McMASTER UNIVERSITY BOOKSTORE

CUSTOM

COURSEWARE

S E R V I C E

COMPUTER ENGINEERING 3KB3

Optimization, Vol. 2

John W. Bandler

Fall 1994

All Custom CourseWare is Non-Returnable

Computer Engineering 3KB3

Optimization, Vol. 2

John W. Bandler

December 1993

Chapter 2

COLLECTED PROBLEMS

2.1 INTRODUCTION

Over the years, optimization techniques have proved to become useful in many more real world applications. Because of this, many problems have accumulated and are still accumulating. With new CAE software, many techniques have also been simplified, or rather problems are now allowed to become more complex as optimization packages take care of tedious tasks. This chapter attempts to cover all the topics associated with optimization including those which make use of available optimization software.

Chapter 2 is divided into the following explanatory sections:

2.2 PROBLEMS

2.2.1 Algorithm Development

Question 1001 Develop an algorithm to efficiently calculate the value of

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

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.

HAND☞ Question 1002 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 $\theta = \pi/2$. 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 1003 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 1004 State Horner's rule for polynomial evaluation. Explain its advantages compared with the direct method of evaluating a polynomial.

Question 1005 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 θ. 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 1006 Write an efficient algorithm for converting binary numbers to decimal numbers. Test it on the numbers 1101, 10111 and 1010101.

Question 1007 Write and test on 44 an efficient algorithm for converting decimal numbers to binary numbers.

Question 1008 Write an algorithm to efficiently evaluate ∇F and $\partial F/\partial s$ where

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

and

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

Question 1009 Write an algorithm to efficiently calculate the value of the objective function

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

and the gradient vector $\nabla U(\phi)$ a total of m times for different ϕ, 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),

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

is the approximating function of time (model response),

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

Question 1010 Write an algorithm to efficiently calculate the frequency response $V_2(j\omega)/V_1(j\omega)$ for the circuit of Fig. Q1010. Use the algorithm to calculate the response when $L_1 = L_2 = 2$ H, $C_1 = C_2 = 0.5$ F, and $\omega = 2$ rad/s. (See Question 1011.)

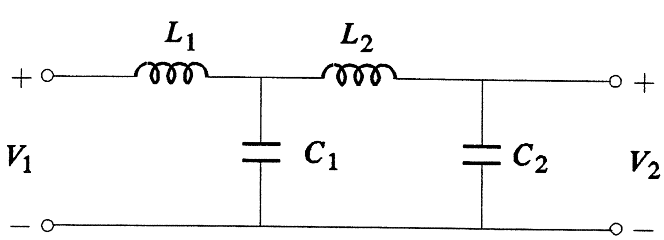

Fig. Q1010 LC ladder network.

OSA☞ Question 1011 Use OSA90/hope to calculate the frequency response $V_2(j\omega)/V_1(j\omega)$ for the circuit of Fig. Q1010. The values of the elements are $L_1 = L_2 = 2$ H, $C_1 = C_2 = 0.5$ F. Calculate $V_2(j\omega)/V_1(j\omega)$ for $\omega = 1.7$-3.2 rad/s with an incremental step of 0.1 rad/s. Display your results numerically and graphically. (See Question 1010.)

Question 1012 Write an algorithm to efficiently evaluate ∇F where

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

and $\phi = [a_0\ a_1\ \ldots\ a_n\ b_0\ b_1\ \ldots\ b_m]^T$.

Question 1013 Write an algorithm to efficiently evaluate ∇T where

$$T(\phi, s) = \frac{1}{R_1R_2C_1C_2s^2 + (R_1C_1 + R_1C_2 + R_2C_2)s + 1}$$

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. Q1013.

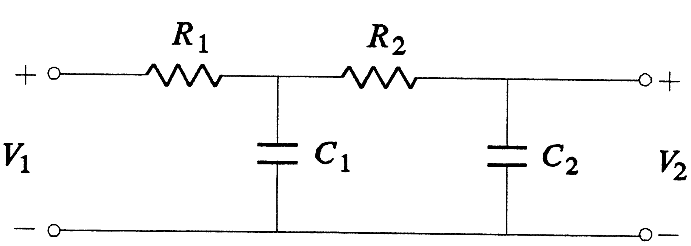

Fig. Q1013 RC ladder network.

Question 1014 Show how the errors propagate in the calculation of

(a) $\dfrac{a}{b - cd}$

(b) $\dfrac{a}{b(c - d)}$

(c) $\dfrac{xy}{u - v}$

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

Question 1015 Derive an expression for the relative error in the computation of x/y. Neglect terms involving products of errors.

Question 1016 Calculate and state the maximum number of multiplications and divisions in the efficient solution for $\boldsymbol{x}$ of the linear system

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

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

HAND☞ Question 1017 Derive from first principles an efficient algorithm for solving the tridiagonal system of equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{d}$$

for $\boldsymbol{x}$, giving arbitrary vector $\boldsymbol{d}$, where

$$\boldsymbol{A} = \begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{bmatrix}$$

using the one-dimensional arrays $a_1, a_2, ..., a_n, b_1, b_2, ..., b_n, c_1, c_2, ..., c_{n-1}$, explicitly.

Question 1018 Write an efficient Fortran program to calculate all the branch voltages and currents in the resistive ladder network of Fig. Q1018, allowing up to 100 resistors. Essential data: $V_g, R_1, R_2, \ldots, R_n$.

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 $V_g = 1$ V using the program written. (See Question 1019.)

OSA☞ Question 1019 Use OSA90/hope to calculate all the branch voltages and currents in the resistive ladder network of Fig. Q1018 with $n = 8$, $R_1 = R_3 = R_5 = R_7 = 3\ \Omega$, $R_2 = R_4 = R_6 = R_8 = 1\ \Omega$. Calculate the voltages and currents for $V_g = 0, 0.5, 1, 1.5, 2, 2.5$ and 3 V. (See Question 1018.)

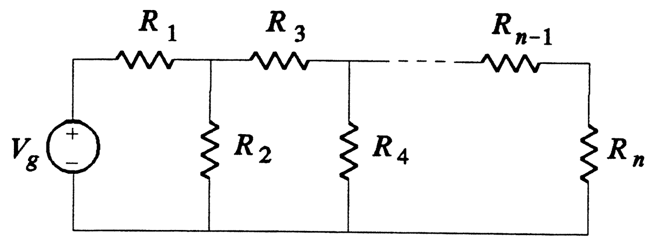

Fig. Q1018 Resistive ladder network.

Question 1020 Write a program to calculate the input resistance of the circuit of Question 1018. Use the program written to calculate the input resistance for the numerical example in Question 1018. (See Question 1021.)

OSA☞ Question 1021 Use OSA90/hope to calculate the input resistance of the circuit of Question 1018. Calculate the input resistances for R_1 = 1, 2, 3, 4 and 5 Ω. Let $n = 8$, $R_3 = R_5 = R_7 = 3$ Ω and $R_2 = R_4 = R_6 = R_8 = 1$ Ω. (See Question 1020.)

Question 1022 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. Q1022, 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, . . ., R_n.

Let $n = 7$, $R_2 = R_4 = R_6 = 1/3$ Ω, $R_1 = R_3 = R_5 = R_7 = 1$ Ω. Calculate the voltages and currents for $V_g = 1$ V using the program. (See Question 1024.)

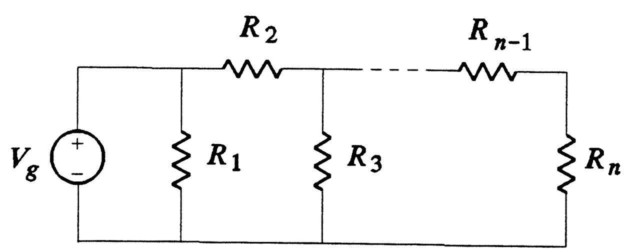

Fig. Q1022 Resistive ladder network.

HAND☞ Question 1023 Consider Question 1022. Evaluate

$$\frac{\partial I_3}{\partial R_i}, \quad i = 1, 2, ..., 7$$

where I_3 is the current flowing in resistor R_3.

OSA☞ Question 1024 Use the LU factorization capability of OSA90/hope to calculate and print out all the branch voltages and currents of the resistive ladder network of Fig. Q1022 with $n = 7$,

$R_2 = R_4 = R_6 = 1/3\ \Omega$, $R_1 = R_3 = R_5 = R_7 = 1\ \Omega$. Calculate the voltages and currents for V_g = 0-3 V with an incremental step of 1 V. (See Question 1022.)

Question 1025 Write a program to calculate the input conductance of the circuit of Question 1022. Use the program written to calculate the input conductance for the numerical example in Question 1022. (See Question 1026.)

OSA☞ Question 1026 Use OSA90/hope to calculate the input conductance of the circuit of Question 1024. Calculate the input conductance for the numerical example in Question 1024. Show the input conductances for R_1 = 1-10 Ω with an incremental step of 1 Ω. (See Question 1025.)

Question 1027 Consider the ladder network of Fig. Q1027.

(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?

(See Question 1028.)

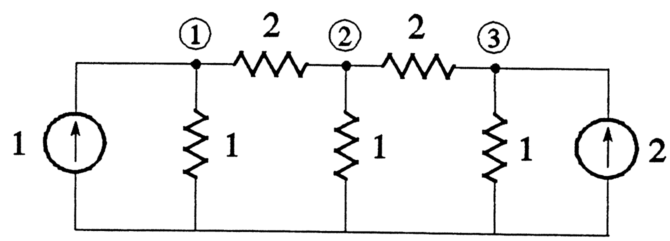

Fig. Q1027 Three–node resistive ladder network.

OSA☞ Question 1028 Consider the ladder network of Fig. Q1027. Use OSA90/hope to calculate the node voltages by (i) matrix inversion and (ii) LU factorization. Set the right-hand source to zero and recalculate the node voltages. *In general*, what would the computational effort be for different excitations? (See Question 1027.)

Question 1029 Is the inverse of a tridiagonal matrix (in general) sparse, dense or tridiagonal? Justify your answer by a physically meaningful example.

Question 1030 Define the term "relaxation method".

Question 1031 State the Gauss-Seidel iterative formula for the solution of the linear system $\boldsymbol{A}\ \boldsymbol{x} = \boldsymbol{b}$, defining precisely any new symbols introduced.

Question 1032 Factorize the following matrix into LU form utilizing available storage locations as much as possible. (See Question 1033.)

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

OSA☞ Question 1033 Use OSA90/hope to factorize the following matrix into LU form. (See Question 1032.)

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

Question 1034 Consider the resistive network shown in Fig. Q1034. Take $G_1 = 2$ S and $G_3 = 1$ S. Showing clearly all major steps, calculate the node voltages by LU factorization.

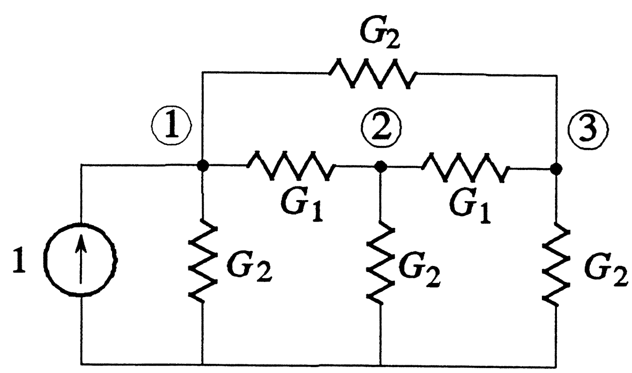

Fig. Q1034 Three–node resistive network.

Question 1035 Apply the Gauss-Seidel (relaxation) method to the circuit of Question 1034. Take the initial node voltages as zero and use two iterations. Repeat with an overrelaxation factor of 1.5.

Question 1036 Consider the resistive network shown in Fig. Q1036. Take $G_1 = G_3 = G_5 = 1$ S and $R_2 = R_4 = 0.5\ \Omega$. 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.

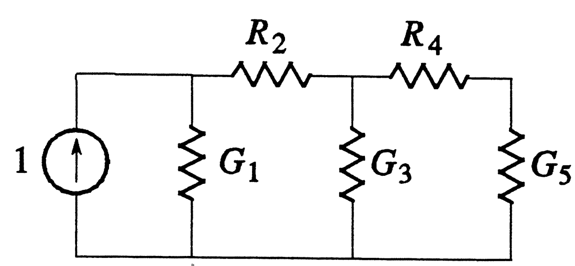

Fig. Q1036 Resistive ladder network.

Question 1037 Consider the resistive network shown in Fig. Q1037. Let $G_1 = 1$ S and $G_2 = 2$ S. Showing clearly all major steps, apply two iterations of the Gauss-Seidel relaxation method starting with $V_1 = 1$ V, $V_2 = 0.5$ V, $V_3 = 0$ V. 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.

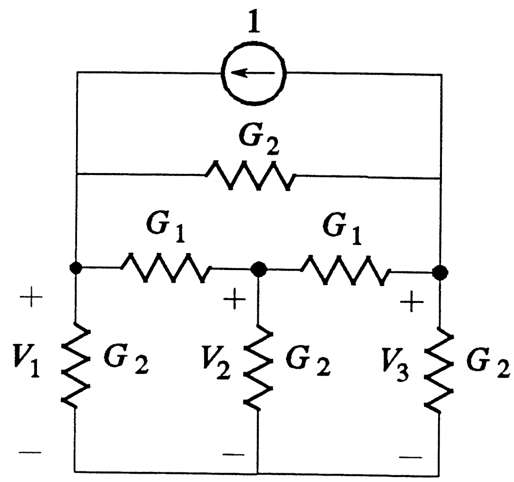

Fig. Q1037 Resistive network.

Question 1038 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 error 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 in Fig. Q1037, starting with $V_1 = 1.0$ V, $V_2 = 0.5$ V, $V_3 = 0$ V. Assume that the solution has been found if the norm of the errors is less than 10^{-4}.

Question 1039 Consider the linear circuit shown in Fig. Q1039, 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 $R_1 = R_2 = R_3 = 2\ \Omega$, $C_1 = C_2 = C_3 = 1$ F. 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.

(See Question 1040.)

OSA☞ Question 1040 Consider the linear circuit shown in Fig. Q1039, which is operating in the sinusoidal steady state. Use OSA90/hope to find V_3/V_1 for this circuit at $\omega = 0\text{–}10$ rad/s with an incremental step of 1 rad/s by (i) matrix inversion, (ii) LU factorization and (iii) direct circuit simulation. Take $R_1 = R_2 = R_3 = 2\ \Omega$, $C_1 = C_2 = C_3 = 1$ F. (See Question 1039.)

Question 1041 Consider the linear circuit shown in Fig. Q1041, which is operating in the sinusoidal steady state. Find V_3/V_1 for this circuit at $\omega = 1$ rad/s in the following ways. Take $R_1 = R_2 = R_3 = 1\ \Omega$, $C_1 = C_2 = C_3 = 2$ F. 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.

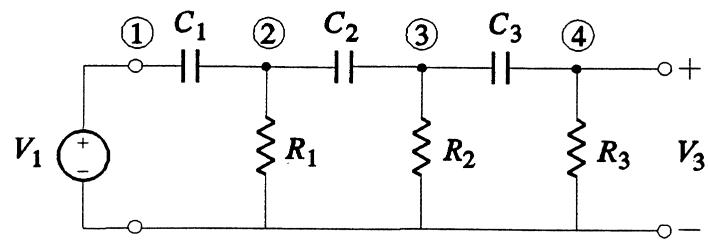

Fig. Q1039 CR ladder network.

(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 backwards through the ladder.
(See Question 1042.)

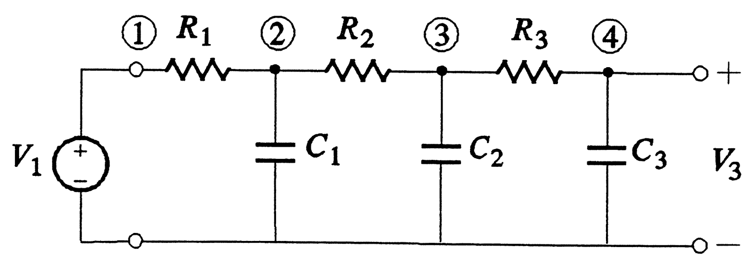

Fig. Q1041 RC ladder network.

OSA☞ Question 1042 Consider the linear circuit shown in Fig. Q1041, which is operating in the sinusoidal steady state. Use OSA90/hope to find V_3/V_1 for this circuit at ω = 0–5 rad/s with an incremental step of 0.2 rad/s by (i) matrix inversion, (ii) LU factorization and (iii) direct circuit simulation. Take $R_1 = R_2 = R_3 = 1\ \Omega$, $C_1 = C_2 = C_3 = 2$ F. (See Question 1041.)

Question 1043 Apply the Gauss-Seidel (relaxation) method to the circuit of Question 1041. Take the initial node voltages to be zero and use two iterations. Repeat over with an overrelaxation factor of 1.5.

Question 1044 Calculate and plot the reflection coefficient of the circuit shown in Fig. Q1044, where $C_1 = 1.0$ F, $C_2 = 0.125$ F, $L = 2.0$ H, $0 \le \omega \le 4$ rad/s. (See Question 1045.)

OSA☞ Question 1045 Use OSA90/hope to calculate and plot the reflection coefficient of the circuit shown in Fig. Q1044, where $C_1 = 1.0$ F, $C_2 = 0.125$ F, $L = 2.0$ H, $0 \le \omega \le 4$ rad/s. (See Question 1044.)

Question 1046 Consider the iterative scheme

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

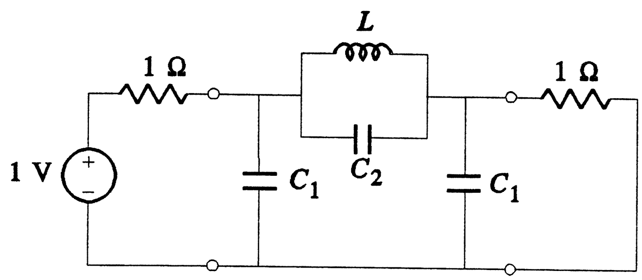

Fig. Q1044 LC filter network.

where the $\boldsymbol{y}$ vectors are of dimension 2 and the $\boldsymbol{A}$ matrices are 2×2 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 $\boldsymbol{y}^1$. (See Question 1044.)

Question 1047 Consider the iterative scheme described in Question 1046. Give the terminating condition

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

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

$$c^n = \frac{y_1^n}{y_2^n}$$

Question 1048 Assume that each matrix $\boldsymbol{A}^i$ in Question 1046 is a function of a single variable x_i. Derive from first principles an approach to calculating $\partial y_1^1/\partial \boldsymbol{x}$ where $\boldsymbol{x}$ is a column vector containing the x_i, $i = 1, 2, \ldots, n$.

Question 1049 Consider the system described by the iterative schemes

$$\boldsymbol{y}^{i+1} = \boldsymbol{A}^i\, \boldsymbol{y}^i, \quad i = 1, 2, \ldots, n, \qquad i \neq j$$
$$\boldsymbol{z}^{i+1} = \boldsymbol{B}^i\, \boldsymbol{z}^i, \quad i = 1, 2, \ldots, m$$

the equation

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

and the terminating conditions

$$z_1^1 = z_2^1$$
$$y_1^1 = y_2^1$$
$$y_1^{n+1} = 1$$

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

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

2.2.2 Minimization

Question 2001 Use the multidimensional Taylor series expansion to show that a turning point of a convex differentiable function is a global minimum. Justify all assumptions.

HAND☞ Question 2002 Given a differentiable function f of many variables $\mathbf{x}$ and a corresponding direction vector $\mathbf{s}$,

$$\lim_{\lambda \to 0^+} \frac{f(\mathbf{x} + \lambda \mathbf{s}) - f(\mathbf{x})}{\lambda} = \text{..........} \text{ (please state) ?}$$

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

Question 2003 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 measured in the Euclidean sense, in the direction of the gradient vector w.r.t. the variables.

Question 2004 Use the method of Lagrange multipliers to minimize w.r.t. ϕ_1 and ϕ_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. ϕ_1 and ϕ_2. Verify your answer by substituting the constraint into the function.

HAND☞ Question 2005 Use the method of Lagrange multipliers to minimize w.r.t. ϕ_1 and ϕ_2 the function

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

subject to

$$\phi_1 + \phi_2 = 1$$

Sketch a diagram to illustrate the problem and its solution w.r.t. ϕ_1 and ϕ_2. Verify your answer by substituting the constraint into the function.

Question 2006 If $\mathbf{g}(\boldsymbol{\phi})$ is concave, verify that $\mathbf{g}(\boldsymbol{\phi}) \geq \mathbf{0}$ describes a convex feasible region.

Question 2007 Under what conditions could equality constraints be included in convex programming?

Question 2008 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 $0 \le \phi \le 3$.

(c) The minimum of $a\phi^2 + b$ contrasted with the minimum of $a\phi^2 + b$ subject to $\phi \ge 0$, 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 2009 Find suitable transformations for the following constraints so that we can use an unconstrained optimization algorithm.

(a) $0 \le \phi_1 \le \phi_2 \le \ldots \le \phi_i \le \ldots \le \phi_k$

(b) $0 < l \le \phi_2/\phi_1 \le u$, $\phi_1 > 0$, $\phi_2 > 0$

Question 2010 Write the following constraints in the form $g_i(\boldsymbol{\phi}) \ge 0$, $i = 1, 2, \ldots, m$.

(a) $l_i \le \phi_i \le u_i$, $i = 1, 2, \ldots, k$

(b) $a \le \phi_i/\phi_{i+1} \le b$, $i = 1, 2, \ldots, k-1$

(c) $1 \le \phi_1 \le \phi_2 \le \ldots \le \phi_k \le 3$

(d) $h_i(\boldsymbol{\phi}) = 0$, $i = 1, 2, \ldots, s$

Question 2011 Discuss the scaling effects of the transformation $\phi_i = \exp \phi'_i$.

Question 2012 Given the derivatives of functions w.r.t. ϕ_i express the gradient vector w.r.t. the ϕ'_i variables and the corresponding incremental change vector for the transformation $\phi_i = \exp \phi'_i$.

HAND☞ Question 2013 Consider the parameter constraints

$$0 \le \phi_1 \le \phi_2 \le \ldots \le \phi_i \le \ldots \le \phi_k$$

as the only constraints applicable in a minimization problem.

(a) Find a suitable transformation of the variables $\phi_1, \phi_2, \ldots, \phi_k$ so that we can use an unconstrained optimization package.

(b) Assuming the new variables are $z_1, z_2, \ldots$, write down $\partial U/\partial z_1, \partial U/\partial z_2, \ldots$ given $\partial U/\partial \phi_1$, $\partial U/\partial \phi_2, \ldots$

(c) You have access to a subprogram to calculate U and ∇U given $\boldsymbol{\phi}$ but you can not alter it. How would you organize your software and data to handle the transformed problem?

Question 2014 Use an appropriate transformation of variables to create the minimization of an unconstrained objective function for the problems

(a) $\underset{\phi}{\text{minimize}}\ U = b\phi + c$ subject to $\phi \ge 0$ with $b > 0$

(b) $\underset{\phi}{\text{minimize}}\ U = a_1\phi_1^2 + a_2\phi_2^2$ subject to $1 \le \phi_i \le 2$, $i = 1, 2$ with $a_1, a_2 > 0$

HAND☞ Question 2015 Derive the gradient vector of $U(\boldsymbol{\phi})$ w.r.t. $\boldsymbol{\phi}$ for the objective functions

$$U = \int_{\psi_l}^{\psi_u} |e(\boldsymbol{\phi},\psi)|^p \, d\psi$$

and

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

where the appropriate error functions are complex.

Question 2016 Prove that $d|z|^2/dx = 2\ Re\{z^*\ dz/dx\}$, where $z(x)$ is complex and x is real.

Question 2017 For the linear function (a polynomial is a special case)

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

formulate the discrete minimax approximation of $S(\psi)$ by $F(\phi, \psi)$ as a linear programming problem w.r.t. ϕ, assuming ϕ to be unconstrained.

Question 2018 For the linear function (a polynomial is a special case)

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

assuming an objective function of the form of

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

derive the gradient vector of U and the Hessian matrix w.r.t. ϕ.

Question 2019 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 DC 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?

HAND☞ Question 2020 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.

Question 2021 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 2022 Derive the exact Newton iteration at ϕ^j for minimization of a differentiable multidimensional function $U(\phi)$. Define all terms used. Under what conditions do you expect a locally downhill step from Newton iteration? Discuss possible pitfalls of the basic Newton method and describe a "damped" Newton method. Illustrate your answers with sketches.

Question 2023 Derive carefully from first principles a numerical approach to finding the gradient vector $\partial f/\partial \boldsymbol{x}$ subject to the system of equations $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}$ given values for $\boldsymbol{x}$, where $f \equiv f(\boldsymbol{y}(\boldsymbol{x}), \boldsymbol{x})$ is a scalar function and where the vector $\boldsymbol{h}$ is nonlinear both in $\boldsymbol{x}$ and $\boldsymbol{y}$. Assume that $\boldsymbol{h}$ and $\boldsymbol{y}$ have the same dimensions and that the Jacobian of $\boldsymbol{h}$ w.r.t. $\boldsymbol{y}$ 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 2024 Define the term "positive definite" as it relates to a square symmetric matrix.

Question 2025 Provide and discuss a link between the Hessian matrix of a differentiable function $U(\boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is a k-vector, with the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{\phi})$, where $\boldsymbol{f}$ is a k-vector of functions of $\boldsymbol{\phi}$.

Question 2026 For the resistor-diode network shown in Fig. Q2026, illustrate with the aid of an I-V diagram an iterative method of finding V at DC. State Newton's method for solving this problem and derive the network model corresponding to the situation at the jth iteration. What is the significance of this model?

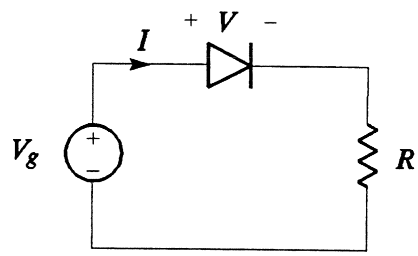

Fig. Q2026 Resistor–diode network.

Question 2027 For the resistor-diode network shown in Fig. Q2027, find the operating point using (a) techniques for solving nonlinear equations, (b) optimization techniques, with E = 10 V, R = 1 kΩ, $I_s = 10^{-2}$ mA, $\lambda = 38.7$ V^{-1} and $I_d = I_s(\exp(\lambda V_d) - 1)$.

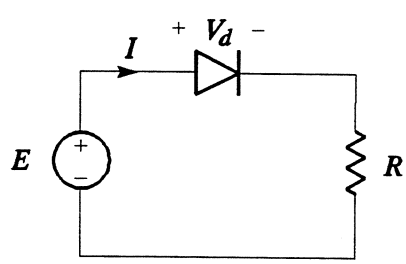

Fig. Q2027 Resistor–diode network.

HAND☞ Question 2028 We wish to calculate $\partial f/\partial \boldsymbol{x}$ subject to $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}$ where $f \equiv f(\boldsymbol{y}(\boldsymbol{x}), \boldsymbol{x})$ given values for $\boldsymbol{x}$.

Explain fully the formula

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

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

$$\left[\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}}\right]\hat{\boldsymbol{y}} = \frac{\partial f}{\partial \boldsymbol{y}}$$

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

$$4x_1^2y_1^2 - 3y_2 - 2 = 0$$
$$-x_1y_1 + 2x_2^2\, y_1y_2 - 3 = 0$$
$$f = y_1^2 + y_2^2 + 2x_2$$

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

Question 2029 Let

$$h_1 = 4\, x_1\, y_1 - 3\, y_2 = 0$$
$$h_2 = -x_1\, y_1\, y_2 + 2\, x_2^2\, y_2 - 3 = 0$$
$$f = y_1^2 + y_1\, x_1$$

Working directly with these functions, construct the Newton equations to solve the nonlinear system and construct the formulas to calculate $\partial f/\partial \boldsymbol{x}$ subject to the given constraints. Let $\boldsymbol{x} = [1 \;\; 1.25]^T$, where T denotes transposition. Apply one iteration of the Newton method starting at $\boldsymbol{y}^0 = [1 \;\; 1]^T$. Assuming the solution to be $\boldsymbol{y} = [1.125 \;\; 1.5]^T$ calculate the appropriate $\partial f/\partial \boldsymbol{x}$. (See Question 2031.)

Question 2030 Let

$$h_1 = -\, x_1y_1y_2 + 2x_2^2y_2 - 3 = 0$$
$$h_2 = 4x_1y_1 - 3y_2 = 0$$
$$f = y_2$$

Working directly with these functions, construct the Newton equations to solve the nonlinear system and construct the formulas to calculate $\partial f/\partial \boldsymbol{x}$ subject to the given constraints. Let $\boldsymbol{x} = [1 \;\; 1.25]^T$, where T denotes transposition. Apply one iteration of the Newton method starting at $\boldsymbol{y}^0 = [1 \;\; 1]^T$. Assuming the solution to be $\boldsymbol{y} = [1.125 \;\; 1.5]^T$ calculate the appropriate $\partial f/\partial \boldsymbol{x}$. (See Question 2032.)

OSA☞ Question 2031 Use OSA90/hope and the functions given in Question 2029 to verify that the solution is $\boldsymbol{y}^0 = [1.125 \;\; 1.5]^T$ starting at $\boldsymbol{y}^0 = [1 \;\; 1]^T$. Working directly with these functions, construct the Newton equations to solve the nonlinear system and construct the formulas to calculate $\partial f/\partial \boldsymbol{x}$ subject to the given constraints. Let $\boldsymbol{x} = [1 \;\; 1.25]^T$, where T denotes transposition. Check derivatives by perturbation. (See Question 2029.)

OSA☞ Question 2032 Use OSA90/hope and the functions given in Question 2030 to verify that the solution is $y^0 = [1.125 \quad 1.5]^T$ starting at $y^0 = [1 \quad 1]^T$. Working directly with these functions, construct the Newton equations to solve the nonlinear system and construct the formulas to calculate $\partial f/\partial x$ subject to the given constraints. Let $x = [1 \quad 1.25]^T$, where T denotes transposition. Check derivatives by perturbation. (See Question 2030.)

Question 2033 Write down and define the first three terms of the multidimensional Taylor series expansion of a scalar function U of many variables ϕ, defining any expressions used appropriately.

Question 2034 Write down the first three terms of the multidimensional Taylor series expansion at ϕ^j of a scalar function U of many variables ϕ. Define all terms used. Write down a linear approximation of the function at ϕ^j. Write down the quadratic approximation of the function at ϕ^j.

Question 2035 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 2036 Derive a formula to approximately calculate all first partial derivatives of a function of k variables by perturbation, using $2k$ function evaluations.

Question 2037 What are the implications of a positive-semidefinite Hessian matrix in minimization problems?

Question 2038 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 2039 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 2040 Write down an objective function which can be minimized in an effort to solve the system of nonlinear equations $f = 0$. Differentiate it w.r.t. the variables and express the gradient vector in compact form.

Question 2041 What is the implication of a negative first-order term in the multidimensional Taylor expansion of a differential function of many variables? Sketch your answer w.r.t. a function of two variables.

Question 2042 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 2043 Write a simple Fortran program to implement steepest descent in the minimization of a scalar differentiable function of many variables and test in on suitable examples.

Question 2044 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 2045 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 2046 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 2047 Describe pitfalls in attempting the solution of constrained optimization problems using the algorithms of Question 2046.

Question 2048 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 2049 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 2050 Sketch the 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 2051

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

Question 2052 Discuss the necessary conditions for an *unconstrained* optimum of a differential function. Derive them from

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

Question 2053 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 2054 Consider the function

$$U = 0.5\,\phi_1 + 2\phi_2 + 3$$

along with the constraints

$$g_1 = \phi_1 + 2\phi_2 - 3 \geq 0$$
$$g_2 = 2\phi_1 + \phi_2 - 3 \geq 0$$

at each of the three points (i) $[2\ 1]^T$; (ii) $[1\ 1]^T$; (iii) $[3\ 0]^T$. Invoke the Kuhn-Tucker (KT) conditions at each of the three points. State the results and comment on them.

Question 2055 Consider each of the following objective functions

(a) $U = 2\phi_1 + 2\phi_2 - 1$
(b) $U = 0.5\phi_1 + 2\phi_2 + 3$
(c) $U = 2\phi_1 + 0.5\phi_2 + 1.5$

along with the constraints

$$g_1 = \phi_1 + 2\phi_2 - 3 \geq 0$$
$$g_2 = 2\phi_1 + \phi_2 - 3 \geq 0$$

at each of the three points (i) $[2\ 1]^T$ (ii) $[1\ 1]^T$ (iii) $[3\ 0]^T$. Invoke the Kuhn-Tucker (KT) conditions for each objective function at each of the three points. State the results and comment on them. For any case which does not satisfy the KT conditions, find the steepest feasible downhill direction.

Question 2056 Consider the voltage divider shown in Fig. Q2055. The design specification is $V \geq 60$ V and the constraint is $R_g \geq 50\ \Omega$. By testing the Kuhn-Tucker conditions find V_g and R_g such that the total power dissipated is minimum.

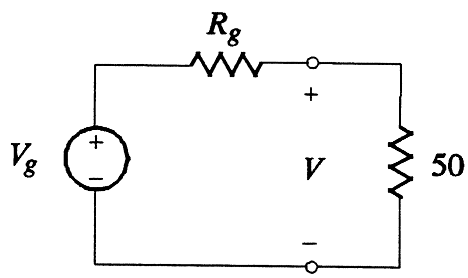

Fig. Q2056 A voltage divider.

Question 2057 Consider the voltage divider shown in Fig. Q2056 with the following objective and specifications:

Objective: minimize total power dissipation
Design specifications: $45 \leq V \leq 55$
Constraints: $50 \leq R_g \leq 60$

By testing the Kuhn-Tucker conditions aided graphically show which constraints are active. What is the solution? Sketch the feasible region and contours of the objective function identifying each function involved clearly.

Question 2058 Sketch curves of $|x - x^0|^p$ against x for $p = 0.5$, 1, 2, 4 and ∞. Discuss the differentiability and convexity of these curves.

Question 2059 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 ∞. Comment on the convexity of these regions and the corresponding one for $p = 0.5$.

Question 2060 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 2061 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 2062 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 2063 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 $r \to 0$. Draw a contour sketch to illustrate the process. Are the conditions for a constrained minimum satisfied?

HAND☞ Question 2064 The updating formula for the Fletcher-Powell-Davidon method is defined by

$$H^0 = 1$$

$$s^j = -H^j \nabla U^j, \qquad j = 0, 1, 2, \ldots$$

where

$$H^{j+1} = H^j + \frac{\Delta\phi^j \Delta\phi^{j^T}}{\Delta\phi^{j^T} g^j} - \frac{H^j g^j g^{j^T} H^j}{g^{j^T} H^j g^j}$$

$$\Delta\phi^j \triangleq \alpha^j s^j = \phi^{j+1} - \phi^j$$

$$g^j \triangleq \nabla U^{j+1} - \nabla U^j$$

(a) What is H^j and what is its relationship with the Hessian matrix of a function $U(\phi)$? How is α^j computed in practice?

(b) Apply the algorithm (using a theoretically justified approach to obtain α^j) to the minimization of

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

w.r.t. ϕ_1 and ϕ_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.

Question 2065 Apply the algorithm described in Question 2064 (using a theoretically justified approach to obtain α^j) to the minimization of

$$2\phi_1^2 + 3\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 2$$

w.r.t. ϕ_1 and ϕ_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.

Question 2066 Apply the Fletcher-Power-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. ϕ_1 and ϕ_2 starting at $\phi_1 = 0$, $\phi_2 = 0$, showing all steps explicitly and commenting on the results obtained. (See Question 2067.)

OSA☞ Question 2067 Use OSA90/hope to create a Fletcher-Powell-Davidon algorithm for minimizing the differentiable function given in Question 2066 and display the results of every iteration. (See Question 2066.)

Question 2068 Apply the Newton method to the minimization of

$$\phi_1^2 + 3\phi_2^2 + \phi_1\phi_2 + 2\phi_2 + 1$$

w.r.t. ϕ_1 and ϕ_2. Select the starting points (a) $\phi_1 = 0$, $\phi_2 = 0$, (b) $\phi_1 = 0$, $\phi_2 = 2$. What is the solution?

Question 2069 Apply 1 iteration of the steepest descent algorithm to the minimization of

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

w.r.t. ϕ_1 and ϕ_2 starting at $\phi_1 = 0$, $\phi_2 = 0$, using an exact analytical line search for the minimum in the direction of search used. Derive and justify all formulas employed.

HAND☞ Question 2070

(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 ∇U^j and ∇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 ϕ.

(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

$$\alpha^{j-1} s^{j-1} \triangleq \phi^{j} - \phi^{j-1} \tag{4}$$

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 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.

Question 2071 Suppose x and p are n-vectors. The conjugate gradient method for the optimization problem

$$\underset{x}{\text{minimize}} \quad U(x) \tag{1}$$

can be described as follows. Let the initial direction be

$$p^0 = -\nabla U^0 = -\frac{\partial U}{\partial x}\bigg|_{x=x^0} \tag{2}$$

Calculate a^k by a line search along the direction p^k using

$$x^{k+1} = x^k + a^k p^k \tag{3}$$

Let

$$\nabla U^{k+1} = \frac{\partial U}{\partial x}\bigg|_{x=x^{k+1}} \tag{4}$$

Take

$$c^k = \frac{(\nabla^T U^{k+1} \, \nabla U^{k+1})}{(\nabla^T U^k \, \nabla U^k)} \tag{5}$$

Update the direction using

$$p^{k+1} = -\nabla U^{k+1} + c^k p^k \tag{6}$$

(a) Consider

$$U(x) = \frac{1}{2}\,(Ax - b)^T A^{-1} (Ax - b) \tag{7}$$

where A is a $n{\times}n$ constant matrix and b is a constant n-vector. Suppose A is symmetrical and positive definite. Derive a formula for calculating a^k. [Result: $a^k = \nabla^T U^k / ((p^k)^T A\, p^k)$. Hint: $\nabla^T U^k\, p^{k-1} = 0$.]

(b) For the $U(x)$ defined in (7), we have

$$\nabla U^k = A\, x^k - b \tag{8}$$

and

$$\nabla U^{k+1} = \nabla U^k + a^k A\, p^k \tag{9}$$

Equations (2)-(6), (8)-(9) and the formula derived in (a) constitute the conjugate gradient method for solving

$$Ax = b \tag{10}$$

Use such a method to obtain numerical solutions to

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{11}$$

Take the initial point as

$$x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{12}$$

Show numerical values of x^k, ∇U^k, c^k, p^k and a^k as used in each iteration.

<u>Question 2072</u> 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. ϕ_1 and ϕ_2 starting at $\phi_1 = 0$, $\phi_2 = 0$, showing all steps explicitly and commenting on the results obtained. (See Question 2073.)

OSA☞ <u>Question 2073</u> Use OSA90/hope to create a conjugate gradient algorithm for minimizing the differentiable function given in Question 2072 and display the results of every iteration. (See Question 2072.)

<u>Question 2074</u> Apply the conjugate gradient algorithm for minimizing a differentiable function of many variables to the following data.

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

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

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

<u>Question 2075</u> Consider the linear programming problem

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

w.r.t. ϕ_1, ϕ_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 2076 Minimize w.r.t. ϕ

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

subject to

$$\phi_1 \geq 0,\ i = 1, 2, 3$$
$$20 - \phi_1 \geq 0,\ 11 - \phi_2 \geq 0,\ 42 - \phi_3 \geq 0$$
$$72 - \phi_1 - 2\phi_2 - 2\phi_3 \geq 0$$

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 2077 Sketch contours of the function

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

w.r.t. ϕ for $U = \phi_1^2 + 4\phi_2^2$ and $h = 2\phi_2 - 1$ in the vicinity of the solution stated in Question 2076 for $\alpha = 0.1$, 1.0 and 100, taking care to indicate points of discontinuous derivatives. [Source: Bandler and Charalambous (1974).]

Question 2078 Minimize w.r.t. ϕ

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

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 $\phi_1 = 20$, $\phi_2 = \phi_3 = 15$. This problem is referred to as the Post Office Parcel problem. [Source: Rosenbrook (1960). See also Bandler and Charalambous (1974).]

Question 2079 Minimize w.r.t. ϕ

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

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 (1956). See also Kowalik and Osborne (1968).]

Question 2080 Minimize w.r.t. $\boldsymbol{\phi}$

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

subject to

$$\phi_i \geq 0, \qquad i = 1, 2, 3$$
$$3 - \phi_1 - \phi_2 - 2\phi_3^2 \geq 0$$

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 2081 Minimize w.r.t. $\boldsymbol{\phi}$ 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 2082 Examine the points $[0\ \ 0]^T$, $[0\ \ 1]^T$ and $[1\ \ 1]^T$ for a 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. What are your conclusions?

Question 2083 Consider the three functions

$$f_1 = \phi_2 - 3$$
$$f_2 = -\phi_1 + 2$$
$$f_3 = \phi_1 - \phi_2 - 1$$

Use a graph to aid your analysis and find the solution to the minimax problem

$$\underset{\boldsymbol{\phi}}{\text{minimize}}\ \max_{i}\ f_i(\boldsymbol{\phi})$$

by starting at $[0\ \ 1]^T$ and using two iterations of a steepest descent algorithm. Test appropriate points for minimax optimality. Why did you choose these points?

Question 2084 Minimize w.r.t. $\boldsymbol{\phi}$ the maximum of

$$f_1 = \phi_1^2 + \phi_2^4$$
$$f_2 = (2 - \phi_1)^2 + (2 - \phi_2)^2$$
$$f_3 = 2\exp(-\phi_1 + \phi_2)$$

The minimax solution occurs at

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

where

$$f_1 = f_2 = 1.95222$$
$$f_3 = 1.57408$$

Suggested starting point: $\phi_1 = \phi_2 = 2$. [Source: Charalambous (1973).]

Question 2085
(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 ω_1, and A_2 dB over the range ω_2 to ω_3, with $0 < \omega_1 < \omega_2 < \omega_3$. At ω_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 unconstrained optimization.
(c) Set up a suitable objective function for the optimization of the notch filter of (a).

Question 2086 Write down explicitly the generalized least pth objective function comprising real functions f_i (not necessarily positive) of $\boldsymbol{\phi}$, level ξ, 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 2087 Derive the gradient vector of the generalized least pth objective of Question 2086 and discuss its features.

Question 2088 Derive necessary conditions for a minimax optimum from the gradient vector of the least pth objective of Question 2087, where the f_i are assumed differentiable functions of $\boldsymbol{\phi}$.

Question 2089 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 ∞ with uniform weighting to all errors.

Question 2090 Solve analytically the problems described in Question 2089 invoking optimality conditions.

Question 2091 Consider the functions e_1 and e_2 of one variable ϕ shown in Fig. Q2091. 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. ϕ.

Question 2092 Consider the functions f_1 and f_2 of one variable ϕ shown in Fig. Q2092. Explain the implications of generalized least pth optimization of f_1 and f_2 w.r.t. ϕ for $p > 0$.

Question 2093 Consider the following two functions of one variable

$$e_1 = -\phi + 4$$
$$e_2 = \phi/3$$

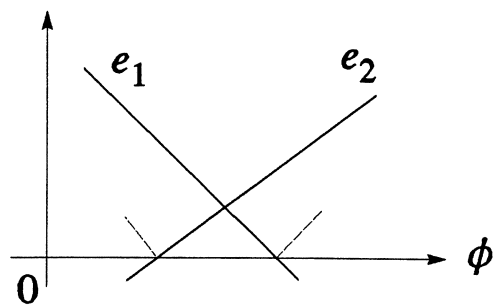

Fig. Q2091 Two error functions of one variable.

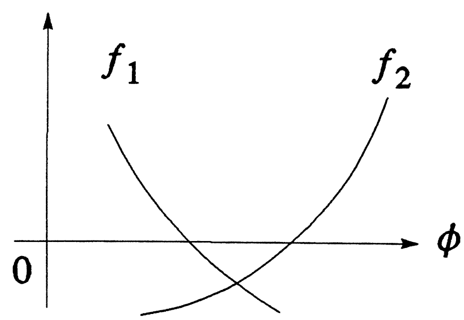

Fig. Q2092 Two functions of one variable.

Expose and explain the distinctive features and implications of

(a) the least pth approximation with $p = 1$ and $p = 2$ of $|e_1|$ and $|e_2|$ w.r.t. ϕ,

(b) the minimax optimization of $|e_1|$ and $|e_2|$ w.r.t. ϕ,

(c) the simultaneous minimization of $|e_1|$ and $|e_2|$ w.r.t. ϕ.

Question 2094 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 α_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

$$S(\omega) \leq 1 \text{ dB} \quad \text{for} \quad 0 \leq \omega \leq 1$$
$$S(\omega) \geq 40 \text{ dB} \quad \text{for} \quad \omega \geq 5$$

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

Question 2095 Suppose that the following table has been derived from impedance measurements at four frequencies.

Frequency (rad/s)	Real Part (Ω)	Imaginary Part (Ω)
1	1.9	1.6
2	2.1	2.9
3	4.5	2.0
4	2.0	6.0

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

Question 2096 Let the Huber approximation objective be represented by

$$U = \frac{1}{2} \sum_{i=1}^{n} \rho(e_i)$$

where

$$\rho(e_i) = \begin{cases} \delta e_i - \dfrac{\delta^2}{2} & \text{if } e_i > \delta \\ \dfrac{e_i^2}{2} & \text{if } |e_i| \le \delta \\ -\delta e_i - \dfrac{\delta^2}{2} & \text{if } e_i < -\delta \end{cases}$$

where e_i is a real error function and δ is a tolerance. Derive a Huber solution for $\delta = 0.25$ for the data of Question 2095. Comment on the solution.

Question 2097 The following table has been derived from impedance measurements at four frequencies. Obtain a uniformly weighted approximation based on *real* approximating functions in (a) the least squares, and (b) the minimax sense for a proposed series RL circuit model with resistance R and inductance L as independent unknowns. Comment on the data in the table.

Frequency (rad/s)	Real Part (Ω)	Imaginary Part (Ω)
1	1.9	1.6
2	2.1	2.9
3	4.4	1.9
4	2.05	6.0

Question 2098 Consider Question 2097, but use least pth approximation with $p = 1$. Comment on your solution.

Question 2099 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 ϕ. 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 2100 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 8005 to fit this specification in the least squares sense.

Question 2101 Suppose we have to minimize

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

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

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 Ω_d is a set of discrete frequencies ω_i. Obtain expressions relating ∇U to $G(j\omega_i)$, where the elements of G are appropriate adjoint sensitivity expressions. Assume convenient values for the excitations of the original and adjoint networks.

Question 2102 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 2103 Consider the circuit shown in Fig. Q2103, which is a linear, time-invariant network with parameters ϕ. 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 ϕ, the optimum of which represents a good match over a band of frequencies Ω. Explain carefully and in detail how the adjoint network method may be used to calculate the gradient vector $\nabla U(\phi)$.

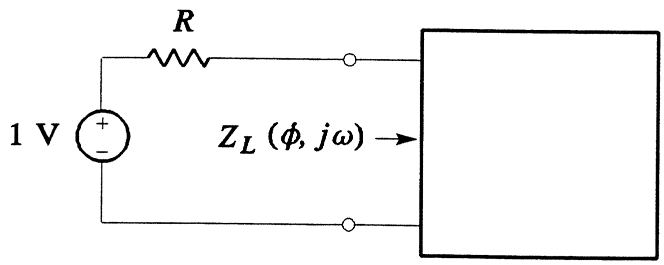

Fig. Q2103 Impedance matching example.

Question 2104 Consider the linear, time-invariant circuit shown in Fig. Q2104 at frequency ω

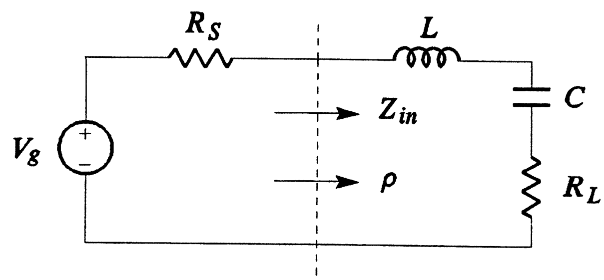

Fig. Q2104 A linear, time-invariant circuit.

We define a complex reflection coefficient ρ as:

$$\rho = \frac{Z_{in} - R_S}{Z_{in} + R_S}$$

Assume that R_s, L and V_s are fixed. Show that $|\rho|^2$ is minimized w.r.t. C for fixed R_L when $\omega^2 LC = 1$. First prove that $d|z|^2/dx = 2Re\{z^* dz/dx\}$.

2.2.3 Sensitivities

Question 3001 Consider the linear circuit shown in Fig. Q3001, which is assumed to be in the sinusoidal steady state. Let $R = 2\ \Omega$, $C = 1$ F, $\omega = 2$ 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.

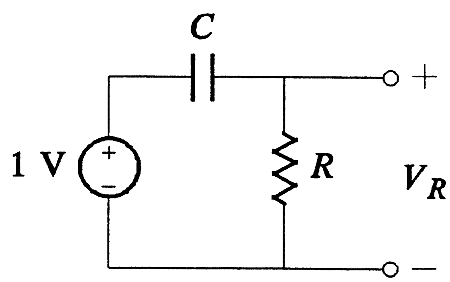

Fig. Q3001 RC circuit.

Question 3002 Consider the linear circuit shown in Fig. Q3002, which is assumed to be in the sinusoidal steady state. Let $V_g = 1$ V, $R_g = 0.5\ \Omega$, $C = 2$ F, $R = 1\ \Omega$, $\omega = 10$ 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?

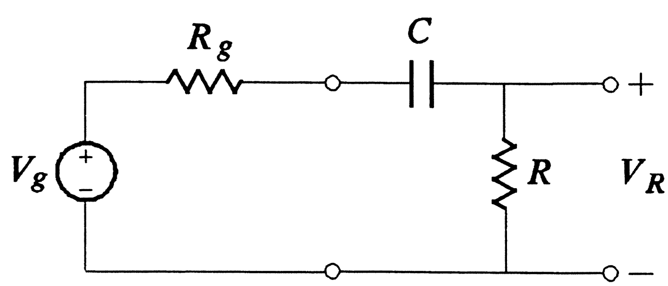

Fig. Q3002 RC circuit.

HAND☞ Question 3003 Consider the linear circuit shown in Fig. Q3002, which is assumed to be in the sinusoidal steady state. Let $V_g = 1$ V, $R_g = 0.5\ \Omega$, $C = 2$ F, $R = 1\ \Omega$, $\omega = 10$ rad/s.

Use the adjoint network approach to evaluate $\partial |V_R|/\partial \omega$. Estimate the change in $|V_R|$ when ω changes by $\pm 1\%$ using this partial derivative and compare with the exact change.

Question 3004 Consider the linear circuit shown in Fig. Q3004, 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_{C2} w.r.t. all the parameters.

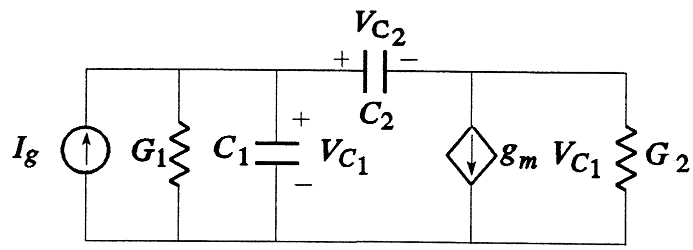

Fig. Q3004 Active circuit.

HAND☞ Question 3005 Derive the first-order sensitivity expression

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

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

Question 3006 Derive from first principles an approach to finding $\partial y_i/\partial \boldsymbol{x}$, where $\boldsymbol{Ay} = \boldsymbol{b}$ is a linear system in $\boldsymbol{y}$, $\boldsymbol{A}$ is a square matrix whose coefficients are nonlinear functions of $\boldsymbol{x}$, the term y_i is the ith component of the column vector $\boldsymbol{y}$ and $\partial y_i/\partial \boldsymbol{x}$ represents a column vector containing partial derivatives of y_i w.r.t. corresponding elements of the column vector $\boldsymbol{x}$. Discuss the computational effort involved.

Question 3007 Derive from first principles an approach to finding $\partial V_i/\partial \omega$, where ω is a frequency, V_i is an ith nodal voltage in the nodal equations of a linear, time-invariant circuit in the frequency domain, namely,

$$\boldsymbol{YV} = \boldsymbol{I}$$

assuming $\boldsymbol{I}$ is independent of ω.

HAND☞ Question 3008 Consider the system of complex linear equations

$$\boldsymbol{YV} = \boldsymbol{I}$$

where $\boldsymbol{Y}$ is a square nodal admittance matrix of constant, complex coefficients, and $\boldsymbol{I}$ 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 $\boldsymbol{V}$.

Question 3009 Consider a nodal system of equations as

$$\begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Express the solution of this system as a least squares problem. Write down the corresponding Jacobian. Write down the gradient vector of the least squares objective.

<u>Question 3010</u> Derive an approach to calculating $\partial y/\partial x_i$, where $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$ is a linear system in $\boldsymbol{y}$, $\boldsymbol{A}$ is a square matrix whose coefficients are nonlinear functions of $\boldsymbol{x}$, and x_i is the ith component of $\boldsymbol{x}$. Discuss the computational effort involved.

<u>Question 3011</u> 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 1012, where x_j and x_k are elements of the vector $\boldsymbol{x}$.

HAND☞ <u>Question 3012</u> Derive from first principles an approach to finding $\partial\lambda/\partial\boldsymbol{x}$, where λ is an eigenvalue of the square matrix $\boldsymbol{A}$ whose coefficients are (in general) nonlinear functions of $\boldsymbol{x}$, i.e.,

$$\boldsymbol{A}\boldsymbol{y} = \lambda\boldsymbol{y}$$

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

<u>Question 3013</u> Derive an approach to calculating

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

for the system described in Question 3012, where x_j and x_k are elements of the vector $\boldsymbol{x}$.

<u>Question 3014</u> Consider the quadratic approximation to a response function given by

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

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

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

<u>Question 3015</u> 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 3016 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 3017 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 3018 Derive from first principles the adjoint element and sensitivity expression for a two-port (see Fig. Q3018(a)) 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. Q3018(b).

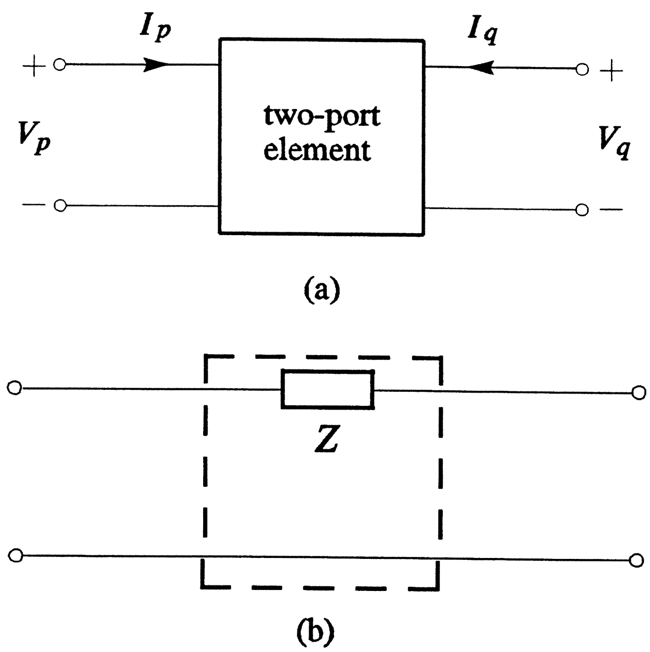

Fig. Q3018 (a) A general two-port network and (b) an example of a two-port.

HAND☞ Question 3019 Consider Question 3018 and apply the result to the element shown in Fig. Q3019 to determine the sensitivity formulas w.r.t. ϕ, where $Y_1 = \phi$ and $Z_2 = 0.5/\phi$.

Question 3020 Consider Question 3018 and apply the result to the element shown in Fig. Q3020 to determine the sensitivity formulas w.r.t. R_1 and R_2, where the elements labelled R_1 are identical.

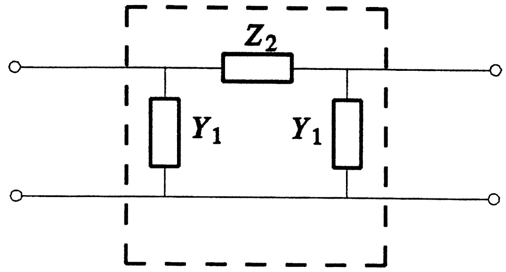

Fig. Q3019 A two-port circuit.

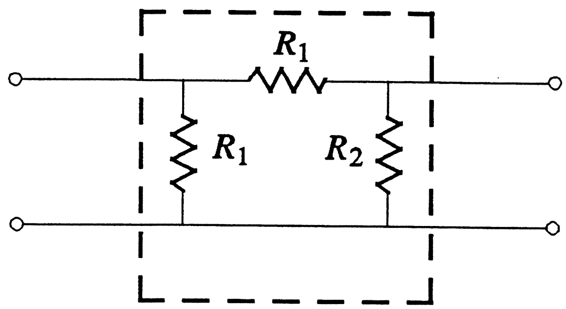

Fig. Q3020 A resistive two-port.

Question 3021 Verify that the adjoint network may be characterized by the hybrid matrix description

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

where the corresponding description for the original network is

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

Question 3022 Verify that, for a network excited by a set of independent voltages J_V and a set of independent currents J_I,

$$G = \sum_{i \in J_V} \hat{V}_i \nabla I_i - \sum_{i \in J_I} \hat{I}_i \nabla V_i$$

where

$$\nabla \triangleq \begin{bmatrix} \frac{\partial}{\partial \phi_1} \\ \frac{\partial}{\partial \phi_2} \\ \vdots \\ \frac{\partial}{\partial \phi_k} \end{bmatrix}$$

implies differentiation w.r.t. k parameters ϕ_1, ϕ_2, ..., ϕ_k and 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. Q3022.

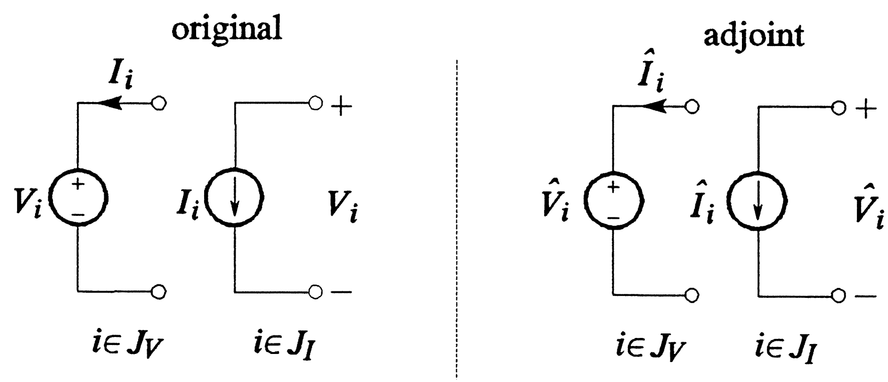

Fig. Q3022 Excitations and responses in the original and adjoint networks.

HAND☞ Question 3023 Consider the formula

$$G = \sum_{\substack{\text{voltage} \\ \text{sources}}} \hat{V}_i \, \nabla I_i - \sum_{\substack{\text{voltage} \\ \text{sources}}} \hat{I}_i \, \nabla V_i$$

where G is a vector of standard sensitivity expressions, i is the index of the sources and ∇ 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 excitations.

Question 3024 Consider the linear circuit shown in Fig. Q3024 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 $v(t)$.

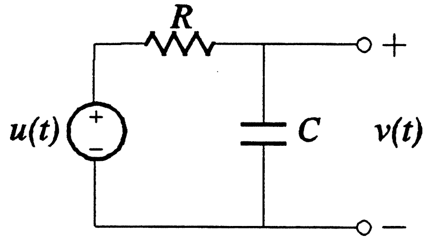

Fig. Q3024 RC circuit.

Question 3025 Consider the linear, time-invariant network shown in Fig. Q3025. Using an abstract approach based on vectors and matrices, derive expressions for $\partial V_1/\partial G_1$, $\partial V_1/\partial G_2$, $\partial V_1/\partial G_3$, $\partial V_1/\partial C_1$, $\partial V_1/\partial C_2$, $\partial V_1/\partial g_m$, and $\partial V_1/\partial \omega$, where ω is the frequency of the excitation.

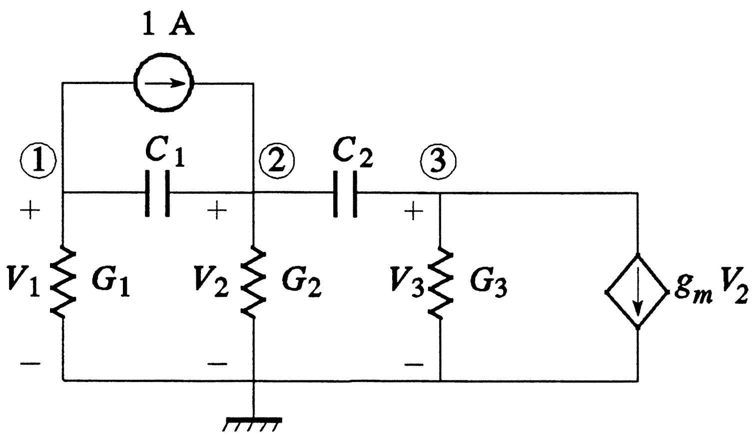

Fig. Q3025 A linear, time-invariant network.

Question 3026 Interpret the expression

$$(u_i - u_j)\, g_m\, (u_k - u_l)^T$$

where u_i, u_j, u_k and u_l are unit column vectors with i, j, k, l respectively, identifying nonzero rows and g_m is a mutual conductance. T denotes transposition.

Question 3027 Consider the linear, time-invariant network shown in Fig. Q3027. Using an abstract approach based on vectors and matrices, derive expressions for $\partial(V_2 - V_3)/\partial G_1$, $\partial(V_2 - V_3)/\partial G_2$, $\partial(V_2 - V_3)/\partial G_3$, $\partial(V_2 - V_3)/\partial L_1$, $\partial(V_2 - V_3)/\partial C_2$, $\partial(V_2 - V_3)/\partial g_m$, and $\partial(V_2 - V_3)/\partial \omega$, where ω is the frequency of the excitation.

Question 3028 Evaluate at 0.5 rad/s the partial derivatives of the input impedance (see Fig. Q3028) w.r.t. the inductors and capacitors of the filter of Question 1044.

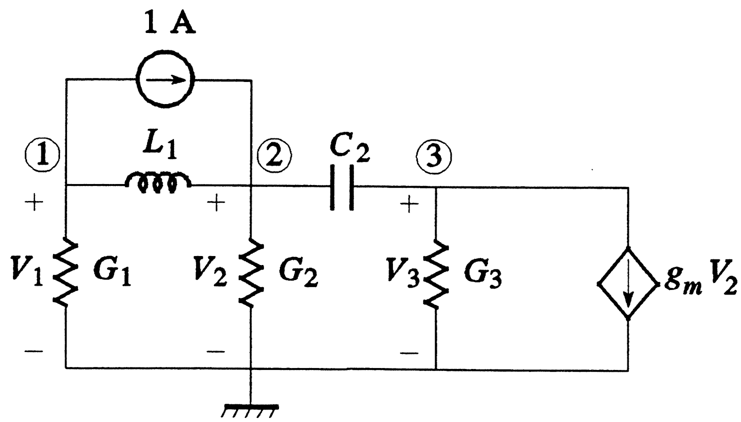

Fig. Q3027 A linear, time-invariant network.

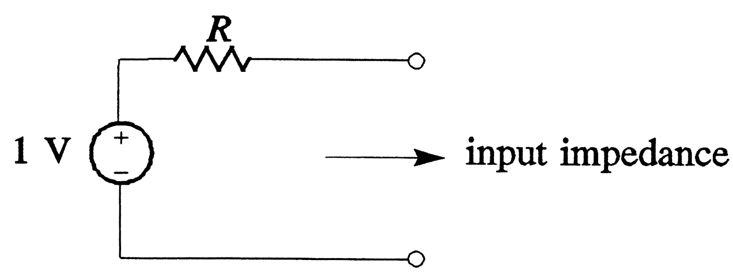

Fig. Q3028 Source for input impedance calculation.

Question 3029 Consider the circuit of Question 8005 at $\omega = 1$ rad/s. Let $L_1 = L_2 = 2$ H, $C = 1$ F. 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 3030 Derive from first principles an approach to finding the exact large change Δy_i due to the large change Δa_{jj}, where $\boldsymbol{A y} = \boldsymbol{b}$ is a linear system in $\boldsymbol{y}$, $\boldsymbol{A}$ is a square matrix, the term y_i is the ith component of the column vector $\boldsymbol{y}$ and a_{jj} represents the jth diagonal element of $\boldsymbol{A}$. Discuss the computational effort involved. [Hint: First find Δy_j.]

HAND☞ Question 3031 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 $\boldsymbol{A y} = \boldsymbol{b}$ is a linear system in $\boldsymbol{y}$, $\boldsymbol{A}$ is a square matrix, the term y_i is the ith component of the column vector $\boldsymbol{y}$ and a_{jk} represents the $\{j, k\}$ element of $\boldsymbol{A}$. Discuss in detail the computational effort involved.

Question 3032 Consider the resistive network of Fig. Q3032. $G_1 = G_3 = G_5 = 1$ S, $R_2 = R_4 = 0.5\ \Omega$.
(a) Calculate the node voltages by LU factorization of the nodal admittance matrix showing all

major steps. Verify that

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

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

(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 ∇V, where

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

using sensitivity formulas shown in the table. (See Question 3033.)

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

OSA☞ Question 3033 Consider the resistive network of Fig. Q3032. $G_1 = G_3 = G_5 = 1$ S, $R_2 = R_4 = 0.5\ \Omega$.
(a) Use OSA90/hope to calculate the node voltages of the original circuit by LU factorization of the nodal admittance matrix. Verify that

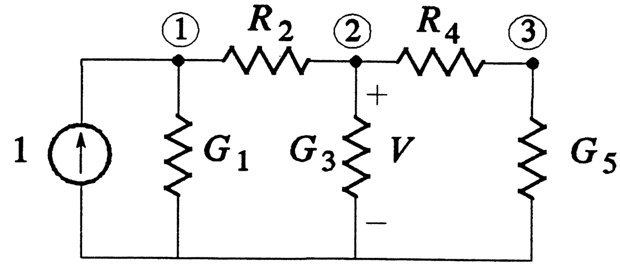

Fig. Q3032 Three–node resistive network.

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

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

(b) Calculate the node voltages of the adjoint circuit using the LU factors already obtained above.
(c) Calculate the node voltages directly from the original and adjoint circuits.
(d) Calculate ∇V, where

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

using sensitivity formulas shown in the table of Question 3032. (See Question 3032.)

HAND☞ Question 3034 Consider the resistive network shown in Fig. Q3034, where G_1 = 1.5 S, G_2 = 2.5 S and i = 10 A. 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.

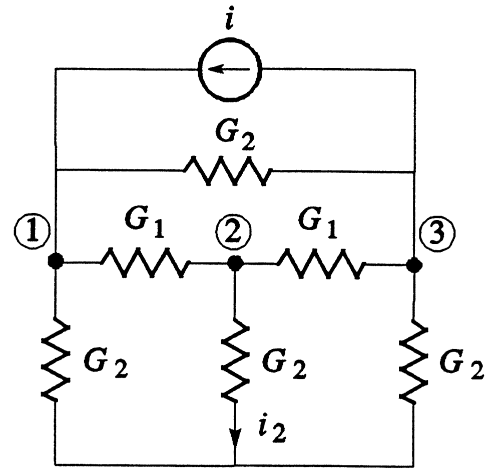

Fig. Q3034 A resistive network.

Question 3035 Consider the resistive network shown in Fig. Q3035, where G_1 = 1 S, G_2 = 1 S, G_3 = 1 S and i = 1 A. Use the adjoint network in conjunction with LU decomposition to evaluate

$$\frac{\partial i_3}{\partial G_1}, \frac{\partial i_3}{\partial G_2}, \frac{\partial i_3}{\partial G_3} \text{ and } \frac{\partial i_3}{\partial i}$$

Check your results by small perturbations. If the tolerances on G_1, G_2 and G_3 are ±3%, estimate the largest and smallest extremes of i_3. Check by direct calculation.

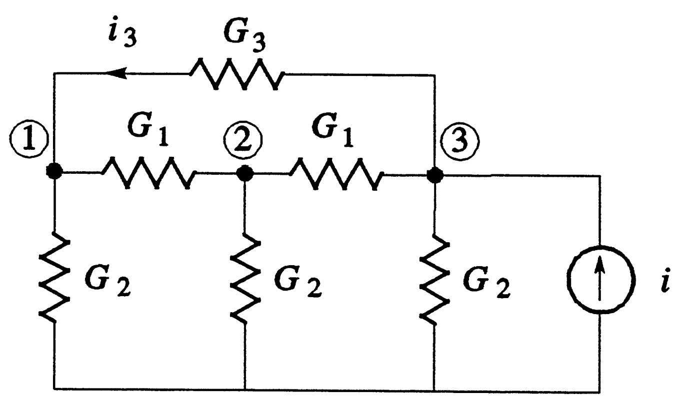

Fig. Q3035 A resistive network.

Question 3036 Consider the resistive network shown in Fig. Q3036, where $G_1 = 3$ S, $G_2 = 5$ S, $G_3 = 4$ S and $i = 1$ A. Use the adjoint network in conjunction with LU decomposition to evaluate

$$\frac{\partial i_2}{\partial G_1}, \frac{\partial i_2}{\partial G_2}, \frac{\partial i_2}{\partial G_3} \text{ and } \frac{\partial i_2}{\partial i}$$

Check your results by small perturbations.

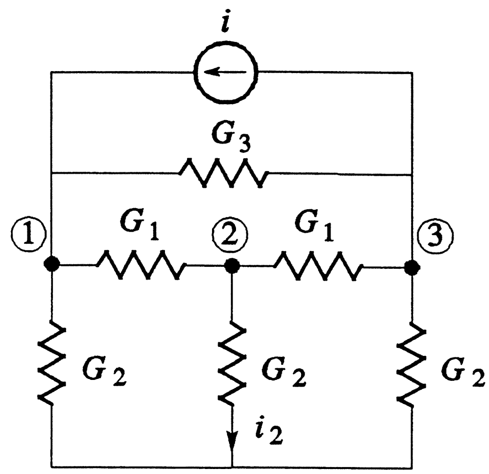

Fig. Q3036 A resistive network.

Question 3037 Consider the resistive network shown in Fig. Q3037, where $G_1 = 1$ S, $G_2 = 1$ S and $G_3 = 1$ S. Use the adjoint network in conjunction with LU decomposition to evaluate

$$\frac{\partial i_3}{\partial G_1}, \frac{\partial i_3}{\partial G_2} \text{ and } \frac{\partial i_3}{\partial G_3}$$

Check your results by small perturbations. If the tolerances on G_1, G_2 and G_3 are ±2%, estimate the largest and smallest extremes of i_3. Check by direct calculation. (See Question 3038.)

OSA☞ Question 3038 Consider the resistive network shown in Fig. Q3037, where $G_1 = 1$ S, $G_2 = 1$ S and $G_3 = 1$ S. Use OSA90/hope to calculate the node voltages of the original circuit by LU factorization of the nodal admittance matrix. Verify the LU matrices in OSA90 and evaluate

$$\frac{\partial i_3}{\partial G_1}, \frac{\partial i_3}{\partial G_2} \text{ and } \frac{\partial i_3}{\partial G_3}$$

If the tolerances on G_1, G_2 and G_3 are ±2%, estimate the largest and smallest extremes of i_3. (See Question 3037.)

OSA☞ Question 3039 Compare the solutions obtained in Question 3038 by creating a circuit model in OSA90/hope.

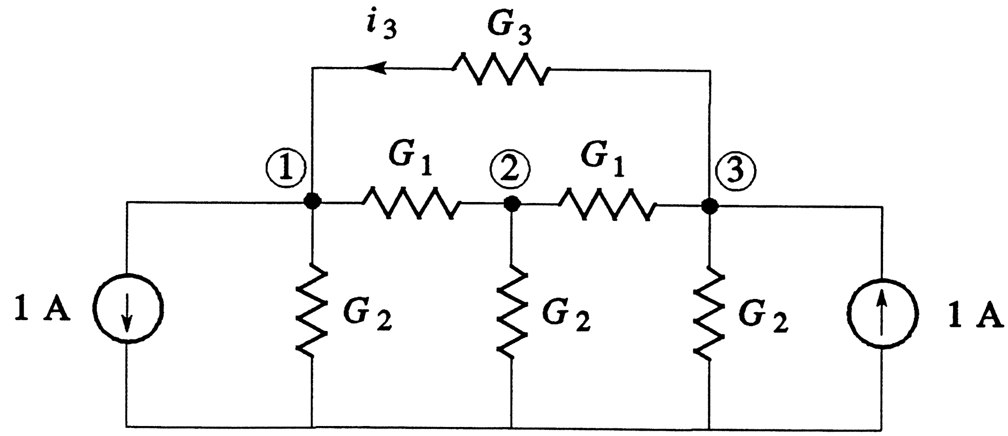

Fig. Q3037 A resistive network.

Question 3040 Consider the resistive network of Question 1036 (Fig. Q1036).

(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 S to 2 S. 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 3041 State the difference form of Tellegen's theorem for the circuit shown in Fig. Q3041. Prove that one adjoint analysis can be used to calculate both Z_{TH}, and V_{TH}, where Z_{TH} is the Thevenin equivalent impedance (set $V_g = 0$) and V_{TH} is the open circuit voltage, as seen at the reference plane. Draw the appropriately excited adjoint network.

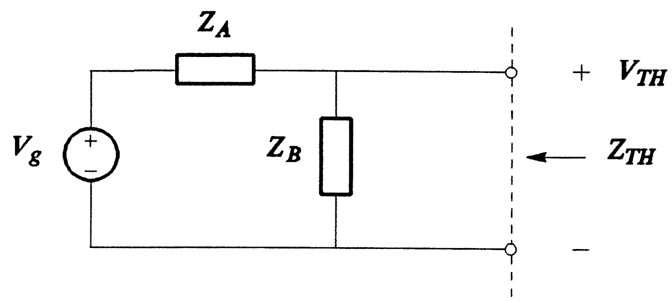

Fig. Q3041 A linear circuit.

Question 3042 Draw the adjoint network for the active circuit shown in Fig. Q3042, 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	$V = RI$	$\hat{V} = R\hat{I}$	$I\hat{I}$	R
Capacitor	$I = j\omega CV$	$\hat{I} = j\omega C\hat{V}$	$-j\omega V\hat{V}$	C
Voltage Controlled Voltage Source	$\begin{bmatrix} I_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \mu & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ I_2 \end{bmatrix}$	$\begin{bmatrix} \hat{I}_1 \\ \hat{V}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\mu \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{V}_1 \\ \hat{I}_2 \end{bmatrix}$	$V_1 \hat{I}_2$	μ

Question 3043 Derive from first principles the adjoint network and sensitivity expressions for the active circuit shown in Fig. Q3042, 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.

Question 3044 Draw the adjoint network for the active circuit shown in Fig. Q3042, which is assumed to be in the sinusoidal steady state and include excitations appropriate to calculating the sensitivities of (a) $V_2(j\omega)$ and (b) $I_{R0}(j\omega)$ w.r.t. all parameters, clearly identifying zero and nonzero

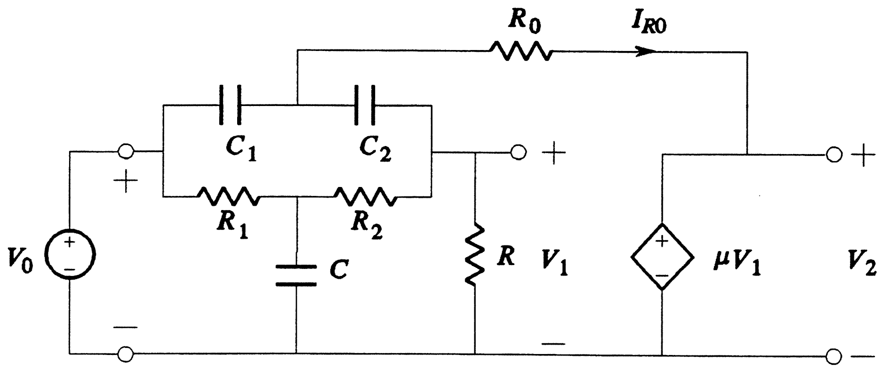

Fig. Q3042 Active circuit example.

excitations. Derive from first principles the sensitivity expression for a capacitor.

Question 3045 Consider the linear circuit of Question 1039, which is assumed to be in the sinusoidal steady state.

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

(a) Write down the nodal equations for the circuit, using the component values and frequency indicated.

(b) Apply the 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.

(c) Factorize the nodal admittance matrix into upper and lower triangular form.

(d) 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.

(e) Estimate ΔV_3 (the total change in V_3) when C_2 changes by +3% and R_1 by —5%. Use $\Delta V_3 \simeq \dfrac{\partial V_3}{\partial C_2}\,\Delta C_2 + \dfrac{\partial V_3}{\partial R_1}\,\Delta R_1$. Check the results by direct perturbation.

(See Question 3046.)

OSA☞ Question 3046 Consider the linear circuit of Question 1039, which is assumed to be in the sinusoidal steady state. Let $V_1 = 1$ V, $\omega = 2$ rad/s, $R_1 = R_2 = R_3 = 2\ \Omega$, $C_1 = C_2 = C_3 = 1$ F. Use OSA90/hope to calculate $\partial V_3/\partial C_2$ and $\partial V_3/\partial R_1$ by the adjoint network method. Estimate ΔV_3 (the total change of V_3) when C_2 changes by +3% and R_1 by —10% . (See Question 3045.)

Question 3047 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 3048 Explain the advantage of the adjoint network method for sensitivity analysis of *very large* linear networks in the frequency domain compared with analytical differentiation.

Question 3049 Discuss carefully the computational effort required in general for each approach used in Question 3045.

Question 3050 Consider the linear equations

$$Ay = b \qquad (1)$$

where A is a constant nxn matrix containing variables ϕ_1 and ϕ_2, b is a constant n-vector and y is a n-vector. The derivative of y w.r.t. ϕ_i is given by

$$\frac{\partial y}{\partial \phi_i} = -A^{-1} \frac{\partial A}{\partial \phi_i} A^{-1} b, \qquad i = 1, 2 \tag{2}$$

Let A_{CONST} be a constant nxn matrix. Let u_i be a unit n-vector containing 1 in its ith position and zeros everywhere else. Suppose $n \geq 4$ and

$$A = A_{CONST} + u_1 u_3^T \phi_1 + u_4 u_2^T \phi_2 \tag{3}$$

Write an algorithm for efficient computation of y, $\dfrac{\partial y_1}{\partial \phi_1}$ and $\dfrac{\partial y_1}{\partial \phi_2}$.

OSA☞ Question 3051 Consider the resistive network shown in Fig. Q3051, with the element values

$$V_{in} = 2V\ , \quad G_1 = 0.5S\ , \quad G_2 = 2.5S \quad \text{and} \quad G_3 = 1.0S\ .$$

Use OSA90/hope to create the adjoint circuit and evaluate

$$\frac{\partial i_{in}}{\partial G_1}, \ \frac{\partial i_{in}}{\partial G_2} \ \text{and} \ \frac{\partial i_{in}}{\partial G_3}$$

If the tolerance on G_1 is +1%, estimate the extreme of i_{in}.

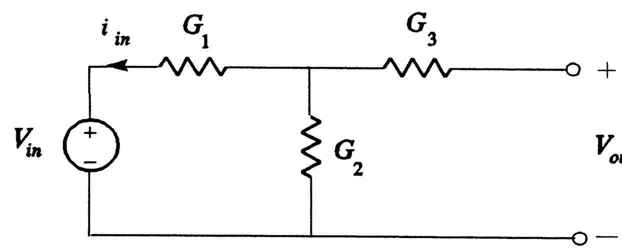

Fig. Q3051 A resistive network.

Question 3052 If $Ay = b$ describes a linear system of equations to be solved for y, what is the corresponding adjoint system? Discuss the implications of LU decomposition as a method of solving the two systems.

2.2.4 Nonlinear Networks

Question 4001 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. Q4001 [Source: Chua and Lin (1975)], where

$$I_d = I_S\,(e^{\lambda V_d} - 1)$$
$$I_S = 10^{-12}\ \text{mA}$$
$$\lambda = 1/V = \frac{1}{0.026}\ \text{V}^{-1}$$
$$E = 10\ \text{V}$$
$$R_1 = R_2 = 1\ \text{k}\Omega$$

Use the results to calculate

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

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

$$V_1 = 5.75 \qquad V_2 = 0.75 \qquad V_3 = 5.0$$

as a test starting point.

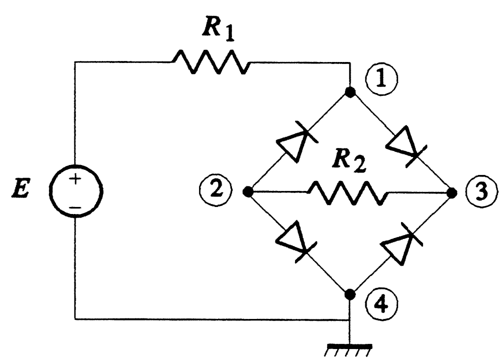

Fig. Q4001 Resistor–diode network.

Question 4002 Consider the circuit in Question 4001 for data, diagram and statement of objectives.

(a) Use only one LU factorization per iteration of Newton's method.

(b) Use the results to find

$$\begin{bmatrix} \dfrac{\partial}{\partial R_1} \\ \dfrac{\partial}{\partial R_2} \\ \dfrac{\partial}{\partial I_s} \\ \dfrac{\partial}{\partial \lambda} \\ \dfrac{\partial}{\partial E} \end{bmatrix} (i_2^2 R_2)$$

subject to satisfying the nonlinear equations.

(c) Check derivatives by perturbation and comment on the results.

(d) Use the test starting point, among others, suggested in Question 4001.

Question 4003 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.

HAND☞ Question 4004 Consider the resistor-diode network shown in Question 4001. Draw the corresponding companion network at the jth iteration for its DC solution. Write down the nodal equations at this iteration.

Question 4005 Consider the resistor-diode network shown in Fig. Q4005. Carefully draw the corresponding companion network at the jth iteration for its DC solution. Write down the nodal equations at this iteration. Create the adjoint system with excitation(s) to calculate the sensitivities of i w.r.t. all parameters. Draw the adjoint circuit. Clearly depict iteration dependent and iteration independent formulas. Each diode is governed by an equation of the form

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

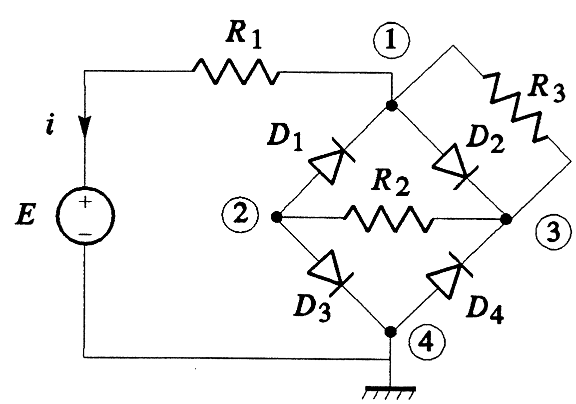

Fig. Q4005 A resistor-diode network.

Question 4006 Consider the resistor-diode network shown in Question 4001. Develop the system of linear equations derived from the nodal equations at the jth iteration for solution by the Newton method. Write down explicitly the Jacobian at the jth iteration.

Question 4007 Consider the resistor-diode network shown in Question 4001. Derive the objective function (or the error function) for optimization to solve the nodal equations.

HAND☞ Question 4008 Consider the nonlinear circuit shown in Fig. Q4008, 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 jth 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 jth iteration and state the corresponding nodal equations.

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

[Source: Chua and Lin (1975).]

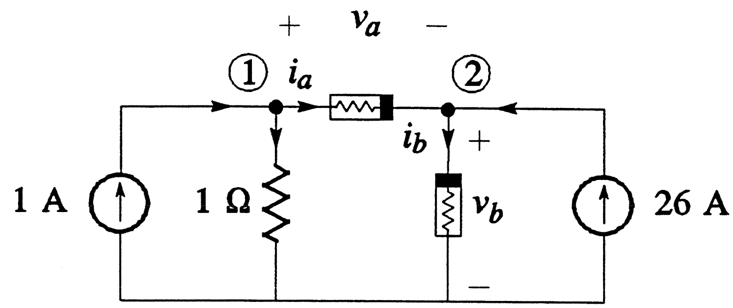

Fig. Q4008 Nonlinear circuit example.

Question 4009 Solve Question 4008 generalized as follows. Let the 1 ohm resistor be R. Let $R = 1$ 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.

2.2.5 One-Dimensional Search Methods

Question 5001 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 5002 Derive from first principles the Golden Section search method for functions of one variable. Discuss any features of the method which you regard as important. Provide diagrams to illustrate your answer.

Question 5003 Apply 3 iterations of the Golden Section search method to the function of one variable given shown in Fig. Q5003. Show clearly all steps and label the diagram 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.

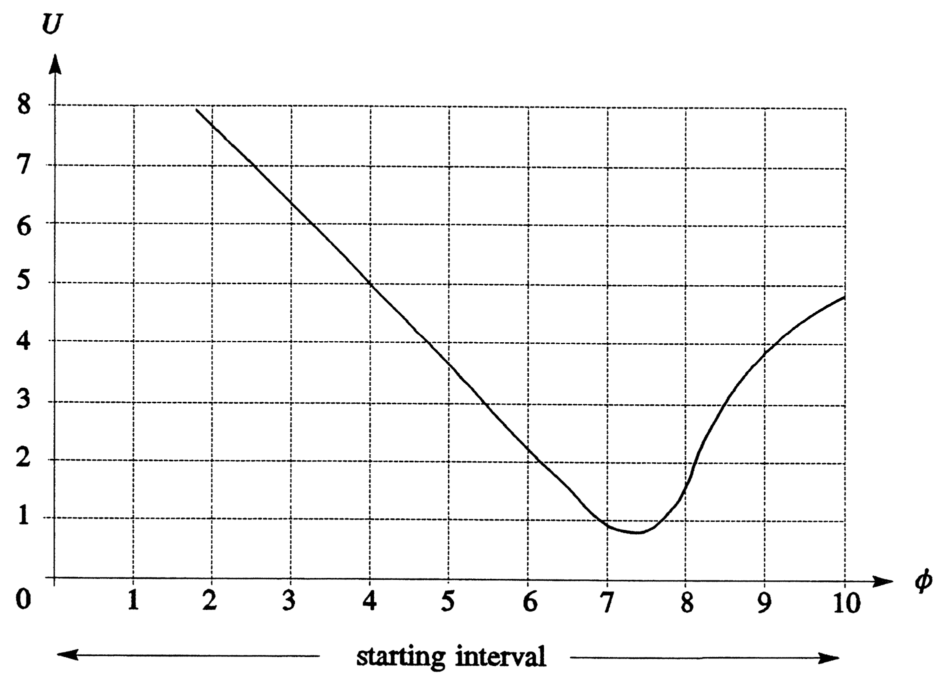

Fig. Q5003 Function of one variable.

Question 5004 Starting with the interval [0, 6], apply 4 iterations of the Golden Section search method to the minimization w.r.t. ϕ of a function described by

$$U = -\phi + 5 \qquad \phi \le 1$$

$$U = 0.5\,(\phi - 3)^2 + 1 \qquad 1 < \phi \le 4$$

$$U = 3 - (\phi - 6)^2/3 \qquad \phi > 4$$

What is the solution obtained? By how much has the interval of uncertainty been reduced? (See Question 5005.)

OSA☞ Question 5005 Starting with the interval [0, 6], use OSA90/hope and apply 20 iterations of the Golden Section search method to the minimization w.r.t. ϕ of the function described in Question 5004. (See Question 5004.)

Question 5006 Consider the minimization of an objective function $U(\phi)$ w.r.t. ϕ. Starting at $\phi = \phi^0$, one dimensional search is performed along a given direction s, i.e.,

$$\phi = \phi^0 + \alpha s$$

Assume that three uniformly spaced points are achieved on the α axis: a, b, c with $a < b < c$ and $U_a > U_b$, $U_b < U_c$.

1) Create a quadratic function of α to fit U at a, b and c.
2) Prove that the minimum of the quadratic function created in 1) is at the point

$$\alpha^* = b + \frac{(b - a)(U_a - U_c)}{2(U_a - 2U_b + U_c)}$$

Note: $b - a = c - b$, $c - a = 2(b - a)$, $c + a = 2b$.

Question 5007 Devise an algorithm for finding the extreme of a well-behaved multimodal function of one variable.

Question 5008 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 ϕ.

2.2.6 Tolerances and Worst-Case Analysis

Question 6001 Consider the voltage divider shown in Fig. Q6001. The specifications are as follows.

$$0.46 \le \frac{R_2}{R_1 + R_2} \le 0.53 \tag{1}$$

$$1.85 \le R_1 + R_2 \le 2.15 \tag{2}$$

Assuming $R_1 \ge 0$, $R_2 \ge 0$, 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.]

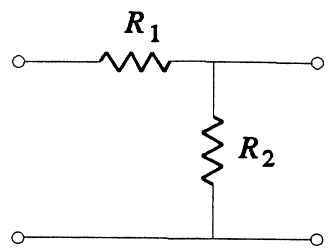

Fig. Q6001 Voltage divider circuit.

Question 6002 Consider the problem defined in Question 6001. Optimize the tolerances ϵ_1 and ϵ_2 on R_1 and R_2 given the cost function

$$C = \frac{R_1^0}{\epsilon_1} + \frac{R_2^0}{\epsilon_2}$$

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

$$R_1 = (R_1^0 + \mu_1\epsilon_1)\,(T^0 + \mu_t\epsilon_t)$$

$$R_2 = (R_2^0 + \mu_2\epsilon_2)\,(T^0 + \mu_t\epsilon_t)$$

where

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

$$T^0 = 1,\ \epsilon_t = 0.05$$

The independent designable variables include R_1^0, R_2^0, ϵ_1, ϵ_2.

Question 6003 Consider the problem defined in Question 6001. Optimize the tolerance ϵ_1 on R_1 given the cost function

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

assuming that R_2 is tunable by ±10% of its nominal value. The independent designable variables include R_1^0, ϵ_1 and R_2^0.

Question 6004 Consider the voltage divider shown in Fig. Q6004 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, ϵ_1 and ϵ_2.

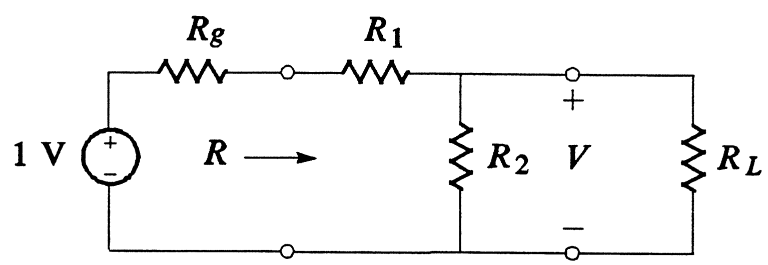

Fig. Q6004 Nonideal voltage divider circuit.

Question 6005 Consider the voltage divider shown in Question 6001. 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 6006 Consider the voltage divider shown in Question 6001. 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.

HAND☞ Question 6007 Consider the voltage divider shown in Question 6001. 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:

$$T \triangleq \frac{V_2}{V_1}, \quad R_1 = 2\ \Omega, \quad R_2 = 1.5\ \Omega$$

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 ±5% and on R_2 is ±10%, estimate the extreme values of T using first partial derivatives. Check the results by direct calculation.

Question 6008 Consider the voltage divider of Question 6001 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 $M \triangleq \max_i f_i$. Show all steps in your calculations.

Question 6009 Describe the concept of optimal design centering, tolerancing and tuning in two dimensions ϕ_1 and ϕ_2, taking ϕ_1 as tunable (zero tolerance) and ϕ_2 as toleranced (zero tuning). Use carefully labelled diagrams involving constraint regions to illustrate the discussion. Suggest objective functions and discuss them. Identify all possible variables, constraints and partial derivatives. [Hint: ϕ_1^0 and ϕ_2^0 are variable nominal values, t_1 is the tuning parameter associated with ϕ_1, and ε_2 is the tolerance associated with ϕ_2.]

Question 6010 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) $\dfrac{1}{\epsilon_1} + \dfrac{1}{\epsilon_2}$

(b) $\log_e \dfrac{\phi_1^0}{\epsilon_1} + \log_e \dfrac{\phi_2^0}{\epsilon_2}$

where ϵ_1 and ϵ_2 are tolerances, and ϕ_1^0 and ϕ_2^0 are nominal values. Formulate the problem as a nonlinear programming problem and give expressions for derivatives.

Question 6011 Consider the voltage divider shown in Question 6001 subject to the same specifications. Optimize the tolerances ϵ_1 and ϵ_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*) $C_1 = \dfrac{R_1^0}{\epsilon_1} + \dfrac{R_2^0}{\epsilon_2}$

(*b*) $C_2 = \dfrac{1}{\epsilon_1} + \dfrac{1}{\epsilon_2}$

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

Question 6012 Consider the voltage divider shown in Fig. Q6001. The specifications are

$$0.46 \le \frac{R_2}{R_1 + R_2} \le 0.53 \qquad (1)$$

$$1.85 \le R_1 + R_2 \le 2.15 \qquad (2)$$

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

(b) Write a suitable objective function for optimal centering and relative tolerance assignment and discuss its features.

(c) Formulate the nonlinear programming problem using the objective function of (b) and the constraints of (a), expressing $U(x)$, $g(x)$, $\nabla_x U(x)$, $\nabla_x g_1(x)$, $\nabla_x g_2(x)$,. . . , where x should represent nominal vector ϕ^0 and tolerance vector ε, $g \ge 0$ and ∇_x is the partial derivative operator w.r.t. all variables.

Question 6013 Consider an acceptable region given by

$$2 + 2\phi_1 - \phi_2 \ge 0$$
$$143 - 11\phi_1 - 13\phi_2 \ge 0$$
$$-60 + 4\phi_1 + 15\phi_2 \ge 0$$

Set up as a nonlinear programming problem the optimal centering of nominal values ϕ^0 with optimal assignment of tolerances ε using the cost function

$$C = \frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2}$$

Derive all the necessary derivatives. Include in your formulation a justifiable selection of candidates for worst-case vertices of the tolerance region.

2.2.7 State Equations

Question 7001 Find the number of state variables and indicate a possible choice of these states for the circuit shown in Fig. Q7001.

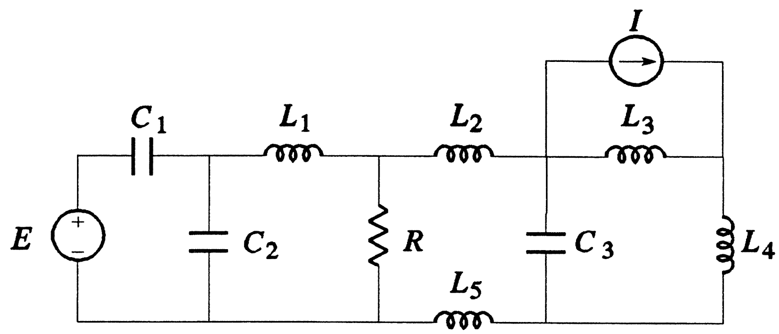

Fig. Q7001 Arbitrary LC network.

Question 7002 The circuit shown in Fig. Q7002 has the state equations

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

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

The parameters are

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

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

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

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

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 7003 Derive the state equations of the circuit shown in Fig. Q7003. Write a Fortran program that implements the fourth-order Runge–Kutta algorithm to solve for the state variables. Take $E = 1$ V 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 your results.

Question 7004 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

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

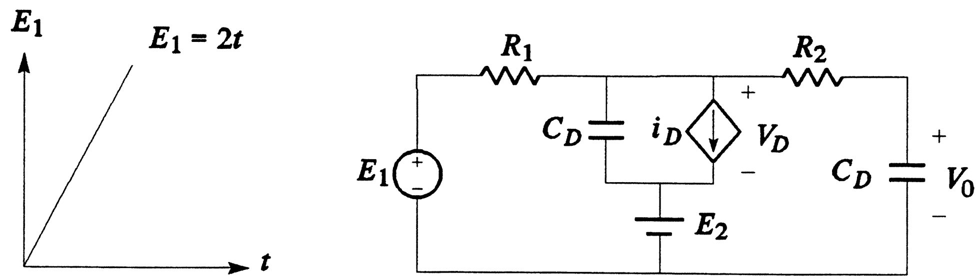

Fig. Q7002 Time domain circuit example.

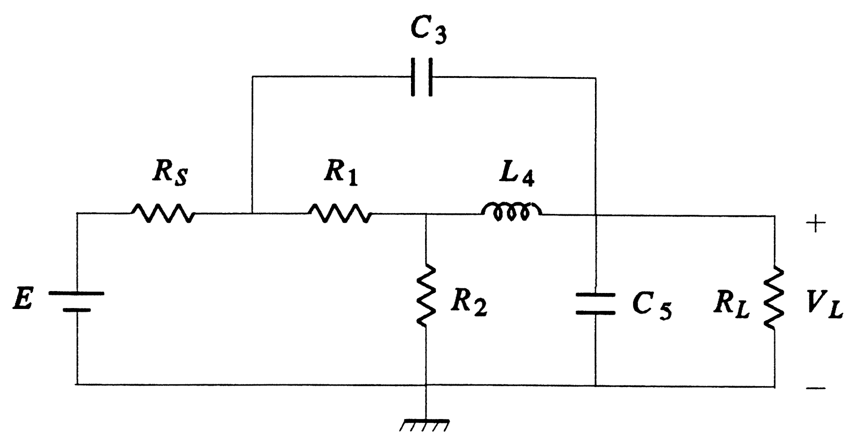

Fig. Q7003 A linear circuit.

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

(a) Use Heun's algorithm.

(b) Use the fourth-order Runge-Kutta method.

2.2.8 Applications

Question 8001 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).] (See Question 8002.)

OSA☞ Question 8002 Use the ℓ_1, ℓ_2 and minimax optimizers of OSA90/hope to approximate in a uniformly weighted sense

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

by

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

on the interval [0, 2]. (See Question 8001.)

Question 8003 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).] (See Question 8004.)

OSA☞ Question 8004 Use the ℓ_1, ℓ_2 and minimax optimizers of OSA90/hope to approximate in a uniformly weighted 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]. (See Question 8003.)

Question 8005 Optimize the LC lowpass filter shown in Fig. Q8005. Write all necessary subprograms to calculate the response and its sensitivities. Verify our results with an available analysis program. (See Question 8006 and Question 8007.)

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

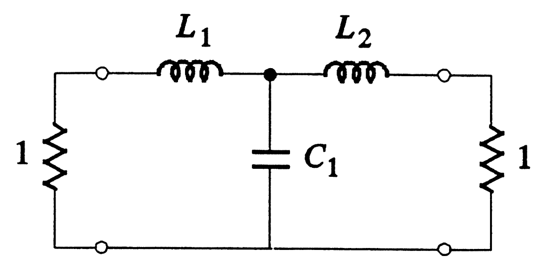

Fig. Q8005 LC lowpass filter.

OSA☞ Question 8006 Use OSA90/hope to perform a nominal point optimization of the three elements in the LC lowpass filter shown in Fig. Q8005 to satisfy the insertion loss design constraints. (See Question 8005.)

OSA☞ Question 8007 Use OSA90/hope to perform a worst-case tolerance optimization of the circuit shown in Fig. Q8005 to satisfy the insertion loss constraints. Use an exact-penalty function formulation to achieve this goal. (See Question 8005.)

Question 8008 Consider a lumped-element LC transformer (Fig. Q8008) 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 $|\rho| = 0.075820$. 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.]

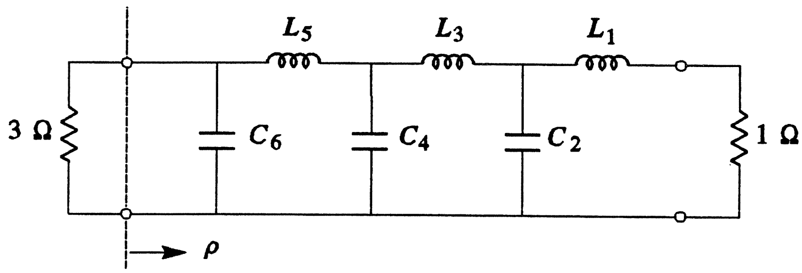

Fig. Q8008 Lumped element LC transformer.

Question 8009 Consider the RC active equalizer shown in Fig. Q8009. 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 $p = 2, 4, 8, ..., \infty$ 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).]

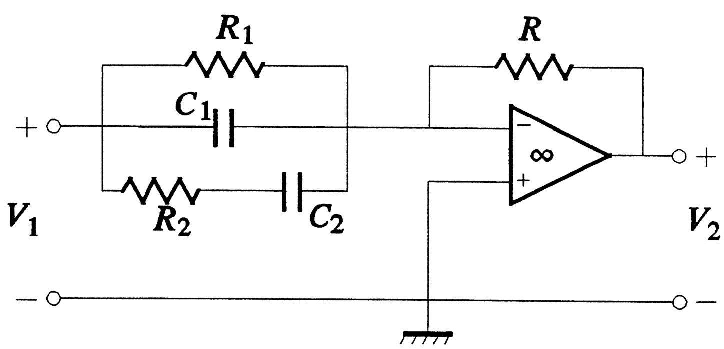

Fig. Q8009 RC active equalizer example.

Question 8010 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

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

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

$$\max_{i\in I} |e_i(\boldsymbol{\phi})|, \; I = \{1, 2, \ldots, 51\}$$

where

$$e_i(\boldsymbol{\phi}) = F(\boldsymbol{\phi}, t_i) - S(t_i)$$

The solution is

$$\phi_1 = 0.68442$$
$$\phi_2 = \pm\, 0.95409$$
$$\phi_3 = 0.12286$$

and the maximum error is 7.9471×10^{-3}. Suggested starting point: $\phi_1 = \phi_2 = \phi_3 = 1$. [See, for example, Bandler (1977).] (See Question 8011.)

OSA☞ <u>Question 8011</u> Use the ℓ_1, ℓ_2 and minimax optimizers of OSA90/hope to make the function

$$F(x) = \frac{a_3}{a_2}\exp(-a_1 x)\sin(a_2 x)$$

best approximate

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

in a uniformly weighted sense. The problem may be discretized on the interval 0 to 10. Suggested starting point: $a_1 = a_2 = a_3 = 1$. (See Question 8010.)

<u>Question 8012</u> Develop a program to calculate and plot the insertion loss of the circuit shown in Fig. Q8012 (elliptic low-pass filter).

Data for the circuit is

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

$C_2 = 0.1022$ F
$C_3 = 1.57677$ F $\qquad L_2 = 1.26033$ H
$C_4 = 0.29139$ F
$C_5 = 0.74177$ F $\qquad L_4 = 1.03950$ H
$0 \le \omega \le 4$ rad/s

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

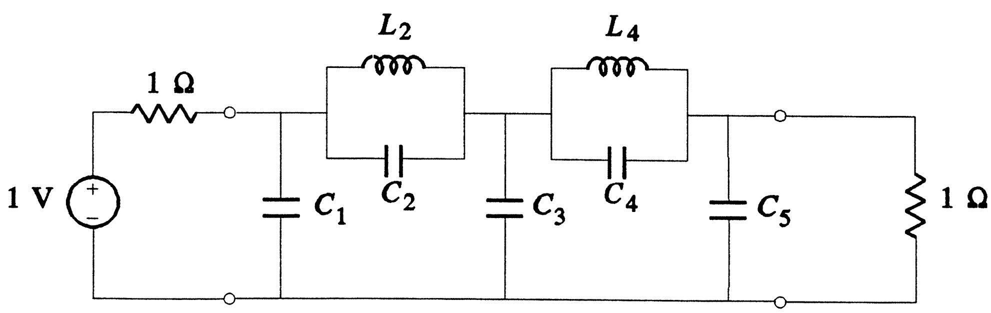

Fig. Q8012 Elliptic low-pass filter.

Question 8013 Consider the LC filter of Question 8005. The minimax solution corresponding to the specifications of Question 8005, 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

$$\frac{L_1^0}{\epsilon_1} + \frac{L_2^0}{\epsilon_2} + \frac{C^0}{\epsilon_C}$$

Parameters	Continuous Solution		Discrete Solution
	Fixed Nominal	Variable Nominal	from {1,2,5,10,15}%
ϵ_1 / L_1^0	3.5%	9.9%	5% 10% 10%
ϵ_C / C_1^0	3.2%	7.6%	10% 5% 10%
ϵ_2 / L_2^0	3.5%	9.9%	10% 10% 5%
L_1^0	1.628	1.999	1.999
C_1^0	1.090	0.906	0.906
L_2^0	1.628	1.999	1.999

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

<u>Question 8014</u> For the circuit of Question 8013 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).]

<u>Question 8015</u> Consider the 10:1 impedance ratio, lossless two-section transmission-line transformer shown in Fig. Q8015. 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 ρ 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 center. 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).] (See Question 8017.)

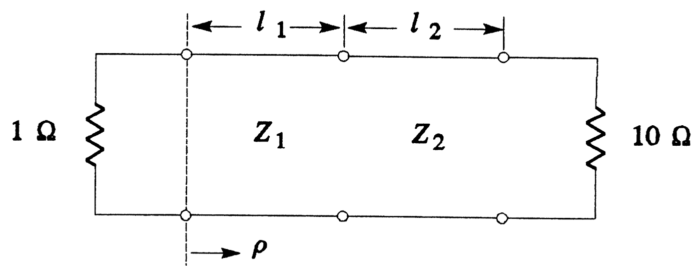

Fig. Q8015 Two-section transmission-line transformer example.

Question 8016 Consider the same circuits, terminations and specifications as in Question 8015. Let ϵ_1 and ϵ_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, ϵ_1 and ϵ_2

(*a*) $C_1 = \dfrac{1}{\epsilon_1} + \dfrac{1}{\epsilon_2}$

(*b*) $C_2 = \dfrac{Z_1^0}{\epsilon_1} + \dfrac{Z_2^0}{\epsilon_2}$

for a worst-case design (yield = 100%). [Source: Bandler, Liu and Chen (1975). See also Abdel-Malek (1977).] (See Question 8018.)

OSA☞ Question 8017 Use OSA90/hope to perform a nominal point optimization of the two section lengths and the characteristic impedances of the circuit shown in Fig. Q8015. Minimize the maximum of the modulus of the reflection coefficient ρ over 100% relative bandwidth. (See Question 8015.)

OSA☞ Question 8018 Use OSA90/hope to perform a worst-case tolerance optimization of the circuit shown in Fig. Q8015 to satisfy the constraint $|\rho| \leq 0.55$. Use an exact-penalty function formulation to achieve this goal for both the relative and absolute cost functions given.

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

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

(See Question 8016.)

Question 8019 Consider the problem described in Question 8015. 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) $1 \le Z_1 \le 3.5$, $3 \le Z_2 \le 6$
(b) $0.8 \le l_1/l_q, l_2/l_q \le 1.2$
(c) $0.8 \le l_1/l_q \le 1.2$, $1 \le Z_1 \le 3.5$
Parameters not specified are held fixed at optimal values. [Source: Bandler and Macdonald (1969).]

Question 8020 Consider the problems described in Questions 8015 and 8019. 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 8021 Consider the same circuit and terminations as in Question 8015 but with three sections. The known quarter-wave solution is given by (See Question 8015 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 8022 Design a recursive digital lowpass filter of the cascade form to best approximate a magnitude response of 1 in the passband, normalized frequency ψ 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 ψ 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 $a_1 = 0$, $b_1 = 0$, $c_1 = 0$, $d_1 = -0.25$, $A = 0.1$, 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 8023 Grow a second section at the solution to Question 8022 and reoptimize appropriately. [See Bandler and Bardakjian (1973).]

Question 8024 Optimize the coefficients of a recursive digital lowpass filter of the cascade form (See Question 8022.) to meet the following specifications:

$0.9 \leq |H| \leq 1.1$ in the passband
$|H| \leq 0.1$ in the stopband

where the passband sample points ψ are

0.0 to 0.18 in steps of 0.02

and the stopband sample points ψ are

0.24
0.3 to 1.0 in steps of 0.1

Begin optimizing with one section starting at $a_1 = 0$, $b_1 = 0$, $c_1 = -1$, $d_1 = 0.5$, $A = 0.1$, 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 8025 Grow a second section at the solution to Question 8024 and reoptimize appropriately. [See Bandler and Bardakjian (1973).]

Question 8026 For the five-section, lossless, transmission-line filter shown in Fig. Q8026, 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

$$Z_1 = Z_3 = Z_5 = 0.2$$
$$Z_2 = Z_4 = 5$$

and the section lengths (normalized to l_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

$$l_1/l_q = l_5/l_q = 0.07$$
$$l_3/l_q = 0.15$$
$$l_2/l_q = l_4/l_q = 0.15$$

[Source for Problem (a): Brancher, Maffioli and Premoli (1970). See also Bandler and Charalambous (1972).] (See Question 8027 and Question 8028.)

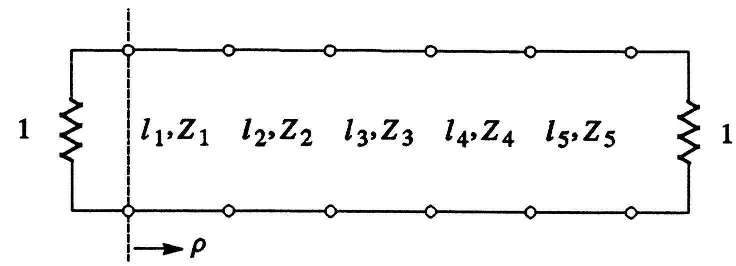

Fig. Q8026 Five-section transmission-line filter.

OSA☞ Question 8027 Use OSA90/hope to perform a nominal point optimization of the five section lengths using the same constraints, characteristic impedances and starting values as Q8026. (See Question 8026.)

OSA☞ Question 8028 Use OSA90/hope to perform a worst-case tolerance optimization of the circuit shown in Fig. Q8026 to satisfy the following insertion loss constraints

Frequency Range (GHz)	Insertion Loss (dB)
0 - 1	< 0.02
5	> 25

Keep the characteristic impedances fixed as in the previous problem. Use an exact-penalty function formulation to achieve this goal for a relative cost function. (See Question 8026.)

<u>Question 8029</u> Solve Question 8026(a) with normalized lengths fixed at 0.2 and impedances variable. [See Levy (1965).]

<u>Question 8030</u> Consider the circuit of Question 8026. 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:

$$l_i = l_q, \quad 0.5 \le Z_i \le 2.0, \qquad i = 1, 2, ..., 5$$

where

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

It is suggested that 21 uniformly spaced frequencies are chosen in the passband. [See Srinivasan (1973) and Carlin (1971).]

<u>Question 8031</u> Reoptimize the example of Question 8030 subject to the constraints

$$\begin{aligned} &0 \le l_i/l_q \le 2, && i = 1, 2, \ldots, 5 \\ &0.4416 \le Z_i \le 4.419, && i = 1, 2, \ldots, 5 \\ &0 \le \sum_{i=1}^{5} l_i/l_q \le 5, && i = 1, 2, \ldots, 5 \end{aligned}$$

where lengths l_i and impedances Z_i are allowed to vary. [See Srinivasan and Bandler (1975).]

<u>Question 8032</u> Consider a third-order lumped-distributed-active lowpass filter as shown in Fig. Q8032. The passband is 0-0.7 rad/s, the stopband 1.415-∞ rad/s. Three design problems are to be solved for minimax results.

(a) 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).

(b) An attenuation and ripple of 1 dB in the passband with the best stopband response.

(c) 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

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

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

Suggested passband sample points are

$$\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.65, 0.7\} \text{ rad/s}$$

Suggested stopband sample points are

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

Let C_2R_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 ϕ_i' are unconstrained while the ϕ_i are positive. [Source: Charalambous (1974).]

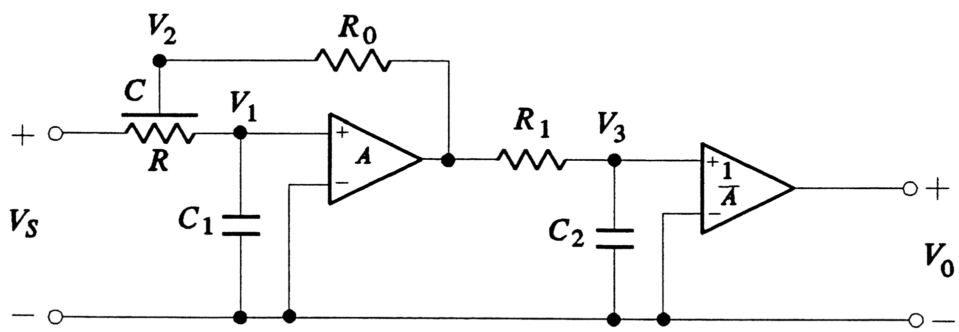

Fig. Q8032 Third-order lumped-distributed-active lowpass filter.

Question 8033 A seven-section, cascaded, lossless, transmission-line filter with frequency-dependent terminations is depicted in Fig. Q8033. The frequency dependence of the terminations is given by

$$R_g = R_L = \frac{377}{\sqrt{1 - \left(\dfrac{f_c}{f}\right)^2}}$$

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 2.16-3 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).] (See Question 8034.)

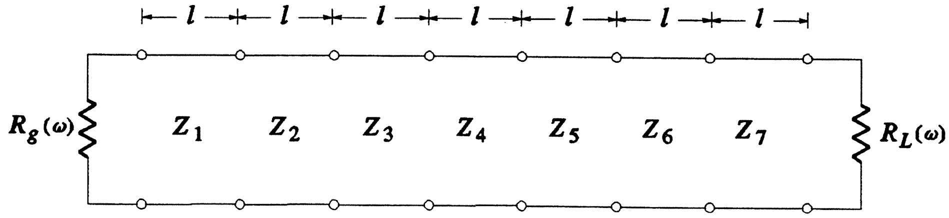

Fig. Q8033 Seven–section, cascaded transmission–line filter.

OSA☞ Question 8034 Use OSA90/hope to perform a nominal point optimization of the seven characteristic impedances in the circuit shown in Fig. Q8033. Satisfy the same insertion loss design constraints as in Q8033. (See Question 8033.)

Question 8035 Consider the active filter shown in Fig. Q8035(a). 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 DC gain and $\omega_a = 12\pi$ rad/s. Use the equivalent circuit shown in Fig. Q8035(b) 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

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

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

$$F \le 1/\sqrt{2} \text{ for } f \le 90 \text{ Hz}$$
$$F \le 1.1 \text{ for } 90 \le f \le 110 \text{ Hz}$$
$$F \le 1/\sqrt{2} \text{ for } f \ge 110 \text{ Hz}$$
$$F \ge 1/\sqrt{2} \text{ for } 92 \le f \le 108 \text{ Hz}$$
$$F \ge 1 \text{ for } f = 100 \text{ Hz}$$

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

$$A_0 = 2 \times 10^5$$
$$R_2 = 2.65 \times 10^4\ \Omega$$
$$C_1 = C_2 = C$$

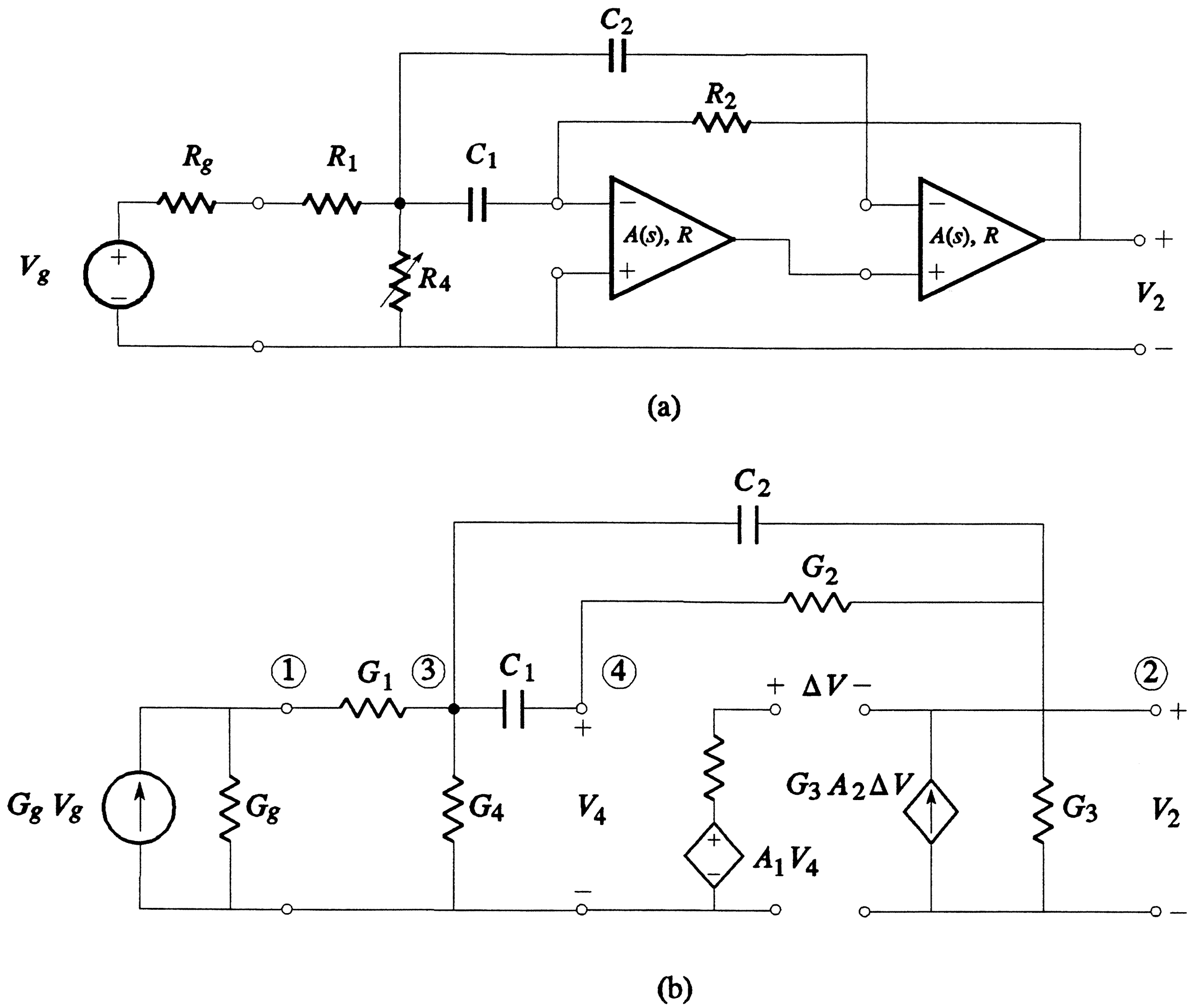

Fig. Q8035 (a) An active filter and (b) its equivalent circuit.

Question 8036 Consider least pth 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 8037 Show, using the generalized least pth 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 8038 Set up and discuss a suitable least pth objective function for approximate minimization of

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

where ϕ contains the adjustable parameters and I denotes an index set relating to the differentiable nonlinear functions f_i, which are not necessarily positive.

Question 8039 Relate the problem formulation of Question 8038 to filter design, taking care to discuss upper and lower response specifications, errors and weighting functions.

Question 8040 Consider the voltage divider example of Question 6012. Express the constraints (1) and (2) suitably for a minimax problem. Take weighting factors as unity.

Consider the point $R_1 = 0.87$, $R_2 = 0.98$.

(a) By suitable calculations, investigate whether this point satisfies the necessary conditions for optimality in the minimax sense.

(b) Calculate the steepest descent direction, if one exists, at this point, assuming that if $|M - f_i| \leq 0.01$ for any f_i then the corresponding f_i is active, where

$$M \triangleq \max_i f_i$$

2.2.9 Various

Question 9001 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 9002 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 9003 Describe all necessary steps required to access the optimization package described in Question 9002 to execute an optimization problem in conjunction with user-supplied programs.

Question 9004 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 9002.

Question 9005 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 9006 For the resistive network of Question 4001, solve the nodal equations by an unconstrained minimization package. Take $G_1 = G_3 = G_5 = 1$ S, $R_2 = R_4 = 0.5\ \Omega$. Write all necessary subprograms.

Question 9007 For the voltage divider of Question 6001, the specifications

$$0.46 \le \frac{R_2}{R_1 + R_2} \le 0.53$$
$$1.85 \le R_1 + R_2 \le 2.15$$

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

Chapter 3

SOLUTIONS TO SOME QUESTIONS

3.1 INTRODUCTION

This chapter contains some solutions done by hand. The sections follow those of the Collected Problems in Chapter 2.

3.2 SOLUTIONS

3.2.1 Algorithm Development

HAND☞ Question 1002 Develop an algorithm to efficiently calculate the value of

$$Z_0 \frac{Z_L + jZ_0 \tan\theta}{Z_0 + jZ_L \tan\theta} \qquad \text{(Q1002.1)}$$

given real Z_0, $0 \le \theta \le \pi$ and complex Z_L, where $\sqrt{-1}$. 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.

Solution

The algorithm can be described as follows.

Step 1 Declare Z_0, Z_L, F and C to be complex.

Step 2 If $|\theta - \frac{\pi}{2}| \ge \varepsilon$, where ε is a small positive number (e.g., $\varepsilon = 10^{-10}$), go to *Step 4*.

Step 3 Set $F \leftarrow Z_0 * \frac{Z_0}{Z_L}$ and stop.

Step 4 Set $C \leftarrow j\tan\theta$.

Step 5 Set $F \leftarrow Z_0 * \frac{(Z_L + C * Z_0)}{(Z_0 + C * Z_L)}$.

Operator count in steps 4 and 5.

Number of multiplications and divisions = 4
Number of additions = 2
Number of trigonometric function evaluations = 1

HAND☞ Question 1017 Derive from first principles an efficient algorithm for solving the tridiagonal system of equations

$$Ax = d \tag{Q1017.1}$$

for x, given arbitrary vector d, where

$$A = \begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{bmatrix} \tag{Q1017.2}$$

using the one-dimensional arrays $a_1, a_2, ..., a_n, b_2, ..., b_n, c_1, ..., c_{n-1}$, explicitly.

Solution

Dividing the first row by a_1, we have

$$\begin{bmatrix} 1 & c_1/a_1 & & & & \\ b_2 & a_2 & c_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & & b_n & a_n \end{bmatrix} x = \begin{bmatrix} d_1/a_1 \\ a_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \tag{Q1017.3}$$

Subtracting b_2×(first row) from the second row, gives

$$\begin{bmatrix} 1 & c_1/a_1 & & & & \\ 0 & a_2 - b_2 c_1/a_1 & c_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & & b_n & a_n \end{bmatrix} x = \begin{bmatrix} d_1/a_1 \\ d_2 - b_2 d_1/a_1 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \tag{Q1017.4}$$

Let

$$c_1/a_1 = c_1', \; a_2' = a_2 - b_2 c_1/a_1, \; d_1' = d_1/a_1 \text{ and } d_2' = d_2 - b_2 d_1/a_1$$

then (Q1017.4) becomes

$$\begin{bmatrix} 1 & c_1' & & & & \\ 0 & a_2' & c_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & & b_n & a_n \end{bmatrix} x = \begin{bmatrix} d_1' \\ d_2' \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \qquad \text{(Q1017.5)}$$

The $(n-1)\times(n-1)$ submatrix in (Q1017.5) has the same structure as that of A. If we continue the same process, we will get

$$\begin{bmatrix} 1 & c_1' & & & & \\ 0 & 1 & c_2' & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & 0 & 1 & c_{n-1}' \\ & & & & 0 & 1 \end{bmatrix} x = \begin{bmatrix} d_1' \\ d_2' \\ \vdots \\ d_{n-1}' \\ d_n' \end{bmatrix} \qquad \text{(Q1017.6)}$$

Then backward substitution can be used to solve (Q1017.6) for x.

The algorithm can be described as

Step 1 Let $c_1 \leftarrow c_1/a_1$, $d_1 \leftarrow d_1/a_1$, $a_1 \leftarrow 1$ and $i \leftarrow 2$.
Step 2 Let $a_i \leftarrow a_i - b_i c_{i-1}$ and $d_i \leftarrow d_i - b_i d_{i-1}$.
Step 3 Let $c_i \leftarrow c_i/a_i$, $d_i \leftarrow d_i/a_i$, $a_i \leftarrow 1$ and $i \leftarrow i + 1$.
Step 4 If $i = N$, go to *Step 5*, otherwise go to *Step 2*.
Step 5 Let $a_n \leftarrow a_n - b_n c_{n-1}$ and $d_n \leftarrow d_n - b_n d_{n-1}$, $d_n \leftarrow d_n/a_n$ and $a_n \leftarrow 1$.
Step 6 Let $x_n \leftarrow d_n$ and $i \leftarrow n - 1$.
Step 7 Let $x_i \leftarrow d_n - c_i x_{i+1}$ and $i \leftarrow i - 1$. If $i < 1$ stop, otherwise go to the beginning of this step.

HAND☞ Question 1023 Consider Question 18. Evaluate

$$\frac{\partial I_3}{R_i}, \qquad i = 1, 2, \ldots, 7 \tag{Q1023.1}$$

for the numerical example, where I_3 is the current flowing in resistor R_3.

Solution

The original network is shown in Fig. SQ1023.1, in which $R_1 = R_3 = R_5 = R_7 = 1\ \Omega$ and $R_2 = R_2 = R_6 = 1/3$.

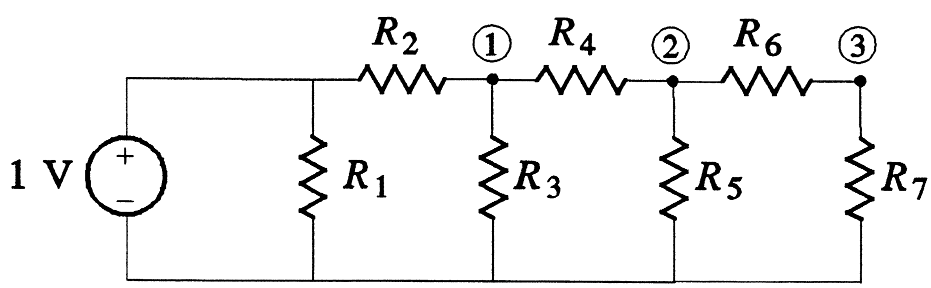

Fig. SQ1023.1 The original network.

The solution vector of the nodal equations of this original network is

$$V = \left[\frac{57}{97} \quad \frac{36}{97} \quad \frac{27}{97}\right]^T \tag{Q1023.2}$$

The branch current vector I_B then can be obtained as

$$I_B = \frac{1}{97}\,[97 \quad 120 \quad 57 \quad 63 \quad 36 \quad 27 \quad 27]^T \tag{Q1023.3}$$

We express I_3, the current flowing in R_3, as

$$I_3 = \frac{V_1}{R_3} \tag{Q1023.4}$$

The gradient vector of I_3 w.r.t. ϕ, where ϕ comprises network parameters, is written using (Q1023.4) as

$$\frac{\partial I_3}{\partial \phi} = \frac{1}{R_3}\frac{\partial V_1}{\partial \phi} - \frac{V_1}{R_3^2}\frac{\partial R_3}{\partial \phi} \tag{Q1023.5}$$

Note that $\frac{\partial R_3}{\partial \phi}$ will have unity when $\phi_1 = R_3$, and otherwise zero.

The adjoint network, in order to find $\frac{\partial V_1}{\partial \phi}$, is shown in Fig. SQ1023.2. The conductance matrix G is unaltered and right-hand side vector of the nodal equations has simply become one-third of the original right hand side vector. Hence,

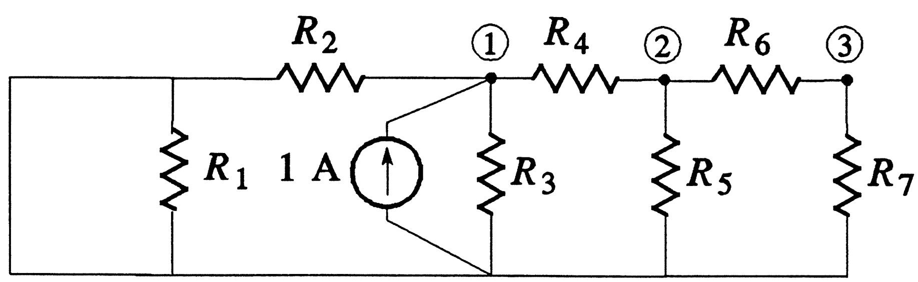

Fig. SQ1023.2 The adjoint network.

$$\hat{V} = \frac{1}{3} V = \frac{1}{97} [19 \quad 12 \quad 9]^T \tag{Q1023.6}$$

and the branch current vector is

$$\hat{I}_B = \frac{1}{97} [0 \quad -57 \quad 19 \quad 21 \quad 12 \quad 9 \quad 9]^T \tag{Q1023.7}$$

Substituting (Q1023.3) and (Q1023.7) into the sensitivity expressions for resistors, i.e.

$$\frac{\partial V_1}{\partial R_i} = I_i \hat{I}_i \qquad i = 1, 2, ..., 7 \tag{Q1023.8}$$

and then substituting the results into (Q1023.5), we obtain

$$\frac{\partial I_3}{\partial \phi} = \frac{1}{97^2} [0 \quad -120\times57 \quad -78\times57 \quad 63\times21 \quad 36\times12 \quad 27\times9 \quad 27\times9]^T \tag{Q1023.9}$$

where $\phi = [R_1 \ R_2 \ R_3 \ R_4 \ R_5 \ R_6 \ R_7]^T$.

3.2.2 Minimization

HAND☞ Question 2002 Given a differentiable function f of many variables $\boldsymbol{x}$ and a corresponding direction vector $\boldsymbol{s}$,

$$\lim_{\lambda \to 0^+} \frac{f(\boldsymbol{x} + \lambda \boldsymbol{s}) - f(\boldsymbol{x})}{\lambda} = \ldots\ldots\ldots\ldots \text{(please state)?} \qquad \text{(Q2002.1)}$$

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

Solution

(Q2002.1) can be written as

$$\lim_{\lambda \to 0^+} \frac{f(\boldsymbol{x} + \lambda \boldsymbol{s}) - f(\boldsymbol{x})}{\lambda} = \nabla f^T \boldsymbol{s} \qquad \text{(Q2002.2)}$$

We can explain (Q2002.2) in two ways

(1) Suppose

$$\nabla f^T \boldsymbol{s} > 0 \qquad \text{(Q2002.3)}$$

then there exists $\sigma > 0$ such that for all λ, $\sigma \geq \lambda > 0$

$$f(\boldsymbol{x} + \lambda \boldsymbol{s}) > f(\boldsymbol{x}) \qquad \text{(Q2002.4)}$$

(2) (Q2002.2) equals the gradient (derivative) of f at point $\boldsymbol{x}$ in the direction of $\boldsymbol{s}$ (directional derivatives).

HAND☞ Question 2005 Use the method of Lagrange multipliers to minimize w.r.t. ϕ_1 and ϕ_2 the function

$$U = \phi_1^2 + \phi_2^2 \tag{Q2005.1}$$

subject to

$$\phi_1 + \phi_2 = 1 \tag{Q2005.2}$$

Sketch a diagram to illustrate the problem and its solution w.r.t. ϕ_1 and ϕ_2. Verify your answer by substituting the constraint into the function.

Solution

Corresponding to (Q2005.1) and (Q2005.2), the Lagrangian function can be written as

$$L(\phi_1, \phi_2, \lambda) = \phi_1^2 + \phi_2^2 + \lambda(\phi_1 + \phi_2 - 1) \tag{Q2005.3}$$

Differentiating (Q2005.3) w.r.t. ϕ_1, ϕ_2, and λ, gives

$$\left.\begin{aligned} \frac{\partial L}{\partial \phi_1} &= 2\phi_1 + \lambda = 0 \\ \frac{\partial L}{\partial \phi_2} &= 2\phi_2 + \lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= \phi_1 + \phi_2 - 1 = 0 \end{aligned}\right\} \tag{Q2005.4}$$

Solving (Q2005.4), we have $\phi_1 = \phi_2 = 1/2$ and $\lambda = -1$. Hence

$$\min_{\phi_1, \phi_2} U(\phi_1, \phi_2) = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2} \tag{Q2005.5}$$

Solution verification

From the constraint (Q2005.2), we have

$$\phi_1 = 1 - \phi_2 \tag{Q2005.6}$$

Substituting (Q2005.6) into (Q2005.1) gives

$$U = (1 - \phi_2)^2 + \phi_2^2 = 2\phi_2^2 - 2\phi_2 + 1 \tag{Q2005.7}$$

Evaluate the derivative of U in (Q2005.7) w.r.t. ϕ_2 and set it to 0. We have

$$\frac{dU}{d\phi_2} = 4\phi_2 - 2 = 0 \tag{Q2005.8}$$

which gives $\phi_2 = 1/2$, thus from (Q2005.6) $\phi_1 = 1/2$. Hence

$$\min_{\phi_1,\, \phi_2} U(\phi_1, \phi_2) = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

The solution in (Q2005.5) is verified. A diagram illustrating the problem and its solution w.r.t. ϕ_1 and ϕ_2 is shown in Fig. SQ2005.

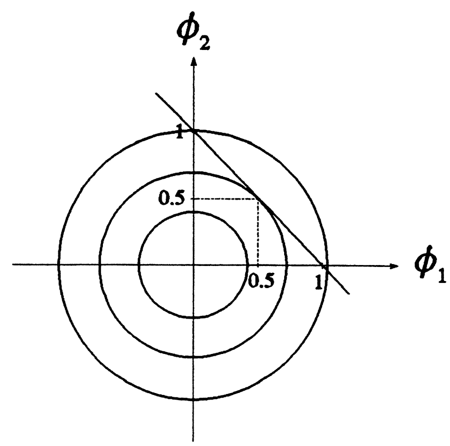

Fig. SQ2005 Problem illustration and its solution.

HAND☞ Question 2013 Consider the parameter constraints

$$0 \le \phi_1 \le \phi_2 \le \dots \le \phi_i \le \dots \le \phi_k \tag{Q2013.1}$$

as the only constraints applicable in a minimization problem.

(a) Find a suitable transformation of the variables ϕ_1, ϕ_2, ..., ϕ_k so that we can use an unconstrained optimization package.

(b) Assuming the new variables are z_1, z_2, ..., write down $\partial U/\partial z_1$, $\partial U/z_2$, ... given $\partial U/\partial \phi_1$, $\partial U/\partial \phi_2$ ...

(c) You have access to a subprogram to calculate U and ∇U given ϕ but you cannot alter it. How would you organize your software and data to handle the transformed problem?

Solution

(a) According to the parameter constraints in (Q2013.1), we can use the following transformation to convert the problem to unconstrained optimization.

$$\begin{gathered}
\phi_1 \ge 0 \;\Rightarrow\; \phi_1 = z_1^2 \\
\phi_2 \ge \phi_1 \;\Rightarrow\; \phi_2 - \phi_1 \ge 0 \;\Rightarrow\; \phi_2 - \phi_1 = z_2^2 \;\Rightarrow\; \phi_2 = z_2^2 + \phi_1 = z_2^2 + z_1^2 \\
\vdots \\
\phi_{j+1} \ge \phi_j \;\Rightarrow\; \phi_{j+1} - \phi_j \ge 0 \;\Rightarrow\; \phi_{j+1} - \phi_j = z_{j+1}^2 \;\Rightarrow\; \phi_{j+1} = z_{j+1}^2 + \phi_j = \sum_{i=1}^{j+1} z_i^2
\end{gathered} \tag{Q2013.2}$$

where z_i is unconstrained.

(b) Given $\partial U/\partial \phi_1$, $\partial U/\partial \phi_2$, ..., $\partial U/\partial \phi_k$, we can write down $\partial U/\partial z_1$, $\partial U/\partial z_2$, ..., $\partial U/\partial z_k$ as

$$\begin{aligned}
\frac{\partial U}{\partial z_i} &= \frac{\partial U}{\partial \phi_1}\frac{\partial \phi_1}{\partial z_1} + \frac{\partial U}{\partial \phi_2}\frac{\partial \phi_2}{\partial z_i} + \dots + \frac{\partial U}{\partial \phi_k}\frac{\partial \phi_k}{\partial z_i} \\
&= \frac{\partial U}{\partial \phi_1}2z_i + \frac{\partial U}{\partial \phi_2}2z_i + \dots + \frac{\partial U}{\partial \phi_k}2z_i \\
&= 2z_i\sum_{j=1}^{k}\frac{\partial U}{\partial \phi_j} \qquad i = 1, 2, \dots, k
\end{aligned} \tag{Q2013.3}$$

(c) We can use the following algorithm to handle the transformed problem.

Step 1 For z given by the optimization package calculate ϕ using (Q2013.2),

Step 2 call the subprogram to calculate U and $\nabla_\phi U$ using ϕ obtained in *Step 1*,

Step 3 calculate $\nabla_z U$ using $\nabla_\phi U$ obtained in *Step 2* and formulas in (Q2013.3).

HAND☞ Question 2015 Derive the gradient vector of $U(\phi)$ w.r.t. ϕ for the objective functions

$$U = \int_{\psi_l}^{\psi_u} |e(\phi, \psi)|^p \, d\psi \tag{Q2015.1}$$

and

$$U = \sum_{i=1}^{n} |e_i(\phi)|^p \tag{Q2015.2}$$

where the appropriate error functions are complex.

Solution

In order to derive the gradient vector of U w.r.t. ϕ, we first derive the gradient vector of $|e|$ w.r.t. ϕ.

$$\begin{aligned} \nabla|e| &= \nabla(ee^*)^{\frac{1}{2}} = \frac{1}{2}(ee^*)^{-\frac{1}{2}}[(\nabla e)e^* + e\nabla e^*] \\ &= \frac{1}{2|e|} 2Re[(\nabla e)e^*] = |e|^{-1} Re[(\nabla e)e^*] \\ &= |e| Re(\frac{1}{e}\nabla e) \end{aligned} \tag{Q2015.3}$$

where * denotes the conjugate of a complex number. Therefore, the gradient vector of U in (Q2015.1) and (Q2015.2) w.r.t. ϕ can be respectively derived as

$$\begin{aligned} \nabla U &= \nabla \int_{\psi_l}^{\psi_u} |e(\phi,\psi)|^p \, d\psi \\ &= p \int_{\psi_l}^{\psi_u} \left[|e(\phi,\psi)|^p \, Re\left(\frac{1}{e}\nabla e\right)\right] d\psi \\ &= p \int_{\psi_l}^{\psi_u} \left[\, |e(\phi,\psi)|^{p-2} \, Re(e^*\nabla e)\right] d\psi \end{aligned} \tag{Q2015.4}$$

$$\begin{aligned}\nabla U &= \nabla \sum_{i=1}^{n} |e_i(\phi)|^p \\ &= p \sum_{i=1}^{n} \left[|e_i(\phi)|^p \, Re\left[\frac{1}{e_i} \nabla e_i\right]\right] \qquad \text{(Q2015.5)} \\ &= p \sum_{i=1}^{n} \left[|e_i(\phi)|^{p-2} \, Re(e_i^* \nabla e_i)\right]\end{aligned}$$

HAND☞ Question 2020 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.

Solution

(a) If the first and second derivatives of the objective $U(\phi)$ to be minimized 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 ϕ, i.e.

$$U(\phi + \Delta\phi) \approx U(\phi) + (\nabla U)^T \Delta\phi + \frac{1}{2}\Delta\phi^T H \Delta\phi \qquad \text{(Q2020.1)}$$

or

$$U(\phi + \Delta\phi) - U(\phi) \approx (\nabla U)^T \Delta\phi + \frac{1}{2}\Delta\phi^T H \Delta\phi \qquad \text{(Q2020.2)}$$

The minimal point can be obtained by solving (Q2020.1) or (Q2020.2) in new variables $\Delta\phi$. This can be done by differentiating (Q2020.1) or (Q2020.2) w.r.t. $\Delta\phi$ and letting it be zero, which results in

$$\Delta\phi = -H^{-1}\nabla U \qquad \text{(Q2020.3)}$$

Then the Newton update formula can be written as

$$\phi^{j+1} = \phi^j - (H^j)^{-1}(\nabla U)^j \qquad \text{(Q2020.4)}$$

(b) For a set of nonlinear equations $F(\phi)$, we have

$$F(\phi + \Delta\phi) = F(\phi) + J\Delta\phi + \ldots \qquad \text{(Q2020.5)}$$

or

$$F(\phi + \Delta\phi) - F(\phi) \approx J\Delta\phi \qquad \text{(Q2020.6)}$$

where J is the Jacobian matrix. We hope that $\Delta\phi$ could make $F(\phi + \Delta\phi) = \mathbf{0}$. Hence, we have

$$J\Delta\phi = -F(\phi) \qquad \text{(Q2020.7)}$$

or

$$\Delta\phi = -J^{-1}F(\phi) \qquad \text{(Q2020.8)}$$

The success of using the Newton's method is based on how adequately the objective function is represented by a quadratic form. If H is positive definite we can expect proper convergence. But for some cases, $-H^{-1}\nabla U$ might not even point downhill. Factor α could be introduced so that

$$\Delta\phi = -\alpha H^{-1}\nabla U \qquad \text{(Q2020.9)}$$

where α is chosen to minimize U.

In the case of slow convergence of the Newton's method or that $-H^{-1}\nabla U$ does not point downhill, a damping technique can be used. (Q2020.3) becomes

$$\Delta\phi = -(H + \lambda 1)^{-1}\nabla U \qquad \text{(Q2020.10)}$$

where λ is a damping factor. If $\lambda = 0$ we have the original problem. If $\lambda = \infty$ we have the steepest descent direction. To state the effect of damping clearly, let us consider one-dimensional problem. From (Q2020.10) we have in one-dimensional case

$$\Delta\phi = -\frac{U'}{U'' + \lambda} \qquad \text{(Q2020.11)}$$

Consider the objective function shown in Fig. SQ2020. At point ϕ^0, $-(U'')^{-1}U'$ points uphill direction since $U' < 0$ and $U'' = 0$. If we choose $\lambda > |U''|$ in (Q2020.11), then $\Delta\phi > 0$ which points to the downhill direction.

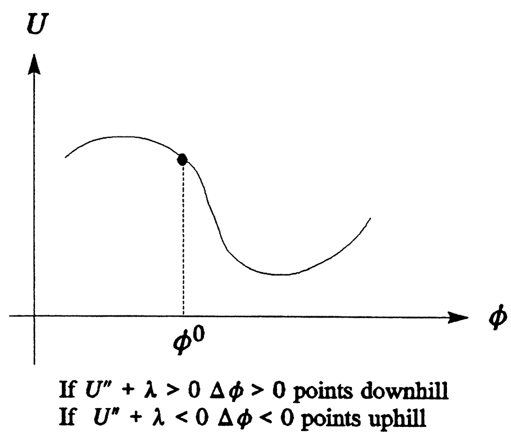

Fig. SQ2020 A one-dimensional objective function.

HAND☞ Question 2028 We wish to calculate $\frac{\partial f}{\partial \boldsymbol{x}}$ subject to $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}$, where $f \equiv f(\boldsymbol{y}(\boldsymbol{x}), \boldsymbol{x})$ given values for $\boldsymbol{x}$. Explain fully the formula

$$\left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\boldsymbol{h}=\boldsymbol{0}} = -\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{x}}\hat{\boldsymbol{y}} + \frac{\partial f}{\partial \boldsymbol{x}}, \tag{Q2028.1}$$

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

$$\left[\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}}\right]\hat{\boldsymbol{y}} = \frac{\partial f}{\partial \boldsymbol{y}}. \tag{Q2028.2}$$

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

$$\begin{aligned} &4x_1^2 y_1^2 - 3y_2 - 2 = 0 \\ &-x_1 y_1 + 2x_2^2 y_1 y_2 - 3y_2 = 0 \\ &f = y_1^2 + x_1 \end{aligned} \tag{Q2028.3}$$

Set up all the matrices and vectors required for both the solution of the nonlinear equations and also for the evaluations of $\frac{\partial f}{\partial \boldsymbol{x}}$ subject to $\boldsymbol{h} = \boldsymbol{0}$.

Solution

We are given

$$f \equiv f(\boldsymbol{y}(\boldsymbol{x}), \boldsymbol{x}) \tag{Q2028.4}$$
$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0} \tag{Q2028.5}$$

Differentiating (Q2028.4) w.r.t. $\boldsymbol{x}$ gives

$$\frac{\partial f}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{x}}\frac{\partial f}{\partial \boldsymbol{y}} + \frac{\partial f}{\partial \boldsymbol{x}} \tag{Q2028.6}$$

Since the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}$, we should eliminate $\frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{x}}$ by using $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}$. Differentiating (Q2028.5) w.r.t. $\boldsymbol{x}$, we have

$$\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}} = \boldsymbol{0} \Rightarrow \frac{\partial \boldsymbol{y}^T}{\partial \boldsymbol{x}} = -\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{x}}\left[\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}}\right]^{-1} \tag{Q2028.7}$$

Substituting (Q2028.7) into (Q2028.6), we obtain

$$\left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\boldsymbol{h}=\boldsymbol{0}} = -\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{x}}\left[\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}}\right]^{-1}\frac{\partial f}{\partial \boldsymbol{y}} + \frac{\partial f}{\partial \boldsymbol{x}} \tag{Q2028.8}$$

If we define

$$\left[\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}}\right]^{-1} \frac{\partial f}{\partial \boldsymbol{y}} = \hat{\boldsymbol{y}} \quad \text{or} \quad \frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}} \hat{\boldsymbol{y}} = \frac{\partial f}{\partial \boldsymbol{y}} \tag{Q2028.9}$$

Then

$$\left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\boldsymbol{h}=\mathbf{0}} = -\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{x}} \hat{\boldsymbol{y}} + \frac{\partial f}{\partial \boldsymbol{x}} \tag{Q2028.10}$$

The analytical effort includes the explicit derivation of $\frac{\partial f}{\partial \boldsymbol{x}}$, $\frac{\partial f}{\partial \boldsymbol{y}}$, $\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{x}}$ and $\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}}$.

The computational effort

If we do not include the computational effort involved in partial derivative computation, we have to solve one linear system $\boldsymbol{Ay} = \boldsymbol{b}$ which needs $(n^3 + 3n^2 - n)/3$ multiplications and another n^2 multiplications. So totally we have $(n^3 + 6n^2 - n)/3$ multiplications. Also the computational effort should include that in solving $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{0}$.

When

$$\boldsymbol{h} = \begin{bmatrix} 4x_1^2 y_1^2 - 3y_2 - 2 \\ -x_1 y_1 + 2x_2^2 y_1 y_2 - 3y_2 \end{bmatrix} = \mathbf{0} \tag{Q2028.11}$$

$$\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{x}} = \begin{bmatrix} 8x_1 y_1^2 & -y_1 \\ 0 & 4x_2 y_1 y_2 \end{bmatrix} \tag{Q2028.12}$$

$$\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}} = \begin{bmatrix} 8x_1^2 y_1 & -x_1 + 2x_2^2 y_2 \\ -3 & 2x_2^2 y_1 - 3 \end{bmatrix} \tag{Q2028.13}$$

$$\frac{\partial f}{\partial \boldsymbol{x}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \frac{\partial f}{\partial \boldsymbol{y}} = \begin{bmatrix} 2y_1 \\ 0 \end{bmatrix} \tag{Q2028.14}$$

The adjoint system is

$$\begin{bmatrix} 8x_1^2 y_1 & -x_1 + 2x_2^2 y_2 \\ -3 & 2x_2^2 y_1 - 3 \end{bmatrix} \hat{\boldsymbol{y}} = \begin{bmatrix} 2y_1 \\ 0 \end{bmatrix} \tag{Q2028.15}$$

Therefore,

$$\left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\boldsymbol{h}=\mathbf{0}} = -\begin{bmatrix} 8x_1 y_1^2 & -y_1 \\ 0 & 4x_2 y_1 y_2 \end{bmatrix} \hat{\boldsymbol{y}} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{Q2028.16}$$

Now we need to solve the nonlinear equation $\boldsymbol{h} = \mathbf{0}$. Using the first-order Taylor series, we have

$$\boldsymbol{h}^{j+1} = \boldsymbol{h}^{j} + \left[\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}}\right]_j^T (\boldsymbol{y}^{j+1} - \boldsymbol{y}^{j}) = \boldsymbol{0} \qquad \text{(Q2028.17)}$$

$$\left[\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}}\right]_j^T \boldsymbol{y}^{j+1} = \left[\frac{\partial \boldsymbol{h}^T}{\partial \boldsymbol{y}}\right]_j^T \boldsymbol{y}^{j} - \boldsymbol{h}^{j} \qquad \text{(Q2028.18)}$$

Substituting (Q2028.13) into (Q2028.18) gives

$$\begin{bmatrix} 8x_1^2 y_1 & -x_1 + 2x_2^2 y_2 \\ -3 & 2x_2^2 y_1 - 3 \end{bmatrix}_j^T \boldsymbol{y}^{j+1} = \begin{bmatrix} 8x^2 & -x_1 + 2x_2^2 y_2 \\ -3 & 2x_2^2 y_1 - 3 \end{bmatrix}_j^T \boldsymbol{y}^{j} - \boldsymbol{h}^{j} \qquad \text{(Q2028.19)}$$

(Q2028.19) is solved iteratively until $\boldsymbol{h}^{j+1} = \boldsymbol{0}$.

HAND☛ Question 2064 The updating formula for the Fletcher-Powell-Davidon method is defined by

$$H^0 = 1 \tag{Q2064.1}$$

$$s^j = -H^j \nabla U^j, \qquad j = 0, 1, 2, \ldots \tag{Q2064.2}$$

where

$$H^{j+1} = H^j + \frac{\Delta\phi^j \Delta\phi^{j^T}}{\Delta\phi^{j^T} g^j} - \frac{H^j g^j g^{j^T} H^j}{g^{j^T} H^j g^j} \tag{Q2064.3}$$

$$\Delta\phi^j \triangleq \alpha^j s^j = \phi^{j+1} - \phi^j \tag{Q2064.4}$$

$$g^j \triangleq \nabla U^{j+1} - \nabla U^j \tag{Q2064.5}$$

(a) What is H^j and what is its relationship with the Hessian matrix of a function $U(\phi)$? How is α^j computed in practice?

(b) Apply the algorithm (using a theoretically justified approach to obtain α^j) to the minimization of

$$2\phi_1^2 + 3\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 2 \tag{Q2064.6}$$

w.r.t. ϕ_1 and ϕ_2 starting at $\phi_1 = 1$ and $\phi_2 = 1$. Show all steps explicitly and comment on the results taken. Draw an accurate diagram showing the path taken.

Solution

(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. α^j is computed using one-dimensional search method where $U(\phi^j + \alpha^j s^j)$ is the objective function and α^j is the variable.

(b) For the function of

$$U = 2\phi_1^2 + 3\phi_2^2 + \phi_1\phi_2 + 2\phi_1 + 2 \tag{Q2064.7}$$

we have

$$\nabla U = \begin{bmatrix} 4\phi_1 + \phi_2 + 2 \\ 6\phi_2 + \phi_1 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} \phi + \begin{bmatrix} 2 \\ 0 \end{bmatrix} \tag{Q2064.8}$$

The first iteration

$$j = 0, \quad \phi^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \nabla U^0 = \begin{bmatrix} 7 \\ 7 \end{bmatrix}, \quad H^0 = 1$$

$$s^0 = -H^0 \nabla U^0 = -\begin{bmatrix} 7 \\ 7 \end{bmatrix}$$

$$\phi^1 = \phi^0 + \alpha^0 s^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \alpha^0 \begin{bmatrix} -7 \\ -7 \end{bmatrix} = \begin{bmatrix} 1 - 7\alpha^0 \\ 1 - 7\alpha^0 \end{bmatrix}$$

In order to find α^0, we let

$$\frac{\partial U}{\partial \alpha} = \left(\frac{\partial U}{\partial \phi}\right)^T \left(\frac{\partial \phi}{\partial \alpha}\right)^T = (\nabla U)^T s = 0$$

That is,

$$\left(\begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} \begin{bmatrix} 1 - 7\alpha^0 \\ 1 - 7\alpha^0 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix}\right)^T \begin{bmatrix} -7 \\ -7 \end{bmatrix} = \begin{bmatrix} 5(1 - 7\alpha^0) + 2 \\ 7(1 - 7\alpha^0) \end{bmatrix}^T \begin{bmatrix} -7 \\ -7 \end{bmatrix} = 0$$

The solution is

$$\alpha^0 = \frac{14}{84} = \frac{1}{6} = 0.16667$$

The second iteration

$$j = 1, \quad \phi^1 = \phi^0 + \alpha^0 s^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \frac{1}{6} \begin{bmatrix} -7 \\ -7 \end{bmatrix} = \begin{bmatrix} -0.16667 \\ -0.16667 \end{bmatrix}$$

$$\Delta\phi^0 = \phi^1 - \phi^0 = \begin{bmatrix} -1.16667 \\ -1.16667 \end{bmatrix}$$

$$\nabla U^1 = \begin{bmatrix} 1.6667 \\ -1.6667 \end{bmatrix}$$

$$g^0 = \nabla U^1 - \nabla U^0 = \begin{bmatrix} -5.8333 \\ -8.1667 \end{bmatrix}$$

$$H^1 = H^0 + \frac{\Delta\phi^0 \Delta\phi^{0^T}}{\Delta\phi^{0^T} g^0} - \frac{H^0 g^0 g^{0^T} H^0}{g^{0^T} H^0 g^0}$$

$$= 1 + \frac{1}{16.334} \begin{bmatrix} 1.3612 & 1.3612 \\ 1.3612 & 1.3612 \end{bmatrix} - \frac{1}{100.72} \begin{bmatrix} 34.627 & 47.639 \\ 47.639 & 66.695 \end{bmatrix}$$

$$= \begin{bmatrix} 0.74550 & -0.38965 \\ -0.38965 & 0.42115 \end{bmatrix}$$

$$s^1 = -H^1 \nabla U^1 = \begin{bmatrix} -1.3244 \\ 0.94596 \end{bmatrix}$$

Let

$$\frac{\partial U}{\partial \alpha^1} = (\nabla U^1)^T s^1 = 0$$

Then we have

$$\left(\begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} (\phi^1 + \alpha^1 s^1) + \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right)^T s^1 = 0$$

$$\phi^{1^T} \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} s^1 + s^{1^T} \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} s^1 \alpha^1 + \begin{bmatrix} 2 \\ 0 \end{bmatrix}^T s^1 = 0$$

Hence,

$$\alpha^1 = \frac{\phi^{1^T} \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} s^1 - \begin{bmatrix} 2 \\ 0 \end{bmatrix} s^1}{s^{1^T} \begin{bmatrix} 4 & 1 \\ 1 & 6 \end{bmatrix} s^1} = \frac{0.00005 + 2.6488}{9.8795} = 0.26811$$

The third iteration

$$j = 2, \quad \phi^2 = \phi^1 + \alpha^1 s^1 = \begin{bmatrix} -0.52176 \\ 0.086957 \end{bmatrix}$$

Since

$$\nabla U \mid_{\phi=\phi^2} = 0$$

result is obtained in 2 iterations. The path taken is shown in Fig. SQ2064.

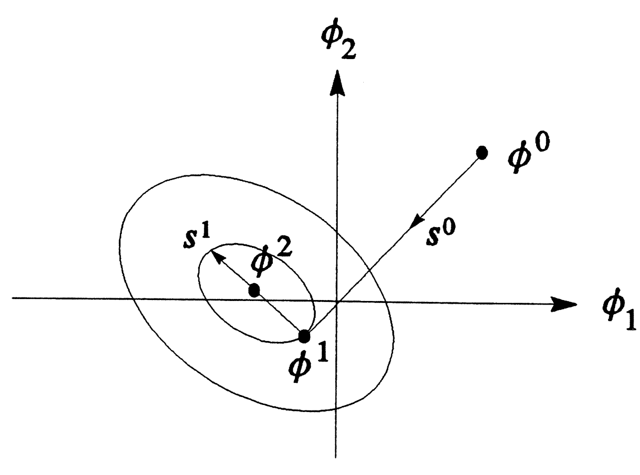

Fig. SQ2064 The path taken for minimization.

HAND☞ Question 2070

(a) State and explain the iterative formulas defining the conjugate gradient method of minimizing a differentiable function $U(\boldsymbol{\phi})$ in terms of direction vectors $\boldsymbol{s}^j$ and $\boldsymbol{s}^{j-1}$, and gradient vectors ∇U^j and ∇U^{j-1}. Hint: take

$$\boldsymbol{s}^j = -\nabla U^j + \beta^j \boldsymbol{s}^{j-1} \tag{Q2070.1}$$

where

$$\beta^j = \frac{(\nabla U^j)^T \nabla U^j}{(\nabla U^{j-1})^T \nabla U^{j-1}}. \tag{Q2070.2}$$

(b) State the formula for a quadratic function $U(\boldsymbol{\phi})$ in terms of Hessian matrix $\boldsymbol{A}$, constant vector $\boldsymbol{b}$, and constant c associated with variable vector $\boldsymbol{\phi}$.

(c) State and explain the formula describing the property of conjugate directions $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ 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 $\boldsymbol{s}^0 = -\nabla U^0$. By using the property

$$\nabla U^j - \nabla U^{j-1} = \alpha^{j-1} \boldsymbol{A} \boldsymbol{s}^{j-1} \tag{Q2070.3}$$

for the quadratic function of (b) where

$$\alpha^{j-1} \boldsymbol{s}^{j-1} \triangleq \boldsymbol{\phi}^j - \boldsymbol{\phi}^{j-1} \tag{Q2070.4}$$

prove that $\boldsymbol{s}^1$ and $\boldsymbol{s}^0$ are conjugate w.r.t. $\boldsymbol{A}$. [Hint: Verify Equation (Q2070.3), explain Equation (Q2070.4) and assume that a full linear search for a minimum is conducted in each search direction $\boldsymbol{s}^j$.]

(e) Discuss and illustrate the implications of conjugate directions in the minimization of an unconstrained differentiable function in many variables. Discuss the properties of the conjugate gradient algorithm, its advantages and disadvantages.

Solution

(a) Given a starting point $\boldsymbol{\phi}^0$, a conjugate gradient iteration is defined by

$$\boldsymbol{\phi}^j = \boldsymbol{\phi}^{j-1} + \alpha^{j-1} \boldsymbol{s}^{j-1} \tag{Q2070.5}$$

where

$$\boldsymbol{s}^j = -\nabla U^j + \beta^j \boldsymbol{s}^{j-1} \tag{Q2070.6}$$

and

$$\beta^j = \frac{(\nabla U^j)^T \nabla U^j}{(\nabla U^{j-1})^T (\nabla U^{j-1})}, \quad \boldsymbol{s}^0 = -\nabla U^0, \quad \nabla U^j \triangleq \nabla U(\boldsymbol{\phi}^j) \tag{Q2070.7}$$

The value of α^{j-1} is determined through a line search along $\boldsymbol{s}^{j-1}$.

(b) The formula for a quadratic function $U(\boldsymbol{\phi})$ in terms of Hessian matrix $\boldsymbol{A}$, constant vector $\boldsymbol{b}$, and constant c associated with variable vector $\boldsymbol{\phi}$ can be written as

$$U(\boldsymbol{\phi}) = \frac{1}{2}\, \boldsymbol{\phi}^T A \boldsymbol{\phi} + \boldsymbol{b}^T \boldsymbol{\phi} + c \qquad \text{(Q2070.8)}$$

(c) The directions $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ are said to be conjugate w.r.t. a positive definite matrix $\boldsymbol{A}$ if

$$\boldsymbol{u}_i^T\, \boldsymbol{A}\, \boldsymbol{u}_j = 0 \qquad \text{(Q2070.9)}$$

(d) From the definition of a conjugate gradient iteration given in (Q2070.5), we have

$$\alpha^{j-1}\, \boldsymbol{s}^{j-1} = \boldsymbol{\phi}^j - \boldsymbol{\phi}^{j-1} \qquad \text{(Q2070.10)}$$

For a quadratic function given by (Q2070.8), we can obtain

$$\nabla U = \boldsymbol{A\phi} + \boldsymbol{b} \qquad \text{(Q2070.11)}$$

Therefore,

$$\nabla U^j - \nabla U^{j-1} = \boldsymbol{A}\,(\boldsymbol{\phi}^j - \boldsymbol{\phi}^{j-1}) = \alpha^{j-1}\, \boldsymbol{A}\, \boldsymbol{s}^{j-1} \qquad \text{(Q2070.12)}$$

For the first iteration

$$\boldsymbol{s}^0 = -\nabla U^0 \qquad \text{(Q2070.13)}$$

A full line search along $\boldsymbol{s}^0$ implies that

$$(\boldsymbol{s}^0)^T\, \nabla U^1 = 0 \quad \Rightarrow \quad (\nabla U^0)^T\, \nabla U^1 = 0 \qquad \text{(Q2070.14)}$$

We wish to show

$$(\boldsymbol{s}^1)^{\mathrm{T}}\, \mathrm{A}\, \boldsymbol{s}^0 = 0 \qquad \text{(Q2070.15)}$$

Notice from (Q2070.6) that

$$\boldsymbol{s}^1 = -\nabla U^1 + \beta^1 \boldsymbol{s}^0 \qquad \text{(Q2070.16)}$$

Let j = 1 in (Q2070.12), we have

$$\boldsymbol{A}\, \boldsymbol{s}^0 = \frac{1}{\alpha^0}\, (\nabla U^1 - \nabla U^0) \qquad \text{(Q2070.17)}$$

Use (Q2070.7), (Q2070.13), (Q2070.14), (Q2070.16) and (Q2070.17), we have

$$(s^1)^T A\, s^0 = (-\nabla U^1 + \beta^1 s^0)^T \frac{1}{\alpha^0} (\nabla U^1 - \nabla U^0)$$

$$= \frac{1}{\alpha^0} (-\nabla U^1 - \beta^1 \nabla U^0)^T (\nabla U^1 - \nabla U^0)$$

$$= \frac{1}{\alpha^0} [-(\nabla U^1)^T \nabla U^1 + (\nabla U^1)^T \nabla U^0 - \beta^1 (\nabla U^0)^T \nabla U^1 + \beta^1 (\nabla U^0)^T \nabla U^0] \quad \text{(Q2070.18)}$$

$$= \frac{1}{\alpha^0} \left[-(\nabla U^1)^T \nabla U^1 + \frac{(\nabla U^1)^T \nabla U^1}{(\nabla U^0)^T \nabla U^0} (\nabla U^0)^T \nabla U^0 \right]$$

$$= 0$$

Therefore, s^1 and s^0 are conjugate w.r.t. $\boldsymbol{A}$.

(e) For a quadratic function described in (b) which has n variables (i.e., $\boldsymbol{\phi} = [\phi_1\ \phi_2\ \dots\ \rho_{n1}]^T$), the conjugate gradient method will find the minimum after no more than n steps, i.e., there is $\boldsymbol{\phi}^j = \boldsymbol{\phi}^*$, $j \le n$, $\boldsymbol{\phi}^*$ 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. The advantages of the conjugate gradient 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). Illustration of optimization for a two-dimensional quadratic function using the conjugate gradient method is shown in Fig SQ2070.

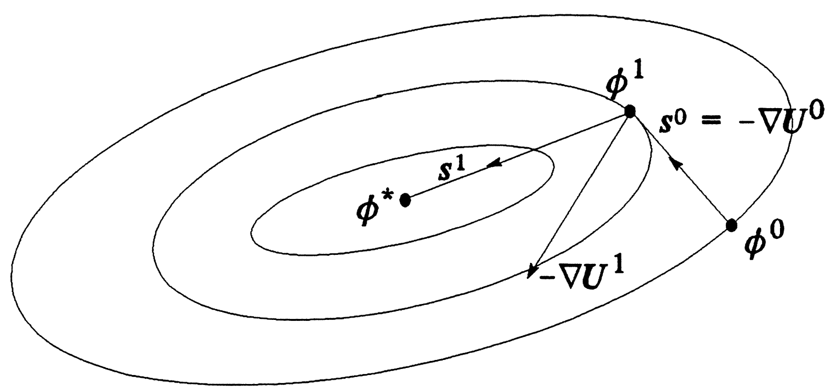

Fig. SQ2070 Illustration of optimization for a two-dimensional quadratic function using the conjugate gradient method.

3.2.3 Sensitivities

HAND☞ Question 3003 (p. 3-27)
HAND☞ Question 3005 (p. 3-29)
HAND☞ Question 3008 (p. 3-30)
HAND☞ Question 3012 (p. 3-31)
HAND☞ Question 3019 (p. 3-32)
HAND☞ Question 3023 (p. 3-34)
HAND☞ Question 3031 (p. 3-36)
HAND☞ Question 3034 (p. 3-37)

HAND☞ Question 3003 Consider the linear circuit shown in Fig. SQ3003.1 which is assumed to be in the sinusoidally steady state. Let $V_g = 1$ V, $R_g = 0.5\ \Omega$, $C = 2$ F, $R = 1\ \Omega$, $\omega = 10$ rad/s. Use the adjoint network approach to evaluate $\partial|V_R|/\partial\omega$. Estimate the changes in $|V_R|$ when ω changes by ±1% using this partial derivative and compare with the exact changes.

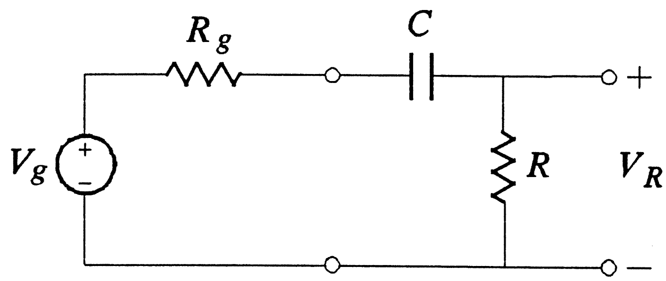

Fig. SQ3003.1 A linear RC circuit.

Solution

From the adjoint network approach, we have

$$\frac{\partial V_R}{\partial \omega} = -V_{R_g}\hat{V}_{R_g}\frac{\partial\left[\frac{1}{R_g}\right]}{\partial\omega} - V_R\hat{V}_R\frac{\partial\left[\frac{1}{R}\right]}{\partial\omega} - V_C\hat{V}_C\frac{\partial(j\omega C)}{\partial\omega} = -jCV_C\hat{V}_C \qquad \text{(Q3003.1)}$$

From Fig. Q3003.1 we can obtain

$$V_R = \frac{RV_g}{R_g + R + \dfrac{1}{j\omega C}} = \frac{1}{1.5 - j0.05} \qquad \text{(Q3003.2)}$$

$$V_C = \frac{\dfrac{1}{j\omega C}V_g}{R_g + R + \dfrac{1}{j\omega C}} = \frac{1}{1 + j30} \qquad \text{(Q3003.3)}$$

The adjoint network is shown in Fig. SQ3003.2 from which we can obtain

$$\hat{V}_C = -\frac{R\dfrac{1}{j\omega C}}{R_g + R + \dfrac{1}{j\omega C}} = -\frac{1}{1 + j30} \qquad \text{(Q3003.4)}$$

Therefore,

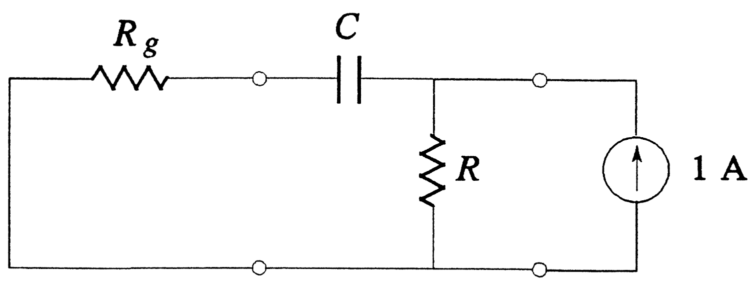

Fig. SQ3003.2 The corresponding adjoint circuit.

$$\frac{\partial |V_R|}{\partial \omega} = |V_R| Re \left[\frac{1}{V_R} \frac{\partial V_R}{\partial \omega} \right]$$

$$= \frac{1}{\sqrt{1.5^2 + 0.05^2}} Re \left[(1.5 - j0.05) \frac{j2}{(1 + j30)^2} \right] \quad \text{(Q3003.5)}$$

$$= 7.39\times10^{-5}$$

When ω changes ±1%, the estimated changes in $|V_R|$ are

$$\Delta |V_R| \approx \frac{\partial |V_R|}{\partial \omega} \Delta\omega = \frac{\partial |V_R|}{\partial \omega} (\pm 0.01\times10) = \pm 7.39\times10^{-6} \quad \text{(Q3003.6)}$$

We can calculate the exact changes of $|V_R|$ from (Q3003.2). The results are

$$0\frac{1}{2}\Delta |V_R| \Big|_{\Delta\omega = 0.1} = \frac{1}{\sqrt{1.5^2 + 0.05^2}} - \frac{1}{\sqrt{1.5^2 + \left(\frac{1}{20.2}\right)^2}} = 1.6411\times10^{-5} \quad \text{(Q3003.7)}$$

$$\frac{1}{2}\Delta |V_R| \Big|_{\Delta\omega = -0.1} = \frac{1}{\sqrt{1.5^2 + 0.05^2}} - \frac{1}{\sqrt{1.5^2 + \left(\frac{1}{19.8}\right)^2}} = -1.6911\times10^{-5} \quad \text{(Q3003.8)}$$

HAND☞ Question 3005 Derive, starting with Tellegen's theorem, the first-order sensitivity expression

$$-V^T \Delta Y^T \hat{V} \tag{Q3005.1}$$

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

Solution

Using the perturbed difference form of Tellegen's Theorem, we have

$$\Delta V_B^T \hat{I}_B - \Delta I_B^T \hat{V}_B = 0 \tag{Q3005.2}$$

Here we are only considering one element whose characteristic is given as

$$Y V = I \tag{Q3005.3}$$

Perturbing both sides of (Q3005.3) gives

$$\Delta I = \Delta Y V + Y \Delta V \tag{Q3005.4}$$

Substituting (Q3005.4) into (Q3005.2), we have

$$\ldots + \Delta V \hat{I} - (\Delta V^T Y^T + V^T \Delta Y^T)\hat{V} + \ldots = 0 \tag{Q3005.5}$$

$$\ldots + \Delta V \hat{I} - \Delta V^T Y^T \hat{V} - V^T \Delta Y^T \hat{V} + \ldots = 0 \tag{Q3005.6}$$

If we let

$$\hat{I} = Y^T \hat{V} \tag{Q3005.7}$$

that is we use Y^T to define the characteristic of the corresponding adjoint element. We obtain

$$\ldots + (-Y^T \Delta Y^T \hat{V}) + \ldots = 0 \tag{Q3005.8}$$

Hence the first-order sensitivity expression for such an element is

$$-V^T \Delta Y^T \hat{V} \tag{Q3005.9}$$

HAND☞ Question 3008 Consider a system of complex linear equations

$$YV = I \tag{Q3008.1}$$

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.

Solution

Let the real and the imaginary parts of V be represented by V_R and V_I, respectively, i.e.

$$V = V_R + jV_I \tag{Q3008.2}$$

and let

$$e(V_R, V_I) = YV - I \tag{Q3008.3}$$

denote the error function. Then the objective function for the least squares solution of $YV = I$ can be defined as

$$U(V_R, V_I) \triangleq e^T e^* = \sum_i |e_i|^2 \tag{Q3008.4}$$

where * stands for the conjugate of a complex number. The derivatives of U w.r.t. ϕ is

$$\begin{aligned}\frac{\partial U}{\partial \phi} &= \frac{\partial e^T}{\partial \phi} e^* + e^T \frac{\partial e^*}{\partial \phi} = 2Re\left\{\frac{\partial e^T}{\partial \phi} e^*\right\} \\ &= 2Re\left\{\frac{\partial (V^T Y^T - I^T)}{\partial \phi}(Y^* V^* - I^*)\right\} = 2Re\left\{\frac{\partial V^T}{\partial \phi} Y^T (Y^* V^* - I^*)\right\}\end{aligned} \tag{Q3008.5}$$

where Y and I are considered constant. Also, notice that

$$\frac{\partial V^T}{\partial V_R} = 1 \quad \text{and} \quad \frac{\partial V^T}{\partial V_I} = j1 \tag{Q3008.6}$$

where 1 is the identity matrix. Therefore, the gradient vector of U w.r.t. the real and imaginary parts of V is given by

$$\begin{bmatrix} \dfrac{\partial U}{\partial V_R} \\ \dfrac{\partial U}{\partial V_I} \end{bmatrix} = 2 \begin{bmatrix} Re\,\{Y^T(Y^*V^* - I^*)\} \\ Re\{jY^T(Y^*V^* - I^*)\} \end{bmatrix} = 2 \begin{bmatrix} Re\{Y^T(Y^*V^* - I^*)\} \\ -Im\{Y^T(Y^*V^* - I^*)\} \end{bmatrix} \tag{Q3008.7}$$

HAND☞ Question 3012 Derive from first principles an approach to finding $\frac{\partial \lambda}{\partial \boldsymbol{x}}$, where λ is an eigenvalue of the square matrix $\boldsymbol{A}$ whose coefficients are (in general) nonlinear functions of $\boldsymbol{x}$, i.e.,

$$\boldsymbol{A}\,\boldsymbol{y} = \lambda \boldsymbol{y} \tag{Q3012.1}$$

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

Solution

By definition, the eigenvalue must satisfy

$$A(\boldsymbol{x})\boldsymbol{y} = \lambda \boldsymbol{y} \tag{Q3012.2}$$

and

$$\boldsymbol{A}^T \boldsymbol{u} = \lambda \boldsymbol{u} \tag{Q3012.3}$$

Differentiating (Q3012.2) w.r.t. x_j, the jth component of $\boldsymbol{x}$, we have

$$\frac{\partial A(\boldsymbol{x})}{\partial x_j}\boldsymbol{y} + A(\boldsymbol{x})\frac{\partial \boldsymbol{y}}{\partial x_j} = \frac{\partial \lambda}{\partial x_j}\boldsymbol{y} + \lambda\frac{\partial \boldsymbol{y}}{\partial x_j} \tag{Q3012.4}$$

Multiplying both sides of (Q3012.4) by $\boldsymbol{u}^T$ gives

$$\boldsymbol{u}^T\frac{\partial A(\boldsymbol{x})}{\partial x_j}\boldsymbol{y} + \boldsymbol{u}^T A(\boldsymbol{x})\frac{\partial \boldsymbol{y}}{\partial x_j} = \boldsymbol{u}^T\frac{\partial \lambda}{\partial x_j}\boldsymbol{y} + \boldsymbol{u}^T\lambda\frac{\partial \boldsymbol{y}}{\partial x_j} \tag{Q3012.5}$$

(Q3012.5) can be rearranged as

$$\boldsymbol{u}^T\frac{\partial A(\boldsymbol{x})}{\partial x_j}\boldsymbol{y} + \boldsymbol{u}^T\left[A(\boldsymbol{x}) - \lambda 1\right]\frac{\partial \boldsymbol{y}}{\partial x_j} = \boldsymbol{u}^T\frac{\partial \lambda}{\partial x_j}\boldsymbol{y} \tag{Q3012.6}$$

Since $(\boldsymbol{A}(\boldsymbol{x}) - \lambda \boldsymbol{I})^T\boldsymbol{u} = \boldsymbol{0}$, we have

$$\boldsymbol{u}^T\frac{\partial A(\boldsymbol{x})}{\partial x_j}\boldsymbol{y} = \boldsymbol{u}^T\frac{\partial \lambda}{\partial x_j}\boldsymbol{y} \tag{Q3012.7}$$

Therefore,

$$\frac{\partial \lambda}{\partial x_j} = \frac{\boldsymbol{u}^T\frac{\partial A(\boldsymbol{x})}{\partial x_j}\boldsymbol{y}}{\boldsymbol{u}^T\boldsymbol{y}} \tag{Q3012.8}$$

HAND☞ Question 3019 Derive from first principles the adjoint element equation and sensitivity expression for a two-port characterized by

$$\begin{bmatrix} V_p \\ I_p \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} \qquad \text{(Q3019.1)}$$

Apply the result to the element shown in Fig. Q3019 to determine the sensitivity formulas w.r.t ϕ, where $Y_1 = \phi$ and $Z_2 = 0.5/\phi$.

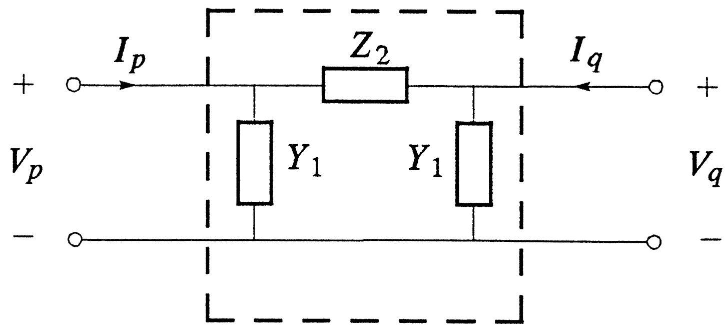

Fig. Q3019 A two-port circuit.

Solution

From the Tellegen's theorem, we have

$$\ldots + [\hat{I}_p \ -\hat{V}_p] \begin{bmatrix} V_p \\ I_p \end{bmatrix} + [\hat{I}_q \ \hat{V}_q] \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + \ldots = 0 \qquad \text{(Q3019.2)}$$

Differentiating (Q3019.1) w.r.t. ϕ gives

$$\ldots + [\hat{I}_p \ -\hat{V}_p] \frac{\partial}{\partial \phi} \begin{bmatrix} V_p \\ I_p \end{bmatrix} + [\hat{I}_q \ \hat{V}_q] \frac{\partial}{\partial \phi} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + \ldots = 0 \qquad \text{(Q3019.3)}$$

Using the branch relation characterized by (Q3019.1), we obtain

$$\frac{\partial}{\partial \phi} \begin{bmatrix} V_p \\ I_p \end{bmatrix} = \frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + \begin{bmatrix} A & B \\ C & D \end{bmatrix} \frac{\partial}{\partial \phi} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} \qquad \text{(Q3019.4)}$$

Substituting (Q3019.4) into (Q3019.3), we have

$$\ldots + [\hat{I}_p \ -\hat{V}_p] \frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + \left([\hat{I}_p \ -\hat{V}_p] \begin{bmatrix} A & B \\ C & D \end{bmatrix} + [\hat{I}_q \ \hat{V}_q] \right) \frac{\partial}{\partial \phi} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} + \ldots = 0 \qquad \text{(Q3019.5)}$$

If we define the adjoint element as

$$[\hat{I}_p \ -\hat{V}_p] \begin{bmatrix} A & B \\ C & D \end{bmatrix} + [\hat{I}_q \ \hat{V}_q] = 0 \tag{Q3019.6}$$

then the sensitivity expression is

$$[\hat{I}_p \ -\hat{V}_p] \frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} \tag{Q3019.7}$$

For the two-port circuit shown in Fig. Q3019, the parameters can be derived as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & \frac{1}{2\phi} \\ \frac{5\phi}{2} & \frac{3}{2} \end{bmatrix} \tag{Q3019.8}$$

Hence,

$$\frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{2\phi^2} \\ \frac{5}{2} & 0 \end{bmatrix} \tag{Q3019.9}$$

Finally

$$[\hat{I}_p \ -\hat{V}_p] \frac{\partial}{\partial \phi} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_q \\ -I_q \end{bmatrix} = -\frac{5}{2} \hat{V}_p V_q + \frac{1}{(2\phi^2)} \hat{I}_p I_q \tag{Q3019.10}$$

HAND☞ Question 3023 Consider the formula

$$G = \sum_{\substack{\text{voltage}\\ \text{sources}}} \hat{V}_i \nabla I_i - \sum_{\substack{\text{current}\\ \text{sources}}} \hat{I}_i \nabla V_i \qquad \text{(Q3023.1)}$$

where G is a vector of standard sensitivity expressions, i is the index of the sources and ∇ 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

$$\sum_{\substack{\text{terminating}\\ \text{resistors}}} \frac{|V_r|^2}{R_r} \qquad \text{(Q3023.2)}$$

where V_r is the response voltage and R_r is the terminating resistor. Draw the adjoint network and state the proper excitation.

Solution

The original six-port network and its corresponding adjoint network can be sketched as Fig. SQ3023.

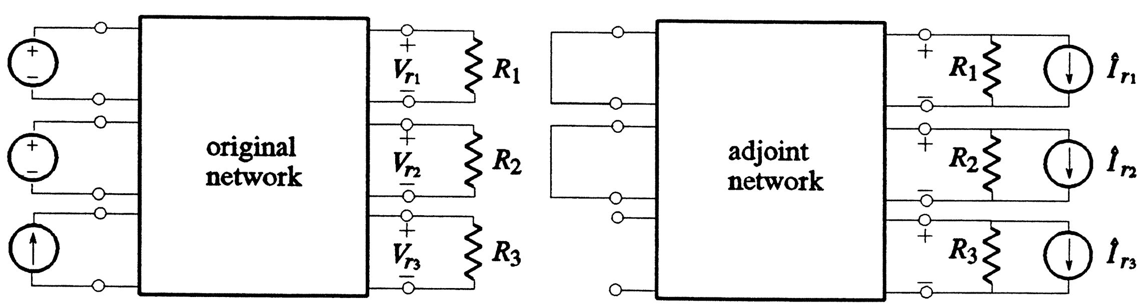

Fig. SQ3023 The original six-port network and its corresponding adjoint network.

Let

$$U = \sum_{\substack{\text{terminating}\\ \text{resistors}}} \frac{|V_r|^2}{R_r} \qquad \text{(Q3023.3)}$$

Following (Q2015.3), the gradient vector of U can be derived as

$$\nabla U = \sum_{\text{resistors}} \frac{1}{R_r} 2Re[V_r^* \nabla V_r] - \sum_{\text{resistors}} \frac{|V_r|^2}{R_r^2} \nabla R_r$$

$$= 2Re\left\{ \sum_{\text{resistors}} \left[\frac{V_r^*}{R_r} \nabla V_r \right] \right\} - \sum_{\text{resistors}} \frac{|V_r|^2}{R_r^2} \nabla R_r \qquad \text{(Q3023.4)}$$

From (Q3023.1) and Fig. SQ3023, we can obtain

$$G = -\sum_r \hat{I}_r \nabla V_r \qquad \text{(Q3023.5)}$$

if

$$\hat{I}_r = \frac{V_r^*}{R_r}$$

then

$$\nabla U = -2Re\{G\} - \sum_r \frac{|V_r|^2}{R_r^2} \nabla R_r \qquad \text{(Q3023.6)}$$

which shows the relationship between G and the gradient vector of U.

HAND☞ Question 3031 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 $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$ is a linear system in $\boldsymbol{y}$, $\boldsymbol{A}$ is a square matrix, the term y_i is the ith component of the column vector $\boldsymbol{y}$ and a_{jk} represents the $\{j, k\}$ element of $\boldsymbol{A}$. Discuss in detail the computational effort involved.

Solution

Let $\boldsymbol{u}_i$ denote a unit vector which has 1 in the ith entry and 0 elsewhere. Then

$$y_i = \boldsymbol{u}_i^T \boldsymbol{y} \tag{Q3031.1}$$

Since $\boldsymbol{b}$ is constant and

$$\boldsymbol{y} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{Q3031.2}$$

we have

$$\frac{\partial y_i}{\partial a_{jk}} = \boldsymbol{u}_i^T \frac{\partial \boldsymbol{y}}{\partial a_{kj}} = \boldsymbol{u}_i^T \frac{\partial \boldsymbol{A}^{-1}}{\partial a_{jk}} \boldsymbol{b} = -\boldsymbol{u}_i^T \boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial a_{kj}} \boldsymbol{A}^{-1} \boldsymbol{b} = -\hat{\boldsymbol{y}}^T \frac{\partial \boldsymbol{A}}{\partial a_{jk}} \boldsymbol{y} \tag{Q3031.3}$$

where

$$\hat{\boldsymbol{y}}^T = \boldsymbol{u}_i^T \boldsymbol{A}^{-1} \quad \text{or} \quad \boldsymbol{A}^T \hat{\boldsymbol{y}} = \boldsymbol{u}_i \tag{Q3031.4}$$

Substituting

$$\frac{\partial \boldsymbol{A}}{\partial a_{jk}} = \boldsymbol{u}_j \boldsymbol{u}_k^T \tag{Q3031.5}$$

into (Q3031.3) gives the first-order sensitivity of y_i w.r.t. a_{jk} as

$$\frac{\partial y_i}{\partial a_{jk}} = -\hat{\boldsymbol{y}}^T \boldsymbol{u}_j \boldsymbol{u}_k^T \boldsymbol{y} = -\hat{y}_j \, y_k \tag{Q3031.6}$$

The computational effort involved:

We solve for $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ from

$$\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$$
$$\boldsymbol{A}^T\hat{\boldsymbol{y}} = \boldsymbol{u}_i$$

This requires one LU factorization which needs $(\frac{n^3}{3} - \frac{n}{3})$ multiplications, and two forward and backward substitutions which need $(n^2 + n^2)$ multiplications. Therefore the total computational effort involved is

$$\frac{n^3}{3} + 2n^2 - \frac{n}{3} + 1$$

multiplications.

HAND☞ Question 3034 Consider the resistive network shown in Fig. Q3034, where G_1 = 1.5 S, G_2 = 2.5 S and i = 10 A. 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.

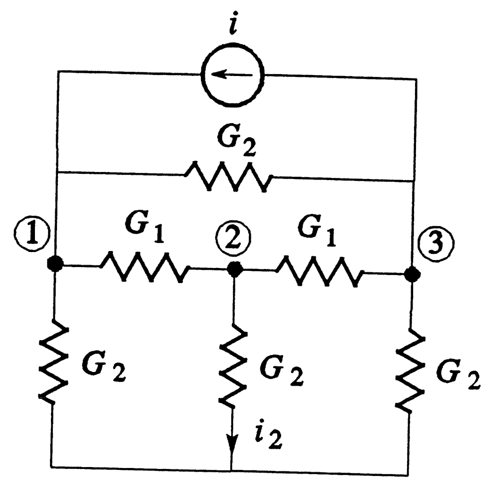

Fig. Q3034 A resistive network.

Solution

The nodal equations of the circuit can be written as

$$\boldsymbol{Y}\boldsymbol{v} = \boldsymbol{i} \tag{Q3034.1}$$

where

$$\boldsymbol{Y} = \begin{bmatrix} G_1 + 2G_2 & -G_1 & -G_2 \\ -G_1 & 2G_1 + G_2 & -G_1 \\ -G_2 & -G_1 & G_1 + 2G_2 \end{bmatrix} = \begin{bmatrix} 6.5 & -1.5 & -2.5 \\ -1.5 & 5.5 & -1.5 \\ -2.5 & -1.5 & 6.5 \end{bmatrix} \tag{Q3034.2}$$

and

$$\boldsymbol{i} = \begin{bmatrix} i \\ 0 \\ -i \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ -10 \end{bmatrix} \tag{Q3034.3}$$

The solution of (Q3034.1) is

$$\boldsymbol{v} = \begin{bmatrix} \frac{10}{9} \\ 0 \\ -\frac{10}{9} \end{bmatrix} \tag{Q3034.4}$$

From the circuit we can calculate

$$\frac{\partial i_2}{\partial \phi} = \frac{\partial (v_2 G_2)}{\partial \phi} = \frac{\partial v_2}{\partial \phi} G_2 + v_2 \frac{\partial G_2}{\partial \phi} \tag{Q3034.5}$$

Differentiating (Q3034.1) w.r.t. ϕ, we have

$$\frac{\partial \boldsymbol{Y}}{\partial \phi} \boldsymbol{v} + \boldsymbol{Y} \frac{\partial \boldsymbol{v}}{\partial \phi} = \frac{\partial \boldsymbol{i}}{\partial \phi} \Rightarrow \frac{\partial \boldsymbol{v}}{\partial \phi} \boldsymbol{Y}^{-1} \left(\frac{\partial \boldsymbol{i}}{\partial \phi} - \frac{\partial \boldsymbol{Y}}{\partial \phi} \boldsymbol{v} \right) \tag{Q3034.6}$$

Hence,

$$\frac{\partial v_2}{\partial \phi} = \hat{\boldsymbol{v}}^T \left(\frac{\partial \boldsymbol{i}}{\partial \phi} - \frac{\partial \boldsymbol{Y}}{\partial \phi} \boldsymbol{v} \right) \tag{Q3034.7}$$

where $\hat{\boldsymbol{v}}$ is solved from the adjoint network

$$\boldsymbol{Y}^T \hat{\boldsymbol{v}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{or} \quad \begin{bmatrix} 6.5 & -1.5 & -2.5 \\ -1.5 & 5.5 & -1.5 \\ -2.5 & -1.5 & 6.5 \end{bmatrix} \hat{\boldsymbol{v}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{Q3034.8}$$

The solution of (Q3034.8) is

$$\hat{\boldsymbol{v}} = K \begin{bmatrix} 3 \\ 8 \\ 3 \end{bmatrix}, \quad \text{where } K = \frac{36}{\det(Y)}$$

Then we can obtain the following results.

Case 1 $\phi = G_1$

$$\frac{\partial \boldsymbol{i}}{\partial G_1} = 0, \quad \frac{\partial V_2}{\partial G_1} = -\hat{\boldsymbol{v}}^T \frac{\partial \boldsymbol{Y}}{\partial G_1} \boldsymbol{v} = -\hat{\boldsymbol{v}}^T \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \boldsymbol{v} = 0$$

$$\Rightarrow \frac{\partial i_2}{\partial G_1} = 0$$

Case 2 $\phi = G_2$

$$\frac{\partial i}{\partial G_2} = 0, \quad \frac{\partial V_2}{\partial G_2} = -\hat{v}^T \frac{\partial Y}{\partial G_2} v = -\hat{v}^T \begin{bmatrix} 2 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix} v = 0$$

$$\Rightarrow \frac{\partial i_2}{\partial G_2} = 0$$

Case 3 $\phi = i$

$$\frac{\partial i}{\partial i} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \quad \frac{\partial v_2}{\partial i} = -\hat{v}^T \frac{\partial Y}{\partial i} v = -\hat{v}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} v = 0$$

$$\Rightarrow \frac{\partial i_2}{\partial i} = 0$$

We can also see directly from Fig. Q3034 that $V_2 \equiv 0$ regardless of changes in G_1, G_2 or i which gives $\frac{\partial v_2}{\partial G_1} = \frac{\partial v_2}{\partial G_2} = \frac{\partial v_2}{\partial i} = 0$. Therefore, $\frac{\partial i_2}{\partial G_1} = 0$, $\frac{\partial i_2}{\partial G_2} = 0$, $\frac{\partial i_2}{\partial i} = 0$.

Check

Analytically, we have

$$v = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \frac{10}{G_1 + G_3} \tag{Q3034.9}$$

Hence

$$v_2 \equiv 0$$

and

$$\frac{\partial i_2}{\partial \phi} = \frac{\partial v_2}{\partial \phi} G_2 + v_2 \frac{\partial G_2}{\partial \phi} \equiv 0$$

3.2.4 Nonlinear Networks

HAND☞ Question 4004 (p. 3-41)
HAND☞ Question 4008 (p. 3-43)

HAND☛ <u>Question 4004</u> Consider the resistor-diode network shown in Fig. SQ4004.1. Draw the corresponding companion network at the jth iteration for its DC solution. Write down the nodal equations at this iteration.

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

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

$$\lambda = \frac{1}{0.026} \; V^{-1}$$

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

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

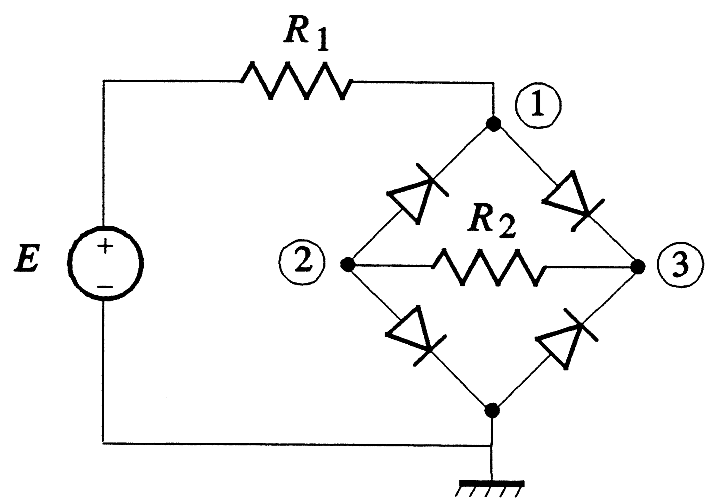

Fig. SQ4004.1 A resistor–diode network.

Solution

The corresponding companion network at the jth iteration is shown in Fig. SQ4004.2. The nodal equations at this iteration can be written as

$$\begin{bmatrix} 0.001 + G_1^j + G_2^j & -G_1^1 & -G_2^j \\ -G_1^j & 0.001 + G_1^j + G_3^j & -0.001 \\ -G_2^j & -0.001 & 0.001 + G_2^j + G_4^j \end{bmatrix} \begin{bmatrix} v_1^{j+1} \\ v_2^{j+1} \\ v_3^{j+1} \end{bmatrix}$$

$$= \begin{bmatrix} 0.01 + (i_{d_1}^j - G_1^j v_{d_1}^j) - (i_{d_2}^j - G_{d_2}^j v_{d_2}^j) \\ -(i_{d_1}^j - G_1^j v_{d_1}^j) - (i_{d_3}^j - G_3^j v_{d_3}^j) \\ (i_{d_2}^j - G_2^j v_{d_2}^j) - (i_{d_4}^j - G_4^j v_{d_4}^j) \end{bmatrix} \qquad \text{(Q4004.1)}$$

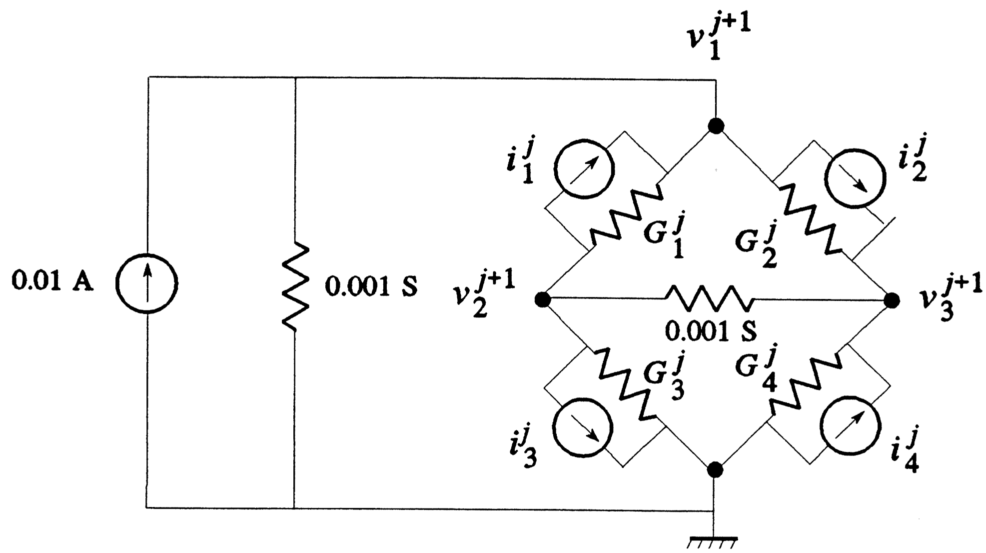

Fig. SQ4004.2 The corresponding companion network at the jth iteration of Fig. SQ4004.1.

where

$$v_{d_1}^j = v_2^j - v_1^j, \quad v_{d_2}^j = v_1^j - v_3^j, \quad v_{d_3}^j = v_2^j, \quad v_{d_4}^j = -v_3^j \tag{Q4004.2}$$

and

$$\left.\begin{aligned} i_{d_k}^j &= I_S(e^{\lambda v_{d_k}^j} - 1) \\ G_k^j &= \lambda I_S\, e^{\lambda v_{d_k}^j} \end{aligned}\right\} \quad k = 1, 2, 3, 4 \tag{Q4004.3}$$

HAND☞ Question 4008 Consider the nonlinear circuit shown in Fig. SQ4008.1, 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 jth 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 jth iteration and state the corresponding nodal equations.

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

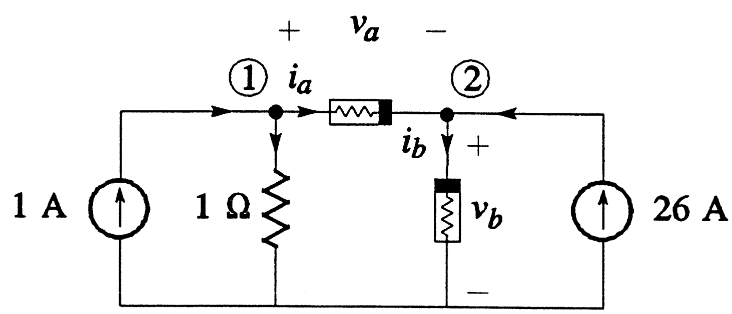

Fig. SQ4008.1 Nonlinear circuit example.

Solution

(a) According to KCL, the nodal equations at nodes 1 and 2 can be written as

$$\begin{aligned} v_1 + 2(v_1 - v_2)^3 - 1 &= 0 \\ v_2^3 + 10\, v_2 - 2(v_1 - v_2)^3 - 26 &= 0 \end{aligned} \tag{Q4008.1}$$

If we let

$$\begin{aligned} f_1 &= v_1 + 2(v_1 - v_2)^3 - 1 \\ f_2 &= v_2^3 + 10\, v_2 - 2(v_1 - v_2)^3 - 26 \end{aligned} \tag{Q4008.2}$$

then the Jacobian matrix at the jth iteration is

$$J^j = \begin{bmatrix} \dfrac{\partial f_1}{\partial v_1} & \dfrac{\partial f_1}{\partial v_2} \\ \dfrac{\partial f_2}{\partial v_1} & \dfrac{\partial f_2}{\partial v_2} \end{bmatrix}^j = \begin{bmatrix} 1 + 6(v_1 - v_2)^2 & -6(v_1 - v_2)^2 \\ -6(v_1 - v_2)^2 & 3v_2^2 + 10 + 6(v_1 - v_2)^2 \end{bmatrix}^j \tag{Q4008.3}$$

and the linearized form of the nodal equations at the jth iteration of the Newton algorithm can be written as

$$J^j(v^{j+1} - v^j) = -f^j \tag{Q4008.4}$$

(b) The starting point is $v^0 = [2\ 1]^T$. From (Q4008.2) and (Q4008.3) we have

$$f^0 = \begin{bmatrix} 3 \\ -17 \end{bmatrix}, \quad J^0 = \begin{bmatrix} 7 & -6 \\ -6 & 19 \end{bmatrix} \tag{Q4008.5}$$

Substituting v^0, f^0 and J^0 into (Q4008.4) and solving the resulting equations, we obtain

$$v^1 = \begin{bmatrix} 2.46 \\ 2.04 \end{bmatrix} \tag{Q4008.6}$$

In the same way, we can obtain

$$v^2 = \begin{bmatrix} 1.60 \\ 1.88 \end{bmatrix} \tag{Q4008.7}$$

(c) The companion network at the jth iteration is shown in Fig. SQ4008.2.

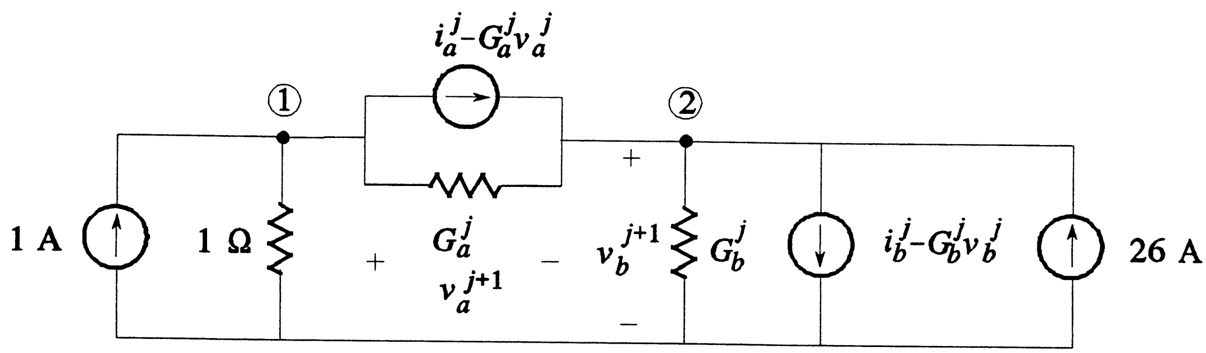

Fig. SQ4008.2 The companion network.

Where

$$i_a^j = 2(v_a^3)^j \tag{Q4008.8}$$

$$i_b^j = (v_b^3)^j + (10\ v_b)^j \tag{Q4008.9}$$

$$G_a^j = 6(v_a^2)^j \tag{Q4008.10}$$

$$G_b^j = (3\ v_b^2)^j + 10 \tag{Q4008.11}$$

$$v_a = v_1 - v_2, \quad v_b = v_2 \tag{Q4008.12}$$

The corresponding nodal equations are

$$\begin{bmatrix} 1 + G_a^j & -G_a^j \\ -G_a^j & G_a^j + G_b^j \end{bmatrix} \begin{bmatrix} v_1^{j+1} \\ v_2^{j+1} \end{bmatrix} = \begin{bmatrix} 1 - (i_a^j - G_a^j v_a^j) \\ (i_a - G_a^j v_a^j) - (i_b^j - G_b^j v_b^j) + 26 \end{bmatrix} \tag{Q4008.13}$$

(d) Let $v^0 = [1.60 \;\; 1.88]^T$. Using (Q4008.13) with (Q4008.9)-(Q4008.12) we can obtain

$$v^1 = \begin{bmatrix} 1.23 \\ 1.89 \end{bmatrix} \text{ and } v^2 = \begin{bmatrix} 1.32 \\ 1.89 \end{bmatrix} \tag{Q4008.14}$$

3.2.5 One-Dimensional Search Methods

3.2.6 Tolerances and Worst-Case Analysis

HAND☞ Question 6007 (p. 3-47)

HAND☞ Question 6007 Consider the voltage divider shown in Fig. SQ6007.1. Deriving all formulas from first principles, use the adjoint network method to calculate $\frac{\partial T}{\partial R_1}$ and $\frac{\partial T}{\partial R_2}$, given

$$T = \frac{V_2}{V_1}, \quad R_1 = 2\ \Omega, \quad R_2 = 1.5\ \Omega \tag{Q6007.1}$$

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 ±5% and on R_2 is ±10%, estimate the extreme values of T using first partial derivatives. Check the results by direct calculation.

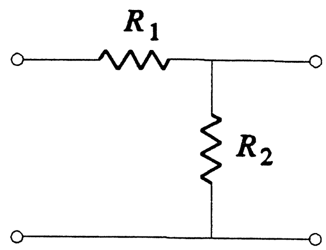

Fig. SQ6007.1 Voltage divider circuit.

Solution

Since the circuit is linear, the transfer function is independent of the numerical value of a V_1. For convenience, assume $V_1 = 1$, therefore $T = \frac{V_2}{1} = V_2$, and we want $\frac{\partial V_2}{\partial R_1}, \frac{\partial V_2}{\partial R_2}$. The original network and its assumed corresponding adjoint network are shown in Fig. SQ6007.2.

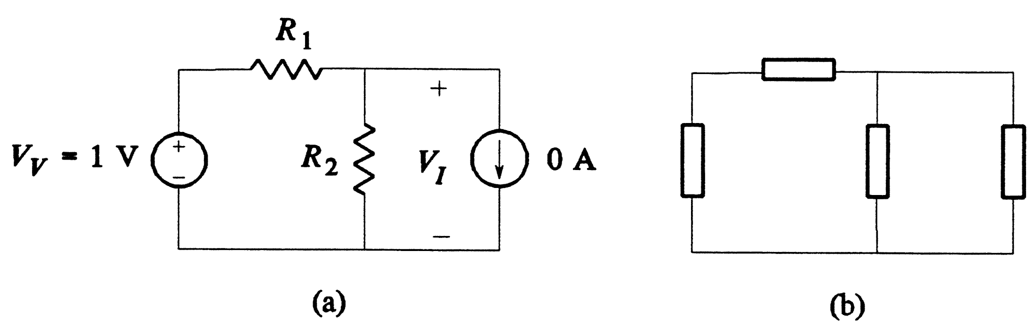

Fig. SQ6007.2 (a) the original network and (b) assumed adjoint network.

From Tellegen's theorem, we have

$$\frac{\partial V_V}{\partial \phi}\hat{I}_V + \frac{\partial V_{R_1}}{\partial \phi}\hat{I}_{R_1} + \frac{\partial V_{R_2}}{\partial \phi}\hat{I}_{R_2} + \frac{\partial V_I}{\partial \phi}\hat{I}_I - \frac{\partial I_V}{\partial \phi}\hat{V}_V$$
$$- \frac{\partial I_{R_I}}{\partial \phi}\hat{V}_{R_1} - \frac{\partial I_{R_2}}{\partial \phi}\hat{V}_{R_2} - \frac{\partial I_I}{\partial \phi}\hat{V}_I = 0 \qquad \text{(Q6007.2)}$$

Since V_V and I_I are constants, we have

$$\frac{\partial V_V}{\partial \phi} = 0 \quad \text{and} \quad \frac{\partial I_I}{\partial \phi} = 0 \qquad \text{(Q6007.3)}$$

Let

$$\hat{I}_I = -1 \quad \text{and} \quad \hat{V}_V = 0 \qquad \text{(Q6007.4)}$$

Then from (Q6007.2) we can obtain

$$\frac{\partial V_I}{\partial \phi} = \frac{\partial V_{R_1}}{\partial \phi}\hat{I}_{R_1} + \frac{\partial V_{R_2}}{\partial \phi}\hat{I}_{R_2} - \frac{\partial I_{R_1}}{\partial \phi}\hat{V}_{R_1} - \frac{\partial I_{R_2}}{\partial \phi}\hat{V}_{R_2} \qquad \text{(Q6007.5)}$$

Use

$$I_{R_1}R_1 = V_{R_1} \quad \text{and} \quad I_{R_2}R_2 = V_{R_2} \qquad \text{(Q6007.6)}$$

We have

$$\frac{\partial V_I}{\partial \phi} = \left(R_1\frac{\partial I_{R_1}}{\partial \phi} + \frac{\partial R_1}{\partial \phi}I_{R_1}\right)\hat{I}_{R_1} + \left(R_2\frac{\partial I_{R_2}}{\partial \phi} + \frac{\partial R_2}{\partial \phi}I_{R_2}\right)\hat{I}_{R_2} - \frac{\partial I_{R_1}}{\partial \phi}\hat{V}_{R_1} - \frac{\partial I_{R_2}}{\partial \phi}\hat{V}_{R_2}$$
$$= \left(R_1\hat{I}_{R_1} - \hat{V}_{R_1}\right)\frac{\partial I_{R_1}}{\partial \phi} + \left(R_2\hat{I}_{R_2} - \hat{V}_{R_2}\right)\frac{\partial I_{R_2}}{\partial \phi} + \frac{\partial R_1}{\partial \phi}I_{R_1}\hat{I}_{R_1} + \frac{\partial R_2}{\partial \phi}I_{R_2}\hat{I}_{R_2} \qquad \text{(Q6007.7)}$$

Let

$$R_1\hat{I}_{R_1} = \hat{V}_{R_1} \quad \text{and} \quad R_2\hat{I}_{R_2} = \hat{V}_{R_2} \qquad \text{(Q6007.8)}$$

Then

$$\frac{\partial V_I}{\partial \phi} = \frac{\partial R_1}{\partial \phi}I_{R_1}\hat{I}_{R_1} + \frac{\partial R_2}{\partial \phi}I_{R_2}\hat{I}_{R_2} \qquad \text{(Q6007.9)}$$

and the adjoint is completely defined as Fig. SQ6007.3.

For $\phi = R_1$

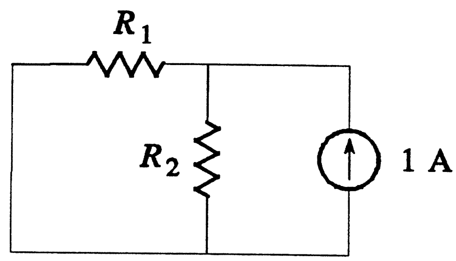

Fig. SQ6007.3 The defined adjoint network.

$$\frac{\partial V_1}{\partial R_1} = \frac{\partial V_2}{\partial R_1} = \frac{\partial T}{\partial R_1} = I_{R_1} \hat{I}_{R_1} \tag{Q6007.10}$$

For $\phi = R_2$

$$\frac{\partial V_1}{\partial R_2} = \frac{\partial V_2}{\partial R_2} = \frac{\partial T}{\partial R_2} = I_{R_2} \hat{I}_{R_2} \tag{Q6007.11}$$

Analyzing the original and the adjoint networks we have

$$I_{R_1} = I_{R_2} = \frac{1}{R_1 + R_2}, \quad \hat{I}_{R_1} = \frac{-R_2}{R_1 + R_2}, \quad \hat{I}_{R_2} = \frac{R_1}{R_1 + R_2} \tag{Q6007.12}$$

Hence

$$\frac{\partial T}{\partial R_1} = \frac{-R_2}{(R_1 + R_2)^2} \quad \text{and} \quad \frac{\partial T}{\partial R_2} = \frac{R_1}{(R_1 + R_2)^2} \tag{Q6007.13}$$

We can check this result by direct differentiation. From the circuit in Fig. SQ6007.2(a), we can obtain

$$T = \frac{R_2}{R_1 + R_2} \tag{Q6007.14}$$

Differentiating (Q6007.14) w.r.t. R_1 and R_2, respectively, gives

$$\frac{\partial T}{\partial R_1} = \frac{-R_2}{(R_1 + R_2)^2}, \quad \frac{\partial T}{\partial R_2} = \frac{R_1}{(R_1 + R_2)^2} \tag{Q6007.15}$$

which is identical to (Q6007.13). Numerically,

$$\frac{\partial T}{\partial R_1} = -0.1224, \quad \frac{\partial T}{\partial R_2} = 0.1633 \tag{Q6007.16}$$

For a quadratic function

$$U = a\phi^2 + b\phi + c \tag{Q6007.17}$$

By perturbing ϕ with $\pm\ \Delta\phi$, we have

$$U(\phi + \Delta\phi) = a(\phi + \Delta\phi)^2 + b(\phi + \Delta\phi) + c \qquad (Q6007.18)$$

and

$$U(\phi - \Delta\phi) = a(\phi - \Delta\phi)^2 + b(\phi - \Delta\phi) + c \qquad (Q6007.19)$$

Then

$$\begin{aligned}\frac{U(\phi + \Delta\phi) - U(\phi - \Delta\phi)}{2\Delta\phi} &= \frac{a[\phi^2 + (\Delta\phi)^2 + 2\phi\Delta\phi - \phi^2 - (\Delta\phi)^2 + 2\phi\Delta\phi]}{2\Delta\phi} \\ &= \frac{4a\phi\Delta\phi + 2b\Delta\phi}{2\Delta\phi} = 2a\phi + b\end{aligned} \qquad (Q6007.20)$$

Differentiating (Q6007.17) w.r.t. ϕ, we have

$$\frac{\partial U}{\partial \phi} = 2a\phi + b \qquad (Q6007.21)$$

Comparing (Q6007.20) with (Q6007.21) gives, for quadratic equations

$$\frac{\partial U}{\partial \phi} = \frac{U(\phi + \Delta\phi) - U(\phi - \Delta\phi)}{2\Delta\phi} \qquad (Q6007.22)$$

For a general function

$$\frac{\partial U}{\partial \phi} \approx \frac{U(\phi + \Delta\phi) - U(\phi - \Delta\phi)}{2\Delta\phi} \qquad (Q6007.23)$$

if $\Delta\phi$ is small.

In this question

$$T(R_1, R_2) = \frac{R_2}{R_1 + R_2} \qquad (Q6007.24)$$

By perturbation we have

$$\begin{aligned}A &= \frac{T(R_1 + \Delta R_1, R_2) - T(R_1 - \Delta R_1, R_2)}{2\Delta R_1} \\ &= \frac{\dfrac{R_2}{R_2 + R_1 + \Delta R_1} - \dfrac{R_2}{R_2 + R_1 - \Delta R_1}}{2\Delta R_1} \\ &= \frac{-R_2}{(R_2 + R_1)^2 - (\Delta R_1)^2}\end{aligned} \qquad (Q6007.25)$$

$$B = \frac{T(R_1, R_2 + \Delta R_2) - T(R_1, R_2 - \Delta R_2)}{2\Delta R_2} = \frac{R_1}{(R_2 + R_1)^2 - (\Delta R_2)^2} \qquad \text{(Q6007.26)}$$

If ΔR_1 and ΔR_2 are small, i.e., $(\Delta R_1)^2 \approx 0$ and $(\Delta R_1)^2 \approx 0$, then

$$A \approx \frac{-R_2}{(R_1 + R_2)^2} = \frac{\partial T}{\partial R_1} \quad \text{and} \quad B \approx \frac{R_1}{(R_1 + R_2)^2} = \frac{\partial T}{\partial R_2} \qquad \text{(Q6007.27)}$$

With tolerances ±5% on R_1 and ±10% on R_2 we have

$$R_1 = R_1 \pm 0.05 R_1$$
$$R_2 = R_2 \pm 0.1 R_2$$

The first-order change of T can be written as

$$\Delta T = \frac{\partial T}{\partial R_1}\Delta R_1 + \frac{\partial T}{\partial R_2}\Delta R_2 \qquad \text{(Q6007.28)}$$

The two extreme changes of T are

$$(\Delta T)_+ = -0.1224 \times (-0.1) + (0.1633) \times 0.15 = 0.0367$$

$$(\Delta T)_- = -0.1224 \times (0.1) + (0.1633) \times (-0.15) = -0.0367$$

At the nominal point

$$T = \frac{1.5}{3.5} = 0.4286$$

Therefore, the two extreme values of T are estimated as

$$T_{max} = 0.4286 + 0.0367 = 0.4653$$
$$T_{min} = 0.4286 - 0.0367 = 0.3919$$

By direct calculation, we have

$$T_{max} = \frac{1.65}{1.65 + 1.9} = 0.4648$$

$$T_{min} = \frac{1.35}{1.35 + 2.1} = 0.3913$$

which are very close to the estimated values.

3.2.7 State Equations

3.2.8 Applications

3.2.9 Various

Chapter 4

SOLUTIONS USING OSA90

4.1 INTRODUCTION

OSA90/hope is a state-of-the-art CAE software system created by Optimization System Associates Inc., Dundas, Ontario, Canada. OSA90/hope offers state-of-the-art optimization, powerful circuit simulation, sophisticated mathematics, intelligent connectivity through Datapipe, all in a friendly environment. OSA90/hope can be used to solve a variety of problems from numerical mathematics to practical engineering design.

In this chapter we will demonstrate how OSA90/hope is used to solve some of the problems of Chapter 2. The input or circuit files, numerical results and plots of responses are given. For details of OSA90/hope, refer to the OSA90/hope User's Manual.

4.2 SOLUTIONS

4.2.1 Algorithm Development

OSA☛ Question 1011 Use OSA90/hope to calculate the frequency response $V_2(j\omega)/V_1(j\omega)$ for the circuit of Fig. Q1010. The values of the elements are $L_1 = L_2 = 2$ H, $C_1 = C_2 = 0.5$ F. Calculate $V_2(j\omega)/V_1(j\omega)$ for ω = 1.7-3.2 rad/s with an incremental step of 0.1 rad/s. Display your results numerically and graphically. (See Question 1010.)

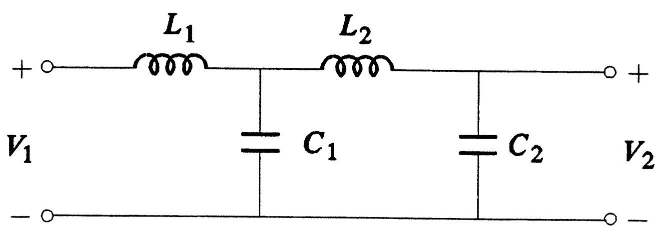

Fig. Q1010 LC ladder network.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 1011

```
! File name: Q1011.ckt
! Circuit file for solving Question 1011

#define MVINP 1V        ! Magnitude of VINP
#define PVINP 0DEG      ! Phase of VINP

Control
   Non_Microwave_Units;
End

Model
   ! Assign values to the circuit elements
   L1 = 2H;
   L2 = 2H;
   C1 = 0.5F;
   C2 = 0.5F;
   Omega = 1;    ! Initialize Omega

   ! Define the circuit
   VSOURCE 1 0  NAME=INP V=MVINP PHASE=PVINP;    ! Source VINP
   IND     1 2  L = L1;
   IND     2 3  L = L2;
   CAP     2 0  C = C1;
   CAP     3 0  C = C2;
   VLABEL  3 0  NAME = V2;            ! Output
   CIRCUIT NAME = Q1011;

   ! Calculate voltage gain for output
   Gain  = MV2[1] / MVINP;     ! Gain  = Output / Source
   Phase = PV2[1] - PVINP;     ! Phase = Output - Source
End

! Define sweep parameter and output responses

Sweep
   HB: Omega: from 1.7 to 3.2 step 0.1 FREQ = (1HZ * Omega / (2 * PI)) Gain Phase
   {Xsweep Title="Frequency Response"
    X_title = "Omega (rad/sec)"  X=Omega
    Y_title = "Gain"  Y = Gain.blue & Gain.red.dot}
```

```
   {Xsweep Title="Frequency Response"
    X_title = "Omega (rad/sec)"  X=Omega
    Y_title = "Phase" Y = Phase.yellow & Phase.blue.dot
    Ymin = -1e-30   Ymax = 1e-30   NYticks = 6};
End

! Define report block

Report
       Gain versus Omega

   --------------------------
     Omega             Gain
   --------------------------
${$%2.1f$       $Omega$ $%6.5f$          $Gain$ }$
   --------------------------


       Phase versus Omega

   --------------------------
     Omega            Phase
   --------------------------
${$%2.1f$       $Omega$ $%6.5f$          $Phase$ }$
   --------------------------
End
```

By selecting the "Generate report" option of OSA90/hope we can generate a report for this question as follows.

Report for Question 1011

Gain versus Omega

Omega	Gain
1.7	1.46606
1.8	0.56256
1.9	0.31230
2.0	0.20000
2.1	0.13854
2.2	0.10095
2.3	0.07625
2.4	0.05918
2.5	0.04692
2.6	0.03785
2.7	0.03098
2.8	0.02568
2.9	0.02151
3.0	0.01818
3.1	0.01550
3.2	0.01331

Phase versus Omega

Omega	Phase
1.7	0.00000
1.8	0.00000
1.9	0.00000
2.0	0.00000
2.1	0.00000
2.2	0.00000
2.3	0.00000
2.4	0.00000
2.5	0.00000
2.6	0.00000
2.7	0.00000
2.8	0.00000
2.9	0.00000
3.0	0.00000
3.1	0.00000
3.2	0.00000

Using the "Display" option of OSA90/hope to simulate the circuit, we can obtain the graphical results as shown in Fig. SQ1011.1 and Fig. SQ1011.2 .

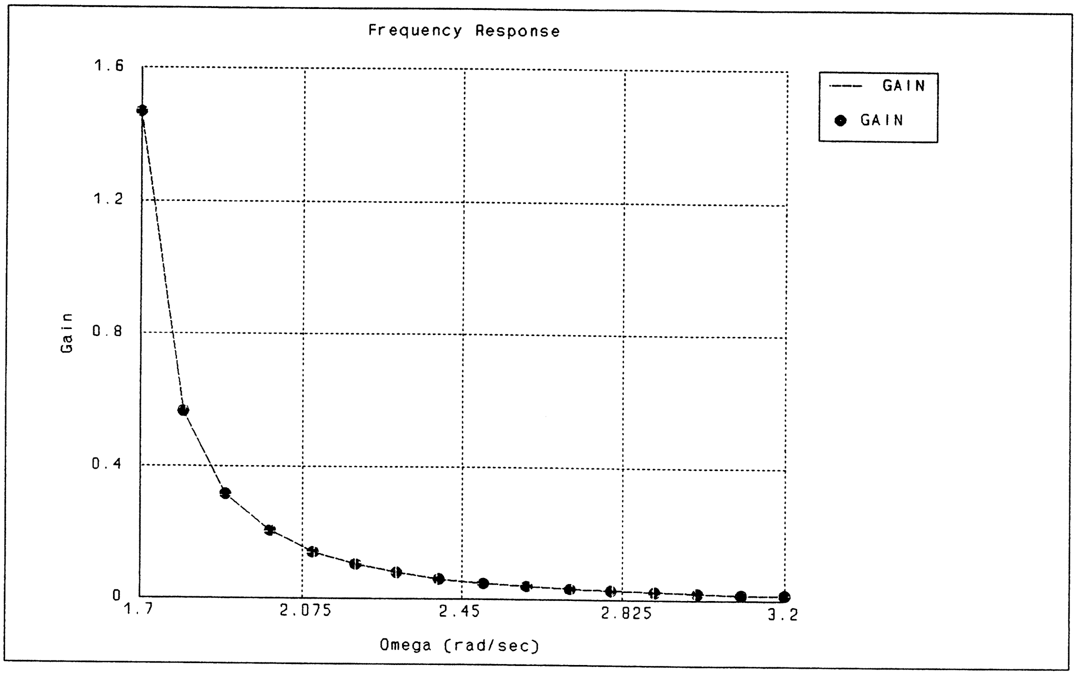

Fig. SQ1011.1 Sweep of gain versus frequency.

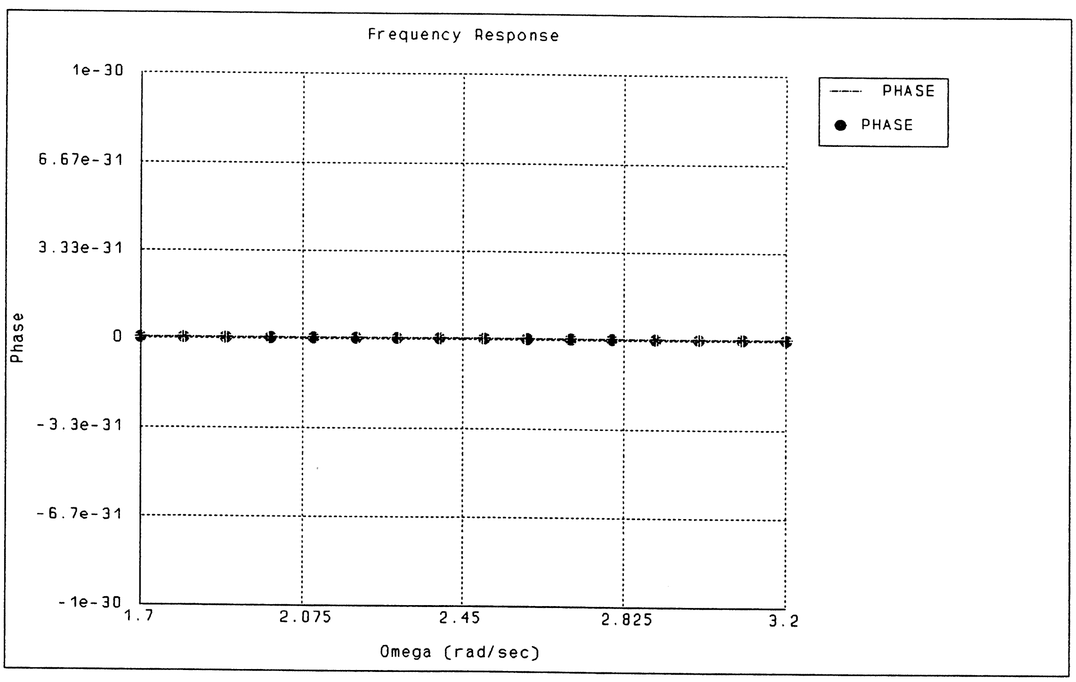

Fig. SQ1011.2 Sweep of phase versus frequency.

OSA☞ Question 1019 Use OSA90/hope to calculate all the branch voltages and currents in the resistive ladder network of Fig. Q1018 with $n = 8$, $R_1 = R_3 = R_5 = R_7 = 3\ \Omega$, $R_2 = R_4 = R_6 = R_8 = 1\ \Omega$. Calculate the voltages and currents for $V_g = 0, 0.5, 1, 1.5, 2, 2.5$ and 3 V. (See Question 1018.)

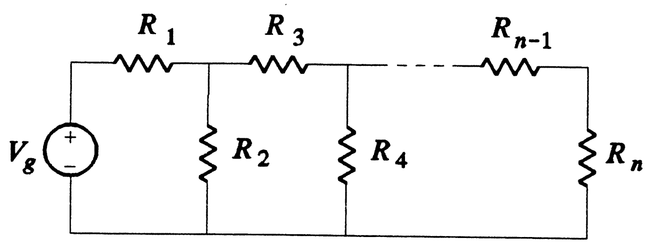

Fig. Q1018 Resistive ladder network.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 1019

```
!  File name: Q1019.ckt
!  Circuit file for solving Question 1019

Model
   ! Assign values to circuit elements
   R1 = 3OH;
   R2 = 1OH;
   R3 = 3OH;
   R4 = 1OH;
   R5 = 3OH;
   R6 = 1OH;
   R7 = 3OH;
   R8 = 1OH;
   Vg = 1V;                 ! Initialize Vg

   ! Define the circuit
   VSOURCE 1 0 NAME = VIN VDC = Vg;
   RES 1 2  R = R1;
   RES 2 0  R = R2;
   RES 2 3  R = R3;
   RES 3 0  R = R4;
   RES 3 4  R = R5;
   RES 4 0  R = R6;
   RES 4 5  R = R7;
   RES 5 0  R = R8;

   ! Using VLABEL to define voltages
   VLABEL 1 2  NAME=V1;
   VLABEL 2 0  NAME=V2;
   VLABEL 2 3  NAME=V3;
   VLABEL 3 0  NAME=V4;
   VLABEL 3 4  NAME=V5;
   VLABEL 4 0  NAME=V6;
   VLABEL 4 5  NAME=V7;
   VLABEL 5 0  NAME=V8;
```

```
   ! Calculate branch currents from branch voltages
   I1 = V1 / R1;
   I2 = V2 / R2;
   I3 = V3 / R3;
   I4 = V4 / R4;
   I5 = V5 / R5;
   I6 = V6 / R6;
   I7 = V7 / R7;
   I8 = V8 / R8;
   CIRCUIT NAME = Q1019;
End

! Define sweep parameters and output responses

Sweep
  DC: Vg: from 0 to 3 step 0.5 V1 V2 V3 V4 V5 V6 V7 V8
  {Xsweep Title = "Sweeps of branch voltages versus input voltage Vg"
   X = Vg  X_title = "Input voltage Vg (V)"  Y_title="Branch voltages (V)"
   Y = V1.white.circle & V2.Red.Triangle & V3.Yellow.Square & V4.Green.Dot
   & V5.LightBlue.diamond & V6.Pink.X & V7.Cream.Cross & V8.Brown.V
   & V1.white & V2.Red & V3.Yellow & V4.Green & V5.LightBlue & V6.Pink
   & V7.Cream & V8.Brown};

  DC: Vg: from 0 to 3 step 0.5 I1 I2 I3 I4 I5 I6 I7 I8
  {Xsweep Title = "Sweeps of branch currents versus input voltage Vg"
   X = Vg  X_title = "Input voltage Vg (V)"  Y_title="Branch Currents (A)"
   Y = I1.white.circle & I2.Red.Triangle & I3.Yellow.Square & I4.Green.Dot
   & I5.LightBlue.diamond & I6.Pink.X & I7.Cream.Cross & I8.Brown.V
   & I1.white & I2.Red & I3.Yellow & I4.Green & I5.LightBlue & I6.Pink
   & I7.Cream & I8.Brown};
End

! Define report block

Report
                   Branch Voltages Versus Input Voltage Vg

   ---------------------------------------------------------------------
     Vg      V1      V2      V3      V4      V5      V6      V7      V8
   ---------------------------------------------------------------------
${$%2.1f$     $Vg$ $%5.4f$ $V1$ $%5.4f$ $V2$ $%5.4f$ $V3$ $%5.4f$ $V4$ $%5.4f$ $V5$ $%5.4f$ $V6$ $%5.4f$ $V7$
$%5.4f$ $V8$}$
   ---------------------------------------------------------------------


                   Branch Currents Versus Input Voltage Vg

   ---------------------------------------------------------------------
     Vg      I1      I2      I3      I4      I5      I6      I7      I8
   ---------------------------------------------------------------------
${$%2.1f$     $Vg$ $%5.4f$ $I1$ $%5.4f$ $I2$ $%5.4f$ $I3$ $%5.4f$ $I4$ $%5.4f$ $I5$ $%5.4f$ $I6$ $%5.4f$ $I7$
$%5.4f$ $I8$}$
   ---------------------------------------------------------------------
End
```

There are two sweep sets in the circuit file, one for branch voltage display and the other for branch current display. Both of them are included in the report block. The report generated with this circuit files follows.

Report for Question 1019

Branch Voltages Versus Input Voltage Vg

Vg	V1	V2	V3	V4	V5	V6	V7	V8
0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.5	0.3956	0.1044	0.0826	0.0218	0.0172	0.0046	0.0034	0.0011
1.0	0.7913	0.2087	0.1651	0.0436	0.0344	0.0092	0.0069	0.0023
1.5	1.1869	0.3131	0.2477	0.0654	0.0516	0.0138	0.0103	0.0034
2.0	1.5826	0.4174	0.3303	0.0872	0.0688	0.0183	0.0138	0.0046
2.5	1.9782	0.5218	0.4128	0.1089	0.0860	0.0229	0.0172	0.0057
3.0	2.3739	0.6261	0.4954	0.1307	0.1032	0.0275	0.0206	0.0069

Branch Currents Versus Input Voltage Vg

Vg	I1	I2	I3	I4	I5	I6	I7	I8
0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.5	0.1319	0.1044	0.0275	0.0218	0.0057	0.0046	0.0011	0.0011
1.0	0.2638	0.2087	0.0550	0.0436	0.0115	0.0092	0.0023	0.0023
1.5	0.3956	0.3131	0.0826	0.0654	0.0172	0.0138	0.0034	0.0034
2.0	0.5275	0.4174	0.1101	0.0872	0.0229	0.0183	0.0046	0.0046
2.5	0.6594	0.5218	0.1376	0.1089	0.0287	0.0229	0.0057	0.0057
3.0	0.7913	0.6261	0.1651	0.1307	0.0344	0.0275	0.0069	0.0069

The sweeps of branch voltages and currents versus input voltage V_g are shown in Fig. SQ1019.1 and Fig. SQ1019.2, respectively.

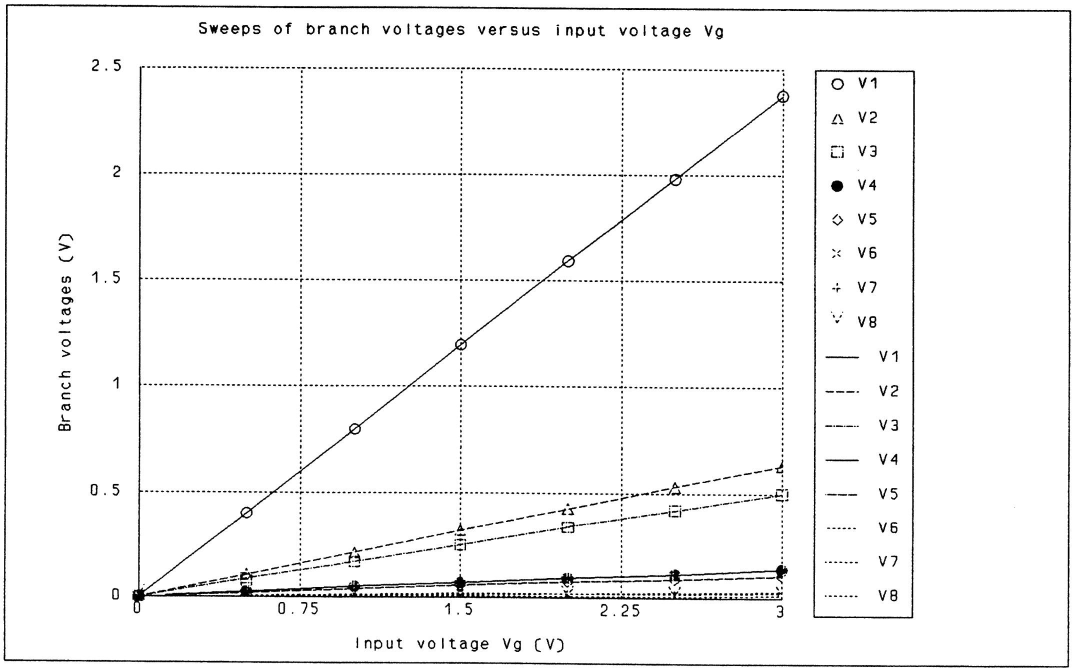

Fig. SQ1019.1 Sweeps of branch voltages versus input voltage V_g.

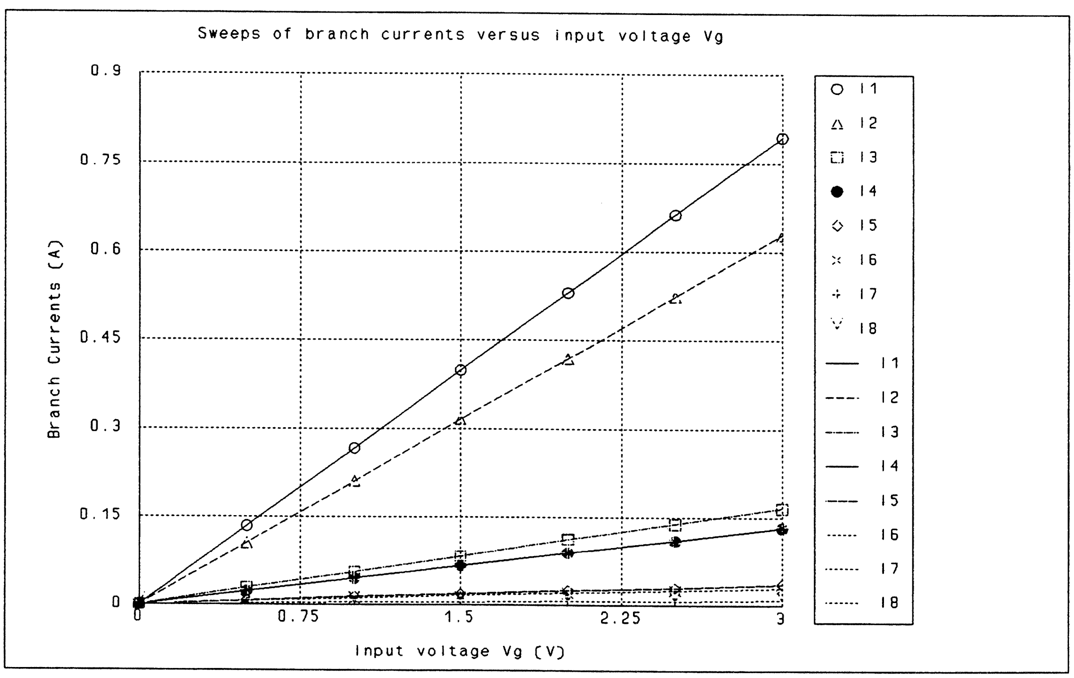

Fig. SQ1019.2 Sweeps of branch currents versus input voltage V_g.

OSA☛ Question 1021 Use OSA90/hope to calculate the input resistance of the circuit of Question 1018. Calculate the input resistances for $R_1 = 1, 2, 3, 4$ and $5\ \Omega$. Let $n = 8$, $R_3 = R_5 = R_7 = 3\ \Omega$ and $R_2 = R_4 = R_6 = R_8 = 1\ \Omega$. (See Question 1020.)

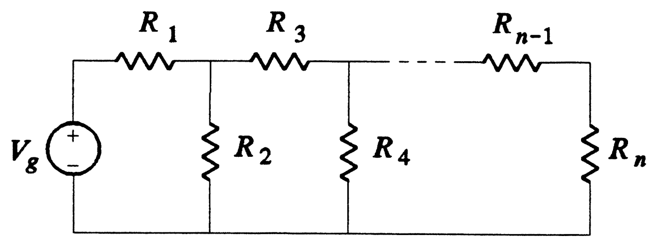

Fig. Q1018 Resistive ladder network.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 1021

```
! File name: Q1021.ckt
! Circuit file for solving Question 1021

Model
   ! Assign values to circuit elements
   R1 = 3OH;
   R2 = 1OH;
   R3 = 3OH;
   R4 = 1OH;
   R5 = 3OH;
   R6 = 1OH;
   R7 = 3OH;
   R8 = 1OH;
   Vg = 1V;

   ! Define the circuit

   VSOURCE 1 0 NAME = INP VDC = Vg;
   RES 1 2 R = R1;
   RES 2 0 R = R2;
   RES 2 3 R = R3;
   RES 3 0 R = R4;
   RES 3 4 R = R5;
   RES 4 0 R = R6;
   RES 4 5 R = R7;
   RES 5 0 R = R8;
   VLABEL 2 0 NAME = V2;
   CIRCUIT NAME = Q1021;

   ! Calculate the input resistance
   InputRes = Vg / IINP_DC;
End

! Define the sweep parameter and output response

Sweep
  DC: R1: from 1 to 5 step 1 InputRes
  {Xsweep  Title = "Sweep of Input Resistance versus R1"
   X = R1  X_title = "R1 (Ohms)"
   Y = InputRes.red  Y_title = "Input Resistance (Ohms)"};
```

```
End

! Define the report block

Report
     Input Resistances Versus R1

   ----------------------------
     R1        Input resistance
   ----------------------------
${$%1g$      $R1$ $%4.3f$          $InputRes$}$
   ----------------------------
End
```

The report follows.

Report for Question 1021

```
     Input Resistances Versus R1

   ----------------------------
     R1        Input resistance
   ----------------------------
     1              1.791
     2              2.791
     3              3.791
     4              4.791
     5              5.791
   ----------------------------
```

The sweep of input resistance versus R_1 is shown in Fig. SQ1021.

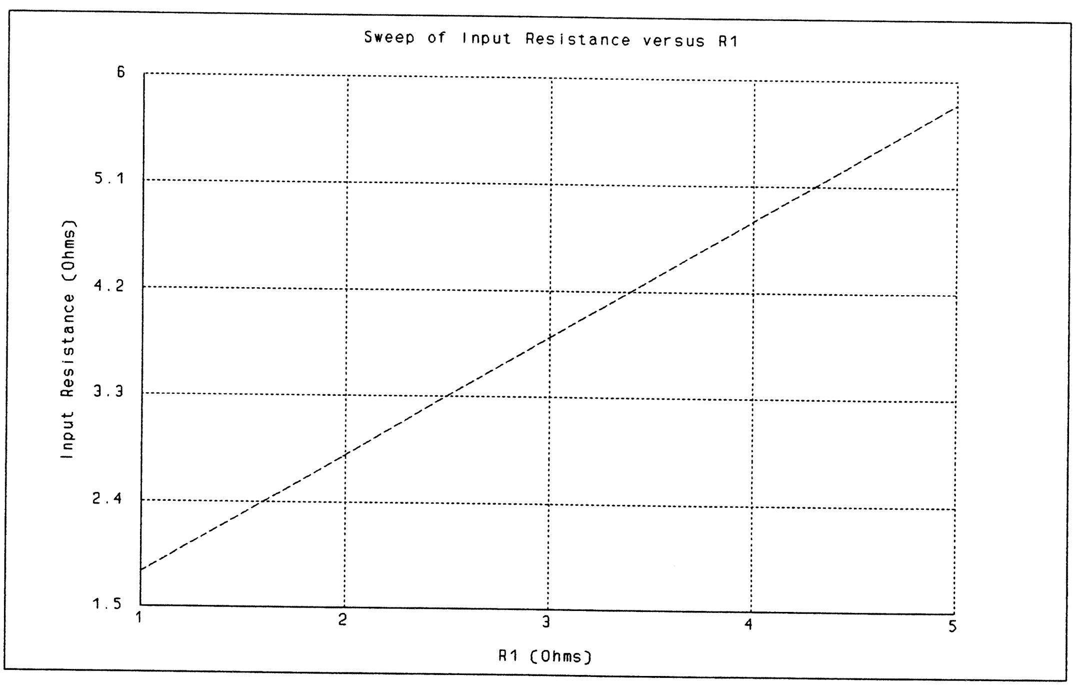

Fig. SQ1021 Sweep of input resistance versus R_1.

OSA☞ Question 1024 Use the LU factorization capability of OSA90/hope to calculate and print out all the branch voltages and currents of the resistive ladder network of Fig. Q1022 with $n = 7$, $R_2 = R_4 = R_6 = 1/3\ \Omega$, $R_1 = R_3 = R_5 = R_7 = 1\ \Omega$. Calculate the voltages and currents for $V_g = 0\text{-}3$ V with an incremental step of 1 V. (See Question 1022.)

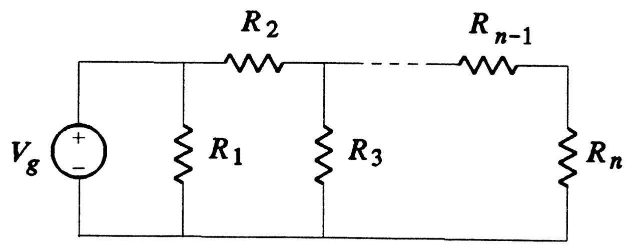

Fig. Q1022 Resistive ladder network.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 1024

```
! File name: Q1024.ckt
! Circuit file for solving Question 1024

Expression
   ! Assign values to circuit elements
   R1 = 1OH;    G1 = 1 / R1;
   R2 = 1/3OH;  G2 = 1 / R2;
   R3 = 1OH;    G3 = 1 / R3;
   R4 = 1/3OH;  G4 = 1 / R4;
   R5 = 1OH;    G5 = 1 / R5;
   R6 = 1/3OH;  G6 = 1 / R6;
   R7 = 1OH;    G7 = 1 / R7;
   Vg = 1V;                      ! Initialize the input voltage

   ! Set up conductance matrix
   G[4,4] = [   1        0         0         0
              -G2   (G2+G3+G4)    -G4        0
               0        -G4    (G4+G5+G6)   -G6
               0         0        -G6     (G6+G7)   ];

   ! Define the right-hand-side of nodal equation
   RHSide[4] = [ VG  0   0   0 ];

   ! LU factorization and substitution
   LUG[5,4] = LU(G);            ! LU factorize G
   V[4] = SUBST(LUG,RHSide);    ! Use substitution to get the nodal voltages V[]

   ! calculate the branch voltages
   V1 = V[1];
   V3 = V[2];
   V5 = V[3];
   V7 = V[4];
   V2 = V1 - V3;
   V4 = V3 - V5;
   V6 = V5 - V7;
```

```
    ! Calculate the branch currents
    I1 = V1 / R1;
    I2 = V2 / R2;
    I3 = V3 / R3;
    I4 = V4 / R4;
    I5 = V5 / R5;
    I6 = V6 / R6;
    I7 = V7 / R7;
End

! Define the sweep parameter and output responses

Sweep
  Vg: from 0 to 3 step 0.5 V1 V2 V3 V4 V5 V6 V7
  {Xsweep Title = "Sweeps of branch voltages versus input voltage Vg"
   X = Vg  X_title = "Input voltage Vg (V)"  Y_title="Branch voltages (V)"
   Y = V1.white.circle & V2.Red.Triangle & V3.Yellow.Square & V4.Green.Dot
   & V5.LightBlue.diamond & V6.Pink.X & V7.Cream.Cross & V1.white
   & V2.Red & V3.Yellow & V4.Green & V5.LightBlue & V6.Pink & V7.Cream};

  Vg: from 0 to 3 step 0.5 I1 I2 I3 I4 I5 I6 I7
  {Xsweep Title = "Sweeps of branch currents versus input voltage Vg"
   X = Vg  X_title = "Input voltage Vg (V)"  Y_title="Branch Currents (A)"
   Y = I1.white.circle & I2.Red.Triangle & I3.Yellow.Square & I4.Green.Dot
   & I5.LightBlue.diamond & I6.Pink.X & I7.Cream.Cross & I1.white
   & I2.Red & I3.Yellow & I4.Green & I5.LightBlue & I6.Pink & I7.Cream};
End

! Define report block

Report
                      Branch Voltages Versus Input Voltage Vg

    ---------------------------------------------------------------
      Vg     V1      V2      V3      V4      V5      V6      V7
    ---------------------------------------------------------------
${$%2.1f$     $Vg$ $%5.4f$ $V1$ $%5.4f$ $V2$ $%5.4f$ $V3$ $%5.4f$ $V4$ $%5.4f$ $V5$ $%5.4f$ $V6$ $%5.4f$ $V7$}$
    ---------------------------------------------------------------

                      Branch Currents Versus Input Voltage Vg

    ---------------------------------------------------------------
      Vg     I1      I2      I3      I4      I5      I6      I7
    ---------------------------------------------------------------
${$%2.1f$     $Vg$ $%5.4f$ $I1$ $%5.4f$ $I2$ $%5.4f$ $I3$ $%5.4f$ $I4$ $%5.4f$ $I5$ $%5.4f$ $I6$ $%5.4f$ $I7$}$
    ---------------------------------------------------------------
End
```

There are two sweep sets, one for branch voltage calculation and the other for branch current calculation. The report follows.

Report for Question 1024

Branch Voltages Versus Input Voltage Vg

Vg	V1	V2	V3	V4	V5	V6	V7
0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.0	1.0000	0.4124	0.5876	0.2165	0.3711	0.0928	0.2784
2.0	2.0000	0.8247	1.1753	0.4330	0.7423	0.1856	0.5567
3.0	3.0000	1.2371	1.7629	0.6495	1.1134	0.2784	0.8351

Branch Currents Versus Input Voltage Vg

Vg	I1	I2	I3	I4	I5	I6	I7
0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.0	1.0000	1.2371	0.5876	0.6495	0.3711	0.2784	0.2784
2.0	2.0000	2.4742	1.1753	1.2990	0.7423	0.5567	0.5567
3.0	3.0000	3.7113	1.7629	1.9485	1.1134	0.8351	0.8351

Fig SQ1024.1 and Fig SQ1024.2 show the branch voltages and currents versus input voltage V_g.

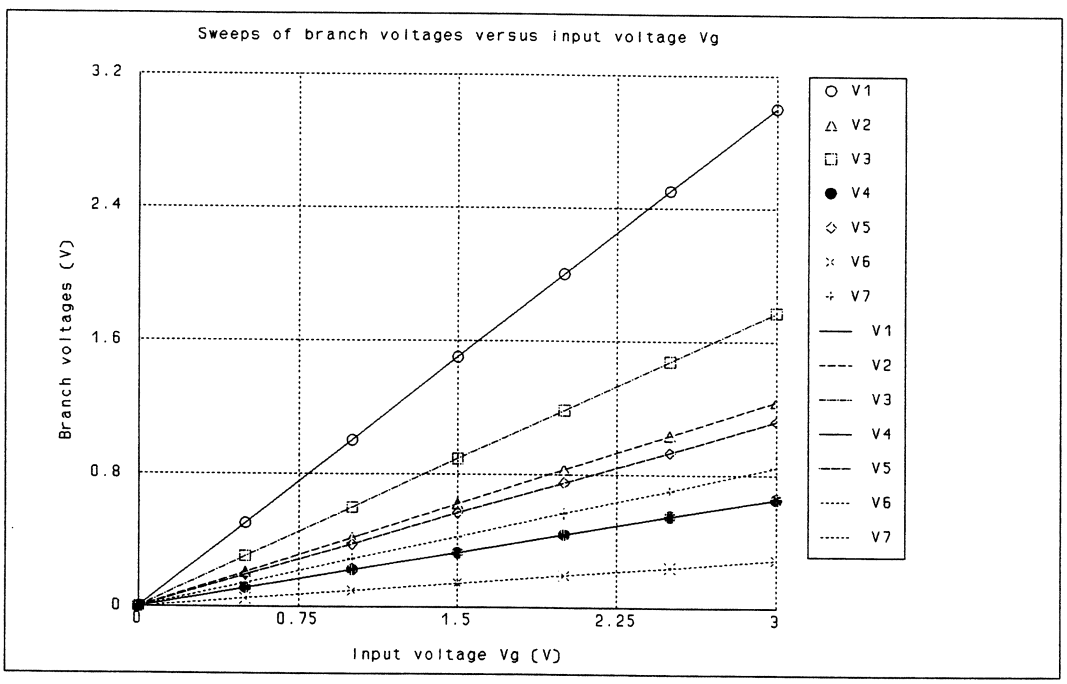

Fig. SQ1024.1 Sweeps of branch voltages versus input voltage V_g.

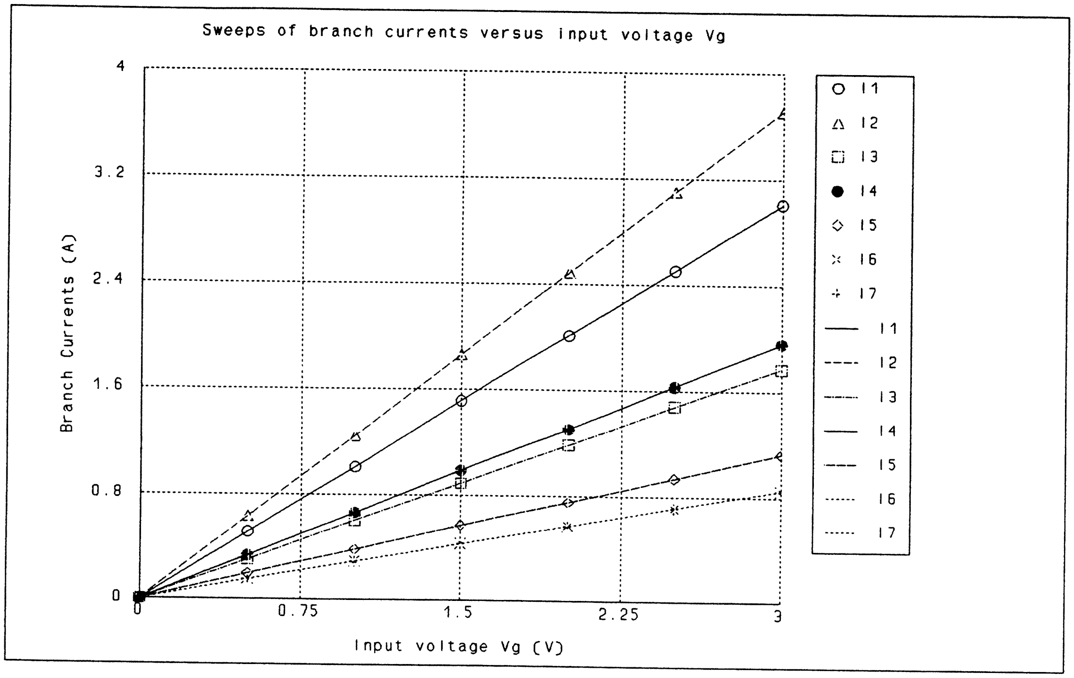

Fig. SQ1024.2 Sweeps of branch currents versus input voltage V_g.

OSA☞ Question 1026 Use OSA90/hope to calculate the input conductance of the circuit of Question 1024. Calculate the input conductance for the numerical example in Question 1024. Show the input conductances for R_1 = 1-10 Ω with an incremental step of 1 Ω. (See Question 1025.)

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 1026

```
! File name: Q1026.ckt
! Circuit file for solving Question 1026

Expression
   ! Assign values for circuit elements
   R1 = 1OH;   G1 = 1 / R1;   ! Initialize R1
   R2 = 1/3OH; G2 = 1 / R2;
   R3 = 1OH;   G3 = 1 / R3;
   R4 = 1/3OH; G4 = 1 / R4;
   R5 = 1OH;   G5 = 1 / R5;
   R6 = 1/3OH; G6 = 1 / R6;
   R7 = 1OH;   G7 = 1 / R7;
   Vg = 1V;

   ! Set up the conductance matrix
   G[4,4] = [   1        0          0         0
              -G2  (G2+G3+G4)     -G4         0
               0       -G4    (G4+G5+G6)   -G6
               0        0         -G6    (G6+G7)   ];

   ! Define the right-hand side of nodal equation
   RHSide[4] = [ VG 0  0  0 ];

   ! LU factorization and substitution
   LUG[5,4] = LU(G);               ! LU factorize G
   VLU[4] = SUBST(LUG,RHSide);     ! Use substitution to get branch voltages V

   ! Calculate the input current
   V1 = VLU[1];
   V3 = VLU[3];
   V2 = V1 - V3;
   I1 = V1 / R1;
   I2 = V2 / R2;
   InputI = I1 + I2;

   ! Calculate the input conductance
   InputCond = InputI/VG;
End

! Define the sweep parameter and output response

Sweep
   R1: from 1 to 10 step 1 InputCond
   {Xsweep  Title = "Sweep of input conductance versus R1"
    X = R1  X_title = "R1 (Ohms)"
    Y = InputCond.blue & InputCond.red.dot
    Y_title = "Input Conductance (Siemens)"};
End
```

```
! Define the report block

Report
     Input Conductance Versus R1

   ------------------------------
    R1         Input conductance
   ------------------------------
${$%2g$     $R1$ $%4.3f$            $InputCond$}$
   ------------------------------
End
```

The report shows the numerical results and Fig. SQ1026 shows graphically the input conductance versus R_1.

Report for Question 1026

Input Conductance Versus R1

R1	Input conductance
1	2.887
2	2.387
3	2.220
4	2.137
5	2.087
6	2.053
7	2.029
8	2.012
9	1.998
10	1.987

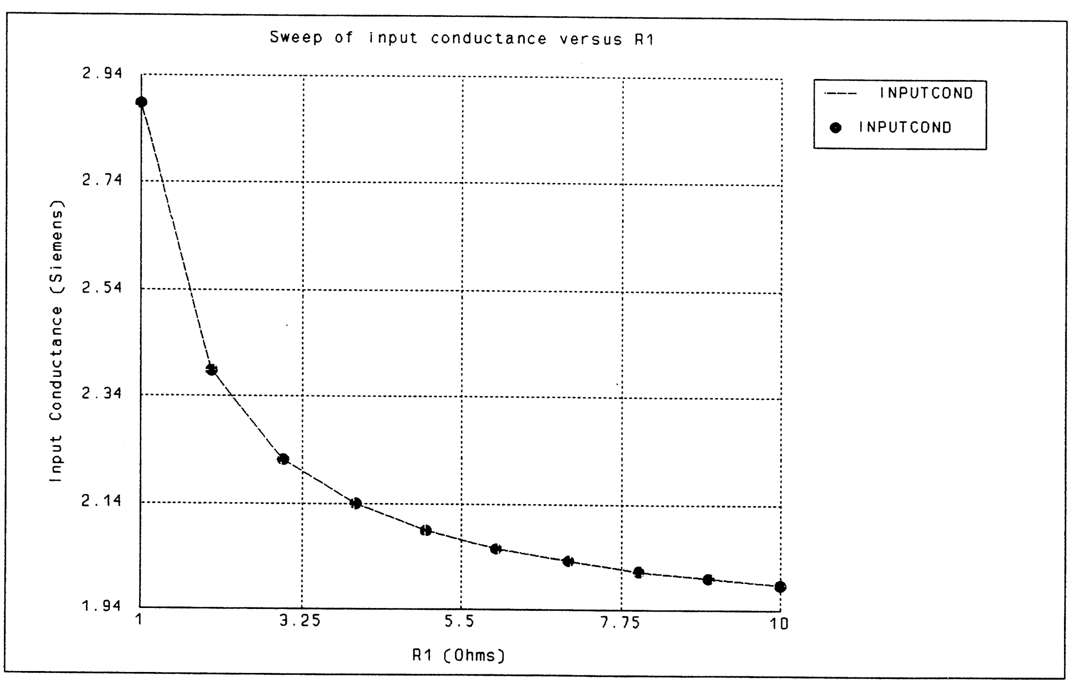

Fig. SQ1026 Sweep of input conductance versus R_1.

OSA☞ Question 1028 Consider the ladder network of Fig. Q1027. Use OSA90/hope to calculate the node voltages by (i) matrix inversion and (ii) LU factorization. Set the right-hand source to zero and recalculate the node voltages. (See Question 1027.)

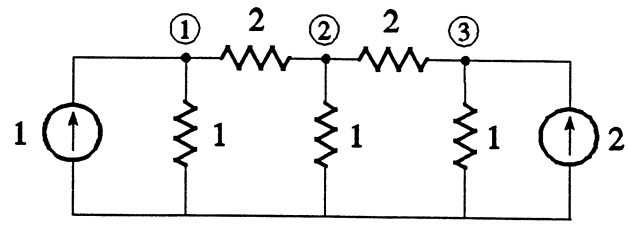

Fig. Q1027 Three-node resistive ladder network.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 1028

```
! File name: Q1028.ckt
! circuit file for solving Question 1028

Control
   Non_Microwave_Units;
End

Expression
   ! Set up conductance matrix
   G[3,3] = [ (1+1/2)   (-1/2)       0
              (-1/2)  (1/2+1+1/2)  (-1/2)
                0       (-1/2)     (1/2+1)  ];

   ! Define the right-hand side of nodal equation
   I[3] = [ 1  0  2 ];

   ! Using matrix inversion to calculate node voltages
   Ginv[3,3] = INVERSE(G);
   Vinv[3] = PRODUCT(Ginv,I);

   ! Using LU factorization to calculate node voltages
   LUG[4,3] = LU(G);         ! LU factorize G
   VLU[3] = SUBST(LUG,I);    ! Use substitution to get branch voltages V

   ! Setting the right-hand source to 0 and recalculating
   I0[3] = [ 1  0  0 ];
   Vinv0[3] = PRODUCT(Ginv,I0);
   VLU0[3] = SUBST(LUG,I0);

   k = 1;   ! Initialize the index for sweep display
End

Model
   ! Assign values to the circuit elements
   Iin1 = 1A;
   Iin2 = 2A;

   ! Define the subcircuit
   SUBCIRCUIT Q1028_ckt 0 1 2 3 {
   RES 1 0 R = 1OH;
   RES 1 2 R = 2OH;
```

```
   RES 2 0 R = 10H;
   RES 2 3 R = 20H;
   RES 3 0 R = 10H;
   };

   ! Define original circuit
   Q1028_ckt 0 1 2 3;
   ISOURCE 1 0 NAME = Iinp1 IDC = Iin1;
   ISOURCE 3 0 NAME = Iinp2 IDC = Iin2;
   VLABEL 1 0 NAME = V1;
   VLABEL 2 0 NAME = V2;
   VLABEL 3 0 NAME = V3;
   V[3] = [ V1 V2 V3 ];

   ! Define circuit with right hand source equal to zero
   Q1028_ckt 0 4 5 6;
   ISOURCE 4 0 NAME = Iinp11 IDC = Iin1;
   VLABEL 4 0 NAME = V11;
   VLABEL 5 0 NAME = V22;
   VLABEL 6 0 NAME = V33;
   V0[3] = [ V11 V22 V33 ];

   CIRCUIT NAME = Q1028;
End

Sweep
   DC:  k: from 1 to 3 step 1 Iin1 = 1A Iin2 = 2A
   Vinv[k] VLU[k] Vinv0[k] VLU0[k] V[k] V0[k] V[k] V0[k];
End

! Define report block

Report
                   Values of Nodal Voltages
                (The right hand source is 2A)

   ----------------------------------------------------------------------------
      Node       Voltage values          Voltage values          Voltage values
     number   (by LU Factorization)   (by matrix inversion)   (by circuit model)
   ----------------------------------------------------------------------------
${$%1g$       $k$      $%5.4f$       $VLU[k]$   $%5.4f$            $Vinv[k]$   $%5.4f$            $V[k]$}$
   ----------------------------------------------------------------------------


                   Values of Nodal Voltages
                (The right hand source is 0A)

   ----------------------------------------------------------------------------
      Node       Voltage values          Voltage values          Voltage values
     number   (by LU Factorization)   (by matrix inversion)   (by circuit model)
   ----------------------------------------------------------------------------
${$%1g$       $k$      $%5.4f$       $VLU0[k]$   $%5.4f$            $Vinv0[k]$   $%5.4f$            $V0[k]$}$
   ----------------------------------------------------------------------------
End
```

The results are shown in the report. From the report we can see that the results obtained by matrix inversion and LU factorization are the same.

Report for Question 1028

Values of Nodal Voltages
(The right hand source is 2A)

Node number	Voltage values (by LU Factorization)	Voltage values (by matrix inversion)	Voltage values (by circuit model)
1	0.8667	0.8667	0.8667
2	0.6000	0.6000	0.6000
3	1.5333	1.5333	1.5333

Values of Nodal Voltages
(The right hand source is 0A)

Node number	Voltage values (by LU Factorization)	Voltage values (by matrix inversion)	Voltage values (by circuit model)
1	0.7333	0.7333	0.7333
2	0.2000	0.2000	0.2000
3	0.0667	0.0667	0.0667

OSA☞ Question 1033 Use OSA90/hope to factorize the following matrix into LU form.

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

(See Question 1032.)

Solution

The circuit file for solving this question follows.

Circuit file for solving Question 1033

```
!  File name: Q1033.ckt
!  Circuit file for solving Question 1033

Expression
   ! The original matrix
   Matrix[4,4] = [  5 -1  0  0
                   -1  6 -1  0
                    0 -1  6 -1
                    0  0 -1  5  ];

   ! LU factorize the matrix without pivoting
   MatrixLU[5,4] = LUF(Matrix);

   ! Extract lower and upper triangular matrix
   L[4,4] = EXTRACT_L(MatrixLU);
   U[4,4] = EXTRACT_U(MatrixLU);

   i = 1;  ! Initialize the row index for matrix display
End

! Define the sweep parameter and output responses

Sweep
   i: from 1 to 4 step 1 L[i,1] L[i,2] L[i,3] L[i,4];
   i: from 1 to 4 step 1 U[i,1] U[i,2] U[i,3] U[i,4];
End

! Define the report block

Report
       Lower Triangular Matrix L

   ----------------------------------
   $% 5.4f$ $L$
   ----------------------------------


       Upper Triangular Matrix U

   ----------------------------------
   $% 5.4f$ $U$
   ----------------------------------
End
```

There are two sweep sets, one of which displays the lower triangular matrix ***L*** and the other one displays the upper triangular matrix ***U***. The results are shown in the report.

Report for Question 1033

Lower Triangular Matrix L

1.0000	0.0000	0.0000	0.0000
-0.2000	1.0000	0.0000	0.0000
0.0000	-0.1724	1.0000	0.0000
0.0000	0.0000	-0.1716	1.0000

Upper Triangular Matrix U

5.0000	-1.0000	0.0000	0.0000
0.0000	5.8000	-1.0000	0.0000
0.0000	0.0000	5.8276	-1.0000
0.0000	0.0000	0.0000	4.8284

OSA☞ Question 1040 Consider the linear circuit shown in Fig. Q1039, which is operating in the sinusoidal steady state. Use OSA90/hope to find V_3/V_1 for this circuit at $\omega = 0\text{-}10$ rad/s with an incremental step of 1 rad/s by (i) matrix inversion, (ii) LU factorization and (iii) direct circuit simulation. Take $R_1 = R_2 = R_3 = 2\ \Omega$, $C_1 = C_2 = C_3 = 1$ F. (See Question 1039.)

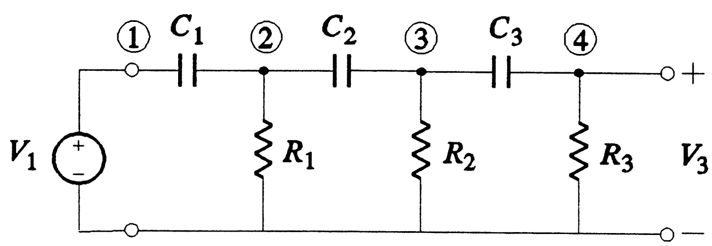

Fig. Q1039 CR ladder network.

Solution

Since this is a problem involving complex numbers, we can not perform LU factorization directly on the original complex admittance matrix using the current version of OSA90/hope. The original nodal equations should be split into real and imaginary parts. Then an equivalent problem involving real numbers can be solved by OSA90/hope. We solve this problem in two ways: by direct splitting and using macros.

The circuit file using direct splitting follows.

Circuit file using direct splitting for solving Question 1040

```
!  File name: Q1040_1.ckt
!  Circuit file for solving Question 1040
!  by splitting complex matrix into real and imaginary matrices

! Illustrate that the units are not microwave units

Control
  Non_Microwave_Units;
End

Expression
   #define w Omega     ! Substitute w by Omega

   ! Assign values to circuit elements
   R1 = 2OH; G1 = 1 / R1;
   R2 = 2OH; G2 = 1 / R2;
   R3 = 2OH; G3 = 1 / R3;
   C1 = 1F;
   C2 = 1F;
   C3 = 1F;
   V1 = 1V;
   Omega = 1;    ! Initialize Omega

   ! Set up the admittance matrix in real and imaginary parts
   Y_real[4,4] = [  1    0    0    0
                 0   G1    0    0
                 0    0   G2    0
                 0    0    0   G3  ];
   Y_imag[4,4] = [  0          0           0           0
            (-C1*w)  (C1*w+C2*w)    (-C2*w)         0
                 0       (-C2*w)   (C2*w+C3*w)     (-C3*w)
                 0          0        (-C3*w)       (C3*w)  ];
```

```
   ! Create an expanded admittance matrix
   Y[8,8] = [ Y_real  -Y_imag
              Y_imag   Y_real ];

   ! Define the right-hand side of nodal equation in real and imaginary parts
   I_real[4] = [ 1  0  0  0 ];
   I_imag[4] = [ 0  0  0  0 ];

   ! Create an expanded right-hand side vector
   I[8] = [ I_real I_imag ];

   ! Calculate the nodal voltages by matrix inversion
   Y_INV[8,8] = INVERSE(Y);
   V_INV[8] = PRODUCT(Y_INV,I);

   ! Calculate the nodal voltages by LU factorization
   Y_LU[9,8] = LU(Y);
   V_LU[8] = SUBST(Y_LU,I);

   ! Convert the voltage at Node 3 from real and imaginary to magnitude and phase
   RI2MP(V_INV[4],V_INV[8],MV3_INV,PV3_INV);
   RI2MP(V_LU[4],V_LU[8],MV3_LU,PV3_LU);

   ! Calculate the voltage gain
   Gain_INV = MV3_INV / V1;
   Gain_LU = MV3_LU / V1;

   ! Calculate the voltage phase shift
   Phase_INV = PV3_INV - 0;
   Phase_LU = PV3_LU - 0;
End

! Calculate the voltage gain by direct circuit simulation

Model
   VSOURCE 1 0 NAME = Vinp V = V1;
   CAP 1 2 C = C1;
   RES 2 0 R = R1;
   CAP 2 3 C = C2;
   RES 3 0 R = R2;
   CAP 3 4 C = C3;
   RES 4 0 R = R3;
   VLABEL 4 0 NAME = V3;
   CIRCUIT NAME = Q1040_1;
   Gain_CKT = MV3[1] / V1;
   Phase_CKT = PV3[1] - 0;
End

! Define the sweep parameter and output responses

Sweep
   HB: Omega: from 0 to 10 step 1 FREQ = (1HZ * Omega / (2 * PI)) Gain_CKT Gain_INV Gain_LU
   {Xsweep
         Title = "Voltage gain vs Frequency"
         X_title = "Omega (rad/sec)"
         X = Omega
         Y_title = "Gain"
         Y = Gain_CKT.red & Gain_INV.blue.dot & Gain_LU.yellow.square};

   HB: Omega: from 0 to 10 step 1 FREQ = (1HZ * Omega / (2 * PI)) Phase_CKT Phase_INV Phase_LU
   {Xsweep
         Title = "Phase Shift vs Frequency"
         X_title = "Omega (rad/sec)"
         X = Omega
         Y_title = "Phase Shift (degrees)"
         Y = Phase_CKT.red & Phase_INV.blue.dot & Phase_LU.yellow.square};
End

! Define report block

Report
```

```
                 Gain versus Frequency

    --------------------------------------------------------
                               Gain
            ---------------------------------------------
             From matrix      From LU        From circuit
     Omega    inversion    factorization      simulation
    --------------------------------------------------------
${$%2g$       $Omega$ $%5.4f%$       $Gain_INV$  $%5.4f$       $Gain_LU$  $%5.4f$       $Gain_CKT$}$
    --------------------------------------------------------


              Phase Shift versus Frequency

    --------------------------------------------------------
                       Phase Shift in degrees
            ---------------------------------------------
             From matrix      From LU        From circuit
     Omega    inversion    factorization      simulation
    --------------------------------------------------------
${$%2g$       $Omega$ $%5.4f%$       $Phase_INV$  $%5.4f$       $Phase_LU$  $%5.4f$       $Phase_CKT$}$
    --------------------------------------------------------
End
```

The results obtained with this circuit file using direct splitting are shown in the first report numerically and in Fig. SQ1040.1 and Fig. SQ1040.2 graphically.

The report for Question 1040 using direct splitting

Gain versus Frequency

	Gain		
Omega	From matrix inversion	From LU factorization	From circuit simulation
0	0.0000	0.0000	0.0000
1	0.3465	0.3465	0.3465
2	0.6113	0.6113	0.6113
3	0.7598	0.7598	0.7598
4	0.8423	0.8423	0.8423
5	0.8904	0.8904	0.8904
6	0.9201	0.9201	0.9201
7	0.9395	0.9395	0.9395
8	0.9527	0.9527	0.9527
9	0.9621	0.9621	0.9621
10	0.9690	0.9690	0.9690

Phase Shift versus Frequency

	Phase Shift in degrees		
Omega	From matrix inversion	From LU factorization	From circuit simulation
0	0.0000	0.0000	0.0000
1	94.9697	94.9697	94.9697
2	65.1484	65.1484	65.1484
3	49.1364	49.1364	49.1364
4	39.0573	39.0573	39.0573
5	32.2325	32.2325	32.2325
6	27.3564	27.3564	27.3564
7	23.7215	23.7215	23.7215
8	20.9179	20.9179	20.9179
9	18.6950	18.6950	18.6950
10	16.8920	16.8920	16.8920

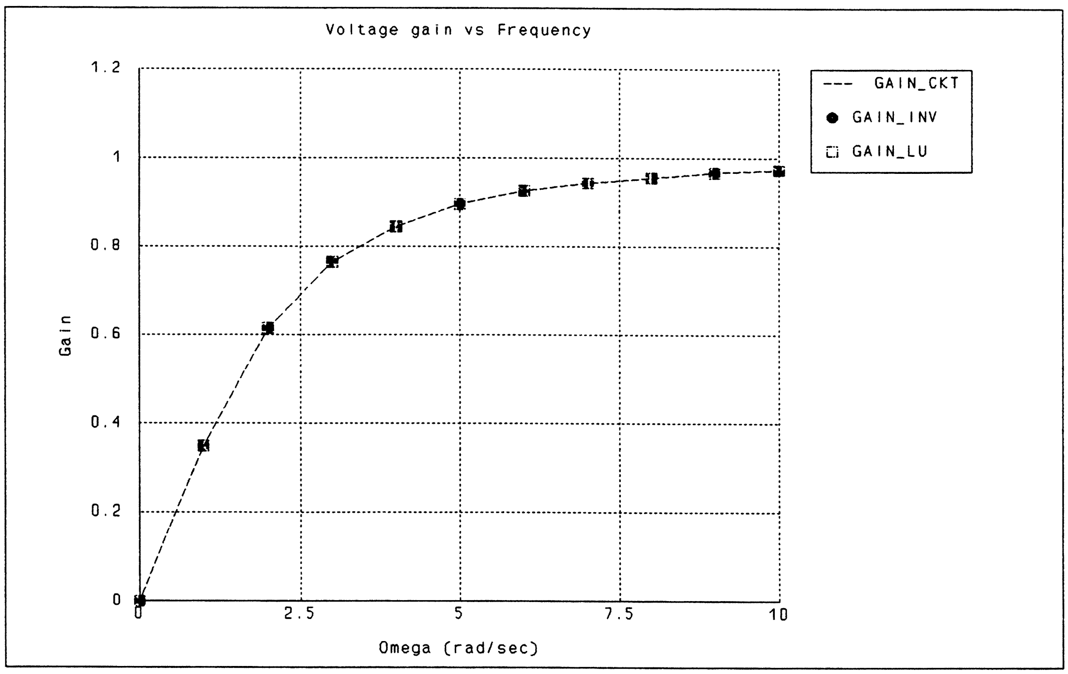

Fig. SQ1040.1 Voltage gain versus frequency.

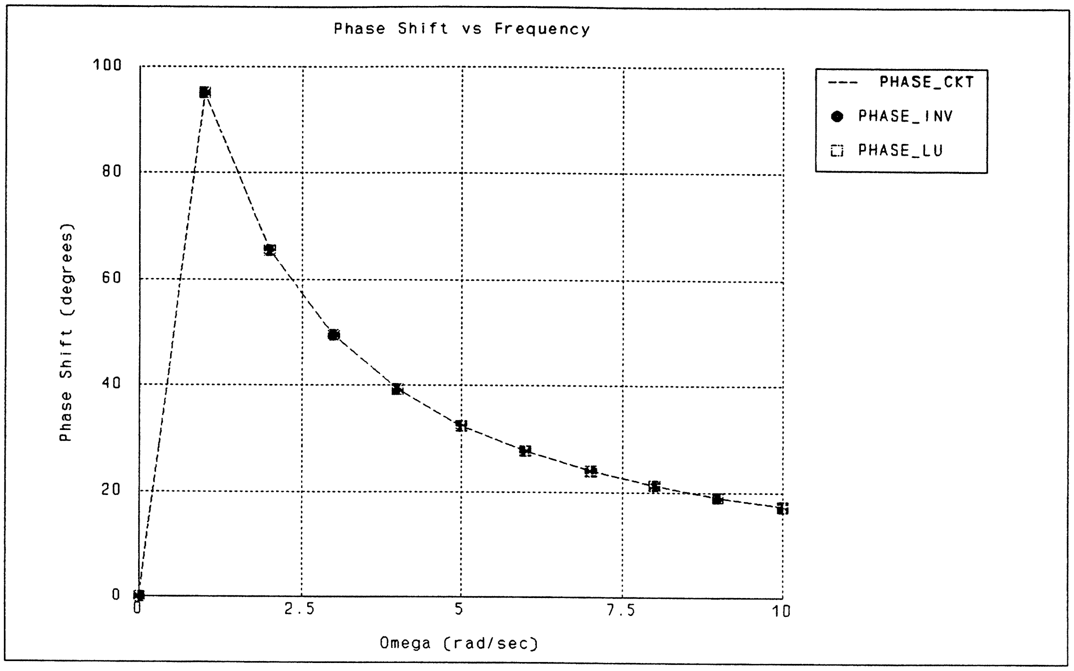

Fig. SQ1040.1 Phase shift versus frequency.

The circuit file using macros follows.

Circuit file using macros for solving Question 1040

```
!  File name: Q1040_2.ckt
!  Circuit file for solving Question 1040
!  by using a macro function substitution in the complex matrix

! Illustrate that the units are not the microwave units

Control
  Non_Microwave_Units;
end

Expression
   #define w Omega     ! Substitute w by Omega

   ! Define a macro for substituting Complex(x,y) by the 2 by 2 matrix [ x -y y x]
   #define Complex(name,x,y)  name[2,2]=[ x -y y x ];

   ! Assign values to circuit elements
   R1 = 2OH; G1 = 1 / R1;
   R2 = 2OH; G2 = 1 / R2;
   R3 = 2OH; G3 = 1 / R3;
   C1 = 1F;
   C2 = 1F;
   C3 = 1F;
   V1 = 1V;
   Omega = 1;    ! Initialize Omega

   ! Use the macro to define matrix entries
   Complex(y11,1,0);
```

```
   Complex(y12,0,0);
   Complex(y13,0,0);
   Complex(y14,0,0);
   Complex(y21,0,(-C1*w));
   Complex(y22,G1,(C1*w+C2*w));
   Complex(y23,0,(-C2*w));
   Complex(y24,0,0);
   Complex(y31,0,0);
   Complex(y32,0,(-C2*w));
   Complex(y33,G2,(C2*w+C3*w));
   Complex(y34,0,(-C3*w));
   Complex(y41,0,0);
   Complex(y42,0,0);
   Complex(y43,0,(-C3*w));
   Complex(y44,G3,(C3*w));

   ! Create the expanded admittance matrix
   Y[8,8] = [ y11 y12 y13 y14
              y21 y22 y23 y24
              y31 y32 y33 y34
              y41 y42 y43 y44 ];

   ! Define the expanded right-hand side of the nodal equation
   I[8]=[ 1  0  0  0  0  0  0  0  ];

   ! Calculate the node voltages.  In this case the odd rows of I & V represent
   ! the real components and the even rows the imaginary ones

   ! Calculate the node voltages by matrix inversion
   Y_INV[8,8]=INVERSE(Y);
   V_INV[8]=PRODUCT(Y_INV,I);

   ! Calculate the node voltages by LU factorization
   Y_LU[9,8]=LU(Y);
   V_LU[8]=SUBST(Y_LU,I);

   ! Convert the voltage of Node 3 from real and imaginary to magnitude and phase
   RI2MP(V_INV[7],V_INV[8],MV3_INV,PV3_INV);
   RI2MP(V_LU[7],V_LU[8],MV3_LU,PV3_LU);

   ! Calculate the voltage gain
   Gain_INV = MV3_INV / V1;
   Gain_LU = MV3_LU / V1;

   ! Calculate the voltage phase shift
   Phase_INV = PV3_INV - 0;
   Phase_LU = PV3_LU - 0;
End

! Calculate the voltage gain by direct circuit simulation

Model
   VSOURCE 1 0 NAME = Vinp V = V1;
   CAP 1 2 C = C1;
   RES 2 0 R = R1;
   CAP 2 3 C = C2;
   RES 3 0 R = R2;
   CAP 3 4 C = C3;
   RES 4 0 R = R3;
   VLABEL 4 0 NAME = V3;
   CIRCUIT NAME = Q1040_2;
   Gain_CKT = MV3[1] / V1;
   Phase_CKT = PV3[1] - 0;
End

! Define the sweep parameter and output responses

Sweep
   HB: Omega: from 0 to 10 step 1 FREQ = (1HZ * Omega / (2 * PI)) Gain_CKT Gain_INV Gain_LU
   {Xsweep
        Title = "Voltage gain vs Frequency"
```

```
        X_title = "Omega (rad/sec)"
        X = Omega
        Y_title = "Gain"
        Y = Gain_CKT.red & Gain_INV.blue.dot & Gain_LU.yellow.square};

   HB: Omega: from 0 to 10 step 1 FREQ = (1HZ * Omega / (2 * PI)) Phase_CKT Phase_INV Phase_LU
   {Xsweep
        Title = "Phase Shift vs Frequency"
        X_title = "Omega (rad/sec)"
        X = Omega
        Y_title = "Phase Shift (degrees)"
        Y = Phase_CKT.red & Phase_INV.blue.dot & Phase_LU.yellow.square};
End

! Define report block

Report
                    Gain Versus Frequency

   -----------------------------------------------------
                              Gain
           -------------------------------------------
            From matrix      From LU      From circuit
     Omega   inversion    factorization    simulation
   -----------------------------------------------------
${$%2g$     $Omega$ $%5.4f%$      $Gain_INV$  $%5.4f$       $Gain_LU$  $%5.4f$       $Gain_CKT$}$
   -----------------------------------------------------


                 Phase Shift versus Frequency

   -----------------------------------------------------
                      Phase Shift in degrees
           -------------------------------------------
            From matrix      From LU      From circuit
     Omega   inversion    factorization    simulation
   -----------------------------------------------------
${$%2g$     $Omega$ $%5.4f%$      $Phase_INV$  $%5.4f$       $Phase_LU$  $%5.4f$       $Phase_CKT$}$
   -----------------------------------------------------
End
```

The numerical results obtained using this circuit file are shown in the second report. Fig. SQ1040.3 shows the gain versus frequency and Fig. SQ1040.4 shows the phase shift versus frequency.

The report for Question 1040 using macros

Gain versus Frequency

	Gain		
Omega	From matrix inversion	From LU factorization	From circuit simulation
0	0.0000	0.0000	0.0000
1	0.3465	0.3465	0.3465
2	0.6113	0.6113	0.6113
3	0.7598	0.7598	0.7598
4	0.8423	0.8423	0.8423
5	0.8904	0.8904	0.8904
6	0.9201	0.9201	0.9201
7	0.9395	0.9395	0.9395
8	0.9527	0.9527	0.9527
9	0.9621	0.9621	0.9621
10	0.9690	0.9690	0.9690

Phase Shift versus Frequency

	Phase Shift in degrees		
Omega	From matrix inversion	From LU factorization	From circuit simulation
0	0.0000	0.0000	0.0000
1	94.9697	94.9697	94.9697
2	65.1484	65.1484	65.1484
3	49.1364	49.1364	49.1364
4	39.0573	39.0573	39.0573
5	32.2325	32.2325	32.2325
6	27.3564	27.3564	27.3564
7	23.7215	23.7215	23.7215
8	20.9179	20.9179	20.9179
9	18.6950	18.6950	18.6950
10	16.8920	16.8920	16.8920

From the two reports and Figs. SQ1040.1, SQ1040.2, SQ1040.3 and SQ1040.4 we can observe that the results obtained by using direct splitting and macros are exactly the same. Also the results obtained by matrix inversion, LU factorization and direct circuit simulation are identical. The important aspect to note is that the method of matrix inversion is the most computationally intensive of the three methods employed. As well, the LU method could have alternatively been implemented using the LU and SUBST functions in place of SOLVE. [Source: Array Functions, OSA90/hope User's Manual.]

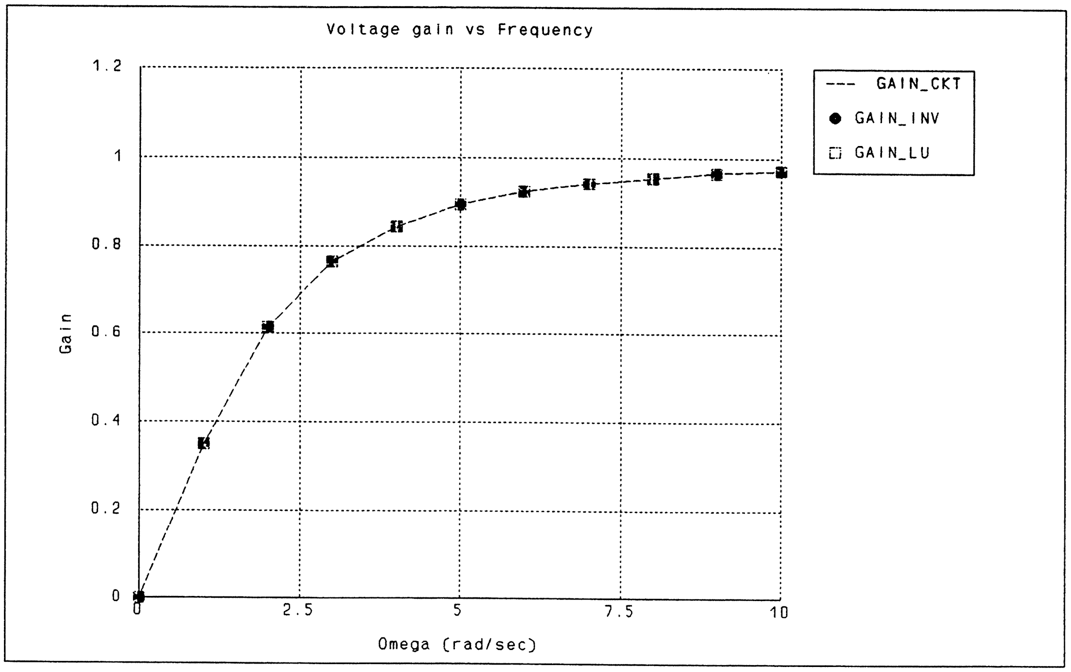

Fig. SQ1040.3 Voltage gain versus frequency.

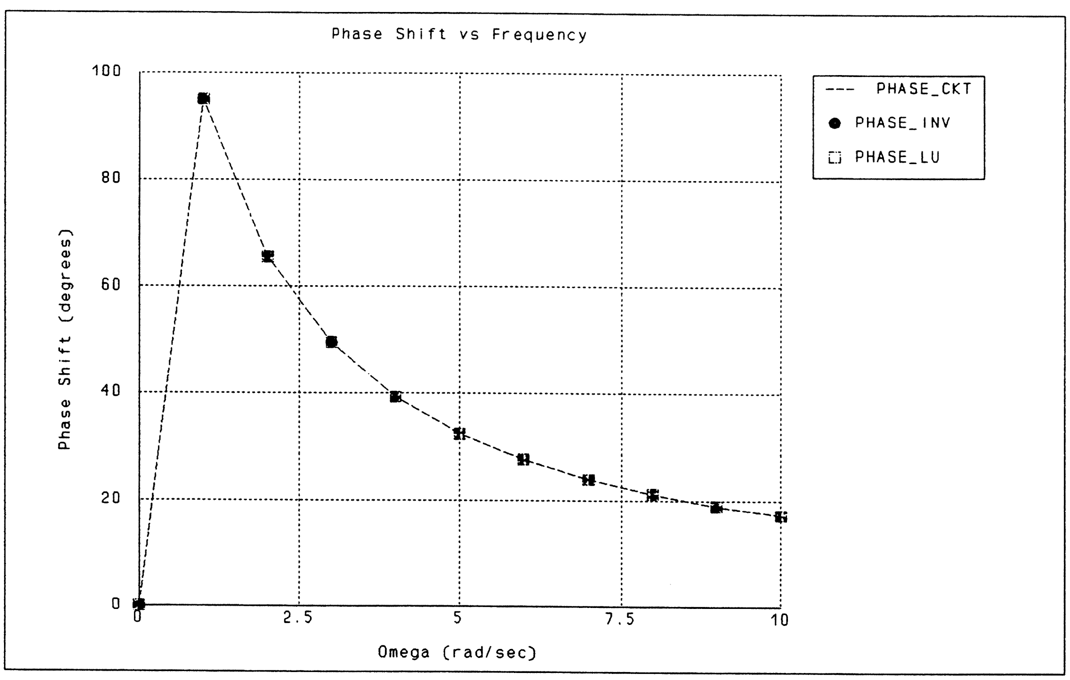

Fig. SQ1040.4 Phase shift versus frequency.

OSA☞ Question 1042 Consider the linear circuit shown in Fig. Q1041, which is operating in the sinusoidal steady state. Use OSA90/hope to find V_3/V_1 for this circuit at ω = 0-5 rad/s with an incremental step of 0.2 rad/s by (i) matrix inversion, (ii) LU factorization and (iii) direct circuit simulation. Take $R_1 = R_2 = R_3 = 1\ \Omega$, $C_1 = C_2 = C_3 = 2$ F. (See Question 1041.)

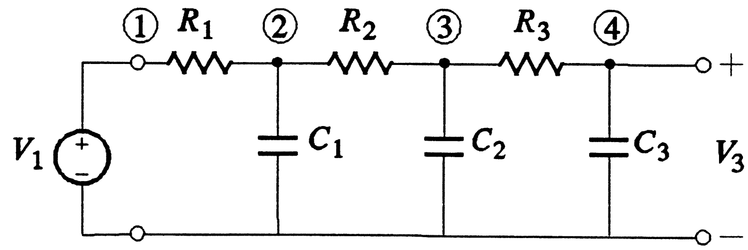

Fig. Q1041 RC ladder network.

Solution

Analogous to the solution of Question 1040, there are two ways to solve this problem. Here, we use only the direct splitting method. The circuit file for solving this problem is

Circuit file for solving Question 1042

```
!  File name: Q1042.ckt
!  Circuit file for solving Question 1042

! Illustrate that the units are not microwave units

Control
  Non_Microwave_Units;
End

Expression
   #define w Omega    ! Substitute w by Omega

   ! Assign values to circuit elements
   R1 = 1OH; G1 = 1 / R1;
   R2 = 1OH; G2 = 1 / R2;
   R3 = 1OH; G3 = 1 / R3;
   C1 = 2F;
   C2 = 2F;
   C3 = 2F;
   V1 = 1V;
   Omega = 1;   ! Initialize Omega

   ! Set up the admittance matrix in real and imaginary parts
   Y_real[3,3] = [ (G1+G2)   -G2       0
                    -G2    (G2+G3)  -G3
                    0       -G3      G3  ];
   Y_imag[3,3] = [ (C1*w)     0        0
                   0       (C2*w)     0
                   0        0     (C3*w) ];

   ! Create an expanded admittance matrix
   Y[6,6] = [ Y_real  -Y_imag
              Y_imag   Y_real ];

   ! Define the right-hand side of nodal equation in real and imaginary parts
   I_real[3] = [ 1  0  0 ];
   I_imag[3] = [ 0  0  0 ];

   ! Create an expanded right-hand side vector
```

```
   I[6] = [ I_real I_imag ];

   ! Calculate the nodal voltages by matrix inversion
   Y_INV[6,6] = INVERSE(Y);
   V_INV[6] = PRODUCT(Y_INV,I);

   ! Calculate the nodal voltages by LU factorization
   Y_LU[7,6] = LU(Y);
   V_LU[6] = SUBST(Y_LU,I);

   ! Convert the voltage at Node 3 from real and imaginary to magnitude and phase
   RI2MP(V_INV[3],V_INV[6],MV3_INV,PV3_INV);
   RI2MP(V_LU[3],V_LU[6],MV3_LU,PV3_LU);

   ! Calculate the voltage gain
   Gain_INV = MV3_INV / V1;
   Gain_LU = MV3_LU / V1;

   ! Calculate the voltage phase shift
   Phase_INV = PV3_INV - 0;
   Phase_LU = PV3_LU - 0;
End

! Calculate the voltage gain by direct circuit simulation

Model
   VSOURCE 1 0 NAME = Vinp V = V1;
   RES 1 2 R = R1;
   CAP 2 0 C = C1;
   RES 2 3 R = R2;
   CAP 3 0 C = C2;
   RES 3 4 R = R3;
   CAP 4 0 C = C3;
   VLABEL 4 0 NAME = V3;
   CIRCUIT NAME = Q1042;
   Gain_CKT = MV3[1] / V1;
   Phase_CKT = PV3[1] - 0;
End

! Define the sweep parameter and output responses

Sweep
   HB: Omega: from 0 to 5 step 0.2 FREQ = (1HZ * Omega / (2 * PI)) Gain_CKT Gain_INV Gain_LU
   {Xsweep
        Title = "Voltage gain vs Frequency"
        X_title = "Omega (rad/sec)"
        X = Omega
        Y_title = "Gain"
        Y = Gain_CKT.red & Gain_INV.blue.dot & Gain_LU.yellow.square};

   HB: Omega: from 0 to 5 step 0.2 FREQ = (1HZ * Omega / (2 * PI)) Phase_CKT Phase_INV Phase_LU
   {Xsweep
        Title = "Phase Shift vs Frequency"
        X_title = "Omega (rad/sec)"
        X = Omega
        Y_title = "Phase Shift (degrees)"
        Y = Phase_CKT.red & Phase_INV.blue.dot & Phase_LU.yellow.square};
End

! Define report block

Report
                    Gain versus Frequency

   -----------------------------------------------------
                             Gain
           ---------------------------------------------
             From matrix     From LU      From circuit
      Omega   inversion    factorization   simulation
   -----------------------------------------------------
${$%2.1f$      $Omega$ $%5.4f%$      $Gain_INV$  $%5.4f$        $Gain_LU$  $%5.4f$        $Gain_CKT$}$
```

```
-----------------------------------------------------

              Phase Shift versus Frequency

-----------------------------------------------------
                   Phase Shift in degrees
        ---------------------------------------------
         From matrix      From LU      From circuit
Omega     inversion    factorization    simulation
-----------------------------------------------------
${$%2.1f$     $Omega$ $%7.2f%$     $Phase_INV$  $%7.2f$     $Phase_LU$  $%7.2f$     $Phase_CKT$}$
-----------------------------------------------------
End
```

The numerical and graphical results are shown in the report and Figs. SQ1042.1 and SQ1042.2, respectively.

Report for Question 1042

Gain versus Frequency

	Gain		
Omega	From matrix inversion	From LU factorization	From circuit simulation
0.0	1.0000	1.0000	1.0000
0.2	0.4265	0.4265	0.4265
0.4	0.2075	0.2075	0.2075
0.6	0.1209	0.1209	0.1209
0.8	0.0768	0.0768	0.0768
1.0	0.0515	0.0515	0.0515
1.2	0.0360	0.0360	0.0360
1.4	0.0259	0.0259	0.0259
1.6	0.0192	0.0192	0.0192
1.8	0.0146	0.0146	0.0146
2.0	0.0113	0.0113	0.0113
2.2	0.0089	0.0089	0.0089
2.4	0.0071	0.0071	0.0071
2.6	0.0058	0.0058	0.0058
2.8	0.0047	0.0047	0.0047
3.0	0.0039	0.0039	0.0039
3.2	0.0033	0.0033	0.0033
3.4	0.0028	0.0028	0.0028
3.6	0.0024	0.0024	0.0024
3.8	0.0021	0.0021	0.0021
4.0	0.0018	0.0018	0.0018
4.2	0.0015	0.0015	0.0015
4.4	0.0014	0.0014	0.0014
4.6	0.0012	0.0012	0.0012
4.8	0.0011	0.0011	0.0011
5.0	0.0009	0.0009	0.0009

Phase Shift versus Frequency

Omega	Phase Shift in degrees: From matrix inversion	From LU factorization	From circuit simulation
0.0	0.00	0.00	0.00
0.2	-85.11	-85.11	-85.11
0.4	-117.16	-117.16	-117.16
0.6	-138.57	-138.57	-138.57
0.8	-154.99	-154.99	-154.99
1.0	-168.11	-168.11	-168.11
1.2	-178.81	-178.81	-178.81
1.4	172.32	172.32	172.32
1.6	164.88	164.88	164.88
1.8	158.56	158.56	158.56
2.0	153.15	153.15	153.15
2.2	148.47	148.47	148.47
2.4	144.39	144.39	144.39
2.6	140.81	140.81	140.81
2.8	137.65	137.65	137.65
3.0	134.84	134.84	134.84
3.2	132.33	132.33	132.33
3.4	130.07	130.07	130.07
3.6	128.04	128.04	128.04
3.8	126.19	126.19	126.19
4.0	124.51	124.51	124.51
4.2	122.97	122.97	122.97
4.4	121.56	121.56	121.56
4.6	120.27	120.27	120.27
4.8	119.07	119.07	119.07
5.0	117.96	117.96	117.96

From the report and Figs. SQ1042.1 and SQ1042.2, we can see that the results obtained by matrix inversion, LU factorization and direct circuit simulation are identical.

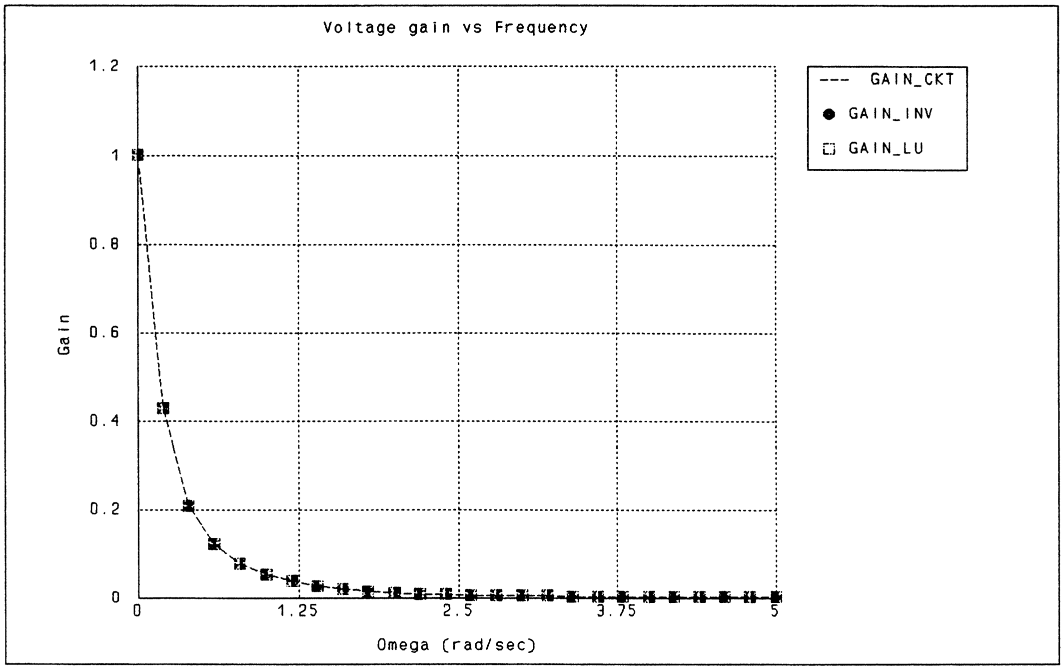

Fig. SQ1042.1 Voltage gain versus frequency.

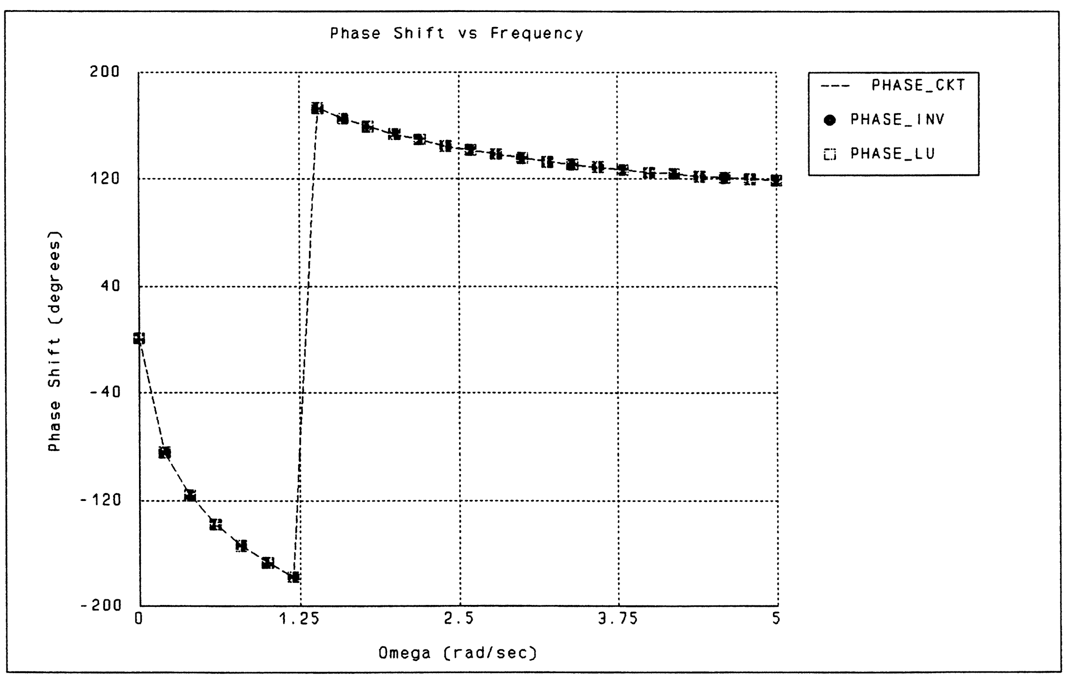

Fig. SQ1042.2 Phase shift versus frequency.

OSA☞ Question 1045 Use OSA90/hope to calculate and plot the reflection coefficient of the circuit shown in Fig. Q1044, where C_1 = 1.0 F, C_2 = 0.125 F, L = 2.0 H, $0 \le \omega \le 4$ rad/s. (See Question 1044.)

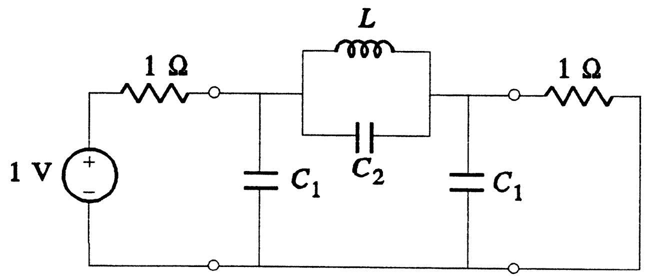

Fig. Q1044 LC filter network.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 1045

```
! File name: Q1045.ckt
! Circuit file for solving Question 1045

Model
   ! Assign values to the circuit elements
   C1 = 1F;
   C2 = 0.125F;
   L = 2H;

   ! Define the circuit
   PORT 1 0 NAME = Vin V = 1V R = 1OH;
   CAP 1 0 C = C1;
   IND 1 2 L = L;
   CAP 1 2 C = C2;
   CAP 2 0 C = C1;
   RES 2 0 R = 1OH;
   CIRCUIT NAME = Q1045;

   Omega = 1;    ! Initialize Omega

   ! Calculate the reflection coefficient
   RefCoef = MS11;
end

Sweep
   AC: Omega: FROM 0 TO 4 n = 100 FREQ = (1HZ*Omega/(2*PI)) RefCoef
   {Xsweep  Title = "Sweep of reflection coefficient versus frequency"
    X_title = "Omega (rad/sec)"  X = Omega
    Y_title = "Reflection coefficient"  Y = Refcoef.blue & Refcoef.red.point};
end
```

The results obtained are plotted in Fig. SQ1045.

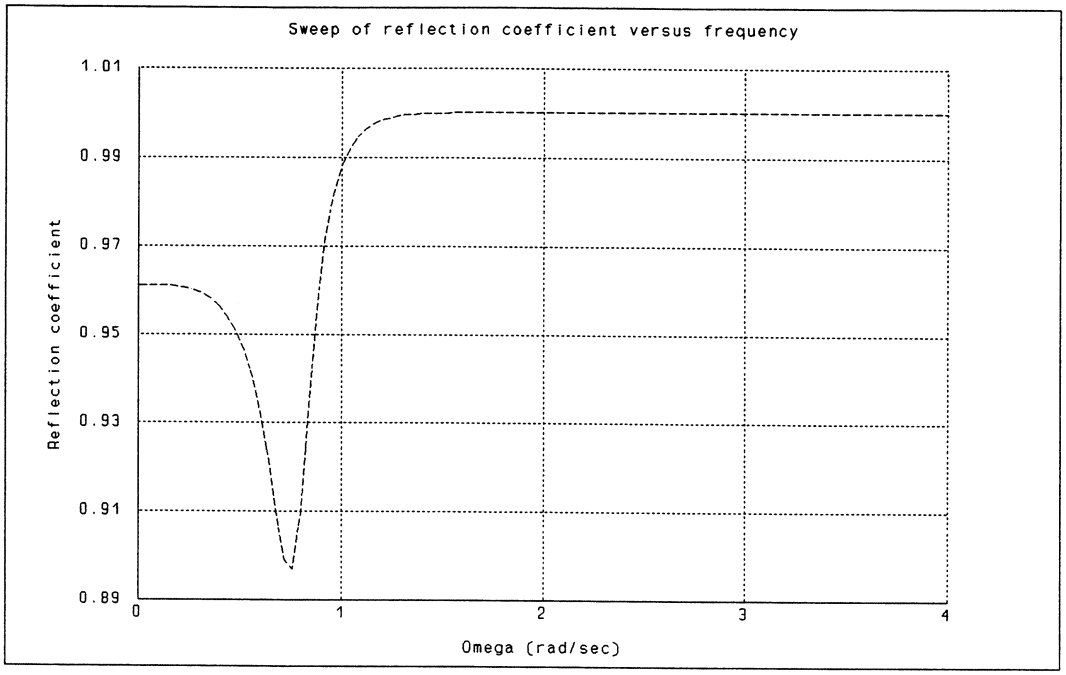

Fig. SQ1045 Sweep of reflection coefficient versus frequency.

4.2.2 Minimization

OSA☞ Question 2031 (p. 4-41)
OSA☞ Question 2032 (p. 4-46)
OSA☞ Question 2067 (p. 4-52)
OSA☞ Question 2073 (p. 4-55)

OSA☛ Question 2031 Use OSA90/hope and the functions given in Question 2029 to verify that the solution is $y^0 = [1.125 \ \ 1.5]^T$ starting at $y^0 = [1 \ \ 1]^T$. Working directly with these functions, construct the Newton equations to solve the nonlinear system and construct the formulas to calculate $\partial f/\partial x$ subject to the given constraints. Let $x = [1 \ \ 1.25]^T$, where T denotes transposition. Check derivatives by perturbation. (See Question 2029.)

$$h_1 = 4\ x_1\ y_1 - 3\ y_2 = 0$$
$$h_2 = -x_1\ y_1\ y_2 + 2\ x_2^2\ y_2 - 3 = 0$$
$$f = y_1^2 + y_1\ x_1$$

Solution

The circuit file for solving the nonlinear system and calculating $\partial f/\partial x$ follows.

Circuit file for solving Question 2031

```
!  File name: Q2031_adjoint.ckt
!  Circuit file for solving the nonlinear equations and df/dX of Q2031

Expression
! Y_j+1
   Y10: 1;       ! initialize to Y_0 for first sweep iteration
   Y20: 1;
! Y_j
   Y1: 0;        ! initialize to anything
   Y2: 0;

! X starting points
   X1 : 1;
   X2 : 1.25;

! objective function
   f = Y1 * Y1 + Y1 * X1;
! its gradient w.r.t. Y
   grad_f_Y[2] = [ (2 * Y1 + X1)  (0) ];
! and its gradient w.r.t. X
   grad_f_X[2] = [ Y1  0 ];

! nonlinear system of constraints
   h1 = 4 * X1 * Y1 - 3 * Y2;
   h2 = -X1 * Y1 * Y2 + 2 * X2 * X2 * Y2 - 3;
   h[2] = [ h1  h2 ];
! its Jacobian w.r.t. Y
   jacobian_Y[2,2] = [ (4 * X1)    (-3)
                       (-X1 * Y2)  (2 * X2 * X2 - X1 * Y1) ];
! and its Jacobian w.r.t. X
   jacobian_X[2,2] = [ (4 * Y1)    (0)
                       (-Y1 * Y2)  (4 * X2 * Y2) ];

! find df/dY with present Y = Y_j  (i.e. find Y_j+1)
   df_dY[2] = SOLVE(jacobian_Y,h);

! calculate (df/dX) s.t.(h=0)
! find Y^
   jacobian_Y_transpose[2,2] = TRANSPOSE(jacobian_Y);
   Y_hat[2] = SOLVE(jacobian_Y_transpose,grad_f_Y);
! df/dX
   jacobian_X_transpose[2,2] = TRANSPOSE(jacobian_X);
   templ[2] = product(jacobian_X_transpose,Y_hat);
   df_dX[2] = grad_f_X - templ;

! initialize sweep parameter
   K: 1;
End
```

```
Sweep
! Newton iterations for solving nonlinear system
   K: from 1 to 5 step=1
   Y1=Y10  Y2=Y20
   Y10=(Y10 - df_dY[1])  Y20=(Y20 - df_dY[2])
   Y1 Y2 f h;

! Adjoint method for calculating df/dX
   K: from 1 to 5 step=1
   Y1=Y10  Y2=Y20
   Y10=(Y10 - df_dY[1])  Y20=(Y20 - df_dY[2])
   df_dX Y_hat;
End

Report

           Iterations of Newton's Method for solving nonlinear system for Y1 and Y2

        --------------------------------------------------------------------------------
            K            Y1            Y2             f             h1             h2
        --------------------------------------------------------------------------------
${
$%12g$ $K$  $Y1$ $Y2$ $f$   $h[1]$   $h[2]$
}$
        --------------------------------------------------------------------------------


                      Calculating df/dX using the Adjoint Method

            -----------------------------------------------------------------
                 K        df/dX1        df/dX2           Y^1           Y^2
            -----------------------------------------------------------------
${
$%12g$      $K$  $df_dX$ $Y_hat$
}$
            -----------------------------------------------------------------

End
```

The report follows.

Report for Question 2031

Iterations of Newton's Method for solving nonlinear system for Y1 and Y2

K	Y1	Y2	f	h1	h2
1	1	1	2	1	-0.875
2	1.09091	1.45455	2.28099	0	-0.0413223
3	1.12377	1.49836	2.38662	0	-0.00143959
4	1.125	1.5	2.39062	0	-2.01875e-06
5	1.125	1.5	2.39062	0	-3.99236e-12

Calculating df/dX using the Adjoint Method

K	df/dX1	df/dX2	Y^1	Y^2
1	-2	-8.18182	1.15909	1.63636
2	-2.38017	-18.4009	1.7155	2.53012
3	-2.52571	-20.7956	1.85166	2.77578
4	-2.53124	-20.8927	1.85714	2.7857
5	-2.53125	-20.8929	1.85714	2.78571

From the report, the solution has been reached in just 4 iterations.

The circuit file used to compare the above adjoint method results for *f* and its derivatives with results obtained by perturbation follows.

Circuit file for comparison in Question 2031

```
!  File name: Q2031_perturb.ckt
!  Circuit file for comparing the derivatives acquired using
!    adjoint and perturbation methods for Q2031

Expression
! Y_j+1
   Y10: 1;        ! (initialize to Y_0 for first sweep iteration)
   Y20: 1;
! Y_j
   Y1: 0;         ! (initialize to anything)
   Y2: 0;

! original and perturbed X parameters
   N: 0;
   M: 0;
   X1_orig = 1;
   X2_orig = 1.25;
! X is constant when solving nonlinear system
   X1 =  X1_orig + N / 100 * X1_orig;
   X2 =  X2_orig + M / 100 * X2_orig;

! objective function
   f = Y1 * Y1 + Y1 * X1;
! its gradient w.r.t. Y
   grad_f_Y[2] = [ (2 * Y1 + X1)  (0) ];
! and its gradient w.r.t. X
   grad_f_X[2] = [ Y1  0 ];

! nonlinear system of constraints
   h1 = 4 * X1 * Y1 - 3 * Y2;
   h2 = -X1 * Y1 * Y2 + 2 * X2 * X2 * Y2 - 3;
   h[2] = [ h1  h2 ];
! its Jacobian w.r.t. Y
   jacobian_Y[2,2] = [ (4 * X1)    (-3)
                      (-X1 * Y2)  (2 * X2 * X2 - X1 * Y1) ];
! and its Jacobian w.r.t. X
   jacobian_X[2,2] = [ (4 * Y1)    (0)
                      (-Y1 * Y2)  (4 * X2 * Y2) ];

! find df/dY with present Y = Y_j  (i.e. find Y_j+1)
   df_dY[2] = SOLVE(jacobian_Y,h);

! calculate (df/dX) s.t.(h=0)
! find Y^
   jacobian_Y_transpose[2,2] = TRANSPOSE(jacobian_Y);
   Y_hat[2] = SOLVE(jacobian_Y_transpose,grad_f_Y);
! df/dX
   jacobian_X_transpose[2,2] = TRANSPOSE(jacobian_X);
   temp1[2] = product(jacobian_X_transpose,Y_hat);
   df_dX[2] = grad_f_X - temp1;

! compare adjoint with perturbation by using predetermined adjoint values
   f_adjoint = 2.39062;
   df_dX1_adjoint = -2.53125;
   df_dX2_adjoint = -20.8929;
   df_dX1_perturb = if (X1 = X1_orig) (-2.53125) else ((f_adjoint - f) / (X1_orig - X1));
   df_dX2_perturb = if (X2 = X2_orig) (-20.8929) else ((f_adjoint - f) / (X2_orig - X2));

! initialize sweep parameter
   K: 1;
End

Sweep
```

```
! Graphical comparison of Adjoint and Perturbation methods for determining df/dX1
  M: 0
  N: from -1 to 1 step=0.01
    K: from 1 to 10 step=1
      Y1=Y10  Y2=Y20
      Y10=(Y10 - df_dY[1])  Y20=(Y20 - df_dY[2])
      df_dX1_perturb df_dX1_adjoint
  {Xsweep
        Title="Calculating df/dX1 by Adjoint method versus by Perturbation"
        X = N
        X_Title="% change in X1"
        NXTicks = 10
        K = 10
        Y = df_dX1_adjoint.blue & df_dX1_perturb.red.circle
        Y_Title="adjoint and perturbed df/dX1"};

! Graphical comparison of Adjoint and Perturbation methods for determining df/dX2
  N: 0
  M: from -1 to 1 step=0.01
    K: from 1 to 10 step=1
      Y1=Y10  Y2=Y20
      Y10=(Y10 - df_dY[1])  Y20=(Y20 - df_dY[2])
      df_dX2_perturb df_dX2_adjoint
  {Xsweep
        Title="Calculating df/dX2 by Adjoint method versus by Perturbation"
        X = M
        X_Title="% change in X2"
        NXTicks = 10
        K = 10
        Y = df_dX2_adjoint.yellow & df_dX2_perturb.green.diamond
        Y_Title="adjoint and perturbed df/dX2"};
End
```

The following Figs. SQ2031.1 and SQ2031.2 show a comparison between the above results with derivatives obtained by perturbation. Note how the derivatives obtained by perturbation are distorted when the change in the X1 variable is very small.

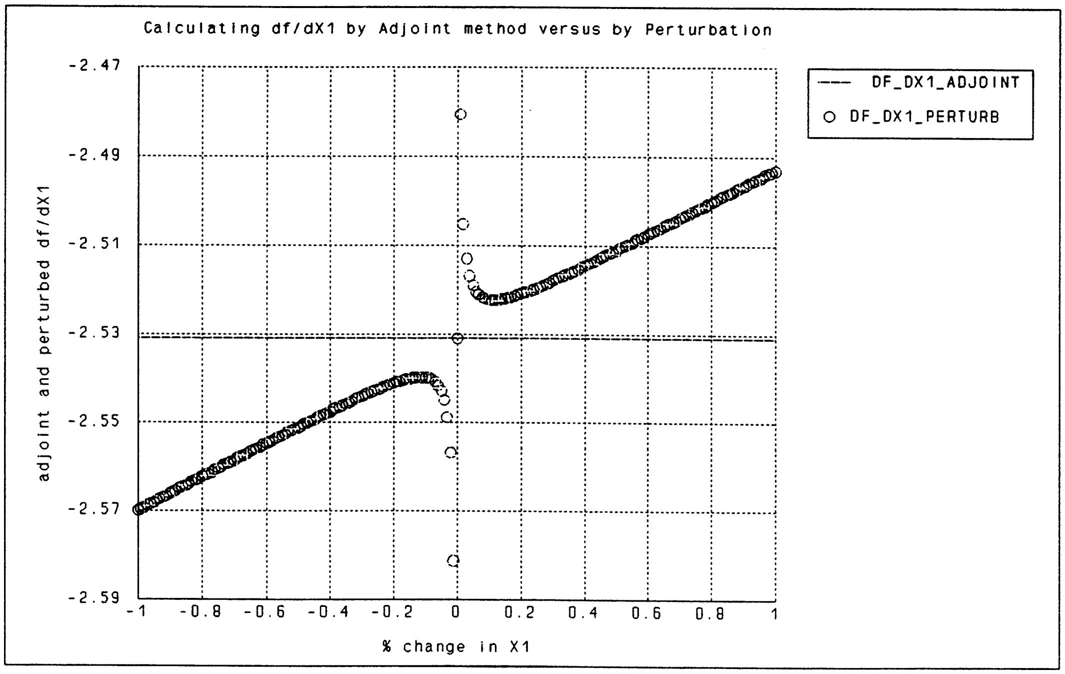

Fig. SQ2031.1 Comparison of adjoint and perturbation methods to obtain df/dX1.

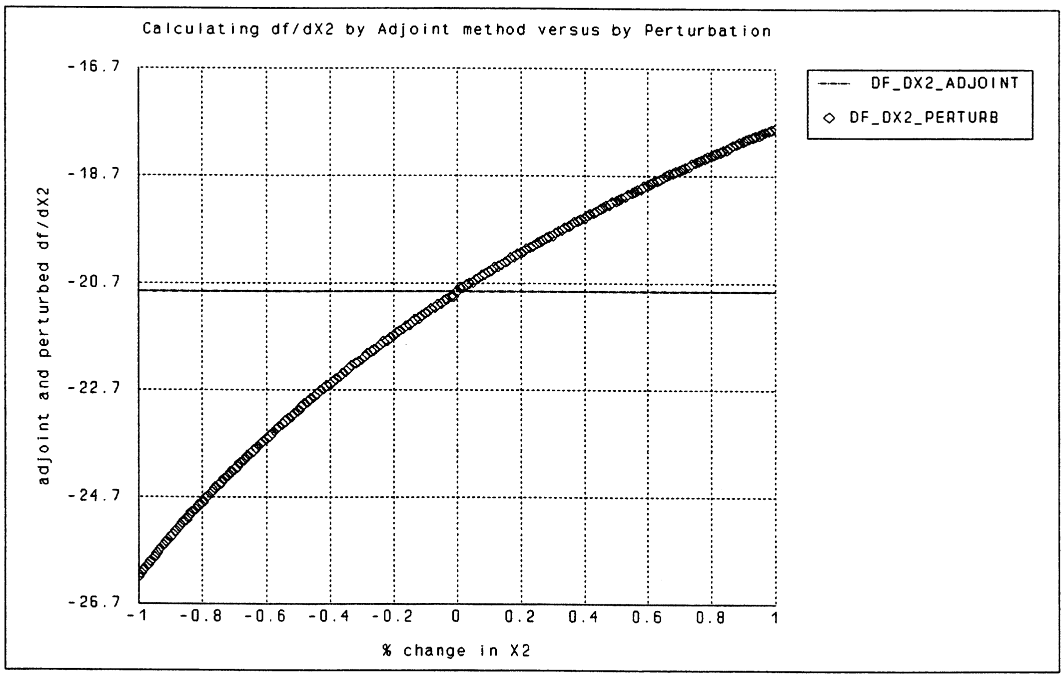

Fig. SQ2031.2 Comparison of adjoint and perturbation methods to obtain df/dX2.

OSA☞ Question 2032 Use OSA90/hope and the functions given in Question 2030 to verify that the solution is $y^0 = [1.125 \;\; 1.5]^T$ starting at $y^0 = [1 \;\; 1]^T$. Working directly with these functions, construct the Newton equations to solve the nonlinear system and construct the formulas to calculate $\partial f/\partial \boldsymbol{x}$ subject to the given constraints. Let $\boldsymbol{x} = [1 \;\; 1.25]^T$, where T denotes transposition. Check derivatives by perturbation. (See Question 2030.)

$$\begin{aligned} h_1 &= -x_1y_1y_2 + 2x_2^2y_2 - 3 = 0 \\ h_2 &= 4x_1y_1 - 3y_2 = 0 \\ f &= y_2 \end{aligned}$$

Solution

The circuit file for solving the nonlinear system and calculating $\partial f/\partial \boldsymbol{x}$ follows.

Circuit file for solving Question 2032

```
!  File name: Q2032_adjoint.ckt
!  Circuit file for solving the nonlinear equations and df/dX of Q2032

Expression
! Y_j+1
   Y10: 1;        ! initialize to Y_0 for first sweep iteration
   Y20: 1;
! Y_j
   Y1: 0;         ! initialize to anything
   Y2: 0;

! Starting X points
   X1 = 1;
   X2 = 1.25;

! objective function
   f = Y2;
! its gradient w.r.t. Y
   grad_f_Y[2] = [ 0  1 ];
! and its gradient w.r.t. X
   grad_f_X[2] = [ 0  0 ];

! nonlinear system of constraints
   h1 = -X1 * Y1 * Y2 + 2 * X2 * X2 * Y2 - 3;
   h2 = 4 * X1 * Y1 - 3 * Y2;
   h[2] = [ h1  h2 ];
! its Jacobian w.r.t. Y
   jacobian_Y[2,2] = [ (-X1 * Y2)   (-X1 * Y1 + 2 * X2 * X2)
                       (4 * X1)     (-3)                       ];
! and its Jacobian w.r.t. X
   jacobian_X[2,2] = [ (-Y1 * Y2)   (4 * X2 * Y2)
                       (4 * Y1)     (0)           ];

! find df/dY with present Y = Y_j  (i.e. find Y_j+1)
   df_dY[2] = SOLVE(jacobian_Y,h);

! calculate (df/dX) s.t.(h=0)
! find Y^
   jacobian_Y_transpose[2,2] = TRANSPOSE(jacobian_Y);
   Y_hat[2] = SOLVE(jacobian_Y_transpose,grad_f_Y);
! df/dX
   jacobian_X_transpose[2,2] = TRANSPOSE(jacobian_X);
   temp1[2] = product(jacobian_X_transpose,Y_hat);
   df_dX[2] = grad_f_X - temp1;

! initialize sweep parameter
   K: 1;
End
```

```
Sweep
! Newton iterations for solving nonlinear system
   K: from 1 to 10 step=1
   Y1=Y10  Y2=Y20
   Y10=(Y10 - df_dY[1])  Y20=(Y20 - df_dY[2])
   Y1 Y2 f h;

! Adjoint method for calculating df/dX
   K: from 1 to 10 step=1
   Y1=Y10  Y2=Y20
   Y10=(Y10 - df_dY[1])  Y20=(Y20 - df_dY[2])
   df_dX Y_hat;
End

Report

           Iterations of Newton's Method for solving nonlinear system for Y1 and Y2

        ------------------------------------------------------------------------------
            K            Y1            Y2             f             h1             h2
        ------------------------------------------------------------------------------
${
$%12g$ $K$  $Y1$ $Y2$ $f$   $h[1]$   $h[2]$
}$
        ------------------------------------------------------------------------------


                      Calculating df/dX using the Adjoint Method

            ----------------------------------------------------------------
                 K        df/dX1       df/dX2          Y^1           Y^2
            ----------------------------------------------------------------
${
$%12g$      $K$  $df_dX$ $Y_hat$
}$
            ----------------------------------------------------------------

End
```

The report follows.

Report for Question 2032

Iterations of Newton's Method for solving nonlinear system for Y1 and Y2

K	Y1	Y2	f	h1	h2
1	1	1	1	-0.875	1
2	1.09091	1.45455	1.45455	-0.0413223	0
3	1.12377	1.49836	1.49836	-0.00143959	0
4	1.125	1.5	1.5	-2.01875e-06	0
5	1.125	1.5	1.5	-3.99236e-12	0
6	1.125	1.5	1.5	4.44089e-16	-8.88178e-16
7	1.125	1.5	1.5	-8.88178e-16	8.88178e-16
8	1.125	1.5	1.5	8.88178e-16	-8.88178e-16
9	1.125	1.5	1.5	-8.88178e-16	8.88178e-16
10	1.125	1.5	1.5	8.88178e-16	-8.88178e-16

Calculating df/dX using the Adjoint Method

K	df/dX1	df/dX2	Y^1	Y^2
1	1.11022e-16	-3.63636	0.727273	0.181818
2	0	-7.71084	1.06024	0.385542
3	0	-8.53799	1.13965	0.4269
4	2.22045e-16	-8.57138	1.14285	0.428569
5	0	-8.57143	1.14286	0.428571
6	-2.22045e-16	-8.57143	1.14286	0.428571
7	2.22045e-16	-8.57143	1.14286	0.428571
8	2.22045e-16	-8.57143	1.14286	0.428571
9	2.22045e-16	-8.57143	1.14286	0.428571
10	2.22045e-16	-8.57143	1.14286	0.428571

From the report, the solution has been reached in just 4 iterations.

The circuit file used to compare the above adjoint method results for f and its derivatives with results obtained by perturbation follows.

Circuit file for comparison in Question 2032

```
!  File name: Q2032_perturb.ckt
!  Circuit file for comparing the derivatives acquired using
!   adjoint and perturbation methods for Q2032

Expression
! Y_j+1
   Y10: 1;       ! (initialize to Y_0 for first sweep iteration)
   Y20: 1;
! Y_j
   Y1: 0;        ! (initialize to anything)
   Y2: 0;

! original and perturbed X parameters
   N: 0;
   M: 0;
   X1_orig = 1;
   X2_orig = 1.25;
! X is constant when solving nonlinear system
   X1 =  X1_orig + N / 100 * X1_orig;
   X2 =  X2_orig + M / 100 * X2_orig;

! objective function
   f = Y2;
```

```
! its gradient w.r.t. Y
   grad_f_Y[2] = [ 0  1 ];
! and its gradient w.r.t. X
   grad_f_X[2] = [ 0  0 ];

! nonlinear system of constraints
   h1 = -X1 * Y1 * Y2 + 2 * X2 * X2 * Y2 - 3;
   h2 = 4 * X1 * Y1 - 3 * Y2;
   h[2] = [ h1  h2 ];
! its Jacobian w.r.t. Y
   jacobian_Y[2,2] = [ (-X1 * Y2)   (-X1 * Y1 + 2 * X2 * X2)
                       (4 * X1)     (-3)                      ];
! and its Jacobian w.r.t. X
   jacobian_X[2,2] = [ (-Y1 * Y2)   (4 * X2 * Y2)
                       (4 * Y1)     (0)           ];

! find df/dY with present Y = Y_j  (i.e. find Y_j+1)
   df_dY[2] = SOLVE(jacobian_Y,h);

! calculate (df/dX) s.t.(h=0)
! find Y^
   jacobian_Y_transpose[2,2] = TRANSPOSE(jacobian_Y);
   Y_hat[2] = SOLVE(jacobian_Y_transpose,grad_f_Y);
! df/dX
   jacobian_X_transpose[2,2] = TRANSPOSE(jacobian_X);
   temp1[2] = product(jacobian_X_transpose,Y_hat);
   df_dX[2] = grad_f_X - temp1;

! compare adjoint with perturbation by using predetermined adjoint values
   f_adjoint = 1.5;
   df_dX1_adjoint = 0;
   df_dX2_adjoint = -8.57143;
   df_dX1_perturb = if (X1 = X1_orig) (0) else (f_adjoint - f) / (X1_orig - X1);
   df_dX2_perturb = if (X2 = X2_orig) (-8.57143) else ((f_adjoint - f) / (X2_orig - X2));

! initialize sweep parameter
   K: 1;
End

Sweep
! Graphical comparison of Adjoint and Perturbation methods for determining df/dX1
   M: 0
   N: from -1 to 1 step=0.01
   K: from 1 to 10 step=1
   Y1=Y10  Y2=Y20
   Y10=(Y10 - df_dY[1])  Y20=(Y20 - df_dY[2])
   df_dX1_perturb df_dX1_adjoint
   {Xsweep
        Title="Calculating df/dX1 by Adjoint method versus by Perturbation"
        X = N
        X_Title="% change in X1"
        NXTicks = 10
        K = 10
        Y = df_dX1_adjoint.blue & df_dX1_perturb.red.square
        Y_Title="adjoint and perturbed df/dX1"
        Ymin = -0.0003  Ymax = 0.0003
        NYTicks = 6};

! Graphical comparison of Adjoint and Perturbation methods for determining df/dX2
   N: 0
   M: from -1 to 1 step=0.01
   K: from 1 to 10 step=1
   Y1=Y10  Y2=Y20
   Y10=(Y10 - df_dY[1])  Y20=(Y20 - df_dY[2])
   df_dX2_perturb df_dX2_adjoint
   {Xsweep
        Title="Calculating df/dX2 by Adjoint method versus by Perturbation"
        X = M
        X_Title="% change in X2"
        NXTicks = 10
        K = 10
```

```
        Y = df_dX2_adjoint.yellow & df_dX2_perturb.green.diamond
        Y_Title="adjoint and perturbed df/dX2"};
End
```

The following Figs. SQ2032.1 and SQ2032.2 show a comparison between the above results with derivatives obtained by perturbation.

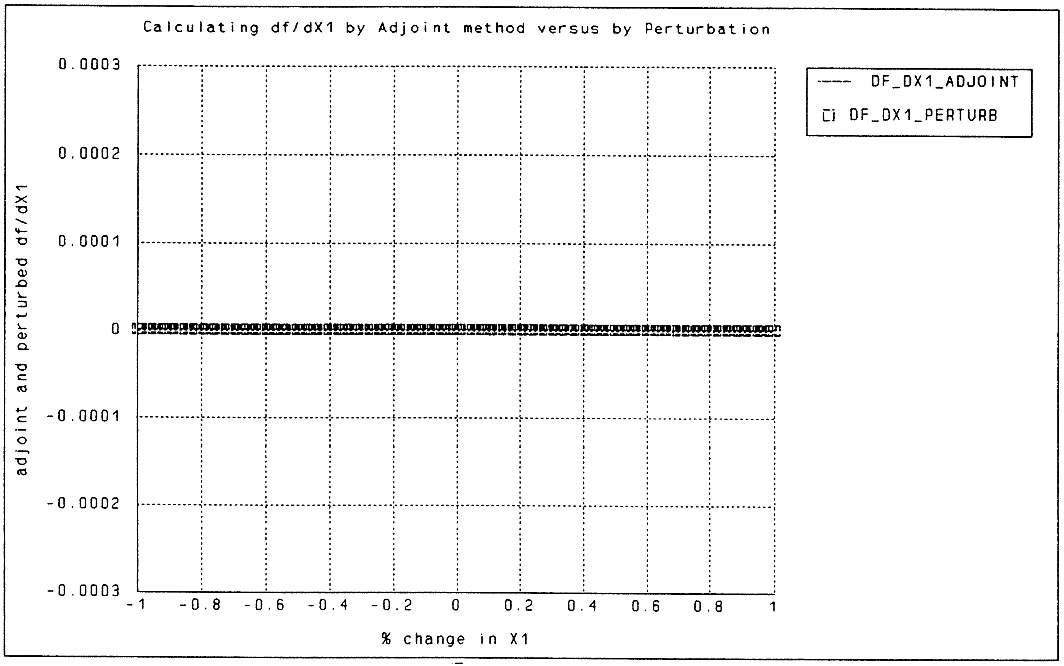

Fig. SQ2032.1 Comparison of adjoint and perturbation methods to obtain df/dX1.

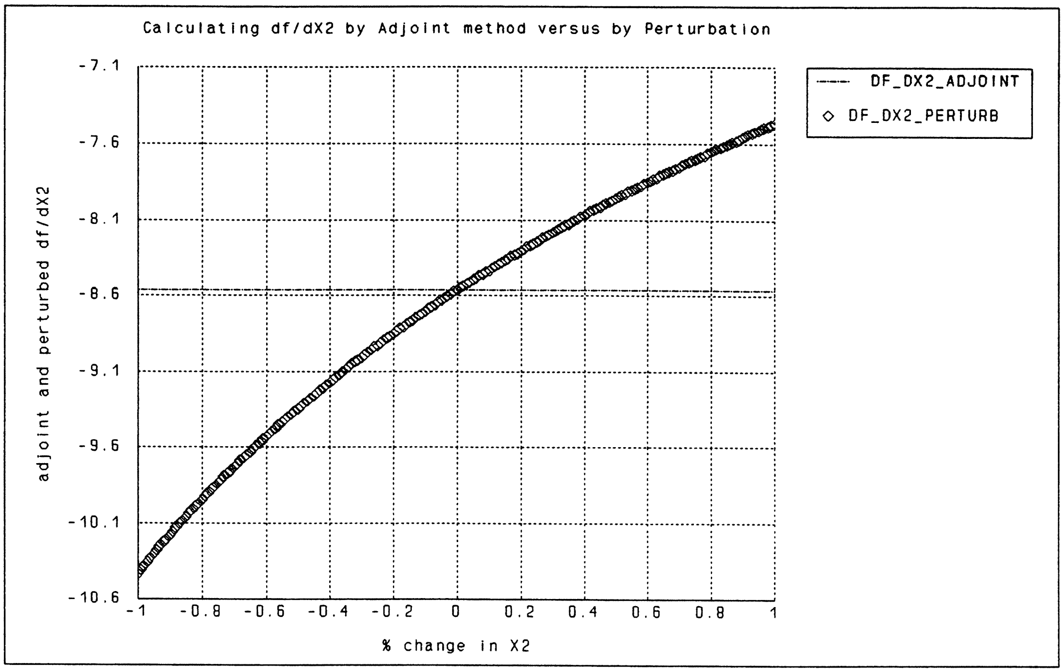

Fig. SQ2032.2 Comparison of adjoint and perturbation methods to obtain df/dX2.

OSA☞ Question 2067 Use OSA90/hope to create a Fletcher-Powell-Davidon algorithm for minimizing the differentiable function given in Question 2066 and display the results of every iteration. (See Question 2066.)

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

Solution

To calculate the parameter α, a line search must be done. The method used in this algorithm incorporates the Hessian, and the current iteration's gradient and search direction. [Source: Luenberger, p.252.]

The circuit file for solving this problem follows.

Circuit file for Question 2067

```
! File name: Q2067.ckt
! Circuit file for solving Question 2067

Expression
   ! Set Sweep Block parameter
   K: 1;

   ! Define iteration solution points
   X1: 0;
   X2: 0;
   X[2] = [ X1  X2 ];

   ! Define intermediate iteration solution points
   X1_new: 0;
   X2_new: 0;
   X_new[2] = [ X1_new  X2_new ];

   ! Define Objective function, its Gradient and its Hessian
   U = X1 * X1 + 2 * X2 * X2 + X1 * X2 + 2 * X1 + 1;
   GRAD_U[2] = [ (2 * X1 + X2 + 2)  (4 * X2 + X1) ];
   HESS_U[2,2] = [ 2  1
                   1  4 ];

   ! Calculate search direction, s = -H * grad(U)

   ! Calculate the new Hessian approximation, H
   ! H = H_old + (delta.delta_T)/(delta_T.g) - (H_old.g.g_T.H_old)/(g_T.H_old.g)
   !   = H_old + H1 - H2

   ! calculate "delta"
   DELTA_X1: 0;
   DELTA_X2: 0;
   DELTA_X[2] = [ DELTA_X1  DELTA_X2 ];

   ! find grad(U) from previous iteration to calculate "g"
   G01: 0;
   G02: 0;
   GRAD_old[2]: [ G01  G02 ];

   g[2] = GRAD_U - GRAD_old;

   ! Define the old Hessian approximation
   H011: 1;
   H012: 0;
   H021: 0;
   H022: 1;
   H_old[2,2]: [ H011  H012
                 H021  H022 ];

   ! Calculate H2
   H21[2] = PRODUCT(H_old,g);
```

```
   H21_m[2,1] = [ H21[1]  H21[2] ];                        ! m = "matrix"
   H22[2] = PRODUCT(g,H_old);
   H22_m[1,2] = [ H22[1]  H22[2] ];
   H2_num[2,2] = PRODUCT(H21_m,H22_m);
   H2_den = PRODUCT(g,H21);
   H2[2,2] = if (K = 1) (0) else (
                                  if (H2_den = 0) (0) else (H2_num / H2_den)
                                );

   ! Calculate H1
   DELTA_X_m[2,1] = [ DELTA_X[1]  DELTA_X[2] ];
   DELTA_X_m_T[1,2] = TRANSPOSE(DELTA_X_m);                ! T = "transpose"
   H1_num[2,2] = PRODUCT(DELTA_X_m,DELTA_X_m_T);
   H1_den = PRODUCT(DELTA_X,g);
   H1[2,2] = if (K = 1) (0) else (
                                  if (H1_den = 0) (0) else (H1_num / H1_den)
                                );

   ! Calculate new Hessian approximation
   H[2,2] = H_old + H1 - H2;

   ! Calculate search direction
   S_neg[2] = PRODUCT(H,GRAD_U);                   ! neg = "negative"
   S[2] = -S_neg;

   ! Minimize function U(X + DELTA_X) with respect to alpha
   ALFA_num = PRODUCT(GRAD_U,S);
   ALFA2[2] = PRODUCT(HESS_U,S);
   ALFA_den = PRODUCT(S,ALFA2);
   ALFA = if (ALFA_den = 0) (0) else ( -ALFA_num / ALFA_den);
End

Sweep
   K: from 1 to 4 step = 1                ! K=1 is starting point
   X1 = X1_new  X2 = X2_new

   ! Prepare variables for next iteration
   X1_new = (X1_new + ALFA * S[1])  X2_new = (X2_new + ALFA * S[2])        ! X_new = X_j+1
   DELTA_X1 = (ALFA * S[1])   DELTA_X2 = (ALFA * S[2])
   H011 = H[1,1]   H012 = H[1,2]   H021 = H[2,1]   H022 = H[2,2]
   G01 = GRAD_U[1]   G02 = GRAD_U[2]

   X  U  GRAD_U;
End

Report
   Fletcher-Powell-Davidon Method for Minimizing U = X1^2 + 2 * X2^2 + x1 * X2 + 2 * X1 + 1

      ---------------------------------------------------------------------------------
          K             X1          X2              U              dU/dX1      dU/dX2
      ---------------------------------------------------------------------------------
${
   $%6g$ $K$ $%10g$    $X1$ $X2$    $U$    $GRAD_U[1]$ $GRAD_U[2]$
}$
      ---------------------------------------------------------------------------------
End
```

The results are shown in the following report.

Report for Question 2067

Fletcher-Powell-Davidon Method for Minimizing U = X1^2 + 2 * X2^2 + x1 * X2 + 2 * X1 + 1

K	X1	X2	U	dU/dX1	dU/dX2
1	0	0	1	2	0
2	-1	0	0	0	-1
3	-1.14286	0.285714	-0.142857	2.22045e-16	2.22045e-16
4	-1.14286	0.285714	-0.142857	0	0

OSA☞ Question 2073 Use OSA90/hope to create a conjugate gradient algorithm for minimizing the differentiable function given in Question 2072 and display the results of every iteration. (See Question 2072.)

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

Solution

To calculate the parameter α, a line search must be done. The method used in this algorithm incorporates the Hessian, and the current iteration's gradient and search direction. [Source: Luenberger, p.252.]

The circuit file for solving this problem follows.

Circuit file for Question 2073

```
! File name: Q2073.ckt
! Circuit file for solving Question 2073

Expression
   ! Define intermediate iteration solution points
   X1_new: 0;
   X2_new: 0;
   X_new[2] = [ X1_new  X2_new ];

   ! Define iteration solution points
   X1: 0;
   X2: 0;
   X[2] = [ X1  X2 ];

   ! Define Objective function, its Gradient and its Hessian at the jth iteration
   U = X1 * X1 + 2 * X2 * X2 + X1 * X2 + 2 * X1 + 1;
   GRAD_U[2] = [ (2 * X1 + X2 + 2)  (4 * X2 + X1) ];
   HESS_U[2,2] = [ 2  1
                   1  4 ];

   ! Define gradient vector from previous iteration needed for Beta calculation
   ! and initialize to anything but zero to avoid a floating point error in first iteration
   G01: 1;
   G02: 1;
   GRAD_old[2] = [ G01  G02 ];

   ! Calculate Beta parameter
   BETA_num = PRODUCT(GRAD_U,GRAD_U);
   BETA_den = PRODUCT(GRAD_old,GRAD_old);
   BETA = BETA_num / BETA_den;           ! BETA = (G_T.G)_j / (G_T.G)_j-1

   ! Define search direction from previous iteration needed for new search
   ! direction calculation and initialize to zero to make s_0 = -(GRAD_U)_0 by definition
   S01: 0;
   S02: 0;
   S_old[2] = [ S01  S02 ];

   ! Calculate new search direction
   S[2] = [ ( -GRAD_U[1] + BETA * S_old[1] )   ( -GRAD_U[2] + BETA * S_old[2] ) ];

   ! Minimize function U(X + DELTA_X) with respect to alpha
   ALFA_num = PRODUCT(GRAD_U,S);
   ALFA2[2] = PRODUCT(HESS_U,S);
   ALFA_den = PRODUCT(S,ALFA2);
   ALFA = -ALFA_num / ALFA_den;          ! ALFA = -(G_T.S)/(S_T.H.S)

   ! Initialize Sweep Block paramater
   K: 1;
End
```

```
Sweep
   K: from 1 to 3 step = 1                                          ! K=1 is starting point
   X1 = X1_new    X2 = X2_new                                       ! X = X_j

   ! Prepare variables for next iteration
   X1_new = (X1_new + (ALFA * S[1]))   X2_new = (X2_new + (ALFA*S[2]))     ! X_new = X_j+1
   S01 = (S[1])         S02 = (S[2])                                ! S_old = S_j-1
   G01 = (GRAD_U[1])  G02 = (GRAD_U[2])                             ! GRAD_old = GRAD_U_j-1

   X1 X2 U GRAD_U;
End

Report
     Conjugate Gradient Method for Minimizing U = X1^2 + 2 * X2^2 + X1 * X2 + 2 * X1 + 1

   ---------------------------------------------------------------------------------------
     K                 X1         X2                 U              dU/dX1     dU/dX2
   ---------------------------------------------------------------------------------------
${ $%6g$ $K$ $%10g$       $X1$ $X2$          $U$        $GRAD_U[1]$ $GRAD_U[2]$ }$
   ---------------------------------------------------------------------------------------
End
```

The results are shown in the following report.

Report for Question 2073

Conjugate Gradient Method for Minimizing U = X1^2 + 2 * X2^2 + X1 * X2 + 2 * X1 + 1

K	X1	X2	U	dU/dX1	dU/dX2
1	0	0	1	2	0
2	-1	0	0	0	-1
3	-1.14286	0.285714	-0.142857	0	0

4.2.3 Sensitivities

OSA☞ Question 3033 Consider the resistive network of Fig. Q3032. $G_1 = G_3 = G_5 = 1$ S, $R_2 = R_4 = 0.5\ \Omega$.

(a) Use OSA90/hope to calculate the node voltages of the original circuit by LU factorization of the nodal admittance matrix. Verify that

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

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

(b) Calculate the node voltages of the adjoint circuit using the LU factors already obtained above.

(c) Calculate the node voltages directly from the original and adjoint circuits.

(d) Calculate ∇V, where

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

using sensitivity formulas shown in the table of Question 3032. (See Question 3032.)

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

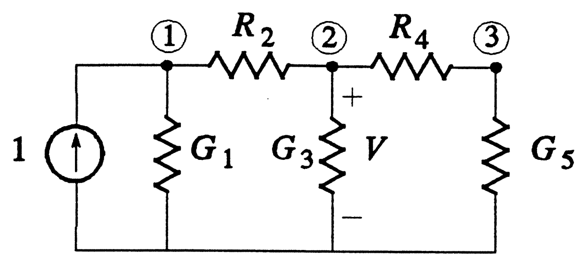

Fig. Q3032 Three–node resistive network.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 3033

```
!  File name: Q3033.ckt
!  Circuit file for solving Question 3033

! Illustrate that the units are not the microwave units

Control
  Non_Microwave_Units;
End

Expression
   ! Assign values to circuit elements
   G1 = 1/OH;
   R2 = 1/2OH;
   G3 = 1/OH;
   R4 = 1/2OH;
   G5 = 1/OH;

   ! Set up the conductance matrix
   G[3,3] = [  3  -2   0
              -2   5  -2
               0  -2   3  ];

   ! L and U matrix to be checked
   L[3,3] = [  3     0      0
              -2  (11/3)    0
               0    -2   (21/11) ];
   U[3,3] = [  1   (-2/3)    0
               0     1    (-6/11)
               0     0       1    ];

   ! Multiplication of L and U
   LxU[3,3] = PRODUCT(L,U);

   ! The right-hand side of the nodal equation for original circuit
   I[3]=[ 1   0   0 ];

   ! LU factorization of G and solving for node voltage V
   GLU[4,3] = LU(G);
   V[3] = SUBST(GLU,I);

   ! Extract L and U from GLU for comparison
   L_extract[3,3] = EXTRACT_L(GLU);
   U_extract[3,3] = EXTRACT_U(GLU);
```

```
   ! Solving the adjoint circuit
   Ihat[3] = [ 0  -1  0 ];
   Vhat[3] = SUBST(GLU,Ihat);    ! Since G transpose = G

   ! Calculate the Sensitivities
   GradV1 = V[1] * Vhat[1];
   GradV2 = -1 * (V[1] - V[2]) * (Vhat[1] - Vhat[2]) / R2^2;
   GradV3 = V[2] * Vhat[2];
   GradV4 = -1 * (V[2] - V[3]) * (Vhat[2] - Vhat[3]) / R4^2;
   GradV5 = V[3] * Vhat[3];
   GradV[5] = [ GradV1 GradV2 GradV3 GradV4 GradV5 ];

   j = 1;   ! Initialize the index for display
End

! Calculate node voltages from direct circuit simulation

Model
   ! Define resistances
   R1 = 1/G1;
   R3 = 1/G3;
   R5 = 1/G5;

   ! Define linear subcircuit
   SUBCIRCUIT  Q3032_CKT  1 2 3 0
   {
        RES 1 0 R = R1;
        RES 1 2 R = R2;
        RES 2 0 R = R3;
        RES 2 3 R = R4;
        RES 3 0 R = R5;
   };

   ! Define original circuit
   Q3032_CKT 1 2 3 0;
   ISOURCE 1 0 IDC = 1A;
   VLABEL 1 0 NAME = V1;
   VLABEL 2 0 NAME = V2;
   VLABEL 3 0 NAME = V3;
   V_NODE[3] = [ V1 V2 V3 ];

   ! Define adjoint circuit for calculating the sensitivities of V across
   ! G3 w.r.t. all the other resistors
   Q3032_CKT 4 5 6 0;
   ISOURCE 5 0 IDC = -1A;
   VLABEL 4 0 NAME = V4;
   VLABEL 5 0 NAME = V5;
   VLABEL 6 0 NAME = V6;
   V_NODE_HAT[3] = [ V4 V5 V6 ];

   CIRCUIT NAME = Q3033;
End

! Define the sweep parameter and output responses

Sweep
   j: from 1 to 3 step 1 LxU[j,1] LxU[j,2] LxU[j,3] G[j,1] G[j,2] G[j,3];
   j: from 1 to 3 step 1 L[j,1] L[j,2] L[j,3] U[j,1] U[j,2] U[j,3] L_extract[j,1] L_extract[j,2] L_extract[j,3]
U_extract[j,1] U_extract[j,2] U_extract[j,3];
   DC:  j: from 1 to 3 step 1 V[j] VHAT[j] V_NODE[j] V_NODE_HAT[j];
   GradV;
End

! Define report block
```

```
Report
      Comparison of Matrix G with LxU

   ------------------------------------
       Matrix G                LxU
   ------------------------------------
${$% 1g$     $G[j,1]$ $% 1g$  $G[j,2]$ $% 1g$  $G[j,3]$ $% 1g$       $LxU[j,1]$ $% 1g$  $LxU[j,2]$ $% 1g$
$LxU[j,3]$}$
   ------------------------------------


                    Comparison of Given and Extracted L and U matrices

   ----------------------------------------------------------------------------------
              Lower Matrix, L                                Upper Matrix, U
          given             extracted                    given             extracted
   ----------------------------------------------------------------------------------
${$% .1f$     $L[j,1]$  $L[j,2]$  $L[j,3]$      $L_extract[j,1]$  $L_extract[j,2]$  $L_extract[j,3]$
$U[j,1]$  $U[j,2]$  $U[j,3]$     $U_extract[j,1]$  $U_extract[j,2]$  $U_extract[j,3]$}$
   ----------------------------------------------------------------------------------


                           Voltage Values of the Original Circuit
                                  and the Adjoint Circuit

   -------------------------------------------------------------------------------------
       Node      Voltages of the original circuit        Voltages of the adjoint circuit
      number        from LU          direct from            from LU         direct from
                 decomposition        circuit            decomposition       circuit
   -------------------------------------------------------------------------------------
${$%1g$        $j$         $%5.4f$      $V[j]$            $V_NODE[j]$                   $VHAT[j]$
$V_NODE_HAT[j]$}$
   -------------------------------------------------------------------------------------


       Sensitivities of V w.r.t. Circuit Elements

   --------------------------------------------------
      dV/dG1     dV/dR2     dV/dG3     dV/dR4     dV/dG5
   --------------------------------------------------
$%5.4f$     $GRADV1$ $%5.4f$  $GRADV2$ $%5.4f$  $GRADV3$  $%5.4f$  $GRADV4$  $%5.4f$ $GRADV5$
   --------------------------------------------------
End
```

The results are shown in the report.

Report for Question 3033

Comparison of Matrix G with LxU

Matrix G			LxU		
3	-2	0	3	-2	0
-2	5	-2	-2	5	-2
0	-2	3	0	-2	3

Comparison of Given and Extracted L and U matrices

Lower Matrix, L						Upper Matrix, U					
given			extracted			given			extracted		
3.0	0.0	0.0	1.0	0.0	0.0	1.0	-0.7	0.0	3.0	-2.0	0.0
-2.0	3.7	0.0	-0.7	1.0	0.0	0.0	1.0	-0.5	0.0	3.7	-2.0
0.0	-2.0	1.9	0.0	-0.5	1.0	0.0	0.0	1.0	0.0	0.0	1.9

Voltage Values of the Original Circuit
and the Adjoint Circuit

Node number	Voltages of the original circuit from LU decomposition	Voltages of the original circuit direct from circuit	Voltages of the adjoint circuit from LU decomposition	Voltages of the adjoint circuit direct from circuit
1	0.5238	0.5238	-0.2857	-0.2857
2	0.2857	0.2857	-0.4286	-0.4286
3	0.1905	0.1905	-0.2857	-0.2857

Sensitivities of V w.r.t. Circuit Elements

dV/dG1	dV/dR2	dV/dG3	dV/dR4	dV/dG5
-0.1497	-0.1361	-0.1224	0.0544	-0.0544

From the report we can see that the multiplication of ***L*** and ***U*** matches ***G*** which verifies that ***L*** and ***U*** are the LU factors of ***G***. This is also verified by using the EXTRACT_L and EXTRACT_U functions to retrieve ***L*** and ***U*** from the ***GLU*** matrix; note however that the extracted matrices are the transposed versions of each other.

OSA☛ Question 3038 Consider the resistive network shown in Fig. Q3037, where G_1 = 1 S, G_2 = 1 S and G_3 = 1 S. Use OSA90/hope to calculate the node voltages of the original circuit by LU factorization of the nodal admittance matrix. Verify the LU matrices in OSA90 and evaluate

$$\frac{\partial i_3}{\partial G_1},\ \frac{\partial i_3}{\partial G_2} \text{ and } \frac{\partial i_3}{\partial G_3}$$

If the tolerances on G_1, G_2 and G_3 are ±2%, estimate the largest and smallest extremes of i_3. (See Question 3037.)

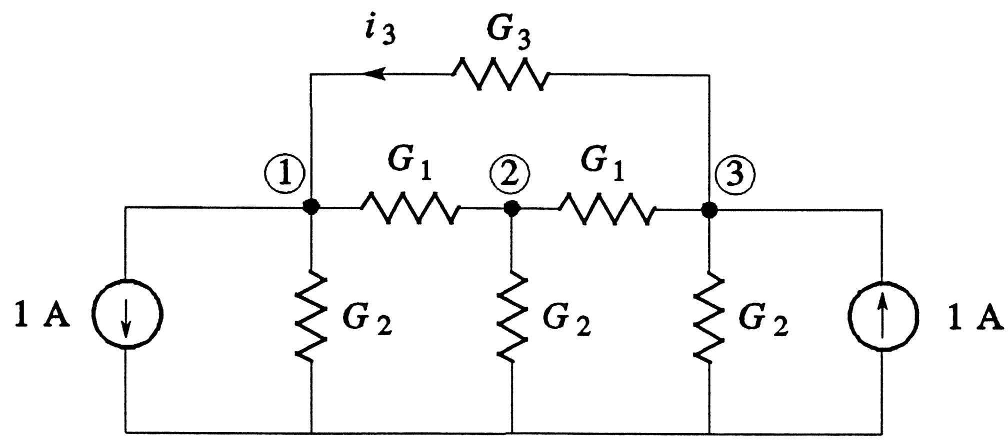

Fig. Q3037 A resistive network.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 3038

```
!  File name: Q3038
!  Circuit file for solving Q3038

!  Illustrate that the units are not microwave units

Control
   Non_Microwave_Units;
End

Expression
   ! Assign values to the circuit elements
   G1 = 1/OH;
   G2 = 1/OH;
   G3 = 1/OH;

   ! Set the conductance Matrix
   Y[3,3] = [ 3  -1  -1
             -1   3  -1
             -1  -1   3 ];

   ! Define the right hand side
   I[3] = [ -1  0  1 ];
```

```
   ! Calculate the node voltages by matrix inversion
   YINV[3,3] = INVERSE(Y);
   VINV[3] = PRODUCT(YINV , I);

   ! LU factorize without pivoting
   MatrixYLU[4,3] = LUF(Y);

   ! Extract Lower and upper matrix display
   L[3,3] = EXTRACT_L(MatrixYLU); ![ 3    0      0
                                  ! -1   (8/3)   0
                                  ! -1  -(4/3)   2!];

   U[3,3] =$EXTRACT_U(MatrixYLU); ![ 1  -(1/3)  -(1/3)
                                  !  0     1    -(1/2)
                                  !  0     0       1  ];

   ! Check if LxU = Y

   LxU[3,3] = PRODUCT(L,U);

   ! LU factorization of Y and solving for node voltage V
   YLU[4,3] = LU(Y);
   V[3] = SUBST(YLU,I);

   ! Solving for the adjoint circuit
   Ihat[3] = [ -1  0  1 ];        ! Norton equivalent current sources
   Vhat[3] = SUBST(YLU,Ihat);

   ! Calculate the sensitivities
   GradI1 = -1 * ( (V[1] - V[2]) * (Vhat[1] - Vhat[2]) + (V[2] - V[3]) * (Vhat[2] - Vhat[3]) );
   GradI2 = -1 * ( V[1] * Vhat[1] + V[2] * Vhat[2] + V[3] * Vhat[3] );
   GradI3 =  1 * ( (V[1] - V[3]) * (Vhat[1] - Vhat[3]) );

   GradI[3] = [ GradI1 GradI2 GradI3 ];

   ! Find largest and smallest extremes of I3 with +/- 2% tolerances in G1, G2, and G3

   DelI3max = -1 * (GradI1 * 0.02) + -1 * (GradI2 * 0.02) + (GradI3 * 0.02);
   DelI3min = (GradI1 * 0.02) + (GradI2 * 0.02) + (GradI3 * 0.02) * -1;

   I3 = (V[3] - V[1]) * G3;      ! Current I3 is equal to current through element G3
   I3max = I3 + DelI3max;       ! Maximum value of I3
   I3min = I3 + DelI3min;       ! Minimum value of I3

   j = 1;     ! Initialize the index of display
End

! Define the sweep parameter and output responses

Sweep
   j: from 1 to 3 step 1 LxU[j,1] LxU[j,2] LxU[j,3] Y[j,1] Y[j,2] Y[j,3];
   j: from 1 to 3 step 1 V[j] Vhat[j];
   GradI, DelI3max, DelI3min, I3, I3max, I3min;
End

! Define report block

Report
        Comparison of Matrix Y with LxU

   -----------------------------------------
           Matrix Y              LxU
   -----------------------------------------
${$% 1g$      $Y[j,1]$ $% 1g$  $Y[j,2]$ $% 1g$  $Y[j,3]$ $% 1g$      $LxU[j,1]$ $% 1g$  $LxU[j,2]$  $% 1g$
$LxU[j,3]$ }$
   -----------------------------------------
```

```
      Voltage Values of the Original Circuit
            and the Adjoint Circuit

  ----------------------------------------------------
   Node          Voltages of             Voltages of
  number    the original circuit    the adjoint circuit
  ----------------------------------------------------
${$%1g$      $j$            $%7.4f$ $V[j]$               $%7.4f$ $Vhat[j]$ }$
  ----------------------------------------------------


      Sensitivities of I3 w.r.t. Circuit Elements

  ----------------------------------------------------
       dI3/dG1          dI3/dG2          dI3/dG3
  ----------------------------------------------------
${$%5.4f$        $GradI1$ $%5.4f$        $GradI2$ $%5.4f$          $GradI3$ }$
  ----------------------------------------------------


      Largest and Smallest extremes of I3 if
       tolerances on  G1, G2 and G3 = +/-2%

  ------------------------------------------------
       I3         Maximum I3        Minimum I3
  ------------------------------------------------
${$%5.4f$       $I3$        $%5.4f$ $I3max$          $%5.4f$ $I3min$ }$
  ------------------------------------------------

End
```

The report follows.

Report for Question 3038

Comparison of Matrix Y with LxU

Matrix Y			LxU		
3	-1	-1	3	-1	-1
-1	3	-1	-1	3	-1
-1	-1	3	-1	-1	3

Voltage Values of the Original Circuit and the Adjoint Circuit

Node number	Voltages of the original circuit	Voltages of the adjoint circuit
1	-0.2500	-0.2500
2	-0.0000	-0.0000
3	0.2500	0.2500

Sensitivities of I3 w.r.t. Circuit Elements

dI3/dG1	dI3/dG2	dI3/dG3
-0.1250	-0.1250	0.2500

Largest and Smallest extremes of I3 if tolerances on G1, G2 and G3 = +/-2%

I3	Maximum I3	Minimum I3
0.5000	0.5100	0.4900

OSA☞ Question 3039 Compare the solutions obtained in Question 3038 by creating a circuit model in OSA90/hope.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 3039

```
! File name: Q3039.ckt
! Circuit file for solving Q3039

! Illustrate that the units are not microwave units

Control
   Non_Microwave_Units;
End

Expression
   ! Assign values to the circuit elements
   G1 = 1/OH;  R1 = 1 / G1;
   G2 = 1/OH;  R2 = 1 / G2;
   G3 = 1/OH;  R3 = 1 / G3;
   Iin1 = -1A;
   Iin2 = 1A;
   Iin3 = 1A;
End

Model

   SUBCIRCUIT RES_CKT 1 2 3 4 0 {
      RES 1 0 R = R2;
      RES 1 2 R = R1;
      RES 2 0 R = R2;
      RES 2 3 R = R1;
      RES 3 0 R = R2;
      RES 3 4 R = R3;
   };

   ! The original circuit
   RES_CKT 1 2 3 4 0;
   ISOURCE 1 0 NAME = Iinp1 IDC = Iin1;
   ISOURCE 3 0 NAME = Iinp2 IDC = Iin2;
   VLABEL 4 0 NAME = V1;
   VLABEL 2 0 NAME = V2;
   VLABEL 3 0 NAME = V3;
   ILABEL 1 4 NAME = I_3;

   ! The adjoint circuit
   RES_CKT 5 6 7 8 0;
   ISOURCE 7 5 NAME = Iinp3 IDC = Iin3;
   VLABEL 8 0 NAME = Vhat1;
   VLABEL 6 0 NAME = Vhat2;
   VLABEL 7 0 NAME = Vhat3;
   ILABEL 5 8 NAME = Ihat_3;

   CIRCUIT NAME = Q3039;

   ! Calculate the sensitivities
   GradI1 = -1 * ( (V1 - V2) * (Vhat1 - Vhat2) + (V2 - V3) * (Vhat2 - Vhat3) );
   GradI2 = -1 * ( (V1 * Vhat1) + (V2 * Vhat2) + (V3 * Vhat3) );
   GradI3 =  1 * ( (V1 - V3) * (Vhat1 - Vhat3) );

   GradI[3] = [ GradI1 GradI2 GradI3 ];
```

```
   ! Find largest and smallest extremes of I3 with +/- 2% tolerances in G1, G2, and G3

   DelI3max = -1 * (GradI1 * 0.02) + -1 * (GradI2 * 0.02) + (GradI3 * 0.02);
   DelI3min = (GradI1 * 0.02) + (GradI2 * 0.02) + (GradI3 * 0.02) * -1;

   I3 = -1 * I_3;  ! Current I3 is equal to current through element G3
   Ihat3 = -1 * Ihat_3;
   I3max = I3 + DelI3max;       ! Maximum value of I3
   I3min = I3 + DelI3min;       ! Minimum value of I3

   j = 1;    ! Initialize the index of display
End

! Define the sweep parameter and output responses

Sweep
   DC: Iin1 = -1A
       Iin2 = 1A
       V1 V2 V3 I3
       Vhat1 Vhat2 Vhat3  Ihat3
       GradI1, GradI2, GradI3, DelI3max, DelI3min, I3max, I3min;
End

! Define report block

Report
        Branch Voltages of the Original Circuit

   -------------------------------------------------
            V1              V2              V3
   -------------------------------------------------
${$%6.4f$        $V1$      $%6.4f$  $V2$       $%6.4f$  $V3$ }$
   -------------------------------------------------


        Branch Voltages of the Adjoint Circuit

   -------------------------------------------------
           Vhat1           Vhat2           Vhat3
   -------------------------------------------------
${$%6.4f$        $Vhat1$        $%6.4f$ $Vhat2$       $%6.4f$  $Vhat3$ }$
   -------------------------------------------------


        Sensitivities of I3 w.r.t. Circuit Elements

   -------------------------------------------------
          dI3/dG1         dI3/dG2         dI3/dG3
   -------------------------------------------------
${$%5.4f$        $GradI1$ $%5.4f$       $GradI2$ $%5.4f$        $GradI3$ }$
   -------------------------------------------------


        Largest and Smallest extremes of I3 if
         tolerances on  G1, G2 and G3 = +/-2%

   -------------------------------------------------
            I3            Max I3          Min I3
   -------------------------------------------------
${$%5.4f$         $I3$       $%5.4f$ $I3max$       $%5.4f$ $I3min$ }$
   -------------------------------------------------
End
```

The report follows.

Report for Question 3039

Branch Voltages of the Original Circuit

V1	V2	V3
-0.2500	-0.0000	0.2500

Branch Voltages of the Adjoint Circuit

Vhat1	Vhat2	Vhat3
-0.2500	0.0000	0.2500

Sensitivities of I3 w.r.t. Circuit Elements

dI3/dG1	dI3/dG2	dI3/dG3
-0.1250	-0.1250	0.2500

Largest and Smallest extremes of I3 if
tolerances on G1, G2 and G3 = +/-2%

I3	Max I3	Min I3
0.5000	0.5100	0.4900

OSA☛ Question 3046 Consider the linear circuit of Question 1039, which is assumed to be in the sinusoidal steady state. Let $V_1 = 1$ V, $\omega = 2$ rad/s, $R_1 = R_2 = R_3 = 2\ \Omega$, $C_1 = C_2 = C_3 = 1$ F. Use OSA90/hope to calculate $\partial V_3/\partial C_2$ and $\partial V_3/\partial R_1$ by the adjoint network method. Estimate ΔV_3 (the total change of V_3) when C_2 changes by +3% and R_1 by —10%. (See Question 3045.)

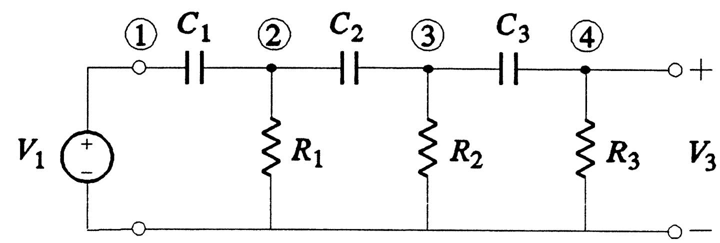

Fig. Q1039 CR ladder network.

Solution

For this problem we use direct circuit simulation to solve the original and adjoint circuits. The circuit file follows.

Circuit file for solving Question 3046

```
! File name: Q3046.ckt
! Circuit file for solving Question 3046

! Illustrate that the units are not the microwave units

Control
  Non_Microwave_Units;
End

Expression
   ! Assign values to circuit elements
   R1 = 2OH; G1 = 1 / R1;
   R2 = 2OH; G2 = 1 / R2;
   R3 = 2OH; G3 = 1 / R3;
   C1 = 1F;
   C2 = 1F;
   C3 = 1F;
   Omega = 2;
End

Model
   ! Define linear subcircuit to use for original and adjoint topology
   SUBCIRCUIT  Q3046_CKT  1 2 3 4 0
   {
        CAP 1 2 C = C1;
        RES 2 0 R = R1;
        CAP 2 3 C = C2;
        RES 3 0 R = R2;
        CAP 3 4 C = C3;
        RES 4 0 R = R3;
   };
```

```
   ! The original circuit
   Q3046_CKT  1 2 3 4 0;
   VSOURCE 1 0  NAME = VIN  V = 1V;
   VLABEL  2 3  NAME = VC2;
   VLABEL  2 0  NAME = VR1;
   OPEN    4;

   ! The adjoint circuit
   Q3046_CKT  0 5 6 7 0;
   ISOURCE 7 0  NAME = IhatIN  I = 1A;
   VLABEL  5 6  NAME = VhatC2;
   VLABEL  5 0  NAME = VhatR1;

   CIRCUIT;

   ! Convert the voltages from magnitude & phase to real & imaginary
   MP2RI(MVC2[1], PVC2[1], RVC2, IVC2);
   MP2RI(MVR1[1], PVR1[1], RVR1, IVR1);
   MP2RI(MVhatC2[1], PVhatC2[1], RVhatC2, IVhatC2);
   MP2RI(MVhatR1[1], PVhatR1[1], RVhatR1, IVhatR1);

   ! Calculate the Sensitivities:

   ! dV3/dC2 = -jwVV^
   RdV3dC2 = Omega * (RVC2 * IVhatC2 + IVC2 * RVhatC2);
   IdV3dC2 = Omega * (-1 * RVC2 * RVhatC2 + IVC2 * IVhatC2);

   ! dV3/dR1 = 1 / R^2 * VV^
   RdV3dR1 = 1 / R1 / R1 * (RVR1 * RVhatR1 - IVR1 * IVhatR1);
   IdV3dR1 = 1 / R1 / R1 * (IVR1 * RVhatR1 + RVR1 * IVhatR1);

   ! Calculate the change of V3
   RDeltaV3 = RdV3dC2 * 0.03 * C2 + RdV3dR1 * -0.10 * R1;
   IDeltaV3 = IdV3dC2 * 0.03 * C2 + IdV3dR1 * -0.10 * R1;
End

! Define the sweep parameter and output responses

Sweep
   HB: Omega: 2 FREQ=(1HZ*Omega/(2*PI)) RdV3dC2 IdV3dC2 RdV3dR1 IdV3dR1 RDeltaV3
       IDeltaV3;
End

! Define report block

Report
        Sensitivities of V3 w.r.t. C2 and R1

   ----------------------------------------------
          dV3/dC2                dV3/dR1
     -------------------    -------------------
      Real   Imaginary        Real   Imaginary
   ----------------------------------------------
$%5.4f$     $RdV3dC2$ $%5.4f$  $IdV3dC2$ $%5.4f$       $RdV3dR1$ $%5.4f$   $IdV3dR1$
   ----------------------------------------------


      Total Change of V3 w.r.t. Changes of C2 and R1

   ---------------------------------------------------
                                       Change of V3
                                   --------------------
     Change of C2  Change of R1    Real     Imaginary
   ---------------------------------------------------
        0.03          -0.02      $%5.4f$ $RDeltaV3$   $%5.4f$  $IdeltaV3$
   ---------------------------------------------------
End
```

The numerical results are shown in the report.

Report for Question Q3046

Sensitivities of V3 w.r.t. C2 and R1

dV3/dC2		dV3/dR1	
Real	Imaginary	Real	Imaginary
0.1834	0.0636	0.0561	0.0016

Total Change of V3 w.r.t. Changes of C2 and R1

		Change of V3	
Change of C2	Change of R1	Real	Imaginary
0.03	-0.02	-0.0057	0.0016

OSA☞ Question 3051 Consider the resistive network shown in Fig. Q3051, with the element values

$$V_{in} = 2V\ ,\quad G_1 = 0.5S\ ,\quad G_2 = 2.5S \text{ and } G_3 = 1.0S\ .$$

Use OSA90/hope to create the adjoint circuit and evaluate

$$\frac{\partial i_{in}}{\partial G_1},\ \frac{\partial i_{in}}{\partial G_2} \text{ and } \frac{\partial i_{in}}{\partial G_3}\ .$$

If the tolerance on G_1 is +1%, estimate the extreme of i_{in}.

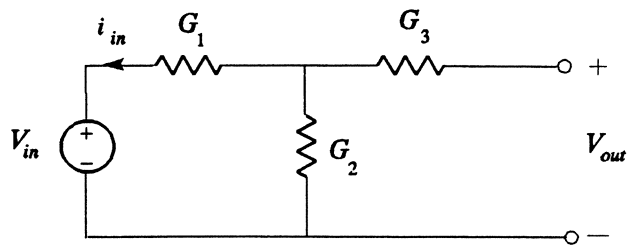

Fig. Q3051 A resistive network.

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 3051

```
!  File name: Q3051.ckt
!  Circuit file to solve Q3051

Model
! Assign values to circuit elements

   G1 = 0.5/OH;  R1 = 1 / G1;
   G2 = 2.5/OH;  R2 = 1 / G2;
   G3 = 1.0/OH;  R3 = 1 / G3;
   Vin = 2V;               ! Voltage source of original circuit
   Vin_hat = 1V;           ! Voltage source of adjoint circuit


! Define subcircuit
      SUBCIRCUIT RES_CKT 1 2 3 0 {
      RES      1 2 R = R1;
      RES      2 0 R = R2;
      RES      2 3 R = R3;
   }

! Define original circuit
   RES_CKT 1 2 3 0;
      VSOURCE 1 0 NAME = VINPUT VDC = Vin;
      VLABEL  1 2 NAME = V1;
      VLABEL  2 0 NAME = V2;
      VLABEL  2 3 NAME = V3;
      VLABEL  3 0 NAME = VOUT;
```

```
! Define Adjoint circuit
   RES_CKT 4 5 6 0;
   VSOURCE 4 0 NAME = VINPUT_hat VDC = VIN_hat;
   VLABEL  4 5 NAME = V1_hat;
   VLABEL  5 0 NAME = V2_hat;
   VLABEL  5 6 NAME = V3_hat;
   VLABEL  6 0 NAME = VOUT_hat;
   CIRCUIT NAME = Q3051;

! Calculate Sensitivities
   dG1 = -V1 * V1_hat;    ! Find dIn/dG1
   dG2 = -V2 * V2_hat;    ! Find dIn/dG2
   dG3 = -V3 * V3_hat;    ! Find dIn/dG3

! Calculate the change in Iin if G1 changes by 1%
   delIin = dG1*(1/100);

End

Sweep
   DC: Vin = 2V
   V1 V2 V3 VOUT
   V1_hat V2_hat V3_hat VOUT_hat Vin_hat
   dG1 dG2 dG3
   delIin;
End

Report
          Branch Voltages of original circuit

   ---------------------------------------------------
        Vin        V1        V2        V3       Vout
   ---------------------------------------------------
${$%2.1f$        $Vin$     $%5.4f$ $V1$  $%5.4f$ $V2$  $%5.4f$ $V3$  $%5.4f$ $VOUT$ }$
   ---------------------------------------------------


          Branch Voltages of adjoint circuit

   ---------------------------------------------------
        Vin        V1        V2        V3       Vout
   ---------------------------------------------------
${$%2.1f$        $Vin_hat$     $%5.4f$ $V1_hat$  $%5.4f$ $V2_hat$  $%5.4f$ $V3_hat$  $%5.4f$ $VOUT_hat$ }$
   ---------------------------------------------------


                   Sensitivities

   -----------------------------------------------
        dIn/dG1          dIn/dG2         dIn/dG3
   -----------------------------------------------
${$%5.4f$        $dG1$ $%5.4f$    $dG2$ $%5.4f$    $dG3$ }$
   -----------------------------------------------


  ----------------------------------------------------
   Tolerance in Iin with 1% change in G1 = ${ $%5.4f$ $delIin$ }$
  ----------------------------------------------------

End
```

The results are shown in the following report.

Report for Question 3051

Branch Voltages

Vin	V1	V2	V3	Vout
2.0	1.6667	0.3333	0.0000	0.3333

Sensitivities

dIn/dG1	dIn/dG2	dIn/dG3
-1.3889	-0.0556	-0.0000

Change in Iin with 1% change in G1 = -0.0139

4.2.4 Nonlinear Networks

4.2.5 One-Dimensional Search Methods

OSA☛ Question 5005 (p. 4-78)

OSA☞ Question 5005 Starting with the interval [0, 6], use OSA90/hope and apply 20 iterations of the Golden Section search method to the minimization w.r.t. ϕ of the function described in Question 5004. (See Question 5004.)

$$U = -\phi + 5 \qquad \phi \le 1$$
$$U = 0.5\,(\phi - 3)^2 + 1 \qquad 1<\phi \le 4$$
$$U = 3 - (\phi - 6)^2/3 \qquad \phi > 4$$

Solution

The circuit file for solving this problem follows.

Circuit file for solving Question 5006

```
! File name: Q5006.ckt
! Circuit file for solving Q147

Expression
! Initialise parameters
   phi: 0;
   I: 0;
   phi_a: 0;
   phi_b: 0;
   phi_l: 0;
   phi_u: 0;

   tau : ((1+SQRT(5))/2);
   phi_ll : 0;
   phi_uu : 6;
   II:6;
   phi_aa : 2.2918924;
   phi_bb : 3.7082818;

! Define the discontinuous function as a vector
   U_a[3] = [ (-phi_a + 5)  (0.5 * (phi_a - 3)^2 + 1)  (3 - (phi_a - 6)^2 / 3) ];

! Calculate the value of Ua
   Ua:    if (phi_a < 1)  (U_a[1])
          else (
                if ((phi_a > 1)*(phi_a < 4)) (U_a[2]) else (U_a[3])
               );
! Define the discontinuous function as a vector
   U_b[3] = [ (-phi_b + 5)  (0.5 * (phi_b - 3)^2 + 1)  (3 - (phi_b - 6)^2 / 3) ];

! Calculate the value of Ua
   Ub:    if (phi_b < 1) (U_b[1])
          else (
                if ((phi_b > 1)*(phi_b < 4)) (U_b[2]) else (U_b[3])
                );

! Calculate the new upper and lower values of phi
   new_phi_ll: If ((Ua > Ub) + (Ua = Ub)) (phi_a) else (phi_l);
   new_phi_uu: If ((Ua > Ub) + (Ua = Ub)) (phi_u) else (phi_b);
   new_II:(new_phi_uu - new_phi_ll);
   new_phi_aa: If ((Ua > Ub) + (Ua = Ub)) (phi_b) else (new_II / tau^2 + new_phi_ll);
   new_phi_bb: If ((Ua > Ub) + (Ua = Ub)) (new_II / tau + new_phi_ll) else (phi_a);

!  Calculate the value of tau by dividing previous interval with new interval
   cal_tau = I/new_II;

   K: 1;

End
```

```
Sweep
   K: from 1 to 20 step=1
   I=II
   phi_a=phi_aa        phi_b=phi_bb
   phi_l=phi_ll        phi_u=phi_uu

   phi_ll=new_phi_ll
   phi_uu=new_phi_uu
   II=new_II
   phi_aa=new_phi_aa
   phi_bb=new_phi_bb

   phi_a phi_b
   phi_u phi_l
   Ua    Ub
   I cal_tau;
End

Report
                          Golden Search iteration outputs

------------------------------------------------------------------------------------------------
Iteration     phi_a       phi_b      phi_l      phi_u        Ua         Ub         I     calculated tau
------------------------------------------------------------------------------------------------
${$%2g$  $K$       $%5.4f$ $phi_a$    $%5.4f$ $phi_b$    $%5.4f$ $phi_l$    $%5.4f$ $phi_u$    $%5.4f$ $Ua$
  $%5.4f$ $Ub$    $%5.4f$ $I$    $%5.4f$ $cal_tau$ }$
------------------------------------------------------------------------------------------------

End
```

In the report that follows, one can see this search converging upon the minimum of 1.0000 on the interval from 3.0000 to 3.0006 after only 12 iterations. As well, the ratio between consecutive iteration intervals is seen to be only approximately 1.618 because of computational approximations.

Report for Question 5006

Golden Search iteration outputs

Iteration	phi_a	phi_b	phi_l	phi_u	Ua	Ub	I	calculated tau
1	2.2919	3.7083	0.0000	6.0000	1.2507	1.2508	6.0000	1.6180
2	1.4164	2.2919	0.0000	3.7083	2.2538	1.2507	3.7083	1.6180
3	2.2919	2.8329	1.4164	3.7083	1.2507	1.0140	2.2918	1.6181
4	2.8329	3.1673	2.2919	3.7083	1.0140	1.0140	1.4164	1.6180
5	2.6263	2.8329	2.2919	3.1673	1.0698	1.0140	0.8754	1.6180
6	2.8329	2.9606	2.6263	3.1673	1.0140	1.0008	0.5410	1.6179
7	2.9606	3.0395	2.8329	3.1673	1.0008	1.0008	0.3344	1.6180
8	2.9118	2.9606	2.8329	3.0395	1.0039	1.0008	0.2067	1.6180
9	2.9606	2.9908	2.9118	3.0395	1.0008	1.0000	0.1277	1.6184
10	2.9908	3.0094	2.9606	3.0395	1.0000	1.0000	0.0789	1.6180
11	2.9793	2.9908	2.9606	3.0094	1.0002	1.0000	0.0488	1.6180
12	2.9908	2.9979	2.9793	3.0094	1.0000	1.0000	0.0301	1.6171
13	2.9979	3.0023	2.9908	3.0094	1.0000	1.0000	0.0186	1.6180
14	2.9952	2.9979	2.9908	3.0023	1.0000	1.0000	0.0115	1.6181
15	2.9979	2.9996	2.9952	3.0023	1.0000	1.0000	0.0071	1.6206
16	2.9996	3.0006	2.9979	3.0023	1.0000	1.0000	0.0044	1.6154
17	3.0006	3.0012	2.9996	3.0023	1.0000	1.0000	0.0027	1.6179
18	3.0002	3.0006	2.9996	3.0012	1.0000	1.0000	0.0017	1.6139
19	3.0000	3.0002	2.9996	3.0006	1.0000	1.0000	0.0010	1.6181
20	3.0002	3.0004	3.0000	3.0006	1.0000	1.0000	0.0006	1.6177

4.2.6 Tolerances and Worst-Case Analysis

4.2.7 State Equations

4.2.8 Applications

OSA☛ Question 8002 Use the ℓ_1, ℓ_2 and minimax optimizers of OSA90/hope to approximate in a uniformly weighted sense

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

by

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

on the interval [0, 2]. (See Question 8001.)

Solution

In this question, a_1 and a_2 are optimizable variables. We choose a starting point $a_1 = 1$ and $a_2 = 1$ which is indicated in the circuit file before optimization. The solutions for a_1 and a_2 are given in the circuit files after optimization. The optimizable variables before and after optimization are listed in the reports.

We let

$$y_1 = x^2$$

$$y_0 = a_1 x + a_2 \exp(x)$$

$$y_2 = y_0 - y_1$$

The circuit file before optimization follows.

Circuit file before optimization

```
! File name: Q8002_0.ckt
! Circuit file for solving Question 8002
! Before optimization

Expression
   ! Define the optimizable variables
   a1 = ?-50  1  50?;     ! Initial value is 1, lower bound -50, upper bound 50
   a2 = ?-50  1  50?;     ! Initial value is 1, lower bound -50, upper bound 50

   x  = 1;                ! Initialize x

   ! Define functions
   y0  = a1 * x + a2 * exp(x);
   y1 = x * x;

   ! Calculate error function
   y2 = y0 - y1;
End

! Define sweep parameter and output responses
```

```
Sweep
   x: from 0 to 2 step 0.01 y0 y1 y2
   {Xsweep  Title = "The curves of y0, y1 and y2"
    X_title = "x"  X = x
    Y_title = "Functions y0, y1 and y2"  Y = y0.green & y1.yellow & y2.red};
End

! Define specifications for optimization

Specification
   x: from 0 to 2 step 0.01 y2 = 0;
End

! Define report block

Report
      Optimizable Variables Before and After Optimization

   --------------------------------------------------------
     Variable    Before optimization    After optimization
   --------------------------------------------------------
        a1               1                 $% 5.4f$ $a1$
        a2               1                 $% 5.4f$ $a2$
   --------------------------------------------------------
End
```

The curves of y_0, y_1 and y_2 before optimization are shown in Fig. SQ8002.1. From Fig. SQ8002.1 we can see that the error y_2 is very large.

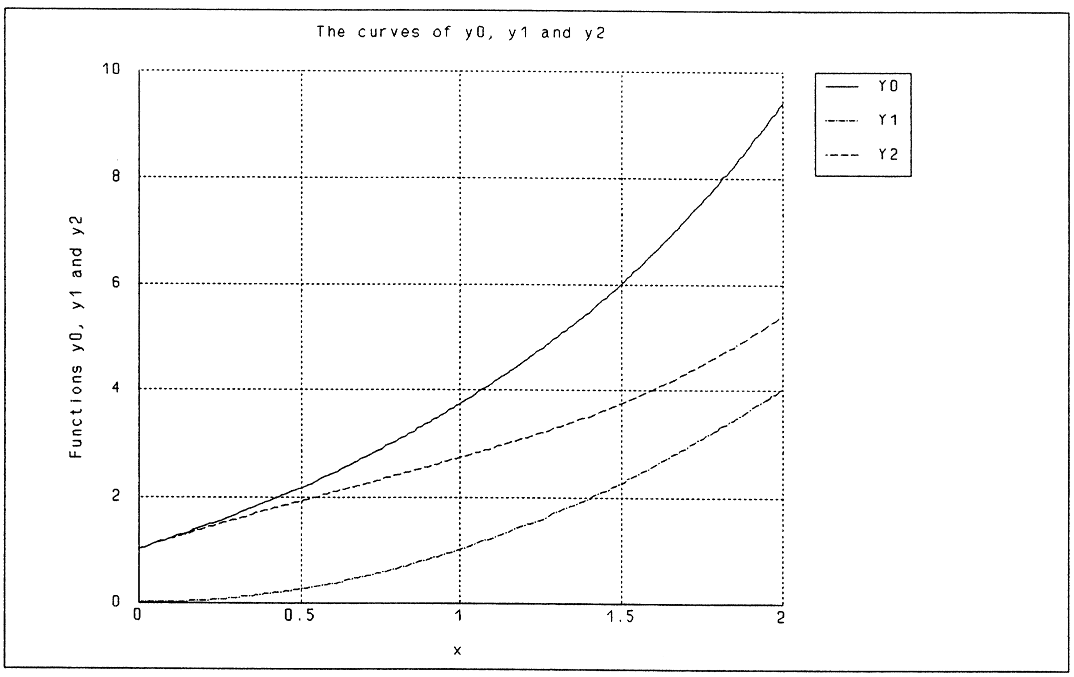

Fig. SQ8002.1 The curves of y_0, y_1 and y_2 before optimization.

First, we use the ℓ_1 optimizer of OSA90/hope to carry out the optimization. Every iteration of optimization is listed as follows (as recorded in the osa90_0.log file).

Iterations of ℓ_1 optimization

```
10:45:08 Read in File Q8002_0.ckt
10:45:11 Parsing Input File...
10:45:11 File Parsing Completed
10:45:13 L1 Optimization
10:45:13 Optimization...  Press any key to interrupt
10:45:15 Iteration   1/30 L1 Objective=575.435
10:45:17 Iteration   2/30 L1 Objective=566.995
10:45:19 Iteration   3/30 L1 Objective=550.112
10:45:21 Iteration   4/30 L1 Objective=516.348
10:45:24 Iteration   5/30 L1 Objective=448.82
10:45:26 Iteration   6/30 L1 Objective=313.763
10:45:28 Iteration   7/30 L1 Objective=72.7544
10:45:30 Iteration   8/30 L1 Objective=69.072
10:45:32 Iteration   9/30 L1 Objective=65.8982
10:45:34 Iteration  10/30 L1 Objective=65.8982
10:45:34 Solution  L1 Objective=65.8982
10:45:34 Optimization Completed
10:45:34 Elapsed Real Time: 00:00:21   CPU Time 00:00:21
10:46:07 Save File Q8002_l1.ckt
```

From the above, we can observe that it takes 10 iterations, 21 seconds CPU time for the optimization to converge. The ℓ_1 error is reduced from 575.435 to 65.8982. The solution is given in the circuit file after ℓ_1 optimization. The optimizable variables before and after optimization are listed in the report. The curves of y_0, y_1 and y_2 after ℓ_1 optimization are plotted in Fig. SQ8002.2.

Circuit file after ℓ_1 optimization

```
!  File name: Q8002_l1.ckt
!  Circuit file for solving Question 8002
!  Before optimization

Expression
   ! Define the optimizable variables
   a1 = ?-50  -0.964438  50?;    ! Initial value is 1, lower bound -50, upper bound 50
   a2 = ?-50  0.808566  50?;    ! Initial value is 1, lower bound -50, upper bound 50

   x  = 1;              ! Initialize x

      .            ! The circuit file remains unchanged from here on
      .
      .
```

Report after ℓ_1 optimization

Optimizable Variables Before and After Optimization

Variable	Before optimization	After optimization
a1	1	-0.9644
a2	1	0.8086

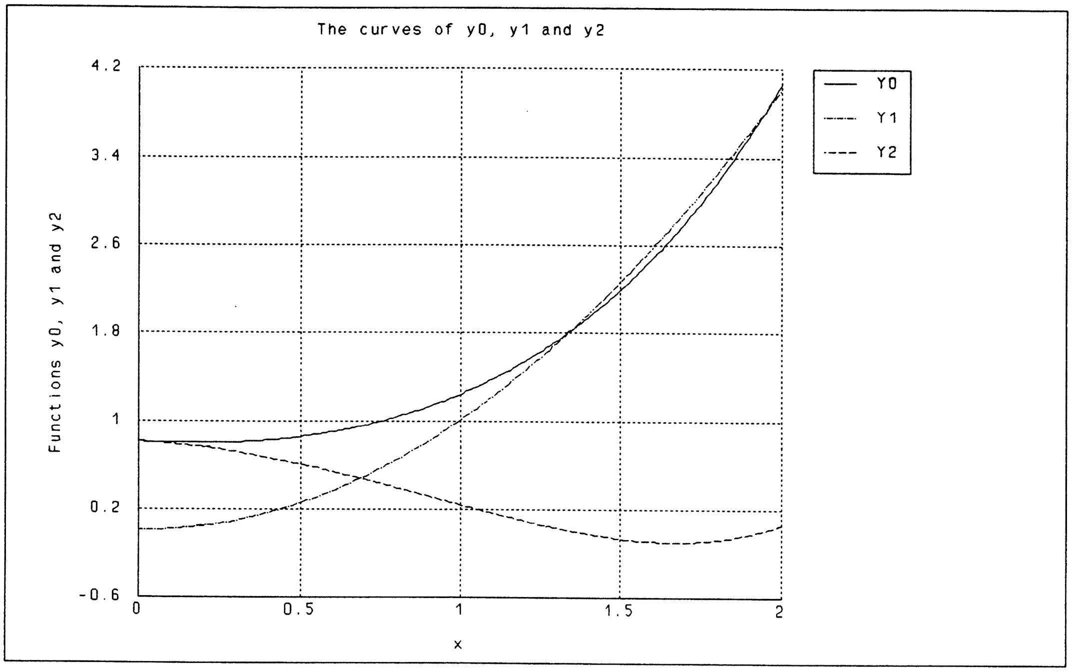

Fig. SQ8002.2 The curves of y_0, y_1 and y_2 after ℓ_1 optimization.

Secondly, we use the ℓ_2 optimizer of OSA90/hope to carry out the optimization. Every iteration of optimization is listed as follows (as recorded in the osa90_0.log file).

Iterations of ℓ_2 optimization

```
10:47:18 Read in File q166a_0..ckt
10:47:21 Parsing Input File...
10:47:21 File Parsing Completed
10:47:25 L2 Optimization
10:47:25 Optimization...  Press any key to interrupt
10:47:27 Iteration   1/30 L2 Objective=1923.82
10:47:29 Iteration   2/30 L2 Objective=32.5019
10:47:31 Iteration   3/30 L2 Objective=31.595
10:47:33 Iteration   4/30 L2 Objective=31.1028
10:47:35 Iteration   5/30 L2 Objective=30.9667
10:47:37 Iteration   6/30 L2 Objective=30.9517
10:47:39 Iteration   7/30 L2 Objective=30.9512
10:47:41 Iteration   8/30 L2 Objective=30.9512
10:47:43 Iteration   9/30 L2 Objective=30.9512
10:47:43 Solution  L2 Objective=30.9512
10:47:43 Optimization Completed
10:47:43 Elapsed Real Time: 00:00:19   CPU Time 00:00:18
10:48:27 Save File Q8002_12.ckt
```

From the above, we can observe that it takes 9 iterations, 18 seconds CPU time for the optimization to converge. The ℓ_2 error is reduced from 1923.82 to 30.9512. The solution is given in the circuit file after ℓ_2 optimization. The optimizable variables are before and after optimization are listed in the report. The curves of y_0, y_1 and y_2 after ℓ_2 optimization are plotted in Fig. SQ8002.3

Circuit file after ℓ_2 optimization

```
!  File name: Q8002_l2.ckt
!  Circuit file for solving Question 8002
!  Before optimization

Expression
   ! Define the optimizable variables
   a1 = ?-50  5.73537e-05  50?;    ! Initial value is 1, lower bound -50, upper bound 50
   a2 = ?-50  0.477359  50?;    ! Initial value is 1, lower bound -50, upper bound 50

   x  = 1;              ! Initialize x

      .             ! The circuit file remains unchanged from here on
      .
      .
```

Report after ℓ_2 optimization

```
      Optimizable Variables Before and After Optimization

   --------------------------------------------------------
     Variable    Before optimization    After optimization
   --------------------------------------------------------
        a1               1                   0.0001
        a2               1                   0.4774
   --------------------------------------------------------
```

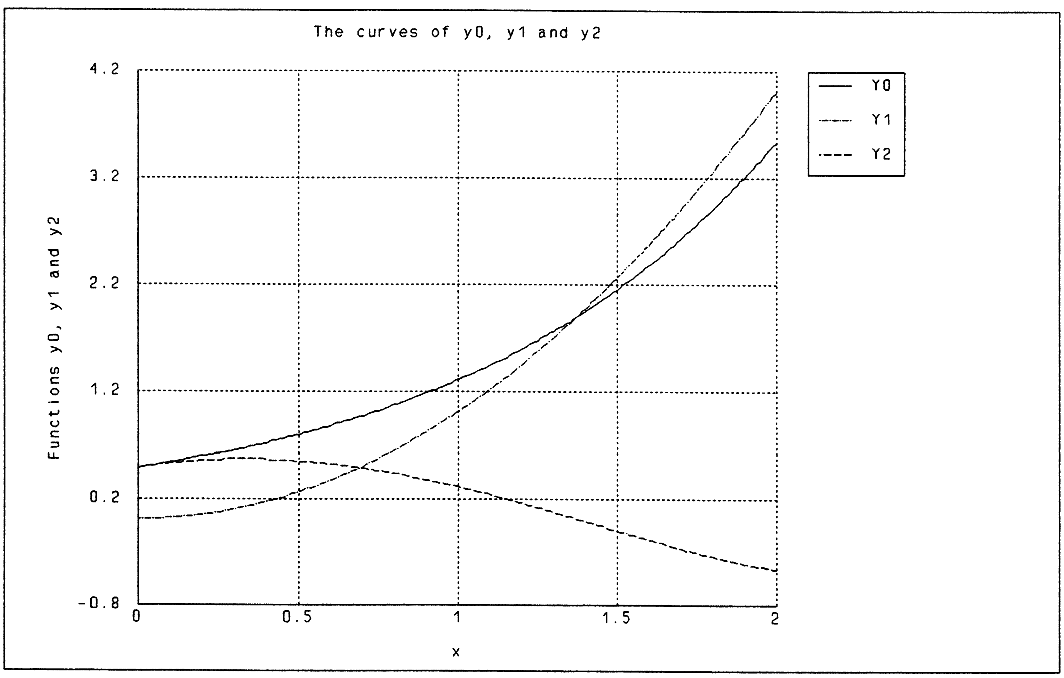

Fig. SQ8002.3 The curves of y_0, y_1 and y_2 after ℓ_2 optimization.

Finally, we use the minimax optimizer of OSA90/hope to carry out the optimization. Every iteration of optimization is listed as follows (as recorded in the osa90_0.log file).

Iterations of minimax optimization

```
10:50:41 Read in File Q8002.ckt
10:50:42 Parsing Input File...
10:50:42 File Parsing Completed
10:50:47 Minimax Optimization
10:50:47 Optimization...  Press any key to interrupt
10:50:49 Iteration   1/30 Max Error=5.38906
10:50:51 Iteration   2/30 Max Error=5.31251
10:50:53 Iteration   3/30 Max Error=5.15941
10:50:55 Iteration   4/30 Max Error=4.85321
10:50:57 Iteration   5/30 Max Error=4.24082
10:50:59 Iteration   6/30 Max Error=3.01602
10:51:02 Iteration   7/30 Max Error=0.904477
10:51:06 Iteration   8/30 Max Error=4.33222
10:51:09 Iteration   9/30 Max Error=0.654596
10:51:12 Iteration  10/30 Max Error=0.556399
10:51:14 Iteration  11/30 Max Error=0.540062
10:51:16 Iteration  12/30 Max Error=0.538233
10:51:18 Iteration  13/30 Max Error=0.538232
10:51:18 Solution  Max Error=0.538232
10:51:18 Optimization Completed
10:51:18 Elapsed Real Time: 00:00:31   CPU Time 00:00:31
10:51:30 Save File Q8002_mm.ckt
```

From the above, we can observe that it takes 13 iterations, 31 seconds CPU time for the optimization to converge. The minimax error is reduced from 5.38906 to 0.538232. The solution is given in the circuit file after minimax optimization. The optimizable variables before and after optimization are list in the report. The curves of y_0, y_1 and y_2 after minimax optimization are plotted in Fig. SQ8002.4

Circuit file after minimax optimization

```
!  File name: Q8002_mm.ckt
!  Circuit file for solving Question 8002
!  After minimax optimization

Expression
   ! Define the optimizable variables
   a1 = ?-50  0.181042  50?;    ! Initial value is 1, lower bound -50, upper bound 50
   a2 = ?-50  0.419496  50?;    ! Initial value is 1, lower bound -50, upper bound 50

   x  = 1;              ! Initialize x

      .             ! The circuit file remains unchanged from here on.
      .
      .
```

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before optimization	After optimization
a1	1	0.1810
a2	1	0.4195

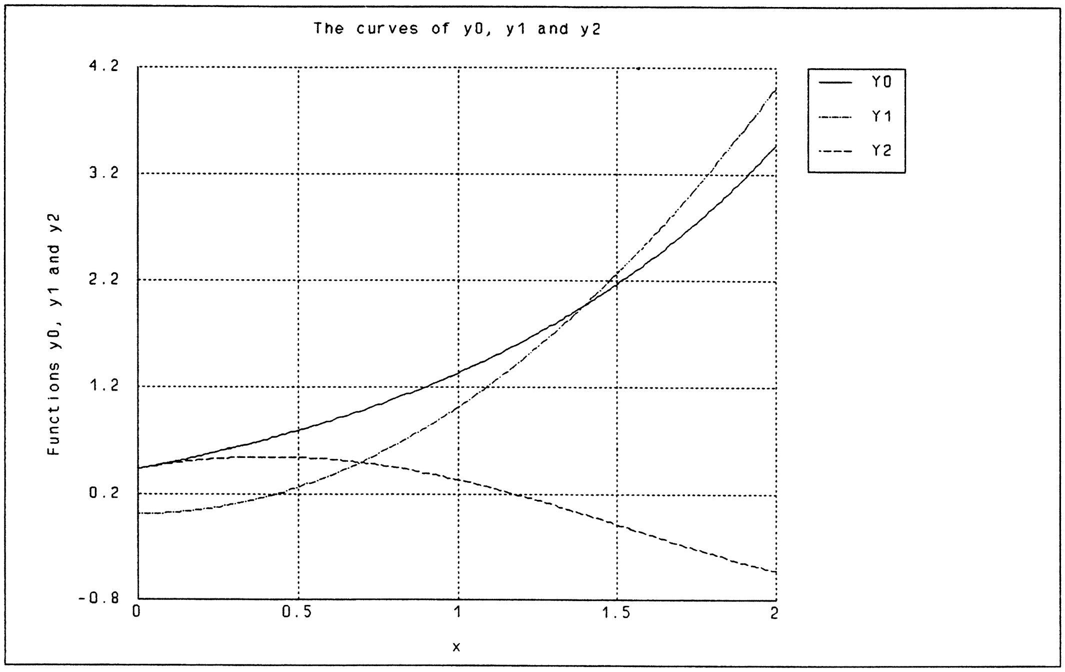

Fig. SQ8002.4 The curves of y_0, y_1 and y_2 after minimax optimization.

OSA☞ Question 8004 Use the ℓ_1, ℓ_2 and minimax optimizers of OSA90/hope to approximate in a uniformly weighted 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]. (See Question 8003.)

Solution

In this question, a_0, a_1, a_2, b_1 and b_2 are optimizable variables. We choose the starting point $a_0 = 0$, $a_1 = -10$, $a_2 = 20$, $b_1 = -1$ and $b_2 = 20$ which is indicated in the circuit file before optimization and in the reports. The solutions for a_0, a_1, a_2, b_1 and b_2 are given in the circuit files after optimization. The optimizable variables before and after optimization are listed in the reports.

We let

$$y_1 = \frac{[(8x - 1)^2 + 1]^{0.5} \tan^{-1}(8x)}{8x}$$

$$y_0 = \frac{a_0 + a_1 x + a_2 x^2}{1 + b_1 x + b_2 x^2}$$

$$y_2 = y_0 - y_1$$

The circuit file before optimization is below.

Circuit file before optimization

```
!  File name: Q8004_0.ckt
!  Circuit file for solving Question 8004
!  Before optimization

Expression
   ! Define optimizable variables
   a0 = ? -1     0      5?;      ! Initial value is 0, lower bound -1, upper bound 5
   a1 = ?-20    -10    -1?;      ! Initial value is -10, lower bound -20, upper bound -1
   a2 = ? 10     20     100?;    ! Initial value is 20, lower bound 10, upper bound 100
   b1 = ?-10    -1      10?;     ! Initial value is -1, lower bound -10, upper bound 10
   b2 = ? 10     20     50?;     ! Initial value is 20, lower bound 10, upper bound 50

   x  = 1;                       ! Initialize x

   ! Define functions
   y0 = (a0 + a1 * x + a2 * x * x) / (1 + b1 * x + b2 * x * x);
   y1 = sqrt((8 * x - 1) * (8 * x - 1) + 1) * atan(8 * x) / (8 * x);

   ! Calculate error function
   y2 = y0 - y1;
End
```

```
! Define sweep parameter and output responses

Sweep
   x:  from -1 to -0.01 step 0.01  from 0.01 to 1 step 0.01 y0 y1 y2
   {Xsweep  Title = "The curves of y0, y1 and y2"
    X_title = "x"  X = x
    Y_title = "Functions y0, y1 and y2"  Y = y0.green & y1.yellow & y2.red
    Ymin = -2  Ymax = 2};
End

! Define specifications for optimization

Specification
  x:  from -1 to -0.01 step 0.01  from 0.01 to 1 step 0.01 y2 = 0;
End

! Define report block

Report
      Optimizable Variables Before and After Optimization

   -------------------------------------------------------
     Variable    Before optimization    After optimization
   -------------------------------------------------------
        a0              0                  $% 5.4f$ $a0$
        a1            -10                  $% 5.4f$ $a1$
        a2             20                  $% 5.4f$ $a2$
        b1             -1                  $% 5.4f$ $b1$
        b2             20                  $% 5.4f$ $b2$
   -------------------------------------------------------
End
```

The curves of y_0, y_1 and y_2 before optimization are shown in Fig. SQ8004.1. From Fig. SQ8004.1 we can see that the error y_2 is very large.

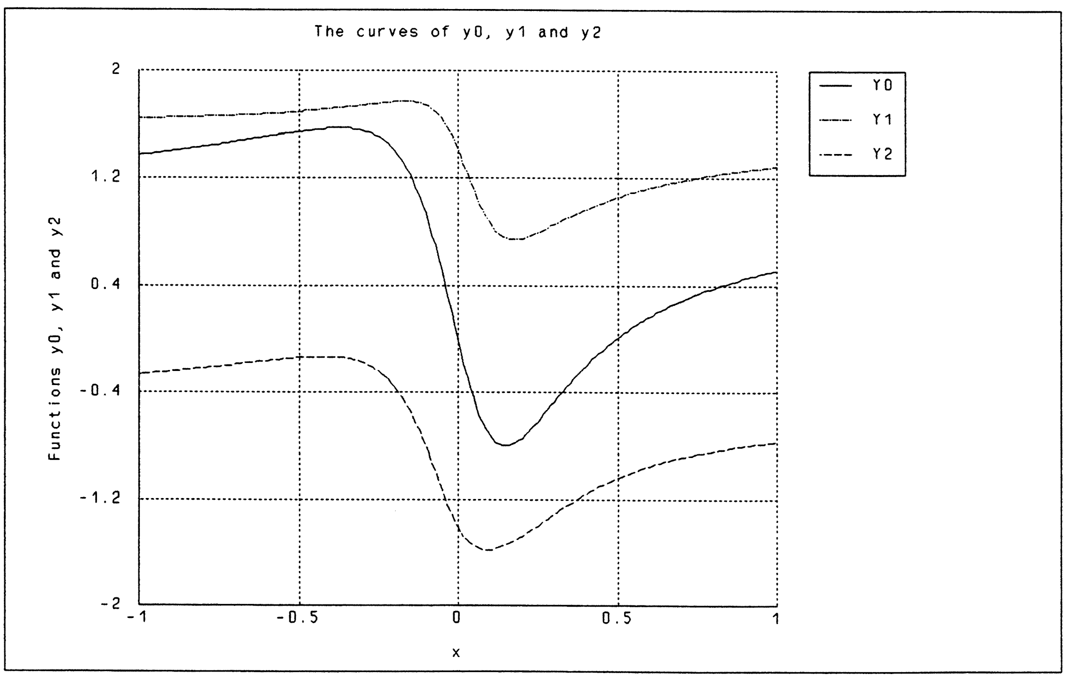

Fig. SQ8004.1 The curves of y_0, y_1 and y_2 before optimization.

First, we use the ℓ_1 optimizer of OSA90/hope to carry out the optimization. Every iteration of optimization is listed as follows (as recorded in the osa90_0.log file).

Iterations of ℓ_1 optimization

```
13:35:19 Read in File Q8004_0.ckt
13:35:23 L1 Optimization
13:35:23 Optimization...  Press any key to interrupt
13:35:27 Iteration   1/30 L1 Objective=144.893
13:35:31 Iteration   2/30 L1 Objective=144.047
13:35:34 Iteration   3/30 L1 Objective=142.354
13:35:38 Iteration   4/30 L1 Objective=138.966
13:35:42 Iteration   5/30 L1 Objective=132.182
13:35:46 Iteration   6/30 L1 Objective=118.566
13:35:50 Iteration   7/30 L1 Objective=97.0407
13:35:53 Iteration   8/30 L1 Objective=63.1862
13:35:58 Iteration   9/30 L1 Objective=24.975
13:36:03 Iteration  10/30 L1 Objective=7.07735
13:36:07 Iteration  11/30 L1 Objective=3.4247
13:36:12 Iteration  12/30 L1 Objective=2.46871
13:36:16 Iteration  13/30 L1 Objective=2.38408
13:36:20 Iteration  14/30 L1 Objective=2.38125
13:36:24 Iteration  15/30 L1 Objective=2.38099
13:36:28 Iteration  16/30 L1 Objective=2.38099
13:36:28 Solution  L1 Objective=2.38099
13:36:28 Optimization Completed
13:36:28 Elapsed Real Time: 00:01:05   CPU Time 00:01:04
13:37:47 Save File Q8004_l1.ckt
```

From the above, we can observe that it takes 16 iterations, 1 minute and 4 seconds CPU time for the optimization to converge. The ℓ_1 error is reduced from 144.893 to 2.38099. The solution is given in the circuit file after ℓ_1 optimization. The optimizable variables before and after optimization are listed in the report. The curves of y_0, y_1 and y_2 after ℓ_1 optimization are plotted in Fig. SQ8004.2.

Circuit file after ℓ_1 optimization

```
!  File name: Q8004_l1.ckt
!  Circuit file for solving Question 8004
!  After l1 optimization

Expression
   ! Define optimizable variables
   a0 = ? -1    1.39873     5?;     ! Initial value is 0, lower bound -1, upper bound 5
   a1 = ?-20   -9.92644    -1?;     ! Initial value is -10, lower bound -20, upper bound -1
   a2 = ? 10    39.2705   100?;     ! Initial value is 20, lower bound 10, upper bound 100
   b1 = ?-10   -3.59109    10?;     ! Initial value is -1, lower bound -10, upper bound 10
   b2 = ? 10    26.8565    50?;     ! Initial value is 20, lower bound 10, upper bound 50

   x  = 1;                        ! Initialize x
        .               ! The circuit file remains unchanged from here on.
        .
        .
```

Report after ℓ_1 optimization

Optimizable Variables Before and After Optimization

Variable	Before optimization	After optimization
a0	0	1.3987
a1	-10	-9.9264
a2	20	39.2705
b1	-1	-3.5911
b2	20	26.8565

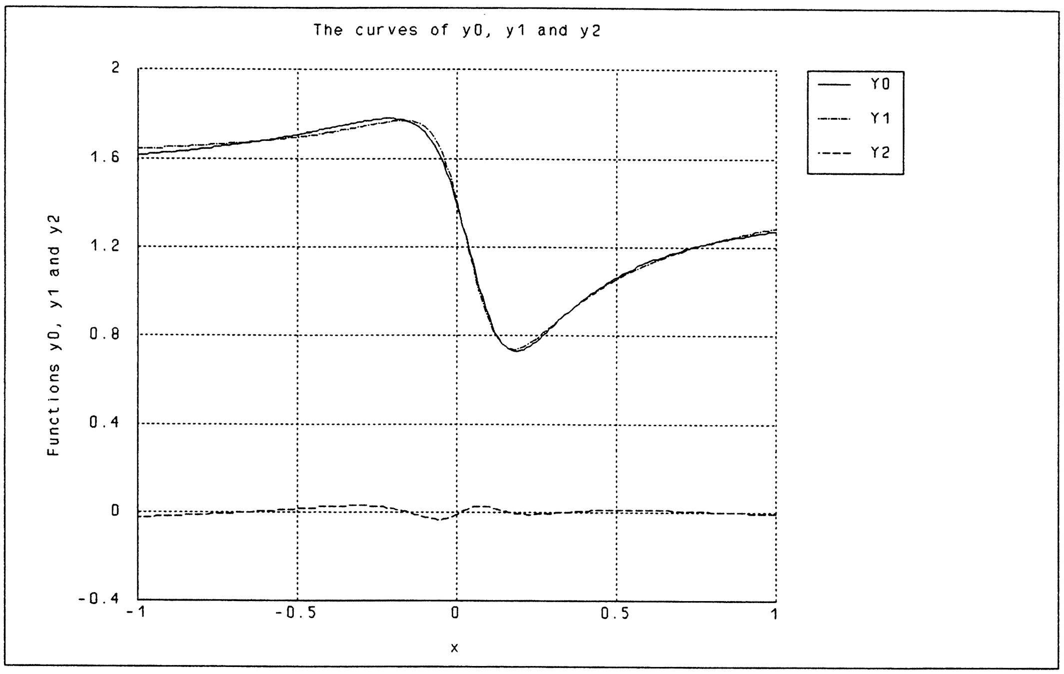

Fig. SQ8004.2 The curves of y_0, y_1 and y_2 after ℓ_1 optimization.

Secondly, we use the ℓ_2 optimizer of OSA90/hope to carry out the optimization. Every iteration of optimization is listed as follows (as recorded in the osa90_0.log file).

Iterations of ℓ_2 optimization

```
13:37:57 Read in File Q8004_0.ckt
13:38:04 Parsing Input File...
13:38:04 File Parsing Completed
13:38:32 L2 Optimization
13:38:32 Optimization...  Press any key to interrupt
13:38:36 Iteration   1/30 L2 Objective=152.698
13:38:39 Iteration   2/30 L2 Objective=51.125
13:38:43 Iteration   3/30 L2 Objective=32.2323
13:38:47 Iteration   4/30 L2 Objective=20.3064
13:38:50 Iteration   5/30 L2 Objective=10.2264
13:38:54 Iteration   6/30 L2 Objective=3.38132
13:38:58 Iteration   7/30 L2 Objective=0.828776
13:39:01 Iteration   8/30 L2 Objective=0.493676
13:39:05 Iteration   9/30 L2 Objective=0.445679
13:39:09 Iteration  10/30 L2 Objective=0.398259
13:39:12 Iteration  11/30 L2 Objective=0.350245
13:39:16 Iteration  12/30 L2 Objective=0.29565
13:39:20 Iteration  13/30 L2 Objective=0.225461
13:39:23 Iteration  14/30 L2 Objective=0.149896
13:39:27 Iteration  15/30 L2 Objective=0.0895716
13:39:30 Iteration  16/30 L2 Objective=0.0561248
13:39:34 Iteration  17/30 L2 Objective=0.0441542
13:39:38 Iteration  18/30 L2 Objective=0.0418578
13:39:41 Iteration  19/30 L2 Objective=0.0416182
13:39:45 Iteration  20/30 L2 Objective=0.0416077
13:39:48 Iteration  21/30 L2 Objective=0.0416074
13:39:51 Iteration  22/30 L2 Objective=0.0416075
13:39:51 Solution  L2 Objective=0.0416074
13:39:51 Optimization Completed
13:39:51 Elapsed Real Time: 00:01:21   CPU Time 00:01:19
13:39:59 Save File Q8004_l2.ckt
```

From the above, we can observe that it takes 22 iterations, 1 minute and 19 seconds CPU time for the optimization to converge. The ℓ_2 error is reduced from 152.698 to 0.0416074. The solution is given in the circuit file after ℓ_2 optimization. The optimizable variables before and after optimization are listed in the report. The curves of y_0, y_1 and y_2 after ℓ_2 optimization are plotted in Fig. SQ8004.3.

Circuit file after ℓ_2 optimization

```
!  File name: Q8004_l2.ckt
!  Circuit file for solving Question 8004
!  After l2 optimization

Expression
   ! Define optimizable variables
   a0 = ? -1    1.40681     5?;    ! Initial value is 0, lower bound -1, upper bound 5
   a1 = ?-20   -10.2085    -1?;    ! Initial value is -10, lower bound -20, upper bound -1
   a2 = ? 10    40.3293    100?;   ! Initial value is 20, lower bound 10, upper bound 100
   b1 = ?-10   -3.73265     10?;    ! Initial value is -1, lower bound -10, upper bound 10
   b2 = ? 10    27.5677    50?;    ! Initial value is 20, lower bound 10, upper bound 50

   x  = 1;                   ! Initialize x

        .               ! The circuit file remains unchanged from here on.
        .
        .
```

Report after ℓ_2 optimization

Optimizable Variables Before and After Optimization

Variable	Before optimization	After optimization
a0	0	1.4068
a1	-10	-10.2085
a2	20	40.3293
b1	-1	-3.7327
b2	20	27.5677

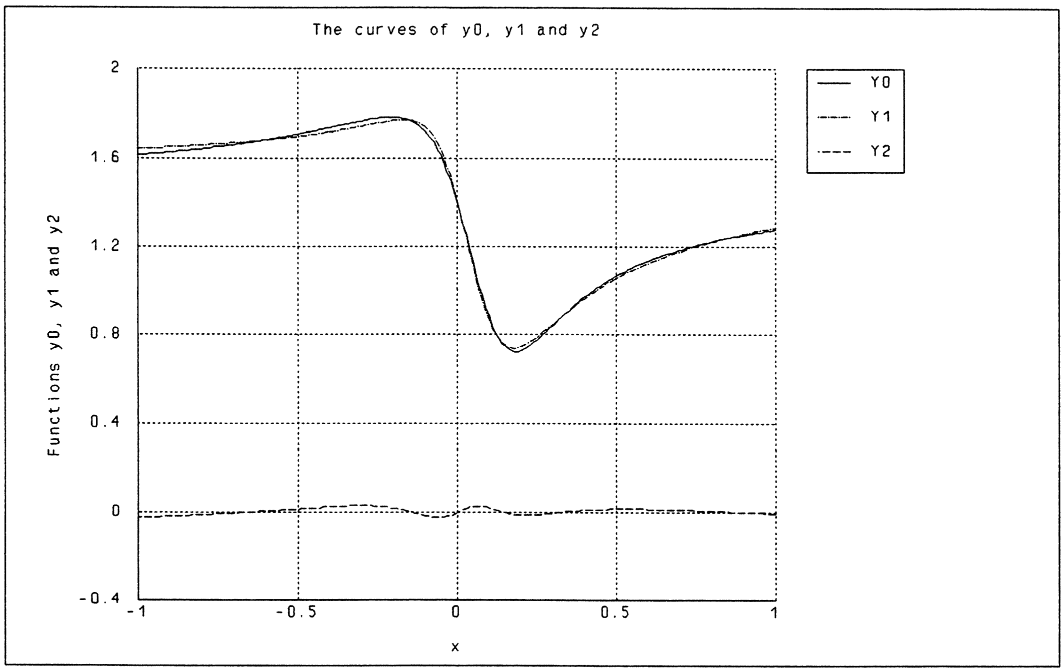

Fig. SQ8004.3 The curves of y_0, y_1 and y_2 after ℓ_2 optimization.

Finally, we use the minimax optimizer of OSA90/hope to carry out the optimization. Every iteration of optimization is listed as follows (as recorded in the osa90_0.log file).

Iterations of minimax optimization

```
13:40:27 Read in File Q8004_0.ckt
13:40:28 Parsing Input File...
13:40:28 File Parsing Completed
13:40:36 Minimax Optimization
13:40:36 Optimization...  Press any key to interrupt
13:40:40 Iteration   1/30 Max Error=1.58846
13:40:43 Iteration   2/30 Max Error=1.57908
13:40:47 Iteration   3/30 Max Error=1.56032
13:40:50 Iteration   4/30 Max Error=1.52331
13:40:54 Iteration   5/30 Max Error=1.45028
13:40:58 Iteration   6/30 Max Error=1.31165
13:41:01 Iteration   7/30 Max Error=1.08806
13:41:06 Iteration   8/30 Max Error=0.814669
13:41:11 Iteration   9/30 Max Error=0.583163
13:41:16 Iteration  10/30 Max Error=0.366531
13:41:22 Iteration  11/30 Max Error=0.410012
13:41:25 Iteration  12/30 Max Error=0.224467
13:41:29 Iteration  13/30 Max Error=0.133198
13:41:34 Iteration  14/30 Max Error=0.290498
13:41:37 Iteration  15/30 Max Error=0.101192
13:41:41 Iteration  16/30 Max Error=0.0861533
13:41:45 Iteration  17/30 Max Error=0.0801742
13:41:49 Iteration  18/30 Max Error=0.0702661
13:41:53 Iteration  19/30 Max Error=0.0532091
13:41:58 Iteration  20/30 Max Error=0.0284787
13:42:02 Iteration  21/30 Max Error=0.0239656
13:42:06 Iteration  22/30 Max Error=0.0237988
13:42:10 Iteration  23/30 Max Error=0.0237942
13:42:14 Iteration  24/30 Max Error=0.0237942
13:42:14 Solution  Max Error=0.0237942
13:42:14 Optimization Completed
13:42:14 Elapsed Real Time: 00:01:38  CPU Time 00:01:34
13:42:20 Save File Q8004_mm.ckt
```

From the above, we can observe that it takes 24 iterations, 1 minute and 34 seconds CPU time for the optimization to converge. The minimax error is reduced from 1.58846 to 0.0237942. The solution is given in the circuit file after minimax optimization. The optimizable variables before and after optimization are listed in the report. The curves of y_0, y_1 and y_2 after minimax optimization are plotted in Fig. SQ8004.4.

Circuit file after minimax optimization

```
!  File name: Q8004_mm.ckt
!  Circuit file for solving Question 8004
!  After minimax optimization

Expression
   ! Define optimizable variables
   a0 = ? -1    1.41457     5?;     ! Initial value is 0, lower bound -1, upper bound 5
   a1 = ?-20   -10.653    -1?;     ! Initial value is -10, lower bound -20, upper bound -1
   a2 = ? 10    41.6058    100?;   ! Initial value is 20, lower bound 10, upper bound 100
   b1 = ?-10   -4.01106     10?;    ! Initial value is -1, lower bound -10, upper bound 10
   b2 = ? 10    28.2548    50?;    ! Initial value is 20, lower bound 10, upper bound 50

   x  = 1;                  ! Initialize x

        .               ! The circuit file remains unchanged from here on.
        .
        .
```

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before optimization	After optimization
a0	0	1.4146
a1	-10	-10.6530
a2	20	41.6058
b1	-1	-4.0111
b2	20	28.2548

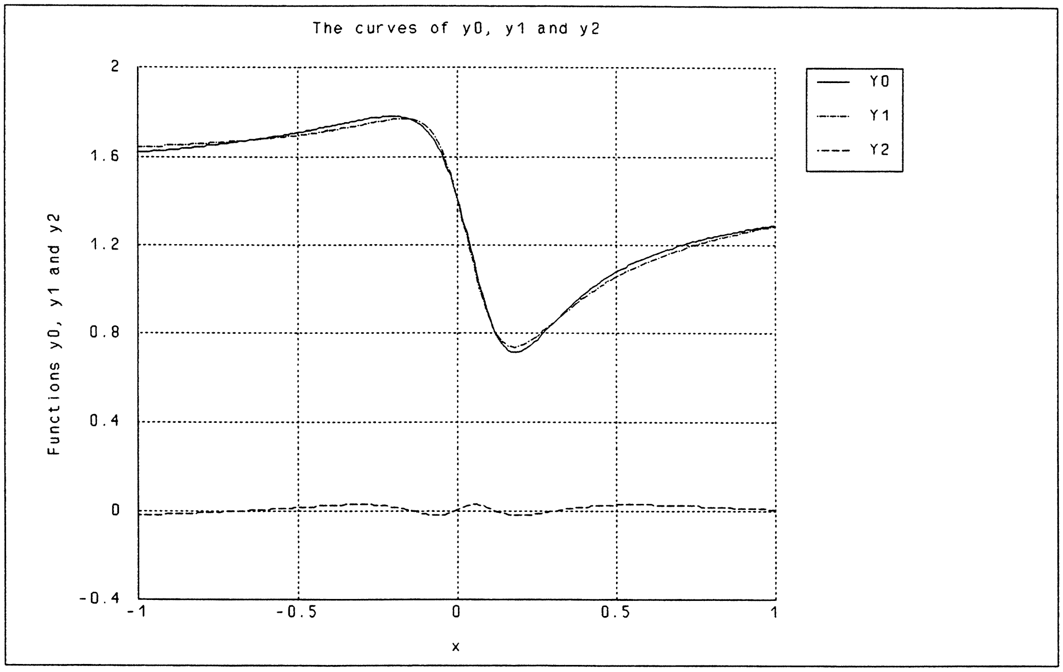

Fig. SQ8004.4 The curves of y_0, y_1 and y_2 after minimax optimization.

OSA☞ Question 8006 Use OSA90/hope to perform a nominal point optimization of the three elements in the LC lowpass filter shown in Fig. Q8005 to satisfy the insertion loss design constraints. (See Question 8005.)

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

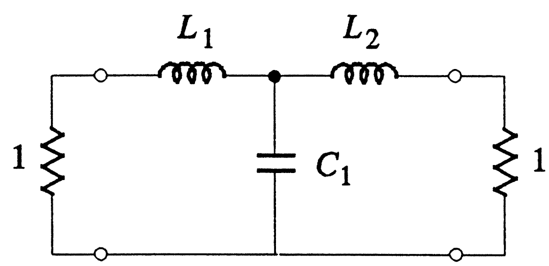

Fig. Q8005 LC lowpass filter.

Solution

In this question, L_1, L_2 and C_1 are the optimizable variables. The following circuit file was implemented to optimize these variables such that the insertion loss constraints were satisfied. The starting points, $L_1 = L_2 = 1.6$ H and $C_1 = 1.0$ F were chosen on the basis that they are close to the final solution.

The circuit file before optimization is below.

Circuit file before optimization

```
!  File name: Q8006_0.ckt
!  Circuit file for solving Question 8006
!  Before optimization

Control
        optimizer = Minimax;
        non_microwave_units;
End

Model
        L1: ? 1.6 ?;
        L2: ? 1.6 ?;
        C1: ? 1.0 ?;

        PORT 1 0 name=input_port R=1oh;
        IND 1 2 L:L1;
        IND 2 3 L:L2;
        CAP 2 0 C:C1;
        RES 3 0 R=1oh;
        circuit;

        omega = 1;
        omega_f = 2.5;
        lower_spec = if ((omega>2.5)+(omega=2.5)) (25) else (-1);
        upper_spec = if ((omega<1)+(omega=1)) (1.5) else (-1);

        INSL = 10*LOG10(1/(1-(MS11)^2));
```

```
End

Sweep
        ac: omega: from 0 to 1 n=30 from 1 to 3 n=30 freq=(1HZ*omega/(2*PI)) RREF=1  INSL lower_spec upper_spec
                { Xsweep TITLE="Insertion Loss vs. Frequency"
                  X_TITLE="Frequency (rad/s)"
                  Y_TITLE="Insertion Loss (dB)"
                  X=omega Xmin=0 Xmax=3.0
                  Y=INSL.green & lower_spec.white.point & upper_spec.white.point
                  Ymin=0 Ymax=26
                  NXticks=6 }

                { Xsweep TITLE="Insertion Loss vs. Frequency"
                  X_TITLE="Frequency (rad/s)"
                  Y_TITLE="Insertion Loss (dB)"
                  X=omega Xmin=0 Xmax=1.5
                  Y=INSL.green & upper_spec.white.point
                  Ymin=0 Ymax=4
                  NXticks=6 NYticks=4 };
End

Specification
        ac: omega: from 0 to 1 n=30 freq=(1HZ*omega/(2*PI)) RREF=1  INSL<1.5;
        ac: freq=(1HZ*omega_f/(2*PI)) RREF=1  INSL>25;
End

Report
          Optimizable Variables Before and After Optimization

    ---------------------------------------------------------------
        Variable        Before Optimization      After Optimization
    ---------------------------------------------------------------
          L1                    1.6                     $% 5.4f$ $L1$
          L2                    1.6                     $% 5.4f$ $L2$
          C1                    1.0                     $% 5.4f$ $C1$
    ---------------------------------------------------------------
End
```

Using minimax optimization, and recording only the best iterations, the following results (as recorded in the osa90_0.log file) were obtained.

Iterations of minimax optimization

```
11:24:51 Read in File Q8006_0.ckt
11:24:59 Parsing Input File...
11:24:59 File Parsing Completed
11:25:05 Minimax Optimization
11:25:05 Optimization...  Press any key to interrupt
11:25:05 Iteration   1/999 Max Error=0.249647
11:25:05 Iteration   2/999 Max Error=0.083347
11:25:05 Iteration   3/999 Max Error=-0.248609
11:25:05 Iteration   4/999 Max Error=-0.897239
11:25:05 Iteration   5/999 Max Error=-0.941367
11:25:05 Iteration   6/999 Max Error=-0.949723
11:25:06 Iteration   7/999 Max Error=-0.966819
11:25:06 Iteration   9/999 Max Error=-0.967953
11:25:06 Iteration  11/999 Max Error=-0.968021
11:25:06 Iteration  12/999 Max Error=-0.968025
11:25:06 Iteration  13/999 Max Error=-0.968025
11:25:06 Solution  Max Error=-0.968025
11:25:06 Optimization Completed
11:25:06 Elapsed Real Time: 00:00:01   CPU Time 00:00:01
```

From the above, we can see that the solution was found in 13 iterations and the minimax error was reduced from 0.249647 to -0.968025. In other words, the constraints were met and exceeded.

Following is the circuit file after minimax optimization.

Circuit file after minimax optimization

```
!  File name: Q8006_mm.ckt
!  Circuit file for solving Question 8006
!  After minimax optimization

Control
        optimizer = Minimax;
        non_microwave_units;
End

Model
        L1: ? 1.62785 ?;
        L2: ? 1.62785 ?;
        C1: ? 1.0898 ?;

        .               ! the circuit file remains unchanged from here on
        .
        .
```

See Fig. SQ8006.1 and SQ8006.2 for the graphical insertion loss results. The first graph shows the loss over the whole frequency range, 0-3 rad/s, and the second graph shows the lower 0-1.5 rad/s region in greater detail.

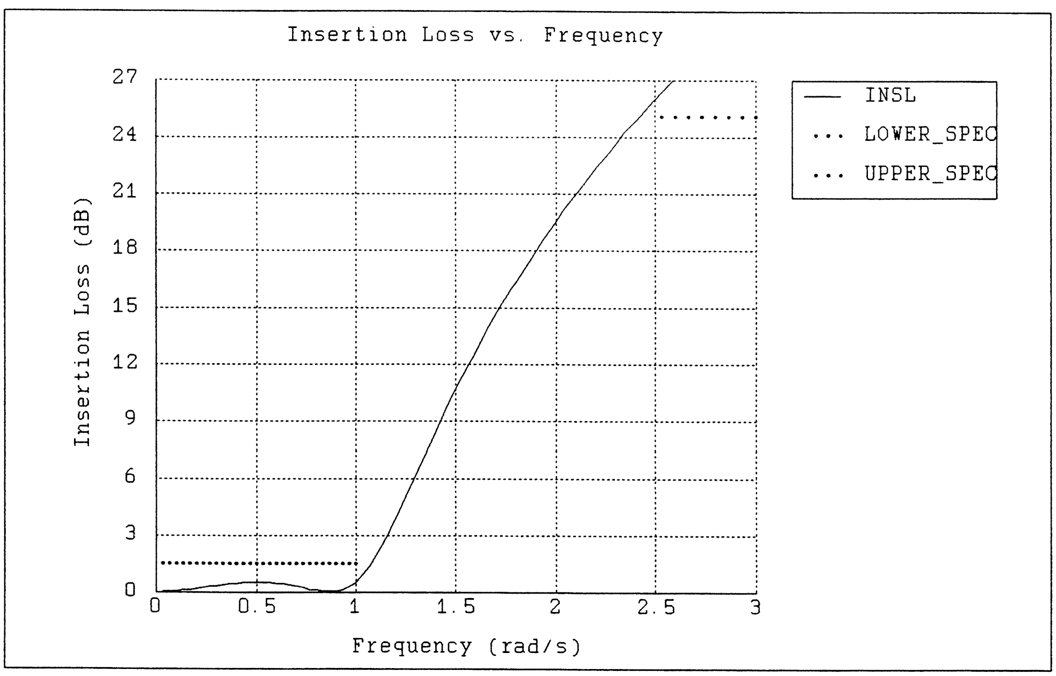

Fig. SQ8006.1 Insertion loss versus frequency.

The report for this problem follows.

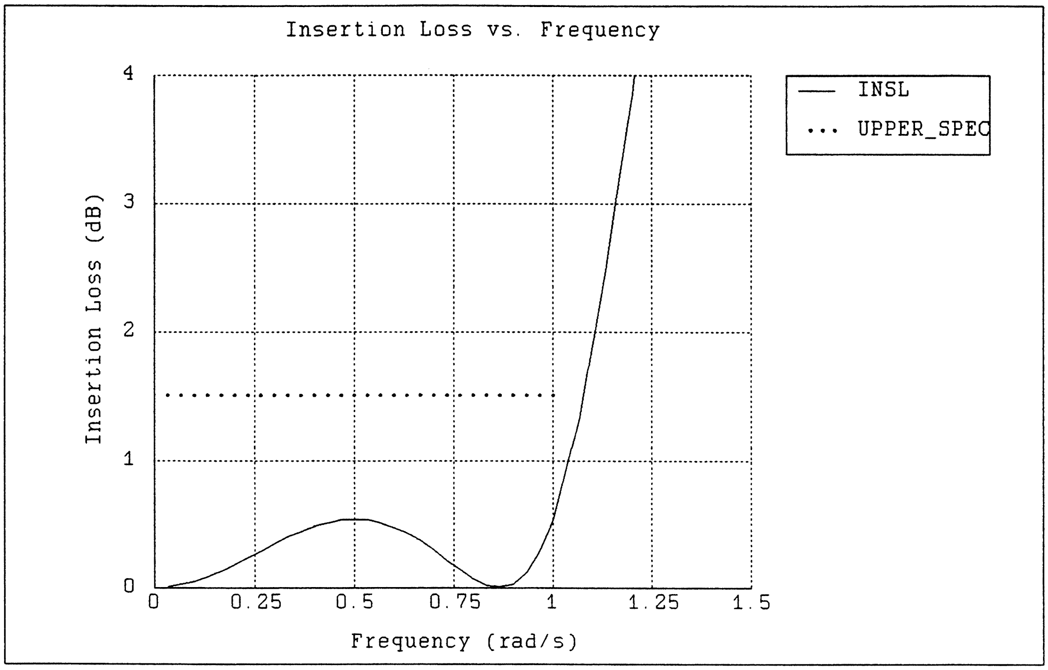

Fig. SQ8006.2 Insertion loss versus frequency (passband detail).

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before Optimization	After Optimization
L1	1.6	1.6279
L2	1.6	1.6279
C1	1.0	1.0898

OSA☞ Question 8007 Use OSA90/hope to perform a worst-case tolerance optimization of the circuit shown in Fig. Q8005 to satisfy the insertion loss constraints. Use an exact-penalty function formulation to achieve this goal. (See Question 8005.)

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

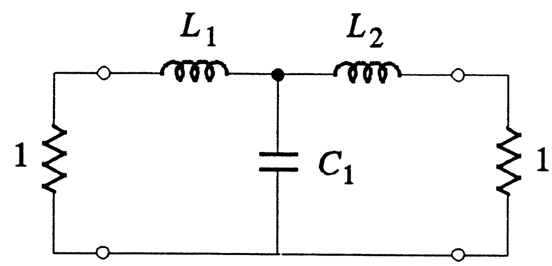

Fig. Q8005 LC lowpass filter.

Solution

This question involves the same circuit as in Question 8006. The optimizable variables include L_1, L_2 and C_1 but the tolerances ϵ_1 and ϵ_2 are also optimizable. The following circuit file was implemented to optimize these variables such that the insertion loss constraints were satisfied. U is the cost function. The exact penalty function, V, is composed of maximum of U, U-αg_1 and U-αg_2. It is minimized using minimax optimization. The results from Question 8006 were chosen as the starting points for the circuit elements and the tolerances were given an initial 10%.

The circuit file before optimization is below.

Circuit file before optimization

```
! File name: Q8007_0.ckt
! Circuit file for solving Question 8007
! Before optimization

Control
        optimizer = Minimax;
        non_microwave_units;
End

Model
        ! Optimizable variables
        L1 : ? 1.62901 ?;
        L2 : ? 1.62669 ?;
        C1 : ? 1.0898 ?;

        e1 : ? 0.0162901 ?;
        e2 : ? 0.0162669 ?;
        e3 : ? 0.010898 ?;

        ! Cost function to minimize
        cost = (L1/e1) + (L2/e2) + (C1/e3);

        ! Multipliers - all 8 vertices
```

```
        u1[8] = [ -1  1 -1  1 -1  1 -1  1 ];
        u2[8] = [ -1 -1  1  1 -1 -1  1  1 ];
        u3[8] = [ -1 -1 -1 -1  1  1  1  1 ];

        ! Calculate element values for each vertex
        I1[8] = L1 + e1*u1;
        I2[8] = L2 + e2*u2;
        CA[8] = C1 + e3*u3;

        k = 0;
        ! Define circuit
        PORT 1 0 name=input_port R=1oh;
        IND 1 2 L=I1[k];
        IND 2 3 L=I2[k];
        CAP 2 0 C=CA[k];
        RES 3 0 R=1oh;
        circuit;

        ! Other labels
        omega = 1;
        omega_f = 2.5;
        lower_spec = if ((omega>2.5)+(omega=2.5)) (25) else (-5);
        upper_spec = if ((omega<1)+(omega=1)) (1.5) else (-5);

        ! For the Report Block only
        ep1 = e1*100/L1;
        ep2 = e2*100/L2;
        ep3 = e3*100/C1;

        ! Define insertion loss (in dB)
        INSL = 10*LOG10(1/(1-(MS11)^2));

        ! Define Exact-Penalty Function Formulation
        alpha = 50;
        U = cost;
        g1 = 1.5 - INSL;
        g2 = INSL - 25;

        Ua = U - (alpha*g1);
        Ub = U - (alpha*g2);
End

Sweep
        ac: k: from 1 to 8 step=1
                omega: from 0 to 1 n=30 from 1 to 3 n=30 freq=(1HZ*omega/(2*PI)) RREF=1  INSL lower_spec
upper_spec
                { Xsweep TITLE="Insertion Loss vs. Frequency"
                  X_TITLE="Frequency (rad/s)"
                  Y_TITLE="Insertion Loss (dB)"
                  X=omega  k=all
                  Xmin=0 Xmax=1.5
                  Y=INSL.green & upper_spec.white.point
                  Ymin=0 Ymax=4
                  NXticks=6 }
                { Xsweep TITLE="Insertion Loss vs. Frequency"
                  X_TITLE="Frequency (rad/s)"
                  Y_TITLE="Insertion Loss (dB)"
                  X=omega  k=all
                  Xmin=0 Xmax=3
                  Y=INSL.green & lower_spec.white.point & upper_spec.white.point
                  Ymin=0 Ymax=27
                  NXticks=6 NYticks=6 };
End

Specification
        U;
        ac: k: from 1 to 8 step=1
                omega: from 0 to 1 n=30 freq=(1HZ*omega/(2*PI)) RREF=1 Ua;
        ac: k: from 1 to 8 step=1
                freq=(1HZ*omega_f/(2*PI)) RREF=1 Ub;
End
```

```
Report
          Optimizable Variables Before and After Optimization

    ---------------------------------------------------------------
        Variable        Before Optimization      After Optimization
    ---------------------------------------------------------------
          L1                 1.62901              $% 5.4f$ $L1$
          L2                 1.62669              $% 5.4f$ $L2$
          C1                 1.0898               $% 5.4f$ $C1$

          e1                   10 %               $% 5.4f$ $ep1$ %
          e2                   10 %               $% 5.4f$ $ep2$ %
          e3                   10 %               $% 5.4f$ $ep3$ %
    ---------------------------------------------------------------
End
```

Using minimax optimization, and recording only the best iterations, the following results (as recorded in the osa90_0.log file) were obtained.

Iterations of minimax optimization

```
17:48:24 Read in File Q8007_0.ckt
17:54:30 Parsing Input File ...
17:54:30 File Parsing Completed
17:55:09 Minimax Optimization
17:55:09 Optimization ...  Press any key to interrupt
17:55:11 Iteration   1/999 Max Error=300
17:55:13 Iteration   2/999 Max Error=297.56
17:55:14 Iteration   3/999 Max Error=292.741
17:55:16 Iteration   4/999 Max Error=283.715
17:55:17 Iteration   5/999 Max Error=271.003
17:55:19 Iteration   6/999 Max Error=247.341
17:55:20 Iteration   7/999 Max Error=206.263
17:55:22 Iteration   8/999 Max Error=144.639
17:55:25 Iteration  10/999 Max Error=86.9464
17:55:26 Iteration  11/999 Max Error=56.5849
17:55:29 Iteration  13/999 Max Error=36.7024
17:55:31 Iteration  14/999 Max Error=35.6761
17:55:32 Iteration  15/999 Max Error=33.4834
17:55:36 Iteration  17/999 Max Error=33.3767
17:55:39 Iteration  19/999 Max Error=33.3571
17:55:40 Iteration  20/999 Max Error=33.3565
17:55:43 Iteration  22/999 Max Error=33.3536
17:55:45 Iteration  23/999 Max Error=33.3526
17:55:49 Iteration  26/999 Max Error=33.3526
17:55:52 Iteration  28/999 Max Error=33.3526
17:55:56 Iteration  30/999 Max Error=33.3526
17:55:57 Iteration  31/999 Max Error=33.3526
17:55:59 Iteration  32/999 Max Error=33.3526
17:56:00 Iteration  33/999 Max Error=33.3526
17:56:00 Solution  Max Error=33.3526
17:56:00 Optimization Completed
17:56:00 Elapsed Real Time: 00:00:51   CPU Time 00:00:50
17:58:12 Save File Q8007_mm.ckt
```

From the above, we can see that the solution was found in 33 iterations and the minimax error was reduced from 300 to 33.3526.

Here is the circuit file after minimax optimization.

Circuit file after minimax optimization

```
!  File name: Q8007_mm.ckt
!  Circuit file for solving Question 8007
!  After minimax optimization
```

```
Control
        optimizer = Minimax;
        non_microwave_units;
End

Model
        ! Optimizable variables
        L1 : ? 1.99903 ?;
        L2 : ? 1.99903 ?;
        C1 : ? 0.905617 ?;

        e1 : ? 0.19719 ?;
        e2 : ? 0.19719 ?;
        e3 : ? 0.0692506 ?;

        .                ! the circuit file remains unchanged from here on
        .
        .
```

See Fig. SQ8007.1 and SQ8007.2 for the insertion loss results graphically. The first graph shows the loss over the whole frequency range, 0-3 rad/s, and the second graph shows the lower 0-1.5 rad/s region in greater detail. Note that eight curves can be seen on both graphs. They correspond to the eight worst-case vertex insertion loss responses of this three element toleranced problem.

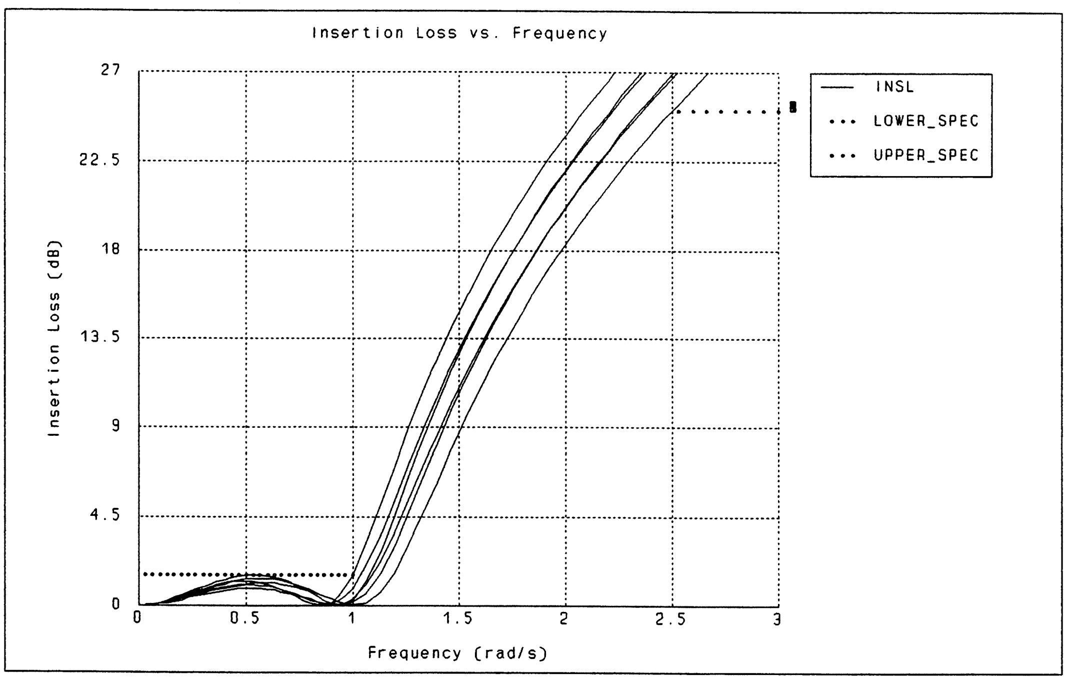

Fig. SQ8007.1 Insertion loss versus frequency.

The report for this problem follows.

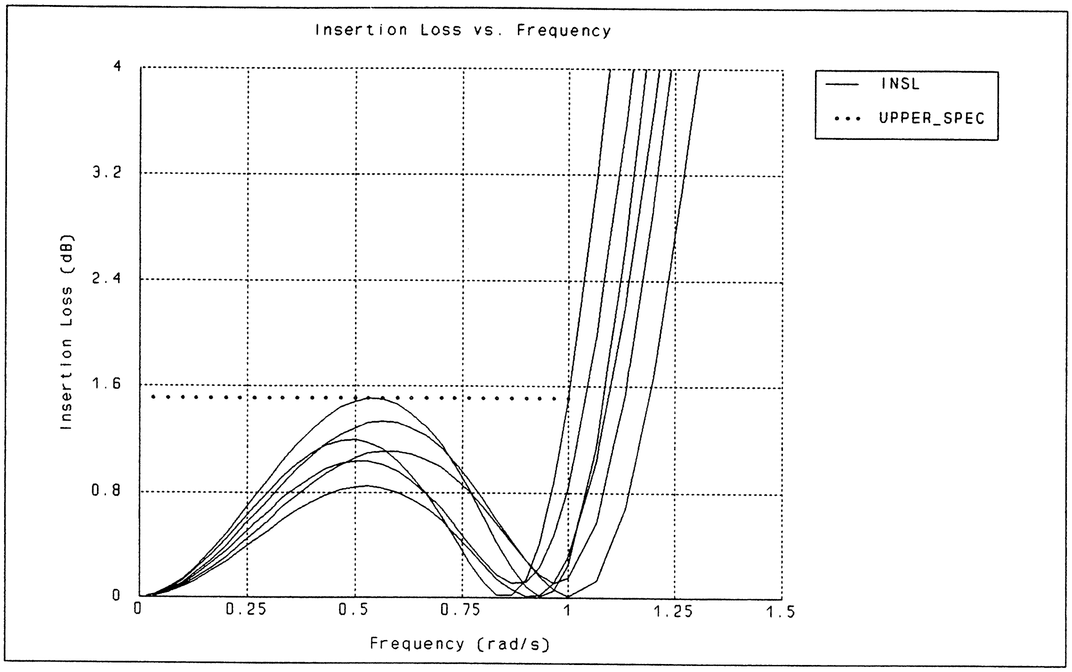

Fig. SQ8007.2 Insertion loss versus frequency (passband detail).

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before Optimization	After Optimization
L1	1.62901	1.9990
L2	1.62669	1.9990
C1	1.0898	0.9056
e1	10 %	9.8643 %
e2	10 %	9.8643 %
e3	10 %	7.6468 %

OSA☞ Question 8011 Use the ℓ_1, ℓ_2 and minimax optimizers of OSA90/hope to make the function

$$F(x) = \frac{a_3}{a_2}\exp(-a_1 x)\sin(a_2 x)$$

best approximate

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

in a uniformly weighted sense. The problem may be discretized on the interval 0 to 10. Suggested starting point: $a_1 = a_2 = a_3 = 1$. (See Question 8010.)

Solution

In this question, a_1, a_2, and a_3 are optimizable variables. The suggested starting point is a_1 $a_2 = a_3 = 1$, which is indicated in the circuit file before optimization. The solutions for a_1, a_2 and a_3 are given in the circuit files after optimization.

We let

$$y_0 = \frac{a_3}{a_2}\exp(-a_1 x)\sin(a_2 x)$$

$$y_1 = \frac{3}{20}\exp(-x) + \frac{1}{52}\exp(-5x) - \frac{\exp(-2x)}{65}[3\sin(2x) + 11\cos(2x)]$$

$$y_2 = y_0 - y_1$$

The circuit file before optimization is below.

Circuit file before optimization

```
!  File name: Q8011_0.ckt
!  Circuit file for solving Question 8011
!  Before optimization

Expression
   ! Define the optimizable variables
   a1 = ?-50    1    50?;    ! Initial value is 1, lower bound -50 and upper bound 50
   a2 = ?-50    1    50?;    ! Initial value is 1, lower bound -50 and upper bound 50
   a3 = ?-50    1    50?;    ! Initial value is 1, lower bound -50 and upper bound 50

   x  = 1;                   ! Initialize x

   ! Define functions
   y0 = a3 / a2 * exp(-a1 * x) * sin(a2 * x);
   y1 = 3.0 / 20 * exp(-x) + 1.0 / 52 * exp(-5 * x) - exp(-2 * x) / 65.0 * (3 * sin(2 * x) + 11 * cos(2 * x));

   ! Calculate error function
   y2 = y0 - y1;
End
```

```
! Define the sweep parameter and the output responses

Sweep
   x: from 0 to 10 step 0.05 y0 y1 y2
   {Xsweep  Title = "The curves of y, y1 and y2"
    X_title = "x"  X = x
    Y_title = "Functions y0, y1 and y2"  Y = y0.green & y1.yellow & y2.red
    Ymin = -0.05  Ymax = 0.35  NYTicks = 8};
End

! Define the specifications for optimization

Specification
   x: from 0 to 10 step 0.05 y2=0;
End

! Define report block

Report
      Optimizable Variables Before and After Optimization

   --------------------------------------------------------
     Variable    Before optimization    After optimization
   --------------------------------------------------------
        a1              0                 $% 5.4f$ $a1$
        a2            -10                 $% 5.4f$ $a2$
        a3             20                 $% 5.4f$ $a3$
   --------------------------------------------------------
End
```

The curves of y_0, y_1 and y_2 before optimization are shown in Fig. SQ8011.1. From Fig. SQ8011.1 we can see that the error y_2 is very large.

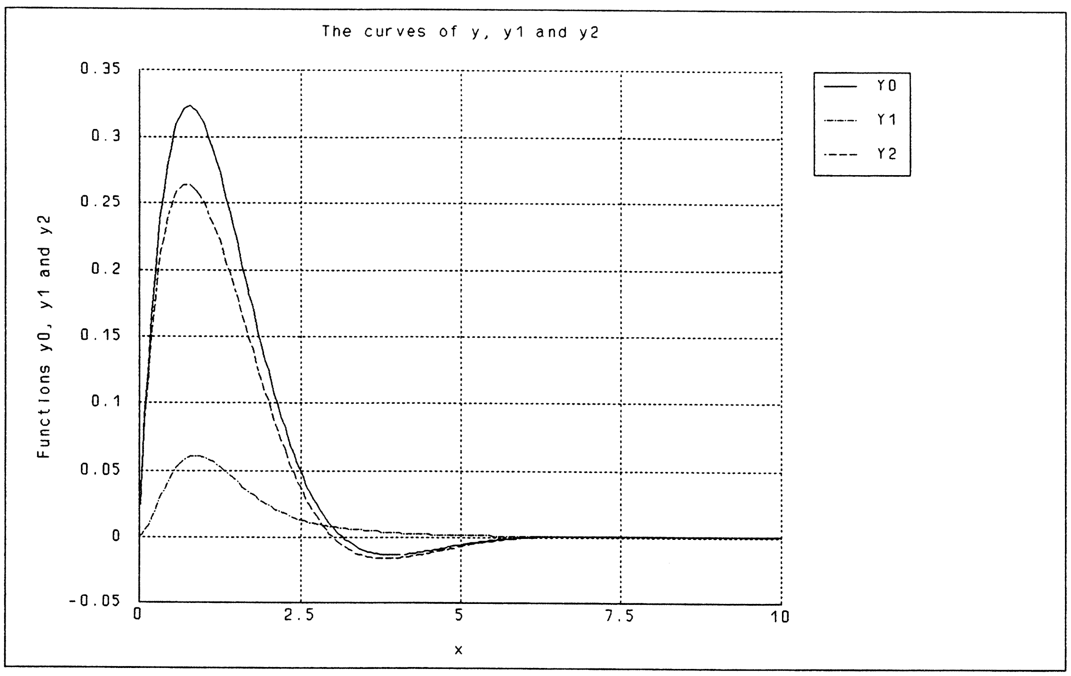

Fig. SQ8011.1 The curves of y_0, y_1 and y_2 before optimization.

First, we use the ℓ_1 optimizer of OSA90/hope to carry out the optimization. Every iteration of optimization is listed as follows (which is recorded in the osa90_0.log file).

Iterations of ℓ_1 optimization

```
11:09:19 Read in File Q8011_0.ckt
11:09:40 Parsing Input File...
11:09:40 File Parsing Completed
11:09:48 L1 Optimization
11:09:48 Optimization... Press any key to interrupt
11:09:51 Iteration   1/30 L1 Objective=9.16184
11:09:54 Iteration   2/30 L1 Objective=8.84924
11:09:57 Iteration   3/30 L1 Objective=8.25477
11:10:00 Iteration   4/30 L1 Objective=7.18039
11:10:03 Iteration   5/30 L1 Objective=5.42004
11:10:06 Iteration   6/30 L1 Objective=3.17153
11:10:09 Iteration   7/30 L1 Objective=0.951022
11:10:12 Iteration   8/30 L1 Objective=0.729988
11:10:15 Iteration   9/30 L1 Objective=0.521008
11:10:19 Iteration  10/30 L1 Objective=0.651634
11:10:22 Iteration  11/30 L1 Objective=0.325556
11:10:25 Iteration  12/30 L1 Objective=0.323995
11:10:28 Iteration  13/30 L1 Objective=0.31974
11:10:31 Iteration  14/30 L1 Objective=0.319505
11:10:34 Iteration  15/30 L1 Objective=0.319505
11:10:34 Solution  L1 Objective=0.319505
11:10:34 Optimization Completed
11:10:34 Elapsed Real Time: 00:00:46   CPU Time 00:00:46
11:11:09 Save File Q8011_l1.ckt
```

From the above, we can observe that it takes 15 iterations, 46 seconds CPU time for the optimization to converge. The ℓ_1 error is reduced from 9.16184 to 0.319505. The solution is given in the circuit file after ℓ_1 optimization. The optimizable variables before and after optimization are listed in the report. The curves of y_0, y_1 and y_2 after ℓ_1 optimization are plotted in Fig. SQ8011.2.

Circuit file after ℓ_1 optimization

```
!  File name: Q8011_l1.ckt
!  Circuit file for solving Question 8011
!  After l1 optimization

Expression
   ! Define the optimizable variables
   a1 = ?-50    1.29769     50?;   ! Initial value is 1, lower bound -50 and upper bound 50
   a2 = ?-50    0.547251     50?;   ! Initial value is 1, lower bound -50 and upper bound 50
   a3 = ?-50    0.20612     50?;   ! Initial value is 1, lower bound -50 and upper bound 50

   x  = 1;                  ! Initialize x

         .               ! The circuit file remains unchanged from here on.
         .
         .
```

Report after ℓ_1 optimization

Optimizable Variables Before and After Optimization

Variable	Before optimization	After optimization
a1	0	1.2977
a2	-10	0.5473
a3	20	0.2061

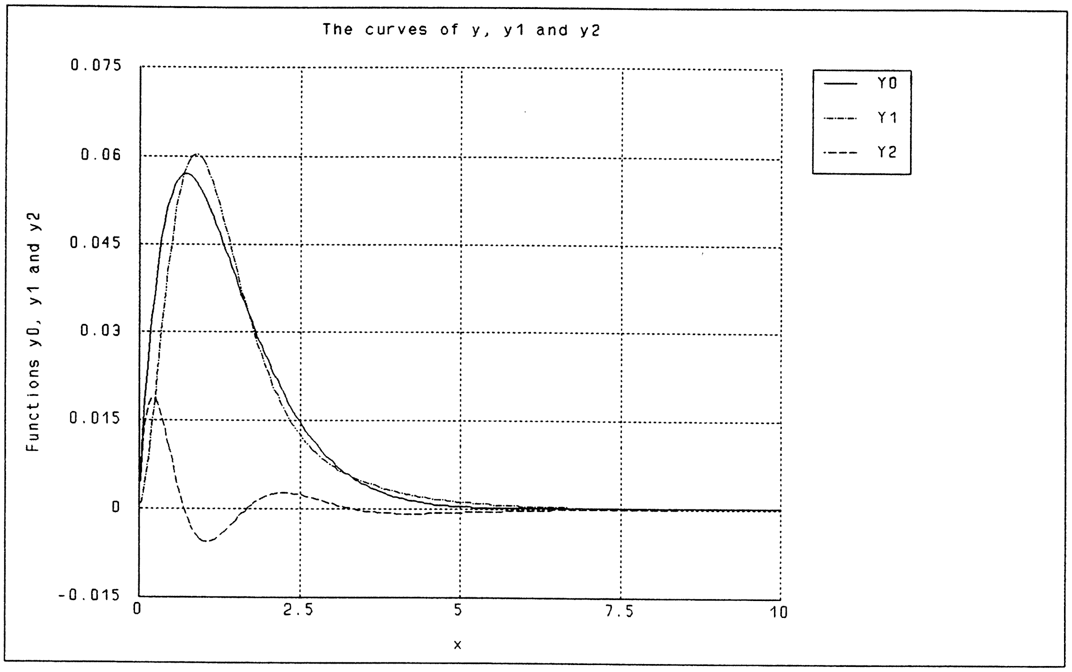

Fig. SQ8011.2 The curves of y_0, y_1 and y_2 after ℓ_1 optimization.

Secondly, we use the ℓ_2 optimizer of OSA90/hope to carry out the optimization. Every iteration of optimization is listed as follows (as recorded in the osa90_0.log file).

Iterations of ℓ_2 optimization

```
11:11:53 Read in File Q8011_0.ckt
11:11:59 Parsing Input File...
11:11:59 File Parsing Completed
11:12:05 L2 Optimization
11:12:05 Optimization...  Press any key to interrupt
11:12:07 Iteration   1/30 L2 Objective=1.70703
11:12:10 Iteration   2/30 L2 Objective=0.981621
11:12:13 Iteration   3/30 L2 Objective=0.493863
11:12:16 Iteration   4/30 L2 Objective=0.208806
11:12:19 Iteration   5/30 L2 Objective=0.0703339
11:12:22 Iteration   6/30 L2 Objective=0.0210645
11:12:25 Iteration   7/30 L2 Objective=0.010412
11:12:28 Iteration   8/30 L2 Objective=0.00809877
11:12:30 Iteration   9/30 L2 Objective=0.00580381
11:12:33 Iteration  10/30 L2 Objective=0.00337154
11:12:36 Iteration  11/30 L2 Objective=0.00204857
11:12:39 Iteration  12/30 L2 Objective=0.00180834
11:12:42 Iteration  13/30 L2 Objective=0.0017962
11:12:42 Solution  L2 Objective=0.0017962
11:12:42 Optimization Completed
11:12:42 Elapsed Real Time: 00:00:37   CPU Time 00:00:37
11:12:54 Save File Q8011_l2.ckt
```

From the above, we can observe that it takes 13 iterations, 37 seconds CPU time for the optimization to converge. The ℓ_2 error is reduced from 1.70703 to 0.0017962. The solution is given in the circuit file after ℓ_2 optimization. The optimizable variables before and after optimization are listed in the report. The curves of y_0, y_1 and y_2 after ℓ_2 optimization are plotted in Fig. SQ8011.3.

Circuit file after ℓ_2 optimization

```
!  File name: Q8011_12.ckt
!  Circuit file for solving Question 8011
!  After 12 optimization

Expression
   ! Define the optimizable variables
   a1 = ?-50    1.01823    50?;   ! Initial value is 1, lower bound -50 and upper bound 50
   a2 = ?-50    0.78765    50?;   ! Initial value is 1, lower bound -50 and upper bound 50
   a3 = ?-50    0.161513    50?;   ! Initial value is 1, lower bound -50 and upper bound 50

   x  = 1;                  ! Initialize x

        .                ! The circuit file remains unchanged from here on.
        .
        .
```

Report after ℓ_2 optimization

```
      Optimizable Variables Before and After Optimization

   --------------------------------------------------------
     Variable   Before optimization   After optimization
   --------------------------------------------------------
        a1              0                  1.0182
        a2            -10                  0.7876
        a3             20                  0.1615
   --------------------------------------------------------
```

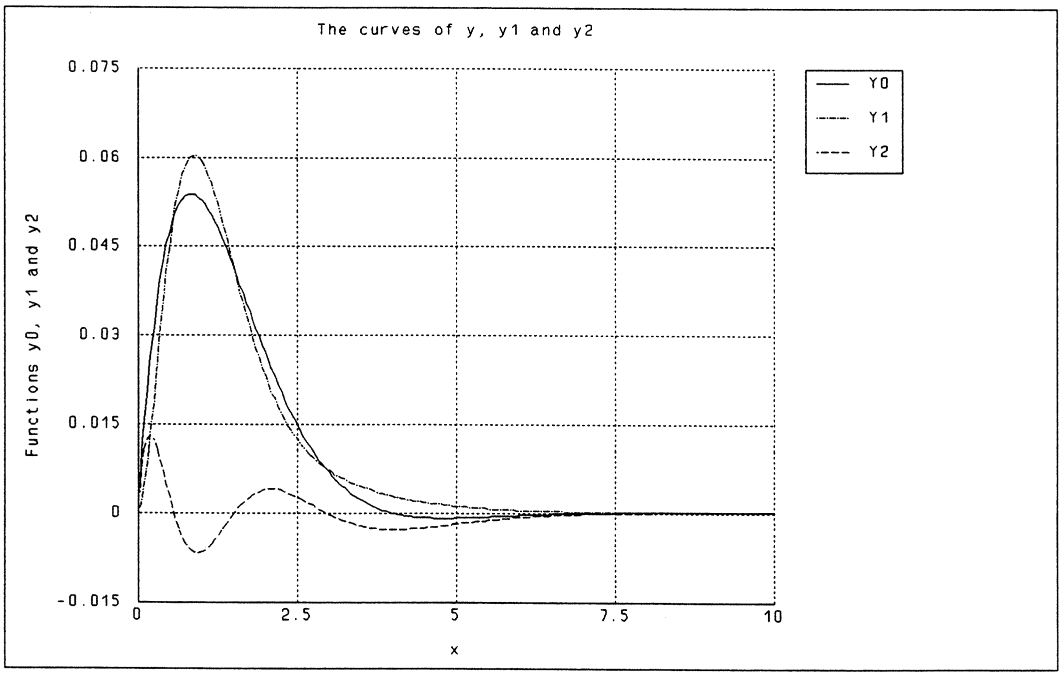

Fig. SQ8011.3 The curves of y_0, y_1 and y_2 after ℓ_2 optimization.

Finally, we use the minimax optimizer of OSA90/hope to carry out the optimization. Every iteration of optimization is listed as follows (as recorded in the osa90_0.log file).

Iterations of minimax optimization

```
11:13:51 Read in File Q8011_0.ckt
11:13:54 Parsing Input File...
11:13:54 File Parsing Completed
11:14:02 Minimax Optimization
11:14:02 Optimization...  Press any key to interrupt
11:14:05 Iteration   1/30 Max Error=0.26362
11:14:08 Iteration   2/30 Max Error=0.259562
11:14:10 Iteration   3/30 Max Error=0.251539
11:14:13 Iteration   4/30 Max Error=0.23623
11:14:16 Iteration   5/30 Max Error=0.207099
11:14:19 Iteration   6/30 Max Error=0.154838
11:14:22 Iteration   7/30 Max Error=0.0670166
11:14:26 Iteration   8/30 Max Error=0.0476161
11:14:30 Iteration   9/30 Max Error=0.0279168
11:14:33 Iteration  10/30 Max Error=0.188955
11:14:37 Iteration  11/30 Max Error=0.045705
11:14:40 Iteration  12/30 Max Error=0.0173404
11:14:43 Iteration  13/30 Max Error=0.0107565
11:14:46 Iteration  14/30 Max Error=0.00990717
11:14:49 Iteration  15/30 Max Error=0.00955978
11:14:51 Iteration  16/30 Max Error=0.00897721
11:14:55 Iteration  17/30 Max Error=0.0090115
11:14:57 Iteration  18/30 Max Error=0.00861224
11:15:00 Iteration  19/30 Max Error=0.00841946
11:15:03 Iteration  20/30 Max Error=0.00825959
11:15:06 Iteration  21/30 Max Error=0.00812111
11:15:09 Iteration  22/30 Max Error=0.00812029
11:15:09 Solution  Max Error=0.00812029
11:15:09 Optimization Completed
11:15:09 Elapsed Real Time: 00:01:07   CPU Time 00:01:07
11:15:20 Save File Q8011_mm.ckt
```

From the above, we can observe that it takes 22 iterations, 1 minute and 7 seconds CPU time for the optimization to converge. The minimax error is reduced from 0.26362 to 0.00812029. The solution is given in the circuit file after minimax optimization. The optimizable variables before and after optimization are listed in the report. The curves of y_0, y_1 and y_2 after minimax optimization are plotted in Fig. SQ8011.4.

Circuit file after minimax optimization

```
!  File name: Q8011_mm.ckt
!  Circuit file for solving Question 8011
!  After minimax optimization

Expression
   ! Define the optimizable variables
   a1 = ?-50    0.675997    50?;   ! Initial value is 1, lower bound -50 and upper bound 50
   a2 = ?-50    0.956314    50?;   ! Initial value is 1, lower bound -50 and upper bound 50
   a3 = ?-50    0.121686    50?;   ! Initial value is 1, lower bound -50 and upper bound 50

   x  = 1;                  ! Initialize x

        .              ! The circuit file remains unchanged from here on.
        .
        .
```

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before optimization	After optimization
a1	0	0.6760
a2	-10	0.9563
a3	20	0.1217

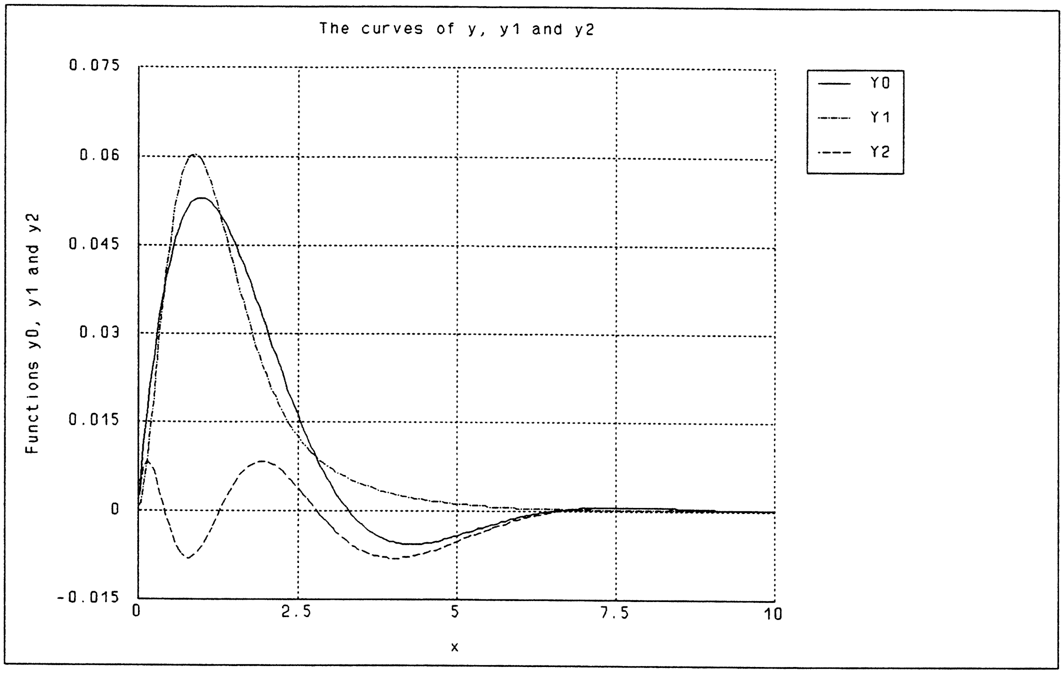

Fig. SQ8011.4 The curves of y_0, y_1 and y_2 after minimax optimization.

OSA☞ Question 8017 Use OSA90/hope to perform a nominal point optimization of the two section lengths and the characteristic impedances of the circuit shown in Fig. Q8015. Minimize the maximum of the modulus of the reflection coefficient ρ over 100% relative bandwidth. (See Question 8015.)

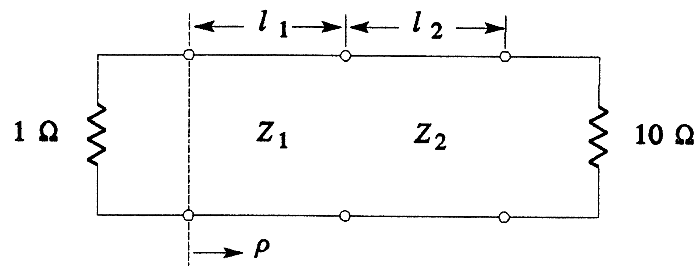

Fig. Q8015 Two–section transmission–line transformer example.

Solution

In this question, the characteristic impedances, Z_1 and Z_2, and their corresponding section lengths, l_1 and l_2, are the optimizable variables. The following circuit file was implemented to optimize these variables such that the modulus of the reflection coefficient was minimized. The starting points, $Z_1 = 2\ \Omega$, $Z_2 = 4\ \Omega$ and $l_1 = l_2 = 7.49481$ cm were chosen.

The circuit file before optimization is below.

Circuit file before optimization

```
!  File name: Q8017_0.ckt
!  Circuit file for solving Question 8017
!  Before optimization

Control
        optimizer = Minimax;
End

Model
        ! Lq = 7.49481cm;          ! quarter wavelength at centre freq. for 1GHz

        ! Parameters of transmission line
        Zo = 1oh;                  ! characteristic impedance of input
        Zload = 10oh;              ! load impedance

        Z1:? 2.0 ?;                ! characteristic impedance of section 1
        Z2:? 4.0 ?;                ! characteristic impedance of section 2

        L1:? 7.49481cm ?;          ! section 1 length
        L2:? 7.49481cm ?;          ! section 2 length

        ! define transmission line parameters
        PORT 1 0 NAME=input R=Zo;
        TRL 1 2 Z=Z1 L=L1;
        TRL 2 3 Z=Z2 L=L2;
        RES 3 0 R=Zload;
        circuit;
End

Sweep
        ac: freq: from 0.5 to 1.5 n=200 RREF=1  MS11
               {      Xsweep TITLE="Reflection Coefficient vs. Frequency"
                       X=freq
                       Y=MS11.green
                       x_title="Frequency (GHz)"
                       y_title="MS11"
```

```
                         xmin=0.5 ymin=0
                         xmax=1.5 ymax=0.5       };
End

Specification
     ac: freq: from 0.5 to 1.5 n=100 RREF=1  MS11;
End

Report
      Optimizable Variables Before and After Optimization

   ---------------------------------------------------------------
      Variable        Before Optimization      After Optimization
   ---------------------------------------------------------------
         Z1                  2.0                   $% 6.5f$ $Z1$
         Z2                  4.0                   $% 6.5f$ $Z2$

         L1               0.074948                 $% 6.5f$ $L1$
         L2               0.074941                 $% 6.5f$ $L2$
   ---------------------------------------------------------------
End
```

Using minimax optimization, and recording only the best iterations, the following results (as recorded in the osa90_0.log file) were obtained.

Iterations of minimax optimization

```
11:09:44 Read in File Q8017_0.ckt
11:09:50 Parsing Input File...
11:09:50 File Parsing Completed
11:09:56 Minimax Optimization
11:09:56 Optimization...  Press any key to interrupt
11:09:56 Iteration   1/999 Max Error=0.439972
11:09:57 Iteration   2/999 Max Error=0.435073
11:09:57 Iteration   3/999 Max Error=0.433493
11:09:57 Iteration   4/999 Max Error=0.431136
11:09:58 Iteration   5/999 Max Error=0.429526
11:09:58 Iteration   7/999 Max Error=0.42866
11:09:59 Iteration   8/999 Max Error=0.428586
11:09:59 Iteration   9/999 Max Error=0.428571
11:09:59 Iteration  10/999 Max Error=0.428571
11:09:59 Solution  Max Error=0.428571
11:09:59 Optimization Completed
11:09:59 Elapsed Real Time: 00:00:03   CPU Time 00:00:03
11:10:22 Save File Q8017_mm.ckt
```

From the above, we can see that the solution was reached in 10 iterations and the minimax error was reduced from.439972 to.428571.

Here is the circuit file after minimax optimization.

Circuit file after minimax optimization

```
!  File name: Q8017_mm.ckt
!  Circuit file for solving Question 8017
!  After optimization

Control
        optimizer = Minimax;
End

Model
        ! Lq = 7.49481cm;          ! quarter wavelength at centre freq. for 1GHz

        ! Parameters of transmission line
        Zo = 1oh;                  ! characteristic impedance of input
        Zload = 10oh;              ! load impedance
```

```
Z1:? 2.23607 ?;          ! characteristic impedance of section 1
Z2:? 4.47214 ?;          ! characteristic impedance of section 2

L1:? 7.50001cm ?;        ! section 1 length
L2:? 7.49999cm ?;        ! section 2 length

.                ! the circuit file remains unchanged from here on
.
.
```

See Fig. SQ8017 for the reflection coefficient (MS11) results graphically over the range of 0.5-1.5 GHz.

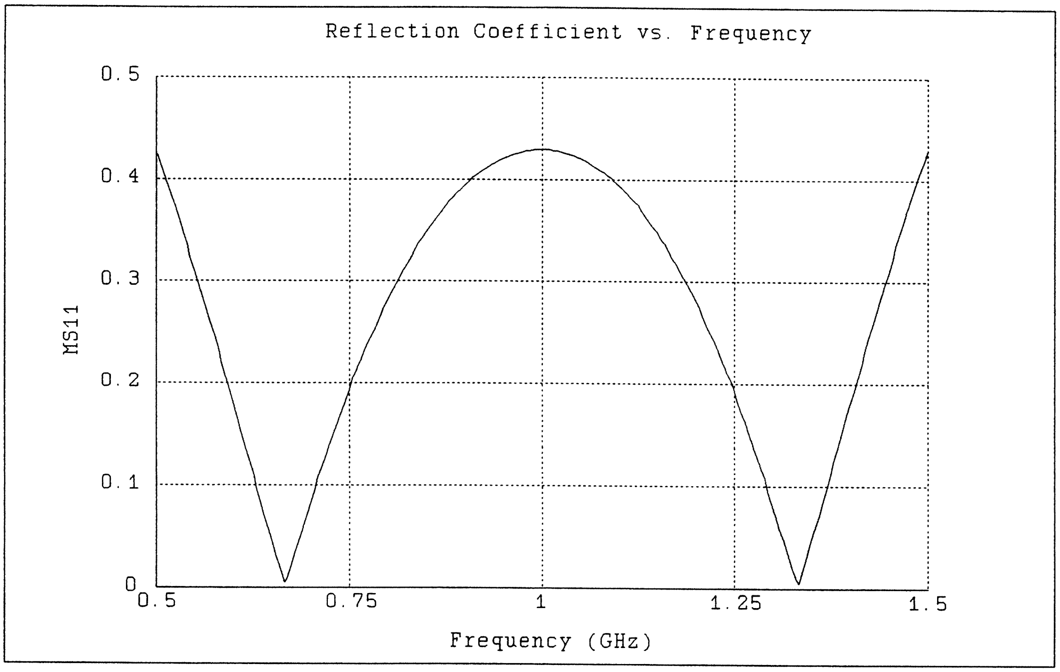

Fig. SQ8017 Reflection coefficient modulus versus frequency.

The report for this problem follows.

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before Optimization	After Optimization
Z1	2.0	2.23607
Z2	4.0	4.47214
L1	0.074948	0.07500
L2	0.074941	0.07500

OSA☛ Question 8018 Use OSA90/hope to perform a worst-case tolerance optimization of the circuit shown in Fig. Q8015 to satisfy the constraint $|\rho| \leq 0.55$. Use an exact-penalty function formulation to achieve this goal for both the relative and absolute cost functions given.

(a) $C_1 = \dfrac{1}{\epsilon_1} + \dfrac{1}{\epsilon_2}$

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

(See Question 8016)

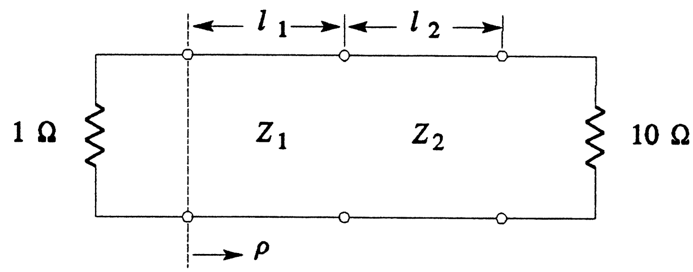

Fig. Q8015 Two-section transmission-line transformer example.

Solution

Here, the nominal characteristic impedances, Z_1 and Z_2, and the tolerances, ϵ_1 and ϵ_2, are the optimizable variables. The corresponding section lengths, l_1 and l_2, are fixed at 7.49481 cm. The following circuit file was implemented to optimize these variables such that the modulus of the reflection coefficient was constrained to be below 0.55 over the range 0.5-1.5 GHz. The starting points for the characteristic impedances were taken from the results of Question 8017.

The circuit file before optimization is below. Note that we have two cost functions and two alphas corresponding to Part 1 and Part 2.

Circuit file for Part 1 before optimization

```
!  File name: Q8018_0.ckt
!  Circuit file for solving Question 8018
!  Before optimization

Control
        optimizer = minimax;
End

Model
        ! Parameters of transmission line
        Zo = 1oh;                       ! char. imp. of input
        Zload = 10oh;                   ! load impedance

        Z1 = ? 2.23836 ?;               ! char. imp. of section 1
        Z2 = ? 4.47672 ?;               ! char. imp. of section 2

        e1 = ? 0.223836 ?;              ! tolerance of Z1
        e2 = ? 0.447672 ?;              ! tolerance of Z2
```

```
        ! for report block only
        ep1 = e1*100/Z1;
        ep2 = e2*100/Z2;

        ! Cost function to minimize.
        cost = 1/e1 + 1/e2;              ! cost for Part 1
        !cost = (Z1/e1) + (Z2/e2);       ! cost for Part 2

        u1[4] = [ -1  1 -1  1 ];
        u2[4] = [ -1 -1  1  1 ];

        a[4] = Z1 + e1*u1;
        b[4] = Z2 + e2*u2;

        k=0;
        limit = 0.55;

        ! define transmission line
        PORT 1 0 NAME=input R=Zo;
        TRL 1 2  Z=a[k]  L=7.49481cm;
        TRL 2 3  Z=b[k]  L=7.49481cm;
        RES 3 0  R=Zload;
        circuit;

        ! Define exact-penalty function formulation
        alpha = 50;              ! alpha for Part 1
        !alpha = 150;            ! alpha for Part 2
        U = cost;
        g = 0.55 - MS11;
        Ua = U  - (alpha*g);
End

Sweep
        ac: k: 1 2 3 4
                freq: from 0.5 to 1.5 n=100 RREF=1  MS11 limit
                        {       Xsweep TITLE="Reflection Coefficient vs. Frequency"
                                k=all
                                x=freq
                                x_title="Frequency (GHz)"
                                y=MS11.green & limit.yellow
                                y_title="MS11"
                                xmin=0.5 xmax=1.5
                                ymin=0 ymax=0.6
                                NXTICKS=4 NYTICKS=6      };
End

Specification
        U;
        ac: k: 1 2 3 4
                freq: from 0.5 to 1.5 n=100 RREF=1 Ua;
End

Report
          Optimizable Variables Before and After Optimization

    ---------------------------------------------------------------
        Variable        Before Optimization     After Optimization
    ---------------------------------------------------------------
           Z1                2.23836               $% 6.5f$ $Z1$
           Z2                4.47672               $% 6.5f$ $Z2$

           e1                  10 %                $% 6.5f$ $ep1$ %
           e2                  10 %                $% 6.5f$ $ep2$ %
    ---------------------------------------------------------------
End
```

Part 1 Results

Using minimax optimization, and recording only the best iterations, the following results (as recorded in the osa90_0.log file) were obtained.

Iterations of minimax optimization

```
18:30:28 Read in File Q8018_0.ckt
18:33:11 Parsing Input File ...
18:33:11 File Parsing Completed
18:33:20 Minimax Optimization
18:33:20 Optimization ...  Press any key to interrupt
18:33:22 Iteration   1/999 Max Error=8.08001
18:33:23 Iteration   2/999 Max Error=7.59745
18:33:25 Iteration   3/999 Max Error=6.6697
18:33:26 Iteration   4/999 Max Error=6.47744
18:33:28 Iteration   5/999 Max Error=6.11445
18:33:29 Iteration   6/999 Max Error=5.47517
18:33:31 Iteration   7/999 Max Error=4.85622
18:33:34 Iteration   9/999 Max Error=4.73793
18:33:36 Iteration  10/999 Max Error=4.73634
18:33:37 Iteration  11/999 Max Error=4.7003
18:33:39 Iteration  12/999 Max Error=4.6927
18:33:40 Iteration  13/999 Max Error=4.688
18:33:42 Iteration  14/999 Max Error=4.68725
18:33:45 Iteration  16/999 Max Error=4.68607
18:33:47 Iteration  17/999 Max Error=4.68575
18:33:50 Iteration  19/999 Max Error=4.6855
18:33:51 Iteration  20/999 Max Error=4.68542
18:33:53 Iteration  21/999 Max Error=4.68537
18:33:54 Iteration  22/999 Max Error=4.68534
18:33:56 Iteration  23/999 Max Error=4.68534
18:33:57 Iteration  24/999 Max Error=4.68534
18:33:57 Solution  Max Error=4.68534
18:33:57 Optimization Completed
18:33:57 Elapsed Real Time: 00:00:37   CPU Time 00:00:37
18:36:44 Save File Q8018_mm_1.ckt
```

From the above, we can see that the solution was reached in 24 iterations and the minimax error was reduced from 8.08001 to 4.68534.

Here is the circuit file for Part 1 after minimax optimization.

Circuit file after minimax optimization

```
!  File name: Q8018_mm_1.ckt
!  Circuit file for solving Question 8018, Part 1
!  After minimax optimization

Control
        optimizer = minimax;
End

Model
        ! Parameters of transmission line
        Zo = 1oh;                        ! char. imp. of input
        Zload = 10oh;                    ! load impedance

        Z1 = ? 2.5239 ?;                 ! char. imp. of section 1
        Z2 = ? 5.43473 ?;                ! char. imp. of section 2

        e1 = ? 0.37694 ?;                ! tolerance of Z1
        e2 = ? 0.49203 ?;                ! tolerance of Z2

        ! for report block only
        ep1 = e1*100/Z1;
        ep2 = e2*100/Z2;
```

```
! Cost function to minimize.
cost = 1/e1 + 1/e2;              ! cost for Part 1
!cost = (Z1/e1) + (Z2/e2);       ! cost for Part 2

.              ! the circuit file remains unchanged from here on
.
.
```

See Fig. SQ8018.1 for the reflection coefficient (MS11) results graphically over the range of 0.5-1.5 GHz. Note that four curves can be seen. They correspond to the four worst-case vertex reflection coefficient results.

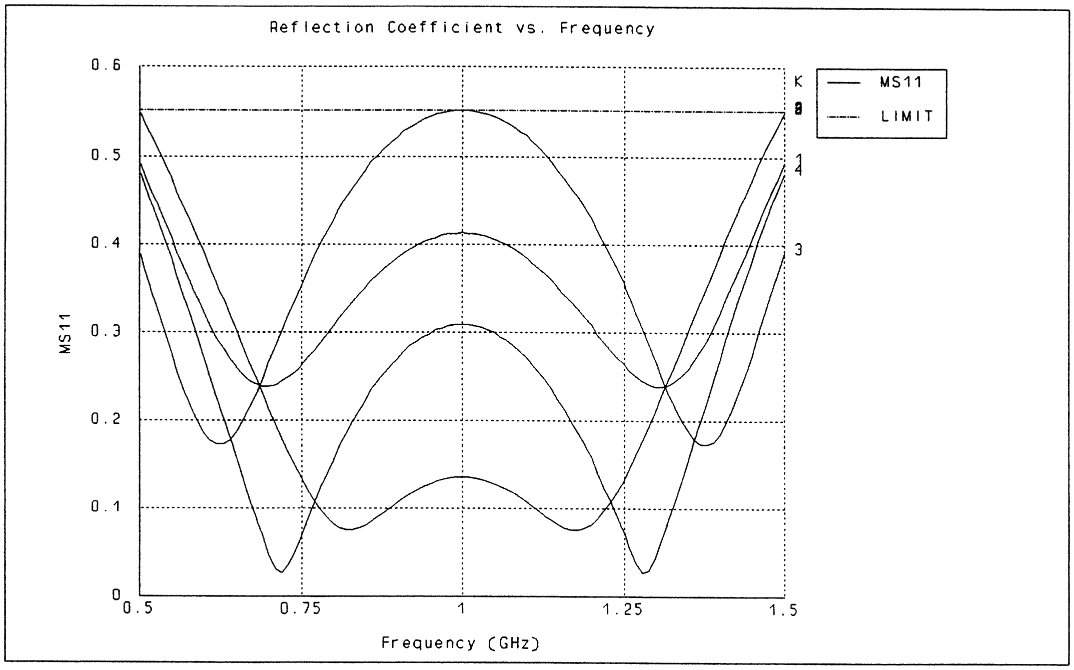

Fig. SQ8018.1 Reflection coefficient modulus versus frequency.

The report for Part 1 of this problem follows.

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before Optimization	After Optimization
Z1	2.23836	2.52390
Z2	4.47672	5.43473
e1	10 %	14.93482 %
e2	10 %	9.05344 %

Part 2 Results

Here, the original circuit file was altered so that the cost and alpha for Part 2 were put in place. Using minimax optimization, and recording only the best iterations the following results (as recorded in the osa90_0.log file) were obtained.

Iterations of minimax optimization

```
12:01:00 Read in File Q8018_0.ckt
12:01:28 Parsing Input File...
12:01:28 File Parsing Completed
12:01:31 Minimax Optimization
12:01:31 Optimization...  Press any key to interrupt
12:01:32 Iteration   1/999 Max Error=24.136
12:01:32 Iteration   2/999 Max Error=22.6973
12:01:33 Iteration   3/999 Max Error=20.2116
12:01:33 Iteration   4/999 Max Error=19.4438
12:01:34 Iteration   5/999 Max Error=18.0216
12:01:34 Iteration   6/999 Max Error=16.1582
12:01:35 Iteration   8/999 Max Error=15.8532
12:01:36 Iteration   9/999 Max Error=15.7768
12:01:37 Iteration  11/999 Max Error=15.7418
12:01:38 Iteration  13/999 Max Error=15.7411
12:01:38 Iteration  14/999 Max Error=15.7406
12:01:39 Iteration  15/999 Max Error=15.7406
12:01:39 Iteration  16/999 Max Error=15.7405
12:01:40 Iteration  17/999 Max Error=15.7405
12:01:41 Iteration  19/999 Max Error=15.7404
12:01:41 Iteration  20/999 Max Error=15.7404
12:01:42 Iteration  21/999 Max Error=15.7404
12:01:42 Solution  Max Error=15.7404
12:01:42 Optimization Completed
12:01:42 Elapsed Real Time: 00:00:11   CPU Time 00:00:10
12:03:08 Save File Q8018_mm_2.ckt
```

From the above, we can see that the solution was reached in 21 iterations and the minimax error was reduced from 24.136 to 15.7404.

Here is the circuit file for Part 2 after minimax optimization.

Circuit file after minimax optimization

```
!  File name: Q8018_mm_2.ckt
!  Circuit file for solving Question 8018, Part 2
!  After minimax optimization

Control
        optimizer = minimax;
End

Model
        ! Parameters of transmission line
        Zo = 1oh;                       ! char. imp. of input
        Zload = 10oh;                   ! load impedance

        Z1 = ? 2.14948 ?;               ! char. imp. of section 1
        Z2 = ? 4.72864 ?;               ! char. imp. of section 2

        e1 = ? 0.273115 ?;              ! tolerance of Z1
        e2 = ? 0.600826 ?;              ! tolerance of Z2

        ! for report block only
        ep1 = e1*100/Z1;
        ep2 = e2*100/Z2;
```

```
! Cost function to minimize.
!cost = 1/e1 + 1/e2;              ! cost for Part 1
cost = (Z1/e1) + (Z2/e2);         ! cost for Part 2

.                ! the circuit file remains unchanged from here on
.
.
```

See Fig. SQ8018.2 for the reflection coefficient (MS11) results graphically over the range of 0.5-1.5 GHz. Note that four curves can be seen which corresponds to the four worst-case vertex reflection coefficient results.

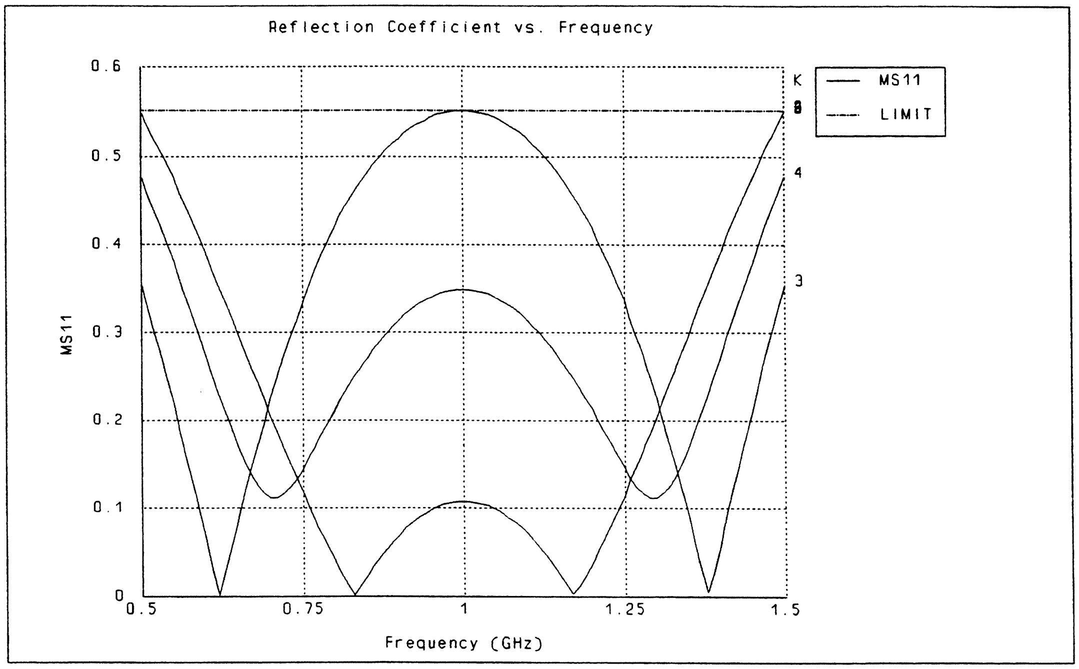

Fig. SQ8018.2 Reflection coefficient modulus versus frequency.

The report for Part 2 of this problem follows.

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before Optimization	After Optimization
Z1	2.23836	2.14948
Z2	4.47672	4.72864
e1	10 %	12.70610 %
e2	10 %	12.70611 %

OSA☞ Question 8027 Use OSA90/hope to perform a nominal point optimization of the five section lengths using the same constraints, characteristic impedances and starting values as Q8026.
(See Question 8026.)

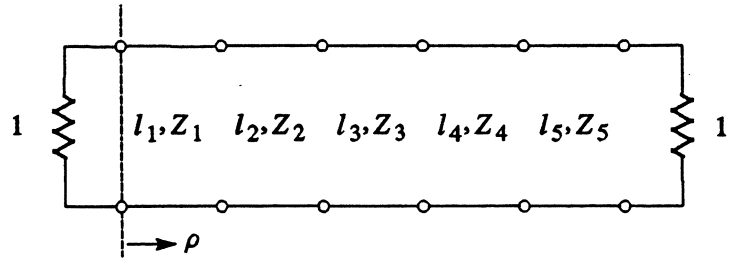

Fig. Q8026 Five-section transmission-line filter.

Solution

In this question, the normalized section lengths, l_{1n}, l_{2n}, l_{3n}, l_{4n} and l_{5n} are the optimizable variables with symmetry enforced. We are trying to achieve a passband insertion loss of no more than 0.01 dB over the band 0-1 GHz and maximize the loss outside this region in two different ways according to Part 1 and Part 2. The following circuit file was implemented to optimize these variables such that the insertion loss constraints given in Question 8026 were satisfied. The starting values, $l_{1n} = l_{5n} = 0.07$, $l_{2n} = l_{4n} = 0.15$ and $l_{3n} = 0.15$ were given. Note the difference between Part 1 and Part 2 in the specification block, everything else being the same.

The circuit file before optimization is below.

Circuit file before optimization

```
!  File name: Q8027_0.ckt
!  Circuit file for solving Question 8027
!  Before optimization

Control
        optimizer = minimax;
End

Model
        Lq = 7.49481cm;                 ! quarter wavelength at 1GHz

        ! Parameters
        Z0 = 1oh;                       ! char. imp. of input
        Zload = 1oh;                    ! load impedance

        Z1 : 0.2oh;                     ! char. imp. of section 1
        Z2 : 5oh;                       ! char. imp. of section 2
        Z3 : 0.2oh;                     ! char. imp. of section 1
        Z4 : 5oh;                       ! char. imp. of section 2
        Z5 : 0.2oh;                     ! char. imp. of section 2

        ! Raqtios of line sections vs. wavelength
        L1_Lq:? 0.07 ?;
        L2_Lq:? 0.15 ?;
        L3_Lq:? 0.15 ?;

        ! Define transmission line parameters
        PORT 1 0 NAME=input R=Z0;
        TRL 1 2 Z=Z1 L=(Lq*L1_Lq);
        TRL 2 3 Z=Z2 L=(Lq*L2_Lq);
        TRL 3 4 Z=Z3 L=(Lq*L3_Lq);
        TRL 4 5 Z=Z4 L=(Lq*L2_Lq);
```

```
        TRL 5 6 Z=Z5 L=(Lq*L1_Lq);
        RES 6 0 R=Zload;
        circuit;

        upper_spec = 0.01;

        ! Define the insertion loss using scattering parameters
        INSL = 10*LOG10(1/(1-MS11^2));
End

Sweep
        ac: freq: from 0 to 1 n=75 from 1 to 10 n=75 RREF=1  INSL upper_spec
                {       Xsweep  X=FREQ
                        Y=INSL.green
                        title="Insertion Loss vs. Frequency  -  0 to 10 GHz"
                        x_title="Frequency in GHz"
                        y_title="Insertion Loss  (dB)"
                        NXTICKS=10
                        xmin=0 xmax=10
                        ymin=0   }
                {       Xsweep  X=FREQ
                        Y=INSL.green & upper_spec.yellow.point
                        title="Insertion Loss vs. Frequency  -  0 to 1 GHz"
                        x_title="Frequency in GHz"
                        y_title="Insertion Loss  (dB)"
                        NXTICKS=10  NYTICKS=11
                        xmin=0 xmax=1
                        ymin=0 ymax=0.011        };
End

Specification
        !ac: freq: from 0 to 1 n=50  RREF=1  INSL<0.01;           ! Part 1
        !ac: freq: 5 RREF=1  MS11=1;                               ! Part 1
        ac: freq: from 0 to 1 n=50  RREF=1  INSL<0.01  w=10;       ! Part 2
        ac: freq: from 2.5 to 10 n=15 RREF=1  MS11=1;              ! Part 2
End

Report
          Optimizable Variables Before and After Optimization

    -----------------------------------------------------------------
        Variable         Before Optimization      After Optimization
    -----------------------------------------------------------------
           L1n                 0.07                 $% 6.5f$ $L1_Lq$
           L2n                 0.15                 $% 6.5f$ $L2_Lq$
           L3n                 0.15                 $% 6.5f$ $L3_Lq$
           L4n                 0.15                 $% 6.5f$ $L2_Lq$
           L5n                 0.07                 $% 6.5f$ $L1_Lq$
    -----------------------------------------------------------------
End
```

Part 1 Results

Here we are trying to maximize the insertion loss at 5 GHz outside the passband region. Using minimax optimization, and recording only the best iterations, the following results (as recorded in the osa90_0.log file) were obtained.

Iterations of minimax optimization

```
12:16:23 Read in File Q8027_0.ckt
12:16:27 Parsing Input File...
12:16:27 File Parsing Completed
12:16:30 Minimax Optimization
12:16:30 Optimization...  Press any key to interrupt
12:16:31 Iteration   1/999 Max Error=0.0351963
12:16:31 Iteration   2/999 Max Error=0.0285303
12:16:31 Iteration   3/999 Max Error=0.0169397
12:16:31 Iteration   4/999 Max Error=0.00250979
```

```
12:16:32 Iteration   5/999 Max Error=0.000141019
12:16:32 Iteration   7/999 Max Error=2.81497e-05
12:16:32 Iteration   8/999 Max Error=2.77183e-05
12:16:32 Iteration   9/999 Max Error=4.44181e-06
12:16:33 Iteration  11/999 Max Error=2.34747e-06
12:16:33 Iteration  12/999 Max Error=2.33064e-06
12:16:33 Iteration  13/999 Max Error=2.16626e-06
12:16:34 Iteration  14/999 Max Error=2.16158e-06
12:16:35 Iteration  21/999 Max Error=2.01227e-06
12:16:38 Iteration  28/999 Max Error=1.87527e-06
12:16:39 Iteration  35/999 Max Error=1.74943e-06
12:16:41 Iteration  42/999 Max Error=1.63375e-06
12:16:43 Iteration  49/999 Max Error=1.52733e-06
12:16:44 Iteration  53/999 Max Error=1.52009e-06
12:16:44 Iteration  54/999 Max Error=1.51862e-06
12:16:44 Iteration  55/999 Max Error=1.50961e-06
12:16:44 Iteration  56/999 Max Error=1.50925e-06
12:16:46 Iteration  63/999 Max Error=1.45599e-06
12:16:48 Iteration  70/999 Max Error=1.3729e-06
12:16:49 Iteration  76/999 Max Error=1.29669e-06
12:16:49 Iteration  77/999 Max Error=1.29471e-06
12:16:51 Iteration  84/999 Max Error=1.20767e-06
12:16:53 Iteration  91/999 Max Error=1.18109e-06
12:16:53 Iteration  92/999 Max Error=1.17932e-06
12:16:55 Iteration  99/999 Max Error=1.09596e-06
12:16:55 Iteration 101/999 Max Error=1.07413e-06
12:16:55 Iteration 102/999 Max Error=1.06776e-06
12:16:55 Solution  Max Error=1.06776e-06
12:16:55 Optimization Completed
12:16:55 Elapsed Real Time: 00:00:25   CPU Time 00:00:24
12:21:17 Save File Q8027_mm_1.ckt
```

From the above, we can see that the solution was reached in 102 iterations and minimax error was reduced from 0.351963 to 1.06776×10^{-06}.

Here is the circuit file for Part 1 after minimax optimization.

Circuit file after minimax optimization

```
!  File name: Q8027_mm_1.ckt
!  Circuit file for solving Question 8027, Part 1
!  After minimax optimization

Control
        optimizer = minimax;
End

Model
        Lq = 7.49481cm;          ! quarter wavelength at 1GHz

        ! Parameters
        Z0 = 1oh;                        ! char. imp. of input
        Zload = 1oh;             ! load impedance

        Z1 : 0.2oh;                      ! char. imp. of section 1
        Z2 : 5oh;                        ! char. imp. of section 2
        Z3 : 0.2oh;                      ! char. imp. of section 1
        Z4 : 5oh;                        ! char. imp. of section 2
        Z5 : 0.2oh;                      ! char. imp. of section 2

        ! Ratios of line sections vs. wavelength
        L1_Lq:? 0.0960079 ?;
        L2_Lq:? 0.162896 ?;
        L3_Lq:? 0.198132 ?;

        .               ! the circuit file remains unchanged from here on
        .
        .
```

See Fig. SQ8027.1 and SQ8027.2 for the insertion loss results graphically. The first graph shows the loss over the whole frequency range, 0-10 GHz, and the second graph shows the lower 0-1 GHz region in greater detail. From these graphs, we can see that the passband has been established but the stopband appears to be a maximum around 6.5 GHz and not the intended 5 GHz.

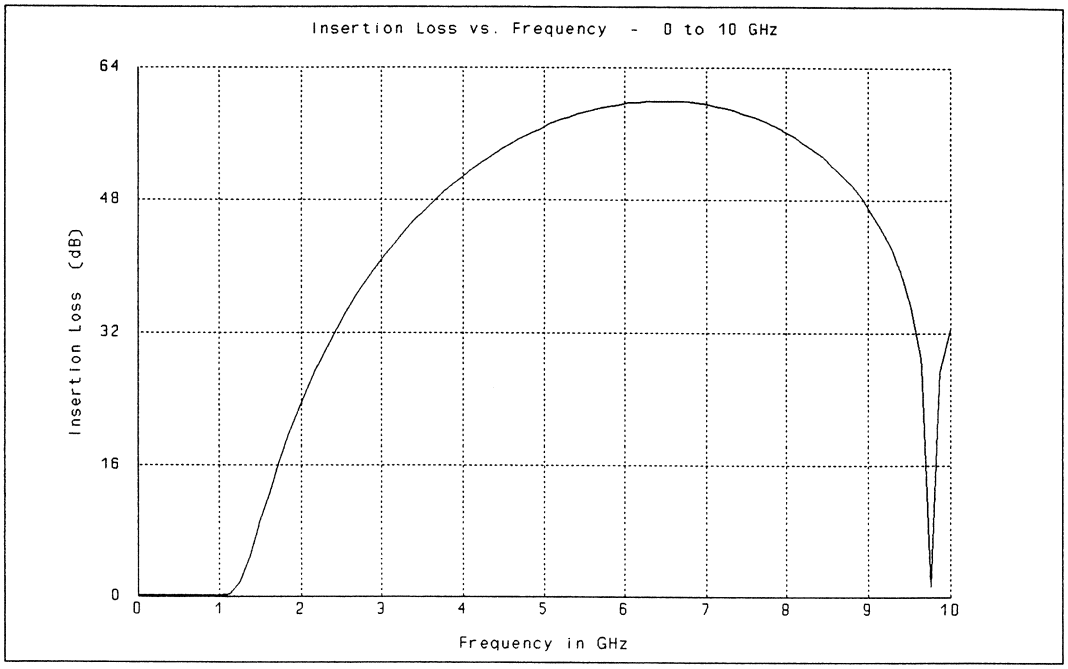

Fig. SQ8027.1 Insertion loss versus frequency.

The report for Part 1 follows.

Report for Part 1 after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before Optimization	After Optimization
L1n	0.07	0.09601
L2n	0.15	0.16290
L3n	0.15	0.19813
L4n	0.15	0.16290
L5n	0.07	0.09601

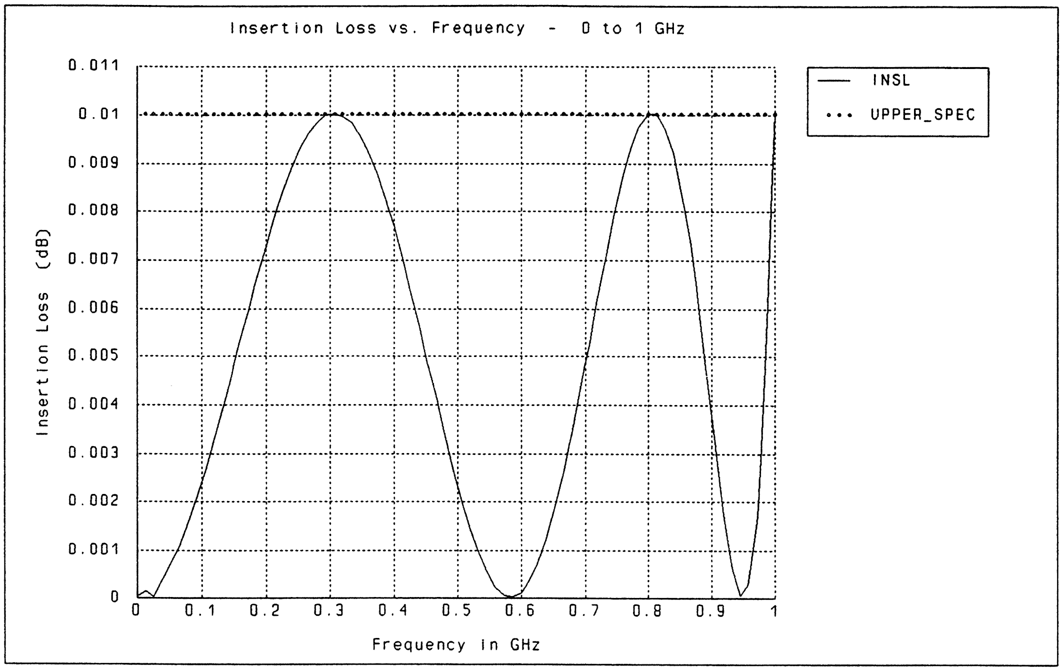

Fig. SQ8027.2 Insertion loss versus frequency (passband detail).

Part 2 Results

Here we are trying to maximize the minimum insertion loss over the range 2.5-10 GHz outside the passband region. The appropriate changes in the circuit file specifications were made before proceeding. Using minimax optimization, and recording only the best iterations, the following results (as recorded in the osa90_0.log file) were obtained.

Iterations of minimax optimization

```
12:34:28 Read in File Q8027_0.ckt
12:34:31 Parsing Input File...
12:34:31 File Parsing Completed
12:34:38 Minimax Optimization
12:34:38 Optimization... Press any key to interrupt
12:34:38 Iteration   1/999 Max Error=0.351963
12:34:38 Iteration   2/999 Max Error=0.285303
12:34:39 Iteration   3/999 Max Error=0.169397
12:34:39 Iteration   4/999 Max Error=0.0240075
12:34:39 Iteration   5/999 Max Error=0.000994723
12:34:40 Iteration   8/999 Max Error=0.00097907
12:34:40 Iteration   9/999 Max Error=0.000885249
12:34:41 Iteration  11/999 Max Error=0.000860411
12:34:41 Iteration  12/999 Max Error=0.000836325
12:34:42 Iteration  14/999 Max Error=0.000824534
12:34:42 Iteration  15/999 Max Error=0.000812921
12:34:43 Iteration  17/999 Max Error=0.000807175
12:34:44 Iteration  19/999 Max Error=0.000800949
12:34:44 Iteration  20/999 Max Error=0.000795896
12:34:44 Iteration  21/999 Max Error=0.000787575
12:34:44 Iteration  22/999 Max Error=0.000779343
12:34:45 Iteration  23/999 Max Error=0.000771198
12:34:46 Iteration  27/999 Max Error=0.00069591
```

```
12:34:46 Iteration  29/999 Max Error=0.000688576
12:34:47 Iteration  30/999 Max Error=0.000685005
12:34:47 Iteration  31/999 Max Error=0.000681452
12:34:47 Iteration  32/999 Max Error=0.000677918
12:34:49 Iteration  39/999 Max Error=0.000538383
12:34:50 Iteration  41/999 Max Error=0.000530358
12:34:50 Iteration  42/999 Max Error=0.00052748
12:34:51 Iteration  43/999 Max Error=0.000524617
12:34:51 Iteration  44/999 Max Error=0.00052177
12:34:53 Iteration  51/999 Max Error=0.00044097
12:34:53 Iteration  53/999 Max Error=0.000418345
12:34:54 Iteration  54/999 Max Error=0.000414256
12:34:54 Iteration  55/999 Max Error=0.000410459
12:34:54 Iteration  56/999 Max Error=0.000406514
12:34:55 Iteration  57/999 Max Error=0.000402624
12:34:55 Iteration  58/999 Max Error=0.000398775
12:34:55 Iteration  59/999 Max Error=0.000394966
12:34:56 Iteration  63/999 Max Error=0.000378762
12:34:57 Iteration  64/999 Max Error=0.000352725
12:34:57 Iteration  66/999 Max Error=0.000350668
12:34:58 Iteration  67/999 Max Error=0.000349653
12:34:58 Iteration  68/999 Max Error=0.00034868
12:34:58 Iteration  69/999 Max Error=0.00034776
12:34:58 Iteration  70/999 Max Error=0.000346889
12:34:59 Iteration  71/999 Max Error=0.000346873
12:34:59 Iteration  72/999 Max Error=0.000346663
12:34:59 Iteration  73/999 Max Error=0.000346423
12:35:00 Iteration  74/999 Max Error=0.000346236
12:35:00 Iteration  75/999 Max Error=0.000346098
12:35:00 Iteration  76/999 Max Error=0.000345665
12:35:01 Iteration  78/999 Max Error=0.00034559
12:35:01 Iteration  79/999 Max Error=0.000345588
12:35:01 Solution  Max Error=0.000345588
12:35:01 Optimization Completed
12:35:01 Elapsed Real Time: 00:00:23   CPU Time 00:00:23
12:37:12 Save File Q8027_mm_2.ckt
```

From the above, we can see that the solution was reached in 79 iterations and the minimax error was reduced from 0.351963 to 0.000345588.

Here is the circuit file for Part 2 after minimax optimization.

Circuit file after minimax optimization

```
! File name: Q8027_mm_2.ckt
! Circuit file for solving Question 8027, Part 2
! After minimax optimization

Control
        optimizer = minimax;
End

Model
        Lq = 7.49481cm;          ! quarter wavelength at 1GHz

        ! Parameters
        Z0 = 1oh;                        ! char. imp. of input
        Zload = 1oh;             ! load impedance

        Z1 : 0.2oh;                      ! char. imp. of section 1
        Z2 : 5oh;                        ! char. imp. of section 2
        Z3 : 0.2oh;                      ! char. imp. of section 1
        Z4 : 5oh;                        ! char. imp. of section 2
        Z5 : 0.2oh;                      ! char. imp. of section 2

        ! Ratios of line sections vs. wavelength
        L1_Lq:? 0.0913218 ?;
        L2_Lq:? 0.157621 ?;
        L3_Lq:? 0.189689 ?;
```

```
.            ! the circuit file remains unchanged from here on
.
.
```

See Fig. SQ8027.3 and SQ8027.4 for the insertion loss results graphically. The first graph shows the loss over the whole frequency range, 0-10 GHz, and the second graph shows the lower 0-1 GHz region in greater detail. From these graphs, we can see that the insertion loss goes slightly above 0.01 dB at 0.83 GHz in the passband, but the stopband arcs evenly about the 2.5-10 GHz range from 32 to 60 back to 32 dB.

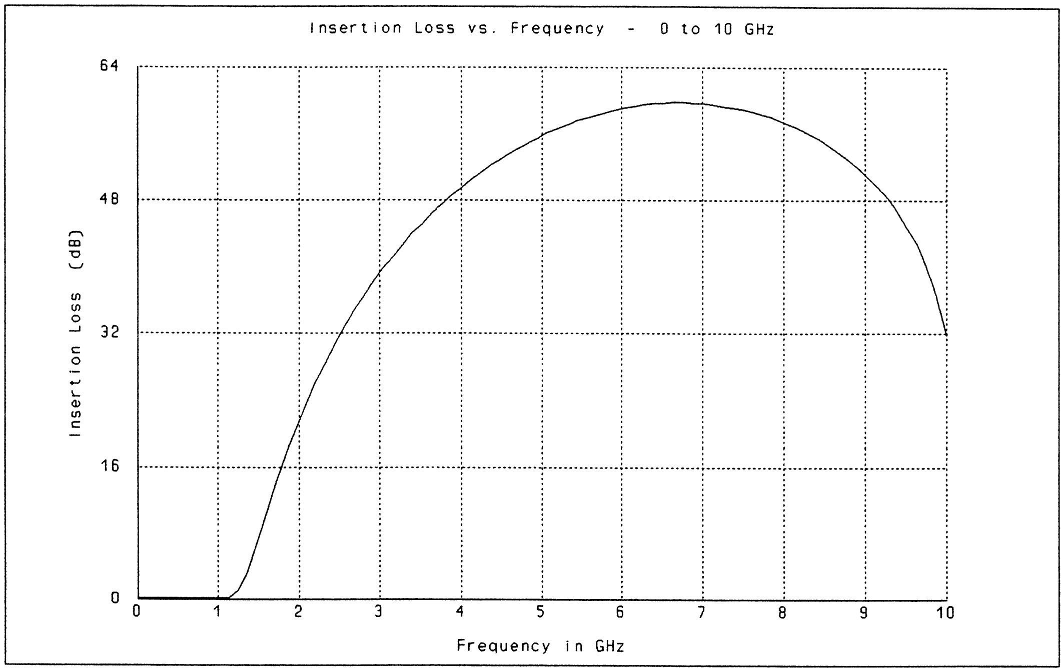

Fig. SQ8027.3 Insertion loss versus frequency.

The report for Part 2 follows.

Report for Part 2 after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before Optimization	After Optimization
L1n	0.07	0.09132
L2n	0.15	0.15762
L3n	0.15	0.18969
L4n	0.15	0.15762
L5n	0.07	0.09132

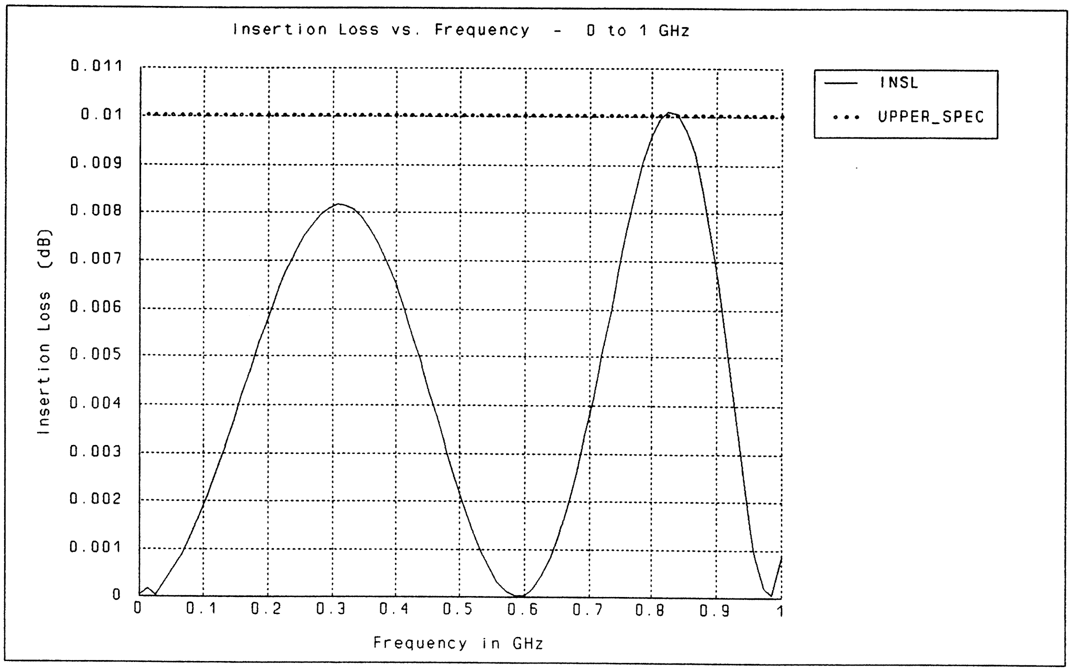

Fig. SQ8027.4 Insertion loss versus frequency (passband detail).

OSA☞ Question 8028 Use OSA90/hope to perform a worst-case tolerance optimization of the circuit shown in Fig. Q8026 to satisfy the following insertion loss constraints.

Frequency Range (GHz)	Insertion Loss (dB)
0 - 1	< 0.02
5	> 25

Keep the characteristic impedances fixed as in the previous problem. Use an exact-penalty function formulation to achieve this goal for a relative cost function. (See Question 8026.)

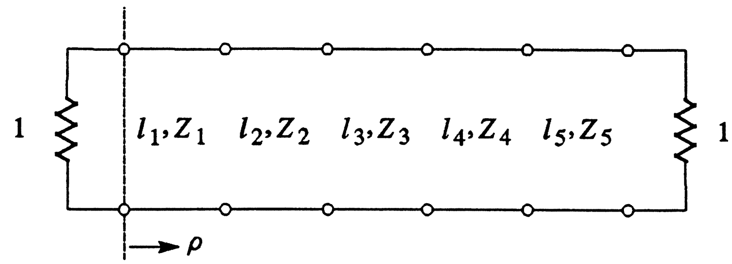

Fig. Q8026 Five–section transmission–line filter.

Solution

This question involves the same circuit as in Question 8027. The optimizable variables are the normalized symmetrical section lengths and their corresponding tolerances, ϵ_1, ϵ_2, and ϵ_2. The characteristic impedances are fixed at the same values as in Question 8027. The following circuit file was implemented to optimize these variables such that the insertion loss constraints were satisfied. U is the cost function. The exact penalty function, V, is composed of maximum of U, U-αg_1 and U-αg_2. It is minimized using minimax optimization. The results from Question 8027, Part 1, were chosen as the starting points for the circuit elements and the tolerances were given an initial 1%.

The circuit file before optimization is below.

Circuit file before optimization

```
!  File name: Q8028_0.ckt
!  Circuit file for solving Question 8028
!  Before optimization

Control
        optimizer = minimax;
End

Model
        Lq:7.49481cm;

        Z0:1oh;                 !  Input impedance
        Zload:1oh;              !  Load impedance

        Z1:.2;   Z3:.2;   Z5:.2;
        Z2:5;    Z4:5;
```

```
! Ratios of the line sections vs. wavelength:
        l1_lq:? 0.0960079 ?;
        l2_lq:? 0.162896 ?;
        l3_lq:? 0.198132 ?;

! Define the matrix of the vertices settings
        V[8,3] = [ -1 -1 -1
                    1 -1 -1
                   -1  1 -1
                    1  1 -1
                   -1 -1  1
                    1 -1  1
                   -1  1  1
                    1  1  1 ];
        I=1;

! Submatrix representing the I-th row of V (i.e. I-th set of vertices)
        Multiplier[3]=Row(V,I);

! Matrix of the tolerances and characteristic impedances vector:
        Eps[3,3]:[ ? 0.000960079 ?         0              0
                        0          ? 0.00162896 ?         0
                        0              0          ? 0.00198132 ? ];

! Define the sections lengths vector:
        L[3]=[ l1_lq  l2_lq  l3_lq ];

! Define the tolerance deviation from nominal value determined by I-th row of Matrix V
        Tolerance[3]=product(Multiplier, Eps);

! Define the vertex:
        Lvert[3]=L+Tolerance;

! Defining the transmission line:
        PORT 1 0 NAME=INPUT  R=Z0;
        TRL 1 2 Z=Z1 L=(Lvert[1]*Lq);
        TRL 2 3 Z=Z2 L=(Lvert[2]*Lq);
        TRL 3 4 Z=Z3 L=(Lvert[3]*Lq);
        TRL 4 5 Z=Z4 L=(Lvert[2]*Lq);
        TRL 5 6 Z=Z5 L=(Lvert[1]*Lq);
        RES 6 0 R=Zload;
        circuit ;

! Set the passband and stoppband insertion loss limits
        INSL_Pas = if ((freq<1)+(freq=1)) (0.02) else (-5);
        INSL_Stop = if ((freq=5)+(freq>5)) (25) else (-5);

! Define the cost function for problem a:
        cost:1/Eps[1,1] + 1/Eps[2,2] + 1/Eps[3,3];

! Define insertion loss
        INSL = 10*log10(1/(1-MS11^2));

! For report block only
        ep1 = Eps[1,1]*100/L[1];
        ep2 = Eps[2,2]*100/L[2];
        ep3 = Eps[3,3]*100/L[3];

! Exact penalty function definitions
        alpha = 15000;

        U = 0.001*cost;

        g1 = 0.02 - INSL;
        g2 = INSL - 25;

        Ua = U - (alpha*g1);
        Ub = U - (alpha*g2);
End
```

```
Sweep
        AC:I:from 1 to 8 step 1
          Freq: FROM 0 TO 1 n=75 from 1 to 10 n=75 Rref=1 INSL INSL_Pas INSL_Stop

          { Xsweep  X=FREQ I=all
             Y=INSL_Pas.yellow.point & INSL.green
             title="Problem 5; Set B;  Insertion Loss vs. Frequency  -  0 to 1 GHz"
             x_title="Frequency in GHz"
             y_title="Insertion Loss"
              NYTICKS=6 xmin=0.1 xmax=1 ymin=0 ymax=0.03}

          { Xsweep  X=FREQ I=all
             Y=Insl.green & INSL_Stop.yellow.point
             title="Problem 5; Set B;  Insertion Loss vs. Frequency  -  0 to 10 GHz"
             x_title="Frequency in GHz"
             y_title="Insertion Loss"
              xmin=0 xmax=10 ymin=0 };
End

Specification
   U;
   ac:I:from 1 to 8  step 1
      freq: from 0 to 1 n=25  Rref=1 Ua;
   ac:I:from 1 to 8  step 1
      freq: 5 Rref=1 Ub
End

Report
          Optimizable Variables Before and After Optimization

    -------------------------------------------------------------
        Variable        Before Optimization      After Optimization
    -------------------------------------------------------------
          L1n                0.09601                $% 6.5f$ $l1_lq$
          L2n                0.16290                $% 6.5f$ $l2_lq$
          L3n                0.19813                $% 6.5f$ $l3_lq$
          L4n                0.16290                $% 6.5f$ $l2_lq$
          L5n                0.09601                $% 6.5f$ $l1_lq$

          e1                   1 %                  $% 6.5f$ $ep1$ %
          e2                   1 %                  $% 6.5f$ $ep2$ %
          e3                   1 %                  $% 6.5f$ $ep3$ %
          e4                   1 %                  $% 6.5f$ $ep2$ %
          e5                   1 %                  $% 6.5f$ $ep1$ %
    -------------------------------------------------------------
End
```

Using minimax optimization, an accuracy of solution set to 10^{-3} and recording only the best iterations, the following results (as recorded in the osa90_0.log file) were obtained.

Iterations of minimax optimization

```
10:50:39 Read in File Q8028_0.ckt
10:50:45 Parsing Input File ...
10:50:45 File Parsing Completed
10:51:07 Minimax Optimization
10:51:07 Optimization ...  Press any key to interrupt
10:51:09 Iteration   1/999 Max Error=64.53
10:51:11 Iteration   2/999 Max Error=5.02933
10:51:13 Iteration   3/999 Max Error=2.14431
10:51:14 Iteration   4/999 Max Error=2.12986
10:51:16 Iteration   5/999 Max Error=2.06289

         .
         .
         .
```

```
        .
        .
        .

10:58:43 Iteration 221/999 Max Error=0.820971
10:58:48 Iteration 223/999 Max Error=0.816395
10:58:50 Iteration 224/999 Max Error=0.815211
10:58:52 Iteration 225/999 Max Error=0.814035
10:58:54 Iteration 226/999 Max Error=0.813955
10:58:54 Solution  Max Error=0.813955
10:58:54 Optimization Completed
10:58:54 Elapsed Real Time: 00:07:47   CPU Time 00:07:11
11:01:08 Save File Q8028_mm.ckt
```

From the above, we can see that the solution was reached (with an accuracy of solution set to 10^{-3}) in 226 iterations. The minimax error was reduced from 64.53 to 0.813955.

Here is the circuit file after minimax optimization.

Circuit file after minimax optimization

```
!  File name: Q8028_mm.ckt
!  Circuit file for solving Question 8028
!  After minimax optimization

Control
        optimizer = minimax;
End

Model
        Lq:7.49481cm;

        Z0:1oh;                 !  Input impedance
        Zload:1oh;              !  Load impedance

        Z1:.2;   Z3:.2;   Z5:.2;
        Z2:5;    Z4:5;

! Ratios of the line sections vs. wavelength:
        l1_lq:? 0.0865775 ?;
        l2_lq:? 0.150828 ?;
        l3_lq:? 0.185632 ?;

! Define the matrix of the vertices settings
        V[8,3] = [ -1 -1 -1
                    1 -1 -1
                   -1  1 -1
                    1  1 -1
                   -1 -1  1
                    1 -1  1
                   -1  1  1
                    1  1  1 ];
        I=1;

! Submatrix representing the I-th row of V (i.e. I-th set of vertices)
        Multiplier[3]=Row(V,I);

! Matrix of the tolerances and characteristic impedances vector:
        Eps[3,3]:[ ? 0.00283667 ?         0              0
                   0          ? 0.00408008 ?            0
                   0              0           ? 0.00464894 ? ];

        .               ! the circuit file remains unchanged from here on
        .
        .
```

See Fig. SQ8028.1 and SQ8028.2 for the insertion loss results graphically. The first graph shows the loss over the whole frequency range, 0-10 GHz, and the second graph shows the lower passband region from 0.1-1 GHz in greater detail. Note that eight curves can be seen on both graphs. They correspond to the eight worst-case vertex insertion loss responses. Looking at the passband, it seems that 1 vertex has only slightly exceeded the upper insertion loss specification of 0.02 dB.

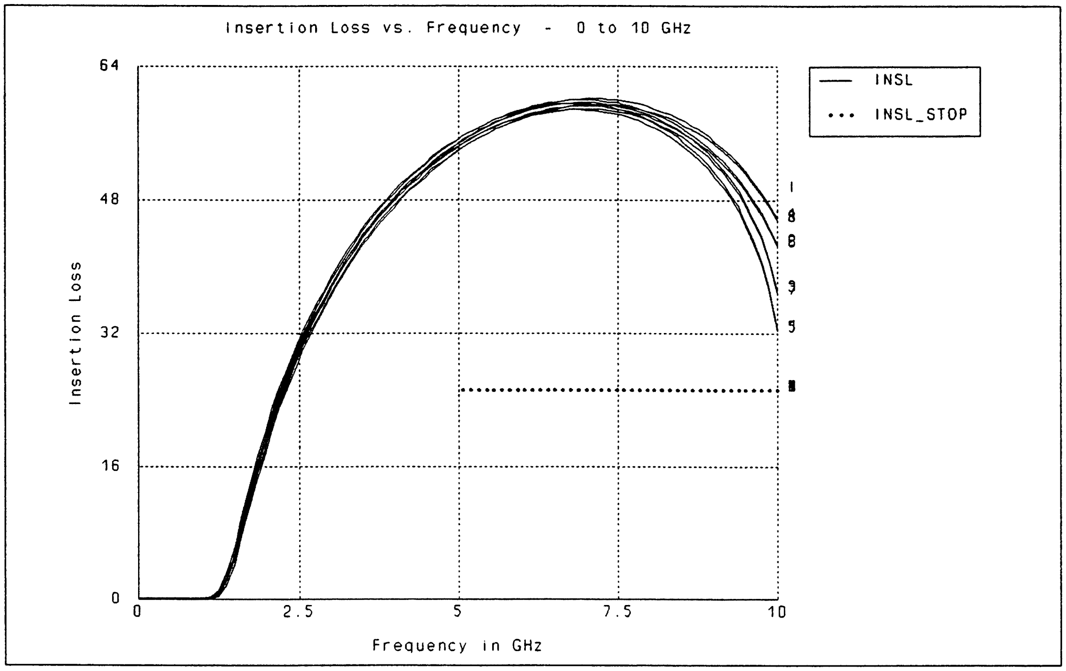

Fig. SQ8028.1 Insertion loss versus frequency.

The report for this problem follows.

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before Optimization	After Optimization
L1n	0.09601	0.08658
L2n	0.16290	0.15083
L3n	0.19813	0.18563
L4n	0.16290	0.15083
L5n	0.09601	0.08658
e1	1 %	3.27645 %
e2	1 %	2.70512 %
e3	1 %	2.50439 %
e4	1 %	2.70512 %
e5	1 %	3.27645 %

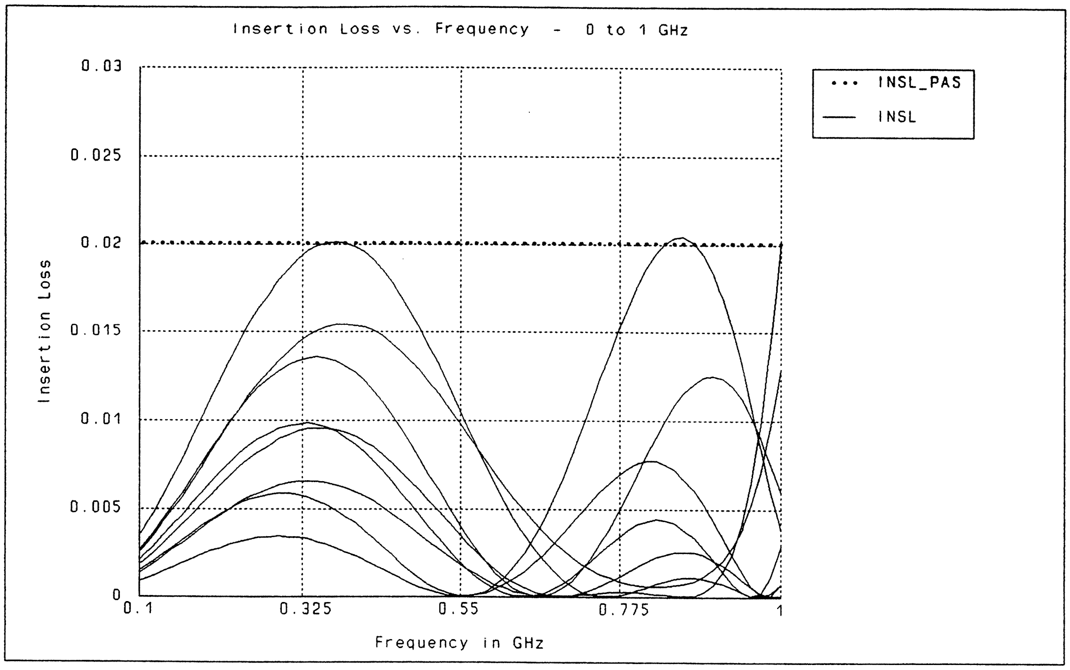

Fig. SQ8028.2 Insertion loss versus frequency (passband detail).

OSA☛ Question 8034 Use OSA90/hope to perform a nominal point optimization of the seven characteristic impedances in the circuit shown in Fig. Q8033. Satisfy the same insertion loss design constraints as in Q8033. (See Question 8033.)

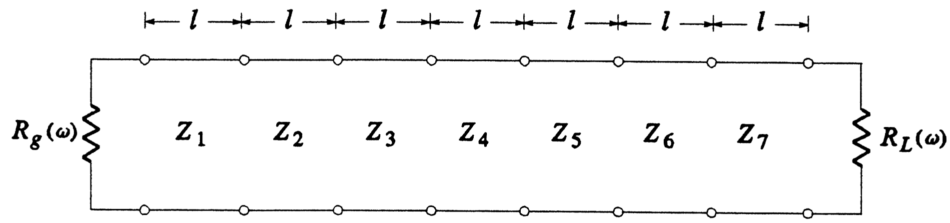

Fig. Q8033 Seven-section, cascaded transmission-line filter.

Solution

In this question, the characteristic impedances, Z_1 to Z_7, are the optimizable variables. Their corresponding section lengths, l_1 to l_7, are fixed at 1.5 cm each. The following circuit file was implemented to optimize the impedances such that the insertion loss is within 0.4 dB between 2.16 and 3 GHz, and maximized at 5 GHz. The starting point is given as

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \\ Z_7 \end{bmatrix} = \begin{bmatrix} 3000 \\ 3000 \\ 25000 \\ 10000 \\ 25000 \\ 3000 \\ 3000 \end{bmatrix}$$

The circuit file before optimization is below.

Circuit file before optimization

```
! File name: Q8034_0.ckt
! Circuit file for solving Question 8034
! Before optimization

Control
        optimizer = minimax;
End

Model
        Length = 1.5cm;                         ! length of each section

        ! Characteristic impedances of each section
        Z1:? 3000 ?;                    ! char. imp. of section 1
        Z2:? 3000 ?;                    ! char. imp. of section 2
        Z3:? 25000 ?;                   ! char. imp. of section 3
        Z4:? 10000 ?;                   ! char. imp. of section 3
        Z5:? 25000 ?;                   ! char. imp. of section 5
        Z6:? 3000 ?;                    ! char. imp. of section 6
        Z7:? 3000 ?;                    ! char. imp. of section 7
```

```
        fc = 2.077;

        Z0 = 377 / sqrt(1-(fc/FREQ)^2);

        ! Define the transmission line
        PORT 1 0 NAME=input R=Z0;
        TRL 1 2 Z=Z1 L=Length;
        TRL 2 3 Z=Z2 L=Length;
        TRL 3 4 Z=Z3 L=Length;
        TRL 4 5 Z=Z4 L=Length;
        TRL 5 6 Z=Z5 L=Length;
        TRL 6 7 Z=Z6 L=Length;
        TRL 7 8 Z=Z7 L=Length;
        PORT 8 0 NAME=output R=Z0;
        circuit;

        ! Define insertion loss - already defined by using a 2nd port
        ! INSL = 10*LOG10(1/(1-MS11^2));

        limit = if (((freq>2.16)+(freq=2.16))*((freq=3)+(freq<3))) (0.4) else (-2)
End

Sweep
        ac: freq: from 2.16 to 3 n=75  from 3 to 5 n=50 RREF=1  INSL limit
                {       Xsweep  X=FREQ
                        Y=INSL.green & limit.yellow.point
                        title="Insertion Loss vs. Frequency  -  2.16 to 3 GHz"
                        x_title="Frequency in GHz"
                        y_title="Insertion Loss  (dB)"
                        Ymin=0 Ymax=0.5 Xmax=3
                        NXTICKS=5        }

                {       Xsweep  X=FREQ
                        Y=INSL.green & limit.yellow.point
                        title="Insertion Loss vs. Frequency  -  2.16 to 5 GHz"
                        x_title="Frequency in GHz"
                        y_title="Insertion Loss  (dB)"
                        NXTICKS=5
                        Xmin=2.16 Xmax=5 Ymin=0  };
End

Specification
        ac: freq: from 2.16 to 3 n=30 RREF=1  INSL<0.4 W=200;
        ac: freq: 5 RREF=1  MS11=1;
End

Report
         Optimizable Variables Before and After Optimization

    ----------------------------------------------------------------
        Variable         Before Optimization       After Optimization
    ----------------------------------------------------------------
           Z1                  3000                $% 6.5f$ $Z1$
           Z2                  3000                $% 6.5f$ $Z2$
           Z3                  25000               $% 6.5f$ $Z3$
           Z4                  10000               $% 6.5f$ $Z4$
           Z5                  25000               $% 6.5f$ $Z5$
           Z6                  3000                $% 6.5f$ $Z6$
           Z7                  3000                $% 6.5f$ $Z7$
    ----------------------------------------------------------------
End
```

Using minimax optimization, and recording only the best iterations, the following results (as recorded in the osa90_0.log file) were obtained.

Iterations of minimax optimization

```
16:30:04 Read in File Q8034_0.ckt
16:30:05 Parsing Input File...
16:30:05 File Parsing Completed
16:30:10 Minimax Optimization
16:30:10 Optimization...  Press any key to interrupt
16:30:10 Iteration   1/999 Max Error=114.12
16:30:10 Iteration   2/999 Max Error=97.4003
16:30:11 Iteration   3/999 Max Error=66.1057
16:30:11 Iteration   4/999 Max Error=15.1805
16:30:12 Iteration   6/999 Max Error=6.99652
16:30:12 Iteration   7/999 Max Error=1.26984
16:30:14 Iteration  10/999 Max Error=0.235732
16:30:14 Iteration  11/999 Max Error=3.8772e-10
16:30:15 Iteration  13/999 Max Error=3.85165e-10
16:30:15 Iteration  14/999 Max Error=3.82625e-10
16:30:16 Iteration  16/999 Max Error=3.81363e-10
16:30:17 Iteration  18/999 Max Error=3.81048e-10
16:30:20 Iteration  25/999 Max Error=3.80689e-10
16:30:20 Solution  Max Error=3.80689e-10
16:30:20 Optimization Completed
16:30:20 Elapsed Real Time: 00:00:10   CPU Time 00:00:08
16:30:36 Save File Q8034_mm.ckt
```

From the above, we can see that the solution was reached in 25 iterations and the minimax error was reduced from 114.12 to 3.80689×10^{-10}.

Here is the circuit file after minimax optimization.

Circuit file after minimax optimization

```
! File name: Q8034_mm.ckt
! Circuit file for solving Question 8034
! After optimization

Control
        optimizer = minimax;
End

Model
        Length = 1.5cm;                         ! length of each section

        ! Characteristic impedances of each section
        Z1:? 3043.66 ?;                 ! char. imp. of section 1
        Z2:? 2811.72 ?;                 ! char. imp. of section 2
        Z3:? 25166 ?;                   ! char. imp. of section 3
        Z4:? 10238.7 ?;                 ! char. imp. of section 3
        Z5:? 25166 ?;                   ! char. imp. of section 5
        Z6:? 2811.72 ?;                 ! char. imp. of section 6
        Z7:? 3043.66 ?;                 ! char. imp. of section 7

        .               ! the circuit file remains unchanged from here on
        .
        .
```

See Fig. SQ8034.1 and Fig. SQ8034.2 for the insertion loss graphically. The first graph shows the results over the range of 2.16-5 GHz and the second over the range of 2.16-3 GHz. From these graphs, we can see that the passband has been established and the stopband insertion loss maximizes at 5 GHz.

The report for this problem follows.

Report after minimax optimization

Optimizable Variables Before and After Optimization

Variable	Before Optimization	After Optimization
Z1	3000	3043.66
Z2	3000	2811.72
Z3	25000	25166.00
Z4	10000	10238.70
Z5	25000	25166.00
Z6	3000	2811.72
Z7	3000	3043.66

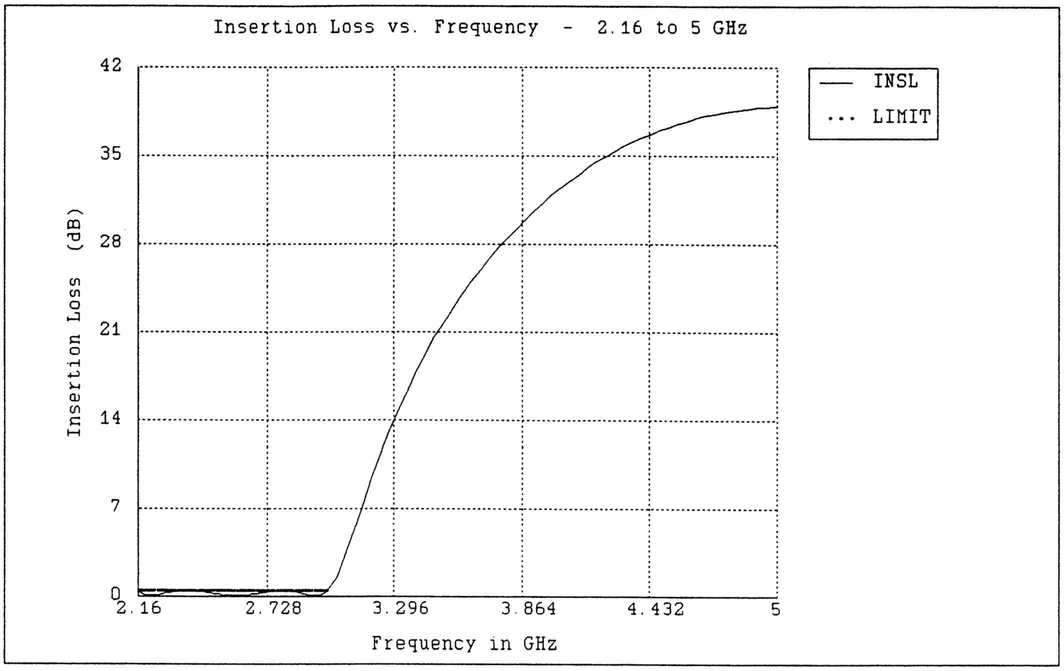

Fig. SQ8034.1 Insertion loss versus frequency.

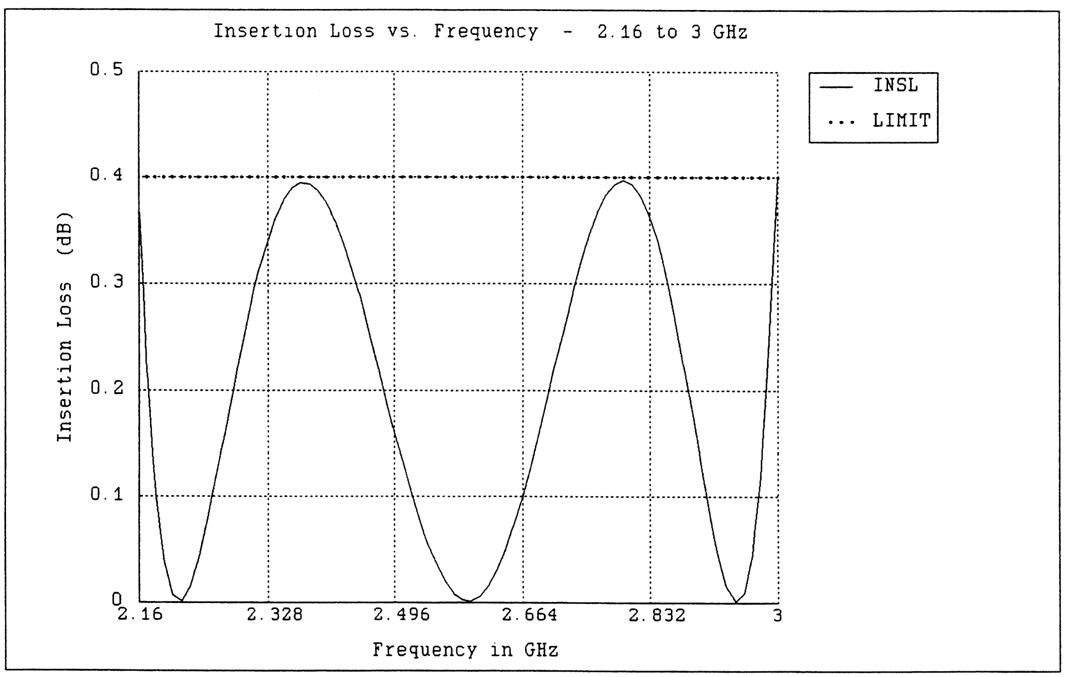

Fig. SQ8034.2 Insertion loss versus frequency (passband detail).

4.2.9 Various

Chapter 5

LABORATORIES

Lab 1 LC Transformer Simulation Using C

Objectives

This lab has been devised for students to learn how to write C programs to simulate simple circuits.

Equipment

Sun SPARCstations (Room JHE 215).

Circuit Diagram and Parameter Values

Consider the LC transformer circuit shown in Fig. L1. Write a C program to compute the branch currents, branch voltages and magnitude of the input reflection coefficient of the LC circuit. The parameter values are $I_s = 1$ A, $R_1 = 3\ \Omega$, $R_2 = 1\ \Omega$, $L_1 = 1.041$ H, $C_2 = 0.979$ F, $L_3 = 2.341$ H, $C_4 = 0.781$ F, $L_5 = 2.937$ H, $C_6 = 0.347$ F. Use 21 uniformly spaced points in the frequency range 0.5 – 1.179 rad/s, i.e.,

$$\omega = 0.5 + k * (1.179 - 0.5) / 20, \;\; k = 0, 1, ..., 20$$

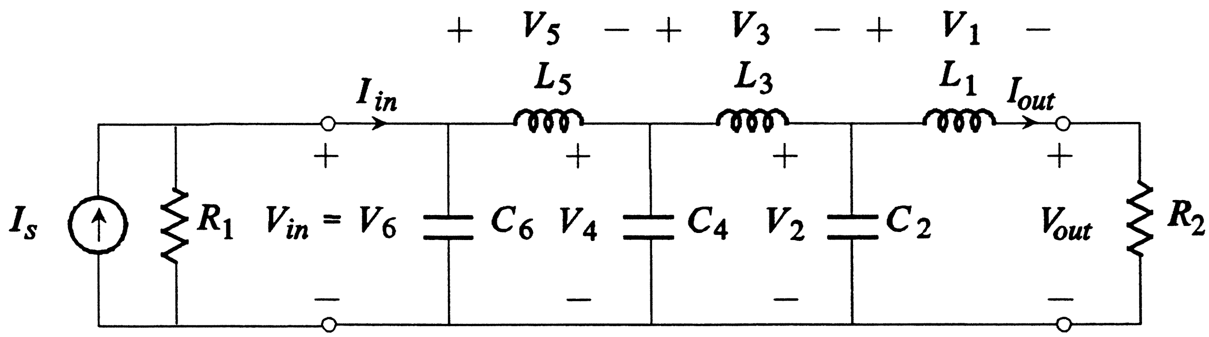

Fig. L1 A lumped element LC transformer.

Procedure

Computation of Branch Currents and Voltages

Assume

$$I_{out}^0 = 1 \text{ A}$$

where the superscript ^{0} denotes assumed values. Then

$$V_{out}^0 = I_{out}^0 * 1\ \Omega$$

$$I_1^0 = I_{out}^0$$

$$V_1^0 = I_1^0 * j\omega L_1$$

$$V_2^0 = V_1^0 + V_{out}^0$$

$$I_2^0 = V_2^0 * j\omega C_2$$

$$I_3^0 = I_2^0 + I_1^0$$

$$V_3^0 = I_3^0 * j\omega L_3$$

$$V_4^0 = V_3^0 + V_2^0$$

$$I_4^0 = V_4^0 * j\omega C_4$$

$$I_5^0 = I_4^0 + I_3^0$$

$$V_5^0 = I_5^0 * j\omega L_5$$

$$V_6^0 = V_5^0 + V_4^0$$

$$I_6^0 = V_6^0 * j\omega C_6$$

$$I_{in}^0 = I_6^0 + I_5^0$$

$$V_{in}^0 = V_6^0$$

$$I_s^0 = I_{in}^0 + V_{in}^0 / 3\ \Omega$$

In the foregoing analysis, I_s^0 is the assumed source current. Since the actual source current is 1 A, the actual branch currents and voltages are obtained by scaling. They are given by

$$I_i = I_i^0 / I_s^0, \quad i = 1, 2, ..., 6$$

$$V_i = V_i^0 / I_s^0, \quad i = 1, 2, ..., 6$$

The input impedance is

$$Z_{in} = V_{in}^0 \ / \ I_{in}^0$$

The input reflection coefficient is

$$\rho = (Z_{in} - R_1) \ / \ (Z_{in} + R_1)$$

The magnitude of the input reflection coefficient is given by $|\rho|$.

C Program

Implement the foregoing analysis in your C program. Use the following command to compile and link your program.

```
cc -o output_file_name input_file_name -lm
```

where the `output_file_name` is the executable file name and the `input_file_name` is the file name of your C source code. The argument "`-lm`" instructs the linker to include the math library. For example,

```
cc -o lab1 lab1.c -lm
```

Complete Table I and Table II

TABLE I BRANCH VOLTAGES*

No.	ω(rad/s)	$\lvert\rho\rvert$	V_1(V)	V_2(V)	V_3(V)	V_4(V)	V_5(V)	V_6(V)
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								

* Voltages should correspond to the reference directions of Fig. L1. Voltages should be in the form (Real(voltage), Imaginary(voltage)).

TABLE II BRANCH CURRENTS*

No.	ω(rad/s)	I_1(A)	I_2(A)	I_3(A)	I_4(A)	I_5(A)	I_6(A)
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							

* The reference directions of currents should correspond to the reference directions of voltages from Table I. The currents should be in the form (Real(current), Imaginary(current)).

Submit a report containing your results and the corresponding source code to the instructor.

Lab 3 Introduction to OSA90/hope™ Version 2.5

Objectives

This introductory lab is intended to familiarize students with OSA90/hope.

Software/Hardware

OSA90/hope on Sun SPARCstations (Room JHE 215)

Procedure

Getting Started

(1) Log into one of the 16 SPARCstations. Your initial password is "monday". At the prompt "Open Windows?" - type "y" and <ENTER>. Change your password using the command "passwd" at the UNIX prompt.

(2) Create a symbolic link to OSA90/hope by executing the command

```
"ln -s /home/engine/ee761/osa90 osa90"
```

Subsequently, you will be able to invoke OSA90/hope by simply typing "osa90" without specifying the full path.

(3) Copy the following OSA90/hope demo examples into your directory using the commands

```
"cp /home/engine/ee761/demo01.ckt ./"
"cp /home/engine/ee761/demo02.ckt ./"
"cp /home/engine/ee761/demo66.ckt ./"
```

Basic Experiments with Open Windows

Open Windows options are available by pressing the left and right mouse buttons (1st/3rd) inside and outside of windows. To "drag" (move) a window, click the left mouse button on the top bar of a window, hold the button and move the mouse. Use the right mouse button on the root window to select other actions.

(1) Open two or three concurrent windows (hold right button on root/Programs/Command Tool...). Move and resize a selected window.

(2) Execute two different commands such as "more demo01.ckt" and "vi demo01.ckt" simultaneously from within two different windows. (To leave vi type ":q" or ":q!".)

(3) Close the windows opened in Step 1 by typing "lo" or (hold right button on top bar of window/Quit).

Starting OSA90/hope

(1) Invoke OSA90/hope by entering "osa90" at the system prompt and <ENTER>.

(2) Center the OSA90/hope window by using the mouse.

(3) Move the curser into the OSA90/hope window then press <ENTER> and <N> ("N" stands for name).

(3) Enter a file name (e.g., "lab3.ckt") at the prompt "File Name:" which appears at the bottom of the window. Press <ENTER>.

(4) If this is a new file the program will ask "Create File (Y/<N>)". Answer <Y>. You are now in the OSA90/hope editor.

Using the OSA90/hope Editor (Manual Chapter 2)

(1) Request help messages by pressing <HELP>. Help messages briefly describing the operation and features of the file editor and parser will be displayed page by page upon pressing <ENTER>.

(2) Editor pull-down menu. In addition to using function keys, many of the editing features are available from a pull-down menu. The left-hand mouse button is used to active the pull-down menu. From this menu, move the color bar by moving the mouse or using the arrow keys to highlight the desired option. The left-hand mouse button is used to select a highlighted option by pressing it. Pressing <ENTER> is equivalent. Cancel the menu without selecting an option by pressing the right-hand mouse button or <ESC> key.

Practice Session

(1) Open the Manual at Page 16-7. Follow the step by step guided tour for example demo01. Compare your results with Figs. 16.1 and 16.2.

(2) Follow the step by step guided tour for demo02. Compare your results with Figs. 16.3 and 16.4.

(3) Complete the following tables applicable to demo02.

TABLE I RESULTS FOR demo02

Optimizer	Number of Iterations	Objective Value	
		Before Optimization	After Optimization
ℓ_1			

TABLE II RESULTS FOR demo02

Variables	Before Optimization	After Optimization
x_1		
x_2		
x_3		
x_4		

HPGL Plotter Files for Graphics Hardcopies

(1) Read in the file demo01.ckt again.

(2) Exit the editor, invoke the Display option, choose the Parametric option, View 1 (several circles).

(3) Press the <PRSC> (or <Ctrl-P>). Select the "Format" option as "HPGL for WP" and press <Enter> on "Ready to go." This will generate an HPGL (.plt) file for this graphics display, which can be incorporated into a WordPerfect document.

Contour Plotting

(1) Read in the file demo66.ckt.

(2) Exit the editor, invoke the Display option, choose the Contour option.

(3) Quit the Contour window. Exit the editor again and choose Optimize - using the minimax optimizer. Then choose Display, and Contour.

(4) Notice the white path taken by the optimizer. Practice generating a contour plot by choosing the Contour option (within the contour window) followed by the "HPGL" option, which generates a plot file called "con3d.plt." Try producing a flat contour plot by rotating the view.

Exit OSA90/hope and Open Windows

(1) Exit OSA90/hope (Menu: Terminate). Log out (Unix: "lo") of all your windows (or use "Quit" using the right mouse button on the top bar of each window).

(2) Then, using the right button on the root window, hold it, and choose "Exit", then "Exit" again.

(3) Select NO to Sunview.

(4) Then, logout by typing "lo" on the grey root screen.

Lab 5 Simulation of an LC Transformer

Objectives

This lab has been designed for students to perform linear circuit simulations using OSA90/hope.

Software/Hardware

OSA90/hope on SUN SPARCstations (Room JHE 215)

Circuit Diagram and Parameter Values

Consider the LC transformer circuit shown in Fig. L5. Use OSA90/hope to compute the branch currents, branch voltages and magnitude of the input reflection coefficient of the LC circuit. Use 21 uniformly spaced points in the frequency range 0.5 – 1.179 rad/s to obtain the reflection coefficient. Use frequency points 0.5 rad/s, 0.8 rad/s and 1.1 rad/s to obtain the requested voltages and currents. The parameter values are $E = 3$ V, $R_1 = 3\ \Omega$, $R_2 = 1\ \Omega$, $L_1 = 1.041$ H, $C_2 = 0.979$ F, $L_3 = 2.341$ H, $C_4 = 0.781$ F, $L_5 = 2.937$ H, $C_6 = 0.347$ F.

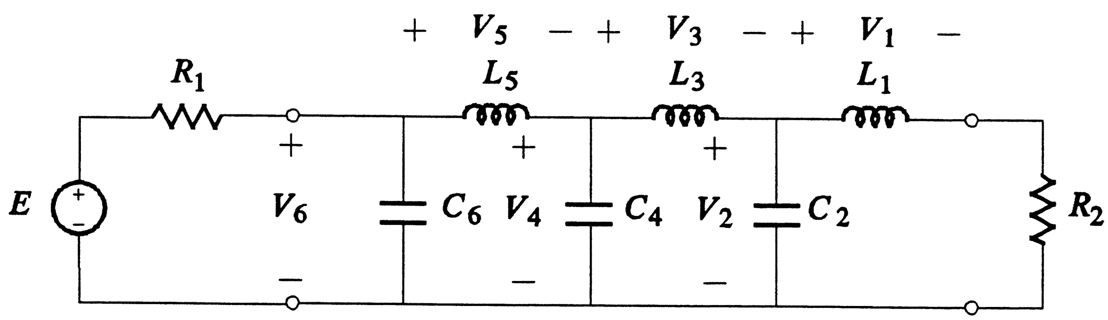

Fig. L5 A lumped element LC transformer.

Procedure

Create Your Circuit File

(1) Create a Control block to set the default unit system to Non_Microwave_Units (Manual Chapter 3, Pages 3–13 to 3–17).

(2) Create a Model block to describe the circuit.

(a) Set the value of the angular frequency to 0.5 by assigning 0.5 to the label Omega (Manual Chapter 4, Page 4–2).

(b) Define the elements R_2, L_1, L_3, L_5, C_2, C_4 and C_6 using the keywords RES, IND and CAP (Manual Chapter 6 – "Elements") (Units on page 3–17) and the topology of the circuit (Manual Chapter 6).

(c) Define the input port with the excitation (both the voltage E and the resistance R_1) (Manual Chapter 6 – "Ports"), e.g.,

```
PORT node1 node2 NAME=portname V=E R=R1;
```

(d) Define complex branch voltages V_1, V_2, V_3, V_4, and V_5 using Voltage Labels (Manual Chapter 6 –"Voltage Labels"). For example,

```
VLABEL node1 node0 NAME=V1;
```

Notice that this implies defining arrays: MV1[], PV1[], ..., MV5[], PV5[] where "M" stands for *magnitude* and "P" stands for *phase* of the complex voltages (Manual Chapter 6, page 6–31). In our case (linear circuit under sinusoidal excitation), only the fundamental components, namely, MV1[1], PV1[1], ..., MV5[1] and PV5[1], are nonzero.

(e) Complete the circuit description using the CIRCUIT statement (Manual Chapter 6 – "The Circuit Statement").

(f) You do not need to define the branch voltage V_6 by VLABEL, because it coincides with the input port. V_6 is automatically available as MV*portname*[1] and PV*portname*[1], where *portname* is the name of the input port you defined at Step 2(c).

(g) Use the MP2RI transformation (Manual Chapter 4 Page 4–12 – "Transformations") to find the real and imaginary parts of the requested voltages. For example:

```
MP2RI(MV1[1], PV1[1], RV1, IV1);
```

where RV1 and IV1 are the real and imaginary parts of the voltage V_1. Apply Ohm's Law to compute the real and imaginary parts of the requested currents.

(h) Terminate your Model block with the "End" keyword.

(3) Define the Sweep block (Manual Chapter 9). Find the magnitude of the reflection coefficient MS11 and real and imaginary part of all branch voltages and currents over the specified range of frequency.

(a) AC simulation (Manual Chapter 9 - "Sweep Block"):

```
AC:  Omega: from 0.5 to 1.179 n=20, FREQ=(Omega/(2*PI)), RREF=3, MS11;
```

AC simulation computes the small-signal S parameters. MS11 is one of them, namely the modulus of S_{11}

(b) HB simulation (Manual Chapter 9 - "Sweep Block"):

```
HB:  Omega: 0.5 0.8 1.1 FREQ=(Omega/(2*PI)) RV1, IV1, ..., V6, IV6, RI1,
     II1, ..., RI6, II6;
```

HB simulation computes the branch voltages and currents.

(c) Terminate your Sweep block with the "End" keyword.

The Sweep block may have the following appearance.

```
Sweep
   AC: Omega: from 0.5 to 1.179 n=20  FREQ=(Omega/(2*PI)), RREF=30H, MS11;
   HB: Omega: 0.5 0.8 1.1 FREQ=(Omega/(2*PI)), RV1, IV1, RV2, IV2, RV3,
       IV3, RV4, IV4, RV5, IV5, RV6, IV6, RI1, II1, RI2, II2, RI3, II3,
       RI4, II4, RI5, IV5, RI6, II6;
End
```

Perform Circuit Simulation

(1) Parse the circuit file by exiting the file editor (press <F7>).

(2) Press <D> to choose the "Display" option.

(3) Press <X> to choose the "Xsweep" option.

(4) Use the "Graphical" option to display the response diagrams (Manual Chapter 9 - "OSA90.Display Menu Option").

(5) Use the "Numerical" option to obtain the numerical outputs. To obtain all the responses, you need to toggle among the three sweep sets in the "Xsweep" menu.

Complete The Following Tables

You may find the Report Generation feature of OSA90/hope helpful in generating the tables. See the Appendix for details.

TABLE I REFLECTION COEFFICIENT

No.	ω (rad/s)	MS11= $\|\rho\|$
1	0.5	
2	0.8055	
3	1.179	

TABLE II BRANCH VOLTAGES*

No.	ω (rad/s)	V_1(V)	V_2(V)	V_3(V)	V_4(V)	V_5(V)	V_6(V)
1	0.5						
2	0.8						
3	1.1						

* The voltages should be presented in the form (Real(voltage), Imaginary(voltage)).

TABLE III BRANCH CURRENTS*

No.	ω (rad/s)	I_1(A)	I_2(A)	I_3(A)	I_4(A)	I_5(A)	I_6(A)
1	0.5						
2	0.8						
3	1.1						

* The currents should be presented in the form (Real(current), Imaginary(current)).

Appendix for Lab 5

Report Generation

The Report Generation feature in OSA90/hope helps the user create elegantly formatted numerical outputs directly from the OSA90/hope text editor.

For example, by adding the following Report block, a numerical report will be generated to help you in filling in the Tables I, II and III on the previous pages.

Report Block Template

```
Report

                    REAL PARTS OF BRANCH VOLTAGES
---------------------------------------------------------------
    Omega    RV1        RV2        RV3        RV4        RV5        RV6
---------------------------------------------------------------
${      $%3.1f$$Omega$Hz   $% -6.4f$$RV1$  $% -6.4f$$RV2$  $% -6.4f$$RV3$   $%
-6.4f$$RV4$  $% -6.4f$$RV5$  $% -6.4f$$RV6$}$
---------------------------------------------------------------


                 IMAGINARY PARTS OF BRANCH VOLTAGES
---------------------------------------------------------------
    Omega    IV1        IV2        IV3        IV4        IV5        IV6
---------------------------------------------------------------
${      $%3.1f$$Omega$Hz   $% -6.4f$$IV1$  $% -6.4f$$IV2$  $% -6.4f$$IV3$   $%
-6.4f$$IV4$  $% -6.4f$$IV5$  $% -6.4f$$IV6$}$
---------------------------------------------------------------


                    REAL PARTS OF BRANCH CURRENTS
---------------------------------------------------------------
    Omega    RI1        RI2        RI3        RI4        RI5        RI6
---------------------------------------------------------------
${      $%3.1f$$Omega$Hz   $% -6.4f$$RI1$  $% -6.4f$$RI2$  $% -6.4f$$RI3$   $%
-6.4f$$RI4$  $% -6.4f$$RI5$  $% -6.4f$$RI6$}$
---------------------------------------------------------------


                 IMAGINARY PARTS OF BRANCH VOLTAGES
---------------------------------------------------------------
    Omega    II1        II2        II3        II4        II5        II6
---------------------------------------------------------------
${      $%3.1f$$Omega$Hz   $% -6.4f$$II1$  $% -6.4f$$II2$  $% -6.4f$$II3$   $%
-6.4f$$II4$  $% -6.4f$$II5$  $% -6.4f$$II6$}$
---------------------------------------------------------------
```

```
MAGNITUDE OF THE REFLECTION COEFFICIENT
--------------------
    Omega   MS11
--------------------
${    $%5.3f$$Omega$Hz  $% -6.4f$$MS11$}$
--------------------

End
```

Please notice that it is not necessary to quit the OSA90/hope editor to generate the report. More details can be found about the report generation feature on Page 9-33 of the OSA90/hope manual.

The disk file for the above report block template is available in the directory

/home/engine/ee761/labs/lab5

with the filename "lab5.rpt". You can incorporate it into your circuit file by the mark and copy features of OSA90/hope (Manual Chapter 2 - "Block Operations").

Lab 6 Simulation and Optimization Using Datapipe with the C Program

Objectives

This laboratory is intended to familiarize students with simulation and optimization using the Datapipe feature of OSA90/hope.

Software/Hardware

OSA90/hope on SUN SPARCstations (Room JHE 215)

Prelab

Prepare the circuit file and the C program for Datapipe communication, as outlined in the Appendix. You can edit the files using the SUN SPARCstations in Room JHE 215 or any other computer available to you, e.g., PC computers in the Computer Centre in JHE 234. In the latter case, transfer the files to the SUN SPARCstations in JHE 215 before the laboratory starts. 20% of the grade will be based on the prelab.

Circuit Diagram and Parameter Values

Consider the LC transformer circuit shown in Fig. L6. Use OSA90/hope to carry out minimax optimization on the magnitude of the input reflection coefficient using all reactive components as optimization variables. Use 21 uniformly spaced points in the frequency range 0.5 – 1.179 (rad/s).

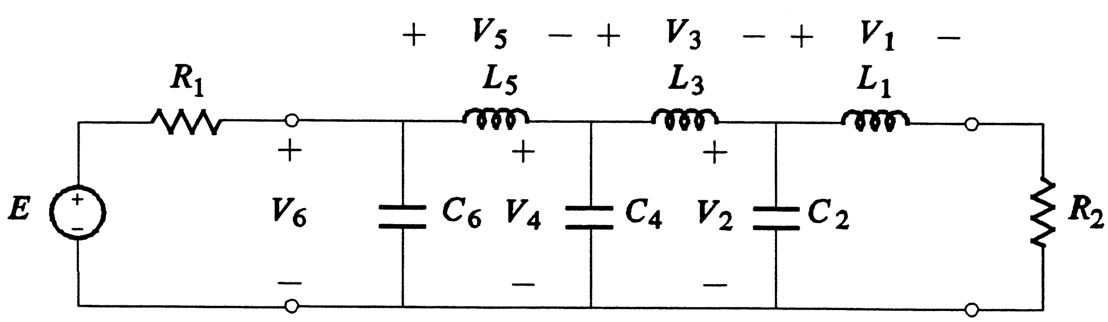

Fig. L6 A lumped element LC transformer.

The suggested starting point is

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

$E = 3$ V, $R_1 = 3\ \Omega$ and $R_2 = 1\ \Omega$. The solution, rounded to 3 decimal places, is

$$L_1 = 1.041$$

$$L_3 = 2.341$$

$$L_5 = 2.937$$

$$C_2 = 0.979$$

$$C_4 = 0.780$$

$$C_6 = 0.347$$

Procedure

Part I: Optimization Using the OSA90/hope Built-in Simulator

(1) Create the circuit file.

(a) Create a Model block to describe the circuit. Define the elements R_2, L_1, L_3, L_5, C_2, C_4 and C_6. Specify the parameters of L_1, L_3, L_5, C_2, C_4 and C_6 as optimization variables (Manual Chapter 6 - "Elements" and Chapter 4 - "Optimization Variables"). For example,

```
L1: ?1?;
IND n1 n2 L = L1;
```

or,

```
IND n1 n2 L = ?1?;
```

(b) Define the input port including both the voltage E and the resistance R_1 (Manual Chapter 6 - "Ports"):

```
PORT node1 node2 NAME=portname V=E R=R1;
```

(c) Complete the circuit description using the CIRCUIT statement (Manual Chapter 6 - "Circuit Statement").

(d) Assign the initial value (0.5) to the label "Omega".

(e) Terminate the Model block with the "End" keyword.

(f) Define the Sweep block (Manual Chapter 9 - "Sweep Block"). This will enable OSA90/hope to display the magnitude of the reflection coefficient MS11 over the specified range of frequency, graphically or numerically (Manual Chapter 6 – "Small-Signal Responses" and Chapter 9 – "OSA90.Display Menu Option"):

```
Sweep
   AC: Omega: from 0.5 to 1.179 n=20, FREQ=(Omega/(2*PI)), RREF=3, MS11;
End
```

(g) Define the Specification block (Manual Chapter 11 – "Specification Block"). The goal for optimization is to minimize the magnitude of the reflection coefficient MS11 over the specified range of frequency.

```
Specification
   AC: Omega: from 0.5 to 1.179 n=20, FREQ=(Omega/(2*PI)), RREF=3,
       MS11=0;
End
```

(2) Save the circuit file you have created: press <Ctrl-S>, enter the file name, and press <ENTER>.

(3) Perform minimax optimization.

(a) Exit the editor and parse the circuit file: press <F7>.

(b) Press <O> to choose the "Optimization" option.

(c) From the pop-up window, toggle to the "Optimizer" option and choose "Minimax". Change the "Number of iterations" option to "999".

(d) Press <ENTER> twice to start the optimization process.

(4) After the optimization is completed, press <F> to return to the circuit file. The optimizable parameters should have been updated with the minimax solution.

(5) Save the optimized circuit file for later comparison: press <Ctrl-S>, enter a new file name and press <ENTER>.

DON'T FORGET TO SAVE YOUR CURRENT FILE BEFORE GOING ON TO THE NEXT STEP.

Part II: Simulation and Optimization via Datapipe

(1) Modify your C program for Lab 1 to conform to the interface requirements outlined in the Appendix. This should have been done in the prelab. You can use the OSA90/hope editor for

any necessary corrections during the lab. If you don't want to use your own C program, there is a model C program in the Appendix.

(2) Exit from OSA90/hope and create a soft link to the C interface file by entering the following command:

```
ln -s /home/engine/ee761/labs/lab6/lab6_sim.o lab6_sim.o
```

This interface file facilitates the Datapipe communication between OSA90/hope and your child program.

(3) Generate the executable program using the following command at the system prompt.

```
cc -o lab6_sim lab6_sim.o filename.c -lm
```

"lab6_sim" is the executable file name you specified in the Datapipe definition in your circuit file. "filename.c" is the source file containing your C program.

(4) Perform minimax optimization within OSA90/hope.

(a) Invoke OSA90/hope again at the system prompt by entering "osa90".

(b) Read in the circuit file you have prepared for Datapipe (or create one using the editor as outlined in the Appendix).

(c) Exit the editor and parse the circuit file (press <F7>).

(d) Press <O> to choose the "Optimization" option.

(e) From the pop-up window, toggle to the "Optimizer" option and choose "Minimax". Change the "Number of iterations" option to "999".

(f) Press <ENTER> twice to start the optimization process.

(5) After the optimization is completed, press <F> to return to the circuit file. The optimizable parameters should have been updated with the minimax solution.

(6) Save the optimized circuit file: press <Ctrl-S>, enter a new file name and press <ENTER>.

Complete the Following Tables

After you finish both Part I and Part II, please place the minimax solutions on the lines, designated as "Case A" in Tables I and II. Then try two additional cases, as indicated below, for minimax optimization and fill in the Tables with the corresponding results.

(1) Case B:

Starting point: solution to Case A.

Range of frequency: 11 uniformly spaced points in the range 0.5 – 1.0.

(2) Case C:

Starting point: original starting point.

Range of frequency: 11 uniformly spaced points in the range 0.5 – 1.0.

Use an upper specification of 0.04 .

While filling in the tables you can find the Report Generation feature of OSA90/hope helpful.

See the Appendix for details.

TABLE I MINIMAX SOLUTION FOR PART I

	L_1	L_3	L_5	C_2	C_4	C_6	max $\|\rho\|$
Case A							
Case B							
Case C							

"max $|\rho|$" is the maximum of the magnitudes of the reflection coefficients at the specified frequency points.

TABLE II MINIMAX SOLUTION FOR PART II

	L_1	L_3	L_5	C_2	C_4	C_6	max $\|\rho\|$
Case A							
Case B							
Case C							

"max $|\rho|$" is the maximum of the magnitudes of the reflection coefficients at the specified frequency points.

Appendix for Lab 6

Templates for the Datapipe Connection

This template assumes that your C program calculates the reflection coefficient at a single frequency point. The frequency range is defined in the OSA90/hope circuit file and the child program is invoked repeatedly within the frequency loop. The advantage of this approach is that it allows you to define an arbitrary frequency range.

Circuit File Template

```
Expression
   L1 = ? 1 ?;
   C2 = ? 1 ?;
   L3 = ? 1 ?;
   C4 = ? 1 ?;
   L5 = ? 1 ?;
   C6 = ? 1 ?;

   Omega: 0.5;

   Datapipe: SIM FILE = "lab3_sim"
             N_INPUT = 7   INPUT = (L1, C2, L3, C4, L5, C6, Omega)
             N_OUTPUT = 1  OUTPUT = (Rout);
End

Sweep
   Omega: From 0.5 to 1.179 N=20 Rout;
End

Specification
   Omega: From 0.5 to 1.179 N=20 Rout=0;
End
```

C Template

```
int trnsfmr ( Par, Omega, ReflectionCoef )
float Par[6], Omega, *ReflectionCoef;
{
   /* this subroutine calculates the lc transformer response at one
      frequency given by Omega.  The values of L1, C2, L3, C4, L5 and
      C6 are supplied in the array Par.  The response is the
      magnitude of the input reflection coefficient.
   */

   ...
   ...
}
```

Example of C Program

```
#include <stdio.h>
#include <math.h>

   typedef struct { float real; float imag; }   COMPLEX;

   int trnsfmr ();
   COMPLEX ComplexAdd (),
           ComplexSub (),
           ComplexMul (),
           ComplexDiv ();
   float ComplexAbs ();

int trnsfmr ( Par, Omega, ReflectionCoef )
float Par[6], Omega, *ReflectionCoef;
{
   /* this subroutine calculates the lc transformer response at one
      frequency given by Omega.  The values of L1, C2, L3, C4, L5 and
      C6 are supplied in the array Par.  The response is the
      magnitude of the input reflection coefficient.
   */

   /* local variables */

   COMPLEX jOmegaPar[6], I0[6], V0[6], Iin, Vin, Iout, Vout,
           Is0, Rin, Zin;
   int i;

   for (i = 0; i < 6; i++) {
      jOmegaPar[i].real = 0.0;
      jOmegaPar[i].imag = Omega * Par[i];
   }

   Iout.real = Vout.real = 1.0;
   Iout.imag = Vout.imag = 0.0;

   I0[0] = Iout;
   V0[0] = ComplexMul(I0[0], jOmegaPar[0]);
   V0[1] = ComplexAdd(V0[0], Vout);
   I0[1] = ComplexMul(V0[1], jOmegaPar[1]);
   I0[2] = ComplexAdd(I0[1], I0[0]);
   V0[2] = ComplexMul(I0[2], jOmegaPar[2]);
   V0[3] = ComplexAdd(V0[2], V0[1]);
   I0[3] = ComplexMul(V0[3], jOmegaPar[3]);
   I0[4] = ComplexAdd(I0[3], I0[2]);
   V0[4] = ComplexMul(I0[4], jOmegaPar[4]);
   V0[5] = ComplexAdd(V0[4], V0[3]);
   I0[5] = ComplexMul(V0[5], jOmegaPar[5]);

   Iin = ComplexAdd(I0[5], I0[4]);
```

```
   Vin = V0[5];

   Rin.real = 3.0;
   Rin.imag = 0.0;

   Zin = ComplexDiv(Vin, Iin);
   *ReflectionCoef = ComplexAbs(ComplexDiv(ComplexSub(Zin, Rin),
                                           ComplexAdd(Zin, Rin)));

   return(0);
}

COMPLEX ComplexAdd ( a, b )
COMPLEX a, b;
{
   COMPLEX c;

   c.real = a.real + b.real;
   c.imag = a.imag + b.imag;

   return(c);
}

COMPLEX ComplexSub ( a, b )
COMPLEX a, b;
{
   COMPLEX c;

   c.real = a.real - b.real;
   c.imag = a.imag - b.imag;

   return(c);
}

COMPLEX ComplexMul ( a, b )
COMPLEX a, b;
{
   COMPLEX c;

   c.real = a.real * b.real - a.imag * b.imag;
   c.imag = a.imag * b.real + a.real * b.imag;

   return(c);
}

COMPLEX ComplexDiv ( a, b )
COMPLEX a, b;
{
   COMPLEX c, d;
   float det;

   det = b.real * b.real + b.imag * b.imag;
```

```
   d.real = b.real / det;
   d.imag = -b.imag / det;

   c = ComplexMul(a, d);

   return(c);
}

float ComplexAbs ( a )
COMPLEX a;
{
   float det;

   det = sqrt(a.real * a.real + a.imag * a.imag);

   return(det);
}
```

Report Generation

The Report Generation feature of OSA90/hope facilitates postprocessing of OSA90/hope results. It allows the user to embed numerical outputs within an arbitrary text document.

In our case we can customize the OSA90/hope output so that it resembles Tables I and II as closely as possible. A possible Report block for our problem follows.

Report Block Template

```
Report
      Table I.  Minimax Solutions for Part I
----------------------------------------------------------------------------
           L1        L3        L5        C2        C4        C6        MS11
----------------------------------------------------------------------------

Case A     $%6.4f$$L1$   $%6.4f$$L3$   $%6.4f$$L5$   $%6.4f$$C2$   $%6.4f$$C4$
 $%6.4f$$C6$   $%6.4f$$max$

----------------------------------------------------------------------------
End
```

max indicates the maximum value of MS11.

The files containing the circuit file, the C program and the report block templates can be found in the directory "/home/engine/ee761/labs/lab6". Their names are "lab6_p2.ckt", "lab6_csim.c" and "lab6.rpt", respectively.

Lab 9 Function Approximation

Objectives

This lab is intended to familiarize students with function approximation using the built-in optimizers of OSA90/hope.

Software/Hardware

OSA90/hope on Sun SPARCstations (Room JHE 215).

Problem to be Solved

Use the ℓ_1, ℓ_2 and minimax optimizers of OSA90/hope to approximate in a uniformly weighted 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].

Let

$$y_1 = \frac{[(8x - 1)^2 + 1]^{0.5} \tan^{-1}(8x)}{8x}$$

$$y_0 = \frac{a_0 + a_1 x + a_2 x^2}{1 + b_1 x + b_2 x^2}$$

$$y_2 = y_0 - y_1$$

where y_2 define the error function.

In this problem, a_0, a_1, a_2, b_1 and b_2 are optimization variables. The suggested starting point is: $a_0 = 0$, $a_1 = -10$, $a_2 = 20$, $b_1 = -1$ and $b_2 = 20$.

Procedure

Create Your OSA90/hope Input File

(1) Create the Expression block. Define the optimization variables with initial values, lower and upper bounds (Manual Chapter 4 - "Optimization Variables"). Define the functions using expressions (Manual Chapter 4 - "Expressions"). Initialize the sweep parameter x with any real number except 0.

(2) Create the Sweep block. Sweep parameter x from -1 to -0.01 with step 0.01 and from 0.01 to 1 with step 0.01 (Manual Chapter 9 - "Sweep Block"). The output responses are y_0, y_1 and y_2.

(3) Create the Specification block. Sweep parameter x as defined in (2). The specification is $y_2 = 0$ (Manual Chapter 11 - "Specification Block").

(4) Save your file as "lab9.ckt".

Optimization

(1) Perform ℓ_1 optimization using "999" for the number of iterations and "1E-06" for the accuracy of solution. After the optimization has terminated copy down the objective error at the first iteration and at the solution, and the number of iterations. Record the values of the optimization variables and save your updated file as "lab9_l1.ckt".

(2) Read in the file "lab9.ckt". Follow Step (1) to perform ℓ_2 optimization. Save your updated file as "lab9_l2.ckt".

(3) Read in the file "lab9.ckt". Follow Step (1) to perform minimax optimization. Save your updated file as "lab9_m.ckt".

Complete the Following Tables

TABLE I RESULTS FOR LAB 9

Number of Optimizer Iterations	Objective Value	
	Before Optimization	After Optimization
ℓ_1		
ℓ_2		
minimax		

TABLE II RESULTS FOR LAB 9

Variables	Before Optimization	After Optimization		
		ℓ_1	ℓ_2	minimax
a_0				
a_1				
a_2				
b_1				
b_2				

Simulation

Perform simulations using the input files: lab9.ckt, lab9_l1.ckt, lab9_l2.ckt and lab9_m.ckt. Observe the response curves. Comment on your observation in the space below.

Lab 10 Advanced Applications of OSA90/hope™ Version 2.5

Objectives

This demonstration lab has been devised to strengthen students' knowledge on optimization, as well as show its applications to engineering design problems.

Software/Hardware

OSA90/hope on Sun SPARCstations (Room JHE 215)

Procedure

Starting the Macro Procedure

(1) Invoke OSA90/hope on one of the 16 SPARCstations. Press <ENTER> to list the input files in your current working directory.

(2) Select any no-syntax-error OSA90/hope input file in this directory. Press <F7> to exit from the file editor. Press <M> to switch from interactive mode to macro mode (Manual Chapter 14). At the prompt "Macro File Name:" which appears at the bottom of the OSA90/hope window, enter the following macro file name, including the entire path:

"/home/engine/ee761/labs/lab9/lab9.mcr"

(3) Now OSA90/hope is running in the macro mode. Please listen carefully to the demonstrator and follow his/her instructions. Hit the space bar once and **only once** as instructed. In case you accidentally exit the macro mode, please restart the macro mode and hit as many keystrokes as possible to catch up with the class.

The Guided Tour

(1) "ex01.ckt": this example demonstrates minimax optimization of a small signal amplifier. In the first part of the circuit file, we see some optimization variables are defined. After we press the space bar, the circuit file scrolls to the next page and shows some definition of the circuit. The next graphic output shows the circuit response before the optimization, two horizontal lines represent the upper and lower specifications. It is clear that the response violates the

specifications. The next graphic output is an enlarged view of the previous picture. The next screen shows the circuit file after the optimization. You may have noticed that the values of the optimization variables are changed. The following picture depicts the circuit response after the optimization. We see that the specifications are satisfied.

(2) "ex02.ckt": this example demonstrates using the ℓ_1 optimization to fit the FET measurement data by changing the FET model parameters. The circuit file shows that the measurement data is put into the array called Id_data, and the specification is to make the FET circuit response equal the measurements. The next picture depicts the scenario before the optimization. The yellow circuit represent the measurement data and the green curves represent the FET model response. The next screen shows the circuit file after the optimization. After you hit the space bar twice, you will see the model response curves perfectly match the measurements.

(3) "mux05.ckt": this example tries to optimize the parameters of a microwave multiplexer.

(4) "demo02.ckt": this example tries to approximate the square-root function

$$Goal = \mathrm{sqrt}(t)$$

by a rational function

$$Function = (x_1 t + x_2 t^2) \;/\; (1 + x_3 t + x_4 t^2)$$

We wish to optimize the coefficients x_1, x_2, x_3 and x_4 so that *Function* will match *Goal* for t = 0, 0.002, 0.004, ..., 1.

The first two diagrams show the absolute errors and the matching before optimization. After you hit the space bar, the circuit file after ℓ_2 optimization shows up. The following diagram displays the errors at the ℓ_2 optimum. As you look closer, you can see that the scales of the errors are vary small. The next picture shows the ℓ_2 matching. The circuit file at the minimax solution is displayed on the next screen. After you hit the space bar, you can see that four maximum errors show up at the solution. Their value is the value of the minimax objective function at the solution. The next picture shows the minimax matching.

(5) "q92a.ckt": in this example, we try to solve a minimax problem for which

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

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

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

The graphic output shows the error functions at the solution. Three error functions have the same value at the solution.

(6) "q93.ckt": this example solves a minimax problem similar to the above example.

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

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

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

Only the roles of ϕ_1 and ϕ_2 in the first error function interchanged. Hit the space bar. This time, two of the error functions (f_1 and f_2) have the maximum value at the solution.

(3) "q92b.ckt": this example solves a minimax problem for which

$$f_1 = \phi_2^2 - 3$$

$$f_2 = - \phi_1 + 2$$

$$f_3 = \phi_1 - \phi_2 - 1$$

The first diagram shows the three error functions at the starting point. The following is a real-time minimax optimization. The upcoming graphic view displays the error functions at the solution: three error functions are of the same height.

(4) "q98.ckt": this example demonstrates a simple data fitting problem. Four data points are at 1, 1, 1.5, and 1. We try to put a line through the data points. The first diagram shows the situation before optimization. The white dots represent the data points; the yellow line is the approximating line and the green bars are the errors between the appoximating function and the data. Hit the space bar, and we will go through a real-time ℓ_2 optimization. And the next screen shows the least-squares fitting. Hit the space bar again, and this time we go through a real-time ℓ_1 optimization. The next diagram shows that at the ℓ_1 solution, the approximating line goes through the majority of the data points, neglecting the point with inconsistent value. The fourth diagram shows the situation after minimax optimization: the line is put exactly in the middle between the points. This means that the point at 1.5 is weighted as important as all three other

points collectively.

(5) "newton.ckt": this example provides a step-by-step demonstration of the Newton-Raphson method for obtaining the roots of a quadratic function. The first pause finds the root of -1, and the second pause finds the root of 1.

(9) "newton1.ckt": this example demonstrates the Newton steps to find the minimum of the function

$$f = x_1^2 + 2x_2^2 + x_1x_2 + 2x_1 + 1$$

Since this is a quadratic function, the problem is solved in two iterations.

(10) "newton2.ckt": this example demonstrates the Newton iterations for minimizing the function

$$f = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Forty iterations are shown.

(11) "demo67.ckt": this example shows how to use OSA90/hope as a child program to provide a circuit simulation result for optimization in a parent OSA90/hope program. A 3-D visualization feature is invoked to assist the analysis of the optimization process.

Appendix for Lab 10

List of Macro File

```
"f", READ, "n/home/engine/ee761/labs/lab9/ex01.ckt", ENTER, PAUSE
PgDn, PAUSE, EXIT, "dxy", LEFT, ENTER, ENTER, PAUSE, ENTER,
"zy5", DOWN, "11", ENTER, "t6", UP, "8", ENTER, ENTER, ENTER,
ENTER, PAUSE, ESC,

"f", READ, "n/home/engine/ee761/labs/lab9/ex01_o.ckt",
ENTER, PAUSE, EXIT, "dxy", LEFT, "zy3", DOWN, "10", ENTER, "t7",
UP, "8", ENTER, ENTER, ENTER, ENTER, PAUSE, ESC,

"f", READ, "n/home/engine/ee761/labs/lab9/ex02.ckt", ENTER, PAUSE,
PgDn, PAUSE, EXIT, "dx", ENTER, PAUSE, ESC,

"f", READ, "n/home/engine/ee761/labs/lab9/ex02_o.ckt", ENTER, PAUSE
PgDn, PAUSE, EXIT, "dx", ENTER, PAUSE, ESC,

"f", READ, "n/home/engine/ee761/labs/lab9/mux05.ckt", ENTER, PAUSE
PgDn, PAUSE, EXIT, "dx", ENTER, PAUSE, ESC

"f", READ, "n/home/engine/ee761/labs/lab9/demo02a.ckt", ENTER, PAUSE,
EXIT, "dx", UP, UP, UP, LEFT, DOWN, LEFT, LEFT, UP, UP, UP, RIGHT,
DOWN, DOWN, DOWN, DOWN, DOWN, ENTER, PAUSE, ENTER, UP, UP, UP,
LEFT, DOWN, LEFT, DOWN, DOWN, ENTER, PAUSE, ESC

"f", READ, "n/home/engine/ee761/labs/lab9/demo02_lsq.ckt", ENTER, PAUSE,
EXIT, "dx", UP, UP, UP, LEFT, DOWN, LEFT, LEFT, UP, UP, UP, RIGHT,
DOWN, DOWN, DOWN, DOWN, DOWN, ENTER, PAUSE, ENTER, UP, UP, UP, LEFT,
DOWN, LEFT, DOWN, DOWN, ENTER, PAUSE, ESC

"f", READ, "n/home/engine/ee761/labs/lab9/demo02_mmo.ckt", ENTER, PAUSE,
EXIT, "dx", UP, UP, UP, UP, UP, RIGHT, DOWN, DOWN, RIGHT, UP, UP,
RIGHT, DOWN, DOWN, DOWN, LEFT, LEFT, DOWN, DOWN, ENTER, PAUSE, ENTER,
UP, UP, UP, LEFT, DOWN, LEFT, DOWN, DOWN, ENTER, PAUSE, ESC

"f", READ, "n/home/engine/ee761/labs/lab9/q92a.ckt", ENTER, PAUSE, EXIT,
"d", ENTER, UP, UP, UP, UP, UP, RIGHT, ENTER, ENTER,  PAUSE, ESC,

"f", READ, "n/home/engine/ee761/labs/lab9/q93.ckt", ENTER, PAUSE, EXIT, "d",
ENTER, UP, UP, UP, UP, UP, RIGHT, ENTER, ENTER, PAUSE, ESC

"f", READ, "n/home/engine/ee761/labs/lab9/q92b.ckt", ENTER, PAUSE, EXIT,
"dx", UP, UP, UP, UP, UP, RIGHT, ENTER, ENTER, PAUSE, ESC, "o", UP,
UP, UP, UP, UP, RIGHT, RIGHT, ENTER, ENTER, PAUSE,

"f", EXIT, "dx", UP, UP, UP, UP, UP, RIGHT, ENTER, ENTER, PAUSE, ESC

"f", READ, "yn/home/engine/ee761/labs/lab9/q98.ckt", ENTER, PAUSE,
EXIT, "dx", ENTER, PAUSE, ESC, "o", "o", RIGHT, DOWN, DOWN, RIGHT,
RIGHT, RIGHT, DOWN, RIGHT, RIGHT, ENTER, ENTER, PAUSE, "dx", ENTER,
```

```
PAUSE, ESC,

"f", READ, "yn/home/engine/ee761/labs/lab9/q98.ckt", ENTER, EXIT,
"o", UP, UP, UP, RIGHT, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, ENTER,
"o", LEFT, ENTER, ENTER, PAUSE, "dx", ENTER, PAUSE, ESC,

"f", READ, "yn/home/engine/ee761/labs/lab9/q98.ckt", ENTER, EXIT,
"o", "o", RIGHT, RIGHT, DOWN, DOWN, RIGHT, RIGHT, RIGHT, DOWN, RIGHT,
RIGHT, ENTER, ENTER, PAUSE, "dx", ENTER, PAUSE, ESC

"f", READ, "yn/home/engine/ee761/labs/lab9/newton.ckt", ENTER, PAUSE, REPORT,
PAUSE, ESC,

READ, "n/home/engine/ee761/labs/lab9/nton.ckt", ENTER, PAUSE, REPORT, PAUSE,
ESC,

READ, "n/home/engine/ee761/labs/lab9/newton1.ckt", ENTER, PAUSE, REPORT, PAUSE,
ESC,

READ, "n/home/engine/ee761/labs/lab9/newton2.ckt", ENTER, PAUSE, REPORT, PAUSE,
PGDN, PAUSE, ESC, ESC,

"f", READ, "n/home/engine/ee761/labs/lab9/demo67_o.ckt", ENTER, PAUSE, EXIT,
"dc", ENTER, ESC,
```

Appendix 1

ESSENCE OF C

Highlights

Portability - compiler included in UNIX systems.
Flexibility - high-level syntax + low-level capabilities
Disadvantage - no built-in data type for complex numbers.

Essence

Identifiers. Data types. Intrinsics. Expressions.
Flow control. Functions.

Identifiers

Names of variables and functions.
Case sensitive: `xMax`, `NodalVoltage`, ...
All statements end with a semicolon ";".

Reserved

auto	break	case	char	continue	default	do
double	else	entry	extern	float	for	goto
if	int	long	register	return	short	sizeof
static	struct	switch	typedef	union	unsigned	while

Comments

```
/* this is a comment */
```

Data Types

char	'A', 'b', '\n'
int	1, 300, -12345, 010 (octal), 0x2f (hex)
float	-1234.56, 1.3E-5

unsigend, short, long (variations of int)
double (double precision float)

All variables must be typed (declared).

Arrays

```
int index[10];
float matrix[4][4], Parameters[3][5][10];
char FileName[64];
```

Array indices start from 0, hence array index[10] consists of
`index[0], index[1], ..., index[9]`

string is an array of chars ended with '\0', e.g., "Hello world"

Struct

```
struct Complex {
   float real;
   float imag;
};
struct Complex Voltage;
```

Fields of struct: `Voltage.real` `Voltage.imag`

Pointers

```
float *x, y;        /* x is a pointer to float, y is a float variable */
x = &y;             /* x now points to y */
*x = 55.8;          /* same as y = 55.8 */
```

Intrinsic Library

Math

```
acos()   asin()   atan()   atan2()   cos()   sin()     tan()
cosh()   sinh()   tanh()   exp()     log()   log10()   pow()
sqrt()   fabs()   ceil()   floor()   ...
```

I/O

```
scanf()   printf()   gets()   getchar()   ...
```

File

```
fopen()   fscanf()   fprintf()   fclose()   ...
```

String

```
strcpy()   strcmp()   ...
```

Operators

Arithmetic

```
+   -   *   /   ++   --   =   +=   -=
```

Logical

```
||   &&   >   <   >=   <=   ==   !
```

Bitwise

```
|   &   ^   ~   >>   <<
```

Spacing for clarity! (e.g., `a * b` for multiplication, `*k` for pointer)
There is no operator for power x^y, use `pow(x,y)`.

Expression

```
3 * (75 + 3);
c1 * (2.5 / cos(A) + sqrt(x25) - V1) + exp(0.25 * t);
```

If Else

```
if (...) ... else ...

if ( x > 5 ) y = 2 * x;
else y = 10;

if ( x > 5 && x < 10 ) {
   y = 2 * x;
   ....;
}
else if ( x >= 10 ) {
   if ( ... ) ...;
   ....;
}
else {
   ....;
}
```

Loops

```
for (initialization; condition; increment) expression;

for (sum = 0.0, i = 0; i < 10; i++) sum += x[i];

for (i = 0; i < 100; i++) {
   ....;
   ....;
}

for (i = 0; i < 100; i++) {
   if (a[i] > 5.0) break;
   ....;
}

for (;;) {   /* infinite loop */
   ....;
   ....;     /* must contains at least one break */
}
```

Main Program

```
#include <stdio.h>
#include <math.h>
#include <string.h>

main ()
{
   ....;
   exit(0);
}
```

Example

```
/* find the maximum of five numbers entered by the user */

#include <stdio.h>
#include <math.h>
#include <string.h>

main ()
{
   float x[5], xMax;
   int i, iMax;

   printf("Please enter five numbers:\n");
   scanf("%f %f %f %f %f", &x[0], &x[1], &x[2], &x[3], &x[4]);

   for (xMax = x[0], iMax = 0, i = 1; i < 5; i++) {
      if (x[i] > xMax) {
         xMax = x[i];
         iMax = i;
      }
   }
   printf("The largest number is x[%d] = %f\n", iMax, xMax);
   exit(0);
}
```

Compilation `cc -o findmax findmax.c -lm`

where `findmax.c` is the file name of the source code, and `findmax` is the file name of the executable code.

Functions

data_type function_name (arguments)
argument_data_type;
```
{
   ....;
   return ( ... );
}
```

Example

```
int FindMax ( n, x, xMax )
int n;
float x[], *xMax;   /* xMax is a pointer, like VAR arguments in Pascal */
{
   int i, iMax;

   for (*xMax = x[0], iMax = 0, i = 0; i < n; i++) {
      if (x[i] > *xMax) {
         *xMax = x[i];
         iMax = i;
      }
   }

   return (iMax);
}
```

Modified Main Program to Call Function FindMax

```
#include <stdio.h>
#include <math.h>
#include <string.h>

main ()
{
   float x[5], xMax;
   int iMax;

   printf("Please enter five numbers:\n");
   scanf("%f %f %f %f %f", &x[0], &x[1], &x[2], &x[3], &x[4]);

   iMax = FindMax(5, x, &xMax);

   printf("The largest number is x[%d] = %f\n", iMax, xMax);
   exit(0);
}
```

Example

```
/* compute the product of a matrix and a vector */

#include <stdio.h>
#include <math.h>
#include <string.h>

void MatrixVectorProduct ();    /* function declaration */

main ()
{
   float matrix[2][2], vector[2], product[2];

   printf("Please enter a 2x2 matrix:\n");
   scanf("%f %f %f %f", &matrix[0][0], &matrix[0][1],
                        &matrix[1][0], &matrix[1][1]);

   printf("Please enter a vector of 2 elements:\n");
   scanf("%f %f", &vector[0], &vector[1]);

   MatrixVectorProduct(matrix, vector, product);

   printf("The product is %f %f\n", product[0], product[1]);
   exit(0);
}


void MatrixVectorProduct ( matrix, vector, product )
float matrix[2][2], vector[2], product[2];
{
   float sum;
   int i, j;

   for (i = 0; i < 2; i++) {
      for (sum = 0.0, j = 0; j < 2; j++) {
         sum += (matrix[i][j] * vector[j]);
      }

      product[i] = sum;
   }

   return ();
}
```

Array arguments are equivalent to pointer arguments.

Input

```
scanf("%f %d %s", &float_var, &int_var, &string_var);
```

(A string variable is defined as a char pointer.)

```
gets(&string_var);
```
gets() reads a whole line of text including blanks.

Output

```
printf("A float number: %f and an integer: %d\n", float_var, int_var);

printf("line 1 %f\nline 2 %f\nline 3 %f\n", x1, x2, x3);
```

Files

```
FILE *file_in, *file_out;

file_in = fopen("input.dat", "r");
if (!file_in) {
   printf("Cannot open input file\n");
   exit(-1);
}

fscanf(file_in, "%f %d %s", &float_var, &int_var, &string_var);

fclose(file_in);

file_out = fopen("output.dat", "w");
if (!file_out) {
   printf("Cannot open output file\n");
   exit(-1);
}

fprintf(file_out, "Results: %f\n", float_var);

fclose(file_out);
```

Complex

The following are a set of functions in C for manipulating complex numbers.

```
#include <stdio.h>
#include <math.h>

struct Complex { float real; float imag; };

struct Complex ComplexAdd (),
               ComplexSub (),
               ComplexMul (),
               ComplexDiv ();

float ComplexAbs ();

main ()
{
   struct Complex a, b, c;
   float x;

   printf("Enter a.real and a.imag: ");
   scanf("%f %f", &(a.real), &(a.imag));

   printf("Enter b.real and b.imag: ");
   scanf("%f %f", &(b.real), &(b.imag));

   c = ComplexAdd(a,b);
   printf("a + b = %f + j %f\n", c.real, c.imag);

   c = ComplexSub(a,b);
   printf("a - b = %f + j %f\n", c.real, c.imag);

   c = ComplexMul(a,b);
   printf("a * b = %f + j %f\n", c.real, c.imag);

   c = ComplexDiv(a,b);
   printf("a / b = %f + j %f\n", c.real, c.imag);

   x = ComplexAbs(a);
   printf("|a| = %f\n", x);

   exit(0);
}

struct Complex ComplexAdd ( a, b )
struct Complex a, b;
{
   struct Complex c;

   c.real = a.real + b.real;
   c.imag = a.imag + b.imag;

   return(c);
}
```

```
struct Complex ComplexSub ( a, b )
struct Complex a, b;
{
   struct Complex c;

   c.real = a.real - b.real;
   c.imag = a.imag - b.imag;

   return(c);
}


struct Complex ComplexMul ( a, b )
struct Complex a, b;
{
   struct Complex c;

   c.real = a.real * b.real - a.imag * b.imag;
   c.imag = a.imag * b.real + a.real * b.imag;

   return(c);
}


struct Complex ComplexDiv ( a, b )
struct Complex a, b;
{
   struct Complex c, d;
   float det;

   det = b.real * b.real + b.imag * b.imag;

   d.real = b.real / det;
   d.imag = -b.imag / det;

   c = ComplexMul(a, d);

   return(c);
}


float ComplexAbs ( a )
struct Complex a;
{
   float det;

   det = sqrt(a.real * a.real + a.imag * a.imag);

   return(det);
}
```

McMASTER UNIVERSITY BOOKSTORE

CUSTOM COURSEWARE

SERVICE

0249 5026

SKU