / \partial \underline{y}$  must be readily available.

#### Solution by LU Factorization for Given $\underline{x}$

Let

$$\underline{L} \underline{U} \underline{y} = \underline{b} .$$

Solve by forward substitution the system

$$\underline{L} \underline{z} = \underline{b}$$

for  $\underline{z}$ , then solve by backward substitution

$$\underline{U} \underline{y} = \underline{z}$$

for  $\underline{y}$ .

To solve

$$(\underline{L} \underline{U})^T \hat{\underline{y}} = \frac{\partial f}{\partial \underline{y}}$$

for  $\hat{y}$  write

$$\tilde{U}^T \tilde{L}^T \hat{y} = \frac{\partial f}{\partial y}$$

and solve by forward substitution the system

$$\tilde{U}^T \hat{z} = \frac{\partial f}{\partial y}$$

for  $\hat{z}$ , then solve by backward substitution

$$\tilde{L}^T \hat{y} = \hat{z}$$

for  $\hat{y}$ .

#### Sensitivity Evaluation

$$\left. \frac{\partial f}{\partial x} \right|_{A y = b} = - \frac{\partial h^T}{\partial x} \hat{y} + \frac{\partial f}{\partial x},$$

where  $\partial h^T / \partial x$  depends upon the structure of the matrix  $A$  and the functional dependence of each coefficient on  $x$  and  $\partial f / \partial x$  must be readily known, and is often 0.

#### Sensitivity Evaluation for $f \equiv y_i$

Since

$$h = A y - b$$

it follows that the  $j$ th row of  $\partial h^T / \partial x$  is given by

$$\frac{\partial h^T}{\partial x_j} = y^T \frac{\partial A^T}{\partial x_j} - \frac{\partial b^T}{\partial x_j}.$$

Assembling these rows, we now have

$$\frac{\partial f}{\partial \underline{x}} \Big|_{\underline{A} \underline{x} = \underline{b}} = \frac{\partial y_i}{\partial \underline{x}} \Big|_{\underline{A} \underline{x} = \underline{b}} = - \begin{bmatrix} \underline{x}^T \frac{\partial \underline{A}^T}{\partial x_1} \hat{y}_i \\ \underline{x}^T \frac{\partial \underline{A}^T}{\partial x_2} \hat{y}_i \\ \cdot \\ \cdot \\ \underline{x}^T \frac{\partial \underline{A}^T}{\partial x_k} \hat{y}_i \end{bmatrix} + \begin{bmatrix} \frac{\partial \underline{b}^T}{\partial x_1} \hat{y}_i \\ \frac{\partial \underline{b}^T}{\partial x_2} \hat{y}_i \\ \cdot \\ \cdot \\ \frac{\partial \underline{b}^T}{\partial x_k} \hat{y}_i \end{bmatrix},$$

where we have let  $\hat{y}_i$  be defined as the solution of the adjoint system associated with the sensitivity of  $y_i$ , namely,

$$\underline{A}^T \hat{y} = \frac{\partial f}{\partial y} = \frac{\partial y_i}{\partial y} = \underline{u}_i,$$

where

$$\underline{u}_i \triangleq \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \leftarrow \text{ith row}$$

and where

$$\frac{\partial f}{\partial x} = \frac{\partial y_i}{\partial x} = 0 .$$

The principal assumption in using this approach to sensitivity evaluation is that few  $y_i$  are being considered and that

$$\frac{\partial A}{\partial x_j} , \quad j = 1, 2, \dots, k$$

are easily formulated and evaluated.

### Network Sensitivity Evaluation

Consider

$$\tilde{Y} \tilde{V} = \tilde{I} ,$$

where

$$\tilde{Y} \triangleq \tilde{A} \tilde{Y}_b \tilde{A}^T$$

is the node admittance matrix,  $\tilde{Y}_b$  is the branch admittance matrix and  $\tilde{A}$  is the reduced incidence matrix of the network. Let

$$\tilde{x} \equiv \tilde{Y}_b \tilde{v} ,$$

where

$$\tilde{v} \triangleq \begin{bmatrix} 1 \\ 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} .$$

Then

$$\begin{aligned} \tilde{A} &\equiv \tilde{Y} \\ \tilde{Y} &\equiv \tilde{V} \\ \tilde{b} &\equiv \tilde{I} \\ \tilde{A}^T &\equiv \tilde{Y}^T \end{aligned}$$

and the adjoint network is given by

$$\tilde{Y}^T \hat{V} = \hat{I},$$

where

$$\hat{I} \equiv \frac{\partial f}{\partial V}.$$

Network Sensitivity Evaluation for  $f \equiv V_i$ ,  $I = \text{constant}$

Here,

$$\left. \frac{\partial V_i}{\partial x_j} \right|_{\tilde{Y} \tilde{V} = \tilde{I}} = - \begin{bmatrix} \tilde{V}^T \frac{\partial \tilde{Y}^T}{\partial x_1} \hat{V}_i \\ \tilde{V}^T \frac{\partial \tilde{Y}^T}{\partial x_2} \hat{V}_i \\ \cdot \\ \cdot \\ \cdot \\ \tilde{V}^T \frac{\partial \tilde{Y}^T}{\partial x_k} \hat{V}_i \end{bmatrix}.$$

where

$$\frac{\partial \tilde{Y}^T}{\partial x_j} = \tilde{\Lambda} \frac{\partial \tilde{Y}_b^T}{\partial x_j} \tilde{\Lambda}^T.$$

But

$$\tilde{V}^T \tilde{\Lambda} \frac{\partial \tilde{Y}_b^T}{\partial x_j} \tilde{\Lambda}^T \hat{V}_i = \tilde{V}_b^T \frac{\partial \tilde{Y}_b^T}{\partial x_j} \hat{V}_{bi},$$

where  $\tilde{V}_b$  and  $\hat{V}_{bi}$  are vectors of branch voltages in the original and adjoint networks. Here, we have used Kirchhoff's voltage law

$$\begin{aligned} \tilde{V}_b &= \tilde{\Lambda}^T \tilde{V}, \\ \hat{V}_{bi} &= \tilde{\Lambda}^T \hat{V}_i, \end{aligned}$$

and the fact that, for the adjoint network



$$\begin{aligned}\tilde{Y}^T &= (\tilde{A} \tilde{Y}_b \tilde{A}^T)^T \\ &= \tilde{A}^T \tilde{Y}_b^T \tilde{A}.\end{aligned}$$

In general,  $\tilde{Y}_b$  is very sparse and  $\partial \tilde{Y}_b / \partial x_j$  has extremely few nonzero elements: only those associated with a network element if  $x_j$  is the element itself.



**SECTION TWELVE**  
**EXAMPLES AND PROBLEMS**

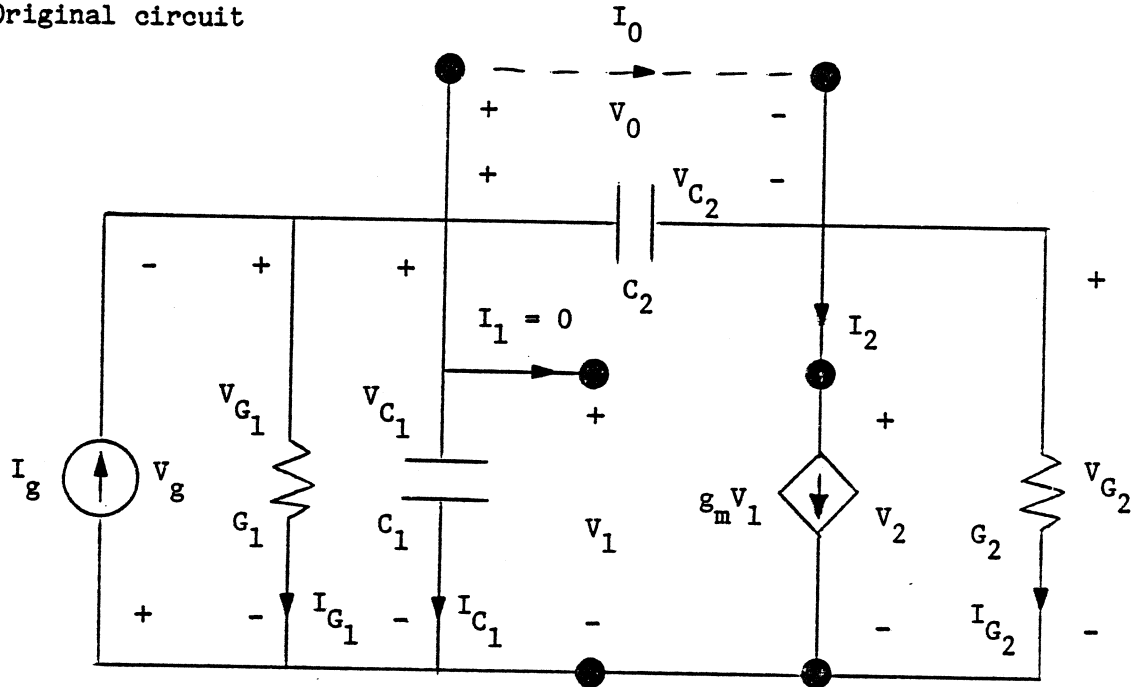
© J.W. Bandler 1988

This material is taken from previous years' assignments and examples. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.



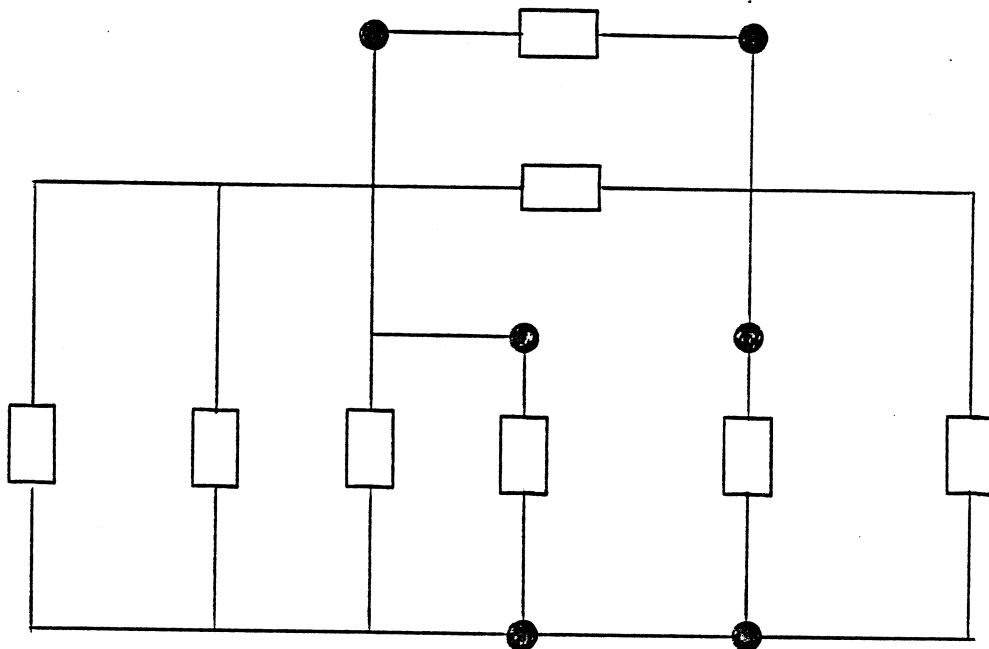
A SOLUTION TO QUESTION 110 OF SECTION TWO I

Original circuit



Adjoint Circuit

(notation to correspond to original circuit)



Tellegen's Theorem

$$\begin{aligned} & V_g \hat{I}_g + V_{G_1} \hat{I}_{G_1} + V_{C_1} \hat{I}_{C_1} + V_{C_2} \hat{I}_{C_2} + V_1 \hat{I}_1 + V_2 \hat{I}_2 \\ & + V_{G_2} \hat{I}_{G_2} + V_0 \hat{I}_0 - I_g \hat{V}_g - I_{G_1} \hat{V}_{G_1} - I_{C_1} \hat{V}_{C_1} \\ & - I_{C_2} \hat{V}_{C_2} - I_1 \hat{V}_1 - I_2 \hat{V}_2 - I_{G_2} \hat{V}_{G_2} - I_0 \hat{V}_0 = 0 \end{aligned}$$

Differentiate w.r.t.  $\phi$

$$\begin{aligned} & \frac{\partial V_g}{\partial \phi} \hat{I}_g + \frac{\partial V_{G_1}}{\partial \phi} \hat{I}_{G_1} + \frac{\partial V_{C_1}}{\partial \phi} \hat{I}_{C_1} + \frac{\partial V_{C_2}}{\partial \phi} \hat{I}_{C_2} + \frac{\partial V_1}{\partial \phi} \hat{I}_1 \\ & + \frac{\partial V_2}{\partial \phi} \hat{I}_2 + \frac{\partial V_{G_2}}{\partial \phi} \hat{I}_{G_2} + \frac{\partial V_0}{\partial \phi} \hat{I}_0 - \frac{\partial I_g}{\partial \phi} \hat{V}_g - \frac{\partial I_{G_1}}{\partial \phi} \hat{V}_{G_1} \\ & - \frac{\partial I_{C_1}}{\partial \phi} \hat{V}_{C_1} - \frac{\partial I_{C_2}}{\partial \phi} \hat{V}_{C_2} - \frac{\partial I_1}{\partial \phi} \hat{V}_1 - \frac{\partial I_2}{\partial \phi} \hat{V}_2 - \frac{\partial I_{G_2}}{\partial \phi} \hat{V}_{G_2} \\ & - \frac{\partial I_0}{\partial \phi} \hat{V}_0 = 0 \end{aligned}$$

Branch relations (original circuit)

$$I_g = \text{constant, hence } \frac{\partial I_g}{\partial \phi} = 0$$

$$I_{G_1} = G_1 V_{G_1}, \text{ hence } \frac{\partial I_{G_1}}{\partial \phi} = \frac{\partial G_1}{\partial \phi} V_{G_1} + G_1 \frac{\partial V_{G_1}}{\partial \phi}$$

$$I_{C_1} = j\omega C_1 V_{C_1}, \text{ hence } \frac{\partial I_{C_1}}{\partial \phi} = \frac{\partial(j\omega C_1)}{\partial \phi} V_{C_1} + j\omega C_1 \frac{\partial V_{C_1}}{\partial \phi}$$

$$I_{C_2} = j\omega C_2 V_{C_2}, \text{ hence } \frac{\partial I_{C_2}}{\partial \phi} = \frac{\partial(j\omega C_2)}{\partial \phi} V_{C_2} + j\omega C_2 \frac{\partial V_{C_2}}{\partial \phi}$$

$$I_1 = 0, \text{ hence } \frac{\partial I_1}{\partial \phi} = 0$$

$$I_2 = g_m V_1, \text{ hence } \frac{\partial I_2}{\partial \phi} = \frac{\partial g_m}{\partial \phi} V_1 + g_m \frac{\partial V_1}{\partial \phi}$$

$$I_{G_2} = G_2 V_{G_2}, \text{ hence } \frac{\partial I_{G_2}}{\partial \phi} = \frac{\partial G_2}{\partial \phi} V_{G_2} + G_2 \frac{\partial V_{G_2}}{\partial \phi}$$

$$I_0 = 0, \text{ hence } \frac{\partial I_0}{\partial \phi} = 0$$

Substitute branch relations directly and collect up terms

$$\begin{aligned} & \frac{\partial v_g}{\partial \phi} \hat{I}_g + \frac{\partial v_{G_1}}{\partial \phi} (\hat{I}_{G_1} - G_1 \hat{V}_{G_1}) + \frac{\partial v_{C_1}}{\partial \phi} (\hat{I}_{C_1} - j\omega C_1 \hat{V}_{C_1}) \\ & + \frac{\partial v_{C_2}}{\partial \phi} (\hat{I}_{C_2} - j\omega C_2 \hat{V}_{C_2}) + \frac{\partial v_1}{\partial \phi} (\hat{I}_1 - g_m \hat{V}_2) \\ & + \frac{\partial v_2}{\partial \phi} \hat{I}_2 + \frac{\partial v_{G_2}}{\partial \phi} (\hat{I}_{G_2} - G_2 \hat{V}_{G_2}) + \frac{\partial v_0}{\partial \phi} \hat{I}_0 \\ & - \frac{\partial G_1}{\partial \phi} V_{G_1} \hat{V}_{G_1} - \frac{\partial(j\omega C_1)}{\partial \phi} V_{C_1} \hat{V}_{C_1} - \frac{\partial(j\omega C_2)}{\partial \phi} V_{C_2} \hat{V}_{C_2} \\ & - \frac{\partial g_m}{\partial \phi} V_1 \hat{V}_2 - \frac{\partial G_2}{\partial \phi} V_{G_2} \hat{V}_{G_2} = 0 \end{aligned}$$

Define adjoint circuit branch relations to simplify this equation

$$\hat{I}_g = 0$$

$$\hat{I}_{G_1} = G_1 \hat{V}_{G_1}$$

$$\hat{I}_{C_1} = j\omega C_1 \hat{V}_{C_1}$$

$$\hat{I}_{C_2} = j\omega C_2 \hat{V}_{C_2}$$

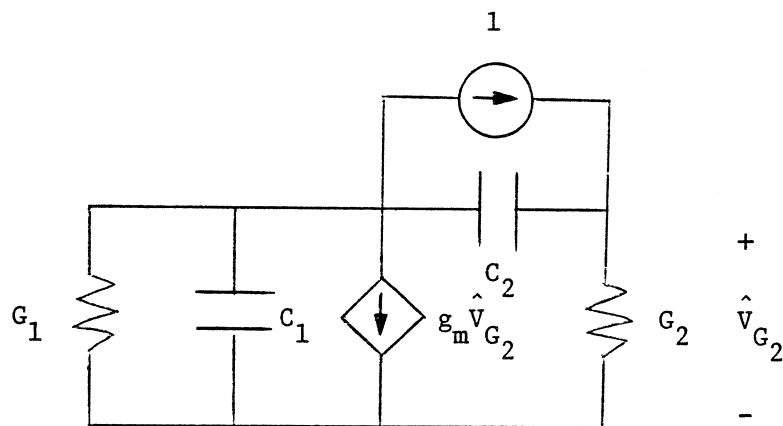
$$\hat{I}_1 = g_m \hat{V}_2$$

$$\hat{I}_2 = 0$$

$$\hat{I}_{G_2} = G_2 \hat{V}_{G_2}$$

$$\hat{I}_0 = 1$$

Adjoint Circuit



Hence,

$$\begin{aligned} \frac{\partial V_0}{\partial \phi} &= \frac{\partial G_1}{\partial \phi} V_{G_1} \hat{V}_{G_1} + \frac{\partial(j\omega C_1)}{\partial \phi} V_C \hat{V}_{C_1} + \frac{\partial(j\omega C_2)}{\partial \phi} V_{C_2} \hat{V}_{C_2} \\ &\quad + \frac{\partial g_m}{\partial \phi} V_{C_1} \hat{V}_{G_2} + \frac{\partial G_2}{\partial \phi} V_{G_2} \hat{V}_{G_2} \end{aligned}$$

Let  $\phi = G_1$ , hence  $\frac{\partial V_0}{\partial G_1} = V_{G_1} \hat{V}_{G_1}$

Let  $\phi = C_1$ , hence  $\frac{\partial V_0}{\partial C_1} = j\omega V_{C_1} \hat{V}_{C_1}$

Let  $\phi = C_2$ , hence  $\frac{\partial V_0}{\partial C_2} = j\omega V_{C_2} \hat{V}_{C_2}$



Let  $\phi = g_m$ , hence  $\frac{\partial V_0}{\partial g_m} = V_{C_1} \hat{V}_{G_2}$  [sic.]

Let  $\phi = G_2$ , hence  $\frac{\partial V_0}{\partial G_2} = V_{G_2} \hat{V}_{G_2}$

$V_0 = V_{C_2}$ , hence  $\partial V_0 / \partial . \equiv \partial V_{C_2} / \partial .$

A SOLUTION TO QUESTION 110 OF SECTION TWO II

Let

$$\underline{Y} \underline{V} = \underline{I}$$

represent the nodal equations, where  $\underline{Y}$  is the nodal admittance matrix and  $\underline{V}$  is the node voltage vector. Assume  $\underline{I}$  is the node current source vector, assumed independent of the variable  $\phi$ . Then

$$\frac{\partial \underline{Y}}{\partial \phi} \underline{V} + \underline{Y} \frac{\partial \underline{V}}{\partial \phi} = \underline{0}.$$

Hence,

$$\underline{Y} \frac{\partial \underline{V}}{\partial \phi} = - \frac{\partial \underline{Y}}{\partial \phi} \underline{V}$$

or

$$\frac{\partial \underline{V}}{\partial \phi} = - \underline{Y}^{-1} \frac{\partial \underline{Y}}{\partial \phi} \underline{V}.$$

Let

$$\underline{u}_1 \triangleq \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \underline{u}_2 \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \dots$$

Then

$$\underline{u}_1^T \frac{\partial \underline{V}}{\partial \phi} = \frac{\partial V_1}{\partial \phi} = - \underline{u}_1^T \underline{Y}^{-1} \frac{\partial \underline{Y}}{\partial \phi} \underline{V}$$

$$\underline{u}_2^T \frac{\partial \underline{V}}{\partial \phi} = \frac{\partial V_2}{\partial \phi} = - \underline{u}_2^T \underline{Y}^{-1} \frac{\partial \underline{Y}}{\partial \phi} \underline{V}$$

etc., so that, subtracting

$$\frac{\partial(\tilde{V}_1 - V_2)}{\partial\phi} = - (\tilde{u}_1 - u_2)^T \tilde{Y}^{-1} \frac{\partial \tilde{Y}}{\partial\phi} \tilde{V} .$$

Let

$$\tilde{Y}^T \hat{\tilde{V}} = \tilde{u}_1 - u_2$$

which implies that

$$\hat{\tilde{V}}^T = (\tilde{u}_1 - u_2)^T \tilde{Y}^{-1} .$$

In this case

$$\frac{\partial(V_1 - V_2)}{\partial\phi} = - \hat{\tilde{V}}^T \frac{\partial \tilde{Y}}{\partial\phi} \tilde{V} .$$

For the 2-node circuit in question

$$\tilde{Y} = \begin{pmatrix} G_1 + j\omega C_1 + j\omega C_2 & -j\omega C_2 \\ -j\omega C_2 + g_m & G_2 + j\omega C_2 \end{pmatrix} ,$$

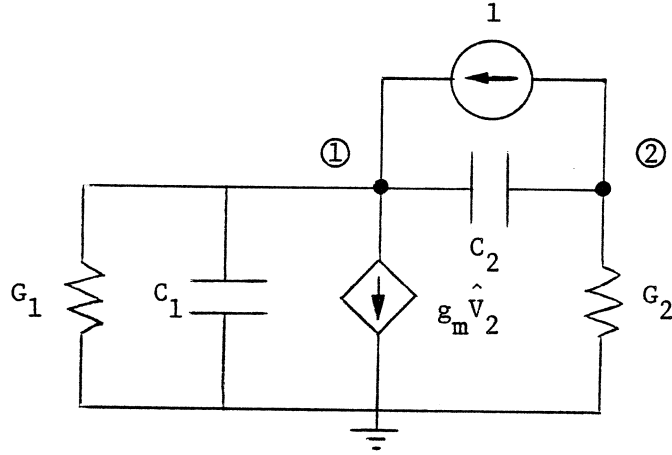
$$\tilde{I} = \begin{pmatrix} I \\ g \\ 0 \end{pmatrix} .$$

Thus the adjoint circuit is represented by

$$\hat{\tilde{Y}} = \tilde{Y}^T = \begin{pmatrix} G_1 + j\omega C_1 + j\omega C_2 & -j\omega C_2 + g_m \\ -j\omega C_2 & G_2 + j\omega C_2 \end{pmatrix} ,$$

$$\hat{\tilde{I}} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} .$$

Adjoint Circuit



$$\frac{\partial(V_1 - V_2)}{\partial G_1} = - [\hat{V}_1 \quad \hat{V}_2] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = - V_1 \hat{V}_1$$

$$\frac{\partial(V_1 - V_2)}{\partial C_1} = - [\hat{V}_1 \quad \hat{V}_2] \begin{bmatrix} j\omega & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = - j\omega V_1 \hat{V}_1$$

$$\frac{\partial(V_1 - V_2)}{\partial C_2} = - [\hat{V}_1 \quad \hat{V}_2] \begin{bmatrix} j\omega & -j\omega \\ -j\omega & j\omega \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = - j\omega(V_1 - V_2)(\hat{V}_1 - \hat{V}_2)$$

$$\frac{\partial(V_1 - V_2)}{\partial g_m} = - [\hat{V}_1 \quad \hat{V}_2] \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = - V_1 \hat{V}_2$$

$$\frac{\partial(V_1 - V_2)}{\partial G_2} = - [\hat{V}_1 \quad \hat{V}_2] \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = - V_2 \hat{V}_2$$

In summary, letting  $V_{C_2} = V_1 - V_2$ ,  $V_1 = V_{G_1} = V_{C_1}$ ,  $V_2 = V_{G_2}$ , with corresponding notation for the adjoint circuit,

$$\partial V_{C_2} / \partial G_1 = - V_{G_1} \hat{V}_{G_1}$$

$$\partial V_{C_2} / \partial C_1 = - j_{\omega} V_{C_1} \hat{V}_{C_1}$$

$$\partial V_{C_2} / \partial C_2 = - j_{\omega} V_{C_2} \hat{V}_{C_2}$$

$$\partial V_{C_2} / \partial g_m = - V_{C_1} \hat{V}_{G_2} \quad [\text{sic.}]$$

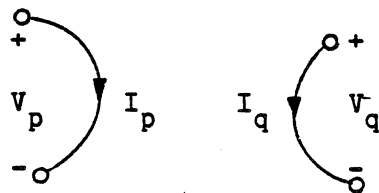
$$\partial V_{C_2} / \partial G_2 = - V_{G_2} \hat{V}_{G_2}$$

A SOLUTION TO QUESTION 123 OF SECTION TWO

Given

$$\begin{pmatrix} V_p \\ I_p \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} V_q \\ -I_q \end{pmatrix}$$

with the graph of the two-port as shown.



Tellegen's theorem can be written for the whole network. The term corresponding to the element in question is indicated by

$$\dots + [\hat{I}_p \quad -\hat{V}_p] \begin{pmatrix} V_p \\ I_p \end{pmatrix} + [\hat{I}_q \quad \hat{V}_q] \begin{pmatrix} V_q \\ -I_q \end{pmatrix} + \dots = 0$$

Differentiating w.r.t.  $\phi$  we have

$$\dots + [\hat{I}_p \quad -\hat{V}_p] \frac{\partial}{\partial \phi} \begin{pmatrix} V_p \\ I_p \end{pmatrix} + [\hat{I}_q \quad \hat{V}_q] \frac{\partial}{\partial \phi} \begin{pmatrix} V_q \\ -I_q \end{pmatrix} + \dots = 0$$

The branch relation yields

$$\frac{\partial}{\partial \phi} \begin{pmatrix} V_p \\ I_p \end{pmatrix} = \frac{\partial}{\partial \phi} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} V_q \\ -I_q \end{pmatrix} + \begin{pmatrix} A & B \\ C & D \end{pmatrix} \frac{\partial}{\partial \phi} \begin{pmatrix} V_q \\ -I_q \end{pmatrix}$$

which is substituted accordingly to give

$$\dots + [\hat{I}_p \quad -\hat{V}_p] \frac{\partial}{\partial \phi} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} V_q \\ -I_q \end{pmatrix} + ([\hat{I}_p \quad -\hat{V}_p] \begin{pmatrix} A & B \\ C & D \end{pmatrix} + [\hat{I}_q \quad \hat{V}_q]) \frac{\partial}{\partial \phi} \begin{pmatrix} V_q \\ -I_q \end{pmatrix} + \dots = 0$$

Define the adjoint element by letting

$$[\hat{I}_q \quad \hat{V}_q] = - [\hat{I}_p \quad -\hat{V}_p] \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

or, rearranging suitably for appearance,

$$\begin{pmatrix} \hat{V}_q \\ \hat{I}_q \end{pmatrix} = \begin{pmatrix} D & B \\ C & A \end{pmatrix} \begin{pmatrix} \hat{V}_p \\ -\hat{I}_p \end{pmatrix}$$

which indicates analysis in the reverse direction. The sensitivity formula is then given by the remaining term

$$[\hat{I}_p \quad -\hat{V}_p] \frac{\partial}{\partial \phi} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} V_q \\ -I_q \end{pmatrix}$$

For a series impedance element

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} 1 & Z \\ 0 & 1 \end{pmatrix}$$

hence

$$\begin{pmatrix} \hat{V}_q \\ \hat{I}_q \end{pmatrix} = \begin{pmatrix} 1 & Z \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{V}_p \\ -\hat{I}_p \end{pmatrix}$$

and

$$\begin{aligned} & [\hat{I}_p \quad -\hat{V}_p] \begin{pmatrix} 0 & \frac{\partial Z}{\partial \phi} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_q \\ -I_q \end{pmatrix} \\ &= - I_q \hat{I}_p \frac{\partial Z}{\partial \phi} \\ &= I_p \hat{I}_p \frac{\partial Z}{\partial \phi} \\ &= I_q \hat{I}_q \frac{\partial Z}{\partial \phi} \end{aligned}$$

since

$$I_p = - I_q.$$

A SOLUTION TO QUESTION 130 OF SECTION TWO

(a) After verification of  $\underline{L}$  and  $\underline{U}$  we check

$$\underline{L} \underline{U} = \begin{pmatrix} 3 & 0 & 0 \\ -2 & 11/3 & 0 \\ 0 & -2 & 21/11 \end{pmatrix} \begin{pmatrix} 1 & -2/3 & 0 \\ 0 & 1 & -6/11 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{pmatrix} = \underline{G}$$

Now

$$\underline{L} \underline{U} \underline{V} = \underline{I} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

hence

$$\underline{L} \underline{z} = \underline{I}$$

gives, by forward substitution

$$z_1 = 1/3, z_2 = 2/11, z_3 = 4/21$$

and

$$\underline{U} \underline{V} = \underline{z}$$

gives, by backward substitution

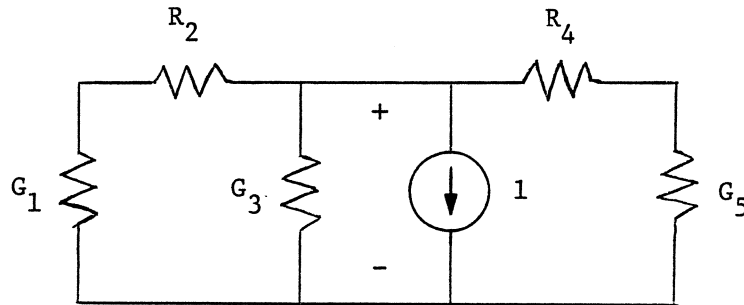
$$v_3 = 4/21, v_2 = 2/7, v_1 = 11/21$$

Check:

$$\frac{1}{21} \begin{pmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{pmatrix} \begin{pmatrix} 11 \\ 6 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



(b) Adjoint circuit



This arrangement fits the general formula

$$\sum_{\text{voltage sources}} \hat{V} \nabla I - \sum_{\text{current sources}} \hat{I} \nabla V = G$$

where  $G$  contains sensitivity expressions which have been tabulated.

We have no voltage sources so that, letting  $\hat{I} = 1$

$$\nabla V = -G$$

(c) We have

$$\tilde{G}^T \hat{V} = \hat{I} = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

But  $G \equiv \tilde{G}^T$  in this example. Hence

$$\underline{L} \underline{U} \hat{V} = \hat{I}$$

and

$$\underline{L} \hat{z} = \hat{I}$$

gives by forward substitution

$$\hat{z}_1 = 0, \hat{z}_2 = -3/11, \hat{z}_3 = -6/21$$

and

$$\underline{U} \hat{V} = \hat{z}$$

gives, by backward substitution

$$\hat{V}_3 = -6/21, \hat{V}_2 = -3/7, \hat{V}_1 = -2/7$$

Thus

$$\begin{aligned} \underline{V} &= - \begin{pmatrix} -V_{G_1} & \hat{V}_{G_1} \\ I_{R_2} & \hat{I}_{R_2} \\ -V_{G_3} & \hat{V}_{G_3} \\ I_{R_4} & \hat{I}_{R_4} \\ V_{G_5} & \hat{V}_{G_5} \end{pmatrix} = \begin{pmatrix} V_1 \hat{V}_1 \\ -(V_1 - V_2)(\hat{V}_1 - \hat{V}_2)/R_2^2 \\ V_2 \hat{V}_2 \\ -(V_2 - V_3)(\hat{V}_2 - \hat{V}_3)/R_4^2 \\ V_3 \hat{V}_3 \end{pmatrix} = \begin{pmatrix} -\frac{22}{147} \\ -\left(\frac{5}{21}\right)\left(\frac{1}{7}\right)4 \\ -\frac{6}{49} \\ +\left(\frac{2}{21}\right)\left(\frac{3}{21}\right)4 \\ -\frac{8}{147} \end{pmatrix} \\ &= \frac{1}{147} \begin{pmatrix} -22 \\ -20 \\ -18 \\ 8 \\ -8 \end{pmatrix} = \begin{pmatrix} -0.1497 \\ -0.1361 \\ -0.1224 \\ +0.0544 \\ -0.0544 \end{pmatrix} \end{aligned}$$

A SOLUTION TO QUESTION 113 OF SECTION TWO

Prob # 113

We write the nodal equations

$$\tilde{Y} \tilde{V} = \tilde{I} \quad (1)$$

where  $\tilde{I}$  is specified to be independent of  $\omega$ .

Differentiating (1) w.r.t.  $\omega$ , we get

$$\frac{\partial \tilde{Y}}{\partial \omega} \tilde{V} + \tilde{Y} \frac{\partial \tilde{V}}{\partial \omega} = \tilde{0}$$

$$\text{or} \quad \tilde{Y} \frac{\partial \tilde{V}}{\partial \omega} = - \frac{\partial \tilde{Y}}{\partial \omega} \tilde{V} \quad (2)$$

Premultiplying both sides of (2) by  $\tilde{Y}^{-1}$ , we write

$$\frac{\partial \tilde{V}}{\partial \omega} = - \tilde{Y}^{-1} \frac{\partial \tilde{Y}}{\partial \omega} \tilde{V} \quad (3)$$

$$\text{Now} \quad \frac{\partial v_i}{\partial \omega} = \hat{u}_i^T \frac{\partial \tilde{V}}{\partial \omega} = - \hat{u}_i^T \tilde{Y}^{-1} \frac{\partial \tilde{Y}}{\partial \omega} \tilde{V}$$

$$\text{or} \quad \frac{\partial v_i}{\partial \omega} = - \hat{V}_i^T \frac{\partial \tilde{Y}}{\partial \omega} \tilde{V} \quad (4)$$

where  $\hat{V}_i$  is obtained from the solution of the adjoint system

$$\tilde{Y}^T \hat{V}_i = \hat{u}_i \quad (5)$$

A SOLUTION TO QUESTION 114 OF SECTION TWO

Prob # 114

The nodal equations are

$$\underline{Y} \underline{V} = \underline{I} \quad , \quad (1)$$

and let  $\underline{f}$  be a complex vector given by

$$\underline{f} = \underline{Y} \underline{V} - \underline{I}$$

or

$$\underline{f} = \underline{Y} (\underline{V}_R + j \underline{V}_I) - \underline{I} \quad , \quad (2)$$

where  $\underline{V} \triangleq \underline{V}_R + j \underline{V}_I$ ,  $\underline{V}_R$  and  $\underline{V}_I$  are vectors comprising real and imaginary parts of the complex nodal voltages, respectively.

The LSO objective is expressed as

$$U = \underline{f}^{*T} \underline{f} \quad , \quad (3)$$

and the gradient vector  $\underline{\nabla} U$  is obtained using (2) and (3) as

$$\underline{\nabla} U = \underline{J}^{*T} \underline{f} + \underline{J}^T \underline{f}^* = 2 \operatorname{Re} \{ \underline{J}^T \underline{f}^* \} \quad , \quad (4)$$

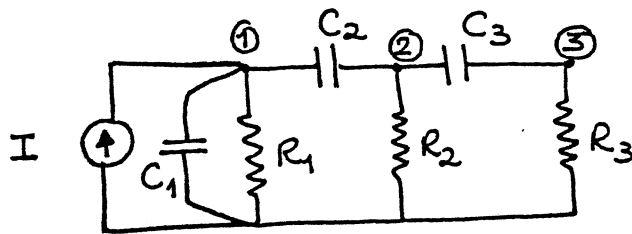
where

$$\underline{J} \triangleq \begin{bmatrix} \underline{Y} & j \underline{Y} \end{bmatrix}$$

A SOLUTION TO QUESTION 133 OF SECTION TWO

Prob# 133

The network voltage source is transformed into a current source and the network is represented as



$$\omega = 2 \text{ rad/sec.}$$

$$I = j2 \text{ A}, \quad C_1 = C_2 = C_3 = 1 \text{ F}, \quad R_1 = R_2 = R_3 = 2 \text{ } \Omega$$

Part a

The nodal equations are

$$\begin{bmatrix} 0.5 + j4.0 & -j2.0 & 0 \\ -j2.0 & 0.5 + j4.0 & -j2.0 \\ 0 & -j2.0 & 0.5 + j2.0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 0 + j2 \\ 0 \\ 0 \end{bmatrix} \quad (1)$$

Part b

Using (1), we write

$$V_1 = \frac{1}{0.5 + j4.0} (j2 + j2V_2) = \frac{1 + V_2}{2.0 - j0.25} \quad (2a)$$

$$V_2 = \frac{1}{0.5 + j4.0} (j2V_1 + j2V_3) = \frac{V_1 + V_3}{2.0 - j0.25} \quad (2b)$$

$$V_3 = \frac{1}{0.5 + j2.0} (j2V_2) = \frac{V_2}{1.0 - j0.25} \quad (2c)$$

We use the given initial guess in (2) and obtain G-S (relaxation method) two iterations as

$$\tilde{V}^1 = \begin{bmatrix} 0.492308 + j 0.061538 \\ 0.238580 + j 0.060592 \\ 0.210289 + j 0.113164 \end{bmatrix}$$

$$\tilde{V}^2 = \begin{bmatrix} 0.606034 + j 0.106050 \\ 0.388392 + j 0.158156 \\ 0.328332 + j 0.240239 \end{bmatrix}$$

The overrelaxation factor  $\alpha = 1.5$  is used in updating the voltages

$$\tilde{V}^n = (1-\omega) \tilde{V}^{n-1} + \omega \bar{V}^n, \quad (3)$$

where  $\bar{V}^n$  are calculated by using (2). The results for two iterations are

$$\tilde{V}^1 = \begin{bmatrix} 0.738462 + j 0.092308 \\ 0.536805 + j 0.136331 \\ 0.709725 + j 0.381928 \end{bmatrix}$$

$$\tilde{V}^2 = \begin{bmatrix} 0.753056 + j 0.196381 \\ 0.758423 + j 0.493919 \\ 0.541528 + j 0.774012 \end{bmatrix}$$

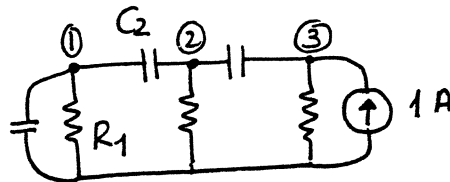
Part c

The LU factors of  $\underline{Y}$  are

$$\underline{\tilde{L}}\underline{U} = \begin{bmatrix} 0.50 + j4.0 & -0.49 - j0.06 & 0 \\ -j2.0 & 0.62 + j3.02 & -0.64 - j1.13 \\ 0 & -j2.0 & 0.76 + j0.73 \end{bmatrix}$$

Part d

The adjoint network for finding sensitivities of  $V_3$  is



Using the adjoint and original solution in the given sensitivity expressions, we get

$$\frac{\partial V_3}{\partial C_2} = 0.183369 + j0.063625$$

$$\frac{\partial V_3}{\partial R_1} = 0.056058 + j0.001602$$

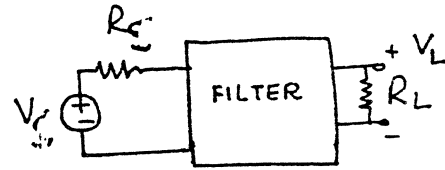
Part e

$$\begin{aligned} \Delta V_3 &= \frac{\partial V_3}{\partial C_2} \cdot 0.03 + \frac{\partial V_3}{\partial R_1} \cdot -0.10 \\ &= -0.00010471 + j0.00174853 \end{aligned}$$

By direct perturbation  $\Delta V_3 = -0.00052464 + j0.00162645$

A SOLUTION TO QUESTION 150 OF SECTION TWO

Prob # 150



The two objective functions specified are

$$U = \left( \sum_{\omega_i \in \Omega_d} |L(\omega_i) - S(\omega_i)|^p \right)^{1/p}, \quad p > 1 \quad (1a)$$

$$U = \sum_{\omega_i \in \Omega_d} [L(\omega_i) - S(\omega_i)]^p, \quad p \text{ even} > 0. \quad (1b)$$

We express the insertion loss  $L(\omega_i)$  of the filter as

$$\begin{aligned} L &= -20 \log_{10} \left| \frac{V_L}{V_g} \right| - 20 \log_{10} \left( \frac{R_g + R_L}{R_L} \right) \\ &= -20 \log_{10} |V_L| + \text{constant}, \end{aligned} \quad (2)$$

where  $R_g$ ,  $R_L$  and  $V_g$  are assumed to be constants.

The gradient vector of  $L$  is written as

$$\underline{\nabla} L = - \frac{20}{\ln 10 |V_L|^2} \text{Re} \{ V_L^* \underline{\nabla} V_L \} \quad (3)$$

and we proceed to find  $\underline{\nabla} V_L$ .

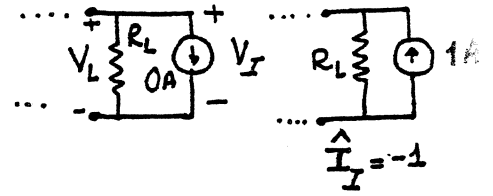
Using the perturbed Tellegen theorem<sup>†</sup>, we write

$$-\Delta V_{\mathbf{I}} \hat{\mathbf{I}}_{\mathbf{I}} = \underline{\mathbf{G}}^T \Delta \underline{\phi}. \quad (4)$$

---

<sup>†</sup> J.W. Bandler, Chapter 6 on Computer-Aided Circuit Optimization  
page 257





The excitations of the original and adjoint networks reduce (4) to simply

$$\tilde{\nabla} V_L = \tilde{G} \quad (5)$$

Rewriting (3) as

$$\tilde{\nabla} L = \frac{-20}{\ln 10 |V_L|^2} \operatorname{Re} \{ V_L^* \tilde{G} \} \quad (6)$$

the expressions relating  $\tilde{\nabla} U$  to  $\tilde{G}$  are

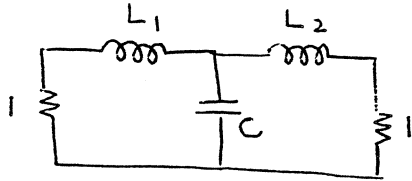
$$\begin{aligned} \tilde{\nabla} U &= \frac{1}{p} \left( \sum_{\omega_i \in \Omega_d} |L(\omega_i) - S(\omega_i)|^p \right)^{\frac{1}{p}-1} \\ &\quad \sum_{\omega_i \in \Omega_d} \operatorname{Re} \{ p |L(\omega_i) - S(\omega_i)|^{p-2} \cdot [L(\omega_i) - S(\omega_i)]^* \tilde{\nabla} L \end{aligned} \quad (7a)$$

$$\tilde{\nabla} U = \sum_{\omega_i \in \Omega_d} p [L(\omega_i) - S(\omega_i)]^{p-1} \tilde{\nabla} L \quad (7b)$$

for the objective functions in (1a) and (1b), respectively.

A SOLUTION TO QUESTION 128 OF SECTION TWO

Q 128



$$L_1 = L_2 = 2 \text{ H}$$

$$C = 1 \text{ F}$$

$$\omega = 1 \text{ rad/s}$$

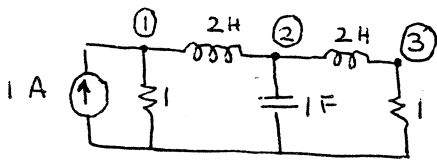
Insertion loss in dB =  $\Lambda = h_1 + h_2 \ln |V_0|$

$$h_1 = 20 \log_{10} (1/2) = -6.0206 \quad h_2 = -20 \log_{10} e = -8.6859$$

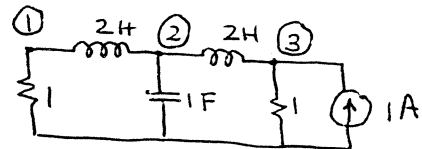
$$\frac{\partial \Lambda}{\partial \phi} = h_2 \operatorname{Re} \left\{ \frac{1}{V_0} \frac{\partial V_0}{\partial \phi} \right\}$$

We need  $\nabla V_0 = \begin{bmatrix} \frac{\partial V_0}{\partial L_1} & \frac{\partial V_0}{\partial L_2} & \frac{\partial V_0}{\partial C} \end{bmatrix}^T$

Original Network



Adjoint Network



$$\frac{\partial V_0}{\partial L_1} = j\omega I_{L_1} \hat{I}_{L_1} = j\omega \left( \frac{V_1 - V_2}{j\omega L_1} \right) \left( \frac{\hat{V}_1 - \hat{V}_2}{j\omega L_1} \right)$$

$$\frac{\partial V_0}{\partial L_2} = j\omega I_{L_2} \hat{I}_{L_2} = j\omega \left( \frac{V_2 - V_3}{j\omega L_2} \right) \left( \frac{\hat{V}_2 - \hat{V}_3}{j\omega L_2} \right)$$

$$\frac{\partial V_0}{\partial C} = -j\omega V_C \hat{V}_C = -j\omega V_2 \hat{V}_2$$

For the original Network :

$$\begin{bmatrix} 1 + \frac{1}{j\omega L_1} & \frac{-1}{j\omega L_1} & 0 \\ \frac{-1}{j\omega L_1} & \frac{1}{j\omega L_1} + j\omega C + \frac{1}{j\omega L_2} & \frac{-1}{j\omega L_2} \\ 0 & \frac{-1}{j\omega L_2} & 1 + \frac{1}{j\omega L_2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

For the adjoint: RHS =  $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

Substituting values we get :

$$V_1 = 0.4 + j0.2 \quad V_2 = -j \quad V_3 = -0.4 - j0.2$$

Since  $L_1 = L_2$ , from symmetry of equations we get

$$\hat{V}_1 = V_3 = -0.4 - j0.2 \quad \hat{V}_2 = V_2 = -j \quad \text{and} \quad \hat{V}_3 = V_1 = 0.4 + j0.2$$

Now :

$$\frac{\partial V_0}{\partial L_1} = \frac{\partial V_0}{\partial L_2} = -0.04 + j0.28$$

$$\frac{\partial V_0}{\partial C} = j$$

We now calculate  $\frac{\partial \Lambda}{\partial \phi}$  using  $\frac{\partial V_0}{\partial \phi}$

$$\frac{\partial \Lambda}{\partial L_1} = \frac{\partial \Lambda}{\partial L_2} = h_2 \operatorname{Re} \left\{ \frac{-0.04 + j0.28}{-0.4 - j0.2} \right\} = -0.2 h_2 = \underline{\underline{1.7372}}$$

$$\frac{\partial \Lambda}{\partial C} = h_2 \operatorname{Re} \left\{ \frac{j}{-0.4 - j0.2} \right\} = -h_2 = \underline{\underline{8.6859}}$$

$$\Delta \Lambda = \sum \frac{\partial \Lambda}{\partial \phi} \Delta \phi = \frac{\partial \Lambda}{\partial L_1} \Delta L_1 + \frac{\partial \Lambda}{\partial L_2} \Delta L_2 + \frac{\partial \Lambda}{\partial C} \Delta C$$

if  $\Delta L_1 = -\Delta L_2$  then  $\Delta \Lambda = \frac{\partial \Lambda}{\partial C} \Delta C = -h_2 \times 0.1 = \underline{\underline{0.8686}}$

Direct Method :

Before change  $\Lambda_1 = h_1 + h_2 \ln |-0.4 - j0.2| = h_1 + \frac{h_2}{2} \ln(0.2) = 0.9691$

After change :  $L_1 = 2.1 \quad L_2 = 1.9 \quad C = 1.1$

Calculate new  $V_0 \Rightarrow V_0 = \frac{1}{-2.4 + j0.711}$

$$\Lambda_2 = h_1 + \frac{h_2}{2} \ln \left( \frac{1}{(2.4)^2 + (0.711)^2} \right) = 1.9490$$

$$\Delta \Lambda = \Lambda_2 - \Lambda_1 = \underline{\underline{0.9799}}$$



**SECTION THIRTEEN**  
**METHOD OF LAGRANGE MULTIPLIERS**

© J.W. Bandler 1984, 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



## METHOD OF LAGRANGE MULTIPLIERS

A classical minimization problem is

$$\text{Minimize } U(\underline{\phi})$$

$$\text{subject to } \underline{h}(\underline{\phi}) = 0.$$

The method of undetermined multipliers or Lagrange multipliers is to find a stationary point for

$$L(\underline{\phi}, \underline{\lambda}) \triangleq U(\underline{\phi}) + \underline{\lambda}^T \underline{h}(\underline{\phi}),$$

where

$$\underline{\lambda} \triangleq \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \cdot \\ \cdot \\ \lambda_s \end{bmatrix}.$$

Thus we must solve the system of  $s+k$  equations

$$\underline{\nabla} U + \sum_{i=1}^s \lambda_i \underline{\nabla} h_i = \underline{0},$$

$$\underline{h} = \underline{0},$$

in  $k$  unknowns  $\underline{\phi}$  and  $s$  unknowns  $\underline{\lambda}$ . The reason is immediately apparent

if we write out these equations in expanded form.

$$\begin{bmatrix} \frac{\partial L}{\partial \phi_1} \\ \frac{\partial L}{\partial \phi_2} \\ \vdots \\ \frac{\partial L}{\partial \phi_k} \\ \frac{\partial L}{\partial \lambda_1} \\ \frac{\partial L}{\partial \lambda_2} \\ \vdots \\ \frac{\partial L}{\partial \lambda_s} \end{bmatrix} = \begin{bmatrix} \nabla U + \sum_{i=1}^s \lambda_i \nabla h_i \\ h \end{bmatrix} = \underline{0}.$$

Let  $(\underline{\phi}^0, \underline{\lambda}^0)$  be a stationary point for the Lagrangian. In the neighbourhood of  $(\underline{\phi}^0, \underline{\lambda}^0)$ , we may write, for  $h(\underline{\phi}) = 0$ ,

$$U(\underline{\phi}) = L(\underline{\phi}, \underline{\lambda}^0) \begin{matrix} \swarrow \text{min } U \\ > \\ < \\ \nwarrow \text{max } U \end{matrix} L(\underline{\phi}^0, \underline{\lambda}^0) = L(\underline{\phi}^0, \underline{\lambda}) = U(\underline{\phi}^0).$$

Thus  $(\underline{\phi}^0, \underline{\lambda}^0)$  is actually a degenerate saddle point. To handle inequality constraints we can generalize the Lagrange multiplier technique. But the computational effort is greatly increased.



**SECTION FOURTEEN**

**A COMPREHENSIVE EXAMPLE OF MINIMAX OPTIMIZATION,  
THE ADJOINT METHOD AND SPARSE MATRIX MANIPULATIONS**

© J.W. Bandler 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



This example illustrates the conceptual details and usage of minimax optimization, adjoint network and sparse matrix manipulations.

Consider the amplifier circuit shown in Fig. 1. It is required to achieve the voltage gain specifications over a prescribed frequency range by adjusting elements, namely,  $C_1$ ,  $C_2$  and  $R_3$ .

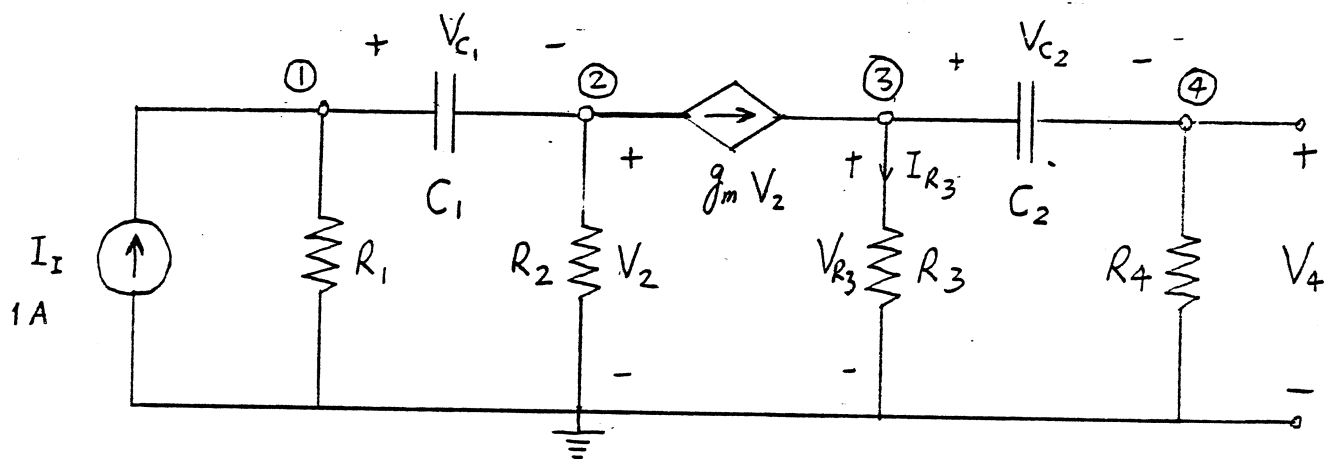


Fig. 1 A four-node nonreciprocal circuit.  
 $R_1 = 40 \Omega$ ,  $R_2 = 270 \Omega$ ,  $R_4 = 1500 \Omega$   
 $g_m = 0.38 \text{ S}$

Specifications:

$19.9 \text{ dB} \leq 20 \log |A_v| \leq 20.1 \text{ dB}$  over  
a frequency range  $2\pi \times 10^3 \text{ rad./sec} \leq \omega \leq 2\pi \times 10^6 \text{ rad./sec}$ ,  
where

$$A_v = \frac{V_4}{R_1 I_I}$$

[ See Fig. 2 ]

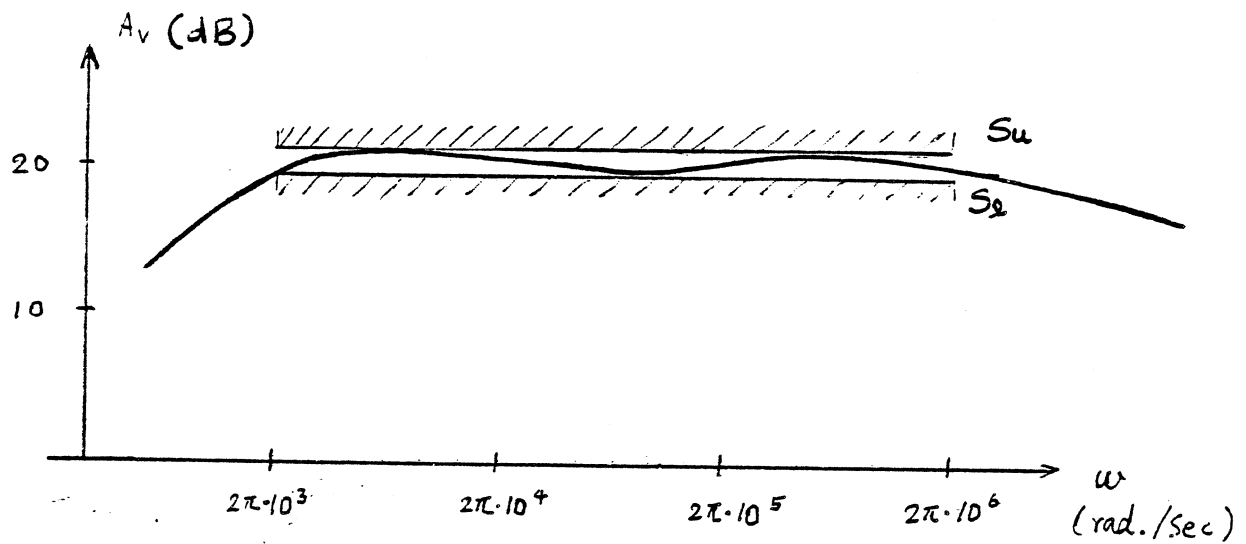


Fig. 2 Amplifier gain versus frequency.

Main features of the approach :

- (i) MMLC package is to be used for optimization.
- (ii) The gradient information and function evaluation are to be obtained using original network solution as well as adjoint network solution.
- (iii) ME28 (sparse matrix) package is to be used efficiently to solve both original and adjoint network nodal equations.

## SOLUTION

### 1. Theoretical analysis

The nodal admittance matrix of the circuit, shown in Fig. 1, can be written by inspection as

$$\underline{Y}(\underline{x}, \omega) = \begin{bmatrix} G_1 + j\omega C_1 & -j\omega C_1 & 0 & 0 \\ -j\omega C_1 & G_2 + j\omega C_1 + g_m & 0 & 0 \\ 0 & -g_m & G_3 + j\omega C_2 & -j\omega C_2 \\ 0 & 0 & -j\omega C_2 & G_4 + j\omega C_2 \end{bmatrix}, \quad (1)$$

where  $\underline{x} = [C_1 \ C_2 \ R_3]^T$ , and  $G_1 = 1/R_1$ ,  $G_2 = 1/R_2$ ,  $G_3 = 1/R_3$   
and  $G_4 = 1/R_4$ .

Remarks: (i) The  $\underline{Y}$  matrix becomes unsymmetrical owing to the presence of active elements, e.g., the voltage-controlled current source (VCCS) in the present problem results in  $Y_{23} \neq Y_{32}$ .

(ii) If any difficulty is encountered in writing  $\underline{Y}$  by inspection, the easier way could be to express (a) the  $\underline{Y}$  matrix without the VCCS and then (b) using KCL at the two nodes of interest, formulate two equations, and finally (c) estimate the location in  $\underline{Y}$  matrix for  $g_m$  terms. For instance, the KCL when applied at nodes ② and ③ yields:

$$(V_2 - V_1)j\omega C_1 + \frac{V_2}{R_2} + g_m V_2 = 0, \quad (2)$$

$$\text{and } -g_m V_2 + \frac{V_3}{R_3} + (V_3 - V_4)j\omega C_2 = 0, \quad (3)$$

respectively.

## 2. Original network solution

The nodal equations associated with the original network are expressed in the vector/matrix notation as

$$\underline{Y} \underline{V} = \underline{I}, \quad (4)$$

where the right-hand side current vector  $\underline{I} = [1 \ 0 \ 0 \ 0]^T$ , the nodal matrix  $\underline{Y}$  is given by (1), and  $\underline{V} = [V_1 \ V_2 \ V_3 \ V_4]^T$  is the unknown vector to be solved by ME28 package.

## 3. Formulation and solution of the adjoint network

The topologically similar adjoint network, corresponding to Fig. 1, for determining the sensitivities of a function involving  $V_4$ , is depicted in Fig. 3. The nodal admittance matrix of the adjoint network is simply

$$\hat{\underline{Y}} = \underline{Y}^T. \quad (5)$$

The rhs vector  $\hat{\underline{I}} = [0 \ 0 \ 0 \ 1]^T$  and the equation similar

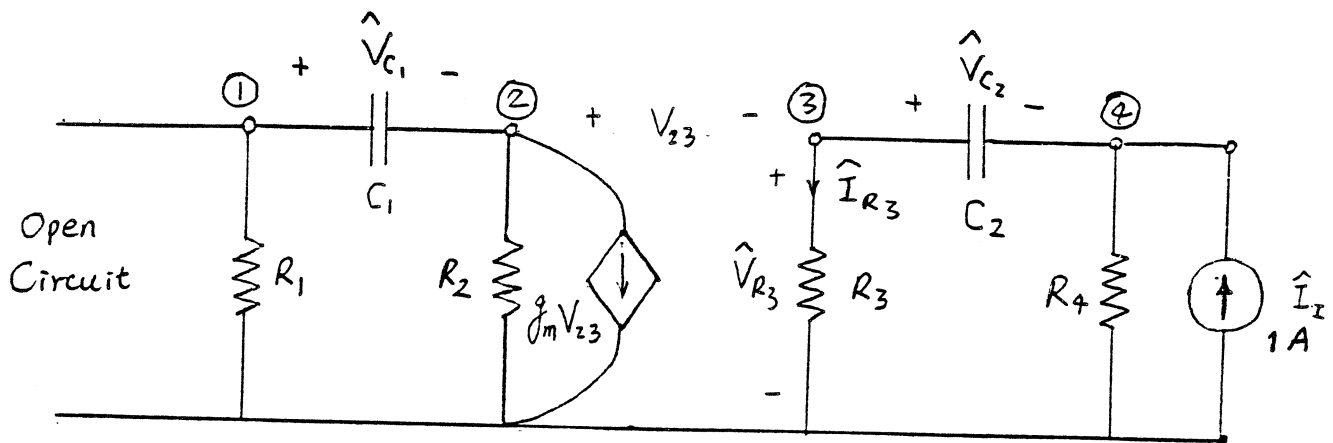


Fig. 3 Details of the adjoint network.

to (4) is

$$\hat{\underline{Y}} \hat{\underline{V}} = \hat{\underline{I}}, \quad (6)$$

where the unknown adjoint voltage vector  $\hat{\underline{V}} = [\hat{V}_1 \hat{V}_2 \hat{V}_3 \hat{V}_4]^T$ . This vector is also to be obtained by using ME2B package.

#### 4. Sensitivity evaluation using the adjoint network method

With reference to Table 6.2 (page 260) in Chapter 6, "Computer-Aided Circuit Optimization", shown below, the sensitivities of  $V_4$  w.r.t.  $\underline{x} = [C_1 C_2 R_3]^T$  are summarized as

$$\frac{\partial V_4}{\partial \underline{x}} = \begin{bmatrix} -j\omega V_{C1} & \hat{V}_{C1} \\ -j\omega V_{C2} & \hat{V}_{C2} \\ I_{R3} & \hat{I}_{R3} \end{bmatrix}. \quad (7)$$

Please notice that other circuit elements (e.g.,  $R_1, g_m, R_4, R_2$ ) are not the optimization variables, and are therefore excluded from  $\underline{x}$ . The voltages and currents in (7) are the respective branch quantities. From Figs. 1 and 3, it is straightforward to obtain the branch quantities as

TABLE 6-2 Sensitivity Expressions for Some Lumped and Distributed Elements

Element	Equation		Sensitivity (component of G)	Increment (component of $\Delta\phi$ )
	Original	Adjoint		
Resistor	$V = RI$ $I = GV$	$\hat{V} = R\hat{I}$ $\hat{I} = G\hat{V}$	$\hat{I}$ $-V\hat{V}$	$\Delta R$ $\Delta G$
Inductor	$V = j\omega LI$ $I = \frac{1}{j\omega} \Gamma V$	$\hat{V} = j\omega L\hat{I}$ $\hat{I} = \frac{1}{j\omega} \Gamma \hat{V}$	$j\omega L\hat{I}$ $-\frac{1}{j\omega} V\hat{V}$	$\Delta L$ $\Delta \Gamma$
Capacitor	$V = \frac{1}{j\omega} SI$ $I = j\omega CV$	$\hat{V} = \frac{1}{j\omega} S\hat{I}$ $\hat{I} = j\omega C\hat{V}$	$\frac{1}{j\omega} \hat{I}$ $-j\omega V\hat{V}$	$\Delta S$ $\Delta C$

$$V_{c1} = V_1 - V_2, \quad \hat{V}_{c1} = \hat{V}_1 - \hat{V}_2 \quad (8a)$$

$$V_{c2} = V_3 - V_4, \quad \hat{V}_{c2} = \hat{V}_3 - \hat{V}_4 \quad (8b)$$

$$\begin{aligned} I_{R3} &= V_{R3} / R_3, \text{ and } \hat{I}_{R3} = \hat{V}_{R3} / R_3 \\ &= V_3 / R_3 \qquad \qquad \qquad = \hat{V}_3 / R_3. \end{aligned} \quad (8c)$$

### 5. Voltage gain and its partial derivatives

The voltage gain of the circuit shown in Fig. 1 is expressed as

$$\begin{aligned} F(\lambda, \omega) &= 20 \log_{10} |A_v| = 20 \log_{10} |V_4 / R_4| \\ &= 20 \log_{10} |V_4| - 20 \log_{10} R_4 \end{aligned} \quad (9a)$$

$$= c_1 + c_2 \log_e |V_4|. \quad (9b)$$

The expression in (9b) is similar to the one used in tackling Assignment 2 EE3K4, and here only the general partial derivative is provided for brevity, that is,

$$\frac{\partial F}{\partial \phi} = c_2 \operatorname{Re} \left\{ \frac{1}{V_4} \frac{\partial V_4}{\partial \phi} \right\} \quad (10)$$

### 6. Selection of sample frequency points for optimization

The sample points are selected within the frequency range of interest, that is,  $2\pi \times 10^3$  to  $2\pi \times 10^6$  rad./sec, as shown in Fig. 2.

Based on experience and judgement, the accuracy of an analysis is warranted when more sample points are assigned in the range exhibiting rapid changes in the function. For well-behaved or smooth functions, fewer sample points are sufficient in order to avoid too many calculations. In the present example, 30 frequency points are chosen.

The spacing of the sample points can be either uniform or non-uniform. The overall frequency range



can be divided into several subintervals. We have 3 subintervals in this example:

$$2\pi \times 10^3 \text{ to } 2\pi \times 10^4, 2\pi \times 10^4 \text{ to } 2\pi \times 10^5 \text{ and } 2\pi \times 10^5 \text{ to } 2\pi \times 10^6.$$

Additionally, each subinterval is divided into 10 points.

The final frequency points are given by

$$\omega_1 = 2\pi \times 10^3, \omega_2 = 2\pi (2 \times 10^3), \dots, \omega_9 = 2\pi (9 \times 10^3), \omega_{10} = 2\pi (9.5 \times 10^3)$$

$$\omega_i = 10 \omega_{i-10} \quad \text{for } i = 11, 12, \dots, 29$$

and

$$\omega_{30} = 2\pi \times 10^6.$$

## 7. Definition of the objective function

The error functions associated with lower and upper specifications are formulated using conventional method, as used in Assignment 2.

$$e_l(x, \omega) = W_l [F(x, \omega) - 19.9] \quad (11a)$$

$$e_u(x, \omega) = W_u [F(x, \omega) - 20.1] \quad (11b)$$

For simplicity, assume  $W_l = W_u = 1$ . The error functions are then expressed as

$$\left. \begin{aligned} f_i &= e_u(x, \omega_i) \\ f_{i+30} &= -e_l(x, \omega_i) \end{aligned} \right\} i = 1, 2, \dots, 30. \quad (12)$$

The partial derivatives of these functions w.r.t.  $\underline{x} = [c_1, c_2, R_3]^T$  are compactly expressed as

$$\frac{\partial f_i}{\partial \underline{x}} = \frac{\partial F(x, \omega_i)}{\partial \underline{x}} \quad (13a)$$

$$\frac{\partial f_{i+30}}{\partial \underline{x}} = - \frac{\partial f_i}{\partial \underline{x}} \quad (13b)$$

Furthermore, the program setup and the flow chart are provided in

Figs. 4 and 5, respectively.

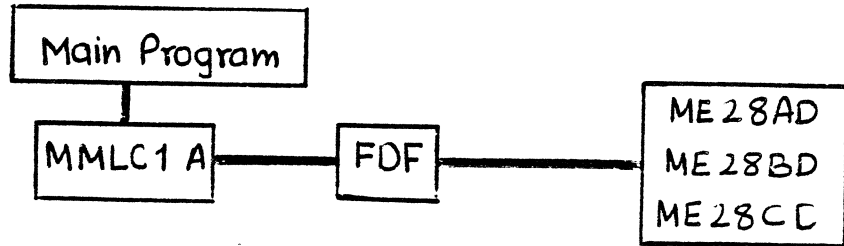


Fig. 4 Computer program structure.

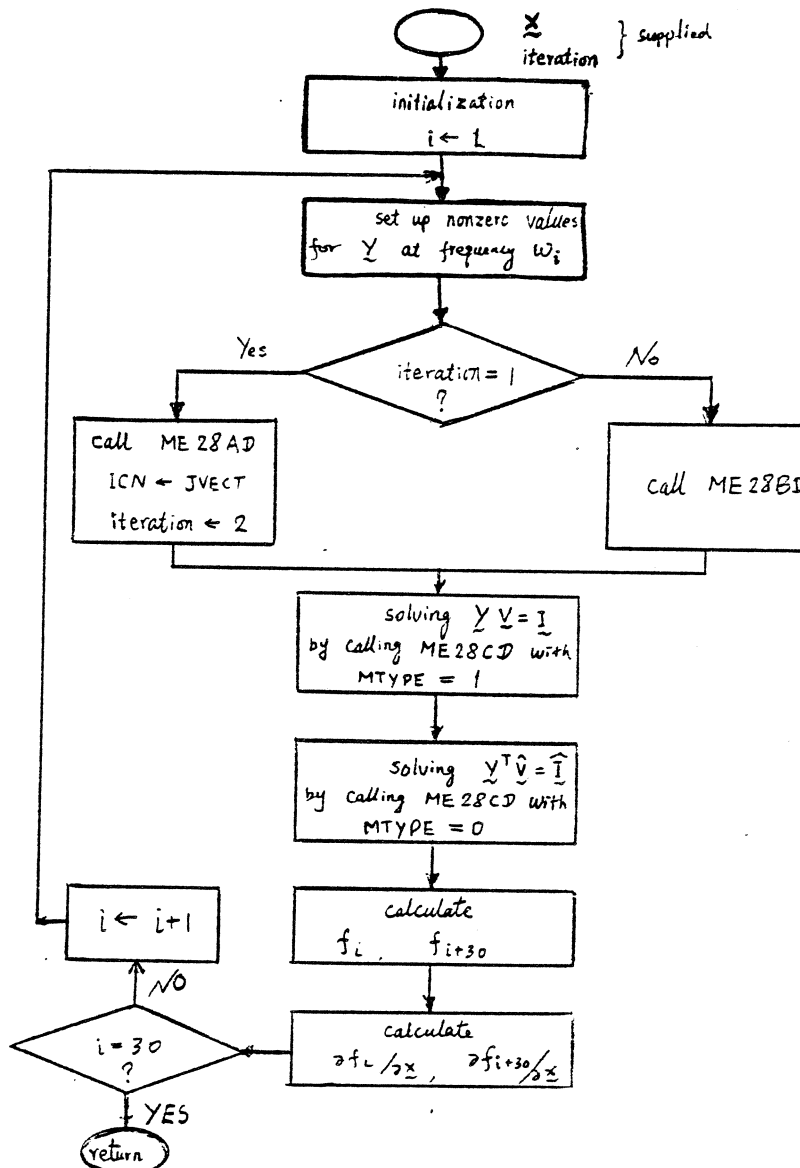


Fig. 5 A flow chart.

**SECTION FIFTEEN**  
**NUMERICAL EXAMPLES OF OPTIMIZATION METHODS**

© J.W. Bandler 1984, 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



Solution of the nodal equations for a resistive network serves to illustrate several interrelated concepts.

Nodal Equations

$$\begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Let

$$\phi = \underset{\sim}{v} \triangleq \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}.$$

Equality Constraints

$$f_1 = 3v_1 - 2v_2 - 1 = 0$$

$$f_2 = -2v_1 + 5v_2 - 2v_3 = 0$$

$$f_3 = -2v_2 + 3v_3 = 0$$

Least Squares Objective (LSO)

$$\begin{aligned} U &= f_1^2 + f_2^2 + f_3^2 \\ &= (3v_1 - 2v_2 - 1)^2 + (-2v_1 + 5v_2 - 2v_3)^2 + (-2v_2 + 3v_3)^2 \end{aligned}$$

Gradient Vectors

$$\tilde{\nabla} f_1 = \begin{bmatrix} 3 \\ -2 \\ 0 \end{bmatrix}$$

$$\tilde{\nabla} f_2 = \begin{bmatrix} -2 \\ 5 \\ -2 \end{bmatrix}$$

$$\tilde{\nabla} f_3 = \begin{bmatrix} 0 \\ -2 \\ 3 \end{bmatrix}$$

Jacobian Matrix

$$\begin{aligned} \tilde{J}^T &\triangleq [\tilde{\nabla} f_1 \quad \tilde{\nabla} f_2 \quad \tilde{\nabla} f_3] \\ &= \begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \end{aligned}$$

Gradient of Least Squares Objective (LSO)

$$\tilde{\nabla} U = 2f_1 \tilde{\nabla} f_1 + 2f_2 \tilde{\nabla} f_2 + 2f_3 \tilde{\nabla} f_3$$

$$= 2[\tilde{\nabla} f_1 \quad \tilde{\nabla} f_2 \quad \tilde{\nabla} f_3] \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = 2\tilde{J}^T \tilde{f}$$

$$= 2 \begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

Taylor Series

$$f_1(\phi + \Delta\phi) = f_1(\phi) + \begin{bmatrix} 3 \\ -2 \\ 0 \end{bmatrix}^T \Delta\phi .$$

$$f_2(\phi + \Delta\phi) = f_2(\phi) + \begin{bmatrix} -2 \\ 5 \\ -2 \end{bmatrix}^T \Delta\phi .$$

$$f_3(\phi + \Delta\phi) = f_3(\phi) + \begin{bmatrix} 0 \\ -2 \\ 3 \end{bmatrix}^T \Delta\phi .$$

Hessian Matrix of LSO

$$\begin{aligned} \underline{H} &= \underline{\nabla}(\underline{\nabla}^T U) = 2[f_1 \underline{\nabla}(\underline{\nabla}^T f_1) + \underline{\nabla} f_1(\underline{\nabla}^T f_1) \\ &\quad + f_2 \underline{\nabla}(\underline{\nabla}^T f_2) + \underline{\nabla} f_2(\underline{\nabla}^T f_2) \\ &\quad + f_3 \underline{\nabla}(\underline{\nabla}^T f_3) + \underline{\nabla} f_3(\underline{\nabla}^T f_3)] \\ &= 2 \left\{ \begin{bmatrix} 3 \\ -2 \\ 0 \end{bmatrix} \begin{bmatrix} 3 & -2 & 0 \end{bmatrix} + \begin{bmatrix} -2 \\ 5 \\ -2 \end{bmatrix} \begin{bmatrix} -2 & 5 & -2 \end{bmatrix} + \begin{bmatrix} 0 \\ -2 \\ 3 \end{bmatrix} \begin{bmatrix} 0 & -2 & 3 \end{bmatrix} \right\} \end{aligned}$$

$$= 2 \left\{ \begin{bmatrix} 9 & -6 & 0 \\ -6 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 4 & -10 & 4 \\ -10 & 25 & -10 \\ 4 & -10 & 4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & -6 \\ 0 & -6 & 9 \end{bmatrix} \right\}$$

(notice that the rank of each matrix is one)

$$= 2 \begin{bmatrix} 13 & -16 & 4 \\ -16 & 33 & -16 \\ 4 & -16 & 13 \end{bmatrix}$$

### Taylor Series of LSO

$$U(\phi + \Delta\phi) = U(\phi) + 2 \left\{ \begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \right\}^T \Delta\phi + \Delta\phi^T \begin{bmatrix} 13 & -16 & 4 \\ -16 & 33 & -16 \\ 4 & -16 & 13 \end{bmatrix} \Delta\phi.$$

### Taylor Series of Gradient Vectors of LSO

$$\nabla U(\phi + \Delta\phi) = \nabla U(\phi) + \underline{H} \Delta\phi$$

$$= 2 \begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} + 2 \begin{bmatrix} 13 & -16 & 4 \\ -16 & 33 & -16 \\ 4 & -16 & 13 \end{bmatrix} \Delta\phi .$$

### Newton Method (Function Minimization)

$$\underline{H}\Delta\phi = -\nabla U$$

i.e.,



$$2 \begin{bmatrix} 13 & -16 & 4 \\ -16 & 33 & -16 \\ 4 & -16 & 13 \end{bmatrix} \Delta \phi = -2 \begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \begin{bmatrix} f_1(\phi) \\ f_2(\phi) \\ f_3(\phi) \end{bmatrix}$$

which is found in a finite number of steps since no higher-order terms have been neglected.

### Relationship of Hessian to Jacobian for Least Squares

Notice that  $\underline{H} = 2 \underline{J}^T \underline{J}$

$$= 2 \begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix}$$

$$= 2 \begin{bmatrix} 13 & -16 & 4 \\ -16 & 33 & -16 \\ 4 & -16 & 13 \end{bmatrix}$$

### Newton Method (Least Squares Approximation)

For a general rectangular  $\underline{J}$  (more functions than variables),

$$\underline{J}^T \underline{J} \Delta \phi = - \underline{J}^T \underline{f}$$

where we assume  $\underline{J}^T \underline{J}$  is nonsingular.

### Newton Method (Solution of Nonlinear Equations)

For a square nonsingular matrix  $\underline{J}$ , we premultiply both sides of the previous equation by  $(\underline{J}^T)^{-1}$  to give

$$\underline{J} \Delta \phi = -\underline{f}$$

$$\begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \begin{bmatrix} v_1 - v_1^0 \\ v_2 - v_2^0 \\ v_3 - v_3^0 \end{bmatrix} = \begin{bmatrix} -f_1(\tilde{y}^0) \\ -f_2(\tilde{y}^0) \\ -f_3(\tilde{y}^0) \end{bmatrix}$$

Let  $\tilde{y}^0 = \tilde{0}$

$$\begin{bmatrix} 3 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

which are the original nodal equations.

### Conclusions

The Newton method of function minimization yields the same results as the Newton method for solving nonlinear equations, which is identical to solving the nodal equations as given.

### Steepest Descent (Viewed as Approximate Newton Method)

Let

$$\underline{H} \Delta \phi = - \nabla U$$

where  $\underline{H}$  has been replaced by the (model) unit matrix.

### Step Size Determination

Let

$$\Delta \phi = - \alpha \nabla U$$

Optimum  $\alpha$  is usually found numerically by one-dimensional optimization. However, we will assume, for academic purposes, that  $\underline{H}$  is known so that an exact steepest descent iteration can be illustrated.

Taylor Series

For the objective function under consideration,

$$U(\phi + \Delta\phi) = U(\phi) + (\nabla U)^T \Delta\phi + 0.5(\Delta\phi)^T \underline{\underline{H}} \Delta\phi.$$

Hence,

$$U(\alpha) = U(\phi) - \alpha (\nabla U)^T \underline{\underline{V}} U + 0.5\alpha^2 (\nabla U)^T \underline{\underline{H}} \underline{\underline{V}} U.$$

A minimum w.r.t.  $\alpha$  is given by

$$\frac{dU}{d\alpha} = -(\nabla U)^T \underline{\underline{V}} U + \alpha (\nabla U)^T \underline{\underline{H}} \underline{\underline{V}} U = 0$$

so that

$$\alpha_{opt} = \frac{(\nabla U)^T \underline{\underline{V}} U}{(\nabla U)^T \underline{\underline{H}} \underline{\underline{V}} U}$$

Function

The function under consideration reduces to

$$U = 13\phi_1^2 + 33\phi_2^2 + 13\phi_3^2 - 32\phi_1\phi_2 - 32\phi_2\phi_3 + 8\phi_1\phi_3 - 6\phi_1 + 4\phi_2 + 1$$

Results for Steepest Descent and Exact Line Search

<u>Iteration</u>	$\phi$	$\nabla U$	U	$\alpha$
1	0.000000 0.000000 0.000000	- 6.000000 4.000000 0.000000	1.000000	.014739
2	.088435 - .058957 0.000000	- 1.814059 - 2.721088 2.594104	.616780	.021518
3	.127470 - .000404 - .055820	- 3.119392 1.680507 - .418623	.429309	.015059
4	.174444 - .025710 - .049516	- 1.037845 - 1.694586 .930866	.333461	.022607
5	.197906 .012598 - .070560	- 1.822063 .756412 - .654459	.279033	.015757
6	.226617 .000680 - .060248	- .611682 - 1.278974 .224754	.244994	.024203
7	.241422 .031635 - .065687	- 1.260847 .464395 - .788810	.220059	.016339
8	.262023 .024047 - .052799	- .379294 - 1.108077 - .046086	.200227	.025003
9	.271507 .051752 - .051646	- 1.010080 .380139 - .826835	.183052	.016520
10	.288193 .045473 - .037987	- .266013 - 1.005375 - .137256	.167784	.025193
.				
.				
.				
95	.517546 .279783 .184213	- .023143 .009370 - .023143	.000117	.016563

96	.517930	- .005143	.000108	.025236
	.279628	- .025406		
	.184596	- .005143		
97	.581060	- .021247	.000099	.016563
	.280269	.008603		
	.184726	- .021247		
98	.518411	- .004722	.000091	.025236
	.280126	- .023324		
	.185078	- .004722		
99	.518531	- .019506	.000083	.016563
	.280715	.007898		
	.185197	- .019506		

Results for QUSNTN Program

No. of iterations = 6

No. of function evaluations = 8

Solution: 0.523810

0.285714

0.190476

Objective:  $\approx 10^{-14}$

Gradients:  $\approx 10^{-9}$



**SECTION SIXTEEN**  
**ONE-DIMENSIONAL STRATEGIES**

© J.W. Bandler 1988

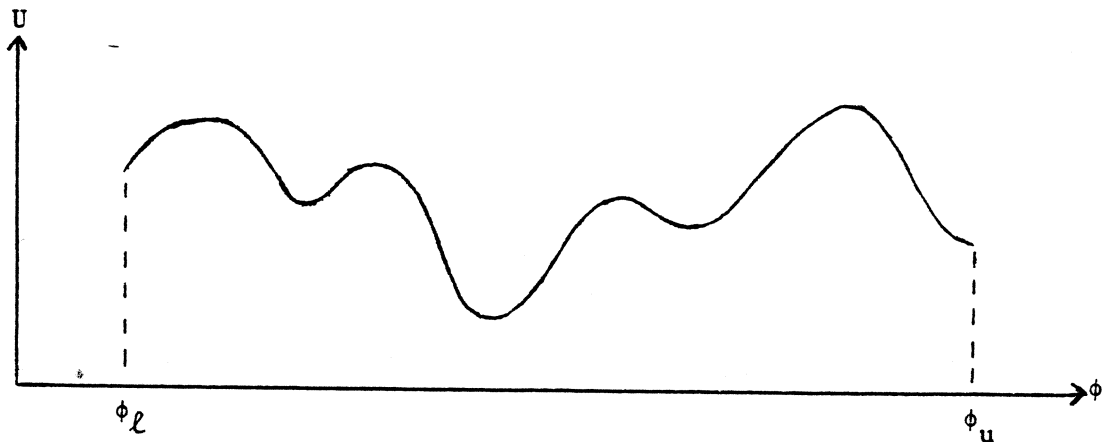
This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



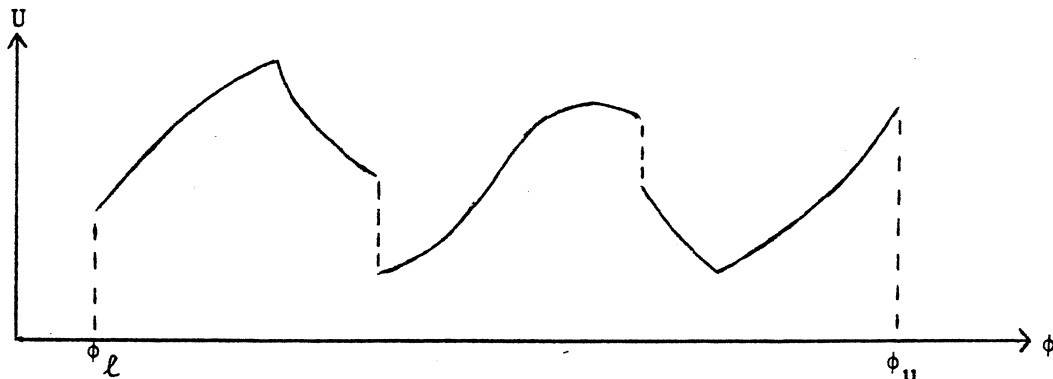


## ONE-DIMENSIONAL STRATEGIES

There are several reasons for investigating the optima of functions of one variable. The most obvious reason is that that is the problem we are given. Another is that the multidimensional optimization strategy we are using may call for one-dimensional techniques for searching along some feasible direction to find the minimum in that direction. A third possibility is that we are dealing with an approximation problem for which we require the extrema of the error or deviation between the specified function and the approximating function.



This is a relatively well-behaved function of one variable. The function is continuous and has continuous derivatives. The first derivative vanishes at the extrema (except at  $\phi_\ell$  and  $\phi_u$ ). The turning points could be found by the indirect method of finding the zeros of  $\frac{dU}{d\phi} = 0$ , assuming that  $U$  is easily differentiated, and the resulting equation easily solved.

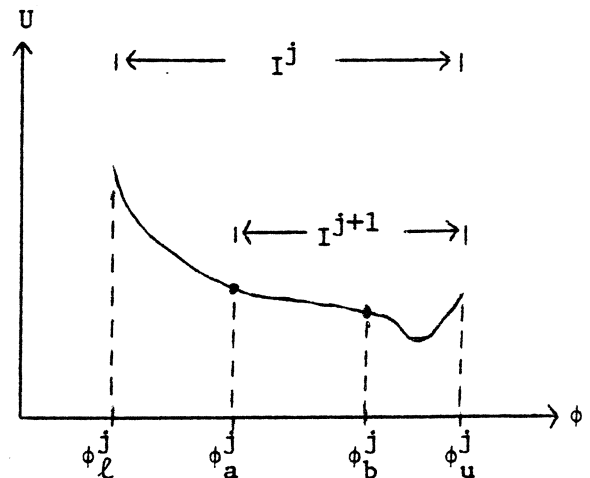
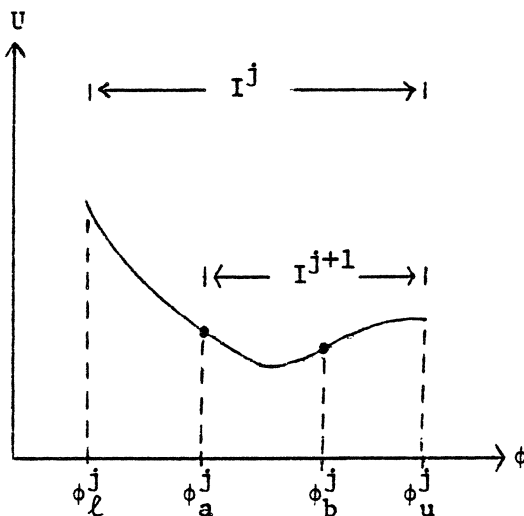


This is a badly-behaved function of one variable. The function is discontinuous, has discontinuous derivatives, and the first derivative vanishes only at one extremum. The extrema would have to be found by a direct method.

Note that although we might, in practice, be dealing with a function of  $\psi$  such as  $e(\phi, \psi)$  for a particular value of  $\psi$ , the discussion here will, without loss of generality, consider a function  $U$  of a single variable  $\phi$ .

Powerful methods are available for functions that are known to be unimodal on a particular interval. These will be discussed first. Then we will consider methods of finding such intervals.

The methods can be divided into two classes: 1) the minimax direct elimination methods - minimax, because they minimize the maximum interval which could contain the minimum, and 2) the approximation or interpolation methods. The latter are generally effective on smooth functions, but the former can be applied to arbitrary unimodal functions.



Suppose that at the start of the  $j$ th iteration of a search method for a minimum we have a unimodal interval  $[\phi_\ell^j, \phi_u^j]$ . The interval of uncertainty  $I^j$  as to where the minimum is located is therefore given by

$$I^j \triangleq \phi_u^j - \phi_\ell^j.$$

In order to reduce the interval of uncertainty using function values only, the function must be evaluated at two interior points, say  $\phi_a^j$  and  $\phi_b^j$ . (Evaluation of the function at only one interior point is not enough - the minimum could lie on either side.) We have

$$\phi_\ell^j < \phi_a^j < \phi_b^j < \phi_u^j.$$

Let  $U_a^j \triangleq U(\phi_a^j)$  and  $U_b^j \triangleq U(\phi_b^j)$ .

If  $U_a^j > U_b^j$  the minimum lies in  $[\phi_a^j, \phi_u^j]$  and  $I^{j+1} = \phi_u^j - \phi_a^j$ .

If  $U_a^j < U_b^j$  the minimum lies in  $[\phi_\ell^j, \phi_b^j]$  and  $I^{j+1} = \phi_b^j - \phi_\ell^j$ .

The difference in the methods to be discussed is in how these interior points are located.

#### Search by Golden Section

In the absence of any relevant a priori knowledge, it would first of all be reasonable to select  $\phi_a^j$  and  $\phi_b^j$  such that, whatever the outcome of comparing  $U_a^j$  with  $U_b^j$

$$I^{j+1} = \phi_u^j - \phi_a^j = \phi_b^j - \phi_\ell^j$$

which is achieved if  $\phi_a^j$  and  $\phi_b^j$  are located symmetrically in  $[\phi_\ell^j, \phi_u^j]$ .

Secondly, in an effort to make  $I^{j+1}$  as small as possible we would arrange

$\phi_a^j$  and  $\phi_b^j$  so that

$$\phi_u^j - \phi_b^j = \phi_a^j - \phi_\ell^j > \phi_b^j - \phi_a^j.$$

Suppose  $U_a^j > U_b^j$ . Then we would set

$$\phi_\ell^{j+1} = \phi_a^j, \quad \phi_a^{j+1} = \phi_b^j, \quad \phi_u^{j+1} = \phi_u^j.$$

Following the above procedure we would find that

$$I^{j+2} = \phi_u^{j+1} - \phi_a^{j+1} = \phi_u^j - \phi_b^j$$

since  $\phi_b^{j+1}$  is to be placed so that

$$\phi_\ell^{j+1} < \phi_a^{j+1} < \phi_b^{j+1} < \phi_u^{j+1}.$$

But

$$\phi_u^j - \phi_b^j = \phi_a^j - \phi_\ell^j$$

so that

$$I^j = I^{j+1} + I^{j+2}.$$

Our next idea would probably be to try to arrange that the interval of uncertainty containing the minimum is reduced by a constant factor with each iteration. Thus,

$$\dots = \frac{I^j}{I^{j+1}} = \frac{I^{j+1}}{I^{j+2}} = \dots \triangleq \tau$$

where  $\tau$  is a constant. Solving these equations we obtain

$$\tau^2 = \tau + 1.$$

The solution of interest is

$$\tau = \frac{1}{2}(1 + \sqrt{5}) = 1.6180\dots$$

Note that earlier we called for

$$I^{j+2} > I^{j+1} - I^{j+2}$$

i.e.,

$$\frac{I^{j+1}}{I^{j+2}} < 2 \quad \text{which is satisfied since } \tau < 2.$$

At the  $j$ th iteration of Golden Section search we have

$$\left. \begin{aligned} \phi_a^j &= \frac{1}{\tau} I^j + \phi_\ell^j \\ \phi_b^j &= \frac{1}{\tau} I^j + \phi_\ell^j \end{aligned} \right\} \quad j = 1, 2, 3, \dots$$

An example involving four function evaluations is shown. Observe that each iteration except the first actually requires only one function evaluation due to symmetry.

$$\text{If } U_a^j > U_b^j \text{ then } \phi_\ell^{j+1} = \phi_a^j, \quad \phi_a^{j+1} = \phi_b^j, \quad \phi_u^{j+1} = \phi_u^j, \quad U_a^{j+1} = U_b^j.$$

$$\text{If } U_a^j < U_b^j \text{ then } \phi_\ell^{j+1} = \phi_\ell^j, \quad \phi_b^{j+1} = \phi_a^j, \quad \phi_u^{j+1} = \phi_b^j, \quad U_b^{j+1} = U_a^j.$$

In either case only one additional point is required with the function value at that point.

After  $n$  function evaluations the reduction ratio of intervals of uncertainty is

$$\frac{I^1}{I^n} = \tau^{n-1}.$$

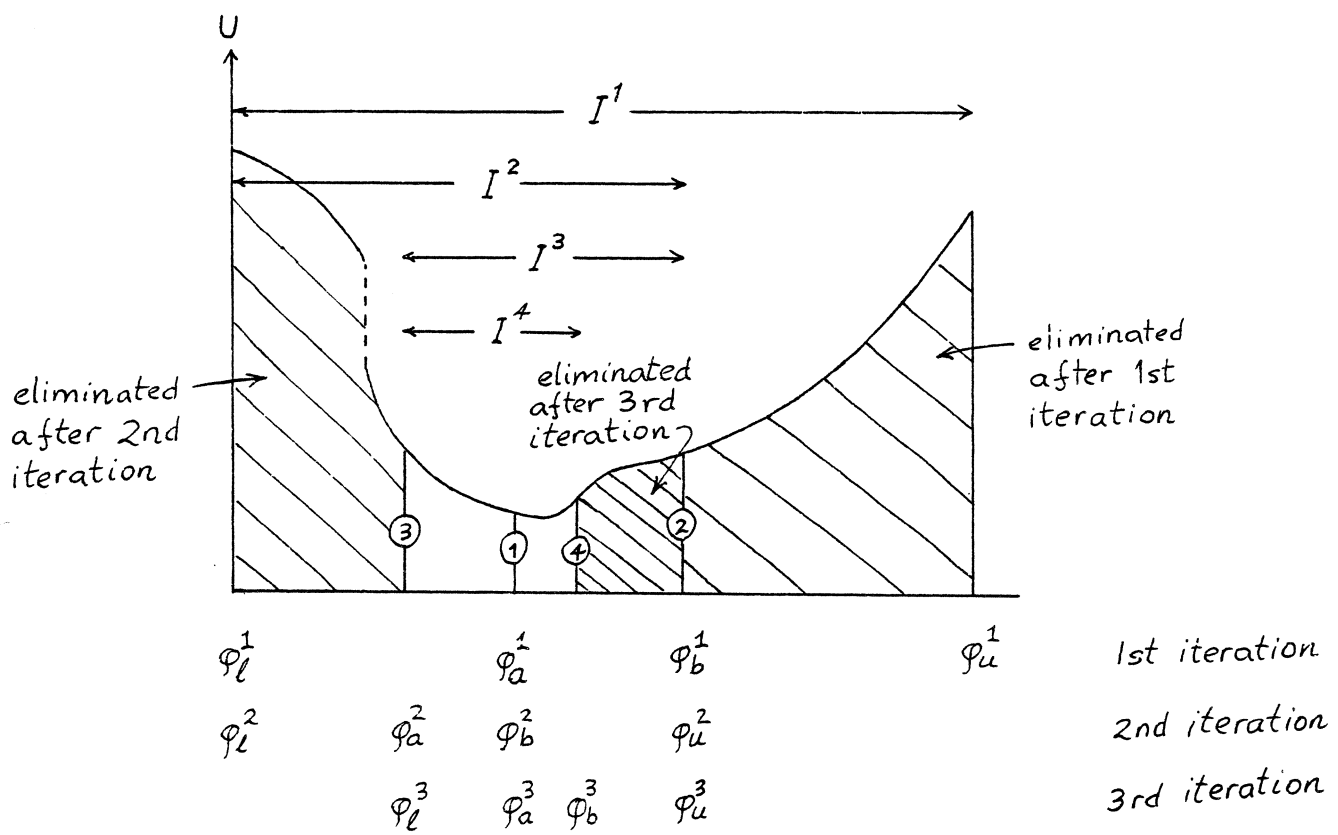
For a desired accuracy of  $\sigma$  we should choose  $n$  so that

$$\tau^{n-2} < \frac{\phi_u^1 - \phi_\ell^1}{\sigma} \leq \tau^{n-1}.$$

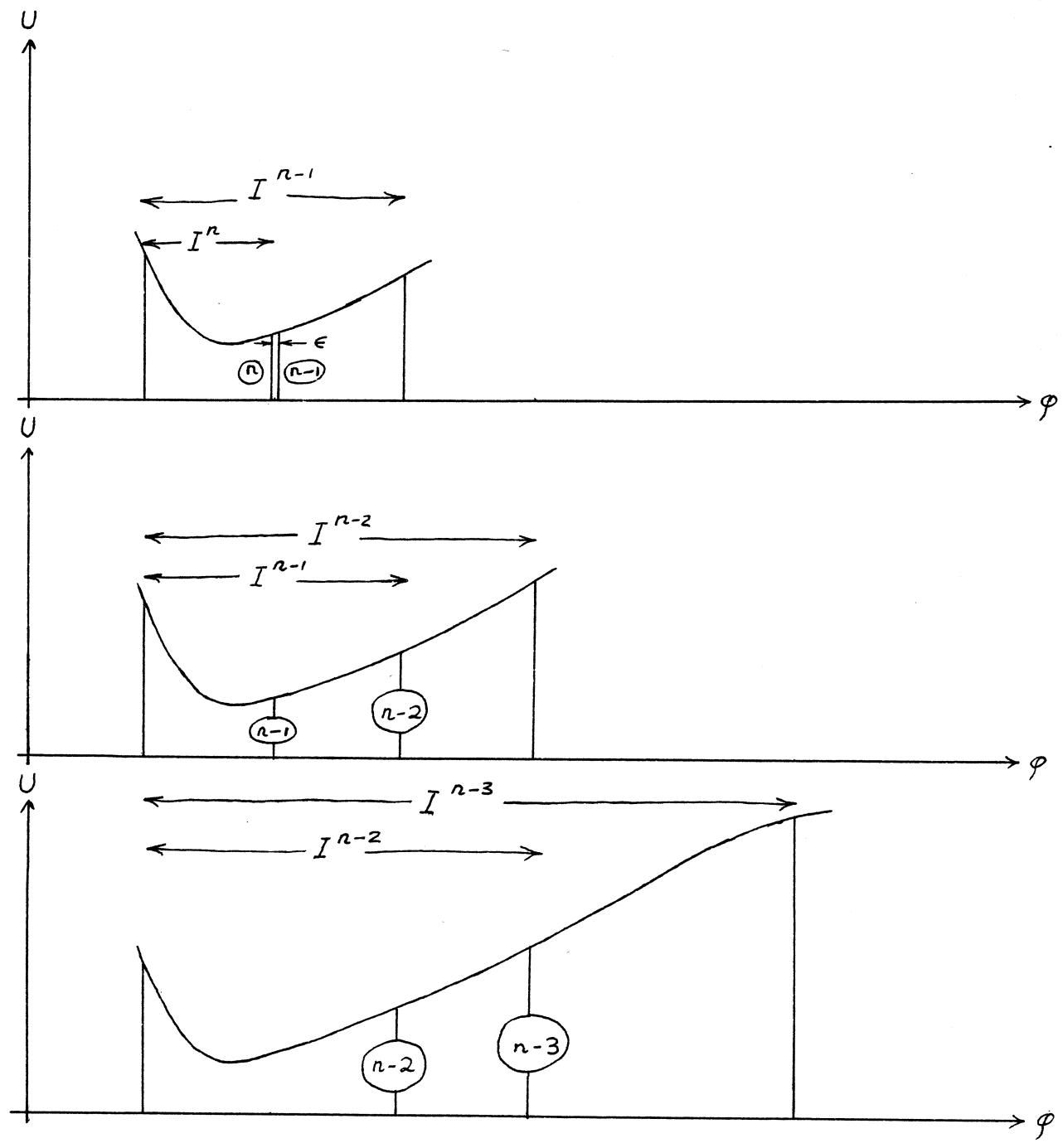
In the example the initial interval has been reduced by a factor of about 4.2. Eleven evaluations would have reduced the interval by about 121.

### Fibonacci Search

Suppose we have already made  $n-1$  function evaluations, the  $(n-1)$ th point being an interior point of an interval of length  $I^{n-1}$ . Obviously,  $I^n$  will be least if the  $(n-1)$ th point was in the middle of the interval



A Golden Section search scheme involving three iterations (four function evaluations) on a unimodal function of one variable.



so that the  $n$ th (and final) point could be placed as close as possible to it. Alternatively we might, in accordance with our previous discussion prefer to think of the  $(n-1)$ th and  $n$ th points to be symmetrically placed with a separation  $\epsilon$  and let  $\epsilon \rightarrow 0$ . In practice, if only the minimum value obtained with its location is desired we could simply omit the  $n$ th function evaluation.

Suppose we have already made  $n-2$  function evaluations, the  $(n-2)$ th point being an interior point of an interval of length  $I^{n-2}$ . Assuming a particular value for  $I^{n-1}$  we could ask what the value is of  $I^{n-2}$  consistent with symmetrical placement of the  $(n-1)$ th point (for reasons already discussed) and optimal placement of the  $n$ th point. The answer is that

$$I^{n-2} = I^{n-1} + I^n$$

(assuming  $\epsilon = 0$ ). Continuing in this vein we see that

$$I^{n-3} = I^{n-2} + I^{n-1}$$

Indeed, as for Golden Section search

$$I^j = I^{j+1} + I^{j+2} \quad \begin{array}{l} j = 1, 2, \dots, n-2 \\ n > 2 \end{array}$$

Define the reduction ratio after  $n$  function evaluations as  $F_n$ , i.e.,

$$F_n \triangleq \frac{I^1}{I^n}$$

Let  $I^n = 1$ , for convenience. Then  $I^{n-1} = 2$ . Using the above relationship

$$\begin{array}{l} I^n = 1 \\ I^{n-1} = 2 \\ I^{n-2} = 3 \\ I^{n-3} = 5 \\ \vdots \\ \vdots \end{array}$$



$$I^{j+1} = F_{n-j}$$

$$I^j = F_{n+1-j}$$

$$\vdots$$

$$\vdots$$

$$I^2 = F_{n-1}$$

$$I^1 = F_n$$

We recognise this sequence of numbers as being in the Fibonacci sequence of numbers defined by

$$F_0 = F_1 = 1$$

$$F_i = F_{i-1} + F_{i-2} \quad i = 2, 3, \dots$$

the first six terms, for example, being 1, 1, 2, 3, 5, 8. For example,  $n = 4$  gives  $I^1 = 5 = F_4$ . Note that  $I^0$ , the interval of uncertainty after 0 evaluations, is equal to  $I^1 = 1$ , as expected.

The solution to the recurrence relationship can be shown to be

$$F_n = \frac{1}{\sqrt{5}} \left\{ \left( \frac{1+\sqrt{5}}{2} \right)^{n+1} - \left( \frac{1-\sqrt{5}}{2} \right)^{n+1} \right\}.$$

At the  $j$ th iteration of Fibonacci search using  $n$  function evaluations ( $n \geq 2$ ) we have

$$\left. \begin{aligned} \phi_a^j &= \frac{F_{n-1-j}}{F_{n+1-j}} I^j + \phi_\ell^j \\ \phi_b^j &= \frac{F_{n-j}}{F_{n+1-j}} I^j + \phi_\ell^j \end{aligned} \right\} \quad j = 1, 2, \dots, n-1$$

An example involving four function evaluations is shown. As with Golden Section search each iteration except the first actually requires only one function evaluation. It may easily be verified that the same relationship

holds as for Golden Section search whether  $U_a^j > U_b^j$  or  $U_a^j < U_b^j$ .

The interval of uncertainty after  $j$  iterations is

$$I^{j+1} = \phi_u^j - \phi_a^j = \phi_b^j - \phi_\ell^j$$

reducing the interval  $I^j$  by a factor

$$\frac{I^j}{I^{j+1}} = \frac{F_{n+1-j}}{F_{n-j}}.$$

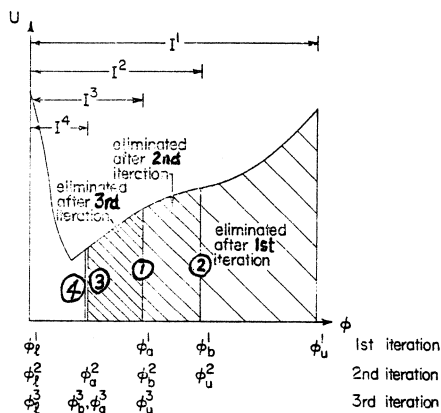
As a check, we find that after  $n-1$  iterations, assuming infinite resolution, the reduction ratio is

$$\frac{I^1}{I^n} = \frac{F_n}{F_{n-1}} \cdot \frac{F_{n-1}}{F_{n-2}} \cdot \dots \cdot \frac{F_2}{F_1} = F_n.$$

For a desired accuracy of  $\sigma$  we should choose  $n$  so that

$$F_{n-1} < \frac{\phi_u^1 - \phi_\ell^1}{\sigma} < F_n.$$

In the example the initial interval has been reduced by a factor of 5 after 4 function evaluations. Eleven function evaluations would have reduced the interval by a factor of 144.



A Fibonacci search scheme involving three iterations  
(four function evaluations) on a unimodal function of  
one variable.

$$\text{1st evaluation} \quad \phi_a^1 = \frac{F_2}{F_4} I^1 + \phi_\ell^1 = \frac{2}{5} I^1 + \phi_\ell^1$$

$$\text{2nd evaluation} \quad \phi_b^1 = \frac{F_3}{F_4} I^1 + \phi_\ell^1 = \frac{3}{5} I^1 + \phi_\ell^1$$

$$U_a^1 < U_b^1 \text{ so} \quad \phi_\ell^2 = \phi_\ell^1, \phi_b^2 = \phi_a^1, \phi_u^2 = \phi_b^1, I^2 = \frac{3}{5} I^1$$

$$\text{3rd evaluation} \quad \phi_a^2 = \frac{F_1}{F_3} I^2 + \phi_\ell^2 = \frac{1}{3} \cdot \frac{3}{5} \cdot I^1 + \phi_\ell^1 = \frac{1}{5} I^1 + \phi_\ell^1$$

$$\phi_b^2 = \frac{F_2}{F_3} I^2 + \phi_\ell^2 = \frac{2}{3} \cdot \frac{3}{5} I^1 + \phi_\ell^1 = \phi_a^1$$

$$U_a^2 < U_b^2 \text{ so} \quad \phi_\ell^3 = \phi_\ell^2, \phi_b^3 = \phi_a^2, \phi_u^3 = \phi_b^2, I^3 = \frac{2}{5} I^1$$

$$\text{4th evaluation} \quad \left\{ \begin{array}{l} \phi_a^3 = \frac{F_0}{F_2} I^3 + \phi_\ell^3 = \frac{1}{2} \cdot \frac{2}{5} I^1 + \phi_\ell^1 = \frac{1}{5} I^1 + \phi_\ell^1 = \phi_a^2 \\ \phi_b^3 = \frac{F_1}{F_2} I^3 + \phi_\ell^3 = \frac{1}{2} \cdot \frac{2}{5} I^1 + \phi_\ell^1 = \frac{1}{5} I^1 + \phi_\ell^1 = \phi_a^2 \end{array} \right.$$

(may be omitted)

Discussion of Fibonacci and Golden Section Methods

For Fibonacci search as  $n \rightarrow \infty$

$$\frac{I^1}{I^n} = F_n \approx \frac{1}{\sqrt{5}} \left( \frac{1 + \sqrt{5}}{2} \right)^{n+1} = \frac{\tau^{n+1}}{\sqrt{5}}$$

and

$$\lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}} = \tau.$$

The ratio of effectiveness of the Fibonacci search as compared with Golden Section is, therefore, given by

$$\lim_{n \rightarrow \infty} \frac{F_n}{\tau^{n-1}} = \frac{\tau^2}{\sqrt{5}} = 1.1708.$$

Consider the reduction ratio per iteration:

$$\frac{I^j}{I^{j+1}} = \lim_{n \rightarrow \infty} \frac{F_{n+1-j}}{F_{n-j}} = \tau.$$

For a very large number of function evaluations the reduction ratio per iteration is practically the same for both methods. It is easily shown, for example, that  $\phi_a^1$  and  $\phi_b^1$  is practically the same for both methods with  $n$  large. Golden Section search ultimately provides an interval of uncertainty only some 17% greater than Fibonacci search.

Thus, Golden Section search is frequently preferred because the number of function evaluations need not be fixed in advance, it is simple to implement and at most one additional function evaluation is required to achieve a specified accuracy, since

$$\tau^n > F_n \quad n = 0, 1, 2, \dots$$

where  $\tau^n$  is the reduction ratio for  $n+1$  evaluations by Golden Section search and  $F_n$  is the reduction ratio for  $n$  evaluations by Fibonacci search.

Suppose an initial unimodal interval is not available. It can be found by the following procedure. Select a convenient increment in the direction of decreasing  $U$  and evaluate  $U(\phi^i)$  for

$$\phi^i = \phi^0 + \sum_{j=1}^i \tau^{j-1} \Delta\phi \quad i = 1, 2, \dots$$

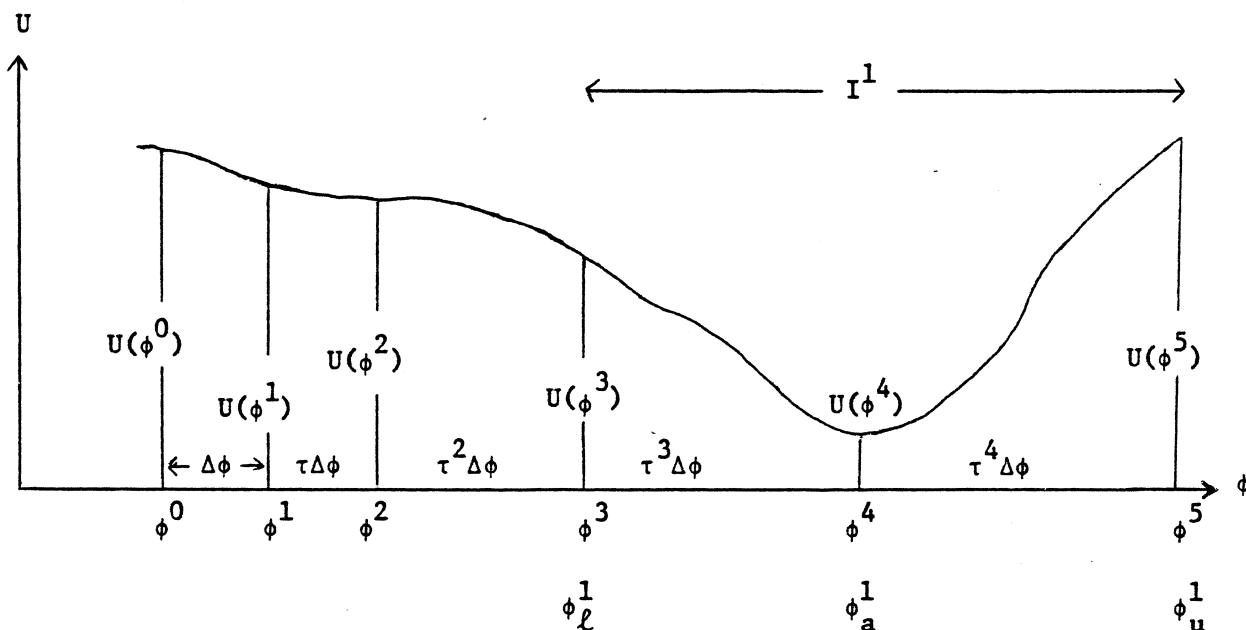
Stop when

$$U(\phi^i) \geq U(\phi^{i-1})$$

for some value of  $i$ . We can now define a unimodal interval and proceed directly with Golden Section search.

$$\text{If } \Delta\phi > 0 \quad \phi_{\ell}^1 = \phi^{i-2}, \quad \phi_a^1 = \phi^{i-1}, \quad \phi_u^1 = \phi^i.$$

$$\text{If } \Delta\phi < 0 \quad \phi_{\ell}^1 = \phi^i, \quad \phi_b^1 = \phi^{i-1}, \quad \phi_u^1 = \phi^{i-2}$$



In the example:

$$\begin{aligned}\phi_a^1 &= \phi_\ell^1 + \frac{\tau^3 \Delta\phi}{\tau^3 \Delta\phi + \tau^4 \Delta\phi} (\phi_u^1 - \phi_\ell^1) \\ &= \phi_\ell^1 + \frac{1}{1 + \tau} (\phi_u^1 - \phi_\ell^1) \\ &= \frac{1}{2} \phi_u^1 + \phi_\ell^1.\end{aligned}$$

Thus, we have the desired unimodal interval with an interior point and we are ready to continue with search by Golden Section.

#### The Method of Davies, Swann and Campey

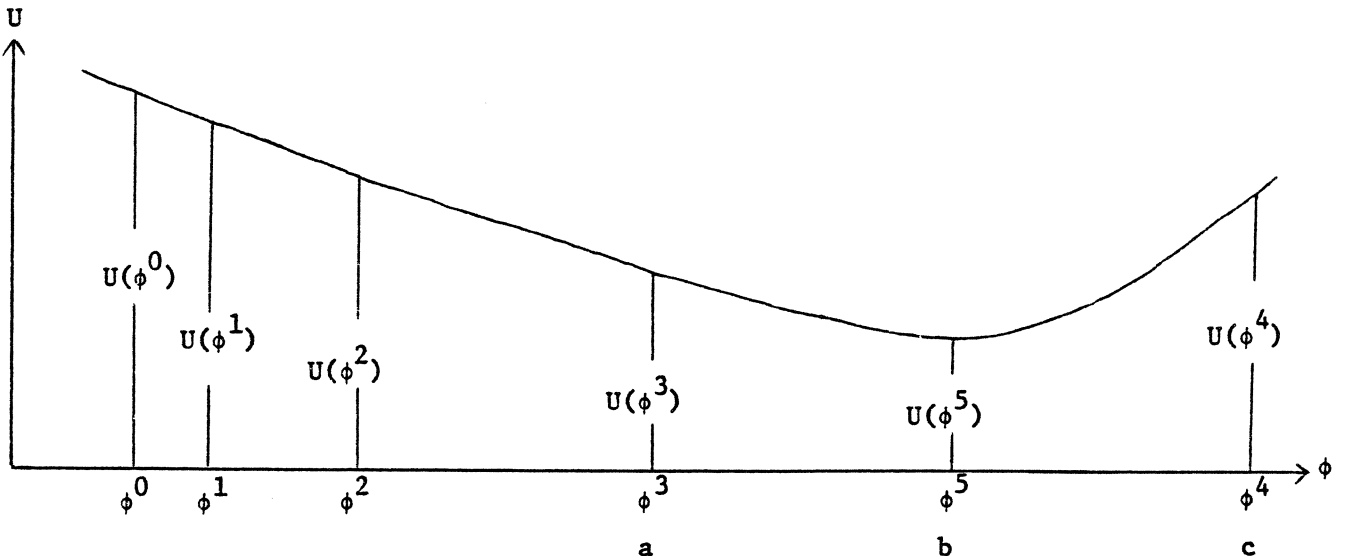
This method does not initially require bounds on the minimum. We select a convenient increment  $\Delta\phi$  in the direction of decreasing  $U$  and evaluate  $U(\phi^i)$  for

$$\phi^i = \phi^0 + \sum_{j=1}^i 2^{j-1} \Delta\phi \quad i = 1, 2, \dots$$

When

$$U(\phi^i) > U(\phi^{i-1})$$

we evaluate  $U$  at  $\phi^{i+1} = \phi^{i-1} + (\phi^{i-1} - \phi^{i-2})$ .



We now have 4 points uniformly spaced along the  $\phi$  axis, namely

$$\phi^{i-2}, \phi^{i-1}, \phi^{i+1}, \phi^i.$$

$$\text{If } \begin{cases} \Delta\phi > 0 \text{ and } \begin{cases} U(\phi^{i+1}) < U(\phi^{i-1}) \\ U(\phi^{i+1}) > U(\phi^{i-1}) \end{cases} \\ \Delta\phi < 0 \text{ and } \begin{cases} U(\phi^{i+1}) < U(\phi^{i-1}) \\ U(\phi^{i+1}) > U(\phi^{i-1}) \end{cases} \end{cases} \begin{cases} a = \phi^{i-1}, b = \phi^{i+1}, c = \phi^i \\ a = \phi^{i-2}, b = \phi^{i-1}, c = \phi^{i+1} \\ a = \phi^i, b = \phi^{i+1}, c = \phi^{i-1} \\ a = \phi^{i+1}, b = \phi^{i-1}, c = \phi^{i-2} \end{cases}$$

Now it is easily shown that the minimum of a quadratic fitted at  $a$ ,  $b$  and  $c$  is at

$$d = b + \frac{(b - a)(U_a - U_c)}{2(U_a - 2U_b + U_c)}.$$

Evaluation of  $U$  at  $d$  resulting in the estimate of the minimum completes one stage of the method. A new stage with reduced  $\Delta\phi$  can then be started at  $b$  or  $d$  whichever corresponds to the smaller value.

#### Quadratic Interpolation Method

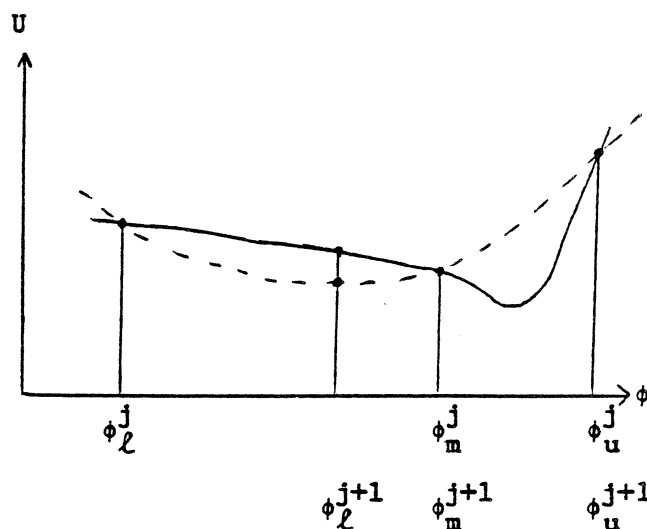
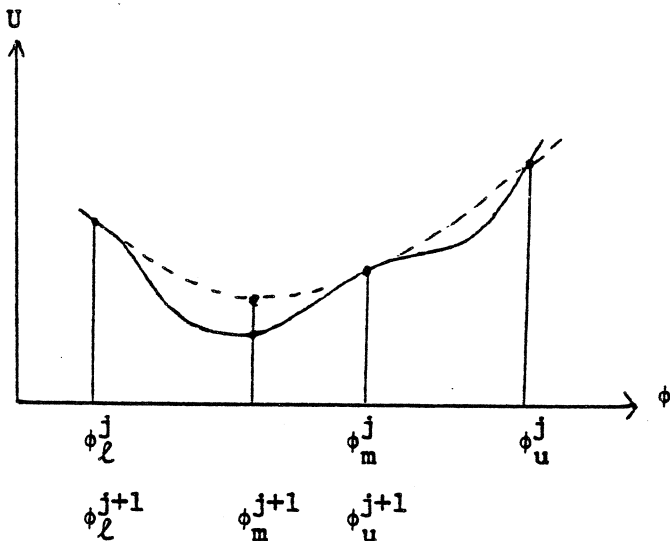
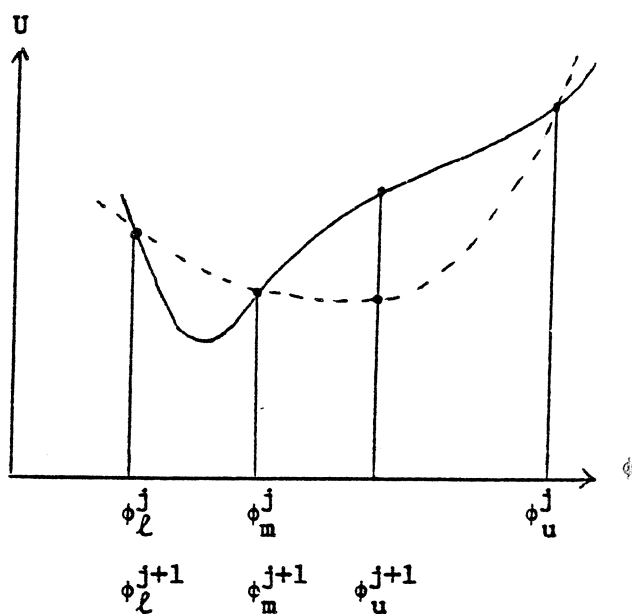
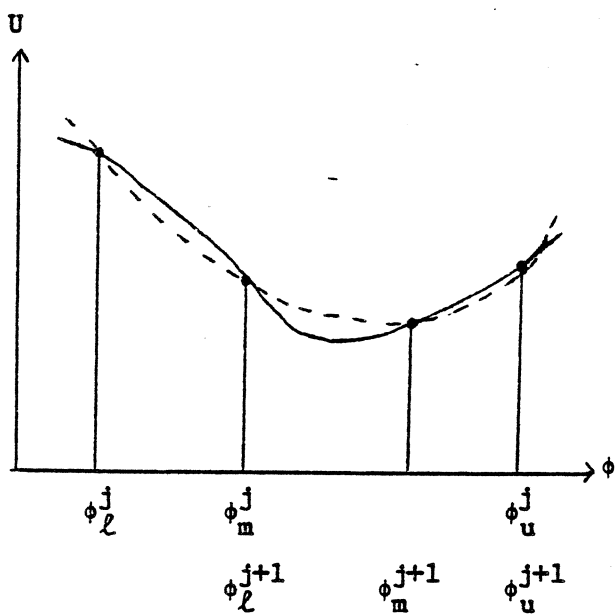
At the  $j$ th iteration of the typical method involving quadratic interpolation we have a unimodal function over  $[\phi_\ell^j, \phi_u^j]$  with an interior point  $\phi_m^j$ . Let  $a = \phi_\ell^j$ ,  $b = \phi_m^j$  and  $c = \phi_u^j$ . Then the minimum of the quadratic through  $a$ ,  $b$  and  $c$  is at

$$d = \frac{1}{2} \frac{(b^2 - c^2)U_a + (c^2 - a^2)U_b + (a^2 - b^2)U_c}{(b - c)U_a + (c - a)U_b + (a - b)U_c}.$$

Then  $\phi_\ell^{j+1}$ ,  $\phi_m^{j+1}$  and  $\phi_u^{j+1}$  are obtained as follows:

$$\text{If } \begin{cases} b > d \text{ and } \begin{cases} U_b > U_d & \phi_\ell^{j+1} = a, \phi_m^{j+1} = d, \phi_u^{j+1} = b \\ U_b < U_d & \phi_\ell^{j+1} = d, \phi_m^{j+1} = b, \phi_u^{j+1} = c \end{cases} \\ b < d \text{ and } \begin{cases} U_b > U_d & \phi_\ell^{j+1} = b, \phi_m^{j+1} = d, \phi_u^{j+1} = c \\ U_b < U_d & \phi_\ell^{j+1} = a, \phi_m^{j+1} = b, \phi_u^{j+1} = d \end{cases} \end{cases}$$

The procedure may be repeated for greater accuracy, convergence being guaranteed.





Powell's version of this algorithm does not initially require bounds on the minimum. Having selected three suitable initial points, an attempt to find the minimum through these points is made. For badly behaved functions or points far from the minimum the extrapolated point could be far from the minimum or could turn out to be an estimate of a maximum. Checks for such possibilities should be made. It is felt that a more reasonable scheme would be a combination of the method of Davies, Swann and Campey with quadratic interpolation such that quadratic interpolation is used only when the minimum has been bounded.

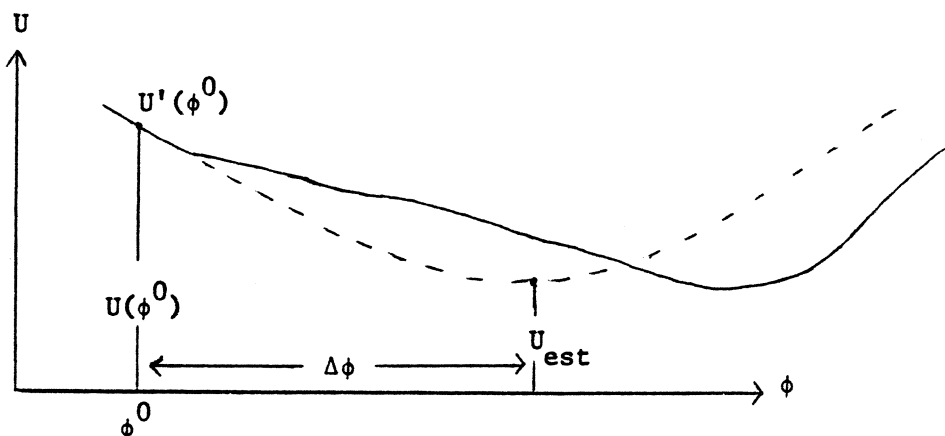
### Cubic Interpolation Method

The method to be presented could be found useful in one-dimensional searches involved in the Davidon, Fletcher-Powell or Fletcher-Reeves methods.

First we assume that  $\phi^0$ ,  $U(\phi^0)$  and  $U'(\phi^0)$  are available. We also have an estimate of the minimum  $U_{\text{est}} < U(\phi^0)$ . Then

$$\Delta\phi = \frac{-2(U(\phi^0) - U_{\text{est}})}{U'(\phi^0)}$$

would give the increment in  $\phi$  necessary to reach  $U_{\text{est}}$ .



To limit the size of the step we could have:

If  $|\Delta\phi| > \delta$  then  $\Delta\phi = \delta s$

where  $s = -U'(\phi^0)/|U'(\phi^0)|$  and  $\delta > 0$ .

This feature may be necessary when  $\phi^0$  is already close to the minimum in which case  $U'(\phi^0) \approx 0$ .

This procedure may be repeated from  $\phi^1 = \phi^0 + \Delta\phi$  and continued until for some  $i$   $U'(\phi^i)/U'(\phi^{i-1}) < 0$ . In this case the minimum has been bounded.

If  $U'(\phi^i) > 0$

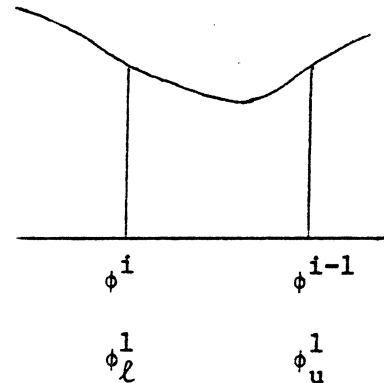
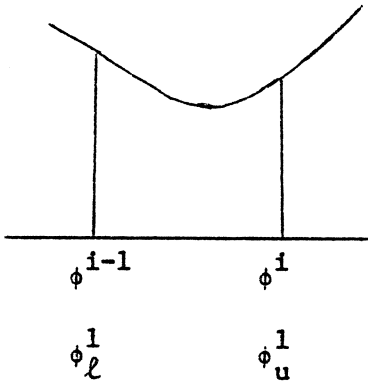
$\phi_\ell^1 = \phi^{i-1}$

$\phi_u^1 = \phi^i$

If  $U'(\phi^i) < 0$

$\phi_\ell^1 = \phi^i$

$\phi_u^1 = \phi^{i-1}$



Alternatively we may prefer to evaluate  $U(\phi^i)$  for

$\phi^i = \phi^0 + 2^{i-1} \Delta\phi$   $i = 1, 2, \dots$

until

$U(\phi^i) \geq U(\phi^{i-1})$ .

If  $\left\{ \begin{array}{l} \Delta\phi > 0 \text{ and } \begin{cases} U'(\phi^{i-1}) > 0 \\ U'(\phi^{i-1}) < 0 \end{cases} \\ \Delta\phi < 0 \text{ and } \begin{cases} U'(\phi^{i-1}) > 0 \\ U'(\phi^{i-1}) < 0 \end{cases} \end{array} \right. \begin{array}{ll} \phi_\ell^1 = \phi^{i-2} & \phi_u^1 = \phi^{i-1} \\ \phi_\ell^1 = \phi^{i-1} & \phi_u^1 = \phi^i \\ \phi_\ell^1 = \phi^i & \phi_u^1 = \phi^{i-1} \\ \phi_\ell^1 = \phi^{i-1} & \phi_u^1 = \phi^{i-2} \end{array}$

Letting  $a = \phi_{\ell}^1$  and  $b = \phi_u^1$  cubic interpolation between  $a$  and  $b$  predicts a minimum at

$$c = b - \frac{(b - a)(U'(b) + x - y)}{U'(b) - U'(a) + 2x}$$

where

$$y = U'(a) + U'(b) + 3 \frac{U(a) - U(b)}{b - a}$$

and

$$x = (y^2 - U'(a) U'(b))^{\frac{1}{2}}$$

If  $U(a) < U(c)$  or  $U(b) < U(c)$  a further interpolation may be required over  $[a, c]$  if  $U'(c) > 0$  or over  $[c, b]$  if  $U'(c) < 0$ . The minimum of a quadratic can be found in a single application of the interpolation formula.



**SECTION SEVENTEEN**

**DIRECT SEARCH**

© J.W. Bandler 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



## DIRECT SEARCH

Methods which do not rely explicitly on evaluation or estimation of partial derivatives of the objective function at any point are usually called direct search methods. Broadly speaking, they rely on the sequential examination of trial solutions in which each solution is compared with the best obtained up to that time, with a strategy generally based on past experience for deciding where the next trial solution should be located.

Falling into the category of direct search are: random search, when points within a region are selected and investigated at random; one-at-a-time search, when one coordinate direction at a time is investigated; pattern search and methods like it which attempt to align a direction of search along a valley; some quadratically convergent methods; and simplex methods, not to be confused with linear programming. Multidimensional extensions of Fibonacci search have been reported. Elimination methods are not as successful as some of the climbing methods to be discussed.

### One-at-a-Time Search

In this method first one parameter is allowed to vary, generally until no further improvement is obtained, and then another one, and so on.

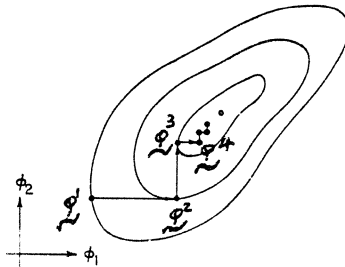
Thus, for a typical iteration

$$\underline{\phi}^{j+1} = \underline{\phi}^j + \alpha^j \underline{s}^j$$

where

$$\underline{s}^j = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -\frac{\partial U^j}{\partial \phi_1} / \left| \frac{\partial U^j}{\partial \phi_1} \right| \\ \vdots \\ 0 \end{bmatrix}$$

is the direction of improvement (decreasing  $U$ ) from  $\underline{\phi}^j$  along the  $\phi_1$  coordinate, and  $\alpha^j$  is a positive scale factor.  $\underline{s}^j$  is usually obtained by trial and error and  $\alpha^j$  obtained from a suitable one-dimensional minimization method to find the minimum in the  $\underline{s}^j$  direction. In practice, simple methods using only function values are employed.



Minimization by a one-at-a-time method.

As the figure shows progress will be slow on narrow valleys which are not oriented in the direction of any coordinate axis.

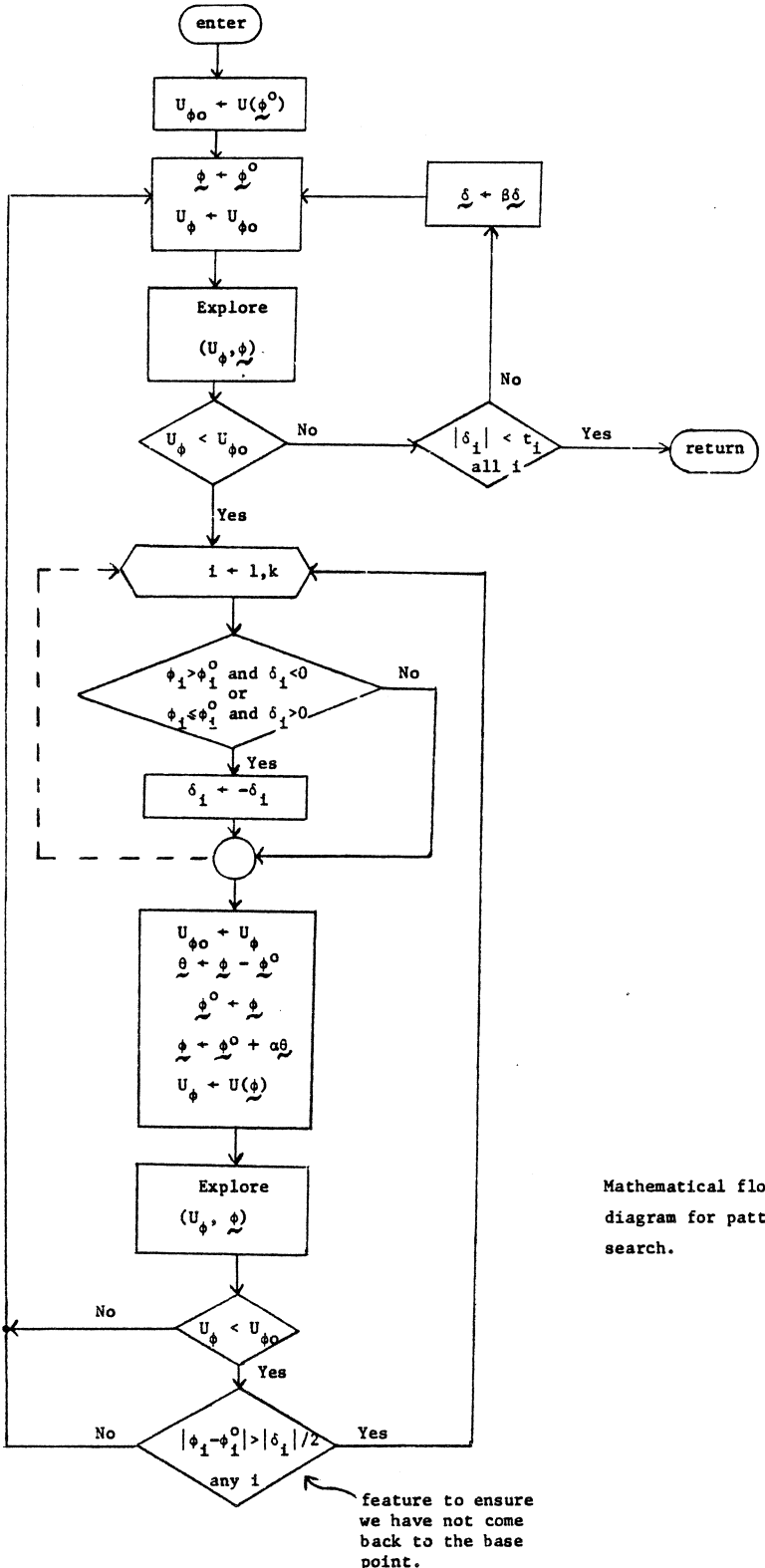


### Pattern Search

The pattern search strategy presented by Hooke and Jeeves is able to follow along fairly narrow valleys because it attempts to align a search direction along the valley.

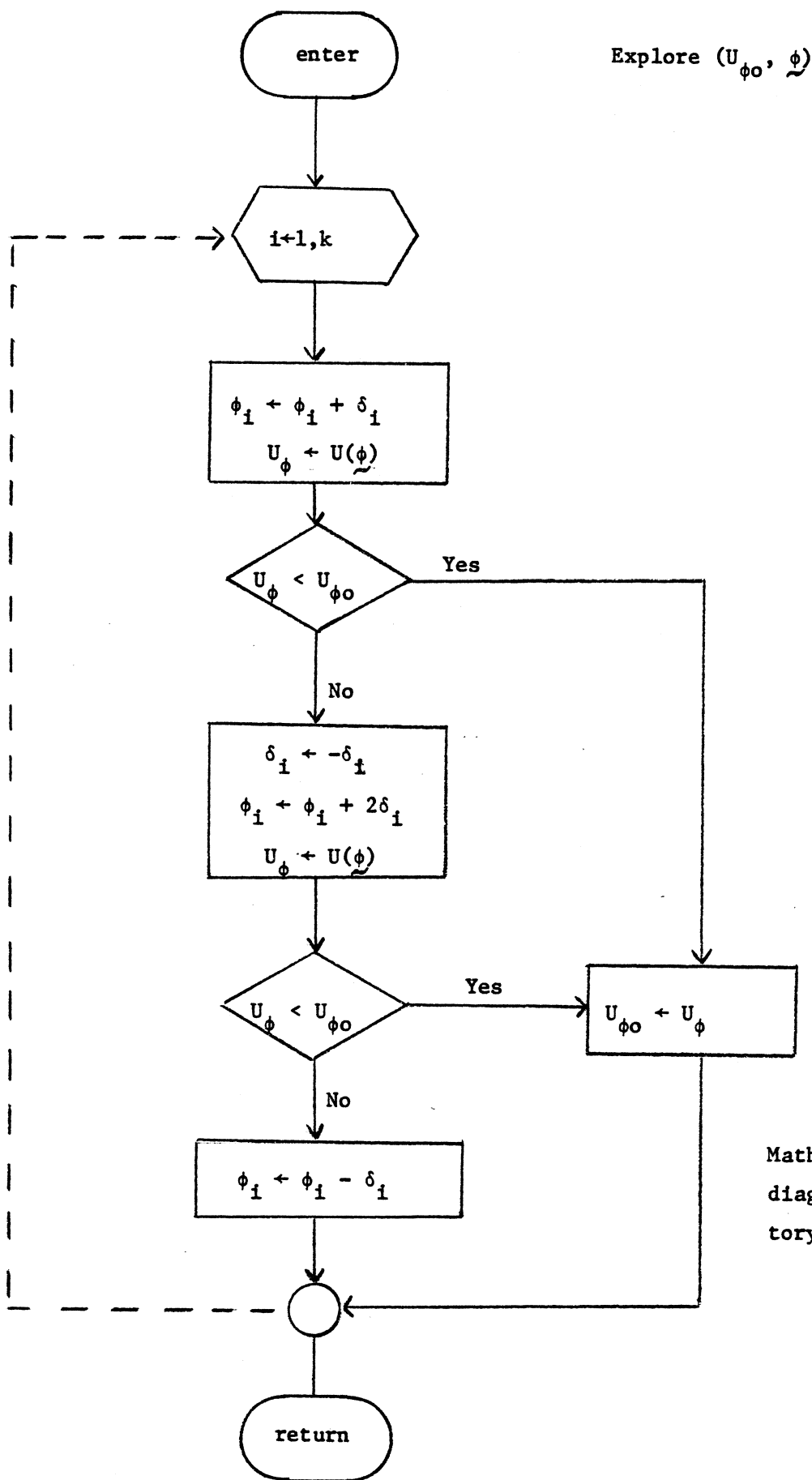
A flow diagram of the method is given. The variables are defined as follows.

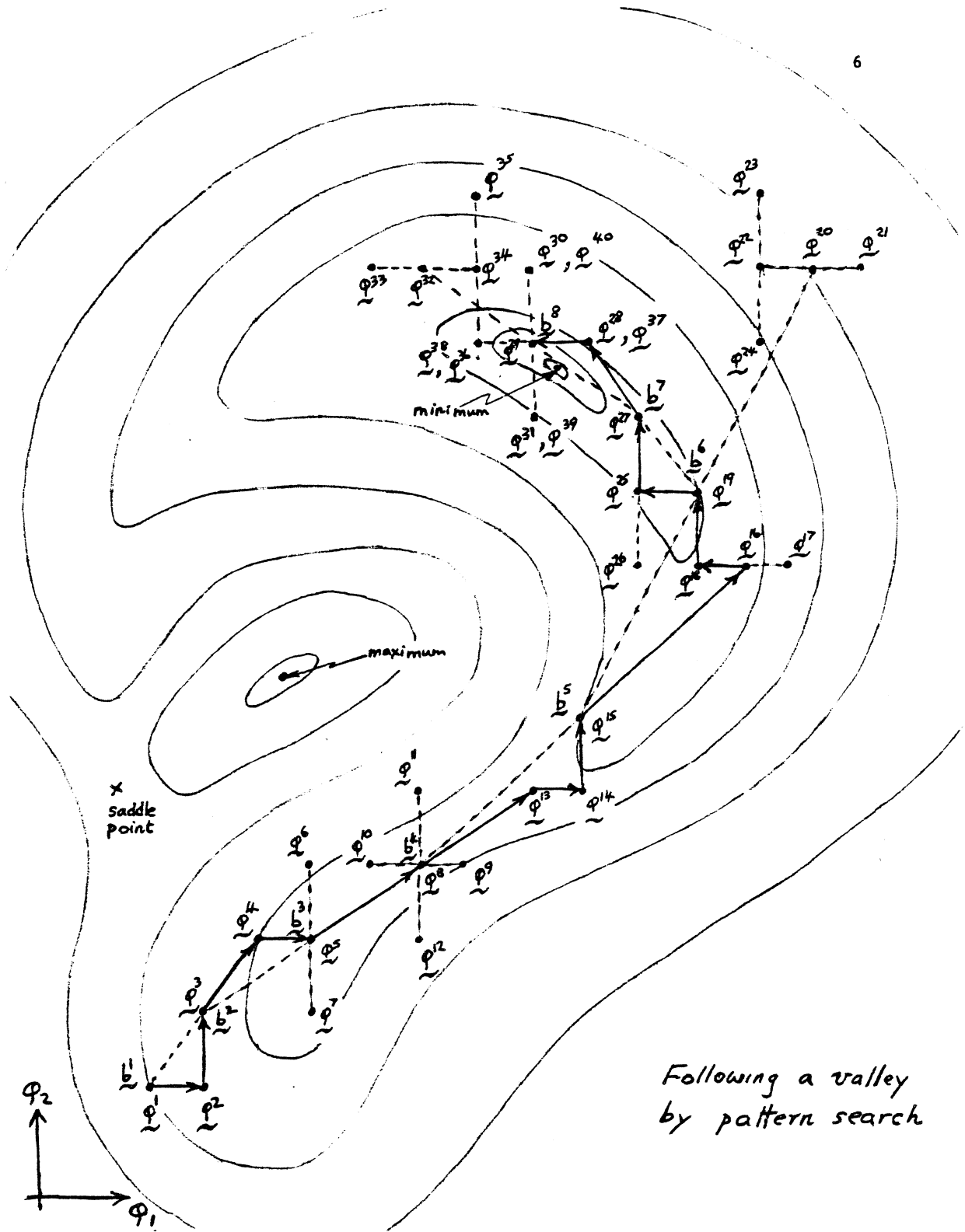
- $i$  subscript
- $k$  dimensionality of space
- $U$  objective function to be minimized
- $U_{\phi}$  value of  $U$  at  $\phi$
- $U_{\phi^0}$  value of  $U$  at  $\phi^0$
- $\alpha$  acceleration factor for pattern move (positive)
- $\beta$  reduction factor for exploratory increments (positive)
- $\delta$  vector containing exploratory increments
- $\delta_i$   $i$ th component of  $\delta$
- $\epsilon_i$   $i$ th component of vector containing minimum required exploratory increments
- $\phi$  current, projected or exploratory point
- $\phi_i$   $i$ th component of  $\phi$
- $\phi^0$  current base point
- $\phi_i^0$   $i$ th component of  $\phi^0$



Mathematical flow diagram for pattern search.

feature to ensure we have not come back to the base point.





Following a valley  
by pattern search

The starting point  $\phi^1$  is the first base point  $\underline{b}^1$ . In the example the first exploratory move from  $\phi^1$  begins by incrementing  $\phi_1$  and resulting in  $\phi^2$ . Since  $U^2 < U^1$ ,  $\phi^2$  is retained and exploration is continued by incrementing  $\phi_2$ .  $U^3 < U^2$  so  $\phi^3$  is retained in place of  $\phi^2$ . The first set of exploratory moves being complete,  $\phi^3$  becomes the second base point  $\underline{b}^2$ . A pattern move is now made to  $\phi^4 = 2\underline{b}^2 - \underline{b}^1$ , i.e., in the direction  $\underline{b}^2 - \underline{b}^1$ , in the hope that the previous success will be repeated. Thus, in this example, and most commonly, the factor  $\alpha = 1$ .  $U^4$  is not immediately compared with  $U^3$ . Instead, a set of exploratory moves is first made to try to improve on the pattern direction. (The usefulness of this feature is shown at  $\phi^{28}$ .) The best point found in the present example is  $\phi^5$  and, since  $U^5 < U^3$ , it becomes  $\underline{b}^3$ , the third base point. The search continues with a pattern move to  $\phi^8 = 2\underline{b}^3 - \underline{b}^2$ .

In an effort to exploit the current direction of success as indicated by the current base point and the point about to replace it, i.e., the pattern direction, subsequent exploratory moves initially try increments in the appropriate directions parallel to the coordinate axes.

When a pattern move and subsequent exploratory moves fail (as around  $\phi^{20}$  and  $\phi^{32}$ ), the pattern is destroyed and the strategy is to return to the current base point. Exploration around the current base point is carried out, the initial directions being those most recently used during the previous exploration. If the exploratory moves about the base point fail (as at  $\phi^{29}$  or  $\underline{b}^8$ ) the parameter increments are reduced and the whole procedure re-started at that point. The search may be terminated when the parameter increments fall below prescribed levels. Alternatively, the search can be terminated when the number of function evaluations or running time have reached upper limits.

### Rotating Coordinates

#### (i) Rosenbrock's Method

Let  $\underline{u}_1^j, \underline{u}_2^j, \dots, \underline{u}_k^j$  be the  $k$  mutually orthogonal directions (unit vectors) of search during the  $j$ th exploratory stage. For convenience, the given coordinate directions are chosen initially, i.e.,

$$\underline{u}_1^1 = \begin{bmatrix} \phi_1/|\phi_1| \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \underline{u}_2^1 = \begin{bmatrix} 0 \\ \phi_2/|\phi_2| \\ \vdots \\ 0 \end{bmatrix} \quad \dots \quad \underline{u}_k^1 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \phi_k/|\phi_k| \end{bmatrix}$$

A step  $e_i$  is taken in the  $\underline{u}_i^j$  direction. At the very beginning of the  $j$ th exploratory stage, for example, we go to  $\underline{\phi}^j + e_1 \underline{u}_1^j$ . If the move is successful (objective function does not become greater than the current best value) the success is noted, the point is retained and a new exploratory increment  $\alpha e_i$ , where  $\alpha > 1$ , is defined for the  $\underline{u}_i^j$  direction. If the move is unsuccessful, the failure is noted, we return to the previous point and a new exploratory increment  $-\beta e_i$ , where  $0 < \beta < 1$ , is defined for the  $\underline{u}_i^j$  direction.

This process is carried out for all  $i = 1, 2, \dots, k$  and repeated if necessary until one success followed by one failure has occurred in each direction. Let the point arrived at finally be  $\underline{\phi}^{j+1}$ .

When the  $j$ th exploratory stage is complete, the coordinates are rotated as follows. First we set

$$\underline{v}_k = d_k \underline{u}_k^j$$

$$\underline{v}_i = d_i \underline{u}_i^j + \underline{v}_{i+1} \quad i = k-1, \dots, 1$$

where  $d_1, d_2, \dots, d_k$  are the distances moved in the respective directions

since the previous rotation of the axes, i.e.,

$$\underline{\phi}^{j+1} - \underline{\phi}^j = d_1 \underline{u}_1^j + d_2 \underline{u}_2^j + \dots + d_k \underline{u}_k^j .$$

The new set of orthogonal unit vectors  $\underline{u}_1^{j+1}, \underline{u}_2^{j+1}, \dots, \underline{u}_k^{j+1}$  are obtained by the Gram-Schmidt procedure:

$$\begin{aligned} \underline{w}_1 &= \underline{v}_1 \\ \underline{u}_1^{j+1} &= \frac{\underline{w}_1}{\|\underline{w}_1\|} \\ \underline{w}_i &= \underline{v}_i - \sum_{p=1}^{i-1} (\underline{v}_i^T \underline{u}_p^{j+1}) \underline{u}_p^{j+1} \\ \underline{u}_i^{j+1} &= \frac{\underline{w}_i}{\|\underline{w}_i\|} \end{aligned} \quad \left. \vphantom{\begin{aligned} \underline{w}_1 &= \underline{v}_1 \\ \underline{u}_1^{j+1} &= \frac{\underline{w}_1}{\|\underline{w}_1\|} \\ \underline{w}_i &= \underline{v}_i - \sum_{p=1}^{i-1} (\underline{v}_i^T \underline{u}_p^{j+1}) \underline{u}_p^{j+1} \\ \underline{u}_i^{j+1} &= \frac{\underline{w}_i}{\|\underline{w}_i\|} \end{aligned}} \right\} i = 2, 3, \dots, k$$

Observe that the first of these directions always lies in the direction of total progress made during the  $j$ th stage since

$$\underline{u}_1^{j+1} = \frac{\underline{\phi}^{j+1} - \underline{\phi}^j}{\|\underline{\phi}^{j+1} - \underline{\phi}^j\|} .$$

Consider the case  $k = 2$ .

$$\underline{v}_1 = d_1 \underline{u}_1^j + d_2 \underline{u}_2^j$$

$$\underline{v}_2 = d_2 \underline{u}_2^j$$

$$\underline{w}_1 = \underline{v}_1$$

$$\underline{u}_1^{j+1} = \frac{d_1}{\sqrt{d_1^2 + d_2^2}} \underline{u}_1^j + \frac{d_2}{\sqrt{d_1^2 + d_2^2}} \underline{u}_2^j$$

$$\begin{aligned}
\tilde{u}_2 &= v_2 - (v_2^T \tilde{u}_1^{j+1}) \tilde{u}_1^{j+1} \\
&= d_2 \tilde{u}_2^j - (d_2 \tilde{u}_2^{jT} [ \frac{d_1}{\sqrt{d_1^2 + d_2^2}} \tilde{u}_1^j + \frac{d_2}{\sqrt{d_1^2 + d_2^2}} \tilde{u}_2^j ] ) ( \frac{d_1}{\sqrt{d_1^2 + d_2^2}} \tilde{u}_1^j + \frac{d_2}{\sqrt{d_1^2 + d_2^2}} \tilde{u}_2^j ) \\
&= d_2 \tilde{u}_2^j - \frac{d_2^2}{\sqrt{d_1^2 + d_2^2}} ( \frac{d_1}{\sqrt{d_1^2 + d_2^2}} \tilde{u}_1^j + \frac{d_2}{\sqrt{d_1^2 + d_2^2}} \tilde{u}_2^j ) \\
&= \frac{d_1^2 d_2 \tilde{u}_2^j + d_2^3 \tilde{u}_2^j - d_2^2 d_1 \tilde{u}_1^j - d_2^3 \tilde{u}_2^j}{d_1^2 + d_2^2} \\
&= \frac{d_1 d_2}{d_1^2 + d_2^2} (d_1 \tilde{u}_2^j - d_2 \tilde{u}_1^j) \\
\tilde{u}_2^{j+1} &= \frac{d_1}{\sqrt{d_1^2 + d_2^2}} \tilde{u}_2^j - \frac{d_2}{\sqrt{d_1^2 + d_2^2}} \tilde{u}_1^j
\end{aligned}$$

$\tilde{u}_1^{j+1}$  and  $\tilde{u}_2^{j+1}$  are clearly orthogonal unit vectors since

$$||\tilde{u}_1^{j+1}|| = ||\tilde{u}_2^{j+1}|| = 1$$

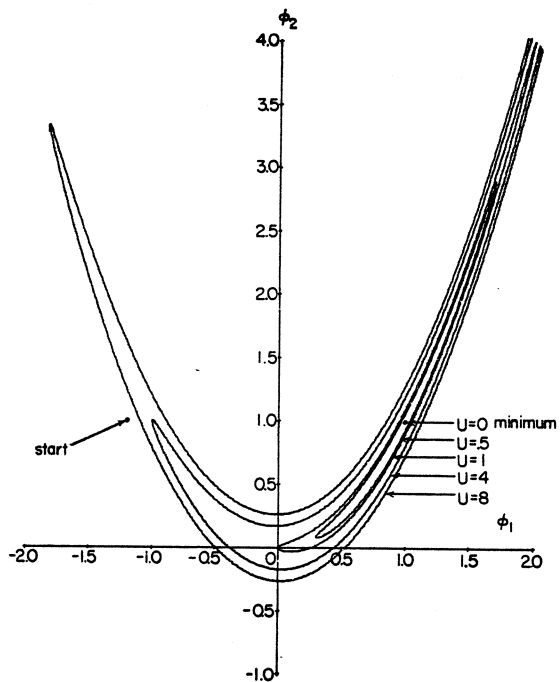
and

$$\tilde{u}_1^{j+1 T} \tilde{u}_2^{j+1} = 0.$$

A new exploratory stage, the  $(j+1)$ th, is started at  $\phi^{j+1}$ . The search may be terminated after a predetermined number of function evaluations or when the total progress made during each of several successive exploratory stages becomes smaller than a predetermined value.

If any of the  $d_i$  are zero the orthogonalization procedure may break down. Rosenbrock's method avoids this however by ensuring that some success is always achieved in every direction.





Contours of a standard test problem: Rosenbrock's function  
 $U = 100(\phi_2 - \phi_1)^2 + (1 - \phi_1)^2$ .

Experimentally, Rosenbrock found that  $\alpha = 3$ ,  $\beta = \frac{1}{2}$ , gives reasonable efficiency.

(ii) The Method of Davies, Swann and Campey

This method was reported by Swann to be more efficient than the methods of Hooke and Jeeves or Rosenbrock. It is essentially an improvement of Rosenbrock's method employing linear minimizations once along each of  $k$  mutually orthogonal directions in turn, after which the coordinates are rotated.

Let  $\phi_i^j$  be the starting point of the  $i$ th minimization during the  $j$ th stage. Then

$$\phi_{i+1}^j = \phi_i^j + d_i u_i^j$$

where  $d_i$  is selected by the one-dimensional method involving quadratic interpolation suggested by Davies, Swann and Campey. It is recommended that just one quadratic interpolation along each direction may be enough. The process is carried out for all  $i = 1, 2, \dots, k$  with  $\phi_1^{j+1}$  being set equal to  $\phi_{k+1}^j$ . When the  $j$ th stage is complete the coordinates can be rotated as already described.

Suppose that after one of the stages one or more of the  $d$ 's were zero. Consider specifically that  $d_r = d_s = 0$ . The following procedure can then be adopted.

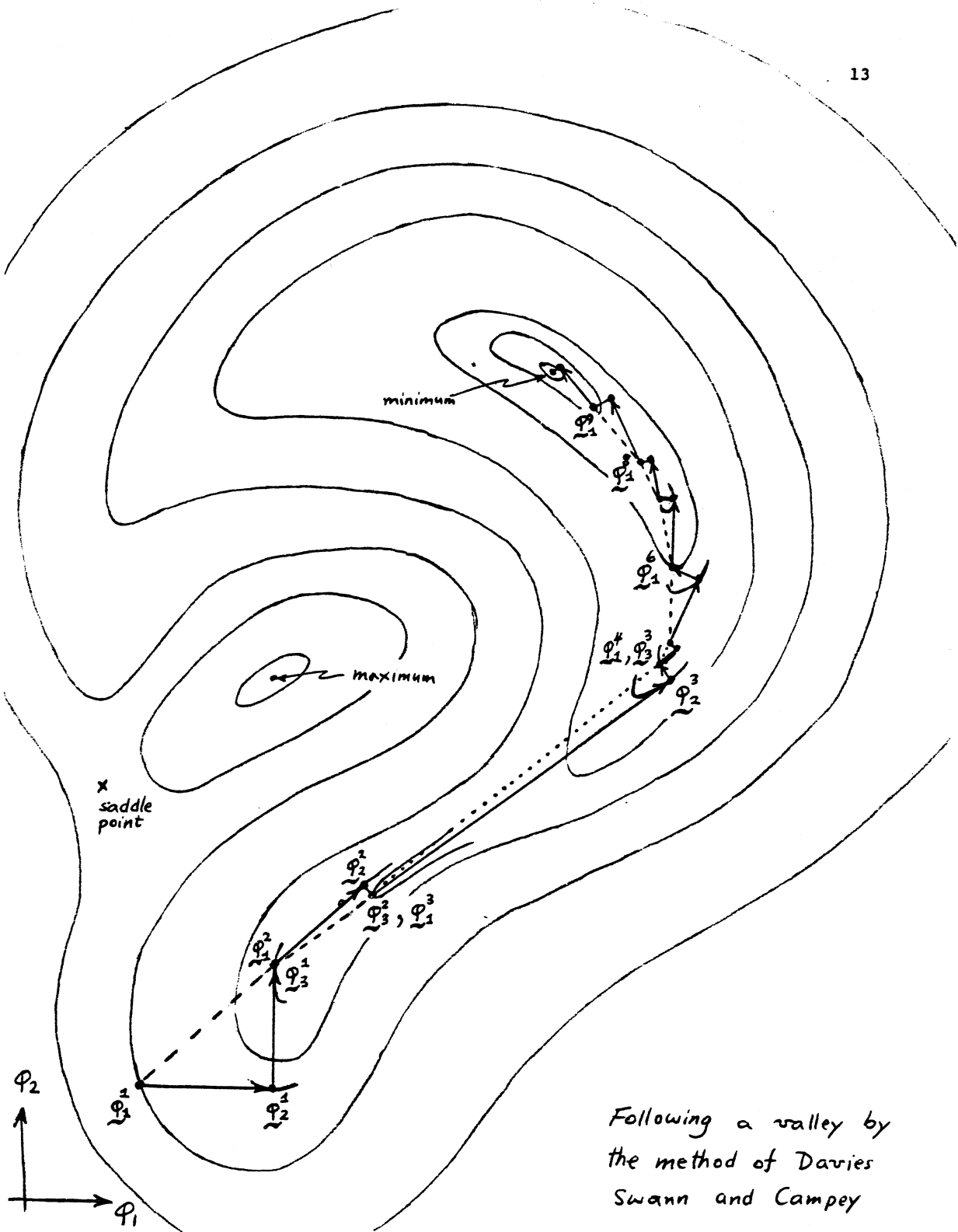
Reorder the directions

$$u_1^j, u_2^j, \dots, u_{r-1}^j, u_r^j, u_{r+1}^j, \dots, u_{s-1}^j, u_s^j, u_{s+1}^j, \dots, u_k^j$$

to

$$u_1^j, u_2^j, \dots, u_{r-1}^j, u_{r+1}^j, \dots, u_{s-1}^j, u_{s+1}^j, \dots, u_k^j, u_r^j, u_s^j.$$

Carry out the orthogonalization on the first  $k-2$  directions so that finally we have  $k-2$  new orthogonal directions plus two previous directions so that  $u_{k-1}^{j+1} = u_r^j$  and  $u_k^{j+1} = u_s^j$ . Because the first  $k-2$  vectors had no components in the  $u_r^j$  and  $u_s^j$  directions all the directions  $u_1^{j+1}, u_2^{j+1}, \dots, u_k^{j+1}$  are mutually orthogonal.



Following a valley by  
 the method of Davies  
 Swann and Campey

A suitable way of deciding whether the increment for the one-dimensional search is to be reduced is to compare it with the total distance moved during the previous stage. If the distance moved is felt to be too small the increment is reduced and the one-dimensional search is repeated along the previous  $k$  directions. Convergence is assumed when the increment has fallen below a prescribed level.

### Simplex Methods

Simplex methods of nonlinear optimization involve the following operations. A set of  $k+1$  points are set up in the  $k$ -dimensional  $\phi$  space to form a simplex. For two dimensions, for example, we would have a triangle and for three we would have a tetrahedron. The simplex is called regular if the points are equidistant.

The objective function is evaluated at each vertex and an attempt to form a new simplex by replacing the vertex with the greatest value of the objective function by another point is made. A simplex method was first introduced by Spendley, Hext and Himsworth. A further development of the idea has been presented by Nelder and Mead. The method, which has very desirable valley following properties, is described by way of an example.

Let

$$U_h = U(\phi_h) = \max_i U(\phi_i) \quad i = 1, 2, \dots, k+1$$

where  $\phi_1, \phi_2, \dots, \phi_{k+1}$  are the vertices of the simplex.

Let

$$U_s = U(\phi_s) = \max_{i \neq h} U(\phi_i)$$

and

$$U_l = U(\phi_l) = \min_i U(\phi_i).$$

Thus,  $U_h$  corresponds to the highest function value,  $U_s$  to the second highest and  $U_l$  to the lowest function value during the current stage.

The centroid of all points excluding  $\phi_h$  is

$$\bar{\phi} = \frac{1}{k} \sum_{\substack{i=1 \\ i \neq h}}^{k+1} \phi_i.$$

Consider the example. Clearly

$$\begin{aligned}\phi_h &= \phi^1 \\ \phi_s &= \phi^2 \\ \phi_l &= \phi^3 \\ \bar{\phi} &= \frac{1}{2}(\phi^2 + \phi^3).\end{aligned}$$

An attempt to replace  $\phi_h$  is carried out by defining a reflection in  $\bar{\phi}$  as

$$\phi_r = \bar{\phi} + \alpha(\bar{\phi} - \phi_h)$$

where  $\alpha > 0$  is called the reflection coefficient. In the example  $\phi_r = \phi^4$ . Let  $U_r = U(\phi_r)$ . If  $U_r < U_l$ , i.e., if the point just obtained is better than the current best one an expansion is attempted:

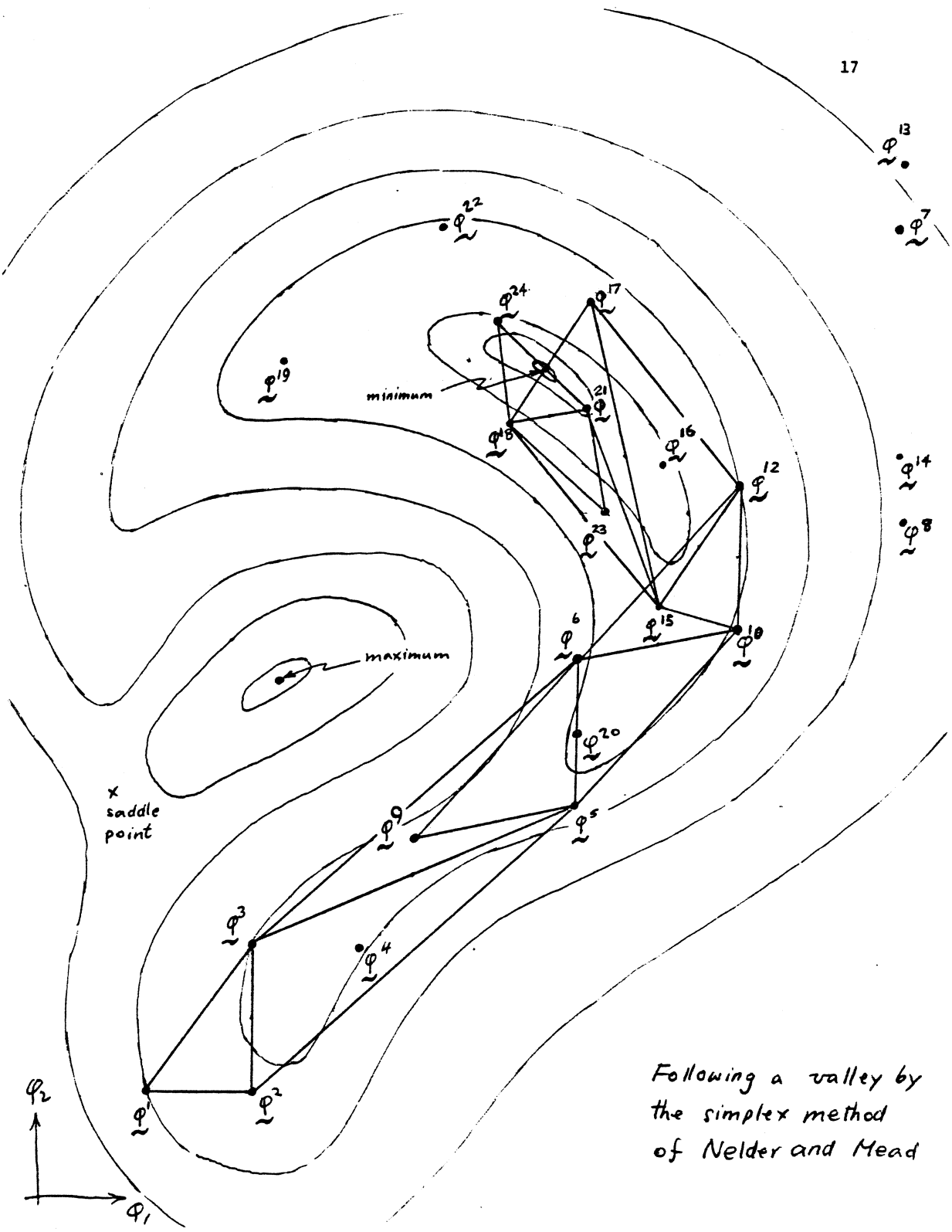
$$\phi_e = \bar{\phi} + \gamma(\phi_r - \bar{\phi})$$

where  $\gamma > 1$  is called the expansion coefficient. In the example  $U^4 < U^3$  so we try  $\phi_e = \phi^5$ . Let  $U_e = U(\phi_e)$ . If  $U_e < U_l$  we replace  $\phi_h$  by  $\phi_e$  and restart the process by redefining the  $\phi_h$ ,  $\phi_s$  and  $\phi_l$ . Since  $U^5 < U^3$  points  $\phi^2$ ,  $\phi^3$  and  $\phi^5$  form the new simplex.

The point  $\phi^6$  is obtained by reflection of  $\phi^2$  in  $\frac{1}{2}(\phi^3 + \phi^5)$ . Since  $U^6 < U^5$  we expand to  $\phi^7$ . However  $U^7 \not< U^5$ , so we return to  $\phi^6$ . Thus, if expansion has failed we replace  $\phi_h$  by  $\phi_r$  and restart the process as before. So our third simplex is formed by  $\phi^3$ ,  $\phi^5$  and  $\phi^6$ . Next we try  $\phi^8$  which is unacceptable. Thus, if  $U_r > U_h$  we attempt a contraction by defining

$$\phi_c = \bar{\phi} + \beta(\phi_h - \bar{\phi})$$

where  $0 < \beta < 1$  is called the contraction coefficient. Contraction is deemed successful if  $U_c < U_h$ , where  $U_c = U(\phi_c)$ . Note that if  $U_h > U_r > U_s$  then we first replace  $\phi_h$  by  $\phi_r$  and redefine  $\phi_h$  accordingly before contracting.



Following a valley by the simplex method of Nelder and Mead

In the example  $\phi^9$  is the result of successful contraction.

If contraction fails, i.e.,  $U_c \geq U_h$  the following shrinking tactic is employed:

$$\phi_i \leftarrow \frac{1}{2}(\phi_i + \phi_\ell) \quad \text{all } i \neq \ell$$

Following successful contraction or shrinking the process is restarted with the new simplex.

If  $U_s \geq U_r \geq U_\ell$  then  $\phi_h$  is replaced by  $\phi_r$ . This occurs with the simplex formed by  $\phi^{18}$ ,  $\phi^{21}$  and  $\phi^{23}$ . Observe that  $U^{18} > U^{24} > U^{21}$ .

A possible terminating criterion could be

$$\frac{1}{k} \sum_{i=1}^{k+1} (U(\phi_i) - U(\bar{\phi}))^2 < \epsilon^2$$

where  $\epsilon$  is prescribed.

In the example  $\alpha = 1$ ,  $\beta = \frac{1}{2}$  and  $\gamma = 2$ .

Some reports state that this method is remarkably efficient for up to four parameters, progress on problems having more dimensions being rather slow. Yet other reports appear much more optimistic.



**SECTION EIGHTEEN**  
**EXAMPLES AND PROBLEMS**

© J.W. Bandler 1988

This material is taken from previous years' assignments and examples. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.



1. Apply the Fletcher-Powell-Davidon updating formula to the minimization of:

$$\phi_1^2 + 2\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 1$$

w.r.t.  $\phi_1$  and  $\phi_2$  starting at  $\phi_1 = 0$ ,  $\phi_2 = 0$ , showing all steps explicitly and commenting on the results obtained.

2. Apply the conjugate gradient algorithm for minimizing a differentiable function of many variables to the minimization of:

$$\phi_1^2 + 2\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 1$$

w.r.t.  $\phi_1$  and  $\phi_2$  starting at  $\phi_1 = 0$ ,  $\phi_2 = 0$ , showing all steps explicitly and commenting on the results obtained.

3. Apply the conjugate gradient algorithm for minimizing a differentiable function of many variables to the following data.

Point:  $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ ,  $\begin{bmatrix} 4 \\ 0 \end{bmatrix}$ ,  $\begin{bmatrix} 8 \\ 2 \end{bmatrix}$ ,  $\begin{bmatrix} 8.4 \\ 2.45 \end{bmatrix}$ , ...

Gradient:  $\begin{bmatrix} -1 \\ 0 \end{bmatrix}$ ,  $\begin{bmatrix} 0 \\ -2 \end{bmatrix}$ ,  $\begin{bmatrix} 1 \\ -2 \end{bmatrix}$ ,  $\begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}$ , ...

Sketch contours of a reasonable function that might have produced these numbers and plot the path taken by the algorithm.

4. Verify that the point  $\phi_1 = \phi_2 = 1$  is a solution to the minimax problem for which

$$f_1 = \phi_1^4 + \phi_2^2$$

$$f_2 = (2 - \phi_1)^2 + (2 - \phi_2)^2$$

$$f_3 = 2 \exp(-\phi_1 + \phi_2)$$

by invoking necessary conditions for a minimax optimum.

5. Starting with the interval  $[0,6]$ , apply 4 iterations of the Golden Section search method to the minimization w.r.t.  $\phi$  of a function described by

$$\begin{aligned} U &= -\phi + 5 & \phi &\leq 1 \\ U &= 0.5(\phi - 3)^2 + 1 & 1 &\leq \phi \leq 4 \\ U &= 3 - (\phi - 6)^2/3 & \phi &\geq 4 \end{aligned}$$

What is the solution obtained? By how much has the interval of uncertainty been reduced?

6. Derive from first principles an approach to calculating  $\partial y_i / \partial \underline{x}$ , where  $\underline{A} \underline{y} = \underline{b}$  is a linear system in  $\underline{y}$ ,  $\underline{A}$  is a square matrix whose coefficients are nonlinear functions of  $\underline{x}$ , the term  $y_i$  is the  $i$ th component of the column vector  $\underline{y}$  and  $\partial y_i / \partial \underline{x}$  represents a column vector containing partial derivative  $y_i$  w.r.t. corresponding elements of the column vector  $\underline{x}$ . Discuss the computational effort involved.

7. Derive from first principles an approach to calculating  $\partial\lambda/\partial\mathbf{x}$ , where  $\lambda$  is an eigenvalue of the square matrix  $\underline{A}$  whose coefficients are nonlinear functions of  $\mathbf{x}$ . The expression  $\partial\lambda/\partial\mathbf{x}$  is a column vector containing all first partial derivatives of  $\lambda$  w.r.t. corresponding elements of the column vector  $\mathbf{x}$ . Discuss the computational effort involved.
8. Consider the voltage divider example of Assignment 2 expressed as a minimax problem. Determine suitable active functions when

$$R_1 = 1.01$$

$$R_2 = 1.14$$

and calculate the steepest descent direction from first principles. Assume that if  $|M - f_i| \leq 0.01$  for any  $f_i$ , then the corresponding  $f_i$  is active, where  $M \stackrel{\Delta}{=} \max f_i$ . Show all steps in your calculations.

## SELECTED PROBLEMS AND THEIR SOLUTION

(Assignment 3, October 1982)

### PROBLEM #1

To minimize  $\phi_1^2 + 2\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 1$  w.r.t  $\phi_1$  and  $\phi_2$  starting at  $\phi_1 = 0, \phi_2 = 0$  by applying DFP updating formula:-

We have 
$$U = \frac{1}{2} \phi^T A \phi + b^T \phi + c$$

where

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$$

$$b = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

$$c = 1$$

$$\text{and } \phi^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Step 1 Find  $\nabla U$

$$\nabla U = A\phi + b = \begin{bmatrix} 2\phi_1 + \phi_2 + 2 \\ \phi_1 + 4\phi_2 \end{bmatrix}$$

$$\nabla U(\phi^0) = \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \nabla U^0$$

Step 2 Determine  $S^j = -H^j \nabla U^j$  [search direction]

$$\text{at } j=0 \quad S^0 = -H^0 \nabla U^0$$

$$H^0 = I \quad \therefore S^0 = -\nabla U^0 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

Step 3 Determine  $\alpha$  by evaluating [step length]

$$U(\phi^0 + \alpha S^0), \text{ where } \phi^0 + \alpha S^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -2\alpha \\ 0 \end{bmatrix}$$

$$U = 4\alpha^2 - 4\alpha + 1$$

$$\frac{dU}{d\alpha} = 8\alpha - 4 = 0 \quad \therefore \alpha^0 = 0.5$$

step 4 Calculate  $\Delta\phi^j = \alpha^j s^j$

$$\Delta\phi^0 = \alpha^0 s^0 = 0.5 \begin{bmatrix} -2 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

and update  $\phi$

$$\phi^1 = \phi^0 + \Delta\phi^0 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

From step 1

$$\nabla U^1 = \begin{bmatrix} -2 & +2 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

step 5 determine the change in gradient

$$g^j = \nabla U^{j+1} - \nabla U^j$$

we have

$$g^0 = \nabla U^1 - \nabla U^0 = \begin{bmatrix} 0 \\ -1 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$$

step 6 Update  $H$  by using DFP formula

$$H^{j+1} = H^j + \frac{\Delta\phi^j (\Delta\phi^j)^T}{(\Delta\phi^j)^T g^j} - \frac{H^j g^j (g^j)^T H^j}{(g^j)^T H^j g^j}$$

we have

$$\begin{aligned} H^1 &= H^0 + \frac{\Delta\phi^0 (\Delta\phi^0)^T}{(\Delta\phi^0)^T g^0} - \frac{H^0 g^0 (g^0)^T H^0}{(g^0)^T H^0 g^0} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{\begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} -1 & 0 \end{bmatrix}}{\begin{bmatrix} -1 & 0 \end{bmatrix} \begin{bmatrix} -2 \\ -1 \end{bmatrix}} - \frac{\begin{bmatrix} -2 & -1 \end{bmatrix} \begin{bmatrix} -2 & -1 \end{bmatrix}}{\begin{bmatrix} -2 & -1 \end{bmatrix} \begin{bmatrix} -2 \\ -1 \end{bmatrix}} \end{aligned}$$

[multiplication with  $I$  ignored]

$$\begin{aligned} H^1 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} - \frac{1}{5} \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{7}{10} & -\frac{4}{10} \\ -\frac{4}{10} & \frac{8}{10} \end{bmatrix} \end{aligned}$$

From step 2:

$$s^1 = -H^1 \nabla U^1$$

$$= - \begin{bmatrix} 0.7 & -0.4 \\ -0.4 & 0.8 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -0.4 \\ 0.8 \end{bmatrix}$$

From step 3:

$$\phi^1 + \alpha s^1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} -0.4 \\ 0.8 \end{bmatrix} = \begin{bmatrix} -1 - 0.4\alpha \\ 0.8\alpha \end{bmatrix}$$

$$U = 1.12\alpha^2 - 0.8\alpha$$

$$\frac{dU}{d\alpha} = 2.24\alpha - 0.8 = 0$$

$$\alpha = \frac{0.8}{2.24} = \frac{5}{14}$$

From step 4:

$$\Delta \phi^1 = \alpha^1 s^1 = \frac{5}{14} \begin{bmatrix} -0.4 \\ 0.8 \end{bmatrix} = \begin{bmatrix} -\frac{1}{7} \\ \frac{2}{7} \end{bmatrix}$$

$$\phi^2 = \phi^1 + \Delta \phi^1$$

$$= \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{1}{7} \\ \frac{2}{7} \end{bmatrix} = \begin{bmatrix} -\frac{8}{7} \\ \frac{2}{7} \end{bmatrix}$$

From step 1

$$\nabla U^2 = \begin{bmatrix} -\frac{16}{7} + \frac{2}{7} + 2 \\ -\frac{8}{7} + \frac{8}{7} + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

From step 5

$$g^j = \nabla U^2 - \nabla U^1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

From step 6

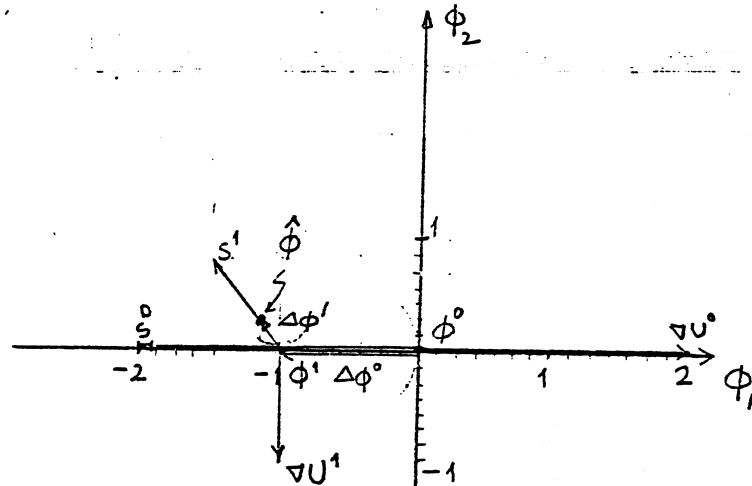
$$H^2 = \begin{bmatrix} \frac{7}{10} & -\frac{4}{10} \\ -\frac{4}{10} & \frac{8}{10} \end{bmatrix} + \frac{\begin{bmatrix} -\frac{1}{7} \\ \frac{2}{7} \end{bmatrix} \begin{bmatrix} -\frac{1}{7} & \frac{2}{7} \end{bmatrix}}{\begin{bmatrix} -\frac{1}{7} \\ \frac{2}{7} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}} - \frac{\begin{bmatrix} \frac{7}{10} & -\frac{4}{10} \\ -\frac{4}{10} & \frac{8}{10} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{7}{10} & -\frac{4}{10} \\ -\frac{4}{10} & \frac{8}{10} \end{bmatrix}}{\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{7}{10} & -\frac{4}{10} \\ -\frac{4}{10} & \frac{8}{10} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}}$$

$$= \begin{bmatrix} .7 & -4 \\ -4 & .8 \end{bmatrix} + \frac{1}{14} \begin{bmatrix} 1 & -2 \\ -2 & 4 \end{bmatrix} - \begin{bmatrix} .2 & -4 \\ -4 & .8 \end{bmatrix}$$



$$H^2 = \begin{bmatrix} \frac{4}{7} & -\frac{1}{7} \\ -\frac{1}{7} & \frac{2}{7} \end{bmatrix}$$

For a check  $H^2 = A^{-1}$ .



$U =$	$\phi^0$	$\phi^1$	$\phi^2$
	1	0	$-\frac{27}{49}$

The exact solution has been obtained in two steps as the objective function is quadratic in nature.

PROBLEM # 2

To minimize  $\phi_1^2 + 2\phi_1\phi_2 + \phi_2^2 + 2\phi_1 + 1$   
w.r.t  $\phi_1$  and  $\phi_2$  starting at  $\phi_1 = 0, \phi_2 = 0$   
by applying CONJUGATE GRADIENT ALGORITHM

Step 1 Find the gradient

$$\nabla U = A\phi + b = \begin{bmatrix} 2\phi_1 + \phi_2 + 2 \\ \phi_1 + 4\phi_2 \end{bmatrix}$$

$$\nabla U^0 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

Step 2

find the direction  $s^j = -\nabla U^j + \beta^j s^{j-1}$   
initially  $\beta^0 = 0$

$$\therefore s^0 = -\nabla U^0 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

Step 3

find  $\alpha$  to give minimum  $U$  in direction  
of  $s^0$

$$U(\phi^0 + \alpha s^0) :-$$
$$U = 4\alpha^2 - 4\alpha + 1$$

$$\frac{dU}{d\alpha} = 8\alpha - 4 \quad \therefore \alpha^0 = 0.5$$

Step 4

Obtain

$$\phi^{j+1} = \phi^j + \alpha^j s^j = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

from step 1

$$\nabla U^1 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

Step 5

Calculate

$$\beta^j = \frac{(\nabla U^j)^T \nabla U^j}{(\nabla U^{j-1})^T \nabla U^{j-1}}$$

we have  $\beta^1 = \frac{(\nabla U^1)^T \nabla U^1}{(\nabla U^0)^T \nabla U^0} = \frac{[0 \ -1] \begin{bmatrix} 0 \\ -1 \end{bmatrix}}{[2 \ 0] \begin{bmatrix} 2 \\ 0 \end{bmatrix}} = \frac{1}{4}$

from step 2

$$s^1 = -\nabla U^1 + \beta^1 s^0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} -2 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 1.0 \end{bmatrix}$$

from step 3

$$U(\phi^1 + \alpha s^1) = 1.75\alpha^2 - \alpha$$

$$\frac{dU}{d\alpha} = 3.5\alpha - 1 = 0 \quad \therefore \alpha_1 = \frac{1}{3.5} = \frac{2}{7}$$

from step 4

$$\phi^2 = \phi^1 + \alpha_1 s^1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \frac{2}{7} \begin{bmatrix} -0.5 \\ 1.0 \end{bmatrix} = \begin{bmatrix} -\frac{8}{7} \\ \frac{2}{7} \end{bmatrix}$$

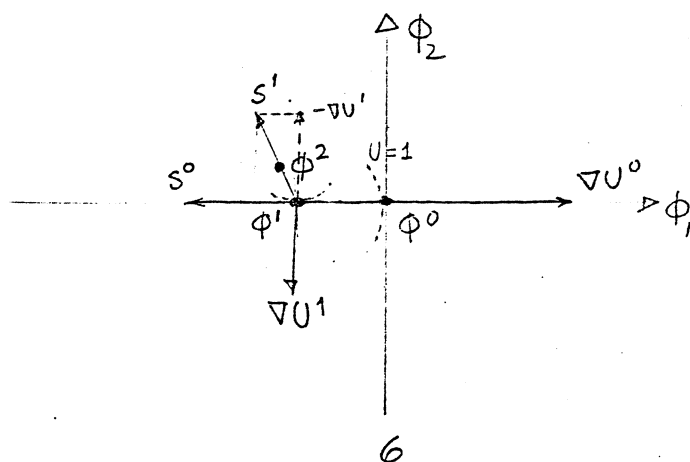
from step 1

$$\nabla U^2 = \begin{bmatrix} -\frac{16}{7} + \frac{2}{7} + 2 \\ -\frac{8}{7} + \frac{8}{7} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\therefore \beta^2 = 0$$

$$\text{and } s^2 = 0$$

hence the search stops as the minima has been reached.

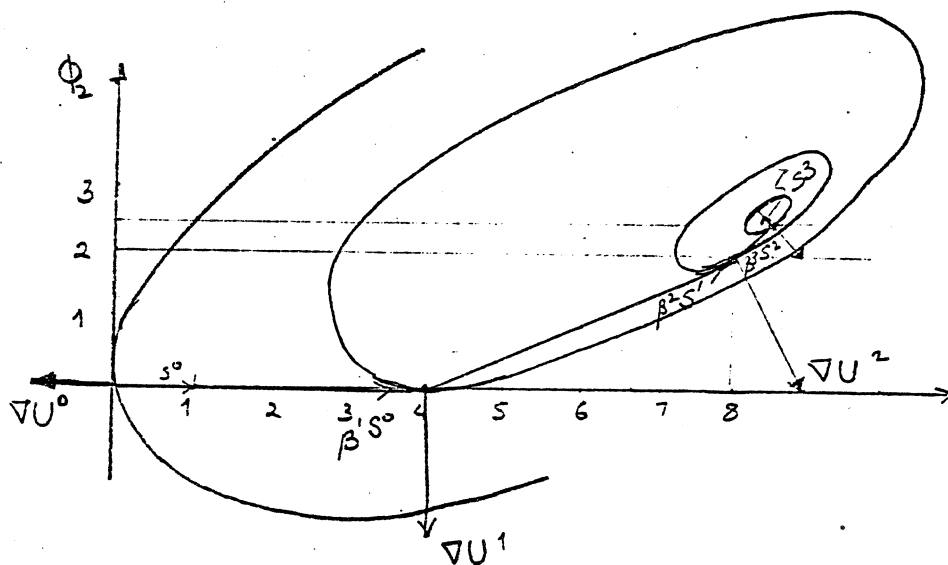


$$\phi^2 = \hat{\phi}$$

PROBLEM # 3.

The data available is

$f$	$0$	$1$	$2$	$3$
$\phi$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 4 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 8 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 8.4 \\ 2.45 \end{bmatrix}$
$\nabla U$	$\begin{bmatrix} -1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -2 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -2 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}$



Step 1

$$s^j = -\nabla U^j + \beta^j s^{j-1}$$

Conjugate gradient method

$$s^0 = -\nabla U^0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\therefore \beta^0 = 0$$

Step 2

$$\beta^1 = \frac{(\nabla U^1)^T \nabla U^1}{(\nabla U^0)^T \nabla U^0} = \frac{[0 \ -2] \begin{bmatrix} 0 \\ -2 \end{bmatrix}}{[-1 \ 0] \begin{bmatrix} -1 \\ 0 \end{bmatrix}} = 4$$

from 1

$$s^1 = -\nabla U^1 + \beta^1 s^0 = \begin{bmatrix} 0 \\ 2 \end{bmatrix} + 4 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

from 2

$$\beta^2 = \frac{(\nabla U^2)^T \nabla U^2}{(\nabla U^1)^T \nabla U^1} = \frac{[1 \ -2] \begin{bmatrix} 1 \\ -2 \end{bmatrix}}{[0 \ -2] \begin{bmatrix} 0 \\ -2 \end{bmatrix}} = \frac{5}{4}$$

from step 1

$$\begin{aligned} s^2 &= -\nabla U^2 + \beta^2 s^1 \\ &= \begin{bmatrix} -1 \\ 2 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4.5 \end{bmatrix} \end{aligned}$$

from step 2

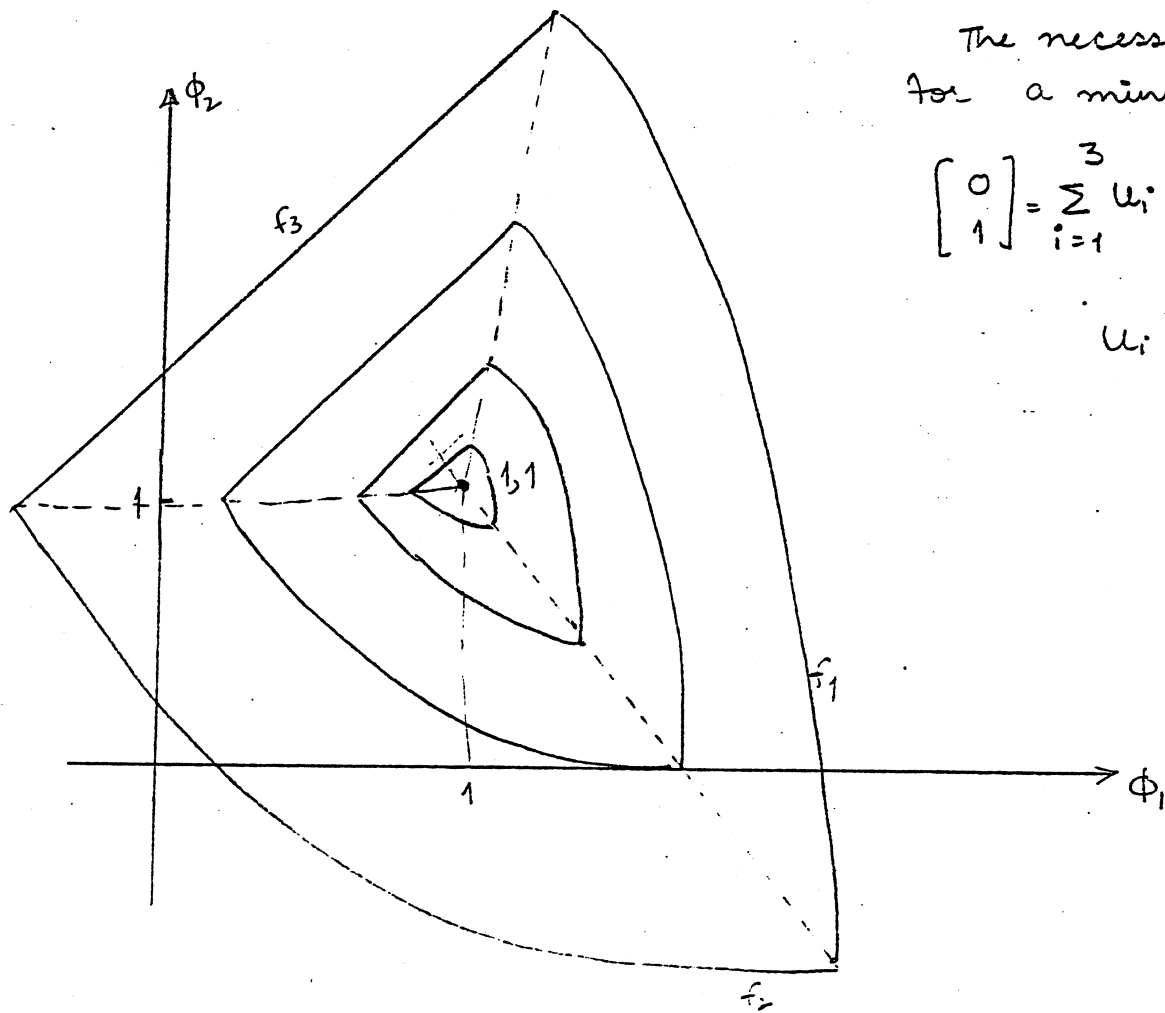
$$\beta^3 = \frac{(\nabla U^3)^T \nabla U^3}{(\nabla U^2)^T \nabla U^2} = \frac{0.5}{5} = 0.1$$

and

$$s^3 = -\nabla U^3 + \beta^3 s^2$$

$$= \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix} + 0.1 \begin{bmatrix} 4 \\ 4.5 \end{bmatrix} = \begin{bmatrix} -0.1 \\ 0.95 \end{bmatrix}$$

### PROBLEM # 4



The necessary conditions for a minimax optimum

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \sum_{i=1}^3 u_i \begin{bmatrix} -\nabla f_i(\phi^0) \\ 1 \end{bmatrix}$$

$$u_i \geq 0$$

The problem is formulated as following

$$\text{minimize } U = \phi_{k+1}$$

$$\text{s.t. } \phi_{k+1} \geq f_i(\phi) \quad i=1,2,3$$

Rewriting the constraints as

$$g_i(\phi) \triangleq \phi_{k+1} - f_i(\phi) \geq 0 \quad i=1,2,3$$

The conditions give the following equations

$$u_1 \nabla f_1 + u_2 \nabla f_2 + u_3 \nabla f_3 = 0$$

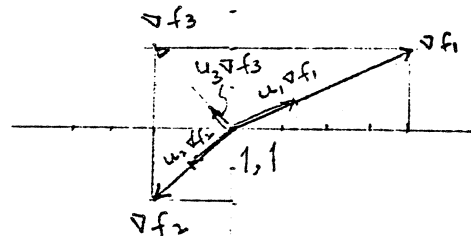
$$\text{or } u_1 \begin{bmatrix} 4\phi_1^3 \\ 2\phi_2 \end{bmatrix} + u_2 \begin{bmatrix} -2(2-\phi_1) \\ -2(2-\phi_2) \end{bmatrix} + u_3 \begin{bmatrix} -2 \exp(-\phi_1 + \phi_2) \\ +2 \exp(-\phi_1 + \phi_2) \end{bmatrix} = 0$$

$$\text{also } u_1 + u_2 + u_3 = 1$$

Evaluating  $\nabla f_i$  at the specified point, we can write the equations in matrix form

$$\begin{bmatrix} 4 & -2 & -2 \\ 2 & -2 & 2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{or } \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}$$



PROBLEM # 5

The interval of uncertainty at start is

$$I^1 = u - l = 6$$

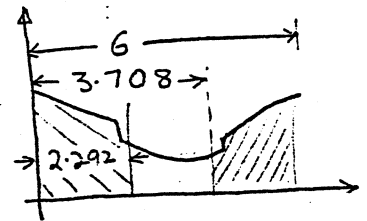
step 1 Take two interior points  $a, b$  apply Golden section

$$\phi_a^1 = \frac{1}{\tau^2} I^1 + \phi_l^1 \quad \text{where } \tau = 1.618034$$

$$= \frac{6.0}{(1.618034)^2} + 0 = 2.292$$

$$\phi_b^1 = \frac{1}{\tau} I^1 + \phi_l^1$$

$$= \frac{6.0}{1.618034} + 0 = 3.708$$



Now  $U(\phi_a^1) = 0.5 (2.292 - 3)^2 + 1 = 0.5 (.708)^2 + 1$

$$U(\phi_b^1) = 0.5 (3.708 - 3)^2 + 1 = 0.5 (.708)^2 + 1$$

we have  $U(\phi_a) = U(\phi_b) = 1.2508$

Next iteration  $\phi_a^2, \phi_b^2$  become the interior points and  $\phi_a^1, \phi_b^1$  as exterior points.

step 2 Applying Golden section again

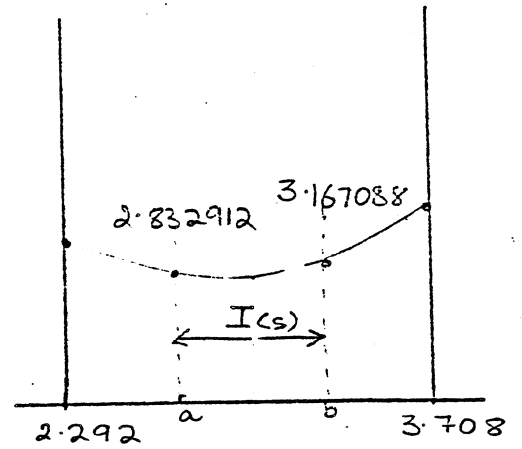
$$\phi_a^2 = \frac{1}{\tau^2} (1.416) + 2.292 = 2.832912$$

$$\phi_b^2 = \frac{1}{\tau} (1.416) + 2.292 = 3.167088$$

step 3

$$\begin{aligned}\phi_a^3 &= \frac{1}{\tau^2} 0.334176 + 2.832912 \\ &= 2.96056\end{aligned}$$

$$\begin{aligned}\phi_b^3 &= \frac{1}{\tau} 0.334176 + 2.832912 \\ &= 3.039432\end{aligned}$$



step 4

$$\phi_a^4 = \frac{1}{\tau^2} \cdot 0.78865 + 2.96056 = 2.99068$$

$$\phi_b^4 = \frac{1}{\tau} \cdot 0.78865 + 2.96056 = 3.00932$$

and so on

$$\hat{\phi} = 3.000$$

The reduction in uncertainty interval:

$$\textcircled{1} \quad \frac{6}{3.708} = \frac{3.708}{2.292} = \tau$$

$$\textcircled{2} \quad \frac{1.416}{3.167088 - 2.292} = \frac{3.167088 - 2.292}{2.832912 - 2.292} = \tau$$



PROBLEM #6

Given that  $\underline{A}(\underline{x})$  is a square matrix whose  
 $a_{ij} = f(\underline{x})$  nonlinear functions

$\underline{A} \underline{y} = \underline{b}$  is a linear system in  $\underline{y}$   
 differentiating both sides w.r.t.  $x_i$

$$\frac{\partial \underline{A}(\underline{x})}{\partial x_i} \underline{y} + \underline{A}(\underline{x}) \frac{\partial \underline{y}}{\partial x_i} = \underline{0}$$

Assuming  $\frac{\partial \underline{A}(\underline{x})}{\partial x_i}$  and  $\underline{y}$  available from above  
 equations

$$\frac{\partial \underline{y}}{\partial x_j} = - \underline{A}^{-1}(\underline{x}) \frac{\partial \underline{A}(\underline{x})}{\partial x_j} \underline{y}$$

In order to find  $i^{\text{th}}$  row of  $\frac{\partial \underline{y}}{\partial x_j}$  vector  
 premultiply the above equation by  $\underline{u}_i^T$  on  
 both sides

$$\frac{\partial y_i}{\partial x_j} = - \underline{u}_i^T \underline{A}^{-1}(\underline{x}) \frac{\partial \underline{A}(\underline{x})}{\partial x_j} \underline{y}$$

To avoid the calculations of  $\underline{A}^{-1}$  we can use

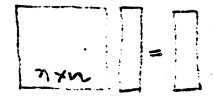
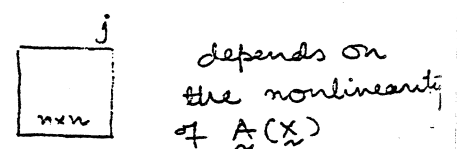
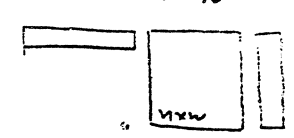
$$\underline{A}^T(\underline{x}) \hat{\underline{y}}_i = \underline{u}_i, \text{ and the equation becomes}$$

$$\frac{\partial y_i}{\partial x_j} = - \hat{\underline{y}}_i^T \frac{\partial \underline{A}(\underline{x})}{\partial x_j} \underline{y}$$

and

$$\frac{\partial y_i}{\partial \underline{x}} = \left[ - \hat{\underline{y}}_i^T \frac{\partial \underline{A}(\underline{x})}{\partial x_1} \underline{y} \quad - \hat{\underline{y}}_i^T \frac{\partial \underline{A}(\underline{x})}{\partial x_2} \underline{y} \quad \dots \quad - \hat{\underline{y}}_i^T \frac{\partial \underline{A}(\underline{x})}{\partial x_n} \underline{y} \right]^T$$

COMPUTATION EFFORT INVOLVED:

- i To solve  $\tilde{A}^T(\tilde{x}) \tilde{y}_i = u_i$  
- ii To get  $\frac{\partial \tilde{A}(\tilde{x})}{\partial x_j}$  
- iii To multiply  $\tilde{y}_i^T \frac{\partial \tilde{A}(\tilde{x})}{\partial x_j} \tilde{y}$  

PROBLEM #7

By definition, the eigen value must satisfy

$$\tilde{A}(\tilde{x}) \tilde{u} = \lambda \tilde{u} \quad \text{and} \quad \tilde{A}^T \tilde{y} = \lambda \tilde{y} \quad \text{①}$$

Differentiating equation ① w.r.t  $x_j$  we have

$$\frac{\partial \tilde{A}(\tilde{x})}{\partial x_j} \tilde{u} + \tilde{A}(\tilde{x}) \frac{\partial \tilde{u}}{\partial x_j} = \frac{\partial \lambda}{\partial x_j} \tilde{u} + \lambda \frac{\partial \tilde{u}}{\partial x_j} \quad \text{②}$$

multiplying both sides of equation ② by  $\tilde{y}^T$  we have

$$\tilde{y}^T \frac{\partial \tilde{A}(\tilde{x})}{\partial x_j} \tilde{u} + \tilde{y}^T \tilde{A}(\tilde{x}) \frac{\partial \tilde{u}}{\partial x_j} = \tilde{y}^T \frac{\partial \lambda}{\partial x_j} \tilde{u} + \tilde{y}^T \lambda \frac{\partial \tilde{u}}{\partial x_j}$$

$$\text{or } \tilde{y}^T \frac{\partial \tilde{A}(\tilde{x})}{\partial x_j} \tilde{u} + \tilde{y}^T \left[ \tilde{A}(\tilde{x}) - \lambda \mathbf{I} \right] \frac{\partial \tilde{u}}{\partial x_j} = \tilde{y}^T \frac{\partial \lambda}{\partial x_j} \tilde{u}$$

$$\text{but } \tilde{A}(\tilde{x}) - \lambda \mathbf{I} = 0$$

∴ the equation reduces to

$$\tilde{y}^T \frac{\partial A(\tilde{x})}{\partial x_j} \tilde{u} = \tilde{y}^T \tilde{u} \frac{\partial \lambda}{\partial x_j}$$

as  $\tilde{y}^T \tilde{u}$  is a scalar

or

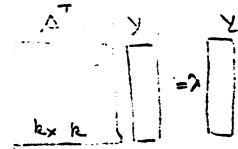
$$\frac{\partial \lambda}{\partial x_j} = \left( \tilde{y}^T \frac{\partial A}{\partial x_j} \tilde{u} \right) \frac{1}{\tilde{y}^T \tilde{u}}$$

Summarizing

$$\frac{\partial \lambda}{\partial \tilde{x}} = \begin{bmatrix} \frac{\tilde{y}^T \frac{\partial A}{\partial x_1} \tilde{u}}{\tilde{y}^T \tilde{u}} & \frac{\tilde{y}^T \frac{\partial A}{\partial x_2} \tilde{u}}{\tilde{y}^T \tilde{u}} & \dots & \frac{\tilde{y}^T \frac{\partial A(\tilde{x})}{\partial x_k} \tilde{u}}{\tilde{y}^T \tilde{u}} \end{bmatrix}^T$$

The computation effort involved is as following :

i, To solve  $\tilde{y}$  from ②



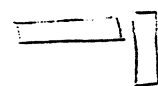
ii, To get  $\frac{\partial A(\tilde{x})}{\partial x_j}$  for  $j = 1, \dots, k$



iii, To multiply  $\tilde{y}^T \frac{\partial A}{\partial x_1} \tilde{u}$  k times

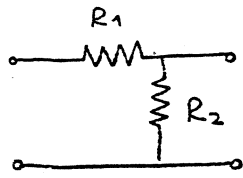


iv, To get the scalar  $\tilde{y}^T \tilde{u}$



v, to divide the numerator obtained in iii by the scalar.

PROBLEM # 8



Voltage divider

$$R_1 = 1.01$$

$$R_2 = 1.14$$

The constraints on these resistances are as follows

$$0.46 \leq \frac{R_2}{R_1 + R_2} \leq 0.53$$

$$1.85 \leq R_1 + R_2 \leq 2.15$$

We formulate four constraint functions out of these

$$f_1 = 1.85 - R_1 - R_2 \quad \rightarrow f_1(1.01, 1.14) = 1.85 - 1.01 - 1.14 = -0.3$$

$$f_2 = R_1 + R_2 - 2.15 \quad \rightarrow f_2(1.01, 1.14) = 1.01 + 1.14 - 2.15 = 0$$

$$f_3 = 0.46 R_1 - 0.54 R_2 \quad \rightarrow f_3(1.01, 1.14) = 0.46 \times 1.01 - 0.54 \times 1.14 = -0.151$$

$$f_4 = -0.53 R_1 + 0.47 R_2 \quad \rightarrow f_4(1.01, 1.14) = -0.53 \times 1.01 + 0.47 \times 1.14 = 0.0005$$

Let  $M \triangleq \max f_i \quad = f_4 = 0.0005$

Then

$$|M - f_1| = |0.0005 + 0.3| = 0.3005 > 0.01 \quad \text{not active}$$

$$|M - f_2| = |0.0005 - 0| = 0.0005 < 0.01 \quad \text{ACTIVE}$$

$$|M - f_3| = |0.0005 + 0.151| = 0.1515 > 0.01 \quad \text{not active}$$

$$|M - f_4| = |0.0005 - 0.0005| = 0 < 0.01 \quad \text{ACTIVE}$$

We have

$$Y = \{ i \mid f_i(\underline{R}) = \max_i f_i(\underline{R}) \quad i \in I \} \quad Y = \{ 2, 4 \}$$

from above

The first order changes

$$\Delta f_i(\underline{R}) = \nabla_{\underline{R}} f_i^T(\underline{R}) \Delta \underline{R} \quad i \in Y$$

For the steepest descent direction for  $\max_{i \in I} f_i(\underline{R})$   
the condition is

$$\nabla_{\underline{R}} f_i^T(\underline{R}) \Delta \underline{R} < 0 \quad i \in Y$$

Consider  $\Delta \underline{R} = - \sum_{i \in Y} \alpha_i \nabla_{\underline{R}} f_i(\underline{R})$

$$\sum_i \alpha_i = 1 \quad \text{and} \quad \alpha_i \geq 0$$

which suggests the linear program

$$\text{maximize } \alpha_{r+1} \geq 0$$

subject to

$$-\nabla_{\underline{R}} f_i^T(\underline{R}) \sum_{i \in Y} \alpha_i \nabla_{\underline{R}} f_i(\underline{R}) \leq -\alpha_{r+1} \quad i \in Y$$

$$\sum_{i \in Y} \alpha_i = 1$$

$$\alpha_i \geq 0 \quad i \in Y$$

changing the index  
of active f  $2 \rightarrow 1$   
 $4 \rightarrow 2$

The solution to the linear program provides  $\Delta \underline{R}$

$$\nabla_{\underline{R}} f_1(\underline{R}) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \nabla_{\underline{R}} f_2(\underline{R}) = \begin{bmatrix} -0.53 \\ 0.47 \end{bmatrix}$$

$$-\nabla_{\underline{R}} f_i^T(\underline{R}) \sum_i \alpha_i \nabla_{\underline{R}} f_i(\underline{R}) \leq -\alpha_3 \quad i=1,2$$

$$1) \quad -[1 \ 1] \left( \alpha_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} -0.53 \\ 0.47 \end{bmatrix} \right) \leq -\alpha_3$$

$$2) \quad -[-0.53 \ 0.47] \left( \alpha_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} -0.53 \\ 0.47 \end{bmatrix} \right) \leq -\alpha_3$$

$$\Rightarrow \quad \alpha_1 + \alpha_2 = 1 \quad \alpha_1, \alpha_2 \geq 0$$

Solving for  $\alpha_1, \alpha_2$

$\alpha_1 = 1 - \alpha_2$  and substitute in ① and ② equations

$$-2(1 - \alpha_2) + 0.06 \alpha_2 \leq -\alpha_3$$

$$0.06(1 - \alpha_2) - 0.5 \alpha_2 \leq -\alpha_3$$

which gives  $\alpha_2 = 0.786$

$\therefore \alpha_1 = 0.214$

and the descent direction is

$$\Delta \underline{R} = -(\alpha_1 \nabla_{\underline{R}} f_1 + \alpha_2 \nabla_{\underline{R}} f_2)$$

$$= -\left(0.214 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0.786 \begin{bmatrix} -0.53 \\ 0.47 \end{bmatrix}\right) = \begin{bmatrix} 0.20 \\ -0.58 \end{bmatrix}$$

As a check  $\nabla_{\underline{R}} f_i^T(\underline{R}) \Delta \underline{R} < 0$  for  $i=1, 2$

$$\nabla_{\underline{R}} f_1^T(\underline{R}) \Delta \underline{R} = [1 \quad 1] \begin{bmatrix} 0.20 \\ -0.58 \end{bmatrix} = -0.38 < 0$$

$$\nabla_{\underline{R}} f_2^T(\underline{R}) \Delta \underline{R} = [-0.53 \quad 0.47] \begin{bmatrix} 0.20 \\ -0.58 \end{bmatrix} = -0.386 < 0$$

**SECTION NINETEEN**  
**SOLUTION OF THE STATE EQUATIONS**

© J.W. Bandler 1984, 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.





## SOLUTION OF STATE EQUATIONS

### The Problem

We are given the equations of motion for a nonlinear dynamic system in the normal form

$$\dot{\tilde{x}} = \tilde{f}(\tilde{x}, t)$$

where

$$\tilde{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}, \quad \dot{\tilde{x}} \triangleq \frac{d}{dt} \tilde{x}$$

This system of first-order differential equations is to be solved numerically using difference methods, given the initial state vector

$$\tilde{x}(t_0) = \tilde{x}_0$$

for  $t \geq t_0$  over the time interval  $T$ .

### Definitions

The function  $\bar{x}(t)$  is a solution to this initial value problem if

$$\bar{x}(t_0) = \tilde{x}_0$$

$$\dot{\bar{x}}(t) = \tilde{f}(\bar{x}(t), t) \quad \forall t \in \{t_0, t_0+T\}$$

The solution time  $T$  is divided for computational purposes into increments

$$h_i \triangleq (\Delta t)_i$$

called the step size. We require

$$\bar{x}(t) \text{ at } t_k \triangleq t_0 + \sum_{i=1}^k h_i, k = 1, 2, \dots, N$$

where

$$t_N = t_0 + T$$

If  $\bar{x}(t_k)$  is the exact value and  $\bar{x}_k$  the computed value, then

$$\| \bar{x}(t_k) - \bar{x}_k \|$$

is the total error at  $t = t_k$ . This will depend on truncation error and roundoff error. The truncation error depends on the algorithm used and the roundoff error on the machine.

#### Local Error

The local error after one time step assuming  $\bar{x}$  is exact at the previous time step is

$$\| \bar{x}(t_1) - \bar{x}_1 \|$$

#### Local Truncation Error

The local error as above but due to the algorithm only is

$$\| \bar{x}(t_1) - \bar{x}_1 \|$$

Note:  $\bar{x}_k$  includes roundoff and truncation errors, whereas  $\bar{x}_k$  has no round off error.

#### Numerical Stability

An algorithm whose local roundoff error decays with an increasing

number of time steps is called numerically stable.

### Taylor Expansion Approach

Such algorithms are usually called Runge-Kutta algorithms.

### Polynomial Approximation Approach

Such algorithms are usually called numerical integration algorithms.

### Uniform Step Size

We will assume the uniform step

$$h = h_i$$

hence

$$t_n = t_0 + nh, n = 1, 2, \dots, N$$

### Taylor Algorithms

Since all formulas are easily extended to many equations in many state variables, we consider only one equation in one state variable.

Let  $\bar{x}(t)$  be the exact solution. Then,

$$\begin{aligned} \bar{x}(t_{n+1}) = & \bar{x}(t_n) + \frac{\bar{x}^{(1)}(t_n)}{1!} (t_{n+1} - t_n) \\ & + \frac{\bar{x}^{(2)}(t_n)}{2!} (t_{n+1} - t_n)^2 \\ & + \dots + \frac{\bar{x}^{(p)}(t_n)}{p!} (t_{n+1} - t_n)^p + \dots \end{aligned}$$

where

$$\bar{x}^{(j)} \triangleq \frac{d^j \bar{x}}{dt^j}$$

Let

$$t_{n+1} - t_n = h$$

then

$$\begin{aligned} \bar{x}(t_{n+1}) &= \bar{x}(t_n) + \frac{h}{1!} \bar{x}^{(1)}(t_n) + \frac{h^2}{2!} \bar{x}^{(2)}(t_n) + \frac{h^3}{3!} \bar{x}^{(3)}(t_n) \\ &+ \dots + \frac{h^p}{p!} \bar{x}^{(p)}(t_n) + \dots \end{aligned}$$

Hence

$$\begin{aligned} \bar{x}(t_{n+1}) &\approx \bar{x}(t_n) + \frac{h}{1!} f(\bar{x}(t_n), t_n) + \frac{h^2}{2!} f^{(1)}(\bar{x}(t_n), t_n) \\ &+ \dots + \frac{h^p}{p!} f^{(p-1)}(\bar{x}(t_n), t_n) \end{aligned}$$

To develop the algorithm, write

$$x_{n+1} = x_n + h T_p$$

where

$$T_p \triangleq T_p(x_n, t_n; h) \triangleq f(x_n, t_n) + \frac{h}{2!} f^{(1)}(x_n, t_n) + \dots + \frac{h^{p-1}}{p!} f^{(p-1)}(x_n, t_n)$$

Taylor: Order 1: Forward Euler

$p = 1$ , hence the Taylor algorithm reduces to

$$x_{n+1} = x_n + h f(x_n, t_n)$$

Exact solution:  $\bar{x}(t_{n+1})$  based on the initial condition  $x_n$  at  $t_n$ .

Approximate solution:  $x_{n+1} = x_n + h \left. \frac{dx}{dt} \right|_{t=t_n}$

Local truncation error:  $x_{n+1} - \bar{x}(t_{n+1})$

Conclusion: seldom used, truncation error too large.

Taylor: Order 2

$p = 2$ , hence

$$\begin{aligned}x_{n+1} &= x_n + h[f(x_n, t_n) + \frac{h}{2} f^{(1)}(x_n, t_n)] \\&= x_n + h[f(x_n, t_n) + \frac{h}{2}[\frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial t}](x_n, t_n)] \\&= x_n + h[f + \frac{h}{2}[\frac{\partial f}{\partial x} f + \frac{\partial f}{\partial t}]](x_n, t_n)\end{aligned}$$

Taylor: Order 3

$p = 3$ , hence

$$\begin{aligned}x_{n+1} &= x_n + h[f + \frac{h}{2}[\frac{\partial f}{\partial x} f + \frac{\partial f}{\partial t}]] + \frac{h^2}{3!} [\frac{d}{dt}(\frac{\partial f}{\partial x} f + \frac{\partial f}{\partial t})] \\&= x_n + \dots\end{aligned}$$

This formula is inconvenient, error-prone and requires explicit second-order partial derivatives.

Runge-Kutta Algorithm

The term

$$T_p \triangleq T_p(x_n, t_n; h)$$

which requires partial derivative evaluation, is replaced by

$$K_p \triangleq K_p(x_n, t_n; h)$$

such that

$$|K_p - T_p| \leq R h^p,$$

where R is a constant. The truncation error in the Taylor algorithm is of order  $h^p$ , hence the Runge-Kutta algorithm has the same order of magnitude of truncation error.

Runge-Kutta: Order 2

$$x_{n+1} = x_n + h K_2$$

where, for  $\alpha_2 \neq 0$

$$K_2 = (1 - \alpha_2) f(x_n, t_n) + \alpha_2 f\left[x_n + \frac{h}{2\alpha_2} f(x_n, t_n), t_n + \frac{h}{2\alpha_2}\right]$$

Heun's Algorithm (Modified Trapezoidal)

For  $\alpha_2 = 0.5$

$$x_{n+1} = x_n + \frac{h}{2} [f(x_n, t_n) + f[x_n + h f(x_n, t_n), t_n + h]]$$

Modified Euler-Cauchy Algorithm

For  $\alpha_2 = 1$

$$x_{n+1} = x_n + h f\left[x_n + \frac{h}{2} f(x_n, t_n), t_n + \frac{h}{2}\right]$$

Fourth-Order Runge-Kutta Algorithm

This algorithm is widely used for its accuracy and for its larger step size requirement h.

$$x_{n+1} = x_n + h K_4$$

where

$$K_4 = \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4]$$

$$k_1 \triangleq f(x_n, t_n)$$

$$k_2 \triangleq f\left[x_n + \frac{h}{2} k_1, t_n + \frac{h}{2}\right]$$

$$k_3 \triangleq f\left[x_n + \frac{h}{2} k_2, t_n + \frac{h}{2}\right]$$

$$k_4 \triangleq f[x_n + h k_3, t_n + h]$$

The procedure takes the weighted average of four separately calculated slopes  $k_1$ ,  $k_2$ ,  $k_3$  and  $k_4$ . This information is not required again, hence relatively a great amount of effort is required per iteration.

If  $f$  is independent of  $x$ , then we have the familiar Simpson's rule for integration.





**SECTION TWENTY**  
**EXAMPLES AND PROBLEMS**

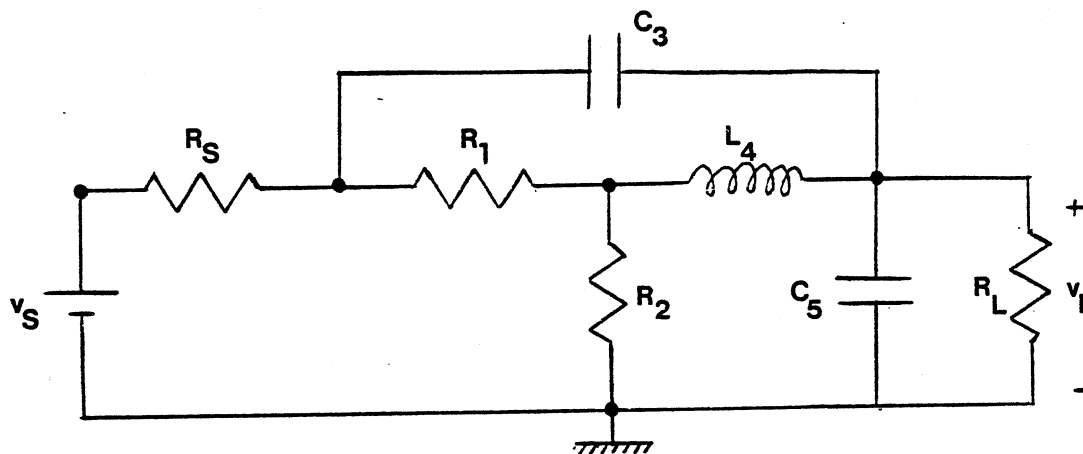
© J.W. Bandler 1988

This material is taken from previous years' assignments and examples. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.

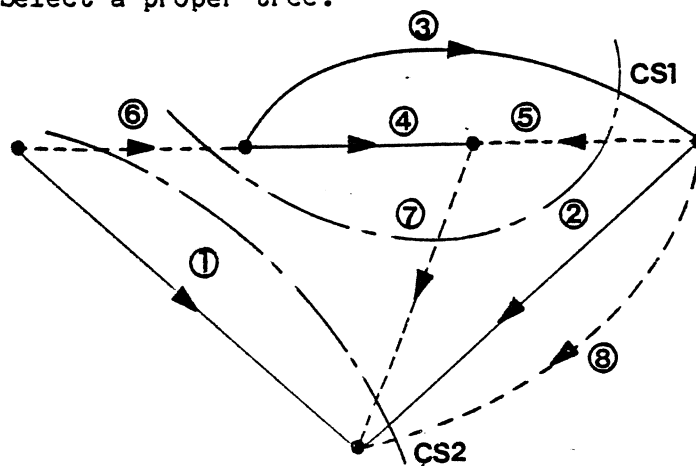


# IMPLEMENTATION OF FOURTH ORDER RUNGE-KUTTA ALGORITHM

(Assignment 5, April 1984)



Step 1 Select a proper tree.



Step 2 (a) Express the currents in the tree capacitors in terms of the link currents

$$i_2 + i_7 + i_8 - i_6 = 0 \quad (\text{CS2}) \quad (1a)$$

$$i_2 = C_5 \frac{dv_2}{dt} = i_6 - i_7 - i_8 \quad (1b)$$

$$i_3 - i_5 + i_7 - i_6 = 0 \quad (\text{CS1}) \quad (1c)$$

$$i_3 = C_3 \frac{dv_3}{dt} = i_6 + i_5 - i_7 \quad (1d)$$

- (b) Express the voltages of the link inductor in terms of the voltage of the tree branches

$$v_5 = v_4 - v_3 \quad (2a)$$

$$L_4 \frac{di_5}{dt} = v_4 - v_3 \quad (2b)$$

- Step 3 (a) Express the voltages across the tree resistors in terms of voltage sources, capacitor voltages, inductor currents and link currents.

$$v_4 = R_1 i_4 = R_1(i_7 - i_5) \quad (3a)$$

- (b) Express the currents in the link resistors in terms of voltage sources, capacitor voltages and inductor currents.

$$i_6 = G_S v_6 = G_S(v_S - v_2 - v_3) \quad (3b)$$

$$i_8 = G_L v_8 = G_L v_2 \quad (3c)$$

$$i_7 = G_2 v_7 = G_2(v_2 + v_3 - v_4)$$

$$i_7 = G_2(v_2 + v_3) - G_2 R_1(i_7 - i_5)$$

$$i_7 = \frac{G_2(v_2 + v_3) + G_2 R_1 i_5}{1 + G_2 R_1} \quad (3d)$$

- Step 4 Substitute in (1) and (2).

$$C_5 \frac{dv_2}{dt} = G_S(v_S - v_3 - v_2) - \frac{G_2(v_2 + v_3) + G_2 R_1 i_5}{1 + G_2 R_1} - G_L v_2 \quad (4a)$$

$$C_3 \frac{dv_3}{dt} = G_S(v_S - v_3 - v_2) + i_5 - \frac{G_2(v_2 + v_3) + G_2 R_1 i_5}{1 + G_2 R_1} \quad (4b)$$

$$L_4 \frac{di_5}{dt} = R_1 \frac{G_2(v_2 + v_3) + G_2 R_1 i_5}{1 + G_2 R_1} - R_1 i_5 - v_3 \quad (4c)$$

For

$$R_S = R_1 = R_2 = R_L = 1 \Omega$$

$$C_5 = C_3 = 1 \text{ F}$$

$$L_4 = 1 \text{ H}$$

the state equations become

$$\begin{bmatrix} \dot{v}_2 \\ \dot{v}_3 \\ \dot{i}_5 \end{bmatrix} = \begin{bmatrix} -\frac{5}{2} & -\frac{3}{2} & -\frac{1}{2} \\ -\frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} v_2 \\ v_3 \\ i_5 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} v_S \quad (5)$$

#### Problem

Write a Fortran program that implements the fourth-order Runge-Kutta algorithm to solve for the state variables  $v_2$ ,  $v_3$  and  $i_5$ . Take  $v_S = 1V$  and assume that the initial capacitor voltages and inductor current are equal to zero. Select a suitable time increment and a final time to highlight the details of the solution. Plot the results on graph paper.

The fourth-order Runge-Kutta algorithm is frequently used in solving the state equations of nonlinear networks.

The algorithm is expressed as

$$\tilde{x}_{n+1} = \tilde{x}_n + h \tilde{K}_4(\tilde{x}_n, t_n; h)$$

where

$$\tilde{K}_4 = [\tilde{k}_1 + 2\tilde{k}_2 + 2\tilde{k}_3 + \tilde{k}_4] / 6$$

$$\tilde{k}_1 \triangleq \tilde{f}(\tilde{x}_n, t_n)$$

$$\tilde{k}_2 \triangleq \tilde{f}(\tilde{x}_n + \frac{h}{2}\tilde{k}_1, t_n + \frac{h}{2})$$

$$\tilde{k}_3 \triangleq \tilde{f}(\tilde{x}_n + \frac{h}{2}\tilde{k}_2, t_n + \frac{h}{2})$$

$$\tilde{k}_4 \triangleq \tilde{f}(\tilde{x}_n + \frac{h}{2}\tilde{k}_3, t_n + \frac{h}{2}) .$$

Since the algorithm is of the order four, a relatively larger step could be chosen with a relatively small truncation error. However, the obvious disadvantage of the algorithm is that the function of interest must be evaluated

four times at each time step, and these function values are not utilized in any subsequent computation.

We write the state equations in matrix notation

$$\dot{\underline{x}} = \underline{A} \underline{x} + \underline{B} u .$$

These equations associated with the given problem are

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -2.5 & -1.5 & -0.5 \\ -1.5 & -1.5 & 0.5 \\ 0.5 & -0.5 & -0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} u$$

where

$$\underline{x} = [v_{C_5} \quad v_{C_3} \quad i_{L_4}]^T$$

$$u = V_S = 1V.$$

The initial states are

$$\underline{x} = \underline{0} .$$

To plot the results, the library routine PLOTPS is used. PLOTPS is an easy-to-use plotting program which can produce a variety of plots, each formatted according to one's requirements.

All PLOTPS requires is a data file containing data points and commands describing the type of plot desired.

Given a data file PLTDATA, a plot is generated by invoking the procedure PLOTPS which is maintained on the procedure file UTILITY. To obtain UTILITY

/GRAB, UTILITY.

Then, to obtain the plot on the VERSATEC plotter:

/BEGIN, PLOTPS, PLTDATA.

The plot of the results was made using the following commands besides the data points.



TITLE "EE3K4 ASSIGNMENT # 5"

TITLESIZE 0.3

LABELSIZE 0.25

XLABEL "TIME (SEC)"

YLABEL "STATE VARIABLES"

XFORMAT F4.1

YFORMAT F5.1

XMAX 20

YMAX 0.6

XMIN 0

YMIN -0.4

DELTA X 2.0

DELTA Y 0.1

C  
C  
C  
C  
C  
C  
C  
C  
C

PROGRAM EE3K4( INPUT, OUTPUT, TAPE6=OUTPUT)  
DIMENSION X(3), X1(3), XK1(3), XK2(3), XK3(3), XK4(3), RK(3)

000001  
000002  
000003  
000004  
000005  
000006  
000007  
000008  
000009  
000010  
000011  
000012

LIST OF MAIN VARIABLES

X VECTOR OF THE STATE VARIABLES  
H THE STEP SIZE  
N NUMBER OF STATE VARIABLES  
T THE TIME  
TMAX THE INTERVAL OF INTEGRATION  
XK1-XK4 COEFFICIENTS COMPUTED BY RUNGE-KUTTA METHOD

DATA X/3\*0.0/  
N=3  
T=0.  
TMAX=20.  
H=.2  
WRITE(6,50)

1 CONTINUE  
IF(T.GE.TMAX)GO TO 70  
CALL RUNKTA(X,H,T,N,X1,XK1,XK2,XK3,XK4,RK)  
WRITE(6,60)T,(X(I),I=1,3)  
T=T+H  
GO TO 1  
50 FORMAT(1H0,12X,"TIME",12X,"X(1)",12X,"X(2)",12X,"X(3)"/  
60 FORMAT(1H0,6X,4(F12.7,4X))  
70 STOP  
END

000013  
000014  
000015  
000016  
000017  
000018  
000019  
000020  
000021  
000022  
000023  
000024  
000025  
000026  
000027  
000028  
000029  
000030

C  
C

SUBROUTINE RUNKTA(X,H,T,N,X1,XK1,XK2,XK3,XK4,RK)  
DIMENSION X(N),X1(N),XK1(N),XK2(N),XK3(N),XK4(N),RK(N)

000031  
000032  
000033

C  
C  
C  
C

SUBROUTINE RUNKTA IMPLEMENTS THE FOURTH ORDER RUNGE-KUTTA  
METHOD FOR SOLVING ORDINARY DIFFERENTIAL EQUATIONS

CALL FUNCT(X,T,XK1)  
DO 10 I=1,N  
X1(I)=X(I)+XK1(I)\*H/2.  
10 CONTINUE  
T1=T+H/2.  
CALL FUNCT(X,T1,XK2)  
DO 20 I=1,N  
X1(I)=X(I)+XK2(I)\*H/2.  
20 CONTINUE  
T1=T+H/2.  
CALL FUNCT(X1,T1,XK3)  
DO 30 I=1,N  
X1(I)=X(I)+XK3(I)\*H.  
30 CONTINUE  
T1=T+H  
CALL FUNCT(X1,T1,XK4)  
DO 40 I=1,N  
RK(I)=(XK1(I)+2.\*XK2(I)+2.\*XK3(I)+XK4(I))/6.  
X(I)=X(I)+H\*RK(I)  
40 CONTINUE  
RETURN  
END

000034  
000035  
000036  
000037  
000038  
000039  
000040  
000041  
000042  
000043  
000044  
000045  
000046  
000047  
000048  
000049  
000050  
000051  
000052  
000053  
000054  
000055  
000056  
000057  
000058  
000059

C  
C

SUBROUTINE FUNCT(X,T,Y)  
DIMENSION X(3),Y(3),A(3,3),B(3)

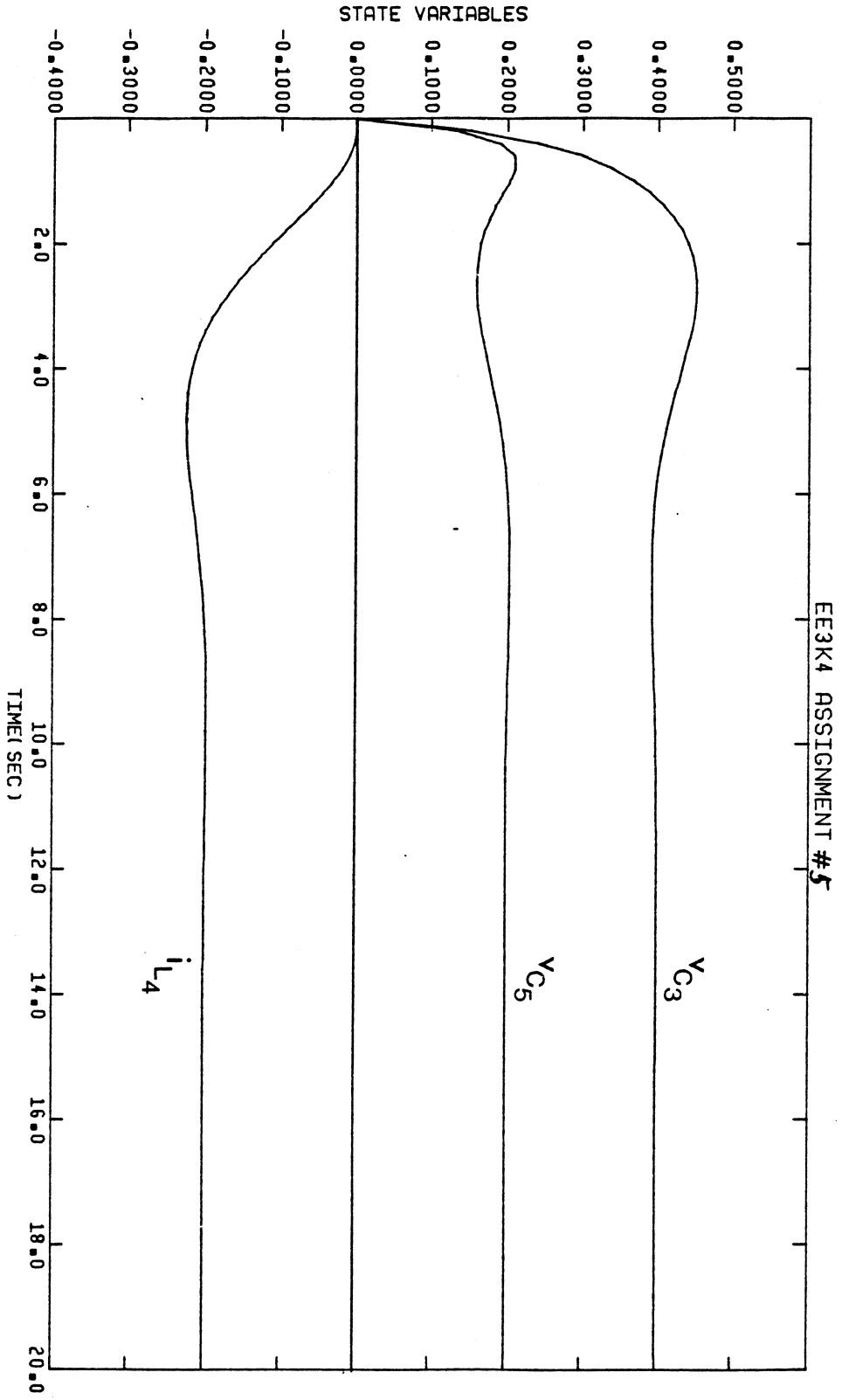
000060  
000061  
000062  
000063

C  
C  
C

SUBROUTINE FUNCT CALCULATES THE VALUE OF F(X,T) FOR CERTAIN X,T

DATA A/-2.5,-1.5,.5,-1.5,-1.5,-.5,-.5,.5,-.5/  
DATA B/1.,1.,0./  
VS=1.  
DO 10 I=1,3  
Y(I)=B(I)\*VS  
DO 10 J=1,3  
Y(I)=Y(I)+A(I,J)\*X(J)  
10 CONTINUE  
RETURN  
END

000064  
000065  
000066  
000067  
000068  
000069  
000070  
000071  
000072  
000073  
000074  
000075





**SECTION TWENTY-ONE**  
**COMPUTER-AIDED CIRCUIT OPTIMIZATION**



# 6

## Computer-Aided Circuit Optimization

---

J. W. Bandler  
McMaster University  
Hamilton, Ontario

---

This chapter deals with formulations and methods which can be implemented in the ever increasing number of situations when the classical synthesis approach, whether analytic or numerical, is inappropriate. When the so-called closed-form solution is, for some reason, out of the question, the modern approach is to use efficient, iterative, automatic optimization methods to achieve a design that meets or exceeds certain requirements. Not infrequently, exact methods may be used to great advantage in providing the initial feasible design for optimization.

In order to make the mathematics tractable, usable synthesis methods are usually restricted to ideal *commensurate* networks. As soon as we have to take into account active devices, a narrow range of element values, parasitic effects, high frequency operation, nonlinearities, frequency-dependent elements, *noncommensurate* elements (e.g., mixed lumped and distributed elements, uniformly distributed transmission lines with unequal or variable lengths, etc.), elements characterized by measurement data, response constraints, and so on, classical methods of design provide, at best, only approximate answers. In some cases these answers adequately approximate the solution to the actual design problem, but in many cases they do not.

The author is not advocating numerical methods for their own sake. Generally speaking, for the same job, iterative methods require more computation time than more specialized methods which do not require iteration (if they are available). Computing time is not, however, the only criterion an engineer has to consider. For example, in deciding whether or not to devote his own time to deriving an *analytic* algorithm, as distinct from a *numerical* algorithm, he also has to ask himself how often the algorithm would be used, how well it would represent real situations, how widely applicable it would be, and last but not least, how accurate the numerical results would be. After all, as engineers, we are ultimately working toward producing meaningful numbers as the solutions to realistic design problems.

Methods for *automating* the optimal design process will be emphasized. *Ad hoc* cut-and-try techniques using a general purpose analysis program are discouraged, particularly for filter design problems with anything other than the simplest of design specifications and a handful of variable parameters. The pitfalls are the same as with automated methods, the strategy for dealing with them is inevitably less sophisticated, and in the long run it will almost certainly cost more. It is desirable that the decision making process should, as far as possible, be left to the computer.

Poor or unacceptable results in computer-aided circuit optimization (or with any design process) are felt to be most likely due to bad preparation of the problem, a lack of understanding of the hazards that can be encountered, and the wrong choice of algorithm. This chapter, therefore, attempts to show how problems within the scope of filter design may be formulated effectively as optimization problems, to explain the differences between these formulations, to indicate appropriate optimization methods, and to indicate how the results might be interpreted. Details of optimization algorithms, proofs of convergence, etc., are beyond the scope of this work. Adequate references to the original papers and relevant text books will permit the reader to investigate these for himself.

Following a section on basic concepts which are essential for an understanding of optimization theory, a formulation and description of typical objectives and objective functions is presented. Constraints and some methods of dealing with them are discussed in fair detail in the next section, including the conditions for a constrained minimum. Minimax approximation, including conditions for a minimax optimum, is then dealt with. This is followed by sections on one-dimensional search methods, direct search methods, and methods using gradient information. Least  $p$ th approximation comes naturally after a discussion of gradient methods. A fairly long section is devoted to the adjoint network method of gradient evaluation.



### 6.1 BASIC CONCEPTS

The problem of optimization may be stated as follows. Minimize the scalar *objective function*\*  $U$  where

$$U \triangleq U(\phi) \quad (6.1)$$

subject to the *inequality constraints*

$$\mathbf{g}(\phi) \geq \mathbf{0} \quad (6.2)$$

and *equality constraints*

$$\mathbf{h}(\phi) = \mathbf{0}. \quad (6.3)$$

In (6.1) to (6.3),  $\phi$  is a vector of  $k$  independent variables or parameters,† thus

$$\phi \triangleq \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_k \end{bmatrix} \quad (6.4)$$

defining a  $k$ -dimensional space. In general, we might have  $m$  inequality constraints and  $s$  equality constraints so that

$$\mathbf{g}(\phi) \triangleq \begin{bmatrix} g_1(\phi) \\ g_2(\phi) \\ \vdots \\ g_m(\phi) \end{bmatrix} \quad (6.5)$$

and

$$\mathbf{h}(\phi) \triangleq \begin{bmatrix} h_1(\phi) \\ h_2(\phi) \\ \vdots \\ h_s(\phi) \end{bmatrix} \quad (6.6)$$

The *feasible region*  $R$  is defined by all vectors  $\phi$  satisfying (6.2) and (6.3). This may be written

$$R \triangleq \{\phi \mid \mathbf{g}(\phi) \geq \mathbf{0}, \mathbf{h}(\phi) = \mathbf{0}\}. \quad (6.7)$$

\* Also called *cost function*, *performance index*, or *error criterion*.

† Typically element values, residues, critical frequencies, etc.

$R$  is said to be *closed* if, as in (6.2), equalities are allowed. If no equalities are allowed it is said to be *open*. A *proper* minimum of  $U$  located by a vector  $\check{\phi}$  on the *response hypersurface* generated by  $U(\phi)$  is such that

$$\check{U} \triangleq U(\check{\phi}) < U(\phi) \quad (6.8)$$

for any feasible  $\phi$  close but not equal to  $\check{\phi}$ .\* Since we cannot generally guarantee to find a *global minimum*, we usually have to resign ourselves to a consideration of *local minima*. Our objective then is to find a feasible  $\check{\phi}$ , if it indeed exists, such that

$$U(\check{\phi}) = \min_{\phi \in R} U(\phi).$$

Figure 6-1 is an illustration of the problem in two dimensions, and it contains a number of features usually encountered in optimization problems. Note that only inequality constraints, i.e., constraints of the form of (6.2), are indicated.

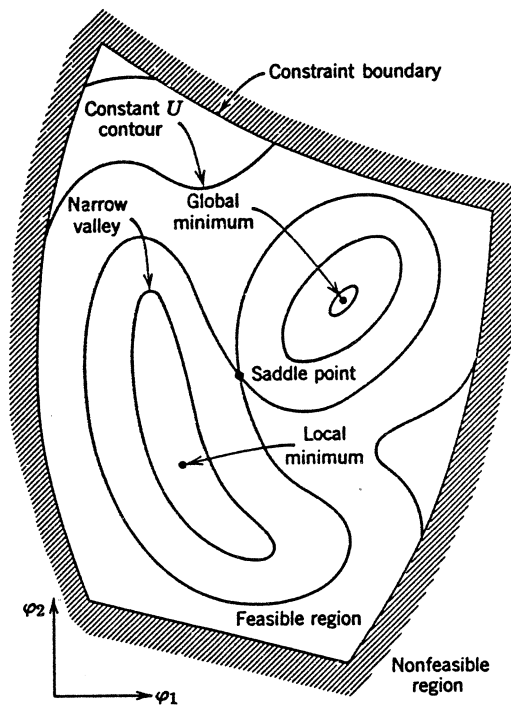


FIGURE 6-1 Some features encountered in optimization problems.

\*  $U(\check{\phi}) \leq U(\phi)$  can also define a minimum, but  $\check{\phi}$  may then be nonunique.

Examples of *unimodal*, *multimodal*, *strictly concave*, and *strictly convex* functions of one variable are shown in Figure 6-2. A unimodal function for our purposes is one having a unique optimum in the feasible region. It may or may not be continuous with continuous derivatives. A *strictly convex*

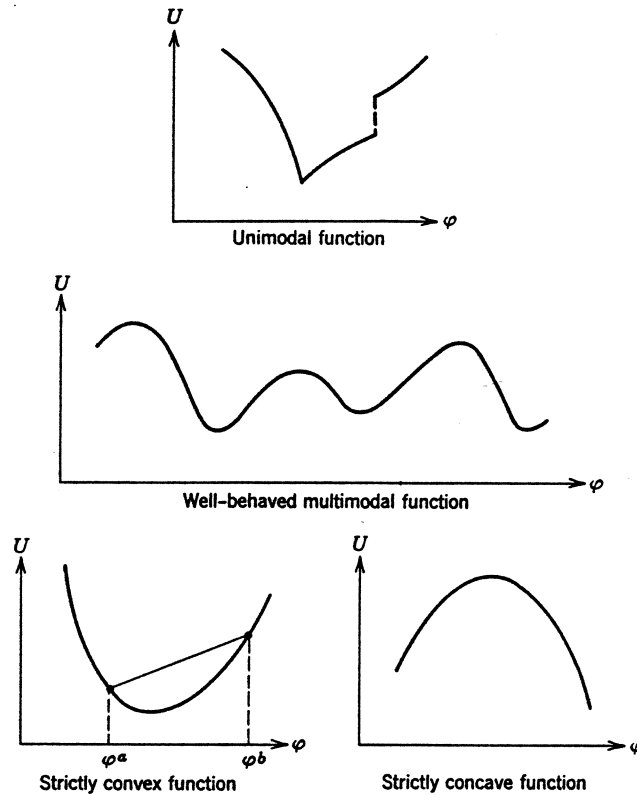


FIGURE 6-2 Functions of one variable.

function is one which can only be overestimated by a linear interpolation between two points on its surface. Thus, for  $\phi^a \neq \phi^b$ ,

$$U(\phi^a + \lambda(\phi^b - \phi^a)) < U(\phi^a) + \lambda(U(\phi^b) - U(\phi^a)) \quad (6.9)$$

$$0 < \lambda < 1$$

for a strictly convex function. See Figure 6-3. A *strictly concave* function is one whose negative is strictly convex. Note that if we omit strictly, then we imply that equality of the function and a linear interpolation can occur, i.e., (6.9) would have to admit equalities.

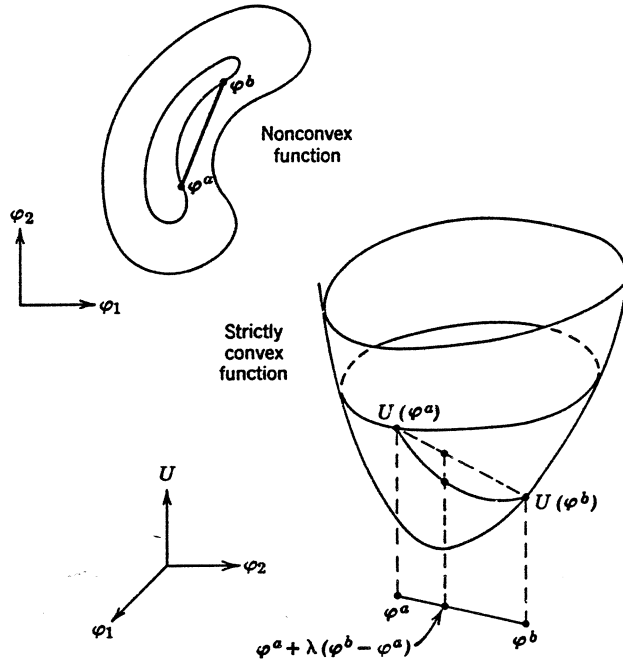


FIGURE 6-3 Illustration of convexity.

A region  $R$  is convex if for all  $\phi^a, \phi^b \in R$  all points

$$\begin{aligned} \phi &= \phi^a + \lambda(\phi^b - \phi^a) \\ 0 &\leq \lambda \leq 1 \end{aligned} \tag{6.10}$$

lie in  $R$ . Illustrations of convex and nonconvex regions are given in Figure 6-4.

The first three terms of a multidimensional Taylor series expansion of  $U(\phi)$  are given by

$$U(\phi + \Delta\phi) = U(\phi) + \nabla U^T \Delta\phi + \frac{1}{2} \Delta\phi^T H \Delta\phi + \dots \tag{6.11}$$

where the vector

$$\Delta\phi \triangleq \begin{bmatrix} \Delta\phi_1 \\ \Delta\phi_2 \\ \vdots \\ \Delta\phi_k \end{bmatrix} \tag{6.12}$$

contains  $k$  parameter increments,  $\Delta\phi^T$  is the transposed (row) vector,

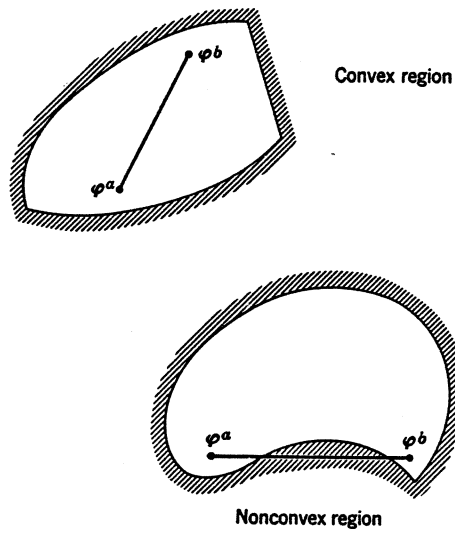


FIGURE 6-4 Convex and nonconvex regions.

$$\nabla U \triangleq \begin{bmatrix} \frac{\partial U}{\partial \phi_1} \\ \frac{\partial U}{\partial \phi_2} \\ \vdots \\ \frac{\partial U}{\partial \phi_k} \end{bmatrix} \quad (6.13)$$

is a vector containing the first partial derivatives of the objective function called the *gradient vector*, and

$$\mathbf{H} \triangleq \begin{bmatrix} \frac{\partial^2 U}{\partial \phi_1^2} & \frac{\partial^2 U}{\partial \phi_1 \partial \phi_2} & \cdots & \frac{\partial^2 U}{\partial \phi_1 \partial \phi_k} \\ \frac{\partial^2 U}{\partial \phi_2 \partial \phi_1} & \frac{\partial^2 U}{\partial \phi_2^2} & \cdots & \frac{\partial^2 U}{\partial \phi_2 \partial \phi_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 U}{\partial \phi_k \partial \phi_1} & \frac{\partial^2 U}{\partial \phi_k \partial \phi_2} & \cdots & \frac{\partial^2 U}{\partial \phi_k^2} \end{bmatrix} \quad (6.14)$$

is a symmetric  $k \times k$  matrix containing the second partial derivatives and called the *Hessian* matrix. At a minimum of a continuous function with continuous first and second partial derivatives  $\nabla U(\phi) = \mathbf{0}$  and  $H(\phi)$  is positive semidefinite.\* Invoking these conditions in (6.11) but with  $H$  taken as positive definite, it may be shown that (6.8) is satisfied, implying that we have a proper minimum.  $U(\phi)$  is strictly convex in a region where  $H$  is positive definite as may be seen by relating (6.9) with (6.11).

The problem formulated in (6.1) to (6.3) is called a *mathematical programming* problem. If all the functions are linear, we have *linear programming*; if not, we have *nonlinear programming*. The term *convex programming* is often used to describe the problem defined by (6.1) and (6.2) when  $U(\phi)$  is convex and  $g(\phi)$  is concave. Under these conditions  $R$  is convex, and  $\bar{U}$  is the global minimum.

In practical situations, it is usually out of the question to determine whether a specific problem falls into the domain of convex programming. Nevertheless, it seems a fair generalization to make, that the most reliable and efficient methods of optimization for practical problems are invariably those which invoke some of the nice properties of convex programming in their proofs of convergence. The better methods usually have built-in safeguards for dealing with the hazards of more general nonlinear programming problems while substantially retaining their desirable convergence features. Note that essentially unconstrained problems are regarded as special cases in the above discussion.

## 6.2 SOME OBJECTIVES AND OBJECTIVE FUNCTIONS

### Optimization by Solving Nonlinear Equations

Classically, to find  $\bar{U}$  we must in general solve  $k$  nonlinear equations in  $k$  unknowns, namely

$$\nabla U = \mathbf{0}.$$

Denoting this set of equations  $\mathbf{f}(\phi) = \mathbf{0}$  where

$$\mathbf{f}(\phi) \triangleq \begin{bmatrix} f_1(\phi) \\ f_2(\phi) \\ \vdots \\ f_k(\phi) \end{bmatrix}, \quad (6.15)$$

we could define a new objective function

$$U(\phi) = \mathbf{f}^T \mathbf{f} \quad (6.16)$$

\*It should be noted that  $H$  might not be positive definite, even in some cases when  $U(\phi)$  is strictly convex.

to be minimized. A minimum of value zero would imply that the solution to  $f(\phi) = 0$  had been found. Now, using a Taylor series expansion

$$f(\phi + \Delta\phi) = f(\phi) + J \Delta\phi + \dots \quad (6.17)$$

where

$$J \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial \phi_1} & \frac{\partial f_1}{\partial \phi_2} & \dots & \frac{\partial f_1}{\partial \phi_k} \\ \frac{\partial f_2}{\partial \phi_1} & \frac{\partial f_2}{\partial \phi_2} & \dots & \frac{\partial f_2}{\partial \phi_k} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_k}{\partial \phi_1} & \frac{\partial f_k}{\partial \phi_2} & \dots & \frac{\partial f_k}{\partial \phi_k} \end{bmatrix} \quad (6.18)$$

is a  $k \times k$  *Jacobian* matrix. The well-known *Newton-Raphson* method of solution is based on the hope that, if we evaluate  $f$  and  $J$  at  $\phi$ , then the incremental change

$$\Delta\phi = -J^{-1}f(\phi) \quad (6.19)$$

brings one closer to the solution. (In Section 6.8 these ideas are extended).

### Quadratic Objective Function

Consider the quadratic objective function

$$U(\phi) = \frac{1}{2}\phi^T A \phi + b^T \phi + c \quad (6.20)$$

where

- $A$  is a  $k \times k$  constant symmetric matrix,
- $b$  is a constant vector with  $k$  components,
- $c$  is a constant.

In this case, it is readily shown that

$$\begin{aligned} \nabla U &= A\phi + b \\ H &= A. \end{aligned}$$

A stationary point of  $U(\phi)$  can be found by solving the linear equations

$$A\phi + b = 0.$$

If  $A$  is nonsingular, the point is unique and can be found in a finite number of operations. The term *quadratic convergence* (Fletcher [1] prefers the term "property  $Q$ ") is used to describe the convergence properties of optimization methods, which guarantee to find the minimum of a quadratic function

in a finite number of steps. Such methods can be expected to be very efficient in minimizing functions adequately representable by positive-definite quadratic forms in the vicinity of a minimum. Some of them are discussed in later sections.

### Error Criteria

Most electrical network design problems can be formulated as approximation problems. Let us, therefore, introduce a weighted error or deviation between a specified function and an approximating function as

$$e(\phi, \psi) \triangleq w(\psi)[F(\phi, \psi) - S(\psi)] \quad (6.21)$$

where

- $S(\psi)$  is the real or complex specified function,
- $F(\phi, \psi)$  is the real or complex approximating function,
- $w(\psi)$  is a weighting function,
- $\psi$  is an independent variable,
- $\phi$  represents the adjustable parameters.

Thus  $F(\phi, \psi)$  may be a network response,  $S(\psi)$  may be the desired response, and  $\psi$  may be frequency or time. See Figure 6-5.

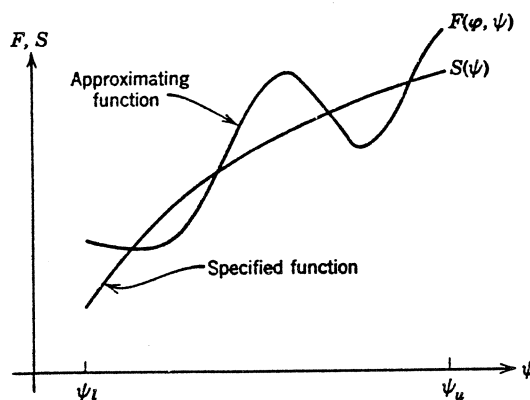


FIGURE 6-5 An approximation problem.

We may define a *norm*

$$\|e\|_p \triangleq \left\{ \int_{\psi_l}^{\psi_u} |e(\phi, \psi)|^p d\psi \right\}^{1/p}, \quad 1 \leq p \leq \infty \quad (6.22)$$



for the continuous case; and a *norm*

$$\|e\|_p \triangleq \left\{ \sum_i |e_i(\Phi)|^p \right\}^{1/p}, \quad \begin{array}{l} i \in I \\ 1 \leq p \leq \infty \end{array} \quad (6.23)$$

for the discrete case, where

$$e(\Phi) \triangleq \begin{bmatrix} e_1(\Phi) \\ e_2(\Phi) \\ \vdots \\ e_n(\Phi) \end{bmatrix}, \quad (6.24)$$

$$e_i(\Phi) \triangleq e(\Phi, \psi_i) = w(\psi_i)[F(\Phi, \psi_i) - S(\psi_i)] \quad (6.25)$$

and

$$I \triangleq \{1, 2, \dots, n\}. \quad (6.26)$$

Thus,  $I$  is an index set relating to discrete values of  $\psi$  on an interval  $[\psi_l, \psi_u]$ , which is closed and finite.

Now for well-behaved functions

$$\max_{[\psi_l, \psi_u]} |e(\Phi, \psi)| = \lim_{p \rightarrow \infty} \left\{ \frac{1}{\psi_u - \psi_l} \int_{\psi_l}^{\psi_u} |e(\Phi, \psi)|^p d\psi \right\}^{1/p} \quad (6.27)$$

when  $|e(\Phi, \psi)|$  is defined on  $[\psi_l, \psi_u]$ . If  $|e(\Phi, \psi)|$  is continuous on a finite interval  $[\psi_l, \psi_u]$ , then (6.27) is certainly valid. Similarly,

$$\max_i |e_i(\Phi)| = \lim_{p \rightarrow \infty} \left\{ \sum_i |e_i(\Phi)|^p \right\}^{1/p}, \quad i \in I. \quad (6.28)$$

Suppose we formulate an objective function as

$$U = \int_{\psi_l}^{\psi_u} |e(\Phi, \psi)|^p d\psi \quad (6.29)$$

for the continuous case and

$$U = \sum_{i \in I} |e_i(\Phi)|^p \quad (6.30)$$

for the discrete case. The minimization of the  $U$  of (6.29) or (6.30) is called *least pth approximation*. A minimum for the continuous case is called a best approximation with respect to  $\|e\|_p$ , defined in (6.22). A minimum for the discrete case is called a best approximation with respect to  $\|e\|_p$ , defined in (6.23). Now  $\|e\|_\infty$  and  $\|e\|_\infty$  are called *Chebyshev* or *uniform* norms. Because of the consequences of (6.27) and (6.28), minimization with respect to  $\|e\|_\infty$  or  $\|e\|_\infty$  is widely referred to as *minimax approximation*. Least  $p$ th approximation tends to minimax approximation as  $p \rightarrow \infty$ .

A word of caution concerning the weighting function  $w(\psi)$  and the index  $p$  is in order. Clearly their purpose is to emphasize or deemphasize the difference between  $F(\phi, \psi)$  and  $S(\psi)$ . Thus, an optimum with respect to one weighting function or value of  $p$  may not be an optimum with respect to another. Large errors will be emphasized by large values of  $p$ . If one knew in advance where these large errors would be,  $w(\psi)$  might also be used to emphasize them. The use of  $w(\psi)$  to do this is a poor approach, however, and should be discouraged.

### 6.3 CONSIDERATION OF CONSTRAINTS

It is rare to find any network design problem which is unconstrained. When physical considerations indicate that the optimum will lie in the interior of the feasible region, the designer is lucky and should take advantage of it. Often this will not be possible, and steps have to be taken to ensure that a realizable and practical design will be achieved. One of the great advantages of computer-aided circuit optimization is that, if the design problem has been properly formulated, a feasible design can always be achieved assuming the initial design is feasible.

Constraints in network design can take a variety of forms. They can include upper and lower bounds on parameters; they can include nonnegativity requirements on network elements. The topology, overall size, the suppression of unwanted modes of operation, considerations for parasitic effects whether reactive or lossy, and the stability of active devices can all result in constraints on parameters. Response constraints such as constraints on the phase while the amplitude is optimized can also occur.

Most network designers seem to treat constraints as an afterthought, and then complain that the optimization process gave them negative resistors, etc. Their faith in automated optimization methods is shattered as a result. The author would like to stress that a thorough consideration should be given to the constraints *before* the selection of an optimization strategy.

In this section we will look at some methods of converting constrained problems into essentially unconstrained ones. For other methods of nonlinear programming, the reader should refer elsewhere [2, 3, 4].

#### Transformations for Parameter Constraints

Various upper and lower bounds on the variable parameters are probably the most common kinds of constraints [5]. In Table 6-1 we show some simple parameter constraints falling into this class with appropriate transformations. It is useful to distinguish between constraints defining open and closed feasible regions. If the optimum is expected to lie away from the boundary or if it is desired to discourage the solution from getting too close, the former type might be chosen.

TABLE 6-1 Simple parameter constraints and transformations

Constraint	Transformation
$\phi_i \geq 0$	$\phi_i = \phi_i'^2$
$\phi_i > 0$	$\phi_i = \exp \phi_i'$
$\phi_i \geq \phi_{ii}$	$\phi_i = \phi_{ii} + \phi_i'^2$
$\phi_i > \phi_{ii}$	$\phi_i = \phi_{ii} + \exp \phi_i'$
$-1 \leq \phi_i \leq 1$	$\phi_i = \sin \phi_i'$
$0 \leq \phi_i \leq 1$	$\phi_i = \sin^2 \phi_i'$
$0 < \phi_i < 1$	$\phi_i = \frac{\exp \phi_i'}{1 + \exp \phi_i'}$
$\phi_{ii} \leq \phi_i \leq \phi_{ui}$	$\phi_i = \phi_{ii} + (\phi_{ui} - \phi_{ii})\sin^2 \phi_i'$
	$\phi_i = \frac{1}{2}(\phi_{ii} + \phi_{ui}) + \frac{1}{2}(\phi_{ui} - \phi_{ii})\sin \phi_i'$
$\phi_{ii} < \phi_i < \phi_{ui}$	$\phi_i = \phi_{ii} + (\phi_{ui} - \phi_{ii}) \frac{\exp \phi_i'}{1 + \exp \phi_i'}$
	$\phi_i = \phi_{ii} + \frac{1}{\pi}(\phi_{ui} - \phi_{ii})\cot^{-1} \phi_i'$ for $0 < \cot^{-1} \phi_i' < \pi$

**More General Considerations**

Parameter constraints of the form

$$\phi_{ii} \leq \phi_i \leq \phi_{ui} \quad (6.31)$$

can if necessary be written as

$$\begin{aligned} \phi_i - \phi_{ii} &\geq 0 \\ \phi_{ui} - \phi_i &\geq 0 \end{aligned} \quad (6.32)$$

in order to fit them into the scheme of (6.2). Frequency- or time-dependent constraints may be put into the form

$$c_j(\Phi, \psi) \geq 0 \quad (6.33)$$

where  $j$  denotes some  $j$ th function, or at discrete points on the  $\psi$ -axis into the form

$$c_j(\Phi, \psi_i) \geq 0 \quad (6.34)$$

where  $i$  denotes an  $i$ th sample point. The form of (6.34) is preferable to that of (6.33), since it allows us to consider a finite rather than an infinite number of constraints.

We might eliminate a number of constraints on physical or logical grounds if, for instance,

1.  $U(\phi) \rightarrow \infty$  as  $g_i(\phi) \rightarrow 0$ . The attenuation of a filter becomes infinite, for example, if a zero valued element short circuits the structure;
2. Some  $h_i(\phi) = 0$  can be explicitly written as  $\phi_j = f(\phi_1, \phi_2, \dots, \phi_{j-1}, \phi_{j+1}, \dots, \phi_k)$ . In this case we can optimize with  $k - 1$  parameters;
3.  $g_i(\phi)$  is known *a priori* to be positive.

Our design problem may be so complicated that we cannot easily find an initial design to serve as a feasible starting point in the optimization process. We could try to find one by unconstrained optimization by minimizing

$$-\sum_{i=1}^m w_i g_i(\phi) + \sum_{j=1}^s h_j^2(\phi) \quad w_i \begin{cases} = 0 & g_i(\phi) \geq 0 \\ > 0 & g_i(\phi) < 0. \end{cases} \quad (6.35)$$

If the minimum is zero we have a feasible point. Failure to converge to zero does not necessarily mean that a feasible point does not exist.

Having obtained a feasible starting point we might decide to simply reject nonfeasible points if they are obtained during optimization. Equivalently we might set  $U(\phi)$  to a most unattractive value if any violation occurs. Alternatively, we could add the term

$$\sum_{i=1}^m w_i g_i^2(\phi) + \sum_{j=1}^s h_j^2(\phi) \quad w_i \begin{cases} = 0 & g_i(\phi) \geq 0 \\ > 0 & g_i(\phi) < 0 \end{cases} \quad (6.36)$$

to the objective function. The objective function is not penalized as long as the constraints are satisfied. This procedure does not, unfortunately, always insure a strictly feasible solution.

The simple approaches just described have other disadvantages also. Discontinuities in the new function or its derivatives may be introduced. Steep walls or valleys may be formed at the boundary of the feasible region which can drastically slow down the optimization process. A method which simply rejects nonfeasible points can easily terminate at a false minimum [6].

### Sequential Unconstrained Minimization Techniques

One of the best known and most highly developed of the sequential unconstrained minimization techniques (SUMT) will be briefly outlined here [7]. Consider first the problem of minimization subject to inequality constraints defined in (6.1) and (6.2). Let

$$P(\phi, r) \triangleq U(\phi) + rG(\mathbf{g}) \quad (6.37)$$

where  $G(\mathbf{g})$  is continuous for  $\mathbf{g} > \mathbf{0}$  and  $G(\mathbf{g}) \rightarrow \infty$  for any  $g_i(\phi) \rightarrow 0$ , and where  $r > 0$ . Two possible candidates for  $G(\mathbf{g})$  immediately suggest themselves, namely

$$G(\mathbf{g}) = \sum_{i=1}^m \frac{1}{g_i(\phi)}, \quad (6.38)$$

and

$$G(\mathbf{g}) = - \sum_{i=1}^m \log g_i(\phi). \quad (6.39)$$

Let us denote the interior of the region  $R$  of feasible points by  $R^\circ$ , where

$$R^\circ \triangleq \{\phi \mid \mathbf{g}(\phi) > \mathbf{0}\} \quad (6.40)$$

and

$$R \triangleq \{\phi \mid \mathbf{g}(\phi) \geq \mathbf{0}\}. \quad (6.41)$$

The procedure is to select a  $\phi$  and a value of  $r$ , initially  $\phi^0 \in R^\circ$  and  $r_1 > 0$ , respectively, and minimize the function  $P$  of (6.37). The form of this equation is such that one would expect the minimum, namely  $\check{\phi}(r_1)$ , to lie in  $R^\circ$ . Repeat the procedure for different values of  $r$  such that

$$r_1 > r_2 > \cdots r_j > 0 \quad \text{and} \quad \lim_{j \rightarrow \infty} r_j = 0, \quad (6.42)$$

each minimization being started at the previous minimum. The minimization of  $P(\phi, r_2)$  would be started at  $\check{\phi}(r_1)$ , and so on.

The effect of the penalty is reduced every time the parameter  $r$  is reduced, so it is reasonable to expect that, under suitable conditions

$$\lim_{j \rightarrow \infty} \check{\phi}(r_j) = \check{\phi}$$

since by (6.42)

$$\lim_{j \rightarrow \infty} r_j = 0$$

so that

$$\lim_{j \rightarrow \infty} U[\check{\phi}(r_j)] = \check{U}$$

the constrained minimum. A minimum of  $P$  should always be available in  $R^\circ$ , so any nonfeasible point that may be encountered can be rejected. This safeguard should not be overlooked, for obvious reasons.

This procedure is termed an interior point unconstrained minimization technique, and it requires an initial  $\phi^0 \in R^\circ$ . If one is not available, the following approach may be adopted. Let

$$S \triangleq \{s | g_s(\phi) \leq 0, \quad s \in \{1, 2, \dots, m\}\}$$

$$T \triangleq \{t | g_t(\phi) > 0, \quad t \in \{1, 2, \dots, m\}\}.$$

Now define a

$$P(\phi, r) = -\sum_{s \in S} g_s(\phi) + r \sum_{t \in T} G_t(g_t(\phi)) \quad (6.43)$$

to be minimized for a sequence of  $r$  values satisfying (6.42). The implications of (6.43) are that any satisfied constraints are prevented from becoming violated while an attempt to satisfy the rest is being made. As soon as any constraint is satisfied the corresponding index is transferred from  $S$  to  $T$ , and the procedure repeated. When  $S$  becomes empty we have obtained a  $\phi^0 \in R^\circ$  and the solution process of the problem can commence.

To prove convergence one must invoke the requirements for convex programming (See Section 6.1). In practice, however, the conditions may be difficult to verify even if they hold. Nevertheless, the method should work successfully on a wide variety of practical problems for which convergence is not readily proved. Bad initial choices of  $r$  and  $\phi$  will slow down convergence. Too large a value of  $r_1$  may render the first few minima of  $P$  to be relatively independent of  $U$ , whereas too small a value may render the penalty ineffective except near the boundary where elongated valleys with steep sides are produced. Because of this and the fact that a sequence of unconstrained problems has to be solved, efficient gradient methods are generally required.

A reduction factor of 10 for the values of  $r$  is probably as good as any once the process has started. The arbitrariness of this can be somewhat alleviated by using the SUMT method without the  $r$  parameters [7, 8].

To include equality constraints the term

$$\frac{1}{r^{1/2}} \sum_{j=1}^s h_j^2(\phi) \quad (6.44)$$

can be added to the right hand side of (6.37). Clearly, as  $r \rightarrow 0$ ,  $h(\phi)$  must approach 0 or a minimum will not be reached.

The reader is referred to a number of selected references which discuss or extend SUMT [7, 8, 9, 10]. A lucid discussion is found in Chapter 5 of Kowalik and Osborne [8].

### Conditions for a Constrained Minimum

Necessary conditions which a stationary point  $\phi^\circ$  must satisfy in the problem of minimizing  $U(\phi)$  subject to  $g(\phi) \geq 0$  can be formulated. Assume  $U(\phi)$  and  $g(\phi)$  to be differentiable in the neighborhood of a feasible stationary point  $\phi^\circ$ , then

$$\nabla U(\phi^\circ) = \sum_{i=1}^m u_i \nabla g_i(\phi^\circ) \quad (6.45)$$

and

$$\mathbf{u}^T \mathbf{g}(\phi^\circ) = 0 \quad (6.46)$$

where

$$\mathbf{u} \triangleq \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \geq 0.$$

These necessary conditions can be interpreted as follows:  $\nabla U(\phi^\circ)$  is a nonnegative linear combination of the gradients  $\nabla g_i(\phi^\circ)$  of those constraints which are active\* at  $\phi^\circ$ .

Under the conditions of convex programming, i.e., if  $U(\phi)$  is convex,  $g(\phi)$  is concave, and  $R^\circ$  is nonempty, the conditions become sufficient for  $\phi^\circ$  to be  $\bar{\phi}$ , the constrained minimum. The relations (6.45) and (6.46) are called the Kuhn-Tucker relations [11]. An interpretation is sketched in Figure 6-6. Note that if we have been using a reliable optimization method, and if the relations are satisfied, we can be reasonably sure that a local minimum has been attained even if the convexity requirements are not met.

For a detailed treatment of the Kuhn-Tucker relations, including their derivation and a discussion of the constraint qualification which must also hold, the reader is referred to an appropriate book such as Zangwill [4].

\* A constraint  $g_i(\phi) \geq 0$  is active at  $\phi^\circ$  if  $g_i(\phi^\circ) = 0$ .

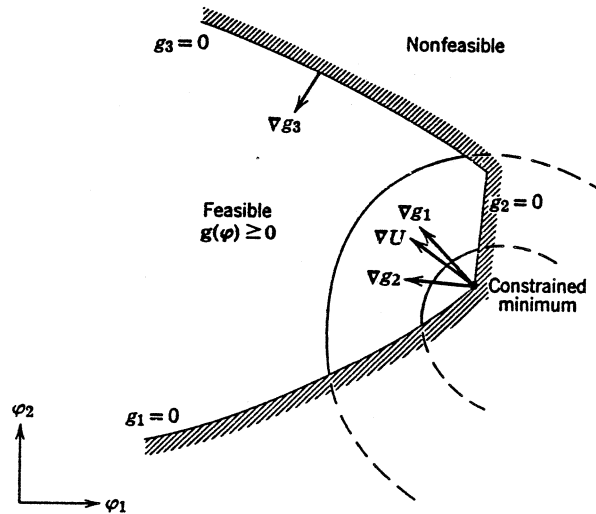


FIGURE 6-6 Sufficient conditions for a constrained minimum,  $u_1 > 0$ ,  $u_2 > 0$ ,  $u_3 = 0$ .

#### 6.4 MINIMAX APPROXIMATION

Classically, minimax approximation (See Section 6.2 for definitions) has implied the selection of the coefficients of a suitable polynomial or rational function so that it fits some desired specification (usually continuous on a closed interval) in an optimal equal-ripple manner. The Remez method and its generalizations are notable examples of iterative processes for obtaining best approximations using polynomials and rational functions.

The number of practical problems in filter design which can be solved by the classical approach is certainly diminishing in comparison with those that need solving, notwithstanding progress in transformations in the frequency variable, Richard's transformation for transmission-line networks, and so on. This section will, therefore, emphasize less specialized methods applicable to a wider range of practical design problems. Recent references are available which discuss in detail methods well-suited to polynomials and rational functions in the context of filter design [12, 13, 14, 15].

##### Formulation in Terms of Inequality Constraints

Figure 6-7 illustrates a typical filter design problem. We would like to find the (constrained) parameters of a suitable network so that certain passband and stopband specifications are met or exceeded. Assuming the approximat-



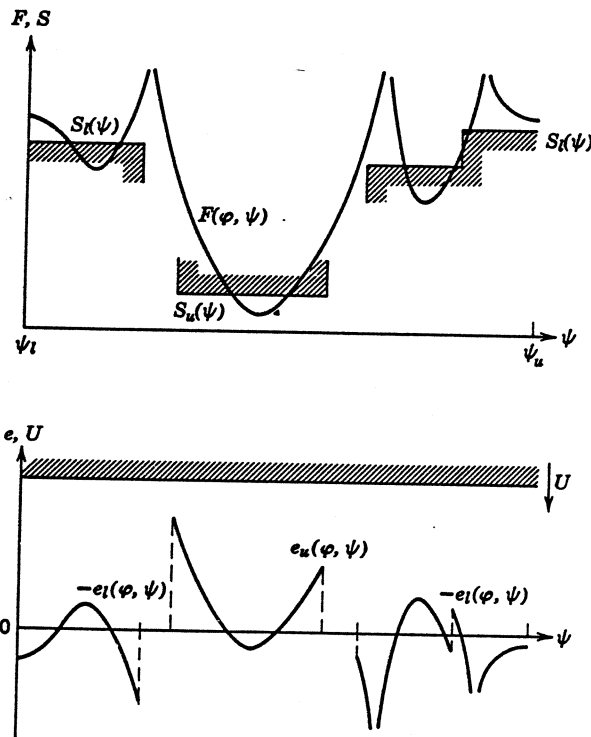


FIGURE 6-7 Typical filter design problem (the specifications are violated).

ing function and the specifications are real, let the error functions  $e_u$  and  $e_l$  be given by

$$\begin{aligned} e_u(\phi, \psi) &\triangleq w_u(\psi)[F(\phi, \psi) - S_u(\psi)] \\ e_l(\phi, \psi) &\triangleq w_l(\psi)[F(\phi, \psi) - S_l(\psi)] \end{aligned} \quad (6.47)$$

so that

$$\begin{aligned} e_{ui}(\phi) &\triangleq e_u(\phi, \psi_i), & i \in I_u \\ e_{li}(\phi) &\triangleq e_l(\phi, \psi_i), & i \in I_l. \end{aligned} \quad (6.48)$$

This is simply a generalization of (6.21), (6.25), and (6.26), where the symbols have the same meaning. In the present case of (6.47) and (6.48), the subscript  $u$  refers to the upper or *passband* specification, the  $l$  to the lower or *stopband* specification.

Since approximation in the time and other domains can also be formulated in these terms,  $\psi$  is used rather than frequency. Furthermore, the index sets  $I_u$  and  $I_l$  are not necessarily disjoint.

The optimization problem can now be specified as: minimize the quantity  $U$  subject to

$$\begin{aligned} U &\geq e_{ui}(\phi), & i \in I_u \\ U &\geq -e_{li}(\phi), & i \in I_l \end{aligned} \quad (6.49)$$

and also to all other constraints, such as on  $\phi$ . Observe that  $U$  is an *additional* independent variable. As shown in Figure 6-7 it may be visualized as a level or ceiling which is forced down on the deviations  $e_u$  and  $-e_l$ .

At a minimum at least one constraint in (6.49) must be an equality. Otherwise  $U$  can be lowered without violation. Further if

1.  $\check{U} < 0$ , the minimum amount by which the network response exceeds the specifications is maximized;
2.  $\check{U} > 0$ , the maximum amount by which the network response violates the specifications is minimized.

For loss or phase equalization, or time-domain approximation, for example, we might have only one specification, namely,  $S(\psi)$ . To treat these special cases we simply drop the subscripts  $u$  and  $l$  in (6.47) to (6.49) and the objective is equivalent to minimizing

$$U = \max_{i \in I} |e_i(\phi)|. \quad (6.50)$$

The weighting functions in (6.47) serve the following purpose. If one is much larger than the other, it emphasizes the deviation associated with it at the expense of the rest of the response if the specifications are violated. When the specifications are satisfied (we can now set the weighting function effectively to infinity if required), effort is switched to the rest of the response.

#### Methods for Minimax Approximation

An approach successfully implemented in optimal filter design [16] and reviewed by Waren, Lasdon, and Suchman [17] is to use sequential unconstrained minimization. We could, for example, define

$$\begin{aligned} P(\phi, U, r) = U + r \left\{ \sum_{i \in I_u} \frac{w_{ui}}{U - e_{ui}(\phi)} \right. \\ \left. + \sum_{i \in I_l} \frac{w_{li}}{U + e_{li}(\phi)} \right. \\ \left. + \text{other terms} \right\} \end{aligned} \quad (6.51)$$

where the other terms might include parameter constraints. Note that in (6.51) the elements of  $g(\phi)$  include

$$\begin{aligned} \frac{1}{w_{ui}} [U - e_{ui}(\phi)] &\geq 0, & i \in I_u \\ \frac{1}{w_{li}} [U + e_{li}(\phi)] &\geq 0, & i \in I_l. \end{aligned} \quad (6.52)$$

Further, it should be remembered that  $U$  is an independent variable. The appropriate formulations described in Section 6.3 are thus applicable to minimax approximation.

Ishizaki and Watanabe [18] have described a method in many respects similar to the more recent one by Osborne and Watson [19], which applies linear programming iteratively to achieve a best approximation in the minimax sense. Let us concern ourselves with the objective suggested by (6.50). This should not, however, be taken to imply that the method is less general than the one already outlined.

Linearizing  $e_i(\phi)$ , which is taken as real, at some point  $\phi^j$  the problem becomes one of minimizing  $U$  subject to

$$\begin{aligned} \frac{1}{w_i} [U - e_i(\phi^j) - \nabla e_i^T(\phi^j) \Delta\phi^j] &\geq 0 \\ \frac{1}{w_i} [U + e_i(\phi^j) + \nabla e_i^T(\phi^j) \Delta\phi^j] &\geq 0 \end{aligned} \quad i = 1, 2, \dots, n > k \quad (6.53)$$

and other (linearized) constraints. Noting that the variables for linear programming should all be nonnegative, and imposing a rather practical constraint that the elements of  $\phi$  should not change sign we have the linear program in  $x \triangleq [x_1 \ x_2 \ \dots \ x_{k+1}]^T$  such as to

$$\text{minimize } U = x_{k+1}$$

subject to

$$\pm \{e_i(\phi^j) + \nabla e_i^T(\phi^j) \begin{bmatrix} \phi_1^j x_1 - \phi_1^j \\ \phi_2^j x_2 - \phi_2^j \\ \vdots \\ \phi_k^j x_k - \phi_k^j \end{bmatrix}\} \leq x_{k+1}, \quad i = 1, 2, \dots, n > k \quad (6.54)$$

$$x \geq 0$$

where

$$x_i \triangleq \frac{\Delta\phi_i^j}{\phi_i^j} + 1, \quad i = 1, 2, \dots, k.$$

The solution produces a direction given by  $\Delta\phi^j$ . Next we find  $\alpha^j$  such that  $\max_i |e_i(\phi^j + \alpha^j \Delta\phi^j)|$  is a minimum, set  $\phi^{j+1} = \phi^j + \alpha^j \Delta\phi^j$  and repeat the process. For conditions for convergence the reader is referred to the original papers [18, 19]. Other linearized constraints can also be considered [14, 15]. Clearly such an approach is directly applicable to linear functions such as polynomials, for which  $k + 1$  equal extrema results at the optimum.

Bandler, Srinivasan, and Charalambous [20] have described a descent type of algorithm for minimax approximation which also employs linear programming. Basically, the algorithm attempts to find a locally optimal downhill direction for the problem of minimizing  $U$ , where

$$U = \max_{i \in I} f_i(\phi), \quad (6.55)$$

where the  $f_i(\phi)$  are real nonlinear differentiable functions generally. Linearizing  $f_i(\phi)$  and letting

$$J \triangleq \{i | f_i(\phi) = \max_i f_i(\phi), \quad i \in I\} \quad (6.56)$$

we can obtain, at some feasible point  $\phi^j$ , the first-order changes

$$\Delta f_i(\phi^j) = \nabla f_i^T(\phi^j) \Delta\phi^j, \quad i \in J. \quad (6.57)$$

In order for  $\Delta\phi^j$  to define a descent direction for  $\max_{i \in I} f_i(\phi)$  we must have

$$\nabla f_i^T(\phi^j) \Delta\phi^j < 0, \quad i \in J.$$

Consider

$$\Delta\phi^j = -\sum_{i \in J} \alpha_i^j \nabla f_i(\phi^j) \quad (6.58)$$

$$\sum_{i \in J} \alpha_i^j = 1 \quad (6.59)$$

$$\alpha_i^j \geq 0, \quad i \in J, \quad (6.60)$$

which suggests the linear program:

$$\text{maximize } \alpha_{r+1}^j \geq 0 \quad (6.61)$$

subject to

$$-\nabla f_i^T(\phi^j) \sum_{i \in J} \alpha_i^j \nabla f_i(\phi^j) \leq -\alpha_{r+1}^j, \quad i \in J \quad (6.62)$$

plus (6.59) and (6.60), where it is assumed that  $J$  has  $r$  elements.

Observe that  $J$  should be nonempty, and that if  $J$  has only one element, we obtain the steepest descent direction for the corresponding maximum of the  $f_i(\phi)$ . The solution to the linear program provides  $\Delta\phi^j$ . We then find  $\gamma^j$  corresponding to the minimum value of  $\max_{i \in I} f_i(\phi^j + \gamma^j \Delta\phi^j)$ .  $\phi^{j+1}$  is set to  $\phi^j + \gamma^j \Delta\phi^j$  and the procedure is repeated.

In practice, we will not have a set of  $f_i(\phi)$  identically equal to the maximum value. An appropriate tolerance must, therefore, be introduced into (6.56) and a more suitable selection procedure for the elements of  $J$  formulated. For further details the original paper should be consulted [20]. It can be proved that the algorithm will, if correctly implemented, converge to the minimax solution.

**Example 6-1.** Figure 6-8 shows an example of minimax approximation [21]. The objective was to find

$$\tilde{U} = \min_{\phi} \left\{ \max_{\{f_i, f_o\}} |\rho(\phi, f)| \right\}$$

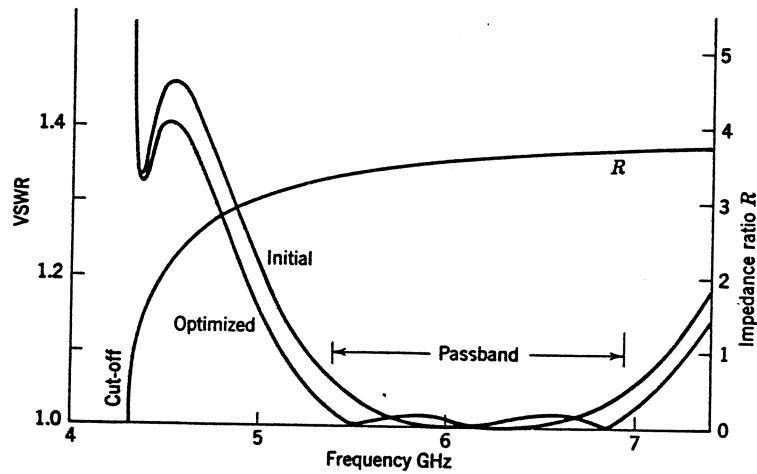
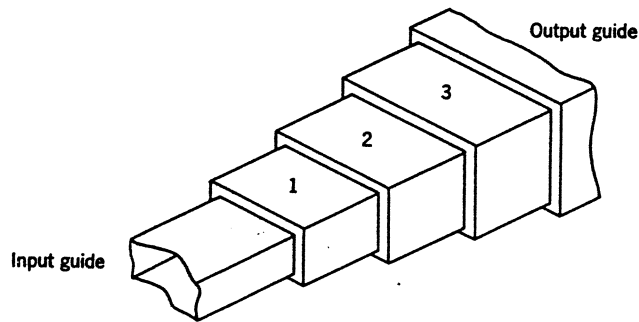


FIGURE 6-8 Example of constrained minimax approximation.

for the 3-section inhomogeneous rectangular waveguide impedance transformer, where  $\rho$  is the reflection coefficient, and  $f$  is frequency in GHz. The parameters  $\phi$  to be varied were the actual geometrical dimensions of the sections. The lower and upper band edges were  $f_l = 5.4$  GHz and  $f_u = 6.95$  GHz. It should be noted that (1) both input and output waveguides had different cut-off frequencies so that an exact synthesis was not possible, (2) severe constraints were placed on the parameters for a variety of physical reasons, (3) discontinuity susceptances could be taken directly into account, and (4) the razor search method [22] (See Section 6.6) was employed. The reader is referred to Bandler [21] for further details of this type of problem and for some other numerical results.

**Example 6-2.** Let us consider in a little more detail, the optimization of a seven-section cascaded transmission-line filter of the type shown in Figure 6-9. It is terminated at each end by

$$R_g(\omega) = R_L(\omega) = \frac{377}{\sqrt{1 - (f_c/f)^2}}$$

where  $f$  is frequency in GHz and  $f_c = 2.077$  GHz. The frequency variation of the terminations is thus like that of rectangular waveguides operating in the  $H_{10}$  mode with cut-off frequency 2.077 GHz. This interesting problem was

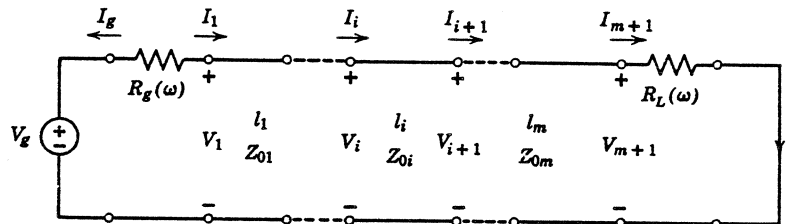


FIGURE 6-9 Cascaded transmission-line filter between frequency-variable resistors.

previously considered by Carlin and Gupta [23]. All section lengths were kept fixed at 1.5 cm so that the maximum stopband insertion loss would occur at about 5 GHz. The passband 2.16 to 3 GHz was selected, for which a maximum of 0.4 dB loss was specified. The solution obtained by the method of Carlin and Gupta was used as the initial design as shown by Figure 6-10.

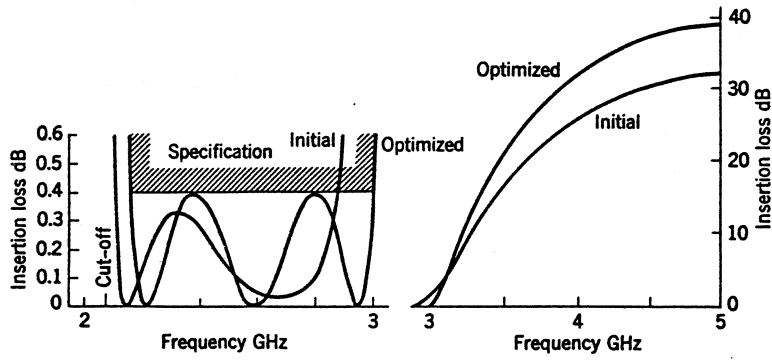


FIGURE 6-10 Comparison between the initial and optimized responses of the filter of Figure 6-9.

As optimized by Bandler and Lee-Chan [24], the problem was to minimize  $\max_i f_i(\phi)$  where

$$f_i(\phi) = \begin{cases} \frac{1}{2}[|\rho_i(\phi)|^2 - r^2] & \text{in the passband} \\ \frac{1}{2}[1 - |\rho_i(\phi)|^2] & \text{in the stopband} \end{cases}$$

$$\phi = \begin{bmatrix} Z_{01} \\ Z_{02} \\ \vdots \\ Z_{07} \end{bmatrix}$$

$r$  is the reflection coefficient magnitude corresponding to an insertion loss of 0.4 dB, and  $\rho_i(\phi)$  is the reflection coefficient of the filter at the  $i$ th frequency point. In particular, 22 uniformly spaced frequencies were selected from the passband and a single frequency, namely, 5 GHz for the stopband. The appropriately optimized response is shown in Figure 6-10. These results have also been reproduced by the method of Bandler, Srinivasan, and Charalambous [20], using

$$\phi = \begin{bmatrix} Z_{01} \\ Z_{02} \\ Z_{03} \\ Z_{04} \end{bmatrix}$$

and letting  $Z_{05} = Z_{03}$ ,  $Z_{06} = Z_{02}$ ,  $Z_{07} = Z_{01}$ .

The method used to analyze the filter at each frequency is suggested in Figure 6-9. A load current of 1 amp was assumed and a simple  $ABCD$  matrix analysis was carried out to find all the other voltage and current variables shown ( $V_g$  will, of course, be generally complex and frequency dependent in

this case). The appropriate partial derivatives were obtained from *one* such analysis per frequency point, using the adjoint network method (Section 6.9).

### Conditions for a Minimax Optimum

To derive some insight into the necessary conditions which a stationary point  $\phi^\circ$  must satisfy in a minimax approximation problem [25], let us reduce it to the form

$$\text{minimize } U = \phi_{k+1} \quad (6.63)$$

subject to constraints of the form

$$\phi_{k+1} \geq f_i(\phi), \quad i = 1, 2, \dots, m. \quad (6.64)$$

Rewriting the constraints as

$$g_i(\phi) \triangleq \phi_{k+1} - f_i(\phi) \geq 0, \quad i = 1, 2, \dots, m, \quad (6.65)$$

allows us to apply the Kuhn-Tucker relations (Section 6.3). Assuming  $U$  and the  $f_i(\phi)$  to be differentiable in the neighborhood of  $\phi^\circ$ , we have at  $\phi = \phi^\circ$

$$\begin{aligned} \left[ \begin{array}{c} \nabla U \\ \frac{\partial U}{\partial \phi_{k+1}} \end{array} \right] &= \sum_{i=1}^m u_i \left[ \begin{array}{c} \nabla \\ \frac{\partial}{\partial \phi_{k+1}} \end{array} \right] (\phi_{k+1} - f_i(\phi)) \\ \mathbf{u}^T \mathbf{g} &= 0, \end{aligned} \quad (6.66)$$

where  $\mathbf{u}$  is defined by (6.46). But

$$\begin{aligned} \nabla U &= \nabla \phi_{k+1} = \mathbf{0} \\ \frac{\partial U}{\partial \phi_{k+1}} &= 1 \\ \frac{\partial f_i(\phi)}{\partial \phi_{k+1}} &= 0 \end{aligned} \quad (6.67)$$

everywhere. Furthermore, at least one constraint must be an equality. For convenience, assume the first  $m_0$  constraints are equalities. Then

$$\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \sum_{i=1}^{m_0} u_i \begin{bmatrix} -\nabla f_i(\phi^\circ) \\ 1 \end{bmatrix} \quad (6.68)$$

since

$$u_i = 0, \quad i = m_0 + 1, m_0 + 2, \dots, m.$$

Alternatively, the necessary conditions may be written as



$$\begin{aligned} \sum_{i=1}^{m_0} u_i \nabla f_i(\phi^\circ) &= \mathbf{0} \\ \sum_{i=1}^{m_0} u_i &= 1 \\ u_i &\geq 0, \quad i = 1, 2, \dots, m_0. \end{aligned} \quad (6.69)$$

An interpretation of these relations is sketched in Figure 6-11. Under the conditions of convex programming, the  $f_i(\phi)$  would have to be convex, and the conditions become sufficient for  $\phi^\circ$  to be  $\phi$ , the minimax optimum. Often  $m_0$  will be equal to  $k + 1$ , but this is not a general requirement. The reader should observe the correspondence between (6.58) to (6.60) for  $\Delta\phi^j = \mathbf{0}$  with (6.69). More insight into these relations, in particular as they relate to filter problems, should be gained by referring to Bandler [25].

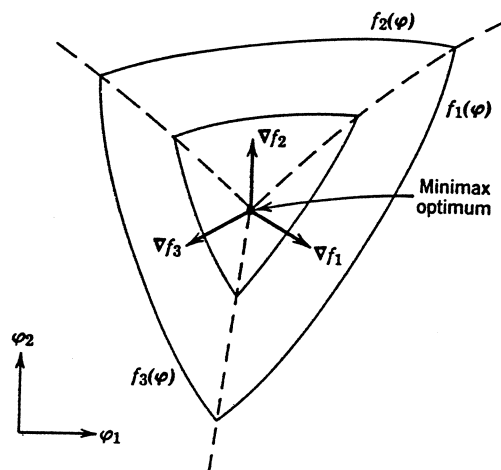


FIGURE 6-11 Sufficient conditions for a minimax optimum,  $u_1 > 0$ ,  $u_2 > 0$ ,  $u_3 > 0$ .

## 6.5 ONE-DIMENSIONAL SEARCH METHODS OF MINIMIZATION

Three main possible reasons spring to mind for investigating the optima of functions of one variable. The obvious one is that this might be the problem we are given. The second is that the multidimensional method we are using may call for a one-dimensional search for a minimum in some feasible downhill direction.\* The third is that we may be dealing with an approximation problem for which the extrema of the error function are required during an optimization process.

\*That is, in a feasible direction for which  $U$  is decreasing.

Powerful methods are available for functions known to be unimodal on an interval. We can broadly distinguish two classes, first the *elimination* methods which chop away subintervals not containing the optimum in an efficient manner with no assumptions except unimodality; second the *approximation* or *interpolation* methods which assume the function is smooth and well-represented by a low-order polynomial near the optimum.

Without loss of generality and to simplify discussions we will assume we have a function  $U$  of a single variable  $\phi$ .

**Elimination Methods**

At the start of the  $j$ th iteration of a search for a minimum of a unimodal function suppose we have an *interval of uncertainty*  $I^j$  where, referring to Figure 6-12,

$$I^j \triangleq u - l \tag{6.70}$$

with  $\phi_a^j = a$ ,  $\phi_l^j = l$ . Further, we have two interior points  $\phi_a^j = a$  and  $\phi_b^j = b$  at which we have evaluated the objective function. Let  $U(a)$  and  $U(b)$  be denoted  $U_a$  and  $U_b$ , respectively. Note that we take

$$l < a < b < u. \tag{6.71}$$

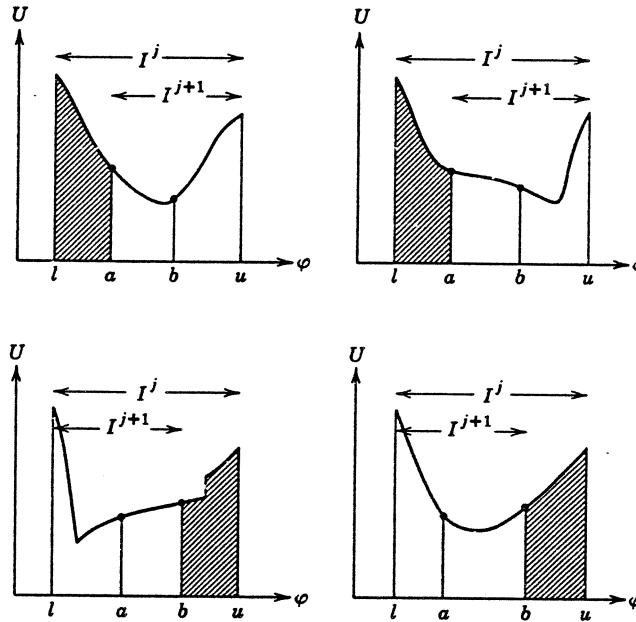


FIGURE 6-12 Reduction of interval of uncertainty.

Two conclusions can be drawn:

1. If  $U_a > U_b$ , the minimum lies in  $[a, u]$  and  $I^{j+1} = u - a$ .
2. If  $U_a < U_b$ , the minimum lies in  $[l, b]$  and  $I^{j+1} = b - l$ .

The difference between two well-known and efficient methods, the Fibonacci search and the Golden Section search, is in how these interior points are located. Let us discuss the slightly less efficient but simpler Golden Section search method. The reader is referred elsewhere for more detailed accounts of the various methods [6, 8, 26].

Whatever the outcome of comparing  $U_a$  and  $U_b$ , we want

$$I^{j+1} = u - a = b - l, \quad (6.72)$$

which is achieved by symmetrical placement of  $a$  and  $b$  on  $[l, u]$ . We want to minimize  $I^{j+1}$  and use one of the points in our new interval again which leads to

$$I^{j+2} = u - b = a - l. \quad (6.73)$$

Combining (6.70) to (6.73)

$$I^j = I^{j+1} + I^{j+2}. \quad (6.74)$$

To reduce the interval of uncertainty by a constant factor  $\tau$  at each iteration:

$$\frac{I^j}{I^{j+1}} = \frac{I^{j+1}}{I^{j+2}} = \tau. \quad (6.75)$$

Equations (6.74) and (6.75) lead to

$$\tau^2 = \tau + 1, \quad (6.76)$$

the solution of relevance being  $\tau = 1/2(1 + \sqrt{5}) \cong 1.618034$ . The division of a line according to (6.74) and (6.75) is called the Golden Section of a line.

At the  $j$ th iteration of this scheme

$$\begin{aligned} \phi_a^j &= \frac{1}{\tau^2} I^j + \phi_l^j \\ \phi_b^j &= \frac{1}{\tau} I^j + \phi_l^j \end{aligned} \quad j = 1, 2, 3, \dots \quad (6.77)$$

Note that each iteration except the first involves only one function evaluation due to symmetry. Depending on the outcome of the  $j$ th iteration, the appropriate quantities are set for the  $(j + 1)$ th iteration and the procedure repeated. After  $n$  function evaluations

$$\frac{I^1}{I^n} = \tau^{n-1}. \quad (6.78)$$

For a desired accuracy of  $\sigma$ ,  $n$  should be chosen such that

$$\tau^{n-2} < \frac{\phi_u^1 - \phi_l^1}{\sigma} \leq \tau^{n-1}. \quad (6.79)$$

It is readily shown that Golden Section provides an interval of uncertainty only about 17% greater than Fibonacci search for large  $n$ . The latter method also has the disadvantage that the number of function evaluations needs to be fixed in advance.

It is also possible to construct a scheme described by Temes [15] whereby the initial interval of uncertainty does not have to be fixed in advance. This scheme has been used with the method of Bandler, Srinivasan and Charalambous [20] (Section 6.4).

### Interpolation Methods

There are several interpolation methods, including quadratic and cubic, which are available [8, 26, 27, 28]. A rather straightforward method suggested by Davies, Swann, and Campey [8, 26] will be described here. The method does not require a unimodal interval containing the minimum to be known in advance, but the unimodality restriction should hold.

Evaluate  $U^i \triangleq U(\phi^0 + \alpha^i s)$  for

$$\begin{aligned} \alpha^0 &= 0 \\ \alpha^i &= \sum_{j=1}^i 2^{j-1} \delta, \quad i = 1, 2, \dots \end{aligned} \quad (6.80)$$

where  $s$  determines the negative gradient direction, i.e.,

$$s \triangleq \frac{-\left. \frac{\partial U}{\partial \phi} \right|_{\phi=\phi^0}}{\left| \left. \frac{\partial U}{\partial \phi} \right|_{\phi=\phi^0} \right|} \quad (6.81)$$

and  $\delta > 0$ , e.g., 1% of  $\phi^0$ , is a convenient increment. Thus  $\alpha^i$  is a positive step in the direction of decreasing  $U$ . When, for some  $i$ ,

$$U(\alpha^i) > U(\alpha^{i-1}), \quad (6.82)$$

evaluate  $U^{i+1}$  at

$$\alpha^{i+1} = \alpha^{i-1} + (\alpha^{i-1} - \alpha^{i-2}). \quad (6.83)$$

It should be clear that we now have four uniformly spaced points on the  $\alpha$  axis, namely,  $\alpha^{i-2}$ ,  $\alpha^{i-1}$ ,  $\alpha^{i+1}$ , and  $\alpha^i$  in order of increasing  $\alpha$ . Note that  $i \geq 2$ .

$$\begin{aligned} \text{If } U(\alpha^{i+1}) < U(\alpha^{i-1}), & \text{ let } a = \alpha^{i-1}, b = \alpha^{i+1}, c = \alpha^i. \\ \text{If } U(\alpha^{i+1}) > U(\alpha^{i-1}), & \text{ let } a = \alpha^{i-2}, b = \alpha^{i-1}, c = \alpha^{i+1}. \end{aligned} \quad (6.84)$$

It is easily shown that the minimum of a quadratic fitted at  $a$ ,  $b$ , and  $c$  is at

$$\alpha_{\min} = b + \frac{(b-a)(U_a - U_c)}{2(U_a - 2U_b + U_c)}. \quad (6.85)$$

Evaluation of  $U$  at  $\alpha_{\min}$  gives the estimate of the minimum and completes one stage of the method. A new stage with reduced  $\delta$  can be started at  $b$  or  $\alpha_{\min}$ , whichever corresponds to a smaller  $U$ .

## 6.6 DIRECT SEARCH METHODS OF MINIMIZATION

*Direct search* methods as interpreted by this author are methods which do not depend explicitly on evaluation or estimation of the gradient vector of the objective function. Such methods have enjoyed fairly wide use in network optimization [6, 21, 22, 29]. To what extent they will remain competitive, however, in the light of currently available methods of evaluating derivatives (See Section 6.9), remains to be seen.

One of the simplest methods is the one-at-a-time method. As Figure 6-13 shows, this process basically consists of letting one parameter vary until no improvement is obtained, and then another one, and so on. Progress is fairly slow on valleys not oriented in the direction of any coordinate axis.

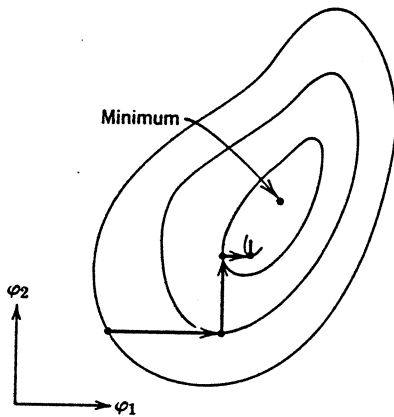


FIGURE 6-13 One-at-a-time search.

Obviously we need to consider more efficient methods. Two widely used methods will be reviewed, namely the pattern search method of Hooke and Jeeves [30] and the simplex method of Nelder and Mead [31]. Other well-known methods are Rosenbrock's method [32], the Powell-Zangwill method [28, 33], and the method of Davies, Swann, and Campey [26]. These methods are discussed in some of the general references [1, 6, 8, 26, 34].

### Pattern Search

An advantage the *pattern search* method has over the one-at-a-time method is that it attempts to detect the presence of a valley and align a direction of search along it. The tactics employed by pattern search will be explained by means of the example shown in Figure 6-14.

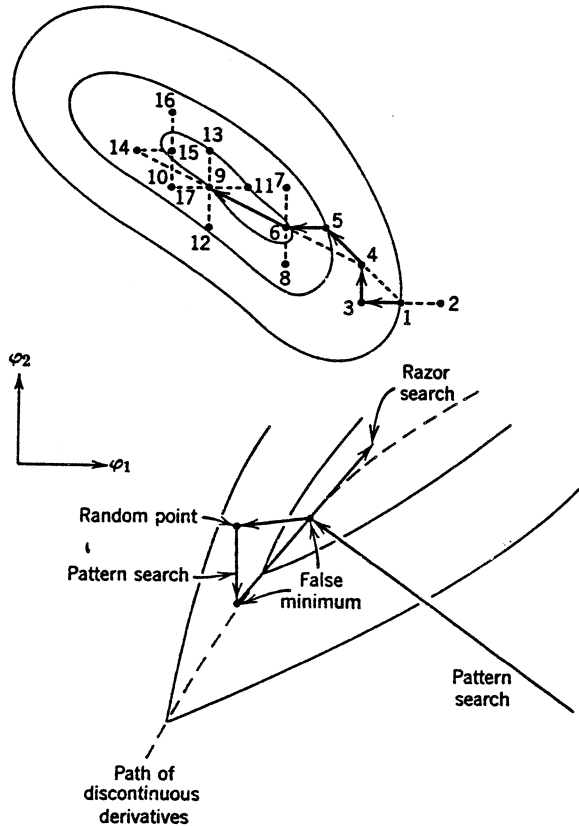


FIGURE 6-14 Following valleys by pattern search and razor search.

The first *base point*  $b^1$  is taken as the starting point  $\phi^1$ . A series of *exploratory moves* from  $\phi^1$  is initiated to find the second base point. In the example,  $\phi_1$  is incremented leading us to  $\phi^2$ . Now  $U^2 > U^1$  so  $\phi^2$  is rejected, and  $\phi_1$  is incremented in the opposite direction to  $\phi^3$ . Exploration with  $\phi_1$  is over.  $U^3 < U^1$  so  $\phi^3$  is retained and exploration with  $\phi_2$  begins.  $U^4 < U^3$  so  $\phi^4$  is retained in place of  $\phi^3$ . The first set of exploratory moves is complete, and so  $\phi^4$  becomes the second base point  $b^2$ . In the expectation that our

success would be repeated we make a *pattern move* to  $\phi^5 = 2b^2 - b^1$ , which is in the direction  $b^2 - b^1$ . By another set of exploratory moves we try to find the most promising point in the vicinity of  $\phi^5$ . Here, this point is  $\phi^6$  which becomes the third base point  $b^3$ , since  $U^6 < U^4$ . The search continues with a pattern move in the direction  $b^3 - b^2$  to  $\phi^9$ .

The pattern direction is destroyed when a pattern move followed by exploration fails, as around  $\phi^{14}$ . The strategy is to return to the previous base point. If the exploratory moves around the base point fail, as around  $\phi^9$ , the parameter increments are reduced and the procedure is restarted at that point. The search may be terminated either when the parameter increments fall below prescribed levels or the number of function evaluations or running time have reached upper limits.

The *razor search* method of Bandler and Macdonald [22] is a development of pattern search suited to direct optimization in the minimax sense without using derivatives. The name was suggested by the fact that "razor sharp" valleys are, in general, generated by an attempt to minimize functions of the form of (6.50). Paths of discontinuous derivatives are found along the bottom of such valleys, as indicated in Figures 6-11 and 6-14.

An investigation of the behavior of pattern search in the optimization of cascaded noncommensurate transmission lines acting as impedance transformers between resistive terminations was carried out [29]. It was observed that pattern search failed only when a sharp valley whose contours lay entirely within a quadrant of the coordinate axes was encountered. In that case no improvement was possible by searching parallel to these axes.

The razor search method makes a random move from a point where pattern search fails (assuming a false minimum) and uses pattern search to return to the path of discontinuous derivatives. (See Figure 6-14.) When pattern search fails again, an attempt is made to establish a pattern in the apparent downhill direction and resume with pattern search. The results shown in Figure 6-8 were produced by the razor search method [21].

An observation worth making here is that manual network optimization in the minimax sense, using an interactive system and employing, say, the one-at-a-time method, can easily terminate at a false minimum. A false minimum in the present context is a point representing a possibly equal-ripple response but which is not a local optimum in the minimax sense.

### The Simplex Method

In simplex methods of nonlinear optimization, the objective function is evaluated at the  $k + 1$  vertices of a *simplex* in  $k$ -dimensional space. In two dimensions, for example, we would have a triangle, for three dimensions a tetrahedron. An attempt is then made to replace the point with the greatest objective function value by another point.

A method having very desirable valley-following properties is the one due to Nelder and Mead [31]. The basic move is to *reflect* the point having the greatest function value in the centroid of the simplex formed by the remaining points. If the reflected point results in a function value lower than the current lowest, an *expansion* is attempted. Otherwise the point is retained if it results in a function value lower than the second highest. *Contraction* is attempted if reflection fails. Finally, *shrinking* of the simplex about the vertex corresponding to the lowest function value occurs following an unsuccessful attempt at contraction. Some of these moves are illustrated in Figure 6-15.

An example of the simplex strategy is shown in Figure 6-16. Observe that  $\phi^4$ ,  $\phi^6$ ,  $\phi^8$ ,  $\phi^9$ , and  $\phi^{10}$  have resulted from reflection;  $\phi^5$  from expansion; and  $\phi^7$  and  $\phi^{11}$  from contraction. The reader should follow the strategy through carefully to ensure his understanding of it. Its desirable valley-following properties result from its ability to align elongated simplexes in the

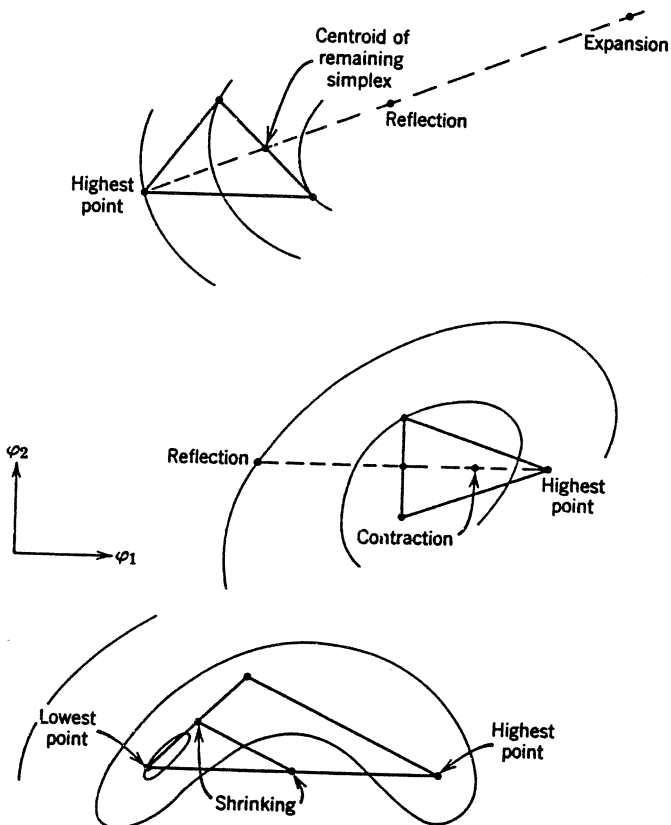


FIGURE 6-15 Examples of moves made by the simplex method.



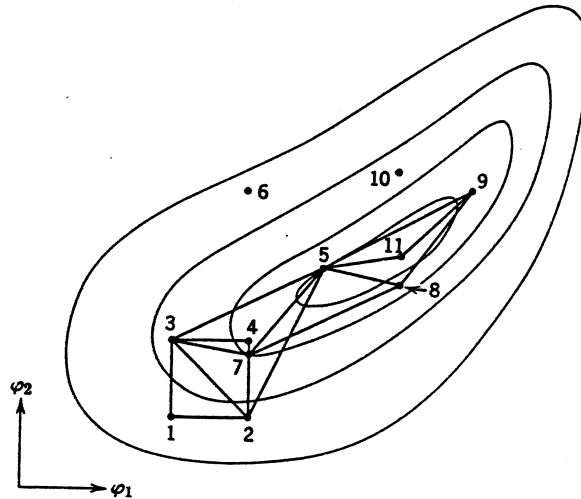


FIGURE 6-16 Optimization by the simplex method.

direction of the valleys. In particular, repeated success, for example, if a long straight valley is being followed, tends to increase the size of the moves, whereas repeated failure, for example, if a bend in the valley is encountered, tends to cause a decrease in the size of the moves.

It has been claimed to the author on a number of occasions that, unlike some other direct search methods, the simplex method can be successfully employed for minimax approximation. In the author's experience the simplex method is no less infallible than pattern search, for example. The principal fallacy in the argument is the assumption that, if the method requires no derivative information, it can necessarily handle problems with discontinuous derivatives.

## 6.7 GRADIENT METHODS OF MINIMIZATION

We turn our attention now to a class of minimization methods which require derivatives. By and large the most efficient algorithms currently available rely on evaluation of the gradient vector [1, 5, 6, 8, 14, 15, 26, 27, 35].

### Steepest Descent

At the  $j$ th iteration of most gradient methods, we proceed to

$$\phi^{j+1} = \phi^j + \alpha^j s^j \quad (6.86)$$

where  $s^j$  is (hopefully) a downhill direction of search and  $\alpha^j > 0$  is a scale factor chosen to minimize  $U(\phi^j + \alpha^j s^j)$ . One-dimensional minimization methods suitable for this purpose were discussed in Section 6.5.

The most obvious choice for  $s^j$  is the *steepest descent* direction at  $\phi^j$ , defined as follows. Referring back to (6.11), we note that a first-order change in the objective function is given by

$$\Delta U = \nabla U^T \Delta \phi. \quad (6.87)$$

If  $\Delta \phi = \alpha s$ , where  $\alpha > 0$  is fixed and  $\|s\| = 1^*$ , then it is easy to show that the  $s$  minimizing  $\Delta U$  is

$$s = -\frac{\nabla U}{\|\nabla U\|} \quad (6.88)$$

The  $s$  in (6.88) is the negative of the *normalized gradient* vector. Although  $-\nabla U/\|\nabla U\|$  provides the greatest local change, success of the steepest descent method is highly dependent on scaling. As Figure 6-17 shows, the

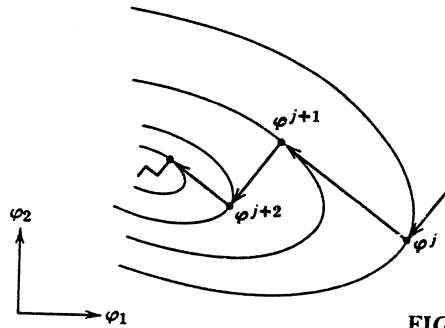


FIGURE 6-17 A steepest-descent strategy.

first few iterations may give good reduction in  $U$ , but subsequently the method usually deteriorates rapidly into oscillations, and progress becomes very slow.

### The Newton Method

This method was already mentioned in Section 6.2 in the context of solution of nonlinear equations. Differentiating the Taylor series (6.11)

$$\nabla U(\phi + \Delta \phi) = \nabla U(\phi) + H \Delta \phi + \dots \quad (6.89)$$

For  $\phi + \Delta \phi$  to be the minimizing point  $\check{\phi}$ ,  $\nabla U(\phi + \Delta \phi)$  should be 0 so that, neglecting higher-order terms,

$$\Delta \phi = -H^{-1} \nabla U. \quad (6.90)$$

\* The expression  $\|\cdot\|$  is the *Euclidean* norm. It has the form of (6.23) with  $p = 2$ .

This incremental change takes us to the minimum in only one iteration if we are dealing with a quadratic function (Section 6.2). It is instructive to compare (6.90) with (6.19).

When  $U$  is not quadratic, we could try the iterative scheme

$$\phi^{j+1} = \phi^j - \mathbf{H}^{-1} \nabla U^j \quad (6.91)$$

where  $\mathbf{H}^{-1}$  is the inverse of the Hessian matrix at the  $j$ th iteration. This scheme has, however, several disadvantages.  $\mathbf{H}$  must be positive definite otherwise divergence could occur. In particular,  $-\mathbf{H}^{-1} \nabla U^j$  might not point downhill. To counteract these possibilities, the modification

$$\phi^{j+1} = \phi^j - \alpha^j \mathbf{H}^{-1} \nabla U^j \quad (6.92)$$

can be employed where  $\alpha^j$  is chosen to minimize  $U^{j+1}$ . This might also be ineffective:  $\alpha^j$  may have to be negative;  $\mathbf{H}$  may be locally singular. Finally, the computation of  $\mathbf{H}$  and its inverse are time consuming.

### Conjugate Directions

Certain gradient methods which exploit the properties of *conjugate directions* associated with quadratic functions and do not explicitly evaluate  $\mathbf{H}$  or its inverse are highly effective. Before discussing them let us define conjugate directions.

The directions  $\mathbf{u}_i$  and  $\mathbf{u}_j$  are said to be conjugate with respect to a positive definite matrix  $\mathbf{A}$  if

$$\mathbf{u}_i^T \mathbf{A} \mathbf{u}_j = 0, \quad i \neq j. \quad (6.93)$$

In Figure 6-18, a two-dimensional interpretation of conjugate directions is given. Methods which generate such directions will minimize a quadratic

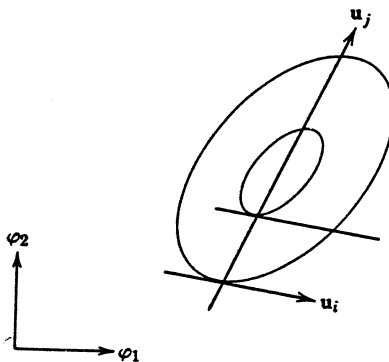


FIGURE 6-18 An illustration of two conjugate directions.

function in a finite number of iterations. It is evident that one linear minimization along each direction in turn locates the minimum.

Three well-known methods which use conjugate directions are the *conjugate gradient* method described by Fletcher and Reeves [35], the Fletcher-Powell-Davidon method [27], and the Powell-Zangwill method [28, 33] which does not require derivatives (See also references [1, 8]).

### The Conjugate Gradient Method

The direction of search  $s^j$  is given by [35]

$$s^j = -\nabla U^j + \beta^j s^{j-1} \quad (6.94)$$

where

$$\beta^j = \frac{(\nabla U^j)^T \nabla U^j}{(\nabla U^{j-1})^T \nabla U^{j-1}}, \quad (6.95)$$

and, initially,  $\beta^0 = 0$ . Thus the first iteration is in the direction of steepest descent. Apart from round-off errors, the procedure will terminate at the minimum of a quadratic in at most  $k$  iterations. In general, however, it is recommended that  $k+1$  iterations be completed before restarting the procedure.

### The Fletcher-Powell-Davidon Method

Redefining  $H$  as any positive definite matrix, we have [27]

$$s^j = -H^j \nabla U^j. \quad (6.96)$$

Note that  $H^j$  is the  $j$ th approximation to the *inverse* of the Hessian matrix. Initially,  $H^0$  is the unit matrix, and again we have the steepest descent direction.

$H$  is continually updated using first derivative information such that

$$\phi^{j+1} - \phi^j = H^{j+1} g^j \quad (6.97)$$

where

$$g^j = \nabla U^{j+1} - \nabla U^j.$$

The following updating procedure is used:

$$H^{j+1} = H^j + \frac{\Delta\phi^j \Delta\phi^{jT}}{\Delta\phi^{jT} g^j} - \frac{H^j g^j g^{jT} H^j}{g^{jT} H^j g^j} \quad (6.98)$$

where

$$\Delta\phi^j = \alpha^j s^j,$$

and  $\alpha^j$  is found by a one-dimensional search (Section 6.5).

Fletcher and Powell prove by induction that if  $H^j$  is positive definite then  $H^{j+1}$  is also positive definite.  $H^0$ , being the unit matrix, is clearly positive definite. On a quadratic function it is further proved that  $H^k$  is the inverse of the Hessian matrix and  $\nabla U^k = 0$ , apart from round-off errors. Both the proof of convergence and success in practice depend on accurate location of the minimum in the linear searches. If necessary,  $H$  may be reset to the unit matrix.

This method is still generally acknowledged to be the best general purpose gradient optimization method.

### 6.8 LEAST $p$ th APPROXIMATION

The material in this section could equally well have been treated under gradient methods. It is useful, however, to distinguish between these problems since special techniques are available for least  $p$ th approximation.

For objective functions in the form of (6.29) and (6.30) we can write

$$\nabla U = \int_{\psi_1}^{\psi_2} \operatorname{Re}\{p |e(\phi, \psi)|^{p-2} e^*(\phi, \psi) \nabla e(\phi, \psi)\} d\psi \quad (6.99)$$

for the continuous case and

$$\nabla U = \sum_{i \in I} \operatorname{Re}\{p |e_i(\phi)|^{p-2} e_i^*(\phi) \nabla e_i(\phi)\} \quad (6.100)$$

for the discrete case. If the appropriate derivatives, namely  $\nabla e$ , are available, we could proceed to optimize with a suitable gradient method (Section 6.7).

In more complicated situations we can envisage a linear combination of functions in the form (6.29) and (6.30), for example,

$$U = \alpha_1 U_1 + \alpha_2 U_2 + \dots \quad (6.101)$$

Simultaneous approximation of more than one response specification might be posed in this way (See Section 6.9). The factors  $\alpha_1, \alpha_2$ , etc. would be given values commensurate with the importance of  $U_1, U_2$ , etc.

Temes and Zai [15, 36] have extended the well-known least squares method of Gauss [6, 8, 14] to a *least  $p$ th method*. Since the former method falls out as a special case, the latter method will be briefly described. For definiteness, assume the objective function is of the form (with real  $e_i(\phi)$ )

$$U = \sum_{i=1}^n [e_i(\phi)]^p \quad (6.102)$$

where  $n > k$  and  $p$  is any positive even integer. Then

$$\nabla U = \sum_{i=1}^n p e_i^{p-1} \nabla e_i \quad (6.103)$$

and

$$\mathbf{H} = \nabla(\nabla U)^T = \sum_{i=1}^n [pe_i^{p-1} \nabla(\nabla e_i)^T + p(p-1)e_i^{p-2} \nabla e_i(\nabla e_i)^T]. \quad (6.104)$$

Now assume that the first term may be neglected in comparison with the second. This really corresponds to a linearization of  $e_i(\phi)$ . Then

$$\mathbf{H} \approx \sum_{i=1}^n p(p-1)e_i^{p-2} \nabla e_i(\nabla e_i)^T.$$

This can be rewritten as

$$\mathbf{H} \approx p(p-1)\mathbf{A}^T\mathbf{B}\mathbf{A} \quad (6.105)$$

where

$$\mathbf{A} \triangleq [\nabla e_1 \quad \nabla e_2 \quad \cdots \quad \nabla e_n]^T$$

and

$$\mathbf{B} \triangleq \begin{bmatrix} e_1^{p-2} & 0 & \cdots & 0 \\ 0 & e_2^{p-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_n^{p-2} \end{bmatrix}.$$

Letting

$$\boldsymbol{\epsilon} \triangleq [e_1^{p-1} \quad e_2^{p-1} \quad \cdots \quad e_n^{p-1}]^T,$$

(6.103) becomes

$$\nabla U = p\mathbf{A}^T\boldsymbol{\epsilon}. \quad (6.106)$$

Using the step given by the Newton method (6.90),

$$\Delta\phi = -(p-1)^{-1}(\mathbf{A}^T\mathbf{B}\mathbf{A})^{-1}\mathbf{A}^T\boldsymbol{\epsilon}. \quad (6.107)$$

Under suitable conditions, it can be shown that  $\Delta\phi$  points in the downhill direction. The modified Newton procedure

$$\phi^{j+1} = \phi^j - \alpha^j(p-1)^{-1}(\mathbf{A}^T\mathbf{B}\mathbf{A})^{-1}\mathbf{A}^T\boldsymbol{\epsilon} \quad (6.108)$$

is recommended where  $\alpha^j$  is chosen to minimize  $U^{j+1}$ .

Damping techniques similar to those used in the Gauss method are applicable [8, 14]. Define, for example,

$$U = \sum_{i=1}^n [e_i(\phi)]^p + \lambda \Delta\phi^T \Delta\phi. \quad (6.109)$$

Then

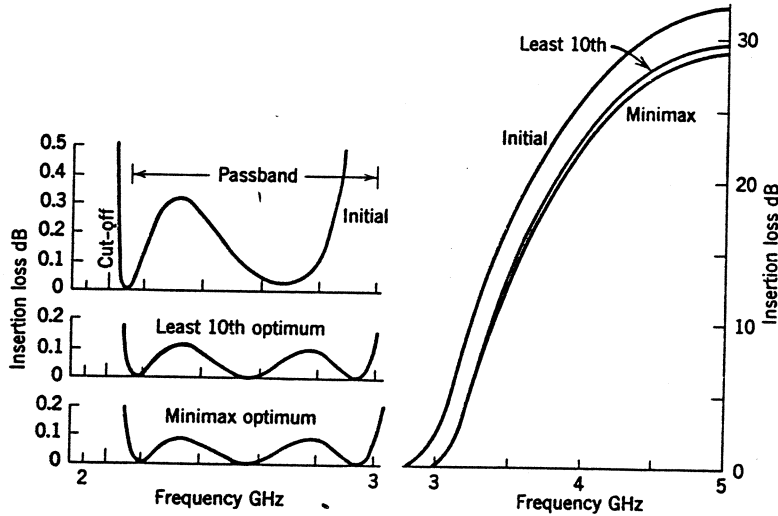
$$\mathbf{H} \approx p(p-1)\mathbf{A}^T\mathbf{B}\mathbf{A} + 2\lambda\mathbf{I}_k \quad (6.110)$$

and

$$\Delta\phi = -p[p(p-1)A^TBA + 2\lambda I_k]^{-1}A^T\epsilon \quad (6.111)$$

It may be shown that the convergence and downhill properties are preserved and that for  $\lambda > 0$  the step is no larger than the undamped step. As  $\lambda \rightarrow 0$  the process is undamped, while for  $\lambda \rightarrow \infty$  the step is in the steepest descent direction. The introduction of  $\alpha^j$  to permit a linear search as in (6.108) is also possible.

**Example 6-3.** An example of least  $p$ th approximation [37] compared with minimax approximation is depicted in Figure 6-19. The structure is the



**FIGURE 6-19** Example of least 10th approximation compared with minimax approximation in optimizing the passband of the filter of Figure 6-9.

seven-section cascade of transmission lines acting as a filter discussed in Section 6.4. The problem here was to see how small the passband insertion loss could be made under the constraints of the problem (if  $R_g$  and  $R_L$  were frequency independent, or the lengths were allowed to vary, the answer would be trivial).

A least  $p$ th objective function was set up with  $p = 10$ , using 51 uniformly spaced points in the passband. The objective function was of the form

$$U = \sum_{i=1}^n \frac{1}{p} |\rho_i(\phi)|^p.$$

The Fletcher-Powell-Davidon method (Section 6.7) was used, the required first derivatives being obtained from *one* network analysis using the adjoint network method (Section 6.9).

Compare the almost equal-ripple passband response obtained with a maximum insertion loss of about 0.1 dB with the equal-ripple response (maximum insertion loss 0.086 dB) produced by minimax approximation. The latter solution was obtained by Bandler and Lee-Chan [24] using a gradient algorithm with quadratic interpolation used to locate the ripple extrema.

The main conclusion to be reached from this example is that acceptable results can be achieved with relatively moderate values of  $p$ . Unless special precautions are taken to avoid ill-conditioning, the use of values of  $p$  much greater than 10 is discouraged.

## 6.9 THE ADJOINT NETWORK METHOD OF GRADIENT EVALUATION

The *adjoint network method* can be used to great advantage in evaluating the gradient vector of objective functions related to gain, insertion loss, reflection coefficient, or any other desired response. A very broad class of networks can be treated by this method. As will be seen, no more than two complete network analyses are required to evaluate the gradient vector regardless of the number of variable parameters.

Director and Rohrer have discussed the concept of the adjoint network and indicated its relevance to automated design of networks in the frequency and time domains [38, 39]. In the frequency domain [39], they considered reciprocal and nonreciprocal, lumped, linear, and time-invariant elements. We will restrict ourselves here to the frequency domain, review Director and Rohrer's results, and extend them to least  $p$ th and minimax approximation. Some uniformly distributed elements will also be included [37, 40].

### Adjoint Networks And Network Sensitivities

Let

$$\mathbf{v} \triangleq \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_b \end{bmatrix} \quad (6.112)$$



contain all the branch voltages in a network and

$$\mathbf{i} \triangleq \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_b \end{bmatrix} \tag{6.113}$$

contain all the corresponding branch currents (using associated reference directions\*).  $\mathbf{v}$  and  $\mathbf{i}$  must satisfy Kirchhoff's voltage and current laws, respectively. Then Tellegen's theorem states [41]

$$\mathbf{v}^T \mathbf{i} = 0. \tag{6.114}$$

As long as the topologies are the same,  $\mathbf{v}$  can refer to one network and  $\mathbf{i}$  to another (See the example in Figure 6-20). Let us, therefore, imagine we have

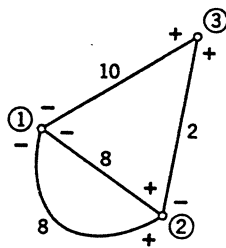
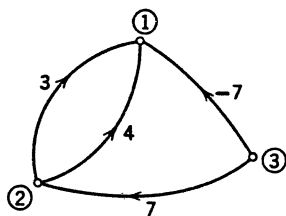


FIGURE 6-20 Illustration of Tellegen's theorem applied to two networks of the same topology. Observe that

$$\mathbf{v}^T \mathbf{i} = 24 + 32 - 70 + 14 = 0.$$

Since the nature of the elements is immaterial, they are replaced by branches.

two networks, the original one which is to be optimized and a topologically equivalent adjoint network. As mentioned earlier we will confine ourselves to a consideration of linear, time-invariant networks in the *frequency domain*. Variables  $V$  and  $I$  will thus denote phasors associated with the original

\* With associated reference directions, the current always enters a branch at the plus sign and leaves at the minus sign.

network, and  $\hat{V}$  and  $\hat{I}$  the corresponding phasors associated with the adjoint network. By Tellegen's theorem

$$\begin{aligned} \mathbf{V}_B^T \hat{\mathbf{I}}_B &= 0 \\ \mathbf{I}_B^T \hat{\mathbf{V}}_B &= 0 \end{aligned} \tag{6.115}$$

where the subscript  $B$  implies that the associated vectors contain all corresponding complex branch voltages and currents. Perturbing elements in the original network we have

$$\Delta \mathbf{V}_B^T \hat{\mathbf{I}}_B = 0 \tag{6.116a}$$

$$\Delta \mathbf{I}_B^T \hat{\mathbf{V}}_B = 0 \tag{6.116b}$$

since Kirchhoff's voltage and current laws must also be applicable to  $\Delta \mathbf{V}_B$  and  $\Delta \mathbf{I}_B$ . Subtracting (6.116b) from (6.116a)

$$\Delta \mathbf{V}_B^T \hat{\mathbf{I}}_B - \Delta \mathbf{I}_B^T \hat{\mathbf{V}}_B = 0. \tag{6.117}$$

Figure 6-21 shows  $N$ -port original and adjoint elements characterized in terms of open-circuit impedance matrices  $\mathbf{Z}$  and  $\hat{\mathbf{Z}}$ , respectively. Letting  $\mathbf{V}$ ,  $\mathbf{I}$ ,

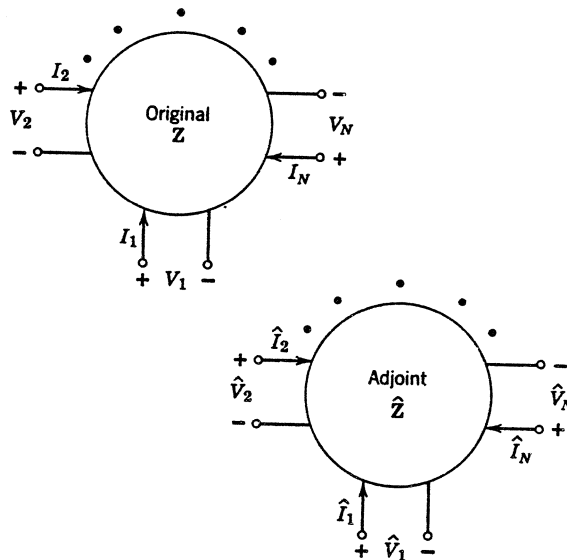


FIGURE 6-21 Original and adjoint elements represented by impedance matrices. In general, many such elements suitably connected form the original and adjoint networks.

$\hat{V}$ , and  $\hat{I}$  denote  $N$ -element vectors containing the relevant port variables

$$V = ZI \tag{6.118}$$

$$\hat{V} = \hat{Z}\hat{I}. \tag{6.119}$$

Perturbing the parameters in the original element and neglecting higher-order terms

$$\Delta V = \Delta ZI + Z \Delta I. \tag{6.120}$$

As indicated by Figure 6-22, the port variables can be thought of as equivalent branch variables, so that, substituting (6.120) into (6.117) we see that

$$(I^T \Delta Z^T + \Delta I^T Z^T)\hat{I} - \Delta I^T \hat{V}$$

reduces to

$$I^T \Delta Z^T \hat{I} \tag{6.121}$$

if

$$\hat{Z} \equiv Z^T. \tag{6.122}$$

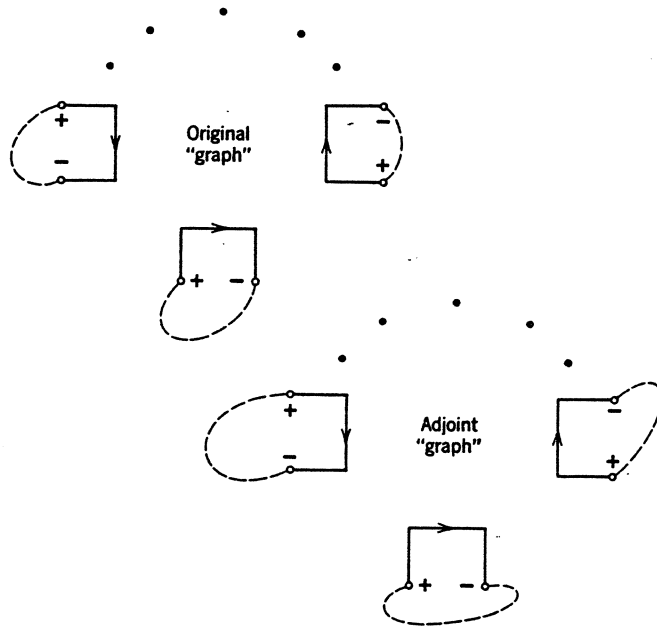


FIGURE 6-22 Representation of the elements of Figure 6-21 for application of Tellegen's theorem. Some  $i$ th equivalent branch might consist of an impedance  $z_{ii}$  in series with voltage generators of value  $z_{ij}I_j$ ,  $j = 1, 2, \dots$ . See, for example, Figure 6-25.

This defines the adjoint element. Observe that expression (6.121), the only term in (6.117) relating to the  $N$ -port element, does not contain  $\Delta I$  or  $\Delta V$ . Further, note that the adjoint of a reciprocal element is identical to the original, since  $Z^T = Z$ .

Next define voltage and current excitation vectors and response vectors as in Figure 6-23. In keeping with the present notation, the hat “ $\wedge$ ” will distinguish the corresponding quantities for the adjoint network. Terms in (6.117) associated with the excitations and responses are

$$\Delta V_V^T \hat{I}_V - \Delta I_V^T \hat{V}_V + \Delta V_I^T \hat{I}_I - \Delta I_I^T \hat{V}_I$$

which reduces to

$$-\Delta I_V^T \hat{V}_V + \Delta V_I^T \hat{I}_I \tag{6.123}$$

since  $\Delta V_V$  and  $\Delta I_I$  become zero when the excitations are held fixed.

Clearly, any network may be thought of as consisting of the interconnection of a number of multiport elements. Thus, several terms of the form of expression (6.121) can appear in (6.117).

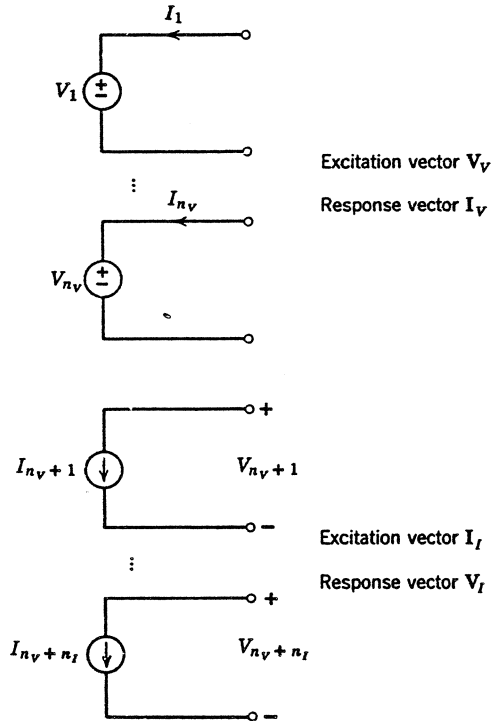


FIGURE 6-23 Port excitation and response vectors.

For an admittance matrix representation we can show that

$$-\mathbf{V}^T \Delta \mathbf{Y}^T \hat{\mathbf{V}} \quad (6.124)$$

corresponds to expression (6.121).  $\mathbf{Y}^T$  is the admittance matrix of the adjoint. Things are slightly more complicated for the hybrid matrix. If we take

$$\begin{bmatrix} \mathbf{I}_a \\ \mathbf{V}_b \end{bmatrix} = \begin{bmatrix} \mathbf{Y} & \mathbf{A} \\ \mathbf{M} & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \mathbf{V}_a \\ \mathbf{I}_b \end{bmatrix}, \quad (6.125)$$

then the corresponding relation for the adjoint is

$$\begin{bmatrix} \hat{\mathbf{I}}_a \\ \hat{\mathbf{V}}_b \end{bmatrix} = \begin{bmatrix} \mathbf{Y}^T & -\mathbf{M}^T \\ -\mathbf{A}^T & \mathbf{Z}^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}}_a \\ \hat{\mathbf{I}}_b \end{bmatrix}. \quad (6.126)$$

The expression corresponding to (6.121) can be shown to be

$$\begin{bmatrix} \mathbf{V}_a^T & \mathbf{I}_b^T \end{bmatrix} \begin{bmatrix} -\Delta \mathbf{Y}^T & \Delta \mathbf{M}^T \\ -\Delta \mathbf{A}^T & \Delta \mathbf{Z}^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}}_a \\ \hat{\mathbf{I}}_b \end{bmatrix}. \quad (6.127)$$

To summarize the results of the above discussion, we note that (6.117) can be written in the form

$$\Delta \mathbf{I}_v^T \hat{\mathbf{V}}_v - \Delta \mathbf{V}_i^T \hat{\mathbf{I}}_i = \mathbf{G}^T \Delta \phi \quad (6.128)$$

where  $\mathbf{G}$  is a vector of sensitivity components related to the adjustable parameters of the network, namely  $\phi$ . Equation (6.128) basically relates changes in port responses due to changes in element values.

Figure 6-24 shows the results of a direct application of the formulas (6.121) and (6.124) to three commonly used elements. Table 6-2 summarizes sensitivity expressions for some commonly used lumped and distributed elements. An element consisting of a single branch is simply viewed as a one-port element.

Consider, for example, a uniformly distributed line (Figure 6-25) having characteristic impedance  $Z_0$  and electrical length  $\theta$ . Since the element is reciprocal:

$$\hat{\mathbf{Z}} = \mathbf{Z}^T = \mathbf{Z} = Z_0 \begin{bmatrix} \coth \theta & \operatorname{csch} \theta \\ \operatorname{csch} \theta & \coth \theta \end{bmatrix}. \quad (6.129)$$

Invoking expression (6.121) we obtain

$$\begin{aligned} \mathbf{I}^T \Delta \mathbf{Z}^T \hat{\mathbf{I}} &= \mathbf{I}^T \left( \Delta Z_0 \begin{bmatrix} \coth \theta & \operatorname{csch} \theta \\ \operatorname{csch} \theta & \coth \theta \end{bmatrix} - \frac{Z_0 \Delta \theta}{\sinh \theta} \begin{bmatrix} \operatorname{csch} \theta & \coth \theta \\ \coth \theta & \operatorname{csch} \theta \end{bmatrix} \right)^T \hat{\mathbf{I}} \\ &= \left( \frac{\Delta Z_0}{Z_0} \mathbf{Z} \mathbf{I} - \frac{\Delta \theta}{\sinh \theta} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{Z} \mathbf{I} \right)^T \hat{\mathbf{I}} = \frac{\Delta Z_0}{Z_0} \mathbf{V}^T \hat{\mathbf{I}} - \frac{\Delta \theta}{\sinh \theta} \mathbf{V}^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \hat{\mathbf{I}}. \end{aligned} \quad (6.130)$$

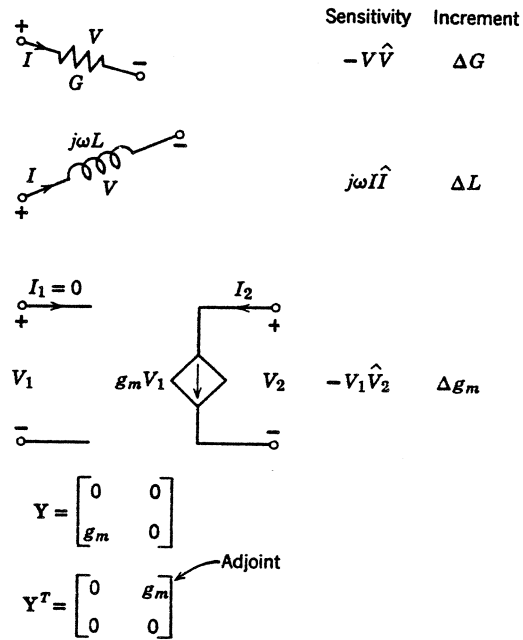


FIGURE 6-24 Sensitivities for three common elements: a resistor of conductance  $G$ , an inductor of inductance  $L$  and a voltage-controlled current source with transfer conductance  $g_m$ .

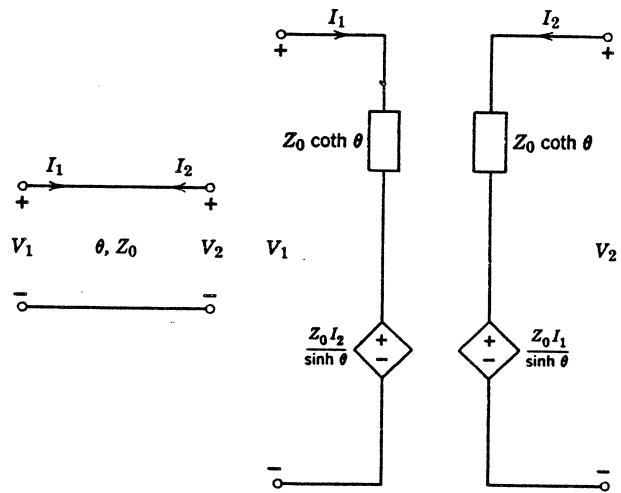


FIGURE 6-25 Uniform line and convenient representation.

Observe that the sensitivities shown in Table 6-2 depend on currents and voltages present in the *unperturbed* original and adjoint networks. At most, two network analyses (using any suitable method) will, therefore, yield the information required to evaluate them. Note that if there is no excitation at a port, the appropriate source is set to zero. If the response at a port is of no interest, the appropriate adjoint excitation should be zero. Elements or parameters not to be varied are simply not represented in  $G$  or  $\phi$ .

**An Application to Minimax Approximation**

Consider the situation depicted in Figure 6-26. Suppose we are given the problem: minimize a positive independent variable  $U$  subject to

$$U \geq f(\phi, \omega_i) \triangleq |\rho(\phi, j\omega_i)|^2, \quad \omega_i \in \Omega_d \tag{6.131}$$

where  $\rho$  is the input reflection coefficient, and  $\Omega_d$  is a discrete set of frequencies in the band of interest. This problem then is effectively to minimize the maximum magnitude of the reflection coefficient over a band [37, 42]. Now

$$\rho = \frac{Z_{in} - R_g}{Z_{in} + R_g} = 1 - \frac{2R_g}{Z_{in} + R_g} = 1 + \frac{2R_g I_g}{V_g} \tag{6.132}$$

so that

$$\begin{aligned} \nabla f(\phi, \omega_i) &= \text{Re}\{2\rho^*(\phi, j\omega_i) \nabla \rho(\phi, j\omega_i)\} \\ &= \text{Re}\left\{\frac{4R_g}{V_g} \rho^*(\phi, j\omega_i) \nabla I_g(\phi, j\omega_i)\right\}. \end{aligned} \tag{6.133}$$

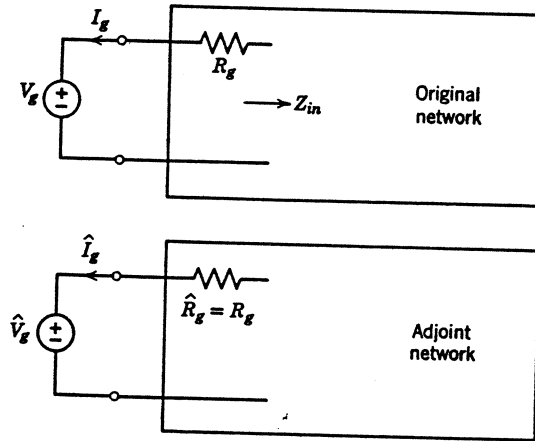


FIGURE 6-26 Possible original and adjoint networks for design on the reflection coefficient basis.

TABLE 6-2 Sensitivity Expressions for Some Lumped and Distributed Elements

Element	Equation		Sensitivity (component of $G$ )	Increment (component of $\Delta\phi$ )
	Original	Adjoint		
Resistor	$V = RI$ $I = GV$	$\hat{V} = RI\hat{I}$ $\hat{I} = G\hat{V}$	$\frac{I}{V} \hat{I}$ $-\frac{I}{V} \hat{V}$	$\Delta R$ $\Delta G$
Inductor	$V = j\omega LI$ $I = \frac{1}{j\omega} \Gamma V$	$\hat{V} = j\omega L\hat{I}$ $\hat{I} = \frac{1}{j\omega} \Gamma \hat{V}$	$\frac{j\omega I}{V} \hat{I}$ $-\frac{1}{j\omega} \frac{V}{V} \hat{V}$	$\Delta L$ $\Delta \Gamma$
Capacitor	$V = \frac{1}{j\omega} SI$ $I = j\omega CV$	$\hat{V} = \frac{1}{j\omega} S\hat{I}$ $\hat{I} = j\omega C\hat{V}$	$\frac{1}{j\omega} \frac{I}{V} \hat{I}$ $-\frac{1}{j\omega} \frac{V}{V} \hat{V}$	$\Delta S$ $\Delta C$
Transformer	$\begin{bmatrix} V_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 0 & n \\ -n & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \end{bmatrix}$	$\begin{bmatrix} \hat{V}_1 \\ \hat{I}_2 \end{bmatrix} = \begin{bmatrix} 0 & n \\ -n & 0 \end{bmatrix} \begin{bmatrix} \hat{I}_1 \\ \hat{V}_2 \end{bmatrix}$	$V_2 I_1 + I_2 V_1$	$\Delta n$
Gyrator	$V = \begin{bmatrix} 0 & \alpha \\ -\alpha & 0 \end{bmatrix} I$	$\hat{V} = \begin{bmatrix} 0 & -\alpha \\ \alpha & 0 \end{bmatrix} \hat{I}$	$I_1 I_2 - I_2 I_1$	$\Delta \alpha$
Voltage controlled voltage source	$\begin{bmatrix} I_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \mu & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ I_2 \end{bmatrix}$	$\begin{bmatrix} \hat{I}_1 \\ \hat{V}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\mu \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{V}_1 \\ \hat{I}_2 \end{bmatrix}$	$V_1 I_2$	$\Delta \mu$
Voltage controlled current source	$I = \begin{bmatrix} 0 & 0 \\ g_m & 0 \end{bmatrix} V$	$\hat{I} = \begin{bmatrix} 0 & g_m \\ 0 & 0 \end{bmatrix} \hat{V}$	$-V_1 \hat{V}_2$	$\Delta g_m$



Table 6-2 Continued

Current controlled voltage source	$\mathbf{V} = \begin{bmatrix} 0 & 0 \\ r_m & 0 \end{bmatrix} \mathbf{I}$	$\hat{\mathbf{V}} = \begin{bmatrix} 0 & r_m \\ 0 & 0 \end{bmatrix} \mathbf{I}$	$I_1, I_2$	$\Delta r_m$
Current controlled current source	$\begin{bmatrix} V_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \beta & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \end{bmatrix}$	$\begin{bmatrix} \hat{V}_1 \\ \hat{I}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\beta \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \end{bmatrix}$	$-I_1, \hat{V}_2$	$\Delta \beta$
Short circuited uniformly distributed line	$V = Z_0 \tanh \theta I$ $I = Y_0 \coth \theta V$	$\hat{V} = Z_0 \tanh \theta \hat{I}$ $\hat{I} = Y_0 \coth \theta \hat{V}$	$\begin{matrix} \tanh \theta l \\ Z_0 \operatorname{sech}^2 \theta l \\ -\coth \theta Y_0 \\ Y_0 \operatorname{csch}^2 \theta Y_0 \end{matrix}$	$\begin{matrix} \Delta Z_0 \\ \Delta \theta \\ \Delta Y_0 \\ \Delta \theta \end{matrix}$
Open circuited uniformly distributed line	$V = Z_0 \coth \theta I$ $I = Y_0 \tanh \theta V$	$\hat{V} = Z_0 \coth \theta \hat{I}$ $\hat{I} = Y_0 \tanh \theta \hat{V}$	$\begin{matrix} \coth \theta l \\ -Z_0 \operatorname{csch}^2 \theta l \\ -\tanh \theta Y_0 \\ -Y_0 \operatorname{sech}^2 \theta Y_0 \end{matrix}$	$\begin{matrix} \Delta Z_0 \\ \Delta \theta \\ \Delta Y_0 \\ \Delta \theta \end{matrix}$
Uniformly distributed line	$\mathbf{V} = Z_0 \begin{bmatrix} \coth \theta & \operatorname{csch} \theta \\ \operatorname{csch} \theta & \coth \theta \end{bmatrix} \mathbf{I}$ $\mathbf{I} = Y_0 \begin{bmatrix} \coth \theta & -\operatorname{csch} \theta \\ -\operatorname{csch} \theta & \coth \theta \end{bmatrix} \mathbf{V}$	same as original network equation but with $\hat{\mathbf{V}}$ and $\hat{\mathbf{I}}$ replacing $\mathbf{V}$ and $\mathbf{I}$ respectively	$\begin{matrix} \frac{1}{Z_0} \mathbf{V}^T \mathbf{I} \\ \frac{1}{\sinh \theta} \mathbf{V}^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{I} \\ -\frac{1}{Y_0} \mathbf{I}^T \hat{\mathbf{V}} \\ \frac{1}{\sinh \theta} \mathbf{I}^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \hat{\mathbf{V}} \end{matrix}$	$\begin{matrix} \Delta Z_0 \\ \Delta \theta \\ \Delta Y_0 \\ \Delta \theta \end{matrix}$

Table 6-2 Continued

Lossless transmission line	$V = -jZ_0 \begin{bmatrix} \cot \beta l & \csc \beta l \\ \csc \beta l & \cot \beta l \end{bmatrix} \mathbf{I}$	$\frac{1}{Z_0} \mathbf{V}^T \mathbf{I}$ $-\frac{\beta}{\sin \beta l} \mathbf{V}^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{I}$	$\Delta Z_0$ $\Delta l$
	$\mathbf{I} = -jY_0 \begin{bmatrix} \cot \beta l & -\csc \beta l \\ -\csc \beta l & \cot \beta l \end{bmatrix} \mathbf{V}$	$-\frac{1}{Y_0} \mathbf{I}^T \mathbf{V}$ $-\frac{\beta}{\sin \beta l} \mathbf{I}^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{V}$	$\Delta Y_0$ $\Delta l$
Uniform RC line	<p>as for uniformly distributed line with</p> $Z_0 = \sqrt{\frac{R}{sC}} \text{ and } \theta = \sqrt{sRC}$	$\frac{1}{2R} \mathbf{V}^T \begin{bmatrix} 1 & -\frac{\theta}{\sinh \theta} \\ -\frac{\theta}{\sinh \theta} & 1 \end{bmatrix} \mathbf{I}$ $-\frac{1}{2C} \mathbf{V}^T \begin{bmatrix} 1 & \frac{\theta}{\sinh \theta} \\ \frac{\theta}{\sinh \theta} & 1 \end{bmatrix} \mathbf{I}$	$\Delta R$ $\Delta C$

same as

original

network

equation

but

with

$\hat{\mathbf{V}}$  and  $\hat{\mathbf{I}}$

replacing

$\mathbf{V}$  and  $\mathbf{I}$

respectively

From (6.128)

$$\Delta I_g \hat{V}_g = \mathbf{G}^T \Delta \phi. \quad (6.134)$$

Hence

$$\Delta I_g = \left( \frac{1}{\hat{V}_g} \mathbf{G}^T \right) \Delta \phi = \nabla I_g^T \Delta \phi,$$

so that

$$\nabla I_g = \frac{1}{\hat{V}_g} \mathbf{G}, \quad (6.135)$$

and, finally,

$$\nabla f(\phi, \omega_i) = \operatorname{Re} \left\{ \frac{4R_g}{V_g \hat{V}_g} \rho^*(\phi, j\omega_i) \mathbf{G}(\phi, j\omega_i) \right\}. \quad (6.136)$$

Observe that we are at liberty to set  $\hat{V}_g = V_g$ . If the original network is reciprocal so that the adjoint network is identical to the original, we need perform only one network analysis to obtain  $\nabla f(\phi, \omega_i)$ .

#### An Application to Least $p$ th Approximation

It can be shown that if there are  $n_v$  independent voltage sources and  $n_I$  independent current sources

$$\mathbf{G} = \sum_{i=1}^{n_v} \hat{V}_i \nabla I_i - \sum_{i=n_v+1}^{n_v+n_I} \hat{I}_i \nabla V_i \quad (6.137)$$

Suppose we are given the objective function [37, 39, 42],

$$U = \sum_{i=1}^{n_v+n_I} \int_{\Omega} |e_i(\phi, j\omega)|^p d\omega, \quad (6.138)$$

where  $\Omega$  defines a frequency range of interest and where  $e_i(\phi, j\omega)$  is an  $i$ th function of the form of (6.21) such that

$$F_i(\phi, j\omega) \triangleq \begin{cases} I_i(\phi, j\omega), & i = 1, 2, \dots, n_v \\ V_i(\phi, j\omega), & i = n_v + 1, \dots, n_v + n_I. \end{cases} \quad (6.139)$$

Equation (6.138) thus represents a summation of functions of the form of (6.101). The specified functions  $S_i(j\omega)$  correspond to desired response currents and voltages. In general,  $F_i(\phi, j\omega)$ ,  $S_i(j\omega)$ , and hence  $e_i(\phi, j\omega)$  may be complex. Now, from (6.99)

$$\nabla U = \sum_{i=1}^{n_v+n_I} \int_{\Omega} \operatorname{Re} \{ p |e_i(\phi, j\omega)|^{p-2} w_i(\omega) e_i^*(\phi, j\omega) \nabla F_i(\phi, j\omega) \} d\omega. \quad (6.140)$$

Comparing (6.137), (6.139), and (6.140), we see that if the adjoint network excitations are taken as

$$p | e_i(\phi, j\omega) |^{p-2} w_i(\omega) e_i^*(\phi, j\omega) = \begin{cases} \hat{V}_i(j\omega) & i = 1, 2, \dots, n_V \\ -\hat{I}_i(j\omega) & i = n_V + 1, \dots, n_V + n_I, \end{cases} \quad (6.141)$$

then

$$\nabla U = \int_{\Omega} \text{Re}\{G\} d\omega. \quad (6.142)$$

The corresponding expression for the discrete case is

$$\nabla U = \sum_{\Omega_d} \text{Re}\{G\} \quad (6.143)$$

where  $\Omega_d$  is the discrete set of frequencies.

### An Application to Group Delay Computation

In group delay computations we are essentially interested in sensitivities with respect to frequency  $\omega$  [43]. This parameter is different from others that we have considered in that it is common throughout the network. Specifically, let us distinguish variables associated with some  $j$ th element of an  $n$ -element network by the subscript  $j$ . Then, assuming only  $\omega$  is varied, (6.128) can be written as

$$\Delta \mathbf{I}_V^T \hat{\mathbf{V}}_V - \Delta \mathbf{V}_I^T \hat{\mathbf{I}}_I = \sum_{j=1}^n [\mathbf{V}_{aj}^T \mathbf{I}_{bj}^T] \begin{bmatrix} -\Delta \mathbf{Y}_j^T & \Delta \mathbf{M}_j^T \\ -\Delta \mathbf{A}_j^T & \Delta \mathbf{Z}_j^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}}_{aj} \\ \hat{\mathbf{I}}_{bj} \end{bmatrix}, \quad (6.144)$$

if each element, for complete generality, is characterized by an appropriate hybrid matrix. Using the rule that  $\Delta x = (\partial x / \partial \omega) \Delta \omega$ , where  $x$  is any quantity depending on  $\omega$ , (6.144) can be more appropriately written

$$\sum_{i=1}^{n_V} \hat{V}_i \frac{\partial I_i}{\partial \omega} - \sum_{i=n_V+1}^{n_V+n_I} \hat{I}_i \frac{\partial V_i}{\partial \omega} = \sum_{j=1}^n G_{\omega j} \quad (6.145)$$

where

$$G_{\omega j} \triangleq [\mathbf{V}_{aj}^T \mathbf{I}_{bj}^T] \begin{bmatrix} -\frac{\partial \mathbf{Y}_j^T}{\partial \omega} & \frac{\partial \mathbf{M}_j^T}{\partial \omega} \\ -\frac{\partial \mathbf{A}_j^T}{\partial \omega} & \frac{\partial \mathbf{Z}_j^T}{\partial \omega} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}}_{aj} \\ \hat{\mathbf{I}}_{bj} \end{bmatrix}. \quad (6.146)$$

If, in particular, the  $k$ th port is to be investigated, and this happens to be a current-excited port,\* then (6.145) reduces to

$$-\hat{I}_k \frac{\partial V_k}{\partial \omega} = \sum_{j=1}^n G_{\omega j} \quad (6.147)$$

if all adjoint excitations except  $\hat{I}_k$  are set to zero. Evaluation of the sensitivity expression  $G_{\omega j}$  is accomplished by the results of two network analyses. The sensitivity formulas from Table 6-2 may be used if appropriate, since

$$\begin{bmatrix} -\frac{\partial \mathbf{Y}_j^T}{\partial \omega} & \frac{\partial \mathbf{M}_j^T}{\partial \omega} \\ -\frac{\partial \mathbf{A}_j^T}{\partial \omega} & \frac{\partial \mathbf{Z}_j^T}{\partial \omega} \end{bmatrix} = \sum_r \begin{bmatrix} -\frac{\partial \mathbf{Y}_j^T}{\partial \phi_r} & \frac{\partial \mathbf{M}_j^T}{\partial \phi_r} \\ -\frac{\partial \mathbf{A}_j^T}{\partial \phi_r} & \frac{\partial \mathbf{Z}_j^T}{\partial \phi_r} \end{bmatrix} \frac{\partial \phi_r}{\partial \omega}, \quad (6.148)$$

where the subscript  $r$  denotes some  $r$ th parameter in the  $j$ th element with respect to which a sensitivity expression is already available.

Consider, for example,  $\theta = j\omega l/c = j\beta l$  where  $c$  is the velocity of propagation. Then the  $\omega$ -sensitivity of a lossless transmission line is  $j/c$  times the  $\theta$ -sensitivity shown in Table 6-2. Consider an inductor as a second example. The lefthand side of (6.148) reduces immediately to  $jL$  using  $Z = j\omega L$ .

Finally, to compute the group delay  $T_G(\omega)$  we note that

$$T_G(\omega) = -\text{Im} \left\{ \frac{1}{V_k} \frac{\partial V_k}{\partial \omega} \right\}, \quad (6.149)$$

where it is assumed that all sources have constant, frequency-independent phase angles. For convenience, letting the excitation  $\hat{I}_k = 1/V_k$ ,

$$T_G(\omega) = \text{Im} \left\{ \sum_{j=1}^n G_{\omega j} \right\}. \quad (6.150)$$

Equation (6.150) is also valid for calculations of group delay if the  $k$ th port is a voltage-excited port.\* All one has to remember is to set all adjoint excitations to zero except  $\hat{V}_k$  which is set to  $-1/I_k$ .

### Extensions and Other Applications

An important point to remember about the adjoint network method is that the analysis of the adjoint, in general, can take considerably less effort than the analysis of the original network. If  $\mathbf{Y}_n$  is, for example, the nodal admittance matrix of the original network, and its inverse  $\mathbf{Y}_n^{-1}$  has been computed, then we can use the result  $(\mathbf{Y}_n^T)^{-1} = (\mathbf{Y}_n^{-1})^T$ . For a further discussion of possible computational efficiency, the reader is referred to Director [44].

\*The value of the excitation could, of course, be zero.

Extensions to second-order sensitivities have been formulated [45], including group-delay sensitivities [43]. Of particular interest to filter designers are the recent applications of the adjoint network concept to the computation of dissipation-induced loss distortion in both lumped and distributed networks [46, 47]. Further extensions include the exploitation of the adjoint network concept in first- and second-order sensitivity computation using wave variables rather than voltages and currents [48, 49, 50]. These results should also be of interest to filter designers.

### 6.10 SUMMARY

A wide range of topics in the field of computer-aided circuit optimization has been discussed. Formulations and methods suitable for automated design, when the classical approach is inappropriate, have been stressed. The formulation of objective functions from design objectives has been discussed, including least  $p$ th and minimax. Methods of dealing with parameter and response constraints by means of transformations or penalties have been considered in some detail. Minimax approximation through linear programming and nonlinear programming has been discussed. Efficient one-dimensional methods and multidimensional gradient and direct search methods have been reviewed. Least  $p$ th approximation has been considered, with emphasis on gradient methods of solution. Finally, the adjoint network method of evaluating derivatives for design in the frequency domain was reviewed.

Most computer centers should have linear programming routines, and at least one efficient gradient algorithm, available as library programs, and possibly other methods also. It is hoped that this chapter has gone a reasonable way towards helping the network designer formulate his problems effectively so that he can take full advantage of the available computer programs.

### ACKNOWLEDGEMENT

The cooperation of R. E. Seviara of the University of Toronto, particularly in the section on adjoint networks, is much appreciated. Dr. E. Della Torre of McMaster University provided considerable constructive criticism.

### PROBLEMS

- 6.1 (a) Prove that  $\Delta U$  is maximized in the direction of  $\nabla U$  for a given step size.  
 (b) Use the multidimensional Taylor series expansion to show that a turning point of a convex differentiable function is a global minimum.

- 6.2 (a) If  $g(\phi)$  is concave, verify that  $g(\phi) \geq 0$  describes a convex feasible region.  
 (b) Under what conditions could equality constraints be included in convex programming?
- 6.3 Find suitable transformations for the following constraints so that we can use unconstrained optimization.  
 (a)  $0 \leq \phi_1 \leq \phi_2 \leq \dots \leq \phi_i \leq \dots \leq \phi_k$ .  
 (b)  $0 < l \leq \phi_2/\phi_1 \leq u$   
 $\phi_1 > 0$   
 $\phi_2 > 0$ .
- 6.4 Derive (6.99) and (6.100).
- 6.5 Derive the sensitivity expression (6.124) from first principles.
- 6.6 Derive the entries of Table 6-2 relating to:  
 (a) A voltage controlled voltage source.  
 (b) An open-circuited uniformly distributed line.  
 (c) A uniform RC line.
- 6.7 Verify that the adjoint network may be characterized by the hybrid matrix description in (6.126).
- 6.8 Obtain the adjoint network in terms of an  $ABCD$  or chain matrix characterization of a two-port. Find sensitivity expressions in these terms for some of the entries of Table 6-2.
- 6.9 Consider the problem of minimizing

$$U = \phi_3(\phi_1 + \phi_2)^2$$

subject to

$$g_1 = \phi_1 - \phi_2^2 \geq 0$$

$$g_2 = \phi_2 \geq 0$$

$$h = (\phi_1 + \phi_2)\phi_3 - 1 = 0.$$

Is this a convex programming problem? Formulate it for solution by the sequential unconstrained minimization method. Starting with a feasible point, show how the constrained minimum is approached as the parameter  $r \rightarrow 0$ . Draw a contour sketch to illustrate the process. Are the conditions for a constrained minimum satisfied?

- 6.10 For the linear function

$$F(\phi, \psi) = \sum_{i=1}^k \phi_i f_i(\psi),$$

- (a) Formulate the discrete minimax approximation of  $S(\psi)$  by  $F(\phi, \psi)$  as a linear programming problem, assuming  $\phi$  to be unconstrained.  
 (b) Assuming an objective function of the form of (6.102), derive  $\nabla U$  and  $H$  (Note that a polynomial is a special case).

- 6.11 Verify (6.137).
- 6.12 Formulate the design of a notch filter in terms of inequality constraints, given the following requirements. The attenuation should not exceed  $A_1$  dB over the frequency range 0 to  $\omega_1$ , and  $A_2$  dB over the range  $\omega_2$  to  $\omega_3$ , with  $0 < \omega_1 < \omega_2 < \omega_3$ . At  $\omega_0$ , where  $\omega_1 < \omega_0 < \omega_2$ , the attenuation must exceed  $A_0$  dB.
- 6.13 Devise an algorithm for finding the extrema of a well-behaved multimodal function of one variable (Figure 6-2), such as the passband response of a filter.
- 6.14 Discuss the scaling effects of the transformation  $\phi_i = \exp \phi'_i$  (Table 6-1).
- 6.15 (a) Are the necessary conditions for a constrained *minimum* satisfied anywhere along the boundary of the feasible region in Figure 6-1?  
 (b) What about the conditions for a constrained *maximum*?
- 6.16 Suppose we have to minimize

$$U = \sum_{\omega_i \in \Omega_d} [L(\omega_i) - S(\omega_i)]^p$$

where  $L(\omega_i)$  is the insertion loss in dB of a filter between  $R_g$  and  $R_L$ ,  $S(\omega_i)$  is the desired insertion loss between  $R_g$  and  $R_L$ ,  $\Omega_d$  is a set of discrete frequencies  $\omega_i$ , and  $p$  is an even positive integer. Obtain an expression relating  $\nabla U$  to  $\mathbf{G}(j\omega_i)$  where the elements of  $\mathbf{G}$  might be as in Table 6-2. Assume convenient values for the excitations of the original and adjoint networks.

## REFERENCES

- [1] R. Fletcher, "A review of methods for unconstrained optimization," *Optimization*, R. Fletcher, Ed., Academic Press, New York, 1969.
- [2] E. M. L. Beale, "Nonlinear programming," *Digital Computer User's Handbook*, M. Klerer and G. A. Korn, Ed., McGraw-Hill, New York, 1967.
- [3] P. Wolfe, "Methods of nonlinear programming," *Nonlinear Programming*, J. Abadie, Ed., John Wiley, New York, 1967.
- [4] W. I. Zangwill, *Nonlinear Programming*, Prentice-Hall, Englewood Cliffs, New Jersey, 1969.
- [5] M. J. Box, "A comparison of several current optimization methods, and the use of transformations in constrained problems," *Computer J.*, **9**, 67-77 (May, 1966).
- [6] J. W. Bandler, "Optimization methods for computer-aided design," *IEEE Trans. Microwave Theory and Techniques*, **MTT-17**, 533-552 (Aug., 1969).
- [7] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley, New York, 1968.



- [8] J. Kowalik and M. R. Osborne, *Methods for Unconstrained Optimization Problems*, Elsevier, New York, 1968.
- [9] F. A. Lootsma, "Logarithmic programming: a method of solving nonlinear-programming problems," *Philips Res. Repts.*, **22**, 329-344 (June, 1967).
- [10] J. Bracken and G. P. McCormick, *Selected Applications of Nonlinear Programming*, John Wiley, New York, 1968.
- [11] H. W. Kuhn and A. W. Tucker, "Non-linear programming," *Proc. 2nd Symp. on Math. Statistics and Probability. Berkeley, Calif.*, University of California Press, 481-493, 1951.
- [12] D. C. Handscomb, Ed., *Methods of Numerical Approximation*, Pergamon, Oxford, 1966.
- [13] G. C. Temes and J. A. C. Bingham, "Iterative Chebyshev approximation technique for network synthesis," *IEEE Trans. Circuit Theory*, CT-14, 31-37 (March, 1967).
- [14] G. C. Temes and D. A. Calahan, "Computer-aided network optimization the state-of-the-art," *Proc. IEEE*, **55**, 1832-1863 (Nov., 1967).
- [15] G. C. Temes, "Optimization methods in circuit design," *Computer Oriented Circuit Design*, F. F. Kuo and W. G. Magnuson, Jr., Ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1969.
- [16] L. S. Lasdon and A. D. Waren, "Optimal design of filters with bounded, lossy elements," *IEEE Trans. Circuit Theory*, CT-13, 175-187 (June, 1966).
- [17] A. D. Waren, L. S. Lasdon, and D. F. Suchman, "Optimization in engineering design," *Proc. IEEE*, **55**, 1885-1897 (Nov., 1967).
- [18] Y. Ishizaki and H. Watanabe, "An iterative Chebyshev approximation method for network design," *IEEE Trans. Circuit Theory*, CT-15, 326-336 (Dec., 1968).
- [19] M. R. Osborne and G. A. Watson, "An algorithm for minimax approximation in the nonlinear case," *Computer J.*, **12**, 63-68 (Feb., 1969).
- [20] J. W. Bandler, T. V. Srinivasan and C. Charalambous, "Minimax optimization of networks by grazor search," *IEEE Trans. Microwave Theory and Techniques*, MTT-20, 596-604 (Sept., 1972).
- [21] J. W. Bandler, "Computer optimization of inhomogeneous waveguide transformers," *IEEE Trans. Microwave Theory and Techniques*, MTT-17, 563-571 (Aug., 1969).
- [22] J. W. Bandler and P. A. Macdonald, "Optimization of microwave networks by razor search," *IEEE Trans. Microwave Theory and Techniques*, MTT-17, 552-562 (Aug., 1969).
- [23] H. J. Carlin and O. P. Gupta, "Computer design of filters with lumped-distributed elements or frequency variable terminations," *IEEE Trans. Microwave Theory and Techniques*, MTT-17, 598-604 (Aug., 1969).
- [24] J. W. Bandler and A. G. Lee-Chan, "Gradient razor search method for optimization," *1971 International Microwave Symp., Digest of Technical Papers*, 118-119 (May, 1971).
- [25] J. W. Bandler, "Conditions for a minimax optimum," *IEEE Trans. Circuit Theory*, CT-18, 476-479 (July, 1971).
- [26] M. J. Box, D. Davies, and W. H. Swann, *Non-linear Optimization Techniques*, Oliver and Boyd, Edinburgh, 1969.

- [27] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Computer J.*, 6, 163-168 (June, 1963).
- [28] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *Computer J.*, 7, 155-162 (July, 1964).
- [29] J. W. Bandler and P. A. Macdonald, "Cascaded noncommensurate transmission-line networks as optimization problems," *IEEE Trans. Circuit Theory*, CT-16, 391-394 (Aug., 1969).
- [30] R. Hooke and T. A. Jeeves, "'Direct search' solution of numerical and statistical problems," *J. ACM*, 8, 212-229 (April, 1961).
- [31] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer J.*, 7, 308-313 (Jan., 1965).
- [32] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *Computer J.*, 3, 175-184 (Oct., 1960).
- [33] W. I. Zangwill, "Minimizing a function without calculating derivatives," *Computer J.*, 10, 293-296 (Nov., 1967).
- [34] R. Fletcher, "Function minimization without evaluating derivatives—a review," *Computer J.*, 8, 33-41 (April, 1965).
- [35] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *Computer J.*, 7, 149-154 (July, 1964).
- [36] G. C. Temes and D. Y. F. Zai, "Least  $p$ th approximation," *IEEE Trans. Circuit Theory*, CT-16, 235-237 (May, 1969).
- [37] J. W. Bandler and R. E. Seviara, "Current trends in network optimization," *IEEE Trans. Microwave Theory and Techniques*, MTT-18, 1159-1170 (Dec., 1970).
- [38] S. W. Director and R. A. Rohrer, "The generalized adjoint network and network sensitivities," *IEEE Trans. Circuit Theory*, CT-16, 318-323 (Aug., 1969).
- [39] S. W. Director and R. A. Rohrer, "Automated network design—the frequency-domain case," *IEEE Trans. Circuit Theory*, CT-16, 330-337 (Aug., 1969).
- [40] J. W. Bandler and R. E. Seviara, "Computation of sensitivities for noncommensurate networks," *IEEE Trans. Circuit Theory*, CT-18, 174-178 (Jan., 1971).
- [41] C. A. Desoer and E. S. Kuh, *Basic Circuit Theory*, McGraw-Hill, New York, 1969, Chapter 9.
- [42] R. E. Seviara, M. Sablatash and J. W. Bandler, "Least  $p$ th and minimax objectives for automated network design," *Electronics Letters*, 6, 14-15 (Jan., 1970).
- [43] G. C. Temes, "Exact computation of group delay and its sensitivities using adjoint-network concept," *Electronics Letters*, 6, 483-485 (July, 1970).
- [44] S. W. Director, "LU factorization in network sensitivity calculations," *IEEE Trans. Circuit Theory*, CT-18, 184-185 (Jan., 1971).
- [45] G. A. Richards, "Second-derivative sensitivity using the concept of the adjoint network," *Electronics Letters*, 5, 398-399 (Aug., 1969).
- [46] G. C. Temes and R. N. Gadenz, "Simple technique for the prediction of dissipation-induced loss distortion," *Electronics Letters*, 6, 836-837 (Dec., 1970).

- [47] R. N. Gadenz and G. C. Temes, "Computation of dissipation-induced loss distortion in lumped/distributed networks," *Electronics Letters*, 7, 258-260 (May, 1971).
- [48] J. W. Bandler and R. E. Seviara, "Sensitivities in terms of wave variables," *Proc. 8th Annual Allerton Conf. on Circuit and System Theory*, 379-387 (Oct., 1970).
- [49] J. W. Bandler and R. E. Seviara, "Wave sensitivities of networks," *IEEE Trans. Microwave Theory and Techniques*, MTT-20, 138-147 (Feb., 1972).
- [50] J. W. Bandler and R. E. Seviara, "Computation of equivalent wave source using the adjoint network," *Electronics Letters*, 7, 235-236 (May, 1971).



**SECTION TWENTY-TWO**  
**CIRCUIT OPTIMIZATION: THE STATE OF THE ART**

© J.W. Bandler and S.H. Chen 1987

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.



# **Circuit Optimization: The State of the Art**

**John W. Bandler**

**Shao Hua Chen**

Reprinted from  
**IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES**  
Vol. 36, No. 2, February 1988

# Circuit Optimization: The State of the Art

JOHN W. BANDLER, FELLOW, IEEE, AND SHAO HUA CHEN, STUDENT MEMBER, IEEE

*Invited Paper*

**Abstract**—This paper reviews the current state of the art in circuit optimization, emphasizing techniques suitable for modern microwave CAD. It is directed at the solution of realistic design and modeling problems, addressing such concepts as physical tolerances and model uncertainties. A unified hierarchical treatment of circuit models forms the basis of the presentation. It exposes tolerance phenomena at different parameter/response levels. The concepts of design centering, tolerance assignment, and postproduction tuning in relation to yield enhancement and cost reduction suitable for integrated circuits are discussed. Suitable techniques for optimization oriented worst-case and statistical design are reviewed. A generalized  $I_p$  centering algorithm is proposed and discussed. Multicircuit optimization directed at both CAD and robust device modeling is formalized. Tuning is addressed in some detail, both at the design stage and for production alignment. State-of-the-art gradient-based nonlinear optimization methods are reviewed, with emphasis given to recent, but well-tested, advances in minimax,  $I_1$ , and  $I_2$  optimization. Illustrative examples as well as a comprehensive bibliography are provided.

## I. INTRODUCTION

COMPUTER-AIDED circuit optimization is certainly one of the most active areas of interest. Its advances continue; hence the subject deserves regular review from time to time. The classic paper by Temes and Calahan in 1967 [102] was one of the earliest to formally advocate the use of iterative optimization in circuit design. Techniques that were popular at the time, such as one-dimensional (single-parameter) search, the Fletcher-Powell procedure and the Remez method for Chebyshev approximation, were described in detail and well illustrated by circuit examples. Pioneering papers by Lasdon, Suchman, and Waren [73], [74], [108] demonstrated optimal design of linear arrays and filters using the penalty function approach. Two papers in 1969 by Director and Rohrer [48], [49] originated the adjoint network approach to sensitivity calculations, greatly facilitating the use of powerful gradient-based optimization methods. In the same period, the work by Bandler [4], [5] systematically treated the formulation of error functions, the least  $p$ th objective, nonlinear constraints, optimization methods, and circuit sensitivity analysis.

Manuscript received May 4, 1987; revised August 20, 1987. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant A7239 and in part by Optimization Systems Associates Inc.

J. W. Bandler is with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7. He is also with Optimization Systems Associates Inc., Dundas, Ontario, Canada L9H 6L1.

S. H. Chen was with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada. He is now with Optimization Systems Associates Inc., Dundas, Ontario, Canada L9H 6L1.  
IEEE Log Number 8717974.

Since then, advances have been made in several major directions. The development of large-scale network simulation and optimization techniques have been motivated by the requirements of the VLSI era. Approaches to realistic circuit design where design parameter tolerances and yield are taken into account have been pioneered by Elias [52] and Karafin [68] and furthered by many authors over the ensuing years. Optimization methods have evolved from simple, low-dimension-oriented algorithms into sophisticated and powerful ones. Highly effective and efficient solutions have been found for a large number of specialized applications. The surveys by Calahan [37], Charalambous [39], Bandler and Rizk [26], Hachtel and Sangiovanni-Vincentelli [63], and Brayton *et al.* [32] are especially relevant to circuit designers.

In the present paper, we concentrate on aspects that are relevant to and necessary for the continuing move to optimization of increasingly more complex microwave circuits, in particular to MMIC circuit modeling and design. Consequently, we emphasize optimization-oriented approaches to deal more explicitly with process imprecision, manufacturing tolerances, model uncertainties, measurement errors, and so on. Such realistic considerations arise from design problems in which a large volume of production is envisaged, e.g., integrated circuits. They also arise from modeling problems in which consistent and reliable results are expected despite measurement errors, structural limitations such as physically inaccessible nodes, and model approximations and simplifications. The effort to formulate and solve these problems represents one of the driving forces of theoretical study in the mathematics of circuit CAD. Another important impetus is provided by progress in computer hardware, resulting in drastic reduction in the cost of mass computation. Finally, the continuing development of gradient-based optimization techniques has provided us with powerful tools.

In this context, we review the following concepts: realistic representations of a circuit design and modeling problem, nominal (single) circuit optimization, statistical circuit design, and multicircuit modeling, as well as recent gradient-based optimization methods.

Nominal design and modeling are the conventional approaches used by microwave engineers. Here, we seek a single point in the space of variables selected for optimization which best meets a given set of performance specifications (in design) or best matches a given set of response measurements (in modeling). A suitable scalar measure



of the deviation between responses and specifications which forms the objective function to be minimized is the ubiquitous least squares measure (see, for example, Morrison [83]), the more esoteric generalized  $l_p$  objective (Charalambous [41]) or the minimax objective (Madsen *et al.* [80]). We observe here that the performance-driven (single-circuit) least squares approach that circuit design engineers have traditionally chosen has proved unsuccessful both in addressing design yield and in serious device modeling.

Recognition that an actual realization of a nominal design is subject to fluctuation or deviation led, in the past, to the so-called sensitivity minimization approach (see, for example, Schoeffler [94] and Laker *et al.* [71]). Employed by filter designers, the approach involves measures of performance sensitivity, typically first-order, that are included in the objective function.

In reality, uncertainties which deteriorate performance may be due to physical (manufacturing, operating) tolerances as well as to parasitic effects such as electromagnetic coupling between elements, dissipation, and dispersion (Bandler [6], Tromp [107]). In the design of substantially untunable circuits these phenomena lead to two important classes of problems: worst-case design and statistical design. The main objective is the reduction of cost or the maximization of production yield.

Worst-case design (Bandler *et al.* [23], [24]), in general, requires that all units meet the design specifications under all circumstances (i.e., a 100 percent yield), with or without tuning, depending on what is practical. In statistical design [1], [26], [30], [47], [97], [98], [100], [101] it is recognized that a yield of less than 100 percent is likely; therefore, with respect to an assumed probability distribution function, yield is estimated and enhanced by optimization. Typically, we either attempt to center the design with fixed assumed tolerances or we attempt to optimally assign tolerances and/or design tunable elements to reduce production cost.

What distinguishes all these problems from nominal designs or sensitivity minimization is the fact that a single design point is no longer of interest: a (tolerance) region of multiple possible outcomes is to be optimally located with respect to the acceptable (feasible, constraint) region.

Modeling, often unjustifiably treated as if it were a special case of design, is particularly affected by uncertainties and errors at many levels. Unavoidable measurement errors, limited accessibility to measurement points, approximate equivalent circuits, etc., result in nonunique and frequently inconsistent solutions. To overcome these frustrations, we advocate a properly constituted multicircuit approach (Bandler *et al.* [12]).

Our presentation is outlined as follows.

In Section II, in relation to a physical engineering system of interest, a typical hierarchy of simulation models and corresponding response and performance functions are introduced. Error functions arising from given specifications and a vector of optimization variables are defined. Performance measures such as  $l_p$  objective functions ( $l_p$

norms and generalized  $l_p$  functions) are introduced and their properties discussed.

We devote to Section III a brief review of the relatively well-known and successful approach of nominal circuit design optimization.

In Section IV, uncertainties that exist in the physical system and at different levels of the model hierarchy are discussed and illustrated by a practical example. Different cases of multicircuit design, namely centering, tolerancing (optimal tolerance assignment), and tuning at the design stage, are identified. A multicircuit modeling approach and several possible applications are described.

Some important and representative techniques in worst-case and statistical design are reviewed in Section V. These include the nonlinear programming approach to worst-case design (Bandler *et al.* [24], Polak [89]), simplicial (Director and Hachtel [47]) and multidimensional (Bandler and Abdel-Malek [7]) approximations of the acceptable region, the gravity method (Soin and Spence [98]), and the parametric sampling method (Singhal and Pinel [97]). A generalized  $l_p$  centering algorithm is proposed as a natural extension to  $l_p$  nominal design. It provides a unified formulation of yield enhancement for both the worst case and the case where yield is less than 100 percent.

Illustrations of statistical design are given in Section VI.

The studies in the last two decades on the theoretical and algorithmic aspects of optimization techniques have produced a great number of results. In particular, gradient-based optimization methods have gained increasing popularity in recent years for their effectiveness and efficiency. The essence of gradient-based  $l_p$  optimization methods is reviewed in Section VII. Emphasis is given to the trust region Gauss-Newton and the quasi-Newton algorithms (Madsen [78], Moré [82], Dennis and Moré [46]).

The subject of gradient calculation and approximation is briefly discussed in Section VIII.

## II. VARIABLES AND FUNCTIONS

In this section, we review some basic concepts of practical circuit optimization. In particular, we identify a physical system and its simulation models. We discuss a typical hierarchy of models and the associated designable parameters and response functions. We also define specifications, error functions, optimization variables and objective functions.

### A. The Physical System

The physical engineering system under consideration can be a network, a device, a process, and so on, which has both a fixed structure and given element types. We manipulate the system through some adjustable parameters contained in the column vector  $\phi^M$ . The superscript  $M$  identifies concepts related to the physical system. Geometrical dimensions such as the width of a strip and the length of a waveguide section are examples of adjustable parameters.

In the production of integrated circuits,  $\phi^M$  may include some fundamental variables which control, say, a doping or photomasking process and, consequently, determine the geometrical and electrical parameters of a chip. External controls, such as the biasing voltages applied to an active device, are also possible candidates for  $\phi^M$ .

The performance and characteristics of the system are described in terms of some measurable quantities. The usual frequency and transient responses are typical examples. These measured responses, or simply measurements, are denoted by  $F^M(\phi^M)$ .

### B. The Simulation Models

In circuit optimization, some suitable models are used to simulate the physical system. Actually, models can be usefully defined at many levels. Tromp [106], [107] has considered an arbitrary number of levels (also see Bandler *et al.* [19]). Here, for simplicity, we consider a hierarchy of models consisting of four typical levels as

$$\begin{aligned} F^H &= F^H(F^L) \\ F^L &= F^L(\phi^H) \\ \phi^H &= \phi^H(\phi^L). \end{aligned} \quad (1)$$

$\phi^L$  is a set of low-level model parameters. It is supposed to represent, as closely as possible, the adjustable parameters in the actual system, i.e.,  $\phi^M$ .  $\phi^H$  defines a higher-level model, typically an equivalent circuit, with respect to a fixed topology. Usually, we use an equivalent circuit for the convenience of its analysis. The relationship between  $\phi^L$  and  $\phi^H$  is either derived from theory or given by a set of empirical formulas.

Next on the hierarchy we define the model responses at two possible levels. The low-level external representation, denoted by  $F^L$ , can be the frequency-dependent complex scattering parameters, unterminated  $y$ -parameters, transfer function coefficients, etc. Although these quantities may or may not be directly measurable, they are very often used to represent a subsystem. The high-level responses  $F^H$  directly correspond to the actual measured responses, namely  $F^M$ , which may be, for example, frequency responses such as return loss, insertion loss, and group delay of a suitably terminated circuit.

A realistic example of a one-section transformer on stripline was originally considered by Bandler *et al.* [25]. The circuits and parameters, physical as well as model, are shown in Fig. 1. The physical parameters  $\phi^M$  (and the low-level model  $\phi^L$ ) include strip widths, section lengths, dielectric constants, and strip and substrate thicknesses. The equivalent circuit has six parameters, considered as  $\phi^H$ , including the effective line widths, junction parasitic inductances, and effective section length. The scattering matrix of the circuit with respect to idealized (matched) terminations is a candidate for a low-level external representation ( $F^L$ ). The reflection coefficient by taking into account the actual complex terminations could be a high-level response of interest ( $F^H$ ).

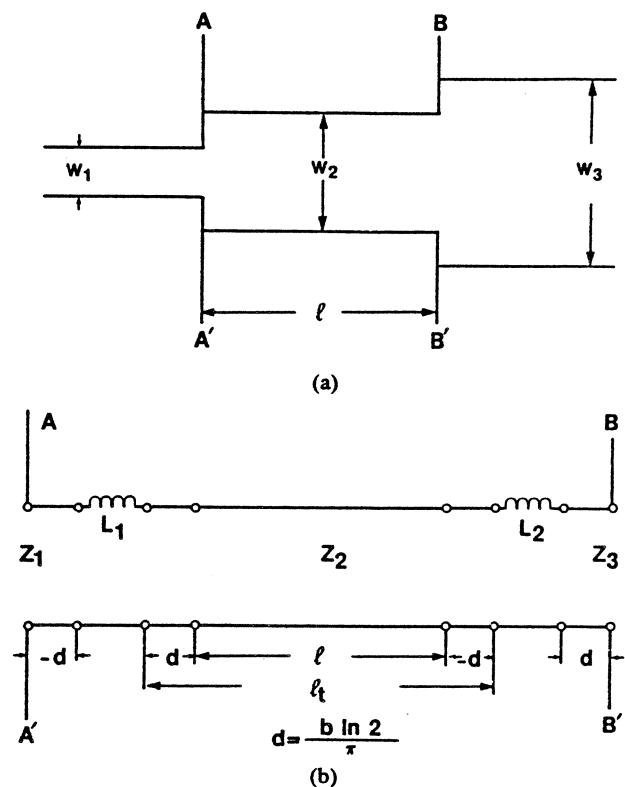


Fig. 1. A microwave stripline transformer showing (a) the physical structure and (b) the equivalent circuit model [25]. The physical parameters are

$$\phi^M = [w_1 w_2 w_3 : l \sqrt{\epsilon_{r1}} \sqrt{\epsilon_{r2}} \sqrt{\epsilon_{r3}} b_1 b_2 b_3 t_{s1} t_{s2} t_{s3}]^T$$

where  $w$  is the strip width,  $l$  the length of the middle section,  $\epsilon_r$  the dielectric constant,  $b$  the substrate thickness, and  $t_s$  the strip thickness.  $\phi^M$  is represented in the simulation model by  $\phi^L$ . The high-level parameters of the equivalent circuit are

$$\phi^H = [D_1 D_2 D_3 L_1 L_2 l_t]^T$$

where  $D$  is the effective linewidth,  $L$  the junction parasitic inductance, and  $l_t$  the effective section length. Suitable empirical formulas that relate  $\phi^L$  to  $\phi^H$  can be found in [25].

For a particular case, we may choose a certain section of this hierarchy to form a design problem. We can choose either  $\phi^L$  or  $\phi^H$  as the designable parameters. Either  $F^L$  or  $F^H$  or a suitable combination of both may be selected as the response functions. Bearing this in mind, we simplify the notation by using  $\phi$  for the designable parameters and  $F$  for the response functions.

### C. Specifications and Error Functions

The following discussion on specifications and error functions is based on presentations by Bandler [5], and Bandler and Rizk [26], where more exhaustive illustrations can be found.

We express the desirable performance of the system by a set of specifications which are usually functions of certain independent variable(s) such as frequency, time, and temperature. In practice, we have to consider a discrete set of samples of the independent variable(s) such that satisfying the specifications at these points implies satisfying them

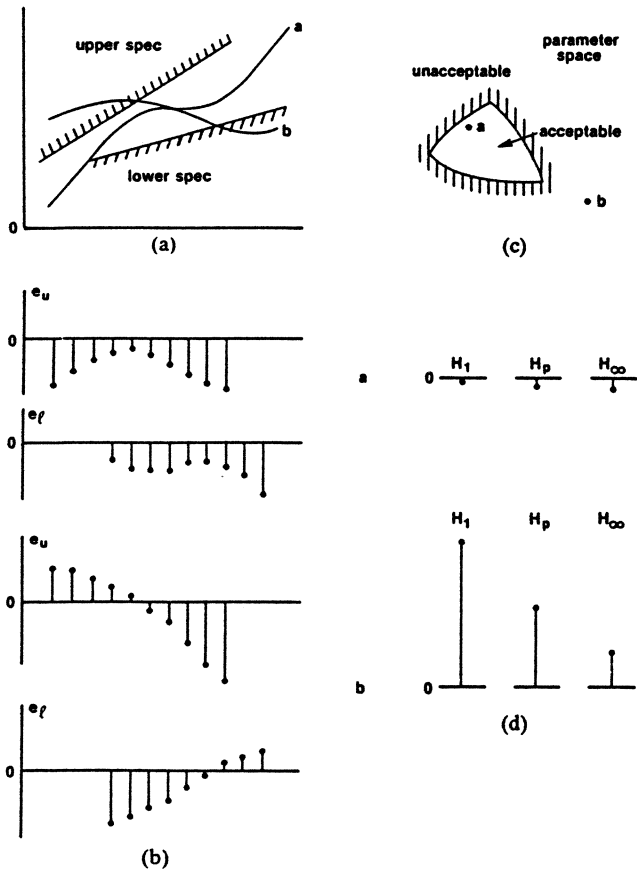


Fig. 2. Illustrations of (a) upper specifications, lower specifications, and the responses of circuits *a* and *b*, (b) error functions corresponding to circuits *a* and *b*, (c) the acceptable region, and (d) generalized  $l_p$  objective functions defined in (13).

almost everywhere. Also, we may consider simultaneously more than one kind of response. Thus, without loss of generality, we denote a set of sampled specifications and the corresponding set of calculated response functions by, respectively,

$$\begin{aligned} S_j, & \quad j=1,2,\dots,m \\ F_j(\phi), & \quad j=1,2,\dots,m. \end{aligned} \quad (2)$$

Error functions arise from the difference between the given specifications and the calculated responses. In order to formulate the error functions properly, we may wish to distinguish between having upper and lower specifications (windows) and having single specifications, as illustrated in Figs. 2(a) and 3(a). Sometimes the one-sidedness of upper and lower specifications is quite obvious, as in the case of designing a bandpass filter. On other occasions the distinction is more subtle, since a single specification may as well be interpreted as a window having zero width.

In the case of having single specifications, we define the error functions by

$$e_j(\phi) = w_j |F_j(\phi) - S_j|, \quad j=1,2,\dots,m \quad (3)$$

where  $w_j$  is a nonnegative weighting factor.

We may also have an upper specification  $S_{uj}$  and a lower specification  $S_{lj}$ . In this case we define the error

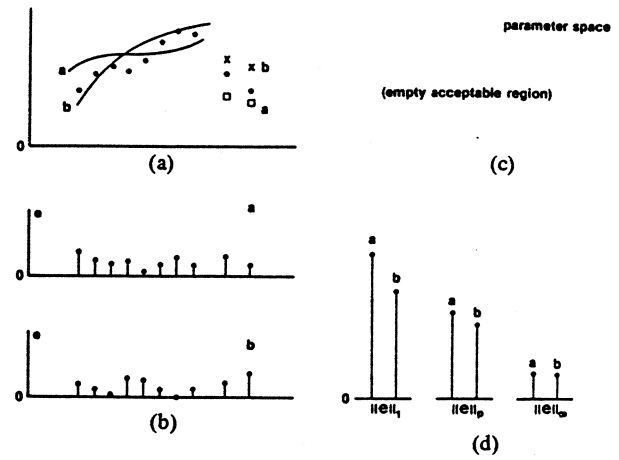


Fig. 3. Illustrations of (a) a discretized single specification and two discrete single specifications (e.g., expected parameter values to be matched), as well as the responses of circuits *a* and *b*, (b) error functions related to circuits *a* and *b*, (c) the (empty) acceptable region (i.e., a perfect match is not possible) and (d) the corresponding  $l_p$  norms.

functions as

$$\begin{aligned} e_{uj}(\phi) &= w_{uj} (F_j(\phi) - S_{uj}), & j \in J_u \\ e_{lj}(\phi) &= w_{lj} (F_j(\phi) - S_{lj}), & j \in J_l \end{aligned} \quad (4)$$

where  $w_{uj}$  and  $w_{lj}$  are nonnegative weighting factors. The index sets as defined by

$$\begin{aligned} J_u &= \{j_1, j_2, \dots, j_k\} \\ J_l &= \{j_{k+1}, j_{k+2}, \dots, j_m\} \end{aligned} \quad (5)$$

are not necessarily disjoint (i.e., we may have simultaneous specifications). In order to have a set of uniformly indexed error functions, we let

$$\begin{aligned} e_i &= e_{uj}(\phi), & j = j_i, \quad i = 1, 2, \dots, k \\ e_i &= -e_{lj}(\phi), & j = j_i, \quad i = k+1, k+2, \dots, m. \end{aligned} \quad (6)$$

The responses corresponding to the single specifications can be real or complex, whereas upper and lower specifications are applicable to real responses only. Notice that, in either case, the error functions are real. Clearly, a positive (nonpositive) error function indicates a violation (satisfaction) of the corresponding specification. Figs. 2(b) and 3(b) depict the concept of error functions.

#### D. Optimization Variables and Objective Functions

Mathematically, we abstract a circuit optimization problem by the following statement:

$$\underset{x}{\text{minimize}} U(x) \quad (7)$$

where  $x$  is a set of optimization variables and  $U(x)$  a scalar objective function.

Optimization variables and model parameters are two separate concepts. As will be elaborated on later in this paper,  $x$  may contain a subset of  $\phi$  which may have been normalized or transformed, it may include some statistical variables of interest, several parameters in  $\phi$  may be tied to one variable in  $x$ , and so on.

Typically, the objective function  $U(x)$  is closely related to an  $l_p$  norm or a generalized  $l_p$  function of  $e(\phi)$ . We shall review the definitions of such  $l_p$  functions and discuss their appropriate use in different contexts.

### E. The $l_p$ Norms

The  $l_p$  norm (Temes and Zai [103]) of  $e$  is defined as

$$\|e\|_p = \left[ \sum_{j=1}^m |e_j|^p \right]^{1/p} \quad (8)$$

It provides a scalar measure of the deviations of the model responses from the specifications. Least-squares ( $l_2$ ) is perhaps the most well-known and widely used norm (Morrison [83]), which is

$$\|e\|_2 = \left[ \sum_{j=1}^m |e_j|^2 \right]^{1/2} \quad (9)$$

The  $l_2$  objective function is differentiable and its gradient can be easily obtained from the partial derivatives of  $e$ . Partly due to this property, a large variety of  $l_2$  optimization techniques have been developed and popularly implemented. For example, the earlier versions of the commercial CAD packages TOUCHSTONE [104] and SUPER-COMPACT [99] have provided designers solely the least-squares objective.

The parameter  $p$  has an important implication. By choosing a large (small) value for  $p$ , we in effect place more emphasis on those error functions ( $e_j$ 's) that have larger (smaller) values. By letting  $p = \infty$  we have the minimax norm

$$\|e\|_\infty = \max_j |e_j| \quad (10)$$

which directs all the attention to the worst case and the other errors are in effect ignored. Minimax optimization is extensively employed in circuit design where we wish to satisfy the specifications in an optimal equal-ripple manner [3], [13], [14], [21], [40], [42], [65], [67], [80], [85].

On the other hand, the use of the  $l_1$  norm, as defined by

$$\|e\|_1 = \sum_{j=1}^m |e_j| \quad (11)$$

implies attaching more importance to the error functions that are closer to zero. This property has led to the application of  $l_1$  to data-fitting in the presence of gross errors [22], [29], [66], [86] and, more recently, to fault location [8], [9], [27] and robust device modeling [12].

Notice that neither  $\|e\|_\infty$  nor  $\|e\|_1$  is differentiable in the ordinary sense. Therefore, their minimization requires algorithms that are much more sophisticated than those for the  $l_2$  optimization.

### F. The One-Sided and Generalized $l_p$ Functions

By using an  $l_p$  norm, we try to minimize the errors towards a zero value. In cases where we have upper and lower specifications, a negative value of  $e_j$  simply indicates that the specification is exceeded at that point which, in a

sense, is better than having  $e_j = 0$ . This fact leads to the one-sided  $l_p$  function defined by

$$H_p^+(e) = \left[ \sum_{j \in J} |e_j|^p \right]^{1/p} \quad (12)$$

where  $J = \{j | e_j \geq 0\}$ . Actually, if we define  $e_j^+ = \max\{e_j, 0\}$ , then  $H_p^+(e) = \|e^+\|_p$ .

Bandler and Charalambous [10], [41] have proposed the use of a generalized  $l_p$  function defined by

$$H_p(e) = \begin{cases} H_p^+(e) & \text{if the set } J \text{ is not empty} \\ H_p^-(e) & \text{otherwise} \end{cases} \quad (13)$$

where

$$H_p^-(e) = - \left[ \sum_{j=1}^m (-e_j)^{-p} \right]^{-1/p} \quad (14)$$

In other words, when at least one of the  $e_j$  is nonnegative we use  $H_p^+$ , and  $H_p^-$  is defined if all the error functions have become negative.

Compared to (12), the generalized  $l_p$  function has an advantage in the fact that it is meaningfully defined for the case where all the  $e_j$  are negative. This permits its minimization to proceed even after all the specifications have been met, so that the specifications may be further exceeded.

A classical example is the design of Chebyshev-type bandpass filters, where we have to minimize the generalized minimax function

$$H_\infty(e) = \max_j \{e_j\}. \quad (15)$$

The current Version 1.5 of TOUCHSTONE [105] offers the generalized  $l_p$  optimization techniques, including minimax.

### G. The Acceptable Region

We use  $H(e)$  as a generic notation for  $\|e\|_p$ ,  $H_p^+(e)$ , and  $H_p^-(e)$ . The sign of  $H(e(\phi))$  indicates whether or not all the specifications are satisfied by  $\phi$ . An acceptable region is defined as

$$R_a = \{ \phi | H(e(\phi)) \leq 0 \} \quad (16)$$

Figs. 2(c), 2(d), 3(c), and 3(d) depict the  $l_p$  functions and the acceptable regions.

## III. NOMINAL CIRCUIT OPTIMIZATION

In a nominal design, without considering tolerances (i.e., assuming that modeling and manufacturing can be done with absolute accuracy), we seek a single set of parameters, called a nominal point and denoted by  $\phi^0$ , which satisfies the specifications. Furthermore, if we consider the functional relationship of  $\phi^H = \phi^H(\phi^L)$  to be precise, then it does not really matter at which level the design is conceived. In fact, traditionally it is often oriented to an equivalent circuit. A classical case is network synthesis

where  $\phi^{H,0}$  is obtained through the use of an equivalent circuit and/or a transfer function. A low-level model  $\phi^{L,0}$  is then calculated from  $\phi^{H,0}$ , typically with the help of an empirical formula (e.g., the number of turns of a coil is calculated for a given inductance). Finally, we try to realize  $\phi^{L,0}$  by its physical counterpart  $\phi^{M,0}$ .

With the tool of mathematical optimization, the nominal point  $\phi^0$  (at a chosen level) is obtained through the minimization of  $U(x)$ , where the objective function is typically defined as an  $l_p$  function  $H(e)$ . The vector  $x$  contains all the elements of or a subset of the elements of  $\phi^0$ . It is a common practice to have some of the variables normalized. It is also common to have several model parameters tied to a single variable. This is true, e.g., for symmetrical circuit structures but, most importantly, it is a fact of life in integrated circuits. Indeed, such dependencies should be taken into account both in design and in modeling to reduce the dimensionality. The minimax optimization of manifold multiplexers as described by Bandler *et al.* [18], [22], [28] provides an excellent illustration of large-scale nominal design of microwave circuits.

Traditionally, the approach of nominal design has been extended to solving modeling problems. A set of measurements made on the physical system serves as single specifications. Error functions are created from the differences between the calculated responses  $F(\phi^0)$  and the measured responses  $F^M$ . By minimizing an  $l_p$  norm of the error functions, we attempt to identify a set of model parameters  $\phi^0$  such that  $F(\phi^0)$  best matches  $F^M$ . This is known as data fitting or parameter identification.

Such a casual treatment of modeling as if it were a special case of design is often unjustifiable, due to the lack of consideration to the uniqueness of the solution. In design, one satisfactory nominal point, possibly out of many feasible solutions, may suffice. In modeling, however, the uniqueness of the solution is almost always essential to the problem. Affected by uncertainties at many levels, unavoidable measurement errors and limited accessibility to measurement points, the model obtained by a nominal optimization is often nonunique and unreliable. To overcome these frustrations, a recent multicircuit approach will be described in Section IV.

#### IV. A MULTICIRCUIT APPROACH

The approach of nominal circuit optimization, which we have described in Section III, focuses attention on a certain kind of idealized situation. In reality, unfortunately, there are many uncertainties to be accounted for. For the physical system, without going into too many details, consider

$$\begin{aligned} F^M &= F^{M,0}(\phi^M) + \Delta F^M \\ \phi^M &= \phi^{M,0} + \Delta\phi^M \end{aligned} \quad (17)$$

where  $\Delta F^M$  represents measurement errors,  $\phi^{M,0}$  a nominal value for  $\phi^M$ , and  $\Delta\phi^M$  some physical (manufacturing, operating) tolerances.

For simulation purposes, we may consider a realistic representation of the hierarchy of possible models as

$$\begin{aligned} F^H &= F^{H,0}(F^L) + \Delta F^H \\ F^L &= F^{L,0}(\phi^H) + \Delta F^L \\ \phi^H &= \phi^{H,0}(\phi^L) + \Delta\phi^H \\ \phi^L &= \phi^{L,0} + \Delta\phi^L \end{aligned} \quad (18)$$

where  $\phi^{L,0}$ ,  $\phi^{H,0}$ ,  $F^{L,0}$ , and  $F^{H,0}$  are nominal models applicable at different levels.  $\Delta\phi^L$ ,  $\Delta\phi^H$ ,  $\Delta F^L$ , and  $\Delta F^H$  represent uncertainties or inaccuracies associated with the respective models.  $\Delta\phi^L$  corresponds to the tolerances  $\Delta\phi^M$ .  $\Delta\phi^H$  may be due to the approximate nature of an empirical formula. Parasitic effects which are not adequately modeled in  $\phi^H$  will contribute to  $\Delta F^L$ , and finally we attribute anything else that causes a mismatch between  $F^{H,0}$  and  $F^{M,0}$  to  $\Delta F^H$ .

These concepts can be illustrated by the one-section stripline transformer example [25] which we have considered in Section II. Tolerances may be imposed on the physical parameters including the strip widths and thicknesses, the dielectric constants, the section length and substrate thicknesses (see Fig. 1). Such tolerances correspond to  $\Delta\phi^M$  and are represented in the model by  $\Delta\phi^L$ . We may also use  $\Delta\phi^H$  to represent uncertainties associated with the empirical formulas which relate the physical parameters to the equivalent circuit parameters (the effective line widths, the junction inductances, and the effective section length). Mismatches in the terminations at different frequencies may be estimated by  $\Delta F^H$  ( $F^H$  being the actual reflection coefficient; see [25] for more details).

The distinction between different levels of model uncertainties can be quite subtle. As an example, consider the parasitic resistance  $r$  associated with an inductor whose inductance is  $L$ . Both  $L$  and  $r$  are functions of the number of turns of a coil (which is a physical parameter). Depending on whether or not  $r$  is modeled by the equivalent circuit (i.e., whether or not  $r$  is included in  $\phi^H$ ), the uncertainty associated with  $r$  may appear in  $\Delta\phi^H$  or in  $\Delta F^L$ .

When such uncertainties are present, a single nominal model often fails to represent satisfactorily the physical reality. One effective solution to the problem is to simultaneously consider multiple circuits. We discuss the consequences for design and modeling separately.

##### A. Multicircuit Design

Our primary concern is to improve production yield and reduce cost in the presence of tolerances  $\Delta\phi^L$  and model uncertainties  $\Delta\phi^H$ . First of all, we represent a realistic situation by multiple circuits as

$$\phi^k = \phi^0 + s^k, \quad k = 1, 2, \dots, K \quad (19)$$

where  $\phi^0$ ,  $\phi^k$ , and  $s^k$  are generic notation for the nominal parameters, the  $k$ th set of parameters, and a deviate due to the uncertainties, respectively. A more elaborate definition is developed as we proceed.

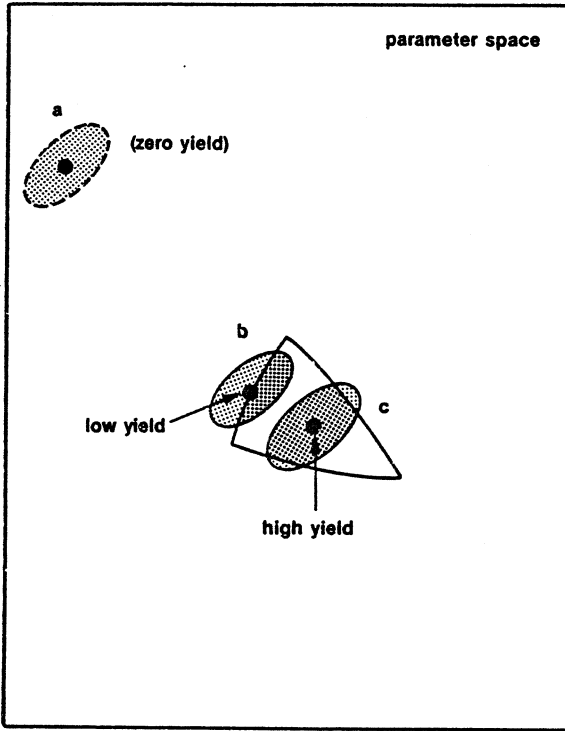


Fig. 4. Three nominal points and the related yield.

For each circuit, we define an acceptance index by

$$I_a(\phi) = \begin{cases} 1, & \text{if } H(e(\phi)) \leq 0 \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

where  $H(e) \leq 0$ , defined in (13), indicates satisfaction of the specifications by  $\phi$ . An estimate of the yield is given by the percentage of acceptable samples out of the total, as

$$Y = \left[ \sum_{k=1}^K I_a(\phi^k) \right] / K. \quad (21)$$

The merit of a design can then be judged more realistically according to the yield it promises, as illustrated in Fig. 4. Now we shall have a closer look at the definition of multiple circuits.

In the Monte Carlo method the deviates  $s^k$  are constructed by generating random numbers using a physical process or arithmetical algorithms. Typically, we assume a statistical distribution for  $\Delta\phi^L$ , denoted by  $D^L(\epsilon^L)$  where  $\epsilon^L$  is a vector of tolerance variables. For example, we may consider a multidimensional uniform distribution on  $[-\epsilon^L, \epsilon^L]$ . Similarly, we assume a  $D^H(\epsilon^H)$  for  $\Delta\phi^H$ . The uniform and Gaussian (normal) distributions are illustrated in Fig. 5.

At the low level, consider

$$\phi^{L,k} = \phi^{L,0} + s^{L,k}, \quad k=1,2,\dots,K^L \quad (22)$$

where  $s^{L,k}$  are samples from  $D^L$ . At the higher level, we have, for each  $k$ ,

$$\phi^{H,k,i} = \phi^{H,0} + s^{H,k,i}, \quad i=1,2,\dots,K^H \quad (23)$$

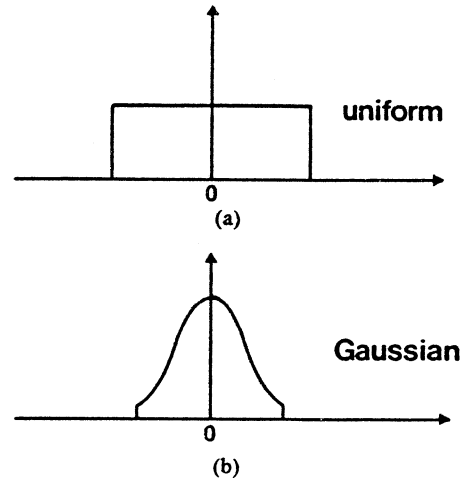


Fig. 5. Typical tolerance distributions: uniform and Gaussian (normal).

where

$$\begin{aligned} \phi^{H,0} &= \phi^{H,0}(\phi^{L,0}) \\ s^{H,k,i} &= \phi^{H,0}(\phi^{L,k}) - \phi^{H,0}(\phi^{L,0}) + \delta^{k,i} \end{aligned} \quad (24)$$

with  $\delta^{k,i}$  being samples from  $D^H$ .

One might propose a distribution for  $s^{H,k,i}$  which presumably encompasses the effect of distribution  $D^L$  and distribution  $D^H$ . But, while we may reasonably assume simple and independent distributions for  $\Delta\phi^L$  and  $\Delta\phi^H$ , the compound distribution is likely to be complicated and correlated.

### B. Centering, Tolerancing, and Tuning

Again, in order to simplify the notation, we use  $\phi^0$  for the nominal circuit and  $\epsilon$  for the tolerance variables.

An important problem involves design centering with fixed tolerances, usually relative to corresponding nominal values. We call this the fixed tolerance problem (FTP). The optimization variables are elements of  $\phi^0$ , the elements of  $\epsilon$  are constant or dependent on the variables, and the objective is to improve the yield. Incidentally, the nominal optimization problem, i.e., the traditional design problem, is sometimes referred to as the zero tolerance problem (ZTP).

Since imposing tight tolerances on the parameters will increase the cost of component fabrication or process operation, we may attempt to maximize the allowable tolerances subject to an acceptable yield. In this case both  $\phi^0$  and  $\epsilon$  may be considered as variables. Such a problem is referred to as optimal tolerancing, optimal tolerance assignment, or the variable tolerance problem (VTP).

Tuning some components of  $\phi^M$  after production, whether by the manufacturer or by a customer, is quite commonly used as a means of improving the yield. This process can also be simulated using the model by introducing a vector of designable tuning adjustments  $\tau^k$  for each circuit, as

$$\phi^k = \phi^0 + s^k + \tau^k, \quad k=1,2,\dots,K. \quad (25)$$

We have to determine, through optimization, the value of

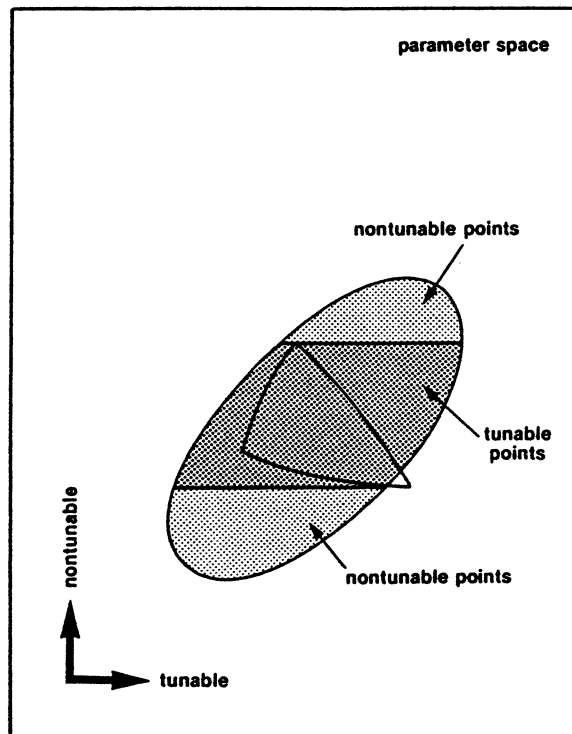
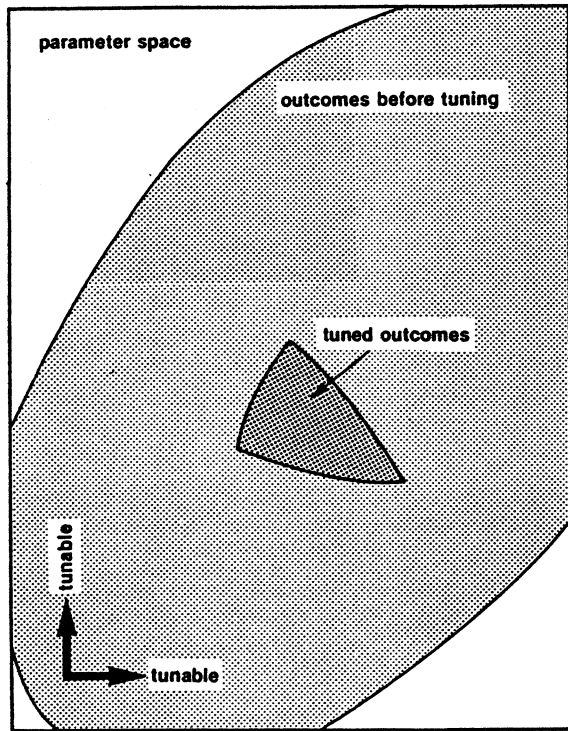


Fig. 6. Illustrations of tuning: (a) both parameters are tunable for a case in which the probability that an untuned design meets the specifications is very low and (b) only one parameter is tunable.

$\tau^k$  such that the specifications will be satisfied at  $\phi^k$  which may otherwise be unacceptable, as depicted in Figs. 6 and 7. The introduction of tuning, on the other hand, also increases design complexity and manufacturing cost. We seek a suitable compromise by solving an optimization problem in which  $\tau^k$  are treated as part of the variables.

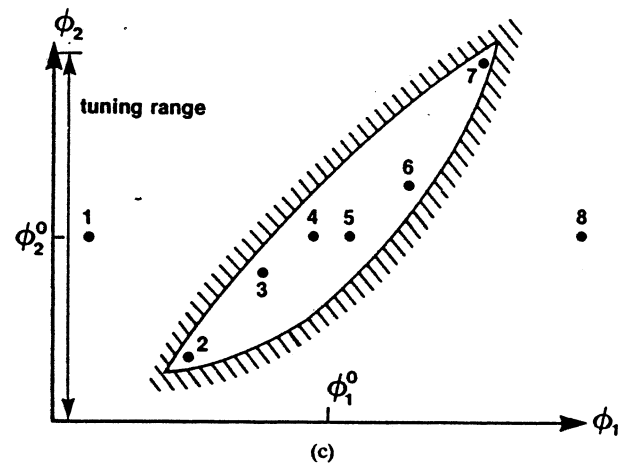
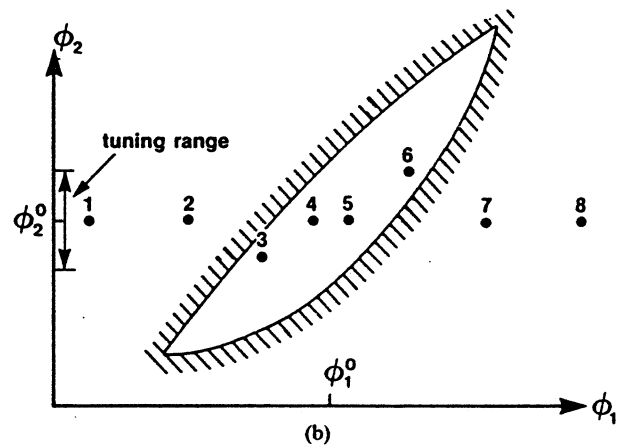
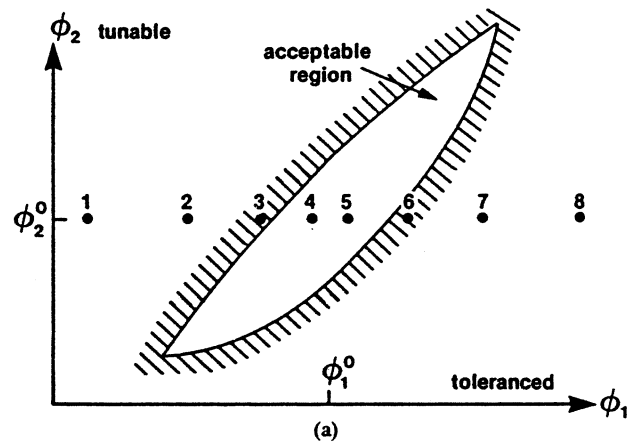


Fig. 7. An illustration of multicircuit design considering eight circuit outcomes.  $\phi_1$  is tolerated and  $\phi_2$  is tunable. (a) Without tuning the yield is 2/8 (25 percent). (b) Tuning on  $\phi_2$  is restricted to a small range. The improved yield is 4/8 (50 percent). (c) A 75 percent yield is achieved by allowing a large tuning range.

From nominal design, centering, optimal tolerancing, to optimal tuning, we have defined a range of problems which lead to increasingly improved yield but, on the other hand, correspond to increasing complexity. Some specific formulations are discussed in Section V. Analogously to ZTP, FTP, and VTP, we can define zero tuning, fixed tuning, and variable tuning problems [20].

### C. Multicircuit Modeling

The uncertainties that affect circuit modeling can be discussed under the following categories.

- 1) Measurement errors will inevitably exist in practice, as represented by  $\Delta F^M$  in (17):  $F^M = F^{M,0}(\phi^M) + \Delta F^M$ .
- 2) Even without measurement errors, the calculated response  $F^{H,0}$  may never be able to match  $F^{M,0}$  perfectly, due to, for example, the use of a model of insufficient order or inadequate complexity. Such an inherent mismatch is accounted for in (18) by  $F^H = F^{H,0} + \Delta F^H$ .
- 3) Even if neither  $\Delta F^M$  nor  $\Delta F^H$  exists so that  $F^{H,0} = F^M$ , we may still not be able to uniquely identify  $\phi$  from the set of measurements that has been selected. This happens when the system of (generally nonlinear) equations  $F^{H,0}(\phi) - F^M = 0$ , where  $F^M$  is the data, is underdetermined. Typically, this problem occurs when, for any reason, many internal nodes are inaccessible to direct measurement. An overcomplicated equivalent circuit, including unknown parasitic elements, is frequently at the heart of this phenomenon.
- 4) The parasitic effects that are not adequately modeled by  $\phi^H$  contribute to the uncertainty  $\Delta F^L$ . This is another source of interference with the modeling process.

First we consider the case in which modeling is applied to obtain a suitable  $\phi$  such that  $F^H(\phi)$  approximates  $F^M$ . The nominal circuit approach may be able to cope with the uncertainties in 1) and 2), and comes up with a  $\phi$  which minimizes the errors  $\Delta F^M$  and  $\Delta F^H$  in a certain sense. But it will not be able to overcome the problem of uniqueness. In practice, we are often unable to determine unambiguously the identifiability of a system, because all these uncertainties can be present at the same time. There will be, typically, a family of solutions which produce reasonable and similar matches between the measured and the calculated responses. We cannot, therefore, rely on any particular set of parameters.

The approach of multicircuit modeling by Bandler *et al.* [12] can be used to overcome these difficulties. Multiple circuits are created by making deliberate adjustments on the physical parameters  $\phi^M$ . For example, we can change the biasing conditions for an active device and obtain multiple sets of measurements. By doing so, we introduce perturbations to the model which cause some parameters in  $\phi$  to change by an unknown amount. For this approach to be successful, each physical adjustment should produce changes in only a few parameters in  $\phi$ .

Although we do not know the changes in  $\phi$  quantitatively, it is often possible to identify which model parameters may have been affected by the physical adjustments. Such a qualitative knowledge may be apparent from the definition of the model or it may come from practical experience. In the attempt to process multiple circuits

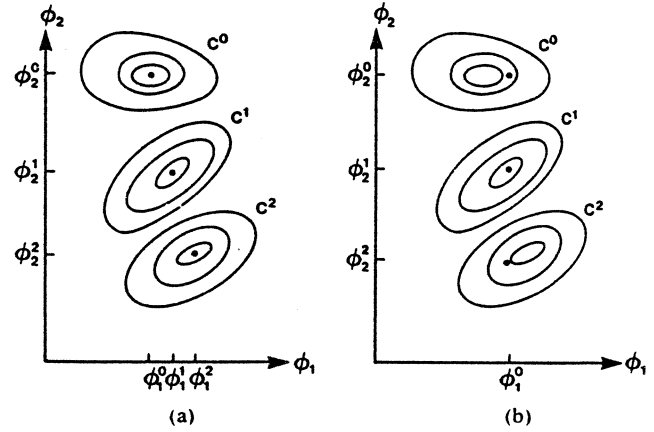


Fig. 8. An illustration of multicircuit modeling. Three circuits are created by making two physical adjustments. Assume that we know that  $\phi_1$  should not be affected by the physical adjustments.  $C^0$ ,  $C^1$ , and  $C^2$  are contours of the error functions corresponding to the three circuits. (a) By treating the three circuits separately, we obtain  $\phi_1^0$ ,  $\phi_1^1$ , and  $\phi_1^2$ .  $\phi_1^0$ ,  $\phi_1^1$ , and  $\phi_1^2$  turn out to have different values (which is inconsistent with our knowledge) because of uncertainties. (b) Consistent results can be obtained by defining  $\phi_1$  as a common variable and processing three circuits simultaneously.

simultaneously, we define those model parameters that are not supposed to change as common variables and, at the same time, allow the others to vary between different circuits. By doing so, we force the solution to exhibit the desired consistency and, therefore, improve the reliability of the result. In other words, from a family of possible solutions we select the one that conforms to the topological constraints. Bandler *et al.* have shown an example [12, Section III-A] in which  $\phi$  can not be uniquely identified due to inaccessible nodes. The problem was effectively addressed using the multicircuit approach.

To formulate this mathematically, let

$$\phi^k = \begin{bmatrix} \phi_c^k \\ \phi_a^k \end{bmatrix} \quad (26)$$

where  $\phi_c^k$  contains the common variables and  $\phi_a^k$  contains the variables which are allowed to vary between the  $k$ th circuit and the reference circuit  $\phi^0$ . We then define the optimization variables by

$$x = [(\phi^0)^T (\phi_a^1)^T \cdots (\phi_a^K)^T]^T \quad (27)$$

and state the optimization problem as to

$$\underset{x}{\text{minimize}} U(x) = \|f\|_p \quad (28)$$

where

$$f = [e^T(\phi^0) e^T(\phi^1) \cdots e^T(\phi^K)]^T. \quad (29)$$

Although any  $l_p$  norm may be used, the unique property of  $l_1$  discussed in detail by Bandler *et al.* [12] can be exploited to great advantage. The concept of common and independent variables is depicted in Fig. 8.

Now, suppose that we do not have a clear idea about which model parameters may have been affected by the



adjustment on  $\phi^M$ . In this case, we let

$$x = [(\phi^0)^T (\phi^1)^T \cdots (\phi^K)^T]^T \quad (30)$$

and change the objective function to an  $l_p$  norm of

$$f = [e^T(\phi^0) \cdots e^T(\phi^K) \alpha_1(\phi^1 - \phi^0)^T \cdots \alpha_K(\phi^K - \phi^0)^T]^T \quad (31)$$

where  $\alpha_1, \alpha_2, \dots, \alpha_K$  are nonnegative multipliers (weights).

Using this formulation, while minimizing the errors  $e$ , we penalize the objective function for any deviates between  $\phi^k$  and  $\phi^0$ , since our only available knowledge is that only a few parameters in  $\phi^k$  should have any significant changes. To be effective, an  $l_1$  norm should be used. A similar principle has been successfully applied to the analog circuit fault location problem [9], [27].

A practical application to FET modeling has been described by Bandler *et al.* in [16], where multiple circuits were created by taking three sets of actual measurements under different biasing conditions.

Another important application of multicircuit modeling is to create analytical formulas which link the model  $\phi$  to the actual physical parameters  $\phi^M$ . Such formulas will become extremely useful in guiding an actual production alignment or tuning procedure. A sequence of adjustments on  $\phi^M$  can be systematically made and multiple sets of measurements are taken. By nominal circuit optimization, these measurements would be processed separately to obtain a set of static models. In the presence of uncertainties, a single change in  $\phi^M$  may seem to cause fluctuations in all the model parameters. Obviously, such results are of very little use. In contrast, multicircuit modeling is more likely to produce models that are consistent and reliable. Since the measurements are made systematically, it certainly makes sense to process them simultaneously. Actually, the variables need not be equivalent circuit model parameters. They can include coefficients of a proposed formula as well.

An example of establishing an experimental relationship between the physical and model parameters for a multicircuit filter using multiple sets of actual measurements has been described by Daijavad [44].

The multicircuit approach can also be applied to model verification. This is typically related to cases where the parasitic uncertainty  $\Delta F^L$  has put the validity of a model in doubt. Instead of defining common and independent variables explicitly, we use the formulation of (30) and (31). If consistent results are obtained, then our confidence in the model is strengthened. Otherwise we should probably reject the current model and consider representing the parasitics more adequately. A convincing example has been demonstrated by Bandler *et al.* [12, section V, test 2].

The commercial packages TOUCHSTONE [104], [105] and SUPER-COMPACT [99] allow a hierarchy of circuit blocks and permit the use of variable labels. Multiple circuits and common variables can be easily defined utilizing these features.

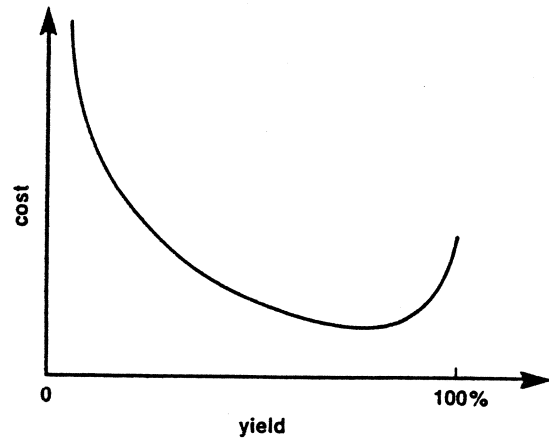


Fig. 9. A typical cost-versus-yield curve [97].

## V. TECHNIQUES FOR STATISTICAL DESIGN

In Section IV we have generally discussed uncertainties at different levels, and, in particular, we have expressed our desire to maximize yield in the presence of uncertainties. Optimal tolerancing and tuning have also been identified as means to further reduce cost in the actual production.

We begin this section with a review of some existing techniques for statistical design. Some of the earliest work in this area came from Karafin [68], Pinel and Roberts [87], Butler [36], Elias [52], Bandler, Liu, and Tromp [24]. During the years, significant contributions have been made by, among others, Director and Hachtel [47] (the simplicial method), Sojn and Spence [98] (the gravity method), Bandler and Abdel-Malek [1], [2], [7] (multidimensional approximation), Biernacki and Styblinski [30] (dynamic constraint approximation), Polak and Sangiovanni-Vincentelli [90] (a method using outer approximation), as well as Singhal and Pinel [97] (the parametric sampling method). Following the review, we propose a generalized  $l_p$  centering algorithm.

A commonly assumed cost versus yield curve [97] is shown in Fig. 9. Actually, hard data are difficult to obtain, and, as we shall see, rather abstract objective functions are often selected for the tolerance-yield design problem. Fig. 10 shows a design with a 100 percent yield and a second design corresponding to the minimum cost.

### A. Worst-Case Design

By this approach, we attempt to achieve a 100 percent yield. Since it means that the specifications have to be satisfied for all the possible outcomes, we need to consider only the worst cases.

Bandler *et al.* [23], [24] have formulated it as a nonlinear programming problem

$$\begin{aligned} & \underset{x}{\text{minimize}} C(x) \\ & \text{subject to } e(\phi^k) \leq 0, \quad \text{for all } k \end{aligned} \quad (32)$$

where  $C(x)$  is a suitable cost function and the points  $\phi^k$

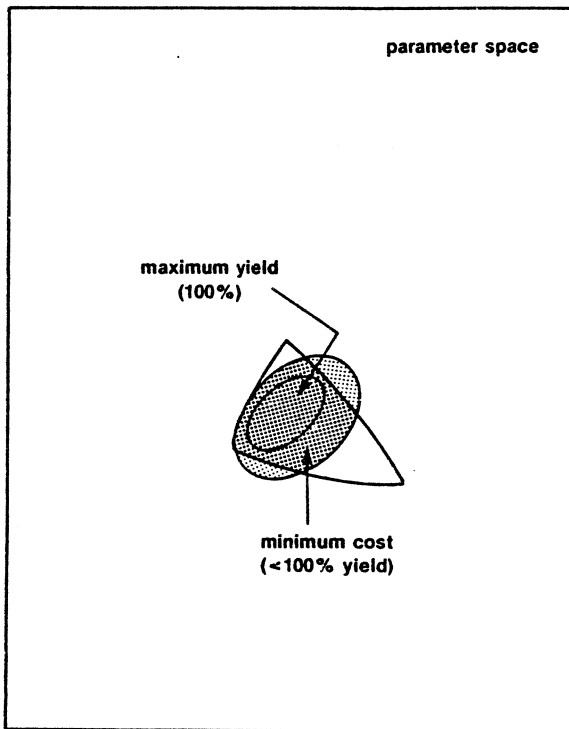


Fig. 10. A maximum yield design and a minimum cost design.

are the worst cases. For instance, we may have

$$C(x) = \sum_{i \in I_e} \frac{a_i}{\epsilon_i} + \sum_{i \in I_t} b_i t_i \quad (33)$$

where  $I_e$  and  $I_t$  are index sets identifying the tolerated and tunable parameters, respectively.  $\epsilon_i$  and  $t_i$  are the tolerance and the tuning range, respectively, associated with the  $i$ th parameter.  $a_i$  and  $b_i$  are nonnegative weights. A cost function can also be defined for relative tolerances and tuning by including  $\phi^0$  into (33). A critical part of this approach is the determination of the worst cases. Vertices of the tolerance region, for example, are possible candidates for the worst cases by assuming one-dimensional convexity. The yield function does not enter (32) explicitly; instead, a 100 percent yield is implied by a feasible solution.

Bandler and Charalambous [11] have demonstrated a solution to (32) by minimax optimization. Polak and Sangiovanni-Vincentelli [90] have proposed a different but equivalent formulation which involves a nondifferentiable optimization.

A worst-case design is not always appropriate. While attempting to obtain a 100 percent yield, the worst-case approach may necessitate unrealistically tight tolerances, or demand excessive tuning. In either case, the cost may be too high. A perfect 100 percent yield may not even be realizable.

### B. Methods of Approximating the Acceptable Region

Since yield is given by the percentage of model outcomes that fall into the acceptable region, we may wish to

find an approximation to that region. The acceptable region has been defined in (16) as  $R_a = \{\phi | H(e(\phi)) \leq 0\}$ .

Director and Hachtel [47] have devised a simplicial approximation approach. It begins by determining points  $\phi^k$  on the boundary of  $R_a$  which is given by  $\Omega_a = \{\phi | H(e(\phi)) = 0\}$ . The convex hull of these points forms a polyhedron. The largest hypersphere inscribed within the polyhedron gives an approximation to  $R_a$  and is found by solving a linear programming problem. Using line searches, more points on the boundary are located and the polyhedron is expanded. The process thus provides a monotonically increasing lower bound on the yield. The center and radius of the hypersphere can be used to determine the centered nominal point and the tolerances, respectively. The application of this method is, however, severely limited by the assumption of a convex acceptable region.

Bandler and Abdel-Malek [1], [2], [7] have presented a method which approximates each  $e_j(\phi)$  by a low-order multidimensional polynomial. Model simulations are performed at some  $\phi^k$  selected around a reference point. From the values of  $e_j(\phi^k)$  the coefficients of the approximating polynomial are determined by solving a linear system of equations. Appropriate linear cuts are constructed to approximate the boundary  $\Omega_a$ . The yield is estimated through evaluation of the hypervolumes that lie outside  $R_a$  but inside the tolerance region. In critical regions these polynomial approximations are updated during optimization. The one-dimensional convexity assumption for this method is much less restrictive than the multidimensional convexity required by the simplicial approach. Sensitivities for the estimated yield are also available.

Recently, Biernacki and Styblinski [30] have extended the work on multidimensional polynomial approximation by considering a dynamic constraint approximation scheme. It avoids the large number of base points required for a full quadratic interpolation by selecting a maximally flat interpolation. During optimization, whenever a new base point is added, the approximation is updated. It shows improved accuracy compared with a linear model as well as reduced computational effort compared with a full quadratic model.

### C. The Gravity Method

Soin and Spence [98] proposed a statistical exploration approach. Based on a Monte Carlo analysis, the centers of gravity of the failed and passed samples are determined as, respectively,

$$\begin{aligned} \phi^f &= \left[ \sum_{k \in J} \phi^k \right] / K_{\text{fail}} \\ \phi^p &= \left[ \sum_{k \notin J} \phi^k \right] / K_{\text{pass}} \end{aligned} \quad (34)$$

where  $J$  is the index set identifying the failed samples.  $K_{\text{fail}}$  and  $K_{\text{pass}}$  are the numbers of failed and passed samples, respectively. The nominal point  $\phi^0$  is then adjusted along the direction  $s = \phi^p - \phi^f$  using a line search.

This algorithm is simple but also heuristic. It is not clear as to how the gravity centers are related to the yield in a general multidimensional problem.

#### D. The Parametric Sampling Method

The parametric sampling approach by Singhal and Pinel [97] has provided another promising direction. A continuous estimate of yield (as opposed to the Monte Carlo estimate, using discrete samples) is given by the following integral:

$$Y(x) = \int_{-\infty}^{+\infty} I_a(\phi) \Gamma(\phi, x) d\phi \quad (35)$$

where  $I_a(\phi)$  is the acceptance index defined in (20) and  $\Gamma(\phi, x)$  the parameter distribution density function which depends on the design variables  $x$  (e.g., the nominal point specifies the mean value and the tolerances control the standard deviations). Normally, in order to estimate the yield, we generate samples  $\phi^k$ ,  $k=1,2,\dots,K$ , from the component density  $\Gamma$ , perform  $K$  circuit analyses, and then take the average of  $I_a(\phi^k)$ . For each new set of variables  $x$  we would have a new density function, and therefore, the sampling and circuit analyses have to be repeated.

The parametric sampling method is based on the concept of importance sampling as

$$Y(x) = \int_{-\infty}^{+\infty} I_a(\phi) \frac{\Gamma(\phi, x)}{h(\phi)} h(\phi) d\phi \quad (36)$$

where  $h(\phi)$  is called the sampling density function. The samples  $\phi^k$  are generated from  $h(\phi)$  instead of  $\Gamma(\phi, x)$ . An estimate of the yield is made as

$$\begin{aligned} Y(x) &= \frac{1}{K} \sum_{k=1}^K I_a(\phi^k) \frac{\Gamma(\phi^k, x)}{h(\phi^k)} \\ &= \frac{1}{K} \sum_{k=1}^K I_a(\phi^k) W(\phi^k, x). \end{aligned} \quad (37)$$

The weights  $W(\phi^k, x)$  compensate for the use of a sampling density different from the component density.

This approach has two clear advantages. First, once the indices  $I_a(\phi^k)$  are calculated, no more model simulations are required when  $x$  is changed. Furthermore, if  $\Gamma$  is a differentiable density function, then gradients of the estimated yield are readily available. Hence, powerful optimization techniques may be employed. In practice, the algorithm starts with a large number of base points sampled from  $h(\phi)$  to construct the initial databank. To maintain a sufficient accuracy, the databank needs to be updated by adding new samples during optimization.

This approach, however, cannot be applied to nondifferentiable density functions such as uniform, discrete, and truncated distributions. It can be extended to include some tunable parameters if the tuning ranges are fixed or practically unlimited. In this case the acceptance index  $I_a(\phi^k)$  is defined as 1 if  $\phi^k$  is acceptable after tuning. If  $\phi^k$  is unacceptable before tuning, then whether it can be tuned and, if so, by how much, may have to be determined

through optimization. Variable tuning ranges (in order to minimize cost) cannot be accommodated by the parametric sampling method.

#### E. Generalized $l_p$ Centering

Here, we propose a generalized  $l_p$  centering algorithm which encompasses, in a unified formulation, problems of 100 percent yield (worst-case design) and less than 100 percent yield.

First, we consider the centering problem where we have fixed tolerances and no tuning. Only the nominal point  $\phi^0$  is to be optimized. Define

$$f = [e^T(\phi^1) \cdots e^T(\phi^K)]^T \quad (38)$$

as the set of multicircuit error functions. We can achieve a worst-case minimax design by

$$\text{minimize}_x U(x) = H_\infty(f) = \max_k \max_j \{e_j(\phi^k)\} \quad (39)$$

where the multiple circuits  $\phi^k$  are related to  $\phi^0$  according to (19).

If a 100 percent yield is not attainable, we would naturally look for a solution where the specifications are met by as many points (out of  $K$  circuits) as possible. For this purpose minimax is not a proper choice, since unless and until the worst case is dealt with nothing else seems to matter. We may attempt to use a generalized  $l_2$  or  $l_1$  function (i.e.,  $H_2(f)$  or  $H_1(f)$ ) instead of  $H_\infty(f)$  in (39), hoping to reduce the emphasis given to the worst case.

In order to gain more insight into the problem, we define, for each  $\phi^k$ , a scalar function which will indicate directly whether  $\phi^k$  satisfies or violates the specifications and by how much. For this purpose, we choose a set of generalized  $l_p$  functions as

$$v_k(x) = H_p(e(\phi^k)), \quad k=1,2,\dots,K. \quad (40)$$

The sign of  $v_k$  indicates the acceptability of  $\phi^k$  while the magnitude of  $v_k$  measures, so to speak, the distance between  $\phi^k$  and the boundary of the acceptable region. For example, with  $p=\infty$  the distance is measured in the worst-case sense whereas for  $p=2$  it will be closer to a Euclidean norm.

We can define a generalized  $l_p$  centering as

$$\text{minimize}_x U(x) = H_p(u(x)) \quad (41)$$

where

$$u(x) = \begin{bmatrix} \alpha_1 v_1 \\ \vdots \\ \alpha_K v_K \end{bmatrix} \begin{bmatrix} \alpha_1 H_q(e(\phi^1)) \\ \vdots \\ \alpha_K H_q(e(\phi^K)) \end{bmatrix} \quad (42)$$

and  $\alpha_1, \alpha_2, \dots, \alpha_K$  are a set of positive multipliers. With different  $p$  and  $q$  it leads to a variety of algorithms for yield enhancement. We discuss separately the case where a nonpositive  $U(x)$  exists and the case where we always have  $U(x) > 0$ .

In the first case, the existence of a  $U(x) \leq 0$  indicates that a 100 percent yield is attainable. We should point out

that for a given  $x$  the sign of  $U(x)$  does not depend on  $p$ ,  $q$ , or  $\alpha_k$ . However, the optimal solution  $x$  at which  $U(x)$  attains its minimum is dependent on  $p$ ,  $q$ , and  $\alpha$ . This means that using any values of  $p$ ,  $q$ , and  $\alpha$  we will be able to achieve a  $U(x) \leq 0$  (i.e., to achieve a 100 percent yield). Furthermore, by using different  $p$ ,  $q$ , and  $\alpha$ , we influence the centering of  $\phi^0$ . Interestingly, the worst-case centering (39) becomes a special case by letting both  $p, q = \infty$  and using unit multipliers.

Now consider the case where the optimal yield is less than 100 percent. In this case we propose the use of  $p = 1$  and  $q = 1$  in (41). Also, given a starting point  $x_0$ , we define the set of multipliers by

$$\alpha_k = 1/|v_k(x_0)|, \quad k = 1, 2, \dots, K. \quad (43)$$

Our proposition is based on the following reasoning (a more complete theoretical justification is reserved for a future paper).

Consider the  $l_p$  sum given by

$$\sum_{k \in J} [u_k(x)]^p \quad (44)$$

where  $J = \{k | u_k > 0\}$ . As  $p \rightarrow 0$  (44) approaches the total number of unacceptable circuits which we wish to minimize. The smallest  $p$  that gives a convex approximation is 1. This leads to the generalized  $l_1$  objective function given by

$$U(x) = \sum_{k \in J} u_k(x) = \sum_{k \in J} \alpha_k v_k(x). \quad (45)$$

With the multipliers defined by (43), the value of the objective function at the starting point, namely  $U(x_0)$ , is precisely the count of unacceptable circuits. Also, notice that the magnitude of  $v_k$  measures the closeness of  $\phi^k$  to the acceptable region. A small  $|v_k|$  indicates that  $\phi^k$  is close to satisfying or violating the specifications. Therefore, we assign a large multiplier to it so that more emphasis will be given to  $\phi^k$  during optimization. On the other hand, we de-emphasize those points that are far away from the boundary of the acceptable region because their contributions to the yield are less likely to change.

One important feature of this approach is its capability of accommodating arbitrary tolerance distributions, since they only influence the generation of  $\phi^k$ . The numerical results we have obtained are very promising. The generalized  $l_p$  centering algorithm can also be extended to include variable tolerances and tuning.

## VI. EXAMPLES OF STATISTICAL DESIGN

### Example 1

The classical two-section 10:1 transmission line transformer, originally proposed by Bandler *et al.* [23] to test minimax optimizers, is a good example for illustrating graphically the basic ideas of centering and tolerancing. An upper specification on the reflection coefficient as  $|\rho| \leq 0.55$  and 11 frequencies  $\{0.5, 0.6, \dots, 1.5 \text{ GHz}\}$  are considered. The lengths of the transmission lines are fixed at the quarter-wavelength while the characteristic impedances  $Z_1$  and  $Z_2$  are to be toleranced and optimized. Fig.

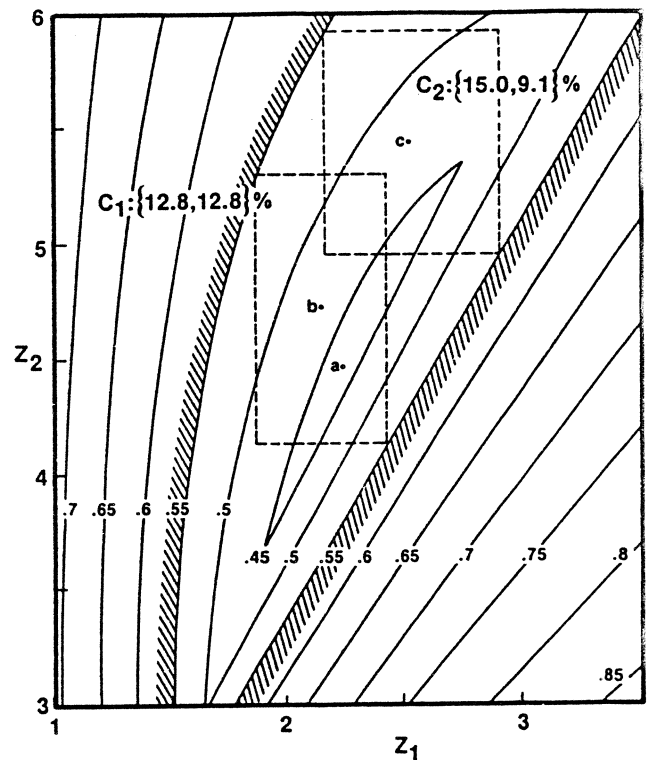


Fig. 11. Contours of  $\max |\rho|$  with respect to  $Z_1$  and  $Z_2$  for the two-section transformer indicating the minimax nominal solution  $a$ , the centered design with relative tolerances  $b$ , and the centered design with absolute tolerances  $c$ . The values in brackets are the optimized tolerances (as percentages of the nominal values). The specification is  $|\rho| \leq 0.55$ .

11 shows the minimax contours, the minimax nominal solution, and the worst-case solutions [23] for

$$P0: \text{minimize } C_1 = Z_1^0/\epsilon_1 + Z_2^0/\epsilon_2$$

subject to  $Y = 100$  percent

$$P1: \text{minimize } C_2 = 1/\epsilon_1 + 1/\epsilon_2 \text{ subject to } Y = 100 \text{ percent}$$

where  $\epsilon_1, \epsilon_2$  denote tolerances on  $Z_1$  and  $Z_2$  (assuming independent uniform distributions), and  $Y$  is the yield. The cost functions  $C_1$  and  $C_2$  correspond to, respectively, relative and absolute tolerancing problems. Two problems of less than 100 percent yield have also been considered by Bandler and Abdel-Malek [7] as

$$P2: \text{minimize } C_2 \text{ subject to } Y \geq 90 \text{ percent}$$

$$P3: \text{minimize } C_2/Y.$$

The optimal tolerance regions and nominal values for  $P2$  and  $P3$  are shown in Fig. 12. For more details see the original paper [7].

### Example 2

The statistical design of a Chebyshev low-pass filter (Singhal and Pinel [97]) is used as the second example. Fifty-one frequencies  $\{0.02, 0.04, \dots, 1.0, 1.3 \text{ Hz}\}$  are considered. An upper specification of 0.32 dB on the insertion loss is defined for frequencies from 0.02 to 1.0 Hz. A lower specification of 52 dB on the insertion loss is defined at 1.3 Hz.

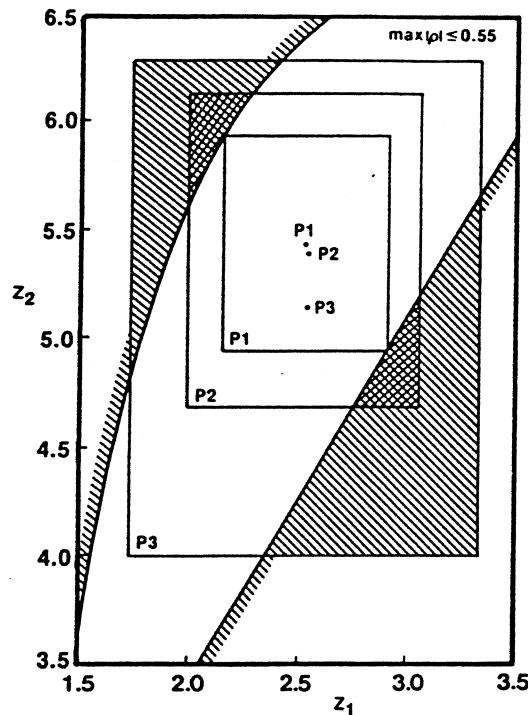


Fig. 12. The optimized tolerance regions and nominal values for the worst case design P1, 90 percent yield design P2, and minimum cost design P3 of the two-section transformer.

Singhal and Pinel [97] have applied the parametric sampling method to the same circuit, assuming normal distributions for the toleranced elements. But, as we have pointed out earlier in this paper, the parametric sampling method cannot be applied to nondifferentiable (such as uniform) distributions. Here, we consider a uniformly distributed 1.5 percent relative tolerance for each component. The generalized  $l_p$  centering algorithm described in Section V is used with  $p=1$ . The nominal solution by standard synthesis as given in [97] was used as starting point, which has a 49 percent yield (w.r.t. the tolerances specified). An 84 percent yield is achieved at the solution which involves a sequence of three design cycles with a total CPU time of 66 seconds on the VAX 8600. Some details are provided in Table I.

## VII. GRADIENT-BASED OPTIMIZATION METHODS

So far we have concentrated on translating our practical concerns into mathematical expressions. Now we turn our attention to the solution methods for optimization problems.

The studies in the last two decades on the theoretical and algorithmic aspects of optimization techniques have produced a great number of results. Modern state-of-the-art methods have largely replaced the primitive trial-and-error-approach. In particular, gradient-based optimization methods have gained increasing popularity in recent years for their effectiveness and efficiency.

The majority of gradient-based methods belong to the Gauss-Newton, quasi-Newton, and conjugate gradient families. All these are iterative algorithms which, from a

TABLE I  
STATISTICAL DESIGN OF A LOW-PASS FILTER USING  
GENERALIZED  $l_1$  CENTERING TECHNIQUE

Component $\phi_i$	Nominal Design $\phi_i^{0,0}$	Case 1 $\phi_i^{0,1}$	Case 2 $\phi_i^{0,2}$	Case 3 $\phi_i^{0,3}$
$x_1$	0.2251	0.21954	0.21705	0.21530
$x_2$	0.2494	0.25157	0.24677	0.23838
$x_3$	0.2523	0.25529	0.24784	0.24120
$x_4$	0.2494	0.24807	0.24019	0.23687
$x_5$	0.2251	0.22042	0.21753	0.21335
$x_6$	0.2149	0.22627	0.23565	0.23093
$x_7$	0.3636	0.36739	0.37212	0.38225
$x_8$	0.3761	0.36929	0.38012	0.39023
$x_9$	0.3761	0.37341	0.38371	0.39378
$x_{10}$	0.3636	0.36732	0.37716	0.38248
$x_{11}$	0.2149	0.22575	0.22127	0.23129
Yield	49%	77.67%	79.67%	83.67%
Number of samples used for design		50	100	100
Starting point		$\phi^{0,0}$	$\phi^{0,1}$	$\phi^{0,2}$
Number of iterations		16	18	13
CPU time (VAX 8600)		10 sec.	30 sec.	26 sec.

Independent uniform distributions are assumed for each component with fixed tolerances  $\epsilon_i = 1.5\% \phi_i^0$ . The yield is estimated based on 300 samples.

given starting point  $x_0$ , generate a sequence of points  $\{x_k\}$ . The success of an algorithm depends on whether  $\{x_k\}$  will converge to a point  $x^*$  and, if so, whether  $x^*$  will be a stationary point. An iterative algorithm is described largely by one of its iterations as how to obtain  $x_{k+1}$  from  $x_k$ .

We use the notation  $U(x)$  for the objective function and  $\nabla U$  for the gradient vector of  $U$ . When  $U(x)$  is defined by an  $l_p$  function, we use  $f$  to denote the set of individual error functions so that  $U = H(f)$ . We also use  $f'_j$  for the first-order derivatives of  $f_j$  and  $G$  for the Jacobian matrix of  $f$ .

### A. $l_p$ Optimization and Mathematical Programming

Of the  $l_p$  family,  $l_1$ ,  $l_2$ , and  $l_\infty$  are the most distinctive and by far the most useful members. Apart from their unique theoretical properties, it is very important from the algorithmic point of view that linear  $l_1$ ,  $l_2$ , and  $l_\infty$  problems can be solved exactly using linear or quadratic programming techniques. Besides, all the other members of the  $l_p$  family have a continuously differentiable function and, therefore, can be treated similarly to the  $l_2$  case.

An  $l_1$ ,  $l_2$ , or  $l_\infty$  optimization problem can be converted into a mathematical program. The concepts of local linearization and optimality conditions are often clarified by the equivalent formulation.

For instance, the minimization of  $\|f\|_1$  is equivalent to

$$\underset{x, y}{\text{minimize}} \sum_{j=1}^m y_j \quad (46)$$

TABLE II  
MATHEMATICAL PROGRAMMING EQUIVALENT FORMULATIONS FOR  
 $l_1$ ,  $l_2$ , AND  $l_\infty$  OPTIMIZATION

The original problem:		
minimize $H(f)$		
$x$		
The equivalent problem:		
minimize $V(x,y)$ subject to the constraints as defined below		
$x,y$		
$H(f)$	$V(x,y)$	constraints (for $j = 1, 2, \dots, m$ )
$l_1$	$\sum_{j=1}^m y_j$	$y_j \geq f_j, y_j \leq -f_j$
$l_2$	$y^T y$	$y_j = f_j$
$l_\infty$	$y$	$y \geq f, y \leq -f$
$H_p^+(f)$	$\sum_{j=1}^m y_j$	$y_j \geq f_j, y_j \geq 0$
$H_p^-(f)$	$y^T y$	$y_j \geq f_j, y_j \geq 0$
$H_p^*(f)$	$y$	$y \geq f, y \geq 0$
$H_p(f)$	$y$	$y \geq f_j$

Note: A generalized  $l_p$  function  $H_p(f)$  is defined through  $H_p^+(f)$  and  $H_p^-(f)$ .  $H_p^-$  is a continuously differentiable function for all  $p < \infty$ .

subject to

$$y_j \geq f_j(x), \quad y_j \leq -f_j(x), \quad j = 1, 2, \dots, m.$$

Other equivalent formulations are summarized in Table II. For the convenience of presentation, we denote these mathematical programming problems by  $P(x, f)$ . One important feature of  $P(x, f)$  is that it has a linear or quadratic objective function. If  $f$  is a set of linear functions, then  $P(x, f)$  becomes a linear or quadratic program which can be solved using standard techniques. Equally importantly, linear constraints can be easily incorporated into the problem. Let  $P(x, f, D)$  be the problem of  $P(x, f)$  subject to a set of linear constraints of the form

$$D: \begin{cases} a_l^T x + b_l = 0, & l = 1, 2, \dots, L_{eq} \\ a_l^T x + b_l \geq 0, & l = L_{eq} + 1, \dots, L \end{cases} \quad (47)$$

where  $a_l$  and  $b_l$  are constants. If  $P(x, f)$  is a linear or quadratic program, so is  $P(x, f, D)$ . In other words, unconstrained and linearly constrained linear  $l_1$ ,  $l_2$ , and  $l_\infty$  problems can be solved using standard linear or quadratic programming techniques.

### B. Gauss-Newton Methods Using Trust Regions

For a general problem, we may, at each iteration, substitute  $f$  with a linearized model  $\tilde{f}$  so that  $P(x, \tilde{f})$  can be solved.

For a Gauss-Newton type method, at a given point  $x_k$ , a linearization of  $f$  is made as

$$\tilde{f}(h) = f(x_k) + G(x_k)h \quad (48)$$

where  $G$  is the Jacobian matrix. We then solve the linear

or quadratic program  $P(h, \tilde{f}, D)$ , where

$$D: \begin{cases} \Lambda_k \geq h_j, & j = 1, 2, \dots, n \\ \Lambda_k \geq -h_j, & j = 1, 2, \dots, n. \end{cases} \quad (49)$$

These additional constraints define a trust region in which the linearized model  $\tilde{f}$  is believed to be a good approximation to  $f$ .

Another way to look at it is that we have applied a semilinearization (Madsen [78]) to  $U(x) = H(f)$  resulting in

$$\bar{U}(h) = H(\tilde{f}(h)). \quad (50)$$

It is important to point out that (50) is quite different from a normal linearization as  $U(h) = U(x_k) + [\nabla U(x_k)]^T h$  which corresponds to a steepest descent method. In fact  $\nabla U$  may not even exist.

Denote the solution of  $P(h, \tilde{f}, D)$  by  $h_k$ . If  $x_k + h_k$  reduces the original objective function, we take it as the next iterate; i.e., if  $U(x_k + h_k) < U(x_k)$  then  $x_{k+1} = x_k + h_k$ . Otherwise we let  $x_{k+1} = x_k$ . In the latter case, the trust region is apparently too large and, consequently, should be reduced. At each iteration, the local bound  $\Lambda_k$  in (49) is adjusted according to the goodness of the linearized model.

The above describes the essence of a class of algorithms due to Madsen, who has called it method 1. Madsen [78] has shown that the algorithm provides global convergence in which the proper use of trust regions constitutes a critical part. Such a method has been implemented as an important element in the minimax and  $l_1$  algorithms of Hald and Madsen [65], [66]. In some other earlier work by Osborne and Watson [85], [86] the problem  $P(h, f)$  was solved without incorporating a trust region and the solution  $h_k$  was used as the direction for a line search. For their methods no convergence can be guaranteed and  $\{x_k\}$  may even converge to a nonstationary point.

Normally for the least-squares objective we have to solve a quadratic program at each iteration, which can be a time-consuming process. A remarkable alternative is the Levenberg-Marquardt [76], [81] method. Given  $x_k$ , it solves

$$\text{minimize}_h h^T (G^T G + \theta_k \mathbf{1}) h + 2 f^T G h + f^T f \quad (51)$$

where  $G = G(x_k)$ ,  $f = f(x_k)$ , and  $\mathbf{1}$  is an identity matrix. The minimizer  $h_k$  is obtained simply by solving the linear system

$$(G^T G + \theta_k \mathbf{1}) h_k = -G^T f \quad (52)$$

using, for example, LU factorization. The Levenberg-Marquardt parameter  $\theta_k$  is very critical for this method. First of all, it is made to guarantee the positive definiteness of (52). Furthermore, it plays, roughly speaking, an inverted role of  $\Lambda_k$  to control the size of a trust region. When  $\theta_k \rightarrow \infty$ ,  $h_k$  gives an infinitesimal steepest descent step. When  $\theta_k = 0$ ,  $h_k$  becomes the solution to  $P(h, \tilde{f})$  without bounds, which is equivalent to having  $\Lambda_k \rightarrow \infty$ .

The concept of trust region has been discussed in a broader context by Moré in a recent survey [82].

### C. Quasi-Newton Methods

Quasi-Newton methods (also known as variable metric methods) are originated in and steadily upgraded from the work of Davidon [45] and Broyden [33], [34], as well as Fletcher and Powell [55].

For a differentiable  $U(x)$ , a quasi-Newton step is given by

$$h_k = -\alpha_k B_k^{-1} \nabla U(x_k) \quad (53)$$

where  $B_k$  is an approximation to the Hessian of  $U(x)$  and the step size controlling parameter  $\alpha_k$  is to be determined through a line search. However, on some occasions such as in the  $l_1$  or minimax case, the gradient  $\nabla U$  may not exist, much less the Hessian.

We can gain more insight to the general case by examining the optimality conditions. Applying the Kuhn–Tucker conditions for nonlinear programming [70] to the equivalent problem  $P(x, f)$ , we shall find a set of optimality equations

$$R(x) = 0. \quad (54)$$

Since a local optimum  $x^*$  must satisfy these equations, we are naturally motivated to solve (54), as a means of finding the minimizer of  $U(x)$ . A quasi-Newton step for solving nonlinear equations (54) is given by

$$h_k = -\alpha_k J_k^{-1} R(x_k) \quad (55)$$

where  $J_k$  is an approximate Jacobian of  $R(x)$ . Only when  $U(x)$  is differentiable will we have the optimality equations as  $R(x) = \nabla U(x) = 0$  and (55) reverts to (53).

Hald and Madsen [65], [66] and Bandler *et al.* [21], [22] have described the implementation of a quasi-Newton method for the minimax and  $l_1$  optimization in which the objective functions are not differentiable. Clarke [43] has introduced the concept of generalized gradient, with which optimality conditions can be derived for a broad range of problems.

Quasi-Newton methods, whether in (53) or (55), all require updates of certain approximate Hessians. Many formulas have been proposed over the years. The best known are the Powell symmetric Broyden (PSB) update [91], the Davidon–Fletcher–Powell (DFP) update [45], [55], and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) update [35], [53], [60], [95]. The merits of these formulas and a great many other variations are often compared in terms of their preservation of positive definiteness, convergence to the true Hessian, and numerical performance (see, for instance, Fletcher [54] and Gill and Murray [59]).

Another important point to be considered is the line search. Ideally,  $\alpha_k$  is chosen as the minimizer of  $U$  in the direction of line search so that  $h_k^T \nabla U(x_k + h_k) = 0$ . If exact line searches are executed, Dixon [50] has shown that theoretically all members of the Broyden family [34], [53] would have the same performance. In practice, however, exact line search is deemed too expensive and is therefore replaced by other methods. An inexact line search usually limits the evaluation of  $U$  and  $\nabla U$  to only a few points.

Interpolation and extrapolation techniques (such as a quadratic or cubic fit) are then incorporated.

### D. Combined Methods

The distinguishing advantage of a quasi-Newton method is that it enjoys a fast rate of convergence near a solution. However, like the Newton method for nonlinear equations, the quasi-Newton method is not always reliable from a bad starting point.

Hald and Madsen [65], [66], [78] have suggested a class of two-stage algorithms. A first-order method of the Gauss–Newton type is employed in stage 1 to provide global convergence to a neighborhood of a solution. When the solution is singular, method 1 suffers from a very slow rate of convergence and a switch is made to a quasi-Newton method (stage 2). Several switches between the two methods may take place and the switching criteria ensure the global convergence of the combined algorithm. Numerical examples of circuit applications have demonstrated a very strong performance of the approach [21], [22], [79], [80].

Powell [92] has extended the Levenberg–Marquardt method and suggested a trust-region strategy which interpolates between a steepest descent step and a Newton step. When far away from the solution, the step is biased toward the steepest descent direction to make sure that it is downhill. Once close to the solution, taking a full Newton step will provide rapid final convergence.

### E. Conjugate Gradient Methods

Some extremely large-scale engineering applications involve hundreds of variables and functions. Although the rapid advances in computer technology have enabled us to solve increasingly larger problems, there may be cases in which even the storage of a Hessian matrix and the solution of an  $n$  by  $n$  linear system become unmanageable.

Conjugate gradient methods [56], [75], [88] provide an alternative for such problems. A distinct advantage of conjugate gradient methods is the minimal requirement of storage. Typically three to six vectors of length  $n$  are needed, which is substantially less than the requirement by the Gauss–Newton or quasi-Newton methods. However, proper scaling or preconditioning, near-perfect line searches and appropriate restart criteria are usually necessary to ensure convergence. In general, we have to pay the price for the reduced storage by enduring a longer computation time.

## VIII. GRADIENT CALCULATION AND APPROXIMATION

The application of gradient-based  $l_p$  optimization methods requires the first-order derivatives of the error functions with respect to the variables.

In circuit optimization, these derivatives are usually obtained from a sensitivity analysis of the network under consideration. For linearized circuits in the frequency domain, it is often possible to calculate the exact sensitivities by the adjoint network approach [5], [31], [48].

However, we ought to recognize that an explicit and elegant sensitivity expression is not always available. For time-domain responses and nonlinear circuits, an exact formula may not exist. Even for linear circuits in the frequency domain, large-scale networks present new problems which need to be addressed.

Often, a large-scale network can be described through compounded and interconnected subnetworks. Many commercial CAD packages such as SUPER-COMPACT [99] and TOUCHSTONE [104], [105] have facilitated such a block structure. In this case, one possible approach would be to assemble the overall nodal matrix and solve the system of equations using sparse techniques (see, e.g., Duff [51], Gustavson [61], Hachtel *et al.* [62]). Another possibility is to rearrange the overall nodal matrix into a bordered block structure which is then solved using the Sherman-Morrison-Woodbury formula [63], [96]. Sometimes it is also possible to develop efficient formulas for a special structure, such as the approach of Bandler *et al.* [17] for branched cascaded networks.

In practice, perhaps the most perplexing and time-consuming part of the task is to devise an index scheme through which pieces of lower level information can be brought into the overall sensitivity expression. It may also require a large amount of memory storage for the various intermediate results. Partly due to these difficulties, methods of exact sensitivity calculations have yet to find their way into general-purpose CAD software packages, although the concept of adjoint network has been in existence for nearly two decades and has had success in many specialized applications.

In cases where either exact sensitivities do not exist or are too difficult to calculate, we can utilize gradient approximations [15], [16], [77], [109]. A recent approach to circuit optimization with integrated gradient approximations has been described by Bandler *et al.* [16]. It has been shown to be very effective and efficient in practical applications including FET modeling and multiplexer optimization.

## IX. CONCLUSIONS

In this review, we have formulated realistic circuit design and modeling problems and described their solution methods. Models, variables, and functions at different levels, as well as the associated tolerances and uncertainties, have been identified. The concepts of design centering, tolerancing, and tuning have been discussed. Recent advances in statistical design, yield enhancement, and robust modeling techniques suitable for microwave CAD have been discussed in detail. State-of-the-art optimization techniques have been addressed from both the theoretical and algorithmic points of view.

We have concentrated on aspects that are felt to be immediately relevant to and necessary for modern microwave CAD. There are, of course, other related subjects that have not been treated or not adequately treated in this paper. Notable among these are special techniques for very large systems (Geoffrion [57], [58], Haimes [64], Lasdon

[72]), third-generation simulation techniques (Hachtel and Sangiovanni-Vincentelli [63]), fault diagnosis (Bandler and Salama [27]), supercomputer-aided CAD (Rizzoli *et al.* [93]), the simulated annealing and combinatorial optimization methods and their application to integrated circuit layout problems [38], [69], [84], and the new automated decomposition approach to large scale optimization (Bandler and Zhang [28]).

The paper is particularly timely in that software based on techniques which we have described is being integrated by Optimization Systems Associates Inc. into SUPER-COMPACT by arrangement with Compact Software Inc.

## ACKNOWLEDGMENT

The authors would like to thank Dr. K. C. Gupta, Guest Editor of this Special Issue on Computer-Aided Design, for his invitation to write this review paper. The useful comments offered by the reviewers are also appreciated. The authors must acknowledge original work done by several researchers which has been integrated into our presentation, including that of Dr. H. L. Abdel-Malek, Dr. R. M. Biernacki, Dr. C. Charalambous, Dr. S. Daijavad, Dr. W. Kellermann, Dr. P. C. Liu, Dr. K. Madsen, Dr. M. R. M. Rizk, Dr. H. Tromp, and Dr. Q. J. Zhang. M. L. Renault is thanked for her contributions, including assistance in preparing data, programs, and results. The opportunity EEsoc Inc. provided to develop state-of-the-art optimizers into practical design tools, through interaction with Dr. W. H. Childs, Dr. C. H. Holmes, and Dr. D. Morton, is appreciated. Thanks are extended to Dr. R. A. Pucel of Raytheon Company, Research Division, Lexington, MA, for reviving the first author's interest and work in design centering and yield optimization. Dr. U. L. Rohde of Compact Software Inc., Paterson, NJ, is facilitating the practical implementation of advanced mathematical techniques of CAD. The stimulating environment provided by Dr. Pucel and Dr. Rohde to the first author is greatly appreciated.

## REFERENCES

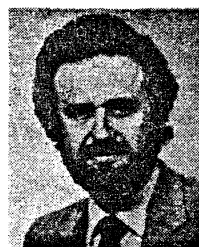
- [1] H. L. Abdel-Malek and J. W. Bandler, "Yield optimization for arbitrary statistical distributions, Part I: Theory," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 245-253, 1980.
- [2] H. L. Abdel-Malek and J. W. Bandler, "Yield optimization for arbitrary statistical distributions, Part II: Implementation," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 253-262, 1980.
- [3] D. Agnew, "Improved minimax optimization for circuit design," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 791-803, 1981.
- [4] J. W. Bandler, "Optimization methods for computer-aided design," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-17, pp. 533-552, 1969.
- [5] J. W. Bandler, "Computer-aided circuit optimization," in *Modern Filter Theory and Design*, G. C. Temes and S. K. Mitra, Eds. New York: Wiley, 1973, pp. 211-271.
- [6] J. W. Bandler, "Engineering modelling and design subject to model uncertainties and manufacturing tolerances," in *Methodology in Systems Modelling and Simulation*, B. P. Zeigler, *et al.*, Eds. Amsterdam: North-Holland, 1979, pp. 399-421.
- [7] J. W. Bandler and H. L. Abdel-Malek, "Optimal centering, tolerancing, and yield determination via updated approximations and cuts," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 853-871, 1978.
- [8] J. W. Bandler, R. M. Biernacki, and A. E. Salama, "A linear programming approach to fault location in analog circuits," in



- Proc. IEEE Int. Symp. Circuits Syst.*, (Chicago, IL), 1981, pp. 256-260.
- [9] J. W. Bandler, R. M. Biernacki, A. E. Salama, and J. A. Starzyk, "Fault isolation in linear analog circuits using the  $L_1$  norm," in *Proc. IEEE Int. Symp. Circuits Syst.*, (Rome, Italy), 1982, pp. 1140-1143.
  - [10] J. W. Bandler and C. Charalambous, "Theory of generalized least  $p$ th approximation," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 287-289, 1972.
  - [11] J. W. Bandler and C. Charalambous, "Nonlinear programming using minimax techniques," *J. Opt. Theory Appl.*, vol. 13, pp. 607-619, 1974.
  - [12] J. W. Bandler, S. H. Chen, and S. Daijavad, "Microwave device modeling using efficient  $l_1$  optimization: A novel approach," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-34, pp. 1282-1293, 1986.
  - [13] J. W. Bandler, S. H. Chen, S. Daijavad, and W. Kellermann, "Optimal design of multicavity filters and contiguous-band multiplexers," in *Proc. 14th European Microwave Conf.* (Liege, Belgium), 1984, pp. 863-868.
  - [14] J. W. Bandler, S. H. Chen, S. Daijavad, W. Kellermann, M. Renault, and Q. J. Zhang, "Large scale minimax optimization of microwave multiplexers," in *Proc. 16th European Microwave Conf.* (Dublin, Ireland), 1986, pp. 435-440.
  - [15] J. W. Bandler, S. H. Chen, S. Daijavad, and K. Madsen, "Efficient gradient approximations for nonlinear optimization of circuits and systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, (San Jose, CA), 1986, pp. 964-967.
  - [16] J. W. Bandler, S. H. Chen, S. Daijavad, and K. Madsen, "Efficient optimization with integrated gradient approximations," pp. 444-455, this issue.
  - [17] J. W. Bandler, S. Daijavad and Q. J. Zhang, "Computer aided design of branched cascaded networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, (Kyoto, Japan), 1985, pp. 1579-1582.
  - [18] J. W. Bandler, S. Daijavad, and Q. J. Zhang, "Exact simulation and sensitivity analysis of multiplexing networks," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-34, pp. 93-102, 1986.
  - [19] J. W. Bandler, M. A. El-Kady, W. Kellermann, and W. M. Zuberek, "An optimization approach to the best alignment of manufactured and operating systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, (Newport Beach, CA), 1983, pp. 542-545.
  - [20] J. W. Bandler and W. Kellermann, "Selected topics in optimal design centering, tolerancing and tuning," Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Rep. SOS-83-28, 1983.
  - [21] J. W. Bandler, W. Kellermann, and K. Madsen, "A superlinearly convergent minimax algorithm for microwave circuit design," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-33, pp. 1519-1530, 1985.
  - [22] J. W. Bandler, W. Kellermann, and K. Madsen, "A nonlinear  $l_1$  optimization algorithm for design, modelling and diagnosis of networks," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 174-181, 1987.
  - [23] J. W. Bandler, P. C. Liu, and J. H. K. Chen, "Worst case network tolerance optimization," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-23, pp. 630-641, 1975.
  - [24] J. W. Bandler, P. C. Liu, and H. Tromp, "A nonlinear programming approach to optimal design centering, tolerancing and tuning," *IEEE Trans. Circuits Syst.*, vol. CAS-23, pp. 155-165, 1976.
  - [25] J. W. Bandler, P. C. Liu, and H. Tromp, "Integrated approach to microwave design," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-24, pp. 584-591, 1976.
  - [26] J. W. Bandler and M. R. M. Rizk, "Optimization of electrical circuits," *Math. Program. Study*, vol. 11, pp. 1-64, 1979.
  - [27] J. W. Bandler and A. E. Salama, "Fault diagnosis of analog circuits," *Proc. IEEE*, vol. 73, pp. 1279-1325, 1985.
  - [28] J. W. Bandler and Q. J. Zhang, "An automatic decomposition technique for device modelling and large circuit design," in *IEEE Int. Microwave Symp. Dig.*, (Las Vegas, NV), 1987, pp. 709-712.
  - [29] R. H. Bartels and A. R. Conn, "An approach to nonlinear  $l_1$  data fitting," Computer Science Department, University of Waterloo, Waterloo, Canada, Rep. CS-81-17, 1981.
  - [30] R. M. Biernacki and M. A. Styblinski, "Statistical circuit design with a dynamic constraint approximation scheme," in *Proc. IEEE Int. Symp. Circuits Syst.*, (San Jose, CA), 1986, pp. 976-979.
  - [31] F. H. Branin, Jr., "Network sensitivity and noise analysis simplified," *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 285-288, 1973.
  - [32] R. K. Brayton, G. D. Hachtel, and A. L. Sangiovanni-Vincentelli, "A survey of optimization techniques for integrated-circuit design," *Proc. IEEE*, vol. 69, pp. 1334-1362, 1981.
  - [33] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Math. Comp.*, vol. 19, pp. 577-593, 1965.
  - [34] C. G. Broyden, "Quasi-Newton methods and their application to function minimization," *Math. Comp.*, vol. 21, pp. 368-381, 1967.
  - [35] C. G. Broyden, "A new double-rank minimization algorithm," *Notices Amer. Math. Soc.*, vol. 16, p. 670, 1969.
  - [36] E. M. Butler, "Realistic design using large-change sensitivities and performance contours," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 58-66, 1971.
  - [37] D. A. Calahan, *Computer-Aided Network Design*, rev. ed. New York: McGraw Hill, 1972.
  - [38] A. Casotto, F. Romeo, and A. L. Sangiovanni-Vincentelli, "A parallel simulated annealing algorithm for the placement of macro-cells," in *Proc. IEEE Int. Conf. Computer-Aided Design* (Santa Clara, CA), 1986, pp. 30-33.
  - [39] C. Charalambous, "A unified review of optimization," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-22, pp. 289-300, 1974.
  - [40] C. Charalambous, "Minimax design of recursive digital filters," *Computer Aided Design*, vol. 6, pp. 73-81, 1974.
  - [41] C. Charalambous, "Nonlinear least  $p$ th optimization and nonlinear programming," *Math. Program.*, vol. 12, pp. 195-225, 1977.
  - [42] C. Charalambous and A. R. Conn, "Optimization of microwave networks," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-23, pp. 834-838, 1975.
  - [43] F. H. Clarke, "Generalized gradients and applications," *Trans. Amer. Math. Soc.*, vol. 205, pp. 247-262, 1975.
  - [44] S. Daijavad, "Design and modelling of microwave circuits using optimization methods," Ph.D. thesis, McMaster University, Hamilton, Canada, 1986.
  - [45] W. C. Davidon, "Variable metric method for minimization," Rep. ANL-5990 Rev., Argonne National Laboratories, Argonne, IL, 1959.
  - [46] J. E. Dennis, Jr., and J. J. Moré, "Quasi-Newton methods, motivation and theory," *SIAM Rev.*, vol. 19, pp. 46-89, 1977.
  - [47] S. W. Director and G. D. Hachtel, "The simplicial approximation approach to design centering," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 363-372, 1977.
  - [48] S. W. Director and R. A. Rohrer, "Generalized adjoint network and network sensitivities," *IEEE Trans. Circuit Theory*, vol. CT-16, pp. 318-323, 1969.
  - [49] S. W. Director and R. A. Rohrer, "Automated network design: The frequency domain case," *IEEE Trans. Circuit Theory*, vol. CT-16, pp. 330-337, 1969.
  - [50] L. C. W. Dixon, "Quasi-Newton algorithms generate identical points," *Math. Program.*, vol. 2, pp. 383-387, 1972.
  - [51] I. S. Duff, "A survey of sparse matrix research," *Proc. IEEE*, vol. 65, pp. 500-535, 1977.
  - [52] N. J. Elias, "New statistical methods for assigning device tolerances," in *Proc. IEEE Int. Symp. Circuits Syst.*, (Newton, MA), 1975, pp. 329-332.
  - [53] R. Fletcher, "A new approach to variable metric algorithms," *Comput. J.*, vol. 13, pp. 317-322, 1970.
  - [54] R. Fletcher, "A survey of algorithms for unconstrained optimization," in *Numerical Methods for Unconstrained Optimization*, W. Murray, Ed. London: Academic Press, 1972.
  - [55] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Comput. J.*, vol. 6, pp. 163-168, 1963.
  - [56] R. Fletcher and C. M. Reeves, "Function minimisation by conjugate gradients," *Comput. J.*, vol. 7, pp. 149-154, 1964.
  - [57] A. M. Geoffrion, "Elements of large-scale mathematical programming—Part I: Concepts," *Management Sci.*, vol. 16, pp. 652-675, 1970.
  - [58] A. M. Geoffrion, "Elements of large-scale mathematical programming—Part II: Synthesis of algorithms and bibliography," *Management Sci.*, vol. 16, pp. 676-691, 1970.
  - [59] P. E. Gill and W. Murray, "Quasi-Newton methods for unconstrained minimization," *J. Inst. Math. Appl.*, vol. 9, pp. 91-108, 1972.
  - [60] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Math. Comp.*, vol. 24, pp. 23-26, 1970.
  - [61] F. G. Gustavson, "Some basic techniques for solving sparse systems of linear equations," in *Sparse Matrices and Their Applications*, D. J. Rose and R. A. Willoughby, Eds. New York: Plenum Press, 1971.
  - [62] G. D. Hachtel, R. K. Brayton, and F. G. Gustavson, "The sparse

- tableau approach to network analysis and design," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 101-113, 1971.
- [63] G. D. Hachtel and A. L. Sangiovanni-Vincentelli, "A survey of third-generation simulation techniques," *Proc. IEEE*, vol. 69, pp. 1264-1280, 1981.
- [64] Y. Y. Haimes, Ed., *Large Scale Systems*. Amsterdam: North Holland, 1982.
- [65] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for minimax optimization," *Math. Program.*, vol. 20, pp. 49-62, 1981.
- [66] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for nonlinear  $l_1$  optimization," *SIAM J. Numer. Anal.*, vol. 22, pp. 68-80, 1985.
- [67] R. Hettich, "A Newton-method for nonlinear Chebyshev approximation," in *Approximation Theory*, R. Schaback and K. Scherer, Eds. (Lecture Notes in Mathematics, 556). Berlin: Springer, 1976, pp. 222-236.
- [68] B. J. Karafin, "The optimum assignment of component tolerances for electrical networks," *Bell Syst. Tech. J.*, vol. 50, pp. 1225-1242, 1971.
- [69] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.
- [70] H.W. Kuhn and A. W. Tucker, "Non-linear programming," in *Proc. 2nd Symp. Math. Statistics Probability* (Berkeley, CA), 1951, pp. 481-493.
- [71] K. R. Laker, M. S. Ghausi, and J. J. Kelly, "Minimum sensitivity active (leapfrog) and passive ladder bandpass filters," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 670-677, 1975.
- [72] L. S. Lasdon, *Optimization Theory for Large Systems*. New York: Macmillan, 1970.
- [73] L. S. Lasdon, D. F. Suchman, and A. D. Waren, "Nonlinear programming applied to linear array design," *J. Acoust. Soc. Amer.*, vol. 40, pp. 1197-1200, 1966.
- [74] L. S. Lasdon and A. D. Waren, "Optimal design of filters with bounded, lossy elements," *IEEE Trans. Circuit Theory*, vol. CT-13, pp. 175-187, 1966.
- [75] D. Le, "A fast and robust unconstrained optimization method requiring minimum storage," *Math. Program.*, vol. 32, pp. 41-68, 1985.
- [76] K. Levenberg, "A method for the solution of certain problems in least squares," *Quart. Appl. Math.*, vol. 2, pp. 164-168, 1944.
- [77] K. Madsen, "Minimax solution of nonlinear equations without calculating derivatives," *Math. Program. Study*, vol. 3, pp. 110-126, 1975.
- [78] K. Madsen, "Minimization of non-linear approximation functions," Dr. techn. thesis, Institute of Numerical Analysis, Tech. Univ. of Denmark, DK2800 Lyngby, Denmark, 1985.
- [79] K. Madsen and H. Schjaer-Jacobsen, "New algorithms for worst case tolerance optimization," in *Proc. IEEE Int. Symp. Circuits Syst.*, (New York), 1978, pp. 681-685.
- [80] K. Madsen, H. Schjaer-Jacobsen, and J. Voldby, "Automated minimax design of networks," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 791-796, 1975.
- [81] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, vol. 11, pp. 431-441, 1963.
- [82] J. J. Moré, "Recent developments in algorithms and software for trust region methods," in *Mathematical Programming, The State of the Art*. Bonn: Springer Verlag, 1982, pp. 258-287.
- [83] D. D. Morrison, "Optimization by least squares," *SIAM J. Numer. Anal.*, vol. 5, pp. 83-88, 1968.
- [84] S. Nahar, S. Sahni, and E. Shragowitz, "Simulated annealing and combinatorial optimization," in *Proc. 23rd Design Automat. Conf.*, (Las Vegas, NV), 1986, pp. 293-299.
- [85] M. R. Osborne and G. A. Watson, "An algorithm for minimax optimization in the nonlinear case," *Comput. J.*, vol. 12, pp. 63-68, 1969.
- [86] M. R. Osborne and G. A. Watson, "On an algorithm for discrete nonlinear  $l_1$  approximation," *Comput. J.*, vol. 14, pp. 184-188, 1971.
- [87] J. F. Pintel and K. A. Roberts, "Tolerance assignment in linear networks using nonlinear programming," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 475-479, 1972.
- [88] E. Polak, *Computational Methods in Optimization: A Unified Approach*. New York: Academic Press, 1971, pp. 53-54.
- [89] E. Polak, "An implementable algorithm for the optimal design centering, tolerancing, and tuning problem," *J. Opt. Theory Appl.*, vol. 37, pp. 45-67, 1982.
- [90] E. Polak and A. L. Sangiovanni-Vincentelli, "Theoretical and computational aspects of the optimal design centering, tolerancing, and tuning problem," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 795-813, 1979.
- [91] M. J. D. Powell, "A new algorithm for unconstrained optimization," in *Nonlinear Programming*, J. B. Rosen, O. L. Mangasarian and K. Ritter, Eds. New York: Academic Press, 1970.
- [92] M. J. D. Powell, "A hybrid method for nonlinear equations," in *Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz, Ed. London: Gordon and Breach, 1970.
- [93] V. Rizzoli, M. Ferlito, and A. Neri, "Vectorized program architectures for supercomputer-aided circuit design," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-34, pp. 135-141, 1986.
- [94] J. Schoeffler, "The synthesis of minimum sensitivity networks," *IEEE Trans. Circuit Theory*, vol. CT-11, pp. 271-276, 1964.
- [95] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Math. Comp.*, vol. 24, pp. 647-656, 1970.
- [96] J. Sherman and W. J. Morrison, "Adjustment of an inverse matrix corresponding to changes in the elements of a given column or row of the original matrix," *Annu. Math. Statist.*, vol. 20, p. 621, 1949.
- [97] K. Singhal and J. F. Pintel, "Statistical design centering and tolerancing using parametric sampling," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 692-701, 1981.
- [98] R. S. Sojn and R. Spence, "Statistical exploration approach to design centering," *Proc. Inst. Elec. Eng.*, vol. 127, pt. G., pp. 260-269, 1980.
- [99] *SUPER-COMPACT User's Manual*, Compact Software Inc., Paterson, NJ 07504, May 1986.
- [100] K. S. Tahim and R. Spence, "A radial exploration approach to manufacturing yield estimation and design centering," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 768-774, 1979.
- [101] T. S. Tang and M. A. Styblinski, "Yield gradient estimation for non-differentiable density functions using convolution techniques and their application to yield optimization," in *Proc. IEEE Int. Symp. Circuits Syst.*, (San Jose, CA), 1986, pp. 1306-1309.
- [102] G. C. Temes and D. A. Calahan, "Computer-aided network optimization the state-of-the-art," *Proc. IEEE*, vol. 55, pp. 1832-1863, 1967.
- [103] G. C. Temes and D. Y. F. Zai, "Least  $p$ th approximation," *IEEE Trans. Circuit Theory*, vol. CT-16, pp. 235-237, 1969.
- [104] *TOUCHSTONE User's Manual*, EEsos Inc., Westlake Village, CA 91362, Aug. 1985.
- [105] *TOUCHSTONE Reference Manual*, (Version 1.5), EEsos Inc., Westlake Village, CA 91362, Mar. 1987.
- [106] H. Tromp, "The generalized tolerance problem and worst case search," in *Proc. Conf. Computer-Aided Design of Electronic and Microwave Circuits Syst.*, (Hull, England), 1977, pp. 72-77.
- [107] H. Tromp, "Generalized worst case design, with applications to microwave networks," Doctoral thesis (in Dutch), Faculty of Engineering, University of Ghent, Ghent, Belgium, 1978.
- [108] A. D. Waren, L. S. Lasdon, and D. F. Suchman, "Optimization in engineering design," *Proc. IEEE*, vol. 55, pp. 1885-1897, 1967.
- [109] W. M. Zuberek, "Numerical approximation of gradients for circuit optimization," in *Proc. 27th Midwest Symp. Circuits Syst.*, (Morgantown, WV), 1984, pp. 200-203.

✱



John W. Bandler (S'66-M'66-SM'74-F'78) was born in Jerusalem, Palestine, on November 9, 1941. He studied at Imperial College of Science and Technology, London, England, from 1960 to 1966. He received the B.Sc. (Eng.), Ph.D. and D.Sc. (Eng.) degrees from the University of London, England, in 1963, 1967, and 1976, respectively.

He joined Mullard Research Laboratories, Redhill, Surrey, England, in 1966. From 1967 to 1969 he was a Postdoctorate Fellow and Ses-

sional Lecturer at the University of Manitoba, Winnipeg, Canada. He joined McMaster University, Hamilton, Canada, in 1969, where he is currently a Professor of Electrical and Computer Engineering. He has served as Chairman of the Department of Electrical Engineering and Dean of the Faculty of Engineering. He currently directs research in the Simulation Optimization Systems Research Laboratory. Dr. Bandler is President of Optimization Systems Associates Inc., which he established in 1983. OSA currently provides consulting services and software, specializing in advanced applications of simulation, sensitivity analysis, and mathematical optimization techniques for CAE of microwave integrated circuits.

Dr. Bandler is a contributor to *Modern Filter Theory and Design* (Wiley-Interscience, 1973) and to the forthcoming *Analog Circuits: Computer-aided Analysis and Diagnosis* (Marcel Dekker). He has more than 220 publications, four of which appear in *Computer-Aided Filter Design* (IEEE Press, 1973), one in *Microwave Integrated Circuits* (Artech House, 1975), one in *Low-Noise Microwave Transistors and Amplifiers* (IEEE Press, 1981), one in *Microwave Integrated Circuits* (2nd ed., Artech House, 1985), one in *Statistical Design of Integrated Circuits* (IEEE Press, 1987), and one to be published in *Analog Fault Diagnosis* (IEEE Press). Dr. Bandler was an Associate Editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES (1969-1974). He was Guest Editor of the Special Issue of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES on Computer-Oriented Microwave Practices (March 1974). Dr. Bandler is a Fellow of the Royal Society of Canada and of the Institution of Electrical Engineers (Great Britain). He is a

member of the Association of Professional Engineers of the Province of Ontario (Canada).

✱



Shao Hua Chen (S'84) was born in Swatow, Guangdong, China, on September 27, 1957. He received the B.S. degree from the South China Institute of Technology, Guangzhou, China, in 1982 and the Ph.D. degree in electrical engineering from McMaster University, Hamilton, Canada, in 1987.

From July 1982 to August 1983, he was a teaching assistant in the Department of Automation at the South China Institute of Technology. He received a graduate scholarship from the Chinese Ministry of Education and worked in the Department of Electrical and Computer Engineering at McMaster University from 1983 to 1987. He held an Ontario Graduate Scholarship for the academic years 1985/86 and 1986/87. Currently he is working as a research engineer for Optimization Systems Associates Inc., Dundas, Ontario, Canada. His research interests include optimization methods, sensitivity analysis, device modeling, design centering, tolerancing and tuning, as well as interactive CAD software.



**SECTION TWENTY-THREE**

**KMOS - A FORTRAN LIBRARY FOR NONLINEAR OPTIMIZATION**

© J.W. Bandler, S.H. Chen and M.L. Renault 1987

This document originally appeared as Report SOS-87-1-R, February 1987. No part of this document may be copied, translated, transcribed or entered in any form into any machine without written permission. Address enquiries in this regard to Dr. J.W. Bandler. Excerpts may be quoted for scholarly purposes with full acknowledgement of source.



# KMOS - A FORTRAN LIBRARY FOR NONLINEAR OPTIMIZATION

J.W. Bandler, S.H. Chen and M.L. Renault

## Abstract

KMOS is a library of Fortran routines for solving nonlinear optimization problems. It includes seven optimization routines, namely MMLC for linearly constrained minimax problems using exact gradients, MMAG for linearly constrained minimax problems using approximate gradients, L1LC for linearly constrained  $\ell_1$  problems using exact gradients, L1AG for linearly constrained  $\ell_1$  problems using approximate gradients, S1LC for linearly constrained one-sided  $\ell_1$  problems using exact gradients, S1AG for linearly constrained one-sided  $\ell_1$  problems using approximate gradients and L2OS for unconstrained least-squares problems using exact gradients. The general theory behind these algorithms has been described by Madsen. The basic iteration uses either a first-order method to solve a linearized subproblem or a quasi-Newton method to solve the appropriate optimality equations. The KMOS library is developed to provide a unified interface to a user's program and a standardized printing service. It has also significantly reduced the size of the combined Fortran codes because the optimizers share many common subroutines.

---

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant A7239.

The authors are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

## I. INTRODUCTION

KMOS is a library of Fortran routines for solving nonlinear optimization problems. It includes seven optimization routines, namely MMLC for linearly constrained minimax problems using exact gradients [1],[2],[3], MMAG for linearly constrained minimax problems using approximate gradients [4],[5],[6], L1LC for linearly constrained  $\ell_1$  problems using exact gradients [7],[8],[9], L1AG for linearly constrained  $\ell_1$  problems using approximate gradients [4],[5],[6], S1LC for linearly constrained one-sided  $\ell_1$  problems using exact gradients, S1AG for linearly constrained one-sided  $\ell_1$  problems using approximate gradients and L2OS for unconstrained least-squares problems using exact gradients [10]. The general theory behind these algorithms has been described by Madsen [11]. The basic iteration uses either a first-order method to solve a linearized subproblem or a quasi-Newton method to solve the appropriate optimality equations. The KMOS library is developed to provide a unified interface to a user's program and a standardized printing service. It has also significantly reduced the size of the combined Fortran codes because the optimizers share many common subroutines.

The Fortran package MMLC for solving linearly constrained minimax problem was first developed by Bandler and Zuberek [1],[2] in 1982 for the CDC 170/730 system. Since then, many changes have taken place, both in the hardware and the software. For hardware, VAX systems have replaced the original CDC machine. The programming language has been upgraded to the current Fortran 77. Most importantly, several new optimization algorithms have been developed and implemented. These include the linearly constrained  $\ell_1$  package [7],[8],[9], the minimax and  $\ell_1$  packages using approximate gradients [4],[5],[6], the 2-stage least-squares package [10] and the one-sided  $\ell_1$  packages. It was felt that in order to facilitate applications in the future the services of these optimizers should be made available in a standardized format. The KMOS library was thus created.

The standard calling sequence to the optimization routines follows exactly that of the original MMLC package. A uniform printing format for reporting intermediate and final



results of optimization is provided. This will undoubtedly make it much easier to apply different optimization methods at the same time. The user only needs to make minimal changes to his/her program and therefore is much less likely to get confused. Also, the size of compact KMOS is much smaller than the combined size of the separate packages since they share many common subroutines.

KMOS is written in Fortran 77 for the VAX machine with VMS operating system. In order to utilize the library, the user should

- 1) write a Fortran program which prepares the relevant parameters and sets up a proper call to an optimization routine in KMOS (see the following sections);
- 2) compile this program using

$$\$FORTRAN \text{ user\_program}$$

- 3) link the object code with the KMOS library using

$$\$LINK \text{ user\_program} + \text{KMOS/LIB}$$

Notice that the name "user\_program" is only symbolic. Certainly the user may instead use other names or the name may represent several Fortran modules edited and compiled separately.

## II. GENERAL DESCRIPTION

Given a set of nonlinear functions

$$f_j(\mathbf{x}), \quad j = 1, \dots, m$$

of  $n$  variables

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T,$$

we try to find a local minimum of the objective function  $F(\mathbf{x})$  which is defined in the minimax, least-squares,  $\ell_1$  or one-sided  $\ell_1$  sense. Except for the case of least-squares, the present packages can also minimize  $F(\mathbf{x})$  subject to linear constraints

$$\mathbf{c}_k^T \mathbf{x} + b_k = 0, \quad k = 1, \dots, \ell_{\text{eq}},$$

$$\mathbf{c}_k^T \mathbf{x} + b_k \geq 0, \quad k = \ell_{\text{eq}} + 1, \dots, \ell,$$

where  $c_k$  and  $b_k$  are constants.

The minimax objective function is defined as

$$F(\mathbf{x}) = \max_j \{f_j(\mathbf{x})\}.$$

The least-squares objective function is defined as

$$F(\mathbf{x}) = \sum_{j=1}^m [f_j(\mathbf{x})]^2.$$

The  $\ell_1$  objective function is defined as

$$F(\mathbf{x}) = \sum_{j=1}^m |f_j(\mathbf{x})|.$$

The one-sided  $\ell_1$  objective function is defined as

$$F(\mathbf{x}) = \sum_{j \in J} f_j(\mathbf{x}).$$

where  $J = \{j | f_j(\mathbf{x}) \geq 0\}$ .

### III. LIST OF ARGUMENTS

It is utterly important for a user to declare double precision for all real values. The user is, therefore, advised to declare, in all his or her program segments,

IMPLICIT REAL\*8 (A-H,O-Z)

The subroutine call to the optimizers from a user's program is

```
CALL MMLC1A (FDF,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL MMAG1A (FUN,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL L1LC1A (FDF,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL L1AG1A (FUN,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL S1LC1A (FDF,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL S1AG1A (FUN,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL L2OS1A (FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
```

For MMAG, L1AG and S1AG, a common block must be defined as

COMMON /APPROX/ IP0,IP1,IP2,IWG,IWRK(5)

The arguments are explained as follows.

**FDF** is the name of a subroutine supplied by the user for MMLC, L1LC, S1LC and L2OS.

It must assume the form

```
SUBROUTINE FDF (N,M,X,DF,F)
```

```
REAL*8 X(N), DF(M,N), F(M)
```

When an optimization routine calls FDF, the variables are given in X(1), X(2), ..., X(N). FDF must calculate the values of the functions as well as their derivatives and store the results in

$$F(J) = f_J(\mathbf{x}), \quad J = 1, \dots, M$$

$$DF(J,I) = df_J/dx_I, \quad I = 1, \dots, N, \quad J = 1, \dots, M.$$

Notice that FDF is only a symbolic name. The actual name of this subroutine is arbitrary and it must be defined in the calling program as EXTERNAL.

**FUN** is the name of a subroutine supplied by the user for MMAG, L1AG and S1AG. It must assume the form

```
SUBROUTINE FUN (N,M,X,F)
```

```
REAL*8 X(N), F(M)
```

When an optimization routine calls FUN, the variables are given in X(1), X(2), ..., X(N). FUN must calculate the values of the functions and store the results in

$$F(J) = f_J(\mathbf{x}), \quad J = 1, \dots, M.$$

Notice that FUN is only a symbolic name. The actual name of this subroutine is arbitrary and it must be defined in the calling program as EXTERNAL.

**N** is an integer argument which must be set to the number of optimization parameters.

Its value must be positive and it is not changed by the package.

**M** is an integer argument which must be set to the number of residual functions defining the appropriate norm's objective function. Its value must be positive and it is not changed by the package.

**L** is an integer argument which must be set to the total number of linear constraints including equality and inequality constraints. Its value must be positive or zero and it is not changed by the package.

**LEQ** is an integer argument which must be set to the number of equality constraints. **LEQ** must not be greater than **N** (otherwise the system is already over-determined), and not greater than **L**.

**B** is a real array of dimension **B(LC)**, where argument **LC** is defined below. The elements **B(K)**,  $K = 1, \dots, L$ , must be set to the constant terms of the linear constraints (see definition of **C** below). The contents of **B** is not changed by the package.

**C** is a real matrix of dimension **C(LC,N)**. It must contain the coefficients of the constraints. The **K**th constraint is defined by

$$C(K,1)*X(1) + \dots + C(K,N)*X(N) + B(K) = 0, \text{ if } K \leq \text{LEQ},$$

$$C(K,1)*X(1) + \dots + C(K,N)*X(N) + B(K) \geq 0, \text{ otherwise.}$$

**LC** is an integer argument which must be set to the first dimension of arrays **B** and **C**. It must be not less than **L**. If **L=0**, **LC** must be at least 1. Its value is not changed by the package.

**X** is a real array of dimension **X(N)**. On entry, it must be set to the initial values of the variables (starting point) before calling **KMOS**. Upon return from **KMOS**, it contains the solution.

**DX** is a real variable which controls the step length of the iteration. On entry, a value between 0.05 - 0.2 is usually used. Upon return, **DX** contains the last value of the bound on the step length.

**EPS** is a real variable which specifies the required accuracy of the solution. A value between 1.D-4 to 1.D-6 is suggested. Upon return, **EPS** contains the length of the last step taken in the iteration.

**MAXF** is an integer argument which limits the maximum number of calls to **FDF** or **FUN**.

Upon return, **MAXF** is set to the actual number of such calls.

KEQS is an integer which controls the use of quasi-Newton iterations (Stage 2). Normally, KEQS=3 is used for MMLC, L1LC, S1LC or L2OS and KEQS=5 is used for MMAG, L1AG or S1AG. Setting KEQS=MAXF will in effect disable Stage 2. Upon return, KEQS contains the number of switches to Stage 2 that have taken place.

W is a real array providing working space for KMOS routines. Its dimension is given by IW. Upon return, the first M elements of W contain the residual function values at the solution, i.e.,

$$W(I) = f_i(\mathbf{x}), \quad I = 1, \dots, M.$$

IW is an integer indicating the size of working space. The minimum size of the working array is

$$\text{For MMLC: } 2*M*N + 5*N*N + 4*M + 8*N + 4*LC + 3$$

$$\text{For MMAG: } 3*M*N + 6*N*N + 5*M + 10*N + 4*LC + 3$$

$$\text{For L1LC and S1LC: } 2*M*N + 5*N*N + 5*M + 10*N + 4*LC$$

$$\text{For L1AG and S1AG: } 3*M*N + 6*N*N + 6*M + 12*N + 4*LC$$

$$\text{For L2OS: } 3*M*N + 2*N*N + 4*M + 9*N$$

It is probably advisable to use  $IW = 3*M*N + 6*N*N + 6*M + 12*N + 4*LC$  which satisfies the requirement of all packages.

ICH is an integer. It must be set to the unit number for printed output generated by KMOS. The user can make a "quiet call" to KMOS by setting  $ICH < 0$ , in which case no printed message will be generated. This in effect emulates the original entries MMLC1Q, etc. Its value is not changed by the package.

IPR is an integer that controls the printed output. Suppose that  $|IPR| = *****\#$ , where \* or # indicates a digit, then

\*\*\*\*\* specifies the frequency of reporting the values of functions and variables in the printed output

if # > 0 then partial derivatives will also be reported

if IPR < 0 then partial derivative verification will be performed at the starting point (ignored by MMAG, L1AG and S1AG)

Examples:

IPR = 100: to report values of the functions and variables for every 10 iterations.

IPR = -50: to verify partial derivatives and to report values of the functions and variables for every 5 iterations.

IPR = 151: to report values of the functions, variables and derivatives for every 15 iterations.

IFALL is an integer which, on return, contains information about the type of the solution.

IFALL = -2: feasible region is empty (conflicting constraints);

IFALL = -1: incorrect data (N < 0, EPS < 0, IW too small, etc.);

IFALL = 0: regular solution reached with required accuracy;

IFALL = 1: singular solution reached with required accuracy;

IFALL = 2: solution reached with machine accuracy;

IFALL = 3: number of calls to FDF or FUN reached MAXF;

IFALL = 4: iteration terminated by the user (see below).

MARK The user may terminate the optimization and force a return from KMOS by setting MARK = 0 in FDF or FUN, where integer MARK must have been declared as

```
COMMON /MML090/ MARK
```

Arguments relating to gradient approximation are discussed as follows. They are applicable to MMAG, L1AG and S1AG and must be declared as

```
COMMON /APPROX/ IP0,IP1,IP2,IWG,IWRK(5)
```

IP0 is an integer that indicates whether the initial approximate gradient should be computed by KMOS (by setting IP0 = 1) or to be supplied by the user (by setting IP0 = 0). If IP0 = 0, the user must supply the approximate derivatives at the starting point in the working array W, from W(2M + 1) to W(2M + N \* M), as follows.

$$W((I + 1) * M + J) = dF(J) / dX(I), \quad I = 1, \dots, N, \quad J = 1, \dots, M.$$

Its value is not changed by the package.

**IP1** is an integer that controls the frequency of perturbations in Stage 1 (see the following Table). Its value is not changed by the package.

**IP2** is an integer that controls the frequency of perturbations in Stage 2 (see Table I). Its value is not changed by the package.

TABLE I. COMBINED EFFECT OF IP1 AND IP2

Arguments	Perturbations Performed	
	Stage 1	Stage 2
IP1 > IP2 > 0	every (IP1)th iteration	every (IP2)th iteration
IP2 > IP1 > 0	every (IP1)th iteration	every (IP1)th iteration
IP1 > 0, IP2 < 0	every (IP1)th iteration	none
IP1 < 0, IP2 > 0	none	every (IP2)th iteration
IP1 < 0, IP2 < 0	none	none

Note: if IP2 > 0, perturbations are performed on entry to Stage 2

**IWG** is an integer that indicates whether the weighted Broyden update should be used. If **IWG** = 0 the original Broyden formula is used. If **IWG** = 1, the user must supply the weights in the working array **W** in the following order

$$W(K + (I - 1) * M + J) = \text{Weight}(I, J), \quad I = 1, \dots, N, \quad J = 1, \dots, M,$$

where

$$K = 2 * M * N + 6 * N * N + 5 * M + 10 * N + 4 * LC + 3 \quad \text{for MMAG,}$$

$$K = 2 * M * N + 6 * N * N + 6 * M + 12 * N + 4 * LC \quad \text{for L1AG and S1AG.}$$

Its value is not changed by the package.

**IWRK** is an integer array of dimension **IWRK**(5). It is used by **KMOS** as additional working space relating to gradient approximation.

#### IV. EXAMPLE

The HALD example [1] is used to illustrate the use of the optimization algorithms available in the KMOS library. The example is used in its original form for the optimization algorithms which are suited to designs, namely entries MMLC, MMAG, S1LC and S1AG. The example has been slightly converted in form for the optimization algorithms which are suited to parameter identification, namely entries L1LC, L1AG and L2OS.

##### The HALD example in its original form used for design purposes

Minimize

$$F(\mathbf{x}) = \|f_i(\mathbf{x})\| \quad \text{for } i = 1, 2, 3$$

subject to

$$-3x_1 - x_2 - 2.5 \geq 0,$$

where

$$f_1(\mathbf{x}) = x_1^2 + x_2^2 + x_1 x_2 - 1,$$

$$f_2(\mathbf{x}) = \sin(x_1),$$

$$f_3(\mathbf{x}) = -\cos(x_2).$$

##### The converted HALD example used for parameter identification purposes

Minimize

$$F(\mathbf{x}) = \|f_i(\mathbf{x}) - \text{spec}_i\| \quad \text{for } i = 1, 2, 3$$

subject to (except for L2OS algorithm)

$$-3x_1 - x_2 - 2.5 \geq 0,$$

where

$$f_1(\mathbf{x}) = x_1^2 + x_2^2 + x_1 x_2 - 1,$$

$$f_2(\mathbf{x}) = \sin(x_1),$$

$$f_3(\mathbf{x}) = -\cos(x_2),$$

and



$$\text{spec}_i = f_i(\mathbf{x}^*) \quad \text{for } i = 1, 2, 3$$

for  $\mathbf{x}^*$  denoting the minimax design solution, known apriori to be

$$\mathbf{x}^{*T} = [-.892857 \quad .178571]$$

### Source code and results

The user written Fortran source code required to solve the HALD problem is listed in the following pages. The user's program is composed of the main segment which prepares parameters and calls the desired optimization algorithm of KMOS (pp. 12-14), and the segment which calculates the values of residual functions and, if required, their first partial derivatives (pp. 15-16). These user written routines are then compiled and linked to the KMOS library.

Results obtained by the optimizers are also presented:

- the original HALD example solved by the minimax algorithm using exact gradients (pp. 17-20).
- the original HALD example solved by the minimax algorithm using gradient approximations (pp. 21-23).
- the original HALD example solved by the one-sided  $\ell_1$  algorithm using exact gradients (pp. 24-26).
- the original HALD example solved by the one-sided  $\ell_1$  algorithm using gradient approximations (pp. 27-29).
- the converted HALD example solved by the  $\ell_1$  algorithm using exact gradients (pp. 30-32).
- the converted HALD example solved by the  $\ell_1$  algorithm using gradient approximations (pp. 33-35).
- the converted HALD example solved by the  $\ell_2$  algorithm using exact gradients (pp. 36-38).

PROGRAM HALD

```
implicit real*8 (a-h,o-z)

parameter (n=2,m=3,lc=1,l=1,leq=0)
parameter (iw=3*m*n+6*n*n+6*m+12*n+4*lc)
dimension x(n),w(iw),b(lc),c(lc,n)
dimension solmm(n),dum(m)
external fdf,fun
common /approx/ ip0,ip1,ip2,iwg,iwrk(5)
common /blk1/ iopt,spec(3)
```

c display program information

```
    write(*,100)
100  format(1h1,
      +/' ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc',
      +/' c',
      +/' c      HALD example for design purposes: c',
      +/' c',
      +/' c Minimize      F(x) = || fi(x) || norm c',
      +/' c              for i=1,2,3 c',
      +/' c',
      +/' c subject to   - 3*x1 - x2 - 2.5 >= 0 c',
      +/' c',
      +/' c where       f1(x) = x1**2 + x2**2 + x1*x2 - 1 c',
      +/' c              f2(x) = sin(x1) c',
      +/' c              f3(x) = -cos(x2) c',
      +/' c',
      +/' c',
      +/' c',
      +/' c',
      +/' c      HALD example converted in form for parameter c',
      +/' c              identification purposes: c',
      +/' c',
      +/' c Minimize      F(x) = || fi(x) - speci || norm c',
      +/' c              for i=1,2,3 c',
      +/' c',
      +/' c subject to   - 3*x1 - x2 - 2.5 >= 0 c',
      +/' c              (except for the L20S optimizer) c',
      +/' c',
      +/' c where       f1(x) = x1**2 + x2**2 + x1*x2 - 1 c',
      +/' c              f2(x) = sin(x1) c',
      +/' c              f3(x) = -cos(x2) c',
      +/' c',
      +/' c and where   spec1 = f1(x) @ minimax design solution c',
      +/' c              spec2 = f2(x) @ minimax design solution c',
      +/' c              spec3 = f3(x) @ minimax design solution c',
      +/' c',
      +/' ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc')
```

c set solmm() to the known minimax solution and obtain the



```
if(iopt.eq.3) call s1lc1a(fdf,n,m,l,leq,b,c,lc,x,dx,eps,  
+                      maxf,keqs,w,iw,ich,ipr,ifall)  
if(iopt.eq.4) call s1ag1a(fun,n,m,l,leq,b,c,lc,x,dx,eps,  
+                      maxf,keqs,w,iw,ich,ipr,ifall)  
if(iopt.eq.5) call l1lc1a(fdf,n,m,l,leq,b,c,lc,x,dx,eps,  
+                      maxf,keqs,w,iw,ich,ipr,ifall)  
if(iopt.eq.6) call l1ag1a(fun,n,m,l,leq,b,c,lc,x,dx,eps,  
+                      maxf,keqs,w,iw,ich,ipr,ifall)  
if(iopt.eq.7) call l2os1a(fdf,n,m,x,dx,eps,  
+                      maxf,keqs,w,iw,ich,ipr,ifall)  
  
goto 20  
end
```

```
SUBROUTINE FDF(N,M,X,DF,F)

implicit real*8 (a-h,o-z)

dimension x(n),f(m),df(m,n)
common /blk1/ iopt,spec(3)

x1 = x(1)
x2 = x(2)

f(1) = x1**2 + x2**2 + x1*x2 - 1.D0
f(2) = sin(x1)
f(3) = -cos(x2)

c if parameter identification is desired construct the error
c functions of the converted HALD problem

    if(iopt.gt.4) then
        do 10 i=1,m
10      f(i)=f(i)-spec(i)
        endif

df(1,1) = 2.D0*x1 + x2
df(1,2) = 2.D0*x2 + x1
df(2,1) = cos(x1)
df(2,2) = 0.D0
df(3,1) = 0.D0
df(3,2) = sin(x2)

return
end
```

```
SUBROUTINE FUN(N,M,X,F)

implicit real*8 (a-h,o-z)

dimension x(n),f(m)
common /blk1/ iopt,spec(3)

x1 = x(1)
x2 = x(2)

f(1) = x1**2 + x2**2 + x1*x2 - 1.D0
f(2) = sin(x1)
f(3) = -cos(x2)

c if parameter identification is desired construct the error
c functions of the converted HALD problem

- if(iopt.gt.4) then
  do 10 i=1,m
10   f(i)=f(i)-spec(i)
  endif

return
end
```



Available optimizers are:

1. MMLC (for design purpose)
2. MMAG (for design purpose)
3. S1LC (for design purpose)
4. S1AG (for design purpose)
5. L1LC (for identification purpose)
6. L1AG (for identification purpose)
7. L2OS (for identification purpose)

\$Enter your choice :

Input : 1



PAGE : 1 4-FEB-1987 15:06:57 MMLC8D PACKAGE V:87.01  
 LINEARLY CONSTRAINED MINIMAX OPTIMIZATION

INPUT DATA  
 -----

NUMBER OF VARIABLES (N) . . . . . 2  
 NUMBER OF FUNCTIONS (M) . . . . . 3  
 TOTAL NUMBER OF LINEAR CONSTRAINTS (L) . . . . . 1  
 NUMBER OF EQUALITY CONSTRAINTS (LEQ) . . . . . 0  
 STEP LENGTH (DX) . . . . . 1.000E-01  
 ACCURACY (EPS) . . . . . 1.000E-06  
 MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . 500  
 NUMBER OF SUCCESSIVE ITERATIONS (KEQS) . . . . . 3  
 WORKING SPACE (IW) . . . . . .88  
 PRINTOUT CONTROL (IPR) . . . . . -500

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

FUNCTION EVALUATION : 1 / 0

MINIMAX OBJECTIVE: 6.000000000000E+00

VARIABLES		FUNCTION VALUES	
1	-2.000000000000E+00	1	6.000000000000E+00
2	-1.000000000000E+00	2	-9.092974268257E-01
		3	-5.403023058681E-01

SOLUTION  
 -----

MINIMAX OBJECTIVE: -3.303571428571E-01

VARIABLES		FUNCTION VALUES	
1	-8.928571428571E-01	1	-3.303571428571E-01
2	1.785714285714E-01	2	-7.788668934368E-01
		3	-9.840984453126E-01

PAGE : 2 4-FEB-1987 15:06:57 MMLC8D PACKAGE V:87.01  
LINEARLY CONSTRAINED MINIMAX OPTIMIZATION

TYPE OF SOLUTION (IFALL)	2
NUMBER OF FUNCTION EVALUATIONS	10
NUMBER OF SHIFTS TO STAGE-2	2
EXECUTION TIME (IN SECONDS)	.0.040

Available optimizers are:

1. MMLC (for design purpose)
2. MMAG (for design purpose)
3. S1LC (for design purpose)
4. S1AG (for design purpose)
5. L1LC (for identification purpose)
6. L1AG (for identification purpose)
7. L2OS (for identification purpose)

\$Enter your choice :

Input : 2



PAGE : 2 4-FEB-1987 15:07:08 MMAG8D PACKAGE V:87.01  
LINEARLY CONSTRAINED MINIMAX OPTIMIZATION WITH GRADIENT APPROXIMATION

TYPE OF SOLUTION (IFALL) . . . . .	1
NUMBER OF ORDINARY ITERATIONS . . . . .	.41
NUMBER OF SPECIAL ITERATIONS . . . . .	7
NUMBER OF PERTURBATIONS . . . . .	.10
TOTAL NUMBER OF FUNCTION EVALUATIONS . . . . .	.68
NUMBER OF SHIFTS TO STAGE-2 . . . . .	4
EXECUTION TIME (IN SECONDS) . . . . .	0.120

Available optimizers are:

1. MMLC (for design purpose)
2. MMAG (for design purpose)
3. S1LC (for design purpose)
4. S1AG (for design purpose)
5. L1LC (for identification purpose)
6. L1AG (for identification purpose)
7. L2OS (for identification purpose)

\$Enter your choice :

Input : 3



PAGE : 2 4-FEB-1987 15:07:22 S1LC8D PACKAGE V:87.01  
LINEARLY CONSTRAINED ONE-SIDED L1 OPTIMIZATION

TYPE OF SOLUTION (IFALL) . . . . .	.2
NUMBER OF FUNCTION EVALUATIONS . . . . .	9
NUMBER OF SHIFTS TO STAGE-2 . . . . .	.0
EXECUTION TIME (IN SECONDS) . . . . .	.0.050



Available optimizers are:

1. MMLC (for design purpose)
2. MMAG (for design purpose)
3. S1LC (for design purpose)
4. S1AG (for design purpose)
5. L1LC (for identification purpose)
6. L1AG (for identification purpose)
7. L2OS (for identification purpose)

\$Enter your choice :

Input : 4

PAGE : 1 4-FEB-1987 15:07:33 S1AG8D PACKAGE V:87.01  
 LINEARLY CONSTRAINED ONE-SIDED L1 OPTIMIZATION WITH GRADIENT APPROXIMATION

INPUT DATA

-----

NUMBER OF VARIABLES (N) . . . . . 2  
 NUMBER OF FUNCTIONS (M) . . . . . 3  
 TOTAL NUMBER OF LINEAR CONSTRAINTS (L) . . . . . 1  
 NUMBER OF EQUALITY CONSTRAINTS (LEQ) . . . . . 0  
 STEP LENGTH (DX) . . . . . 1.000E-01  
 ACCURACY (EPS) . . . . . 1.000E-06  
 MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . 500  
 INITIAL GRADIENT APPROXIMATION FLAG (IPO) . . . . . 1  
 FREQUENCY OF PERTURBATIONS IN STAGE1 (IP1) . . . . . 5  
 FREQUENCY OF PERTURBATIONS IN STAGE2 (IP2) . . . . . 5  
 WEIGHTED OR NON-WEIGHTED FORMULA (IWEIGH) . . . . . 0  
 NUMBER OF SUCCESSIVE ITERATIONS (KEQS) . . . . . 3  
 WORKING SPACE (IW) . . . . . 88  
 PRINTOUT CONTROL (IPR) . . . . . -500

FUNCTION EVALUATION : 1 / 0  
 ONE-SIDED L1 OBJECTIVE: 6.000000000000E+00

VARIABLES		FUNCTION VALUES	
1	-2.000000000000E+00	1	6.000000000000E+00
2	-1.000000000000E+00	2	-9.092974268257E-01
		3	-5.403023058681E-01

SOLUTION

-----

ONE-SIDED L1 OBJECTIVE: 1.743050148661E-14

VARIABLES		FUNCTION VALUES	
1	-9.621201098209E-01	1	1.743050148661E-14
2	-7.188087385925E-02	2	-8.204056516677E-01
		3	-9.974176821468E-01

PAGE : 2    4-FEB-1987    15:07:33    S1AG8D PACKAGE V:87.01  
LINEARLY CONSTRAINED ONE-SIDED L1 OPTIMIZATION WITH GRADIENT APPROXIMATION

TYPE OF SOLUTION (IFALL) . . . . .	2
NUMBER OF ORDINARY ITERATIONS . . . . .	10
NUMBER OF SPECIAL ITERATIONS . . . . .	2
NUMBER OF PERTURBATIONS . . . . .	2
TOTAL NUMBER OF FUNCTION EVALUATIONS . . . . .	14
NUMBER OF SHIFTS TO STAGE-2 . . . . .	0
EXECUTION TIME (IN SECONDS) . . . . .	0.060

Available optimizers are:

1. MMLC (for design purpose)
2. MMAG (for design purpose)
3. S1LC (for design purpose)
4. S1AG (for design purpose)
5. L1LC (for identification purpose)
6. L1AG (for identification purpose)
7. L2OS (for identification purpose)

\$Enter your choice :

Input : 5

PAGE : 1 4-FEB-1987 15:07:47 L1LC8D PACKAGE V:87.01  
 LINEARLY CONSTRAINED L1 OPTIMIZATION

INPUT DATA

-----

NUMBER OF VARIABLES (N) . . . . . 2  
 NUMBER OF FUNCTIONS (M) . . . . . 3  
 TOTAL NUMBER OF LINEAR CONSTRAINTS (L) . . . . . 1  
 NUMBER OF EQUALITY CONSTRAINTS (LEQ) . . . . . 0  
 STEP LENGTH (DX) . . . . . 1.000E-01  
 ACCURACY (EPS) . . . . . 1.000E-06  
 MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . 500  
 NUMBER OF SUCCESSIVE ITERATIONS (KEQS) . . . . . 3  
 WORKING SPACE (IW) . . . . . 88  
 PRINTOUT CONTROL (IPR) . . . . . -500

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

FUNCTION EVALUATION : 1 / 0

L1 OBJECTIVE: 6.904583901216E+00

VARIABLES		FUNCTION VALUES	
1	-2.000000000000E+00	1	6.330357196429E+00
2	-1.000000000000E+00	2	-1.304305602684E-01
		3	4.437961445195E-01

SOLUTION

-----

L1 OBJECTIVE: 7.133382227964E-08

VARIABLES		FUNCTION VALUES	
1	-8.928571000000E-01	1	5.357143999241E-08
2	1.785713000000E-01	2	0.000000000000E+00
		3	-1.776238228723E-08

PAGE : 2    4-FEB-1987    15:07:47    L1LC8D PACKAGE V:87.01  
LINEARLY CONSTRAINED L1 OPTIMIZATION

TYPE OF SOLUTION (IFALL) . . . . .	0
NUMBER OF FUNCTION EVALUATIONS . . . . .	10
NUMBER OF SHIFTS TO STAGE-2 . . . . .	0
EXECUTION TIME (IN SECONDS) . . . . .	0.040

Available optimizers are:

1. MMLC (for design purpose)
2. MMAG (for design purpose)
3. S1LC (for design purpose)
4. S1AG (for design purpose)
5. L1LC (for identification purpose)
6. L1AG (for identification purpose)
7. L2OS (for identification purpose)

\$Enter your choice :

Input : 6

PAGE : 1 4-FEB-1987 15:07:58 L1AG8D PACKAGE V:87.01  
 LINEARLY CONSTRAINED L1 OPTIMIZATION WITH GRADIENT APPROXIMATION

INPUT DATA  
 -----

NUMBER OF VARIABLES (N) . . . . . 2  
 NUMBER OF FUNCTIONS (M) . . . . . 3  
 TOTAL NUMBER OF LINEAR CONSTRAINTS (L) . . . . . 1  
 NUMBER OF EQUALITY CONSTRAINTS (LEQ) . . . . . 0  
 STEP LENGTH (DX) . . . . . 1.000E-01  
 ACCURACY (EPS) . . . . . 1.000E-06  
 MAX NUMBER OF FUNCTION EVALUATIONS (MAXF) . . . . . 500  
 INITIAL GRADIENT APPROXIMATION FLAG (IPO) . . . . . 1  
 FREQUENCY OF PERTURBATIONS IN STAGE1 (IP1) . . . . . 5  
 FREQUENCY OF PERTURBATIONS IN STAGE2 (IP2) . . . . . 5  
 WEIGHTED OR NON-WEIGHTED FORMULA (IWEIGH) . . . . . 0  
 NUMBER OF SUCCESSIVE ITERATIONS (KEQS) . . . . . 3  
 WORKING SPACE (IW) . . . . . 88  
 PRINTOUT CONTROL (IPR) . . . . . -500

FUNCTION EVALUATION : 1 / 0

L1 OBJECTIVE: 6.904583901216E+00

VARIABLES		FUNCTION VALUES	
1	-2.000000000000E+00	1	6.330357196429E+00
2	-1.000000000000E+00	2	-1.304305602684E-01
		3	4.437961445195E-01

SOLUTION  
 -----

L1 OBJECTIVE: 7.133382229352E-08

VARIABLES		FUNCTION VALUES	
1	-8.928571000000E-01	1	5.357143999241E-08
2	1.785713000000E-01	2	-1.110223024625E-16



PAGE : 2      4-FEB-1987    15:07:58      3    -1.776238219009E-08  
L1AG8D PACKAGE V:87.01  
LINEARLY CONSTRAINED L1 OPTIMIZATION WITH GRADIENT APPROXIMATION

TYPE OF SOLUTION (IFALL) . . . . .	0
NUMBER OF ORDINARY ITERATIONS . . . . .	12
NUMBER OF SPECIAL ITERATIONS . . . . .	3
NUMBER OF PERTURBATIONS . . . . .	3
TOTAL NUMBER OF FUNCTION EVALUATIONS . . . . .	21
NUMBER OF SHIFTS TO STAGE-2 . . . . .	0
EXECUTION TIME (IN SECONDS) . . . . .	0.060

Available optimizers are:

1. MMLC (for design purpose)
2. MMAG (for design purpose)
3. S1LC (for design purpose)
4. S1AG (for design purpose)
5. L1LC (for identification purpose)
6. L1AG (for identification purpose)
7. L2OS (for identification purpose)

\$Enter your choice :

Input : 7

PAGE : 1 4-FEB-1987 15:34:38 L2OS8D PACKAGE V:87.01  
UNCONSTRAINED LEAST-SQUARES OPTIMIZATION

INPUT DATA  
-----

NUMBER OF VARIABLES (N)	2
NUMBER OF FUNCTIONS (M)	3
STEP LENGTH (DX)	1.000E-01
ACCURACY (EPS)	1.000E-06
MAX NUMBER OF FUNCTION EVALUATIONS (MAXF)	500
NUMBER OF SUCCESSIVE ITERATIONS (KEQS)	3
WORKING SPACE (IW)	88
PRINTOUT CONTROL (IPR)	-500

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

FUNCTION EVALUATION : 1 / 0  
LEAST-SQUARES OBJECTIVE: 4.028738938332E+01

VARIABLES		FUNCTION VALUES	
1	-2.000000000000E+00	1	6.330357196429E+00
2	-1.000000000000E+00	2	-1.304305602684E-01
		3	4.437961445195E-01

SOLUTION  
-----

LEAST-SQUARES OBJECTIVE: 7.000233645414E-14

VARIABLES		FUNCTION VALUES	
1	-8.928568040142E-01	1	6.164087730520E-08
2	1.785703969812E-01	2	1.856391597388E-07
		3	-1.781595943828E-07

TYPE OF SOLUTION (IFALL)	0
NUMBER OF FUNCTION EVALUATIONS	11
NUMBER OF SHIFTS TO STAGE-2	0
EXECUTION TIME (IN SECONDS)	0.030

Available optimizers are:

1. MMLC (for design purpose)
2. MMAG (for design purpose)
3. S1LC (for design purpose)
4. S1AG (for design purpose)
5. L1LC (for identification purpose)
6. L1AG (for identification purpose)
7. L2OS (for identification purpose)

\$Enter your choice :

Input : 0

FORTRAN STOP

## REFERENCES

- [1] J. Hald (Adapted and Edited by J.W. Bandler and W.M. Zuberek), "MMLA1Q – A Fortran package for linearly constrained minimax optimization", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-81-14-UL, 1981.
- [2] J.W. Bandler and W.M. Zuberek, "MMLC – A Fortran package for linearly constrained minimax optimization", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-82-5-U2, 1983.
- [3] J.W. Bandler, W. Kellermann and K. Madsen, "A superlinearly convergent minimax algorithm for microwave circuit design", IEEE Trans. Microwave Theory Tech., vol. MTT-33, pp. 1519-1530, 1985.
- [4] J.W. Bandler, S.H. Chen, S. Daijavad and K. Madsen, "Efficient gradient approximations for nonlinear optimization of circuits and systems", Proc. IEEE Int. Symp. Circuits and Systems (San Jose, CA), pp. 964-967, 1986.
- [5] J.W. Bandler, S.H. Chen, S. Daijavad and K. Madsen, "Efficient optimization with integrated gradient approximations, Part I: algorithms", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-86-12-R, 1986.
- [6] J.W. Bandler, S.H. Chen, S. Daijavad and K. Madsen, "Efficient optimization with integrated gradient approximations, Part II: implementation", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-86-13-R, 1986.
- [7] J. Hald, "A 2-stage algorithm for nonlinear  $\ell_1$  optimization", Report No. NI-81-03, Institute for Numerical Analysis, Technical University of Denmark, Lyngby, Denmark, 1981.
- [8] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for nonlinear  $\ell_1$  optimization", SIAM J. on Numerical Analysis, vol. 22, pp.68-80, 1985.
- [9] J.W. Bandler, W. Kellermann and K. Madsen, "A nonlinear  $\ell_1$  optimization algorithm for design, modelling and diagnosis of networks", IEEE Trans. Circuits and Systems, vol. CAS-34, pp. 174-181, 1987.
- [10] K. Madsen, Lecture notes and private communications, 1986/87.
- [11] K. Madsen, "Minimization of Non-linear Approximation Functions", Thesis for den tekniske doktorgrad, Institute for Numerical Analysis, Technical University of Denmark, Lyngby, Denmark, 1986.



**SECTION TWENTY-FOUR**  
**PAST EXAMS, TESTS AND SOLUTIONS**

© J.W. Bandler 1988

This material may not be used without written permission for any purpose other than scholarship and private study in connection with courses taught by J.W. Bandler.





## EE3K4 SIMULATION AND OPTIMIZATION I

DURATION OF TEST: 2 hours

Wednesday, March 25, 1987

### THIS IS A CLOSED BOOK TEST

Candidates must attempt Questions

1 or 2 or 3, 4 or (5 and 6), 7 or 8 9 or 10 11

Write your name here. NAME: \_\_\_\_\_

Write your student number here. NO: \_\_\_\_\_

- Note: (1) All scripts and question papers must be turned in.  
(2) Estimated times required to complete the questions are indicated.  
(3) Please encircle questions attempted in the following table.

Questions Attempted (please encircle)	Weighting	Estimated Time (min.)	Examiner's Use only
1 or 2 or 3	15%	ten	
4 or (5 and 6)	20%	thirty	
7 or 8	15%	twenty	
9 or 10	30%	thirty	
11	20%	thirty	
<b>TOTAL</b>	<b>100%</b>	<b>2 hours</b>	

**Question 1** Given a differentiable function  $f$  of many variables  $\mathbf{x}$  and a corresponding direction vector  $\mathbf{s}$ ,

$$\lim_{\lambda \rightarrow 0^+} \frac{f(\mathbf{x} + \lambda \mathbf{s}) - f(\mathbf{x})}{\lambda} = \dots \dots \dots \text{(please state)?}$$

Explain in a few words the meaning of the above expression.

**Answer** (10 min.)

$$(1) \quad \lim_{\lambda \rightarrow 0^+} \frac{f(\underline{x} + \lambda \underline{s}) - f(\underline{x})}{\lambda} = \nabla f^T \underline{s}$$

Suppose

$$\nabla f^T \underline{s} > 0,$$

then there exists  $\sigma > 0$  such that for all  $\lambda, 0 \leq \lambda < \sigma$

$$f(\underline{x} + \lambda \underline{s}) > f(\underline{x})$$

or, (2)

$$\lim_{\lambda \rightarrow 0^+} \frac{f(\underline{x} + \lambda \underline{s}) - f(\underline{x})}{\lambda} = \nabla f^T \underline{s}$$

The above expression equals the gradient (derivative) of  $f$  at point  $\underline{x}$  in the direction of  $\underline{s}$ .

EE3K4

**Question 2** Use the method of Lagrange multipliers to minimize w.r.t.  $\phi_1$  and  $\phi_2$  the function

$$U = \phi_1^2 + \phi_2^2$$

subject to

$$\phi_1 + \phi_2 = 1$$

Sketch a diagram to illustrate the problem and its solution w.r.t.  $\phi_1$  and  $\phi_2$ . Verify your answer by substituting the constraint into the function.

**Answer (10 min.)**

Lagrange function

$$L(\phi_1, \phi_2, \lambda) = \phi_1^2 + \phi_2^2 + \lambda(\phi_1 + \phi_2 - 1)$$

$$\frac{\partial L}{\partial \phi_1} = 2\phi_1 + \lambda = 0$$

$$\frac{\partial L}{\partial \phi_2} = 2\phi_2 + \lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = \phi_1 + \phi_2 - 1 = 0$$

} solve this set of linear equations

we have  $\phi_1 = \phi_2 = \frac{1}{2}$ ,  $\lambda = -1$ . So

$$\min_{\phi_1, \phi_2} U(\phi_1, \phi_2) = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

Verify the solution:

From the constraint

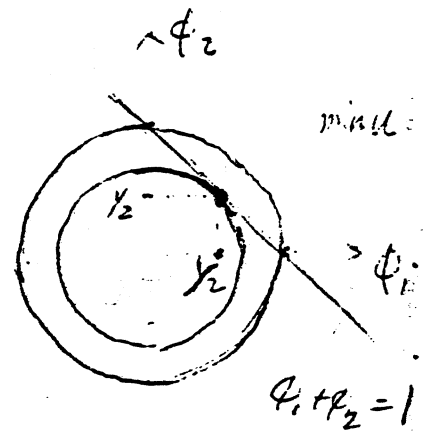
$$\phi_1 = 1 - \phi_2$$

$$U = (1 - \phi_2)^2 + \phi_2^2$$

$$= 2\phi_2^2 - 2\phi_2 + 1$$

$$\frac{dU}{d\phi_2} = 4\phi_2 - 2 = 0 \Rightarrow \phi_2 = \frac{1}{2} \Rightarrow \phi_1 = \frac{1}{2}$$

$$\text{So } \min_{\phi_1, \phi_2} U(\phi_1, \phi_2) = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$



**Question 3** Derive the gradient vector of  $U(\phi)$  w.r.t.  $\phi$  for the objective functions

$$U = \int_{\psi_l}^{\psi_u} |e(\phi, \psi)|^p d\psi$$

and

$$U = \sum_{i=1}^n |e_i(\phi)|^p,$$

where the appropriate error functions are complex.

**Answer** (10 min.)

$$\begin{aligned} \textcircled{1} \quad \frac{\partial |e|}{\partial \phi} &= \frac{\partial (e e^*)^{\frac{1}{2}}}{\partial \phi} = \frac{1}{2} (e e^*)^{-\frac{1}{2}} \left( \frac{\partial e}{\partial \phi} e^* + e \frac{\partial e^*}{\partial \phi} \right) \\ &= \frac{1}{2|e|} \cdot 2 \operatorname{Re} \left( \frac{\partial e}{\partial \phi} e^* \right) = |e|^{-1} \operatorname{Re} \left( \frac{\partial e}{\partial \phi} e^* \right) \\ &= |e| \cdot \operatorname{Re} \left( \frac{1}{e} \frac{\partial e}{\partial \phi} \right) \end{aligned}$$

$$\begin{aligned} \textcircled{2} \quad \nabla U &= \nabla \int_{\psi_l}^{\psi_u} |e(\phi, \psi)|^p d\psi = p \int_{\psi_l}^{\psi_u} \left[ |e(\phi, \psi)|^{p-2} \operatorname{Re} \left( \frac{1}{e} \nabla e \right) \right] d\psi \\ &= p \int_{\psi_l}^{\psi_u} \left[ |e(\phi, \psi)|^{p-2} \operatorname{Re} (e^* \nabla e) \right] d\psi \end{aligned}$$

$$\begin{aligned} \textcircled{3} \quad \nabla U &= \nabla \sum_{i=1}^n |e_i(\phi)|^p = p \sum_{i=1}^n \left[ |e_i(\phi)|^{p-2} \operatorname{Re} \left( \frac{1}{e_i} \nabla e_i \right) \right] \\ &= p \sum_{i=1}^n \left[ |e_i(\phi)|^{p-2} \operatorname{Re} (e_i^* \nabla e_i) \right] \end{aligned}$$

**Question 4** Derive, starting with Tellegen's theorem, the first-order sensitivity expression

$$-\mathbf{V}^T \Delta \mathbf{Y}^T \hat{\mathbf{V}}$$

for linear time-invariant networks in the frequency domain, where  $\mathbf{Y}$  is the s.c. admittance matrix of an element,  $\mathbf{V}$  the voltage vector in the original network and  $\hat{\mathbf{V}}$  the corresponding vector in the adjoint network of the element under consideration.

**Answer (30 min.)**

From Tellegen's Theorem

$$\sum \mathbf{V}_B \hat{\mathbf{I}}_B = 0 \quad \text{and} \quad \sum \mathbf{I}_B^T \hat{\mathbf{V}}_B = 0$$

When  $\mathbf{V}_B$  and  $\mathbf{I}_B$  change to  $\mathbf{V}_B + \Delta \mathbf{V}_B$  and  $\mathbf{I}_B + \Delta \mathbf{I}_B$  respectively, we have

$$\Delta \sum \mathbf{V}_B \hat{\mathbf{I}}_B - \Delta \sum \mathbf{I}_B^T \hat{\mathbf{V}}_B = 0 \quad (*)$$

Here we are only considering one element whose characteristic is given as

$$\mathbf{Y} \mathbf{V} = \mathbf{I}$$

Hence 
$$\Delta \mathbf{I} = \Delta \mathbf{Y} \mathbf{V} + \mathbf{Y} \Delta \mathbf{V}$$

put it into (\*)

$$\begin{aligned} \dots + \Delta \sum \mathbf{V}_B \hat{\mathbf{I}}_B - (\Delta \sum \mathbf{V}_B^T \mathbf{Y}^T + \sum \mathbf{V}_B^T \Delta \mathbf{Y}^T) \hat{\mathbf{V}}_B + \dots &= 0 \\ \dots + \Delta \sum \mathbf{V}_B \hat{\mathbf{I}}_B - \Delta \sum \mathbf{V}_B^T \mathbf{Y}^T \hat{\mathbf{V}}_B - \sum \mathbf{V}_B^T \Delta \mathbf{Y}^T \hat{\mathbf{V}}_B + \dots &= 0 \end{aligned}$$

If Let

$$\hat{\mathbf{I}} = \mathbf{Y}^T \hat{\mathbf{V}}$$

that is we use  $\mathbf{Y}^T$  to define the characteristic of the corresponding adjoint element, we obtain

$$\dots + (-\sum \mathbf{V}_B^T \Delta \mathbf{Y}^T \hat{\mathbf{V}}_B) + \dots = 0$$

Question 5 Consider the formula

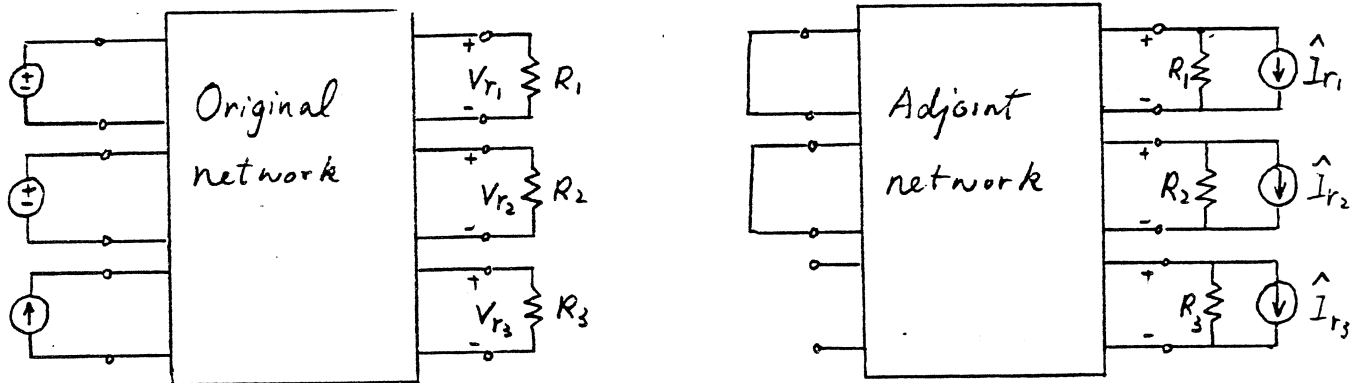
$$G = \sum_{\substack{\text{voltage} \\ \text{sources}}} \hat{V}_i \nabla I_i - \sum_{\substack{\text{current} \\ \text{sources}}} \hat{I}_i \nabla V_i$$

where  $G$  is a vector of standard sensitivity expressions,  $i$  is the index of the sources and  $\nabla$  is the partial derivative operator w.r.t. circuit parameters corresponding to  $G$ . Consider a six port network having two constant voltage sources, one constant current source, the remaining ports being terminated by resistors. Use the formula to show how to relate to  $G$  the gradient vector of

$$\sum_{\substack{\text{terminating} \\ \text{resistors}}} |V_r|^2 / R_r$$

where  $V_r$  is the response voltage and  $R_r$  is the terminating resistor. Draw the adjoint network and state the proper excitation.

Answer (15 min.)



$$U = \sum_{\substack{\text{(terminating)} \\ \text{resistors}}} |V_r|^2 / R_r$$

$$\begin{aligned} \nabla U &= \sum_{\text{resistors}} \frac{1}{R_r} 2 \operatorname{Re} [V_r^* \nabla V_r] - \sum_{\text{resistors}} \frac{|V_r|^2}{R_r^2} \nabla R_r \\ &= 2 \operatorname{Re} \left\{ \sum_{\text{resistors}} \left[ \frac{V_r^*}{R_r} \nabla V_r \right] \right\} - \sum_{\text{resistors}} \frac{|V_r|^2}{R_r^2} \nabla R_r \end{aligned}$$

$$\therefore G = - \sum_r \hat{I}_r \nabla V_r$$

if  $\hat{I}_r = V_r^* / R_r$ , then

$$\nabla U = -2 \operatorname{Re} \{ G \} - \sum_r \frac{|V_r|^2}{R_r^2} \nabla R_r$$

**Question 7** Derive from first principles, using manipulation of vectors and matrices, an approach to finding the first-order sensitivity of  $y_i$  w.r.t.  $a_{jk}$ , where  $\mathbf{A} \mathbf{y} = \mathbf{b}$  is a linear system in  $\mathbf{y}$ ,  $\mathbf{A}$  is a square matrix, the term  $y_i$  is the  $i$ th component of the column vector  $\mathbf{y}$  and  $a_{jk}$  represents the  $\{j, k\}$  element of  $\mathbf{A}$ . Discuss in detail the computational effort involved.

**Answer** (20 min.)

$$\therefore \underline{\mathbf{y}} = \underline{\mathbf{A}}^{-1} \underline{\mathbf{b}}, \quad y_i = \underline{u}_i^T \underline{\mathbf{y}}, \quad \text{and } \underline{\mathbf{b}} \text{ is constant,}$$

$$\begin{aligned} \therefore \frac{\partial y_i}{\partial a_{jk}} &= \underline{u}_i^T \frac{\partial \underline{\mathbf{y}}}{\partial a_{jk}} = \underline{u}_i^T \frac{\partial \underline{\mathbf{A}}^{-1}}{\partial a_{jk}} \underline{\mathbf{b}} = -\underline{u}_i^T \underline{\mathbf{A}}^{-1} \frac{\partial \underline{\mathbf{A}}}{\partial a_{jk}} \underline{\mathbf{A}}^{-1} \underline{\mathbf{b}} \\ &= -\underline{\hat{\mathbf{y}}}^T \frac{\partial \underline{\mathbf{A}}}{\partial a_{jk}} \underline{\mathbf{y}} \end{aligned}$$

$$\text{where } \underline{\hat{\mathbf{y}}}^T = \underline{u}_i^T \underline{\mathbf{A}}^{-1}, \quad \text{or } \underline{\mathbf{A}}^T \underline{\hat{\mathbf{y}}} = \underline{u}_i.$$

$$\therefore \frac{\partial \underline{\mathbf{A}}}{\partial a_{jk}} = \underline{u}_j \underline{u}_k^T$$

$$\therefore \frac{\partial y_i}{\partial a_{jk}} = -\underline{\hat{\mathbf{y}}}^T \underline{u}_j \underline{u}_k^T \underline{\mathbf{y}} = -\hat{y}_j \cdot y_k$$

$$\text{where } \underline{\hat{\mathbf{y}}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_j \\ \vdots \\ \hat{y}_n \end{bmatrix}, \quad \underline{\mathbf{y}} = \begin{bmatrix} y_1 \\ \vdots \\ y_j \\ \vdots \\ y_n \end{bmatrix}.$$

The computational effort involved:

Solve for  $\underline{\mathbf{y}}$  and  $\underline{\hat{\mathbf{y}}}$  from

$$\begin{aligned} \underline{\mathbf{A}} \underline{\mathbf{y}} &= \underline{\mathbf{b}} \\ \underline{\mathbf{A}}^T \underline{\hat{\mathbf{y}}} &= \underline{u}_i. \end{aligned}$$

So it needs 1 LU factorization  $\left(\frac{n^3}{3} - \frac{n}{3}\right)$ , and 2 forward and backward substitutions  $(n^2 + n^2)$ .

$\therefore$  Total effort

$$\frac{n^3}{3} + 2n^2 - \frac{n}{3} + 1$$

**Question 8** Consider the parameter constraints

$$0 \leq \phi_1 \leq \phi_2 \leq \dots \leq \phi_i \leq \dots \leq \phi_k$$

as the only constraints applicable in a minimization problem.

- (a) Find a suitable transformation of the variables  $\phi_1, \phi_2, \dots, \phi_k$  so that we can use an unconstrained optimization package.
- (b) Assuming the new variables are  $z_1, z_2, \dots$ , write down  $\partial U / \partial z_1, \partial U / \partial z_2, \dots$  given  $\partial U / \partial \phi_1, \partial U / \partial \phi_2, \dots$
- (c) You have access to a subprogram to calculate  $U$  and  $\nabla U$  given  $\phi$  but you can not alter it. How would you organize your software and data to handle the transformed problem?

**Answer (20 min.)**

$$(a) \quad \phi_i \geq 0 \Rightarrow \phi_i \triangleq z_1^2 \quad (1)$$

$$\begin{aligned} \phi_{j+1} \geq \phi_j &\Rightarrow \phi_{j+1} - \phi_j \geq 0 \Rightarrow \phi_{j+1} - \phi_j \triangleq z_{j+1}^2 \Rightarrow \\ &\Rightarrow \phi_{j+1} = z_{j+1}^2 + \phi_j \Rightarrow \phi_{j+1} = \sum_{i=1}^{j+1} z_i^2 \quad (2) \end{aligned}$$

where  $z_i$  is unconstrained.

$$\begin{aligned} \frac{\partial U}{\partial z_1} &= \frac{\partial U}{\partial \phi_1} \frac{\partial \phi_1}{\partial z_1} + \frac{\partial U}{\partial \phi_2} \frac{\partial \phi_2}{\partial z_1} + \dots + \frac{\partial U}{\partial \phi_k} \frac{\partial \phi_k}{\partial z_1} \\ &= \frac{\partial U}{\partial \phi_1} 2z_1 + \frac{\partial U}{\partial \phi_2} 2z_1 + \dots + \frac{\partial U}{\partial \phi_k} 2z_1 = 2z_1 \left( \frac{\partial U}{\partial \phi_1} + \frac{\partial U}{\partial \phi_k} \right) \end{aligned}$$

$$\frac{\partial U}{\partial z_i} = 2z_i \sum_{j=i}^k \frac{\partial U}{\partial \phi_j} \quad j=1, 2, \dots, k \quad (3)$$

- (c) 1) For  $z$  given by the optimization package, calculate  $\phi$  using (1) and (2);
- 2) call the subprogram to calculate  $U$  and  $\nabla_{\phi} U$  using  $\phi$  given in 1);
- 3) calculate  $\nabla_z U$  using  $\nabla_{\phi} U$  given in 2) and formulas in (3).



Question 10

- (a) State and explain the iterative formulas defining the conjugate gradient method of minimizing a differentiable function  $U(\phi)$  in terms of direction vectors  $s^j$  and  $s^{j-1}$ , and gradient vectors  $\nabla U^j$  and  $\nabla U^{j-1}$ . [Hint: take

$$s^j = -\nabla U^j + \beta^j s^{j-1} \quad (1)$$

where

$$\beta^j = \frac{(\nabla U^j)^T \nabla U^j}{(\nabla U^{j-1})^T \nabla U^{j-1}}. \quad (2)$$

- (b) State the formula for a quadratic function  $U(\phi)$  in terms of Hessian matrix  $A$ , constant vector  $b$ , and constant  $c$  associated with variable vector  $\phi$ .
- (c) State and explain the formula describing the property of conjugate directions  $u_i$  and  $u_j$  w.r.t. a positive definite matrix  $A$ .
- (d) Let  $j = 0$  for the first iteration of the conjugate gradient method. Let the first direction of search  $s^0 = -\nabla U^0$ . By using the property

$$\nabla U^j - \nabla U^{j-1} = \alpha^{j-1} A s^{j-1} \quad (3)$$

for the quadratic function of (b) where

$$\alpha^{j-1} s^{j-1} \triangleq \phi^j - \phi^{j-1} \quad (4)$$

prove that  $s^1$  and  $s^0$  are conjugate w.r.t.  $A$ . [Hint: Verify Equation (3), explain Equation (4) and assume that a full linear search for a minimum is conducted in each search direction  $s^j$ .]

- (e) Discuss and illustrate the implications of conjugate directions in the minimization of an unconstrained differentiable function of many variables. Discuss the properties of the conjugate gradient algorithm, its advantages and disadvantages.

Answer (30 min.)

(a) Given a starting point  $\underline{\phi}^0$ , a conjugate gradient iteration is defined by

$$\underline{\phi}^j = \underline{\phi}^{j-1} + \alpha^{j-1} \underline{s}^{j-1},$$

$$\underline{s}^j = -\nabla U^j + \beta^j \underline{s}^{j-1},$$

where

$$\beta^j = \frac{(\nabla U^j)^T \nabla U^j}{(\nabla U^{j-1})^T \nabla U^{j-1}}, \quad \underline{s}^0 = -\nabla U^0, \quad \nabla U^j \triangleq \nabla U(\underline{\phi}^j).$$

The value of  $\alpha^{j-1}$  is determined through a line search along  $\underline{s}^{j-1}$ .

(b)

$$U(\underline{\phi}) = \frac{1}{2} \underline{\phi}^T \underline{A} \underline{\phi} + \underline{b}^T \underline{\phi} + c$$

(c) The directions  $\underline{u}_i$  and  $\underline{u}_j$  are said to be conjugate w.r.t. a positive definite matrix  $\underline{A}$  if

$$\underline{u}_i^T \underline{A} \underline{u}_j = 0$$

(d) From the definition of a conjugate gradient iteration given in (a)

$$\underline{\phi}^j = \underline{\phi}^{j-1} + \alpha^{j-1} \underline{z}^{j-1}, \text{ we have } \alpha^{j-1} \underline{z}^{j-1} = \underline{\phi}^j - \underline{\phi}^{j-1} \quad (\text{d.1})$$

For a quadratic function given by (b), we have

$$\underline{\nabla} U = \underline{A} \underline{\phi} + \underline{b} \quad \therefore \underline{\nabla} U^j - \underline{\nabla} U^{j-1} = \underline{A} (\underline{\phi}^j - \underline{\phi}^{j-1}) \quad (\text{d.2})$$

$$= \alpha^{j-1} \underline{A} \underline{z}^{j-1} \quad (\text{d.3})$$

For the first iteration,  $\underline{z}^0 = -\underline{\nabla} U^0$ . (d.4)

A full line search along  $\underline{z}^0$  implies that  $(\underline{z}^0)^T \underline{\nabla} U^1 = 0 \Rightarrow (\underline{\nabla} U^0)^T \underline{\nabla} U^1 = 0$  (d.5)

we wish to show  $(\underline{z}^1)^T \underline{A} \underline{z}^0 = 0$ . Notice  $\underline{z}^1 = -\underline{\nabla} U^1 + \beta^1 \underline{z}^0$  (d.6)

Let  $j=1$  in (d.3), we have  $\underline{A} \underline{z}^0 = \frac{1}{\alpha^0} (\underline{\nabla} U^1 - \underline{\nabla} U^0)$ . (d.7)

Use (d.6) and (d.7)

$$(\underline{z}^1)^T \underline{A} \underline{z}^0 = (-\underline{\nabla} U^1 + \beta^1 \underline{z}^0)^T \frac{1}{\alpha^0} (\underline{\nabla} U^1 - \underline{\nabla} U^0)$$

$$= \frac{1}{\alpha^0} (-\underline{\nabla} U^1 + \beta^1 \underline{\nabla} U^0)^T (\underline{\nabla} U^1 - \underline{\nabla} U^0) \quad \because \underline{z}^0 = -\underline{\nabla} U^0$$

$$= \frac{1}{\alpha^0} [-(\underline{\nabla} U^1)^T \underline{\nabla} U^1 + (\underline{\nabla} U^1)^T \underline{\nabla} U^0 - \beta^1 (\underline{\nabla} U^0)^T \underline{\nabla} U^1 + \beta^1 (\underline{\nabla} U^0)^T \underline{\nabla} U^0]$$

$$= \frac{1}{\alpha^0} \left[ -(\underline{\nabla} U^1)^T \underline{\nabla} U^1 + \frac{(\underline{\nabla} U^1)^T \underline{\nabla} U^1}{(\underline{\nabla} U^0)^T \underline{\nabla} U^0} (\underline{\nabla} U^0)^T \underline{\nabla} U^0 \right] \quad (\because \beta^1 = \frac{(\underline{\nabla} U^1)^T \underline{\nabla} U^1}{(\underline{\nabla} U^0)^T \underline{\nabla} U^0})$$

$$= 0.$$

$\therefore \underline{z}^1$  and  $\underline{z}^0$  are conjugate w.r.t.  $\underline{A}$ .

[Solution to Question 10 continued]

(e) For a quadratic function described in (b) which has  $n$  variables (i.e.,  $\underline{\phi} = [\phi_1 \ \phi_2 \ \dots \ \phi_n]^T$ ), the conjugate gradient method will find the minimum after no more than  $n$  steps, i.e., there is  $\underline{\phi}^j = \underline{\phi}^*$ ,  $j \leq n$ ,  $\underline{\phi}^*$  is the solution.

For a general nonlinear function, such a property can be discussed only for a local quadratic approximation, and the method usually takes more than  $n$  steps to converge.

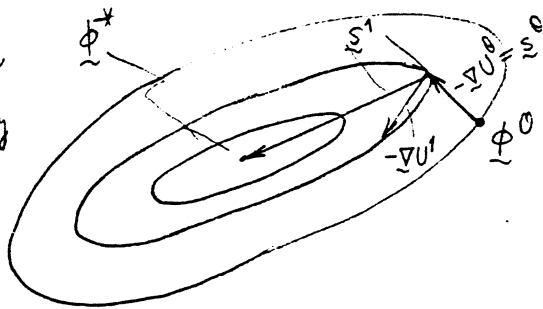


illustration for a two-dimensional quadratic function

The advantages of the C-G method

- are
- ①. its form is fairly simple,
  - ②. no linear equations need to be solved,
  - ③. no matrix needs to be stored.

Its disadvantages are the relatively slow convergence rate for a general nonlinear function as compared with more sophisticated methods (e.g., quasi-Newton method).

Question 11 Consider a system of complex linear equations

$$\underline{Y} \underline{V} = \underline{I}$$

where  $\underline{Y}$  is a square nodal admittance matrix of constant, complex coefficients, and  $\underline{I}$  is a specified excitation vector. Set up the appropriate objective function for the least squares solution of this system of equations and derive the gradient vector w.r.t. the real and imaginary parts of the components of  $\underline{V}$ .

Answer (30 min.)

Let the real and the imaginary parts of  $\underline{V}$  be represented by  $\underline{V}_R$  and  $\underline{V}_I$ , respectively, i.e.,

$$\underline{V} = \underline{V}_R + j \underline{V}_I.$$

Let

$$\underline{e}(\underline{V}_R, \underline{V}_I) = \underline{Y} \underline{V} - \underline{I}.$$

Objective function for least squares solution of  $\underline{Y} \underline{V} = \underline{I}$  is

$$U(\underline{V}_R, \underline{V}_I) \triangleq \underline{e}^T \underline{e}^* = \sum_i |e_i|^2.$$

$$\frac{\partial U}{\partial \phi} = \frac{\partial \underline{e}^T}{\partial \phi} \underline{e}^* + \underline{e}^T \frac{\partial \underline{e}^*}{\partial \phi} = 2 \operatorname{Re} \left\{ \frac{\partial \underline{e}^T}{\partial \phi} \underline{e}^* \right\}$$

$$= 2 \operatorname{Re} \left\{ \frac{\partial (\underline{Y}^T \underline{Y}^T - \underline{I}^T)}{\partial \phi} (\underline{Y}^* \underline{V}^* - \underline{I}^*) \right\} = 2 \operatorname{Re} \left\{ \frac{\partial \underline{Y}^T}{\partial \phi} \underline{Y}^T (\underline{Y}^* \underline{V}^* - \underline{I}^*) \right\},$$

where  $\underline{Y}$  has been considered constant. Also, notice that

$$\frac{\partial \underline{V}^T}{\partial \underline{V}_R} = \underline{1} \quad \text{and} \quad \frac{\partial \underline{V}^T}{\partial \underline{V}_I} = j \underline{1},$$

where  $\underline{1}$  is an identity matrix. Therefore, the gradient vector of  $U$  w.r.t. the real and imaginary parts of  $\underline{V}$  is

$$\begin{bmatrix} \frac{\partial U}{\partial \underline{V}_R} \\ \frac{\partial U}{\partial \underline{V}_I} \end{bmatrix} = 2 \begin{bmatrix} \operatorname{Re} \{ \underline{Y}^T (\underline{Y}^* \underline{V}^* - \underline{I}^*) \} \\ \operatorname{Re} \{ j \underline{Y}^T (\underline{Y}^* \underline{V}^* - \underline{I}^*) \} \end{bmatrix} = 2 \begin{bmatrix} \operatorname{Re} \{ \underline{Y}^T (\underline{Y}^* \underline{V}^* - \underline{I}^*) \} \\ -\operatorname{Im} \{ \underline{Y}^T (\underline{Y}^* \underline{V}^* - \underline{I}^*) \} \end{bmatrix}$$

THE END! □

**EE3K4 COMPUTATIONAL METHODS AND DESIGN I**

Dr. J.W. Bandler

DURATION OF TEST: 2 hours

Wednesday, March 28, 1984

**THIS IS AN OPEN BOOK TEST**

Candidates must attempt Questions

1, 2, 3, 4, 5 or 6, 7

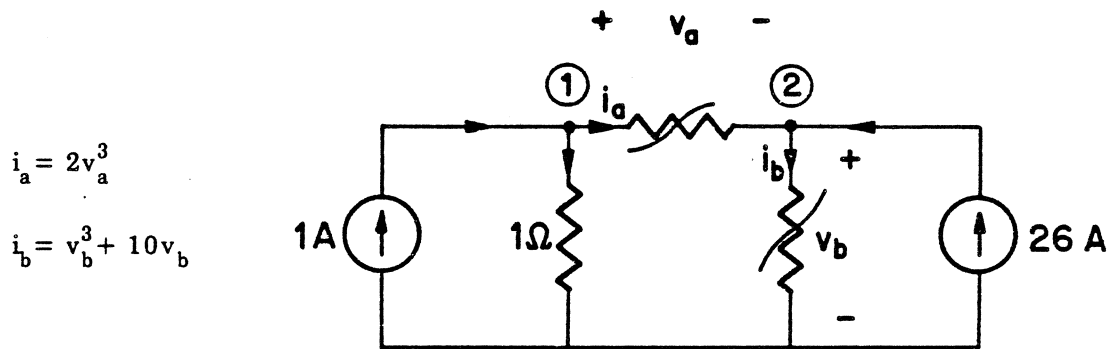
Write your name here. NAME: \_\_\_\_\_

Write your student number here. NO.: \_\_\_\_\_

- Note: (1) All scripts and question papers must be turned in.  
(2) Estimated times required to complete the questions are indicated.  
(3) Please encircle questions attempted in the following table.

Questions Attempted (please encircle)	Weighting	Estimated Time (min.)	Examiner's Use Only
1	10%	ten	
2	10%	ten	
3	15%	fifteen	
4	20%	thirty	
5 or 6	20%	twenty	
7	25%	thirty-five	
<b>TOTAL</b>	<b>100%</b>	<b>2 hours</b>	

**Question 4** Consider the nonlinear circuit shown.



- Express the nodal equations in the linearized form required at the  $j$ th iteration of the Newton algorithm.
- Apply two iterations of the Newton method, starting at  $v_1 = 2$ ,  $v_2 = 1$ .
- Draw the companion network at the  $j$ th iteration and state the corresponding nodal equations.
- Continue with two iterations of the companion network method.

Answer (30 min)

Question 4

2

$$(a) \quad f_1 = v_1 + 2(v_1 - v_2)^3 - 1$$

$$f_2 = v_2^3 + 10v_2 - 2(v_1 - v_2)^3 - 26$$

$$\underline{J}^j = \begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \frac{\partial f_1}{\partial v_2} \\ \frac{\partial f_2}{\partial v_1} & \frac{\partial f_2}{\partial v_2} \end{bmatrix}^j = \begin{bmatrix} 1 + 6(v_1 - v_2)^2 & -6(v_1 - v_2)^2 \\ -6(v_1 - v_2)^2 & 3v_2^2 + 10 + 6(v_1 - v_2)^2 \end{bmatrix}^j$$

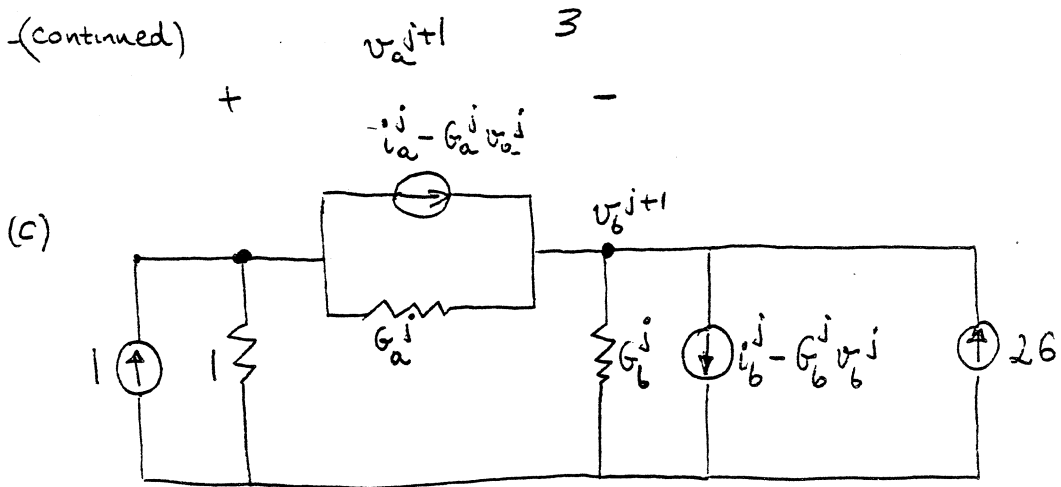
$$\underline{J}^j (\underline{v}^{j+1} - \underline{v}^j) = -\underline{f}^j$$

$$(b) \quad \underline{v}^0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \underline{J}^0 = \begin{bmatrix} 7 & -6 \\ -6 & 19 \end{bmatrix} \quad \underline{f}^0 = \begin{bmatrix} 3 \\ -17 \end{bmatrix}$$

$$\underline{v}^1 = \begin{bmatrix} 2.46 \\ 2.04 \end{bmatrix}$$

$$\underline{v}^2 = \begin{bmatrix} 1.60 \\ 1.88 \end{bmatrix}$$

Q4 (continued)



$$i_a^j = 2(v_a^3)^j$$

$$G_a^j = 6(v_a^2)^j$$

$$i_b^j = (v_b^3)^j + (10v_b)^j$$

$$G_b^j = (3v_b^2)^j + 10$$

$$v_a = v_1 - v_2$$

$$v_b = v_2$$

$$\begin{bmatrix} 1 + G_a^j & -G_a^j \\ -G_a^j & G_a^j + G_b^j \end{bmatrix} \begin{bmatrix} v_1^{j+1} \\ v_2^{j+1} \end{bmatrix} = \begin{bmatrix} 1 - (i_a^j - G_a^j v_a^j) \\ (i_a^j - G_a^j v_a^j) - (i_b^j - G_b^j v_b^j) + 26 \end{bmatrix}$$

(d)

$$v_0 = \begin{bmatrix} 1.60 \\ 1.88 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} 1.23 \\ 1.89 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} 1.32 \\ 1.89 \end{bmatrix}$$



Question 5 Refer to Assignment 2, namely, Question 18 of the COLLECTED PROBLEMS.

Evaluate

$$\partial I_3 / \partial R_i, \quad i = 1, 2, \dots, 7$$

for the numerical example, where  $I_3$  is the current flowing in resistor  $R_3$ .

Answer (20 min)

## Question 5

The original network is shown in Fig. 5.1. The solution

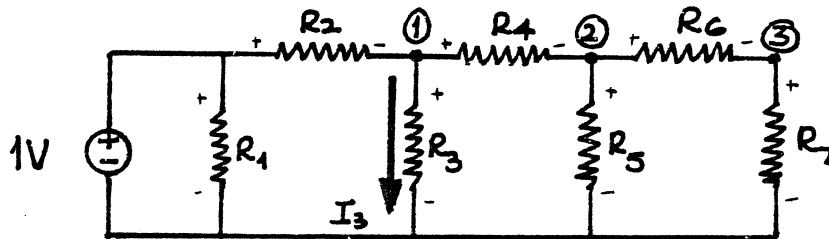


Fig. 5.1 Original network.

vector is given:  $\underline{V} = \left[ \frac{57}{97} \quad \frac{36}{97} \quad \frac{27}{97} \right]^T$  and the branch current vector  $\underline{I}^B$  is determined, i.e.

$$\underline{I}^B = \frac{1}{97} \left[ 97 \quad 120 \quad 57 \quad 63 \quad 36 \quad 27 \quad 27 \right]^T. \quad (5.1)$$

We express  $I_3$ , the current flowing in  $R_3$  as

$$I_3 = V_1 / R_3. \quad (5.2)$$

The sensitivities of  $I_3$  w.r.t  $\underline{\phi}$ , where  $\underline{\phi}$  comprises network parameters, is written using (5.2) as

$$\frac{\partial I_3}{\partial \underline{\phi}} = \frac{1}{R_3} \frac{\partial V_1}{\partial \underline{\phi}} - \frac{V_1}{R_3^2} \frac{\partial R_3}{\partial \underline{\phi}}. \quad (5.3)$$

(Note that  $\frac{\partial R_3}{\partial \underline{\phi}}$  will have unity, when  $\phi_i = R_3$ , and otherwise zero)

The adjoint network, in order to find  $\frac{\partial V_1}{\partial \underline{\phi}}$ , is shown in Fig. 5.2.

The  $G$ -matrix is unaltered and RHS vector has simply become one-third of the original RHS vector. Hence

$$\underline{V}^{\wedge} = \frac{1}{3} \underline{V} = \frac{1}{97} \left[ 19 \quad 12 \quad 9 \right]^T, \quad (5.4)$$

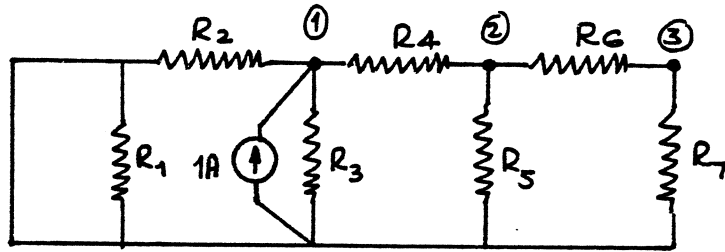


Fig. 5.2 Adjoint network.

and the branch currents are

$$\frac{\hat{I}}{\sim} B = \frac{1}{97} [0 \quad -57 \quad 19 \quad 21 \quad 12 \quad 9 \quad 9]^T. \quad (5.5)$$

Substituting (5.1) and (5.5) in the sensitivity expressions for resistors, we obtain (5.3) as

$$\frac{\partial I_3}{\partial \phi} = \frac{1}{97^2} [0 \quad -120 \times 57 \quad -78 \times 57 \quad 63 \times 21 \quad 36 \times 12 \quad 27 \times 9 \quad 27 \times 9]^T, \quad (5.6)$$

where  $\phi = [R_1 \ R_2 \ R_3 \ R_4 \ R_5 \ R_6 \ R_7]^T$ .

Question 6 Develop an algorithm to efficiently calculate the value of

$$Z_0 \frac{Z_L + jZ_0 \tan\theta}{Z_0 + jZ_L \tan\theta}$$

given real  $Z_0$ ,  $0 \leq \theta \leq \pi$  and complex  $Z_L$ , where  $j = \sqrt{-1}$ . Avoid  $\theta = \pi/2$ . State the number of needed multiplications and divisions, the number of additions and subtractions and the number of calls to a trigonometric function evaluation routine.

For 100 consecutive values of  $0 < \theta < \pi$ , estimate the minimum execution time on the University's Cyber or VAX.

Answer (20 min)

## Question 6

## Algorithm

- 1- Declare  $z_0$ ,  $z_L$ ,  $F$  and  $C$  to be complex.
- 2- IF  $|\theta - \frac{\pi}{2}| \gg \epsilon$ ,  
when  $\epsilon$  is a small positive number ( $\epsilon = 10^{-10}$  say),  
go to 4
- 3- Set  $F \leftarrow z_0 * z_0 / z_L$  and stop.
- 4-  $C \leftarrow j \tan \theta$
- 5- Set  $F \leftarrow z_0 * (z_L + C * z_0) / (z_0 + C * z_L)$

Operation count in steps 5 and 6

Number of multiplications and divisions = 4

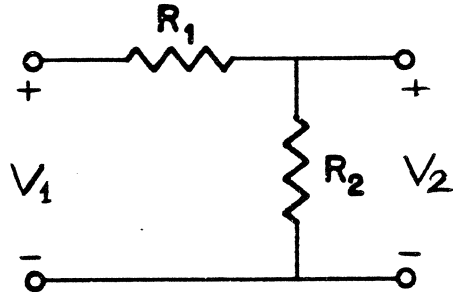
Number of additions = 2

Number of trigonometric function evaluations = 1

Execution time in seconds =  $3 * 0.0023 + 0.0034 + 0.0024 + 71 * 10^{-6}$

= 0.01277

Question 7 Consider the voltage divider shown.



Deriving all formulas from first principles, use the adjoint network method to calculate  $\partial T/\partial R_1$  and  $\partial T/\partial R_2$ , given

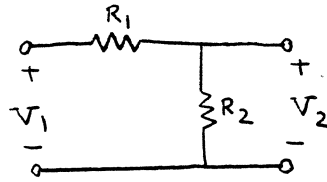
$$T = \frac{V_2}{V_1}, \quad R_1 = 2 \Omega, \quad R_2 = 1.5 \Omega.$$

Show both original and adjoint networks appropriately excited and verify your result by direct differentiation.

Derive an appropriate quadratic approximation formula from first principles and apply it to verify the two partial derivative values.

If the tolerance on  $R_1$  is  $\pm 5\%$  and on  $R_2$  is  $\pm 10\%$ , estimate the extreme values of  $T$  using first partial derivatives. Check the results by direct calculation.

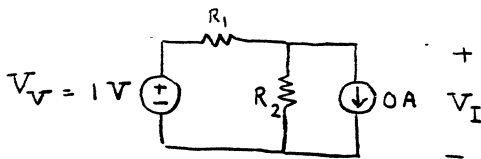
Answer (35 min)

Question 7

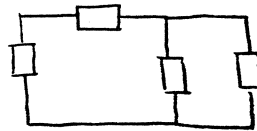
$$T = \frac{V_2}{V_1}, \quad R_1 = 2\Omega, \quad R_2 = 1.5\Omega$$

Since the circuit is linear, transfer function is independent of the numerical value of  $V_1$ . For convenience, assume  $V_1 = 1$ , therefore

$$T = \frac{V_2}{1} = V_2, \quad \text{and we want } \frac{\partial V_2}{\partial R_1}, \quad \frac{\partial V_2}{\partial R_2}$$



Original Network



Adjoint Network

From Tellegen's theorem:

$$\frac{\partial V_V}{\partial \phi} \hat{I}_V + \frac{\partial V_{R_1}}{\partial \phi} \hat{I}_{R_1} + \frac{\partial V_{R_2}}{\partial \phi} \hat{I}_{R_2} + \frac{\partial V_I}{\partial \phi} \hat{I}_I - \frac{\partial I_V}{\partial \phi} \hat{V}_V - \frac{\partial I_{R_1}}{\partial \phi} \hat{V}_{R_1} - \frac{\partial I_{R_2}}{\partial \phi} \hat{V}_{R_2} - \frac{\partial I_I}{\partial \phi} \hat{V}_I = 0$$

$$\text{Now: } \frac{\partial V_V}{\partial \phi} = 0, \quad \text{and} \quad \frac{\partial I_I}{\partial \phi} = 0$$

$$\text{Assume: } \hat{I}_I = -1, \quad \text{and} \quad \hat{V}_V = 0$$

$$\therefore \frac{\partial V_I}{\partial \phi} = \frac{\partial V_{R_1}}{\partial \phi} \hat{I}_{R_1} + \frac{\partial V_{R_2}}{\partial \phi} \hat{I}_{R_2} - \frac{\partial I_{R_1}}{\partial \phi} \hat{V}_{R_1} - \frac{\partial I_{R_2}}{\partial \phi} \hat{V}_{R_2}$$

$$\text{Use } I_{R_1} R_1 = V_{R_1}, \quad I_{R_2} R_2 = V_{R_2}$$

$$\begin{aligned} \therefore \frac{\partial V_I}{\partial \phi} &= \left( R_1 \frac{\partial I_{R_1}}{\partial \phi} + \frac{\partial R_1}{\partial \phi} I_{R_1} \right) \hat{I}_{R_1} + \left( R_2 \frac{\partial I_{R_2}}{\partial \phi} + \frac{\partial R_2}{\partial \phi} I_{R_2} \right) \hat{I}_{R_2} - \frac{\partial I_{R_1}}{\partial \phi} \hat{V}_{R_1} - \frac{\partial I_{R_2}}{\partial \phi} \hat{V}_{R_2} \\ &= \left( R_1 \hat{I}_{R_1} - \hat{V}_{R_1} \right) \frac{\partial I_{R_1}}{\partial \phi} + \left( R_2 \hat{I}_{R_2} - \hat{V}_{R_2} \right) \frac{\partial I_{R_2}}{\partial \phi} + \frac{\partial R_1}{\partial \phi} I_{R_1} \hat{I}_{R_1} + \frac{\partial R_2}{\partial \phi} I_{R_2} \hat{I}_{R_2} \end{aligned}$$

$$\text{Assume } R_1 \hat{I}_{R_1} = \hat{V}_{R_1}, \quad \text{and} \quad R_2 \hat{I}_{R_2} = \hat{V}_{R_2}$$

$$\therefore \frac{\partial V_I}{\partial \phi} = \frac{\partial R_1}{\partial \phi} I_{R_1} \hat{I}_{R_1} + \frac{\partial R_2}{\partial \phi} I_{R_2} \hat{I}_{R_2}$$

and the adjoint is completely // defined as



$$\text{for } \phi = R_1 \quad \frac{\partial V_1}{\partial R_1} = \frac{\partial V_2}{\partial R_1} = \frac{\partial T}{\partial R_1} = I_{R_1} \hat{I}_{R_1}$$

$$\text{for } \phi = R_2 \quad \frac{\partial V_1}{\partial R_2} = \frac{\partial V_2}{\partial R_2} = \frac{\partial T}{\partial R_2} = I_{R_2} \hat{I}_{R_2}$$

Now: analyzing the original and adjoint networks :

$$I_{R_1} = I_{R_2} = \frac{1}{R_1 + R_2} \quad , \quad \hat{I}_{R_1} = \frac{-R_2}{R_1 + R_2} \quad , \quad \hat{I}_{R_2} = \frac{R_1}{R_1 + R_2}$$

$$\therefore \frac{\partial T}{\partial R_1} = \frac{-R_2}{(R_1 + R_2)^2} \quad , \quad \frac{\partial T}{\partial R_2} = \frac{R_1}{(R_1 + R_2)^2}$$

Direct differentiation:

$$T = \frac{R_2}{R_1 + R_2}$$

$$\frac{\partial T}{\partial R_1} = \frac{-R_2}{(R_1 + R_2)^2}$$

$$\frac{\partial T}{\partial R_2} = \frac{R_1}{(R_1 + R_2)^2}$$

Numerically  $\frac{\partial T}{\partial R_1} = -0.1224$  ,  $\frac{\partial T}{\partial R_2} = 0.1633$

For a quadratic function

$$U = a\phi^2 + b\phi + c$$

$$U(\phi + \Delta\phi) = a(\phi + \Delta\phi)^2 + b(\phi + \Delta\phi) + c \quad , \quad U(\phi - \Delta\phi) = a(\phi - \Delta\phi)^2 + b(\phi - \Delta\phi) + c$$

$$\frac{U(\phi + \Delta\phi) - U(\phi - \Delta\phi)}{2\Delta\phi} = \frac{a[\phi^2 + (\Delta\phi)^2 + 2\phi\Delta\phi - \phi^2 - (\Delta\phi)^2 + 2\phi\Delta\phi] + b[\phi + \Delta\phi - \phi + \Delta\phi]}{2\Delta\phi}$$

$$= \frac{4a\phi\Delta\phi + 2b\Delta\phi}{2\Delta\phi} = 2a\phi + b$$

Also,  $\frac{\partial U}{\partial \phi} = 2a\phi + b$



12

∴ For quadratic functions  $\frac{\partial U}{\partial \phi} = \frac{U(\phi + \Delta\phi) - U(\phi - \Delta\phi)}{2\Delta\phi}$

For a general function  $\frac{\partial U}{\partial \phi} \approx \frac{U(\phi + \Delta\phi) - U(\phi - \Delta\phi)}{2\Delta\phi}$  if  $\Delta\phi$  is small

In this question  $T(R_1, R_2) = \frac{R_2}{R_1 + R_2}$

$$A = \frac{T(R_1 + \Delta R_1, R_2) - T(R_1 - \Delta R_1, R_2)}{2\Delta R_1} = \frac{\frac{R_2}{R_2 + R_1 + \Delta R_1} - \frac{R_2}{R_2 + R_1 - \Delta R_1}}{2\Delta R_1} = \frac{-R_2}{(R_2 + R_1)^2 - (\Delta R_1)^2}$$

$$B = \frac{T(R_1, R_2 + \Delta R_2) - T(R_1, R_2 - \Delta R_2)}{2\Delta R_2} = \frac{R_1}{(R_2 + R_1)^2 - (\Delta R_1)^2}$$

If  $\Delta R_1, \Delta R_2$  are small,  $(\Delta R_1)^2 \approx 0$ ,  $(\Delta R_2)^2 \approx 0$

and

$$A \approx \frac{-R_2}{(R_1 + R_2)^2} = \frac{\partial T}{\partial R_1}, \quad B \approx \frac{R_1}{(R_1 + R_2)^2} = \frac{\partial T}{\partial R_2}$$

$$R_1 = R_1 \pm 0.05 R_1$$

$$R_2 = R_2 \pm 0.1 R_2$$

$$\Delta T = \frac{\partial T}{\partial R_1} \Delta R_1 + \frac{\partial T}{\partial R_2} \Delta R_2$$

$$(\Delta T)_+ = -0.1224 \times (-0.1) + (0.1633) \times 0.15 = 0.0367$$

$$(\Delta T)_- = -0.1224 \times (0.1) + (0.1633) \times (-0.15) = -0.0367$$

$$T = \frac{1.5}{3.5} = 0.4286$$

$$T_{\max} = 0.4286 + 0.0367 = 0.4653$$

$$T_{\min} = 0.4286 - 0.0367 = 0.3919$$

Direct calculation:

$$T_{\max} = \frac{1.65}{1.65 + 1.9} = 0.4648$$

$$T_{\min} = \frac{1.35}{1.35 + 2.1} = 0.3913$$

□

## Question 1

Derive from first principles an approach to finding  $\partial\lambda/\partial\mathbf{x}$ , where  $\lambda$  is an eigenvalue of the square matrix  $\mathbf{A}$  whose coefficients are (in general) nonlinear functions of  $\mathbf{x}$ , i.e.,

$$\mathbf{A} \mathbf{y} = \lambda \mathbf{y}$$

The expression  $\partial\lambda/\partial\mathbf{x}$  is a column vector containing all first partial derivatives of  $\lambda$  w.r.t. corresponding elements of the column vector  $\mathbf{x}$ . Discuss the computational effort involved. Give interpretations of any new symbols introduced. [Hint:  $\lambda$  is also an eigenvalue of  $\mathbf{A}^T$ .]

Answer (20 minutes)

By definition, the eigen value must satisfy

$$\underline{\mathbf{A}}(\underline{\mathbf{x}}) \underline{\mathbf{y}} = \lambda \underline{\mathbf{y}} \quad (1) \quad \text{and} \quad \underline{\mathbf{A}}^T \underline{\mathbf{u}} = \lambda \underline{\mathbf{u}} \quad (2)$$

Differentiating equation (1) w.r.t.  $x_j$  we have

$$\frac{\partial \underline{\mathbf{A}}(\underline{\mathbf{x}})}{\partial x_j} \underline{\mathbf{y}} + \underline{\mathbf{A}}(\underline{\mathbf{x}}) \frac{\partial \underline{\mathbf{y}}}{\partial x_j} = \frac{\partial \lambda}{\partial x_j} \underline{\mathbf{y}} + \lambda \frac{\partial \underline{\mathbf{y}}}{\partial x_j} \quad (3)$$

Multiplying both sides by  $\underline{\mathbf{u}}^T$ :

$$\underline{\mathbf{u}}^T \frac{\partial \underline{\mathbf{A}}(\underline{\mathbf{x}})}{\partial x_j} \underline{\mathbf{y}} + \underline{\mathbf{u}}^T \underline{\mathbf{A}}(\underline{\mathbf{x}}) \frac{\partial \underline{\mathbf{y}}}{\partial x_j} = \underline{\mathbf{u}}^T \frac{\partial \lambda}{\partial x_j} \underline{\mathbf{y}} + \underline{\mathbf{u}}^T \lambda \frac{\partial \underline{\mathbf{y}}}{\partial x_j}$$

Rearranging it

$$\underline{\mathbf{u}}^T \frac{\partial \underline{\mathbf{A}}(\underline{\mathbf{x}})}{\partial x_j} \underline{\mathbf{y}} + \underline{\mathbf{u}}^T [\underline{\mathbf{A}}(\underline{\mathbf{x}}) - \lambda \underline{\mathbf{I}}] \frac{\partial \underline{\mathbf{y}}}{\partial x_j} = \underline{\mathbf{u}}^T \frac{\partial \lambda}{\partial x_j} \underline{\mathbf{y}}$$

Because  $\underline{\mathbf{A}}(\underline{\mathbf{x}}) - \lambda \underline{\mathbf{I}} = 0$

we have

$$\underline{\mathbf{u}}^T \frac{\partial \underline{\mathbf{A}}(\underline{\mathbf{x}})}{\partial x_j} \underline{\mathbf{y}} = \underline{\mathbf{u}}^T \frac{\partial \lambda}{\partial x_j} \underline{\mathbf{y}}$$

# ELECTRICAL ENGINEERING 3K4

DAY CLASS

Dr. J.W. Bandler

DURATION OF EXAMINATION: 3 Hours  
McMaster University Final Examination

April 1987

---

This examination paper includes 11 pages and 10 questions. You are responsible for ensuring that your copy of the paper is complete. Bring any discrepancy to the attention of your invigilator.

---

## SPECIAL INSTRUCTIONS:

Candidates may use slide rules, calculators and log books.

Candidates must not use preprogrammed algorithms, such as those for solving linear or nonlinear equations, etc.

Candidates must attempt Questions

1 OR 2 OR 3, 4 OR 5, 6, 7 OR 8, 9, 10

Write your name here. NAME: \_\_\_\_\_

Write your student number here. NO: \_\_\_\_\_

Date and hour of examination: \_\_\_\_\_

- Note: (1) All scripts and question papers must be turned in.  
(2) Estimated times required to complete the questions are indicated.  
(3) Please encircle questions attempted in the following table.

Questions Attempted (please encircle)	Weighting	Estimated Time (min.)	Examiner's Use only
1 or 2 or 3	10%	twenty	
4 or 5	15%	twenty	
6	15%	thirty-five	
7 or 8	20%	thirty	
9	20%	thirty	
10	20%	forty-five	
TOTAL	100%	3 hours	

or

$$\frac{\partial \lambda}{\partial x_j} = \left( \underbrace{u^T}_{\sim} \frac{\partial A}{\partial x_j} \underbrace{y}_{\sim} \right) \frac{1}{\underbrace{u^T y}_{\sim}}$$

So for all  $x_j$ 's

$$\frac{\partial \lambda}{\partial \underline{x}} = \left[ \frac{\underbrace{u^T}_{\sim} \frac{\partial A}{\partial x_1} \underbrace{y}_{\sim}}{\underbrace{u^T y}_{\sim}} \quad \frac{\underbrace{u^T}_{\sim} \frac{\partial A}{\partial x_2} \underbrace{y}_{\sim}}{\underbrace{u^T y}_{\sim}} \quad \dots \quad \frac{\underbrace{u^T}_{\sim} \frac{\partial A(x)}{\partial x_R} \underbrace{y}_{\sim}}{\underbrace{u^T y}_{\sim}} \right]^T$$

The computational effort involved is as following:

i) To solve  $\underline{u}$  from (2)

ii) To obtain  $\frac{\partial A(x)}{\partial x_i}$ ,  $i = 1, 2, \dots, R$ .

iii) To multiply  $\underbrace{u^T}_{\sim} \frac{\partial A(x)}{\partial x_i} \underbrace{y}_{\sim}$   $R$  times

iv) To get  $\underbrace{u^T y}_{\sim}$

v) To divide  $\underbrace{u^T}_{\sim} \frac{\partial A}{\partial x_i} \underbrace{y}_{\sim}$  by  $\underbrace{u^T y}_{\sim}$  for  $i = 1, 2, \dots, R$ .

**Question 2**

Consider the resistor-diode network shown. Draw the corresponding companion network at the  $j$ th iteration for its d.c. solution. Write down the nodal equations at this iteration.

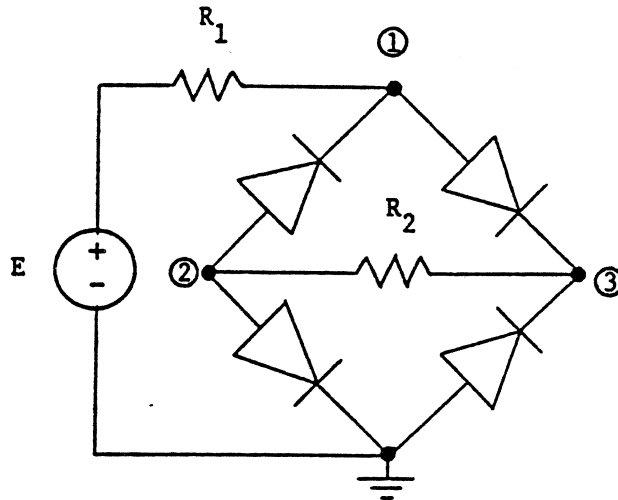
$$i_d = I_S (e^{\lambda v_d} - 1)$$

$$I_S = 10^{-12} \text{ mA}$$

$$\lambda = 1/0.026 \text{ V}^{-1}$$

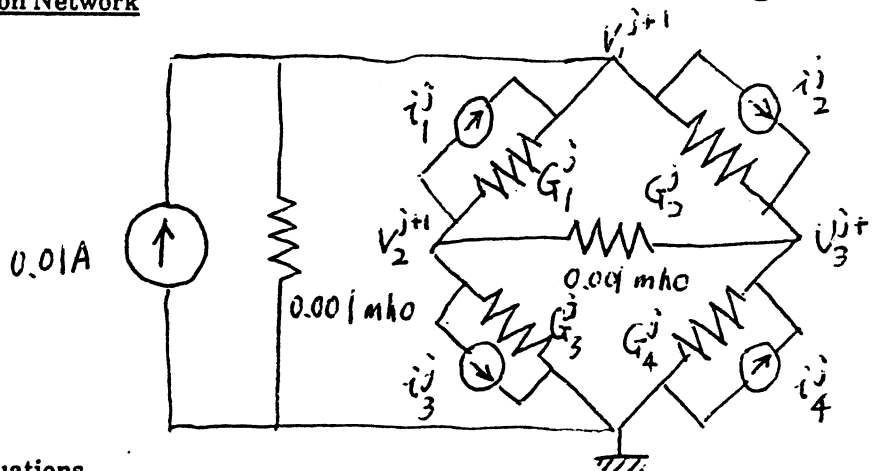
$$E = 10 \text{ V}$$

$$R_1 = R_2 = 1 \text{ k}\Omega$$



**Answer (20 minutes)**

Companion Network



Nodal Equations

$$\begin{bmatrix} 0.001 + G_1^j + G_2^j & -G_1^j & -G_2^j \\ -G_1^j & 0.001 + G_1^j + G_3^j & -0.001 \\ -G_2^j & -0.001 & 0.001 + G_2^j + G_4^j \end{bmatrix} \begin{bmatrix} v_1^{j+1} \\ v_2^{j+1} \\ v_3^{j+1} \end{bmatrix}$$

$$= \begin{bmatrix} 0.01 + (i_{d1}^j - G_1^j v_{d1}^j) - (i_{d2}^j - G_2^j v_{d2}^j) \\ - (i_{d1}^j - G_1^j v_{d1}^j) - (i_{d3}^j - G_3^j v_{d3}^j) \\ (i_{d2}^j - G_2^j v_{d2}^j) - (i_{d4}^j - G_4^j v_{d4}^j) \end{bmatrix}$$

where

$$v_{d_1}^j = v_2^j - v_1^j, \quad v_{d_2}^j = v_1^j - v_3^j, \quad v_{d_3}^j = v_2^j, \quad v_{d_4}^j = -v_3^j$$

and

$$\left. \begin{aligned} i_{d_k}^j &= I_s (e^{\lambda v_{d_k}^j} - 1) \\ G_R^j &= \lambda I_s e^{\lambda v_{d_k}^j} \end{aligned} \right\} \quad k=1, 2, 3, 4$$

## Question 3

Derive from first principles Newton's method (a) for function minimization w.r.t. many variables and (b) for solving nonlinear equations. Under what conditions would you expect proper convergence? State carefully and discuss the effects and theoretical interpretation of damping. Use diagrams to illustrate your results.

Answer (20 minutes)

Solution:

(a) If the first and second derivatives of  $u(\underline{\phi})$  are available, a quadratic model of  $u$  can be obtained by taking the first three terms of the Taylor-series expansion about the current point  $\underline{\phi}_0$ , i.e.

$$u(\underline{\phi}_0 + \Delta\underline{\phi}) \approx u(\underline{\phi}_0) + \nabla u^T \Delta\underline{\phi} + \frac{1}{2} \Delta\underline{\phi}^T H \Delta\underline{\phi}$$

$$\text{or } u(\underline{\phi}_0 + \Delta\underline{\phi}) - u(\underline{\phi}_0) \approx \nabla u^T \Delta\underline{\phi} + \frac{1}{2} \Delta\underline{\phi}^T H \Delta\underline{\phi}$$

The minimal point can be obtained by solving above equation in new variables  $\Delta\underline{\phi}$ . This can be done by differentiating above equation and letting it be zero:

$$H \Delta\underline{\phi} = - \nabla u$$

(b) For a set of nonlinear equations  $F(\underline{\phi})$ , we have

$$F(\underline{\phi} + \Delta\underline{\phi}) = F(\underline{\phi}) + J \Delta\underline{\phi} + \dots$$

$$\text{or } F(\underline{\phi} + \Delta\underline{\phi}) - F(\underline{\phi}) \approx J \Delta\underline{\phi}$$

We hope that  $\Delta\phi$  could make  $F(\phi + \Delta\phi) = 0$ . So we have

$$\underline{J}\Delta\phi = -F(\phi).$$

or

$$\Delta\phi = -\underline{J}^{-1}F(\phi).$$

The success of using the Newton's method is based on how adequately the objective function is represented by a quadratic form. If  $\underline{H}$  is positive definite we can expect proper convergence. But for some cases  $-\underline{H}^{-1}\nabla u$  might not point downhill, the damping factor  $\alpha$  could be introduced

$$\Delta\phi = -\alpha \underline{H}^{-1}\nabla u$$

where  $\alpha$  is chosen to minimize  $u$ .



## Question 4

Consider the linear circuit shown which is assumed to be in the sinusoidal steady state.

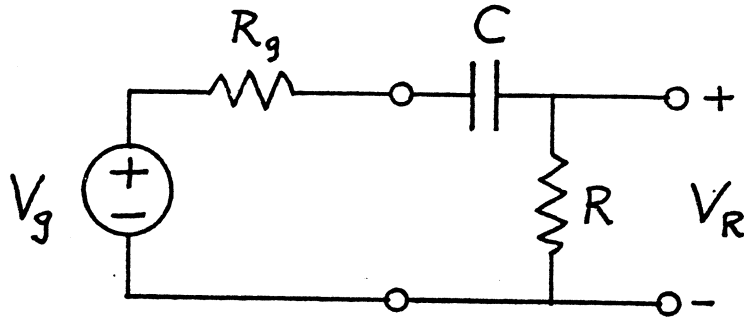
$$V_g = 1 \text{ V}$$

$$R_g = 0.5 \Omega$$

$$C = 2 \text{ F}$$

$$R = 1 \Omega$$

$$\omega = 10 \text{ rad/s}$$



Use the adjoint network approach to evaluate  $\partial|V_R|/\partial\omega$ . Estimate the changes in  $|V_R|$  when  $\omega$  changes by  $\pm 1\%$  using this partial derivative and compare with the exact changes.

Answer (20 minutes)

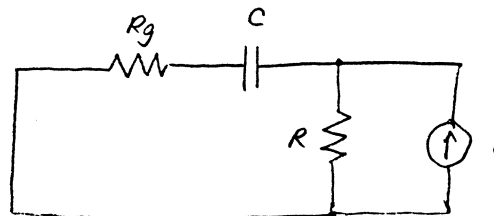
$$\therefore \frac{\partial V_R}{\partial \omega} = -V_{Rg} \hat{V}_{Rg} \cdot \frac{\partial(\frac{1}{R_g})}{\partial \omega} - V_R \hat{V}_R \frac{\partial(\frac{1}{R})}{\partial \omega} - V_C \hat{V}_C \frac{\partial(j\omega C)}{\partial \omega} = -j\omega C V_C \hat{V}_C$$

$$V_R = \frac{R V_g}{R_g + R + \frac{1}{j\omega C}} = \frac{1}{1.5 - j0.05}$$

$$V_C = \frac{\frac{1}{j\omega C} V_g}{R_g + R + \frac{1}{j\omega C}} = \frac{1}{1 + j30}$$

Adjoint network:

$$\hat{V}_C = \frac{R(R_g + \frac{1}{j\omega C})}{R + R_g + \frac{1}{j\omega C}} = \frac{1}{1 + j30}$$



$$\begin{aligned} \therefore \frac{\partial |V_R|}{\partial \omega} &= |V_R| \cdot \text{Re} \left( \frac{1}{V_R} \frac{\partial V_R}{\partial \omega} \right) = \frac{1}{|V_R|} \cdot \text{Re} \left( V_R^* \frac{\partial V_R}{\partial \omega} \right) \\ &= \sqrt{1.5^2 + 0.05^2} \cdot \text{Re} \left[ \frac{1}{1.5 + j0.05} \cdot \frac{-2j}{(1 + j30)^2} \right] \\ &= -7.39 \times 10^{-5} \end{aligned}$$

Continued on page 6

Question 4 (cont.)

$$\Delta|V_R| \approx \frac{\partial|V_R|}{\partial\omega} \cdot \Delta\omega = \pm \frac{\partial|V_R|}{\partial\omega} \cdot 0.01 \times 10 = \pm 7.39 \times 10^{-6}$$

Exact change:

$$\Delta|V_R| \Big|_{\Delta\omega=0.1} = \frac{1}{\sqrt{1.5^2 + 0.05^2}} - \frac{1}{\sqrt{1.5^2 + \left(\frac{1}{20.2}\right)^2}} = 1.6411 \times 10^{-5}$$

$$\Delta|V_R| \Big|_{\Delta\omega=-0.1} = \frac{1}{\sqrt{1.5^2 + 0.05^2}} - \frac{1}{\sqrt{1.5^2 + \left(\frac{1}{19.8}\right)^2}} = -1.6911 \times 10^{-5}$$

## Question 5

Consider a system of complex linear equations

$$YV = I$$

where  $Y$  is a square nodal admittance matrix of constant, complex coefficients, and  $I$  is a specified excitation vector. Set up the appropriate objective function for the least squares solution of this system of equations and derive the gradient vector w.r.t. the real and imaginary parts of the components of  $V$ .

Answer (20 minutes)

Let the real and the imaginary parts of  $\underline{V}$  be represented by  $\underline{V}_R$  and  $\underline{V}_I$ , respectively, i.e.,

$$\underline{V} = \underline{V}_R + j\underline{V}_I$$

Let

$$\underline{e}(\underline{V}_R, \underline{V}_I) = Y\underline{V} - \underline{I}$$

Objective function for least squares solution of  $Y\underline{V} = \underline{I}$  is

$$U(\underline{V}_R, \underline{V}_I) \triangleq \underline{e}^T \underline{e}^* = \sum_i |e_i|^2$$

$$\frac{\partial U}{\partial \phi} = \frac{\partial \underline{e}^T}{\partial \phi} \underline{e}^* + \underline{e}^T \frac{\partial \underline{e}^*}{\partial \phi} = 2 \operatorname{Re} \left\{ \frac{\partial \underline{e}^T}{\partial \phi} \underline{e}^* \right\}$$

$$= 2 \operatorname{Re} \left\{ \frac{\partial (\underline{V}^T Y^T - \underline{I}^T)}{\partial \phi} (Y^* \underline{V}^* - \underline{I}^*) \right\} = 2 \operatorname{Re} \left\{ \frac{\partial \underline{V}^T}{\partial \phi} Y^T (Y^* \underline{V}^* - \underline{I}^*) \right\}$$

Where  $Y$  has been considered constant. Also, notice that

$$\frac{\partial \underline{V}^T}{\partial \underline{V}_R} = \underline{1} \quad \text{and} \quad \frac{\partial \underline{V}^T}{\partial \underline{V}_I} = j\underline{1}$$

where  $\underline{1}$  is an identity matrix. Therefore, the gradient vector of  $U$  w.r.t. the real and imaginary parts of  $\underline{V}$  is

$$\begin{bmatrix} \frac{\partial U}{\partial \underline{V}_R} \\ \frac{\partial U}{\partial \underline{V}_I} \end{bmatrix} = 2 \begin{bmatrix} \operatorname{Re} \{ Y^T (Y^* \underline{V}^* - \underline{I}^*) \} \\ \operatorname{Re} \{ j Y^T (Y^* \underline{V}^* - \underline{I}^*) \} \end{bmatrix} = 2 \begin{bmatrix} \operatorname{Re} \{ Y^T (Y^* \underline{V}^* - \underline{I}^*) \} \\ -\operatorname{Im} \{ Y^T (Y^* \underline{V}^* - \underline{I}^*) \} \end{bmatrix}$$

Continued on page 7

**Question 6**

Derive from first principles an efficient algorithm for solving the tridiagonal system of equations

$$Ax = d$$

for  $x$ , given arbitrary vector  $d$ , where

$$A = \begin{bmatrix} a_1 & & & c_1 & & & \\ & b_2 & & a_2 & & c_2 & \\ & & \cdot & & \cdot & & \cdot \\ & & & \cdot & & \cdot & \\ & & & & b_{n-1} & & a_{n-1} & & c_{n-1} \\ & & & & & b_n & & & a_n \end{bmatrix}$$

using the one-dimensional arrays  $a_1, a_2, \dots, a_n, b_2, \dots, b_n, c_1, \dots, c_{n-1}$ , explicitly.

**Answer (35 minutes)**

*Solution:*

I) Divide the first row by  $a_1$  i.e.  $0/a_1$ .

$$\begin{bmatrix} 1 & & & & & & \\ b_2 & a_2 & & c_2 & & & \\ & & \cdot & & \cdot & & \\ & & & \cdot & & \cdot & \\ & & & & b_{n-1} & & a_{n-1} & & c_{n-1} \\ & & & & & b_n & & & a_n \end{bmatrix} x = \begin{bmatrix} d_1/a_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

② -  $b_2 \times$  ①

$$\begin{bmatrix} 1 & c_1/a_1 & & & & & \\ 0 & a_2 - b_2 c_1/a_1 & & c_2 & & & \\ & & \cdot & & \cdot & & \\ & & & \cdot & & \cdot & \\ & & & & b_{n-1} & & a_{n-1} & & c_{n-1} \\ & & & & & b_n & & & a_n \end{bmatrix} x = \begin{bmatrix} d_1/a_1 \\ d_2 - b_2 d_1/a_1 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}$$

Let  $c_1/a_1 = c'_1$ ,  $a_2' = a_2 - b_2 c_1/a_1$ ,  $d_1' = d_1/a_1$  and  
 $d_2' = d_2 - b_2 d_1/a_1$ .

$$\left[ \begin{array}{c|ccc} 1 & c'_1 & & \\ 0 & a_2' & c_2 & \\ & & \ddots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{array} \right] \underline{x} = \begin{bmatrix} d_1' \\ d_2' \\ \vdots \\ d_n \end{bmatrix}$$

The  $(n-1) \times (n-1)$  submatrix in the above expression has the same structure as that of  $\underline{A}$ . If we continue the same process, we'll get

$$\left[ \begin{array}{c|ccc} 1 & c'_1 & & \\ & 1 & c'_2 & \\ & & \ddots & \\ & & & 1 & c'_{n-1} \\ & & & & 1 \end{array} \right] \underline{x} = \begin{bmatrix} d_1' \\ d_2' \\ \vdots \\ d_n' \end{bmatrix}$$

The backward substitution can solve this set of equations for  $\underline{x}$ .

II) Algorithm:

Step 1:  $c_1 \leftarrow c_1/a_1$ ,  $d_1 \leftarrow d_1/a_1$ ,  $a_1 \leftarrow 1$ ;

$i \leftarrow 2$ ;

Step 2:  $a_i \leftarrow a_i - b_i c_{i-1}$ ,  $d_i \leftarrow d_i - b_i d_{i-1}$ ;

Step 3:  $c_i \leftarrow c_i/a_i$ ,  $d_i \leftarrow d_i/a_i$ ,

$a_i \leftarrow 1$ .

$i \leftarrow i+1$ ;

Step 4: If  $i = N$ , goto step 5;  
otherwise goto step 2;

Step 5:  $a_n \leftarrow a_n - b_n c_{n-1}$ ,  
 $d_n \leftarrow d_n - b_n d_{n-1}$ ,  
 $d_n \leftarrow d_n / a_n$ ,

$a_n \leftarrow i$ ;  
Step 6:  $x_n \leftarrow d_n$ ,

$i = n - 1$ ;

Step 7:  $x_i \leftarrow d_i - C_i x_{i+1}$ ;

$i = i - 1$ ;

If  $i < 1$  stop, otherwise goto the beginning of this step.

## Question 7

We wish to calculate  $\partial f / \partial \underline{x}$  subject to  $h(\underline{x}, \underline{y}) = 0$ , where  $f \equiv f(\underline{y}(\underline{x}), \underline{x})$  given values for  $\underline{x}$ .

Explain fully the formula

$$\left. \frac{\partial f}{\partial \underline{x}} \right|_{\underline{h}=0} = - \frac{\partial h^T}{\partial \underline{x}} \underline{\hat{y}} + \frac{\partial f}{\partial \underline{x}},$$

where  $\underline{\hat{y}}$  is the solution to

$$\left( \frac{\partial h^T}{\partial \underline{y}} \right) \underline{\hat{y}} = - \frac{\partial f}{\partial \underline{y}}.$$

Describe the computational and analytical effort required in any given problem.

Let

$$4x_1^2 y_1^2 - 3y_2 - 2 = 0,$$

$$-x_1 y_1 + 2x_2^2 y_1 y_2 - 3y_2 = 0,$$

$$f = y_1^2 + x_1.$$

Set up all the matrices and vectors required both for the solution of the nonlinear equations and also for the evaluation of  $\partial f / \partial \underline{x}$  subject to  $h = 0$ .

Answer (30 minutes)

$$\text{Given } f \equiv f(\underline{y}(\underline{x}), \underline{x}),$$

$$h(\underline{x}, \underline{y}) = 0.$$

$$\therefore \frac{\partial f}{\partial \underline{x}} = \frac{\partial \underline{y}^T}{\partial \underline{x}} \cdot \frac{\partial f}{\partial \underline{y}} + \frac{\partial f}{\partial \underline{x}}$$

The relationship between  $\underline{x}$  and  $\underline{y}$  is given by  $h(\underline{x}, \underline{y}) = 0$ .

$\therefore$  We should eliminate  $\frac{\partial \underline{y}^T}{\partial \underline{x}}$  by using  $h(\underline{x}, \underline{y}) = 0$ .

$$\frac{\partial h^T}{\partial \underline{x}} + \frac{\partial \underline{y}^T}{\partial \underline{x}} \frac{\partial h^T}{\partial \underline{y}} = 0 \quad \rightarrow \quad \frac{\partial \underline{y}^T}{\partial \underline{x}} = - \frac{\partial h^T}{\partial \underline{x}} \left( \frac{\partial h^T}{\partial \underline{y}} \right)^{-1}$$

Question 7 (cont.)

$$\therefore \left. \frac{\partial f}{\partial \underline{x}} \right|_{\underline{h}=0} = - \frac{\partial \underline{h}^T}{\partial \underline{x}} \left( \frac{\partial \underline{h}^T}{\partial \underline{y}} \right)^{-1} \frac{\partial f}{\partial \underline{y}} + \frac{\partial f}{\partial \underline{x}}$$

$$\text{Define } \left( \frac{\partial \underline{h}^T}{\partial \underline{y}} \right)^{-1} \frac{\partial f}{\partial \underline{y}} = \hat{\underline{y}} \quad \text{or} \quad \frac{\partial \underline{h}^T}{\partial \underline{y}} \hat{\underline{y}} = \frac{\partial f}{\partial \underline{y}}$$

$$\therefore \left. \frac{\partial f}{\partial \underline{x}} \right|_{\underline{h}=0} = - \frac{\partial \underline{h}^T}{\partial \underline{x}} \hat{\underline{y}} + \frac{\partial f}{\partial \underline{x}}$$

The analytical effort: derive explicitly  $\frac{\partial f}{\partial \underline{x}}$ ,  $\frac{\partial f}{\partial \underline{y}}$ ,  $\frac{\partial \underline{h}^T}{\partial \underline{x}}$  and  $\frac{\partial \underline{h}^T}{\partial \underline{y}}$ .

The computational effort:

If we do not include the computational effort involved in partial derivative computation, we have to solve one linear system  $\underline{A} \underline{y} = \underline{b}$  and another  $n^2$  multiplications. So totally

Also the computational effort should include those in solving  $\underline{h}(\underline{x}, \underline{y}) = 0$ .

$$\text{When } \underline{h} = \begin{bmatrix} 4x_1^2 y_1^2 - 3y_2 - 2 \\ -x_1 y_1 + 2x_2^2 y_1 y_2 - 3y_2 \end{bmatrix} = \underline{0}$$

$$\frac{\partial \underline{h}^T}{\partial \underline{x}} = \begin{bmatrix} 8x_1 y_1^2 & -y_1 \\ 0 & 4x_2 y_1 y_2 \end{bmatrix}$$



Question 7 (cont.)

$$\frac{\partial \underline{h}^T}{\partial \underline{y}} = \begin{bmatrix} 8x_1^2 y_1 & -x_1 + 2x_2^2 y_2 \\ -3 & 2x_2^2 y_1 - 3 \end{bmatrix},$$

$$\frac{\partial f}{\partial \underline{x}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \frac{\partial f}{\partial \underline{y}} = \begin{bmatrix} 2y_1 \\ 0 \end{bmatrix}.$$

the adjoint system

$$\begin{bmatrix} 8x_1^2 y_1 & -x_1 + 2x_2^2 y_2 \\ -3 & 2x_2^2 y_1 - 3 \end{bmatrix} \underline{\hat{y}} = \begin{bmatrix} 2y_1 \\ 0 \end{bmatrix}$$

$$\therefore \left. \frac{\partial f}{\partial \underline{x}} \right|_{\underline{h}=\underline{0}} = - \begin{bmatrix} 8x_1 y_1^2 & -y_1 \\ 0 & 4x_2 y_1 y_2 \end{bmatrix} \underline{\hat{y}} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Solution of the nonlinear equation  $\underline{h} = \underline{0}$  :

Use first order Taylor series,

$$\underline{h}^{j+1} = \underline{h}^j + \left. \left( \frac{\partial \underline{h}^T}{\partial \underline{y}} \right)^T \right|_j (\underline{y}^{j+1} - \underline{y}^j) = \underline{0},$$

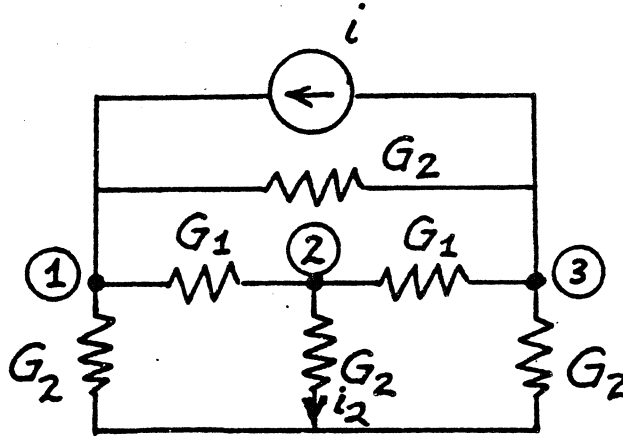
$$\left. \left( \frac{\partial \underline{h}^T}{\partial \underline{y}} \right)^T \right|_j \underline{y}^{j+1} = \left. \left( \frac{\partial \underline{h}^T}{\partial \underline{y}} \right)^T \right|_j \underline{y}^j - \underline{h}^j,$$

$$\begin{bmatrix} 8x_1^2 y_1 & -x_1 + 2x_2^2 y_2 \\ -3 & 2x_2^2 y_1 - 3 \end{bmatrix}_j^T \underline{y}^{j+1} = \begin{bmatrix} 8x_1^2 y_1 & -x_1 + 2x_2^2 y_2 \\ -3 & 2x_2^2 y_1 - 3 \end{bmatrix}_j^T \underline{y}^j - \underline{h}^j,$$

until  $\underline{h}^{j+1} = \underline{0}$ .

Question 8

Consider the resistive network shown.



$G_1 = 1.5$

$G_2 = 2.5$

$i = 10$

Use the adjoint network method to evaluate

$$\frac{\partial i_2}{\partial G_1}, \frac{\partial i_2}{\partial G_2}, \text{ and } \frac{\partial i_2}{\partial i}$$

Check your results by small perturbations.

Answer (30 minutes)

$$\underline{Y} \underline{V} = \underline{I}, \text{ where } \underline{Y} = \begin{bmatrix} G_1 + 2G_2 & -G_1 & -G_2 \\ -G_1 & 2G_1 + G_2 & -G_1 \\ -G_2 & -G_1 & G_1 + 2G_2 \end{bmatrix} = \begin{bmatrix} 6.5 & -1.5 & -2.5 \\ -1.5 & 5.5 & -1.5 \\ -2.5 & -1.5 & 6.5 \end{bmatrix}$$

$$\underline{I} = \begin{bmatrix} i \\ 0 \\ -i \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ -10 \end{bmatrix}$$

$$\underline{V} \text{ is solved as } \underline{V} = \begin{bmatrix} 10/9 \\ 0 \\ -10/9 \end{bmatrix}$$

$$\frac{\partial i_2}{\partial \phi} = \frac{\partial (V_2 G_2)}{\partial \phi} = \frac{\partial V_2}{\partial \phi} G_2 + V_2 \frac{\partial G_2}{\partial \phi}$$

$$\therefore \frac{\partial \underline{Y}}{\partial \phi} \underline{V} + \underline{Y} \frac{\partial \underline{V}}{\partial \phi} = \frac{\partial \underline{I}}{\partial \phi} \rightarrow \frac{\partial \underline{V}}{\partial \phi} = \underline{Y}^{-1} \left( \frac{\partial \underline{I}}{\partial \phi} - \frac{\partial \underline{Y}}{\partial \phi} \underline{V} \right)$$

$$\therefore \frac{\partial V_2}{\partial \phi} = \hat{V}^T \left( \frac{\partial \underline{I}}{\partial \phi} - \frac{\partial \underline{Y}}{\partial \phi} \underline{V} \right)$$

## Question 8 (cont.)

where  $\hat{\underline{V}}$  is solved from the adjoint network

$$\underline{Y}^T \hat{\underline{V}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{or} \quad \begin{bmatrix} 6.5 & -1.5 & -2.5 \\ -1.5 & 5.5 & -1.5 \\ -2.5 & -1.5 & 6.5 \end{bmatrix} \hat{\underline{V}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

But it can be seen that  $V_2 \equiv 0$  regardless of changes in  $G_1$ ,  $G_2$  or  $i$ . Therefore,  $\frac{\partial V_2}{\partial G_1} = \frac{\partial V_2}{\partial G_2} = \frac{\partial V_2}{\partial i} = 0$ .

$$\text{Therefore, } \frac{\partial i_2}{\partial G_1} = 0, \quad \frac{\partial i_2}{\partial G_2} = 0, \quad \frac{\partial i_2}{\partial i} = 0.$$

Or

$$\text{Solve } \underline{Y}^T \hat{\underline{V}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \rightarrow \hat{\underline{V}} = K \begin{bmatrix} 3 \\ 8 \\ 3 \end{bmatrix}, \quad \text{where } K = \frac{36}{\det(\underline{Y})}$$

$$\text{Case 1. } \phi = G_1, \quad \frac{\partial I}{\partial G_1} = 0, \quad \frac{\partial V_2}{\partial G_1} = -\hat{\underline{V}}^T \frac{\partial \underline{Y}}{\partial G_1} \underline{V} = -\hat{\underline{V}}^T \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \underline{V} = 0$$

$$\therefore \frac{\partial i_2}{\partial G_1} = 0.$$

$$\text{Case 2. } \phi = G_2, \quad \frac{\partial I}{\partial G_2} = 0, \quad \frac{\partial V_2}{\partial G_2} = -\hat{\underline{V}}^T \frac{\partial \underline{Y}}{\partial G_2} \underline{V} = -\hat{\underline{V}}^T \begin{bmatrix} 2 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix} \underline{V} = 0$$

$$\therefore \frac{\partial i_2}{\partial G_2} = 0.$$

$$\text{Case 3. } \phi = i, \quad \frac{\partial I}{\partial i} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \quad \frac{\partial V_2}{\partial i} = -\hat{\underline{V}}^T \frac{\partial \underline{Y}}{\partial i} \underline{V} = -\hat{\underline{V}}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \underline{V} = 0.$$

$$\therefore \frac{\partial i_2}{\partial i} = 0.$$

Check Results: Analytical:  $\underline{V} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \frac{10}{G_1 + 3G_2}$

$$V_2 \equiv 0$$

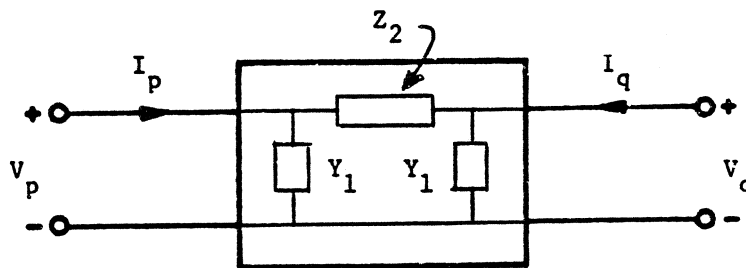
$$\frac{\partial i_2}{\partial \phi} = \frac{\partial V_2}{\partial \phi} G_2 + V_2 \frac{\partial G_2}{\partial \phi} \equiv 0.$$

## Question 9

Derive from first principles the adjoint element equation and sensitivity expression for a two-port characterized by

$$\begin{bmatrix} V_p \\ I_p \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix}$$

Apply the result to the element shown in the diagram to determine the sensitivity formulas w.r.t.  $\phi$ , where  $Y_1 = \phi$  and  $Z_2 = 0.5/\phi$ .



Answer (30 minutes)

Solution:

From the Tellegen's sum

$$\dots + [\hat{I}_p \quad -\hat{V}_p] \begin{bmatrix} V_p \\ I_p \end{bmatrix} + [\hat{I}_q \quad \hat{V}_q] \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + \dots = 0$$

Differentiating

$$\dots + [\hat{I}_p \quad -\hat{V}_p] \frac{\partial}{\partial \phi} \begin{bmatrix} V_p \\ I_p \end{bmatrix} + [\hat{I}_q \quad \hat{V}_q] \frac{\partial}{\partial \phi} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + \dots = 0$$

The branch relation to be used is

$$\frac{\partial}{\partial \phi} \begin{bmatrix} V_p \\ I_p \end{bmatrix} = \frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + \begin{bmatrix} A & B \\ C & D \end{bmatrix} \frac{\partial}{\partial \phi} \begin{bmatrix} V_q \\ -I_q \end{bmatrix}$$

Then we have

$$\dots + [\hat{I}_p \hat{V}_p] \frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + ([\hat{I}_p \hat{V}_p] \begin{bmatrix} A & B \\ C & D \end{bmatrix} + [\hat{I}_q \hat{V}_q]) \frac{\partial}{\partial \phi} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + \dots = 0.$$

The adjoint element can be defined as

$$[\hat{I}_p \hat{V}_p] \begin{bmatrix} A & B \\ C & D \end{bmatrix} + [\hat{I}_q \hat{V}_q] = 0.$$

The sensitivity expression is

$$[\hat{I}_p \hat{V}_p] \frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} -$$

The parameters can be derived

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 3/2 & 1/2\phi \\ 5/2 & 3/2 \end{bmatrix}$$

Hence

$$\frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{2\phi^2} \\ 5/2 & 0 \end{bmatrix}$$

Finally

$$\begin{aligned} & [\hat{I}_p \hat{V}_p] \frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} \\ & = -\frac{5}{2} \hat{V}_p V_q + 1/(2\phi^2) \cdot \hat{I}_p I_q \end{aligned}$$

## Question 10

The updating formula for the Fletcher-Powell-Davidon method is defined by

$$H^0 = I$$

$$s^j = -H^j \nabla U^j, \quad j = 0, 1, 2, \dots$$

where

$$H^{j+1} = H^j + \frac{\Delta\phi^j \Delta\phi^{jT}}{\Delta\phi^{jT} g^j} - \frac{H^j g^j g^{jT} H^j}{g^{jT} H^j g^j}$$

$$\Delta\phi^j \triangleq \alpha^j s^j = \phi^{j+1} - \phi^j$$

$$g^j \triangleq \nabla U^{j+1} - \nabla U^j$$

- (a) What is  $H^j$  and what is its relationship with the Hessian matrix of a function  $U(\phi)$ ? How is  $\alpha^j$  computed in practice?
- (b) Apply the algorithm (using a theoretically justified approach to obtain  $\alpha^j$ ) to the minimization of

$$2\phi_1^2 + 3\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 2$$

w.r.t.  $\phi_1$  and  $\phi_2$  starting at  $\phi_1 = 1, \phi_2 = 1$ . Show all steps explicitly and comment on the results obtained. Draw an accurate diagram showing the path taken.

**Answer (45 minutes)**

(a).  $H^j$  is an approximation to the inverse of the Hessian matrix of  $U$ . At the optimum,  $H^j$  equals the inverse of Hessian for a quadratic objective function.

$\alpha^j$  is computed using one dimensional search method where  $U(\phi^j + \alpha^j \underline{s}^j)$  is the objective function and  $\alpha^j$  is the variable.

$$(b) U = 2\phi_1^2 + 3\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 2$$

$$\underline{\phi}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Page 11A

$$\underline{\nabla} U = \begin{bmatrix} 4\phi_1 + \phi_2 + 2 \\ 6\phi_2 + \phi_1 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} \underline{\phi} + \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

$$(1) j=0, \underline{\phi}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \underline{\nabla} U^0 = \begin{bmatrix} 7 \\ 7 \end{bmatrix}, \underline{H}^0 = \underline{1}$$

$$\underline{\xi}^0 = -\underline{H}^0 \underline{\nabla} U^0 = -\begin{bmatrix} 7 \\ 7 \end{bmatrix}$$

$$\underline{\phi}^1 = \underline{\phi}^0 + \alpha^0 \underline{\xi}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \alpha^0 \begin{bmatrix} -7 \\ -7 \end{bmatrix} = \begin{bmatrix} 1 - 7\alpha^0 \\ 1 - 7\alpha^0 \end{bmatrix}$$

$$\therefore \frac{\partial U}{\partial \alpha} = \left( \frac{\partial U}{\partial \underline{\phi}} \right)^T \left( \frac{\partial \underline{\phi}}{\partial \alpha} \right) = (\underline{\nabla} U)^T \underline{\xi}$$

$$\therefore \left( \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} \begin{bmatrix} 1 - 7\alpha^0 \\ 1 - 7\alpha^0 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right)^T \begin{bmatrix} -7 \\ -7 \end{bmatrix} = \begin{bmatrix} 5(1 - 7\alpha^0) + 2 \\ 7(1 - 7\alpha^0) \end{bmatrix}^T \begin{bmatrix} -7 \\ -7 \end{bmatrix} = 0$$

$$12(1 - 7\alpha^0) + 2 = 0, \therefore 84\alpha^0 = 14 \quad \alpha^0 = \frac{14}{84} = \frac{1}{6} = 0.16667$$

$$(2) j=1, \underline{\phi}^1 = \underline{\phi}^0 + \alpha^0 \underline{\xi}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \frac{1}{6} \begin{bmatrix} -7 \\ -7 \end{bmatrix} = \begin{bmatrix} -0.16667 \\ -0.16667 \end{bmatrix}$$

$$\begin{cases} \Delta \underline{\phi}^0 = \underline{\phi}^1 - \underline{\phi}^0 = \begin{bmatrix} -1.16667 \\ -1.16667 \end{bmatrix}, \underline{\nabla} U^1 = \begin{bmatrix} 1.16667 \\ -1.16667 \end{bmatrix} \\ \underline{g}^0 = \underline{\nabla} U^1 - \underline{\nabla} U^0 = \begin{bmatrix} -5.83333 \\ -8.16667 \end{bmatrix} \end{cases}$$

$$\underline{H}^1 = \underline{H}^0 + \frac{\Delta \underline{\phi}^0 \Delta \underline{\phi}^{0T}}{\Delta \underline{\phi}^{0T} \underline{g}^0} - \frac{\underline{H}^0 \underline{g}^0 \underline{g}^{0T} \underline{H}^0}{\underline{g}^{0T} \underline{H}^0 \underline{g}^0}$$

$$= \underline{1} + \begin{bmatrix} 1.3612 & 1.3612 \\ 1.3612 & 1.3612 \end{bmatrix} / 16.334 - \begin{bmatrix} 34.627 & 47.6397 \\ 47.639 & 66.695 \end{bmatrix} / 100.72$$

$$= \begin{bmatrix} 0.74550 & -0.38965 \\ -0.38965 & 0.42115 \end{bmatrix}$$

$$\underline{\xi}^1 = -\underline{H}^1 \underline{\nabla} U^1 = \begin{bmatrix} -1.3244 \\ 0.94596 \end{bmatrix}$$

$$(\nabla U)^T \underline{\xi}' = \frac{\partial U}{\partial \alpha'} = 0$$

$$\left( \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} (\underline{\phi}' + \alpha' \underline{\xi}') + \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right)^T \underline{\xi}' = 0$$

$$\underline{\phi}'^T \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} \underline{\xi}' + \underline{\xi}'^T \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} \underline{\xi}' \alpha' + \begin{bmatrix} 2 \\ 0 \end{bmatrix}^T \underline{\xi}' = 0$$

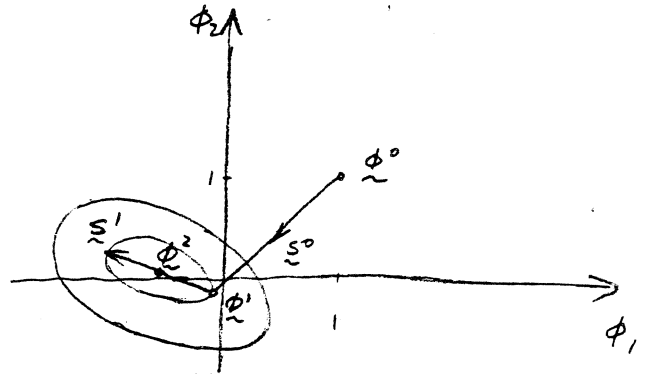
$$\alpha' = \frac{-\underline{\phi}'^T \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} \underline{\xi}' - \begin{bmatrix} 2 \\ 0 \end{bmatrix}^T \underline{\xi}'}{\underline{\xi}'^T \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} \underline{\xi}'} = \frac{0.00005 + 2.6488}{9.8795} = 0.26811$$

③  $j=2$ .

$$\underline{\phi}^2 = \underline{\phi}' + \alpha' \underline{\xi}' = \begin{bmatrix} -0.52176 \\ 0.086957 \end{bmatrix}$$

check:  $\nabla U|_{\underline{\phi}=\underline{\phi}^2} = \underline{0}$ .

Comment: Results obtained in 2 iterations.





## INDEX

<u>Section</u>		<u>No. of Pages</u>
1	Introduction to Simulation and Optimization	13
2	Collected Problems in Computational Methods, Design and Optimization	88
3	Notes on Vectors, Matrices and Sensitivities	40
4	Examples and Problems	34
	Simple Analysis and Optimization of an LC Transformer	9
	One at a Time Optimization of an Active Filter	12
	Response Calculation and Contour Plotting	13
5	Gauss Elimination	17
6	Examples and Problems	14
	Efficient Computation of Circuit Functions	4
	A Solution to Problem 18 of Section Two	10
7	Iterative Methods	5
8	Examples and Problems	9
	Relaxation Method of Solving Equations I	3
	Relaxation Method of Solving Equations II	6
9	Nonlinear System Simulation	5
	Gradient Concepts	1
	Nonlinear System Simulation	4
10	Examples and Problems	42
	Simple Resistor Diode Circuit	7
	Analysis and Sensitivity Evaluation for Nonlinear Systems	4
	Numerical Example of Analysis and Sensitivity Evaluation for Nonlinear Systems	4
	Resistor Diode Circuit Simulation Including Sensitivity Analysis	14
	Resistor Diode Circuit Simulation, Including Companion Network	7

	Nonlinear Circuit Simulation and Sensitivity Analysis	6
11	Linear System Simulation	7
12	Examples and Problems	23
	A Solution to Question 110 of Section Two I	5
	A Solution to Question 110 of Section Two II	4
	A Solution to Question 123 of Section Two	2
	A Solution to Question 130 of Section Two	3
	A Solution to Question 113 of Section Two	1
	A Solution to Question 114 of Section Two	1
	A Solution to Question 133 of Section Two	3
	A Solution to Question 150 of Section Two	2
	A Solution to Question 128 of Section Two	2
13	Method of Lagrange Multipliers	2
14	A Comprehensive Example of Minimax Optimization, the Adjoint Method and Sparse Matrix Manipulations	8
15	Numerical Examples of Optimization Methods	9
16	One-Dimensional Strategies	19
17	Direct Search	18
18	Examples and Problems	20
	Selected Problems and Their Solution	20
19	Solution of the State Equations	7
20	Examples and Problems	9
	Implementation of Fourth Order Runge-Kutta Algorithm	9
21	Computer-Aided Circuit Optimization	31
22	Circuit Optimization: the State of the Art	20
23	KMOS - A Fortran Library for Nonlinear Optimization	39
24	Past Exams, Tests and Solutions	49

