

MINIMAX SYSTEM MODELLING AND DESIGN

by

Thandangorai V. Srinivasan, B. Tech. (Hons.), M.E.

A Thesis

Submitted to the Faculty of Graduate Studies'

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

1974

July 1973

© Thandangorai V. Srimivasan

DOCTOR OF PHILOSOPHY (1973)

McMaster University Hamilton, Ontario

TITLE

₹.,

: Minimax System Modelling and Design

AUTHOR	: Thandangorai V. Srinivasan B. Tech. (Hons.). (Indian Institute of Technology, Kharagpur) M.E. (Birla Institute of Technology and Science, Pilani)	
SUPERVISOR	: J.W. Bandler B.Sc. (Eng.), Ph.D. (University of London) D.L.C. (Imperial College)	

NUMBER OF PAGES: x', 175

SCOPE AND CONTENTS:

Computer-aided system modelling and design for minimax objectives have been considered in detail. A new algorithm for minimax approximation, called the grazor search method, has been proposed and successfully used on a number of network design problems to test the reliability and efficiency of the method. A critical comparison of the method with existing algorithms has shown the grazor search algorithm tobe reliable in most of the problems considered. Practical ideas have been presented to deal with constrained minimax optimization problems and to investigate a solution for minimàx optimality. Two user-oriented computer programs incorporating these ideas have been included as part of the thesis. Lower-order modelling of a high-order system has been considered for minimax objectives, and the suggested ideas make it feasible to design automated models for a variety of transient and steady-state constraint specifications.

ii

ACKNOWLEDGEMENTS

The author is greatly indebted to Dr. J.W. Bandler for his expert guidance and supervision throughout the course of this work. The constant encouragement and inspiration the author derived from him are thankfully acknowledged.

The author wishes to acknowledge, in particular, Dr. N.K. Sinha for his advice and interest during the research effort. Special thanks are due to Dr. C.M. Crowe for his constructive criticisms and useful suggestions for the thesis.

Thanks are due to Dr., C. Charalambous for helping with the proof of convergence of the grazor search algorithm and many discussions the author had with him on a number of occasions. The author is very thankful to Dr. G. T. Bereznai for timely advice and helpful suggestions on various aspects of the research work.

The author wishes to acknowledge with thanks the useful discussions he had with his colleagues, in particular, B.L. Bardakjian, M. Beshai, J. Chen, V.K. Jha, A.G. Lee-Chan, P.C. Liu, N.D. Markettos, J.R. Popović, J. Roitman and S.K. Tam.

The financial assistance provided by the National Research Council of Canada through grants A7239 and C154, and the Department of Electrical Engineering is gratefully acknowledged.

The competence and reliability of Mrs. Hazel Coxall, who typed this thesis, has been very much appreciated.

iii .

TABLE OF CONTENTS

· · ·	· · ·	Page
CHAPTER I - INTE	RODUCTION	1
CHAPTER II - REVI	IEW OF MINIMAX METHODS	4
2.1 2.2 2.3	 Introduction Function Minimization Least pth Approximation for Single Specified Function 	4 4 5
	2.3.1 - The Error Function 2.3.2 - Continuous Approximation 2.3.3 - Discrete Approximation	5 5 6
2.4 2.5	- The Minimax Problem - Minimax Methods	7 7
	2.5.1 - The Razor Search Method 2.5.2 - Sequential Unconstrained	8
	Minimization Technique 2.5.3 - Algorithm due to Osborne and	9
-	Watson 2.5.4 - Method due to Bandler and Lee-Chan	11 13
. 2.6	Near-Minimax Methods	14
CHAPTER III - NEI	N APPROACHES TO THE MINIMAX PROBLEM	17
3.1 3.2	- Introduction - The Grazor Search Strategy	17 - 19
•	3.2.1 - Theoretical Considerations 3.2.2 - Proof of Convergence 3.2.3 - Practical Implementation 3.2.4 - Example	19 21 22 28
3.3	- Constrained Minimax Optimization 3.3.1 - Statement of the Problem 3.3.2 - Formulation 1 3.3.3 - Formulation 2 3.3.4 - Comments	- 34 34 35 36 37

- 7

7

iv

	•					-			
				٠.	,	-		-	ι,
				•	· · · ·	• .			
-			· _		,		i.		
	•			j č	,				
		•			•			•	
	► Î	3 A D+2/	tical Tayor	, tigation of	Minimau				r
č -		-3.4 = rrac	imality Cond	itions	MINIMAX	1	38		
`		op of					70	•	
	· · ·	· 3.4.	,1 → Introdu 2 - Conditi	ction one for a Mi	nimar		38		
	•	- J.4.	0ntimum	UNS TOT A MI			38	1	
		3.4	.3 - Practic	al Implement	ation		39	-	
7		3.4	.4 - Method	1	• •		40		~
	-	3.4	.5 - Method	2		,	41		
		3.4	.6 - Comment	S	•		. 41		
	1	3.4	. / - Example		-		42		
•		3.5 - Conc	clusions				45		
CH/	APTER- IV	- COMPUTER-	AIDED CIRCUI	TDESIGN			46		
-		4.1 - Int	roduction	Form			46		
× .		4.2 - 1000	Ded LL ITANS	IOTMET	ine		46		
с ,	1	Trai	nsformer	5 5 5 1 UII = L		· .	48		
· .		4.4 - Cas	caded Transm	ission-Line	Filters		63		
•		- 4.4	.l - Problem	1 1			63		
		4.4	.2 - Problem	12-	$\mathbf{x} = \mathbf{x}$		66		
		- 4.4	.3 - Problem	1.3	•		73		
-		4.5 - Con	clusions		<u> </u>		76		
CHA	APTER V -	· Systen mod	ELLING	r			77.		
	•				-	•			
	I	5.1 - Int	roduction	· • •		· .	77		
		5.2 - Sta	tement of th	Ne Problem	•		78 78		
		5.3 - Min	max system	Modelling			80	.•	
•					¥	•	0 U (<u>.</u>	
	~	5.4 5.4	2 - Ontinal	and Inira-Un	Deremeter	6	01 94		2
	,	5.4	.3 - Discus	sion	i ralancici		101		
•	2						102	•	•
		_~5.5 - New	Approaches	co minimax 3	SYSTEM MODE.	i i i ng	102		
		£ 5.5	11 - A Gener	alized Object	ctive Funct	ion	103		
		- S+S	.Z - Automat	ed Lower-Oro	ier Models	··	104	•	
		r r			3 m #		100	•	•
	•	5.5 c c	$\frac{1}{4} = \frac{1}{2} $	lity Conditi(t	ons	•	106		
,	·	5.5 5.5 5.5	.5 - Optimal .4 - Results .5 - Discuss	lity Conditions sion	ons		106 106 108		ſ
, , •		5.5 5.5 5.5	.3 - Optimal .4 - Result: .5 - Discuss	lity Conditions sion	0 ң 5		106 106 108		ſ
, .	÷	5.5 5.5 5.5 5.6 - Con	 .3 - Optimal .4 - Result: .5 - Discuss clusions 	lity Conditions sion	ons		106 106 108 117		ſ
, .		5.5 5.5 5.5 £.6 - Con	.3 - Optimal .4 - Result: .5 - Discus: clusions	lity Conditions	ons		106 106 108 117		ſ
,		5.5 5.5 ع.6 - Con	1.3 - Optimal 1.4 - Result: 1.5 - Discus: 1.5 clusions	lity Conditions	ons		106 106 108 117		ſ
	•	5.5 5.5 5.5 £.6 - Con	1.3 - Optimal 1.4 - Result: 1.5 - Discus: 1.5 clusions	lity Conditions	ons		106 106 108		ſ
) , - 7		5.5 5.5 5.6 - Con	1.3 - Optimal 1.4 - Result: 1.5 - Discus: 1.5 - Discus:	v	ons		106 106 108 117	•	ſ
7		5.5 5.5 5.5 5.6 - Con	1.3 - Optimal 1.4 - Result: 1.5 - Discuss Iclusions	v	ons	•	106 106 108 117		ſ
7		5.5 5.5 5.6 - Con	1.3 - Optimal 1.4 - Result: 1.5 - Discuss clusions	v	ons	•	106 106 108 117	•	ſ
, -	•	5.5 5.5 5.6 - Con	1.3 - Optimal 1.4 - Result: 1.5 - Discus: 1.5 clusions	v	ons	•	106 106 108 117		ſ
, - - 2		5.5 5.5 5.6 - Con	1.3 - Optimal 1.4 - Result: 1.5 - Discus: 1.5 clusions	v	ons	•	106 106 108 117		,

÷				
	CHAPTER VI -	DISCUSSION AND CONCLUSIONS	· 118 J	•
	APPENDIX A -	GRAZOR SEARCH PROGRAM FOR MINIMAX		
		OPTIMIZATION	120	7
, , ,		A.1 Introduction	120	le la companya de la comp
	(A.2 Nomenclature	120	•
		A.3 Program Description	1,22	
	•	A_4 Subprograms	126	•
	-	A.5 Comments	126	
		A.6 Discussion	127	
		A.7 Grazor Search Fortran Program Listing	128	
			,	<i></i>
	APPENDIX B	PROCRAM FOR INVESTIGATING MINIMAY	. –	
		OPTIMALITY CONDITIONS	145	
		P. L. Introduction	145	
			145	
	1	B.2 Program Description	145	,
	-	B.3 Required Subprograms	148	•
	. ,	B.4 Comments	148	
		B.5 Fortran Listing for MINIMAX Program	*150	
•	REFERENCES		164	t
	AUTHOR INDEX		173	
	,		• •	,
-			-	0
	1-			•
•		· · ·	•	•1
			•	•
	• .	V vi		
		· · · · · · · · · · · · · · · · · · ·	•	
		-		

LIST OF FIGURES

Figure	<u>نې</u> .	C	Page	
Fig. 3.1		Block diagram summarizing the computer program structure and illustrating the relative hierarchy of the subprograms.	23	
Fig. 3.2)	Mathematical flow diagram of subroutine GRAZOR $(a_0, a, \beta, \epsilon, \epsilon', n, \phi, \psi_i, k, k_r, n, a_r, U_{\phi 0}, TERM)$.	24	
Fig. 3.3		Mathematical flow diagram of subroutine SELEC $(\phi^0, \psi_1, \psi_m, k, B, n_r, \psi_m)$.	25	1 - - -
Fig. 3.4	۰ ۲	Mathematical flow diagram of subroutine GOLDEN $(\gamma^*, n, \phi, \phi, \Delta \phi, \psi_i, k, n, U_{\phi}, B_{\phi})$.	26	•
Fig. 3.5	~	Example illustrating how the grazor search strategy follows the narrow, path of discontinuous derivatives.	31	÷
Fig. 3.6	•	2-section 100 to 10 quarter-wave transmission- line transformer.	33	•
Fig. 4.1		3-section LC transformer problem. Optimum match- ing over a frequency range of 0.5-1.179 radians/ sec occurs at the following parameter values: $L_1=1.04088$, $C_2=0.979035$, $L_3=2.34044$, $C_4=0.780157$,	47	
,		$L_5 = 2.93714$, $C_6 = 0.346960$ and $U = \max_{1} \left[\rho(\check{\phi}, \psi_1) \right]$ = 0.075820.	•	
Fig. 4.2		3-section LC transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. Starting point: $L_1 = L_3 = L_5 = C_2 = C_4 = C_6 = 1$.	49-	
Fig. 4.3		The m-section resistively terminated cascade of transmission lines. Optimum matching over 100 pe cent band centred at 1 GHz for R=10 occurs for th following parameter values. 2-section: $l_1 = l_2 = l_q, Z_1 = 2.23605, Z_2 = 4.4721$	51 r-	
		3-section: $l_1 = l_2 = l_3 = l_q, Z_1 = 1.63471, Z_2 = 3.16228, Z_3 = 6.11729$		
•		1 = 7.49481 cm is the quarter-wavelength at cent g frequency.	re	

j:

vii

		-	
•			
			-
2 · · · · · · · · · · · · · · · · · · ·			
			y
▶ Figure		Page	
Fig. 4.4(a)	The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. l_1 , l_2 fixed at l_1 and impedances varied. Start- ing point $Z_1=1.0$, $Z_2=3.0$.	53	- - -
Fig. 4.4(b)	The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Natson method. Z_1 , Z_2 fixed at optimum values and lengths varied. Starting point $L_1/L_q=0.8$, $L_2/L_q=1.2$.	54	
Fig. 4.4(c)	The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. t_2 , z_2 fixed at optimum values and t_1 , z_1 varied. Starting point $t_1/t_q = 1.2$, $z_1 = 3.5$.	55	
Fig. 4.4(d)	The 2-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. All 4 parameters varied. Starting point $t_1/t_q = 1.2$, $t_2/t_q = 0.8$, $2_1 = 3.5$, $2_2 = 3.0$.	56 °	
Fig. 4.5(a)	The 3-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. t_1, t_2, t_3 fixed at t_1 and impedances varied. Starting point $Z_1 = 1.0$, $Z_2 = 3.46228$, $Z_3 = 10.0$.	57 Ø	, ,
Fig. 4.5(b)	The 3-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. All 6 parameters varied. Starting point $t_1/t_{=0}$. $t_2/t_{=1.2}$, $t_3/t_{=0.8}$, $Z_1=1.5$, $Z_2=3.0$, $Z_3=6.0.9$.	58 8,	•
Fig. 4.5(c)	The 3-section transformer problem. Solid points distinguish the grazor search algorithm from the algorithm based on the Osborne and Watson method. All 6 parameters varied. Starting point $t_1/t_1 = t_2/t_1 = t_3/t_1 = 1.0$, $Z_1 = 1.0$, $Z_2 = 3.16228$, $Z_3 = 10.0$	59	1 , •
Fig. 4.6	Problem 1. Cascaded transmission-line filter operating between $R_{c}(\omega) = R_{1}(\omega) = 377/\sqrt{1-(f_{c}/f)^{2}}$, where $f_{c}=2.077$ GHz and $t=1.5$ cm.	64 2	-
•	viii		
2		•	

ig. 4.7 ig. 4.7 ig. 4.8 ig. 4.9	Responses of response of initial once obtained by minimax response of Fig. 4.6 wid duced by th Optimal responses of and lengths values at so The initial obtained by and the opt	of the networ f Carlin and c. The least y Bandler and sponse was prinod. f the minimarian ith 0.4 dB prino he grazor set sponse for Prinpedances, var $528=25$, $Z_2=1$ for Problem s are allowed start and fight l response conversed	rk of Fig. Gupta (19) t 10th resid Sevioral roduced by x design of assband, in arch metho roblem 2 w ried. Opt 0.254=24, 2 when imp d to vary, nish are so	4.6. The 69) is the ponse was (1970). The the grazor f the network servicen lose d ith lengths imal parame $Z_3 = 4.842$ - edances are The parame hown in Tab	rk of s pro- fixed ters fixed eter 1e 4.5	Page 65 68 70 71		•	
gure ig. 4.7 ig. 4.8 ig. 4.9 ig. 4.10	Responses of response of initial one obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a t and in are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	of the networ f Carlin and e. The least y Bandler and sponse was pr hod. f the minimar ith 0.4 dB pr he grazor set sponse for Pr npedances var .528=25, Z ₂ = for Problem s are allowed start and fir l response co y Brancher	rk of Fig. Gupta (19 t 10th res d Sevioran roduced by x design o assband, in arch metho roblem 2 w ried. Opt 0.254=24, 2 when imp d to vary, nish are s orresponds	4.6. The 69) is the ponse was (1970). The the grazor f the network servicen loss d ith lengths imal parame $Z_3 = 4.842$. edances are The parame hown in Tab	rk of s pro- fixed ters fixed eter 1e 4.5	Page 65 68_ 70 71		•	
gure ig. 4.7 ig. 4.8 ig. 4.9	Responses of response of initial one obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a 1 and im are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	of the networ f Carlin and c. The least y Bandler and sponse was point ith 0.4 dB path he grazor set sponse for Print npedances, van $528=25$, $Z_2=1$ for Problem s are allowed start and fig 1 response converse	rk of Fig. Gupta (19) t 10th res d Sevioral roduced by x design o assband, in arch metho roblem 2 w ried. Opt 0.254=Z ₄ , 2 when imp d to vary, nish are s orresponds	4.6. The 69) is the ponse was (1970). The the grazor f the networ servicen loss d. ith lengths imal parame Z ₃ =4.842. edances are The param hown in Tab	rk of s pro- fixed ters fixed eter 1e 4.5	Page 65 68 70 71			
ig. 4.7 ig. 4.8 ig. 4.9 ig. 4.10	Responses of response of initial one obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a t and im are ⁹ : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	of the networ f Carlin and c. The least y Bandler and sponse was pr hod. f the miniman ith 0.4 dB pr he grazor set sponse for Pr npedances, van .528=25, Z ₂ = for Problem s are allowed start and fin l response co y Brancher	rk of Fig. Gupta (19) t 10th res d Sevioran roduced by x design o assband, in arch metho roblem 2 w ried. Opt 0.254=24, 2 when imp d to vary, nish are s orresponds	4.6. The 69) is the ponse was (1970). The the grazor f the network servion lose d ith lengths imal parame $Z_3 = 4.84 \frac{2}{4}$ edances are The parame hown in Tab	rk of s pro- fixed ters fixed eter 1e 4.5	Page 65 68 70 71		•	
ig. 4.7 ig. 4.8 ig. 4.9 ig. 4.10	Responses of response of initial one obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a t and in are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	of the networ f Carlin and c. The least y Bandler and sponse was pond. f the miniman ith 0.4 dB path he grazor set sponse for Pr npedances van .528=25, Z ₂ = for Problem s are allowed start and fin 1 response converse	rk of Fig. Gupta (19 t 10th res d Sevioral roduced by x design o assband, in arch metho roblem 2 w ried. Opt 0.254=24, 2 when imp d to vary, nish are s orresponds	4.6. The 69) is the ponse was (1970). The the grazor f the network servicen losed d ith lengths imal parame $Z_3 = 4.842$ edances are The parame hown in Tab	rk of s pro- fixed ters fixed eter 1e 4.5	Page 65 68_ 70 71		•	
ig. 4.7 ig. 4.8 ig. 4.9	Responses of response of initial one obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a t and im are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	of the networ f Carlin and e. The least y Bandler and sponse was pr hod. f the minima: ith 0.4 dB pr he grazor set sponse for Pr npedances, var .528=25, 22=1 for Problem s are allowed start and fig 1 response co	rk of Fig. Gupta (19) t 10th res Sevioral roduced by c design o assband, in arch metho roblem 2 w ried. Opt 0.254=Z ₄ , 2 when imp d to vary, nish are s orresponds	4.6. The 69 is the ponse was (1970). The the grazor f the network servicen losed d.F ith lengths imal parame $Z_3=4.842$. redances are The parame hown in Tab	rk of s pro- fixed ters fixed eter le 4.5	Page 65 68 70 71			
ig. 4.7 ig. 4.8 ig. 4.9 ig. 4.10	Responses of response of initial one obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a t and in are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	of the netwo: f Carlin and c. The least y Bandler and sponse was p: hod. f the minima: ith 0.4 dB p: he grazor set sponse for Pri npedances.va: $528=25, Z_2=1$ for Problem s are allowed start and fight 1 response conversed	rk of Fig. Gupta (19) t 10th resid Sevioral roduced by c design o assband, in arch metho roblem 2 w ried. Opt 0.254=24, 2 when imp d to vary, nish are s	4.6. The 69) is the ponse was (1970). The the grazor f the networ servicen lose d.p ith lengths imal parame $Z_3 = 4.842$. edances are The parame	rk of s pro- fixed ters fixed eter le 4.5	65 68 70 71		•	
ig. 4.7 ig. 4.8 ig. 4.9 ig. 4.10	Responses of response of initial one obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a t and in are ⁹ : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	of the networ f Carlin and c. The leas y Bandler and sponse was prode hod. f the minimar ith 0.4 dB prode he grazor set sponse for Proper npedances var .528=25, Z2=1 for Problem s are allowed start and fir 1 response converted	rk of Fig. Gupta (19 t 10th res d Sevioral roduced by x design o assband, in arch metho roblem 2 w ried. Opt 0.254=24, 2 when imp d to vary, nish are s orresponds	4.6. The 69) is the ponse was (1970). The the grazor f the network servicen losed d ith lengths imal parame $Z_3 = 4.842$ edances are The parame hown in Tab	rk of s pro- fixed ters fixed eter le 4.5	65 68 70 71		•	
ig. 4.8 ig. 4.9 ig. 4.10	response of initial onc obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a t and im are ^q : Z ₁ =2. Responses f and lengths values at s The initial obtained by and the opt	f Carlin and c. The leasy y Bandler any sponse was pro- hod. f the minimar ith 0.4 dB pro- he grazor set sponse for Pro- npedances, var .528=25, Z2=1 for Problem s are allowed start and fir 1 response conversed	Gupta (19) Gupta (19) t 10th resid Sevioral roduced by x design of assband, in arch methon roblem 2 wr roblem 2 wr roblem 2 wr arch methon collem 2 wr arch methon to vary, nish are so	69) is the ponse was (1970). The the grazor f the netwo servicen loss d. ith lengths imal parame Z ₃ =4.842. edances are The param	rk of s pro- fixed ters fixed eter 1e 4.5	68 70 71			-
ig. 4.8 ig. 4.9 ig. 4.10	initial one obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a t and in are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	c. The leasy y Bandler any sponse was prode hod. f the minimary ith 0.4 dB prode he grazor set sponse for Prophem sponse for Prophem s are allowed start and fir l response converse y Brancher	t 10th resid Sevioral roduced by assband, in arch metho roblem 2 w ried. Opt 0.254=24, 2 when imp d to vary, nish are s orresponds	ponse was (1970). The the grazor f the networ servicen loss d.p ith lengths imal parame $Z_3 = 4.842$. edantes are The param hown in Tab	rk of s pro- fixed ters fixed eter le 4.5	68_ 70 71		•	
ig. 4.8 ig. 4.9 ig. 4.10	obtained by minimax res search meth Response of Fig. 4.6 wi duced by th Optimal res a t and in are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	y Bandler and sponse was p: hod. f the minima: ith 0.4 dB' pa he grazor set sponse for Pr npedances van .528=Z ₅ , Z ₂ = for Problem s are allowed start and fin 1 response co y Brancher	d Seviorai roduced by x design o assband, in arch metho roblem 2 w ried. Opt 0.254=24, 2 when imp d to vary, nish are s orresponds	(1970). The the grazor f the network servicen loss d.= ith lengths imal parame Z_3 =4.842. edances are the parame hown in Tab	rk of s pro- fixed ters fixed eter le 4.5	68_ 70 71		•	
1g. 4.8 ig. 4.9 ig. 4.10	 minimax ressearch meth Response of Fig. 4.6 widuced by th Optimal ressat and im are^q: Z₁=2. Responses fand lengths values at s The initial obtained by and the optimal solution 	sponse was p hod. f the minima: ith 0.4 dB p he grazor se sponse for P npedances, van .528=25, Z2=1 for Problem s are allowed start and fin 1 response co y Brancher	roduced by x design o assband, in arch metho roblem 2 w ried. Opt 0.254=2 ₄ , 2 when imp d to vary, nish are s orresponds	the grazor f the networs servion loss d ith lengths imal parame Z ₃ =4.842 edances are The param hown in Tab	rk of s pro- fixed ters fixed eter le 4.5	68_ 70 71	•		
ig. 4.8 ig. 4.9	search meth Response of Fig. 4.6 wi duced by th Optimal res a t and in are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	hod. f the minima: ith 0.4 dB p he grazor se sponse for P npedances,va .528=Z ₅ , Z ₂ = for Problem s are allowed start and fig 1 response co y Brancher	x design o assband, in arch metho roblem 2 w ried. Opt 0.254=2 ₄ , 2 when imp d to vary, nish are s orresponds	f the networ service loss d. ith lengths imal parame 23=4.842- edances are The param hown in Tab	rk of s pro- fixed ters fixed eter le 4.5	68_ 70 71			
ig. 4.8 ig. 4.9 ig. 4.10	Response of Fig. 4.6 wi duced by th Optimal res a t and in are ⁹ : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt	f the minima: ith 0.4 dB'p; he grazor se: sponse for P: mpedances,va: .528=Z ₅ , Z ₂ = for Problem s are allowed start and fig 1 response co y Brancher	x design o assband, in arch metho roblem 2 w ried. Opt 0.254=2 ₄ , 2 when imp d to vary, nish are s orresponds	f the networ service loss d. ith lengths imal parame Z ₃ =4.842 edances are The param hown in Tab	rk of s pro- fixed ters fixed eter le 4.5	68_ 70 71	· · · ·	•	
ig. 4.9 ig. 4.10	Response of Fig. 4.6 wi duced by th Optimal res a t and in are ^q : Z ₁ =2. Responses f and lengths values at s The initial obtained by and the opt timal solut	ith 0.4 dB' p he grazor se sponse for P: mpedances,va: .528=Z ₅ , Z ₂ = for Problem s are allowed start and fig 1 response co y Brancher	assband, in arch metho roblem 2 w ried. Opt 0.254=24, 2 when imp d to vary, nish are s orresponds	ith lengths imal parame 23=4.842 edances are The param	fixed fixed fixed eter le 4.5	70	· · · · · · · · · · · · · · · · · · ·	•	
ig. 4.9 ig. 4.10	duced by the Optimal rest at and integrated and integrated and integrated and integrated and integrated and lengths values at the initial obtained by and the optimal solution	he grazor se sponse for P: mpedances,va: .528=Z ₅ , Z ₂ = for Problem s are allowed start and fig 1 response co v Brancher	arch metho roblem 2 w ried. Opt 0.254=2 ₄ , 2 when imp d to vary, nish are s orresponds	d. ith lengths imal parame Z ₃ =4.842 edances are The param hown in Tab	fixed ters fixed eter le 4.5.	70 71	• •	•	
ig. 4.9 ig. 4.10	Optimal res a t and im are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt timal solut	sponse for P mpedances,va: .528=Z ₅ , Z ₂ = for Problem s are allowed start and fig 1 response co y Brancher	roblem 2 w ried. Opt 0.254=2 ₄ , 2 when imp d to vary, nish are s orresponds	ith lengths imal parame Z ₃ =4.842 edances are The param hown in Tab	fixed ters fixed eter le 4.5	70 71		•	
ig. 4.9 ig. 4.10	Optimal res a t and in are ^q : $Z_1=2$. Responses f and lengths values at s The initial obtained by and the opt timal solut	sponse for P mpedances,va: $528=2_5$, $Z_2=$ for Problem s are allowed start and fig 1 response co y Brancher	roblem 2 w ried. Opt 0.254=2 ₄ , 2 when imp d to vary, nish are s orresponds	ith lengths imal parame 23=4.842 edances are The param hown in Tab	fixed ters fixed eter le 4.5	70 71		•	
ig. 4.10	a t and in are ^q : Z ₁ =2. Responses f and lengths values at s The initial obtained by and the opt timal solut	mpedances,va: .528=25, Z ₂ = for Problem s are allowed start and fig 1 response co y Brancher	ried. Opt 0.254=2 ₄ , 2 when imp d to vary, nish are s orresponds	imal parame Z ₃ =4.842 edances are The param hown in Tab	fixed eter 1e 4.5	71		•	
ig. 4.10	are ⁻ : Z ₁ =2. Responses f and lengths values at s The initial obtained by and the opt timal solut	for Problem s are allowed start and fig 1 response co y Brancher	U.254=2 ₄ , 2 when imp d to vary, nish are s orresponds	Z ₃ =4.842 edances are The param hown in Tab	fixed eter le 4.5	; 71			
ig. 4.10	Responses f and lengths values at f The initial obtained by and the opt timal solut	for Problem s are allowed start and find l response converse y Brancher	2 when imp d to vary, nish are s orresponds	edances are The param hown in Tab	fixed eter le 4.5.	71			
* P • ••10	The initial obtained by and the opt	s are allowed start and fig l response of y Brancher	d to vary, nish are s orresponds	The param hown in Tab	eter 1e 4.5.				
	values at s The initial obtained by and the opt timal solut	start and fi l response co v Brancher	nish are s orresponds	hown in Tab	1e 4.5			• .	
•	The initial obtained by and the opt timal solut	l response c v Brancher	orresponds				-		
	obtained by and the opt timal solut	v Brancher		to best re	sults	-			
	and the opt timal solut	/	Maffioli	and Premoli	(1970)	•			
•	timal solut	timized resp	onse corre	sponds to t	he op-				
		tion obtaine	d by the g	razor searc	h	÷			
-	method.			•		r '		-	
ig. 5.1(a)	Seventh-or	der system m	odelling e	xample. 1-2-		83			
0··(")	parameter (optimum resp	onse.	· · · · · · · · · · · · · · · · · · ·					•
	-	· · ·		-		_			
ig. 5.1(b)	Seventh-or	der system m	odelling e	example. 2-		84			
	parameter, (optimum erro	r curve.	a t					
10 - 5 - 2(n)	Seventh-OT	der system m	delling e	Tample 3.	·	87			1
18. o. c (a)	narameter (optimum resp	ouerring e	campie. 54	-	.		-	
•	Par time rer .		*						
ig. 5.2(b) *	Seventh-or	der system m	odelling e	example. 3-		88			
`	parameter	optimum erro	r curve.						۹
· · · · · ·						01			
1g. 5.3(a)	Seventh-or	der system m	odelling e	example. 5-		AT			
>	parameter	ary-tribie 0	PLIMUM TCS	shause.		•			
ie. 5.3(b)	Seventh-or	der system m	odelling e	example. 5-		92		•	
-01. (9)	parameter	six-ripple o	ptimum ern	for curve.				· .	
	•	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~			-	•			
ig. 5.4(a)	-Seventh-or	der system m	odelling e	example. 5-	• •	9 5			
	parameter	rive-rippie	solution i	response.		N .			
ia 5 1(K)	Soventiant	der system a	odalling 4	example. 5.	•	96	۲.		
*B** 0.14 (11)	narameter	five-ripple	solution (error curve.					
•	hardmeter	rr-v							
			`		-				
		·		-					
		ix		-				2	
		ix		-					_
	ig. $5.2(a)$ ig. $5.2(b)$ ig. $5.3(a)$ ig. $5.3(b)$ ig. $5.4(a)$ ig. $5.4(b)$	 ig. 5.2(a) Seventh-or parameter ig. 5.2(b) Seventh-or parameter ig. 5.3(a) Seventh-or parameter ig. 5.3(b) Seventh-or parameter ig. 5.4(a) Seventh-or parameter ig. 5.4(b) Seventh-or parameter 	 ig. 5.2(a) Seventh-order system m parameter optimum resp ig. 5.2(b) Seventh-order system m parameter optimum error ig. 5.3(a) Seventh-order system m parameter six-ripple of seventh-order system m parameter six-ripple of ig. 5.4(a) Seventh-order system m parameter five-ripple ig. 5.4(b) Seventh-order system m parameter five-ripple 	 ig. 5.2(a) Seventh-order system modelling e parameter optimum response ig. 5.2(b) Seventh-order system modelling e parameter optimum error curve. ig. 5.3(a) Seventh-order system modelling e parameter six-ripple optimum response ig. 5.3(b) Seventh-order system modelling e parameter six-ripple optimum error system modelling e parameter six-ripple optimum error system modelling e parameter five-ripple solution i parameter five-ripple soluti parameter five-ri	 ig. 5.2(a) Seventh-order system modelling example. 3-parameter optimum response ig. 5.2(b) Seventh-order system modelling example. 3-parameter optimum error curve. ig. 5.3(a) Seventh-order system modelling example. 5-parameter six-ripple optimum response. ig. 5.3(b) Seventh-order system modelling example. 5-parameter six-ripple optimum error curve. ig. 5.4(a) Seventh-order system modelling example. 5-parameter five-ripple solution response. ig. 5.4(b) Seventh-order system modelling example. 5-parameter five-ripple solution response. 	 ig. 5.2(a) Seventh-order system modelling example. 3-parameter optimum response ig. 5.2(b) Seventh-order system modelling example. 3-parameter optimum error curve. ig. 5.3(a) Seventh-order system modelling example. 5-parameter six-ripple optimum error curve. ig. 5.3(b) Seventh-order system modelling example. 5-parameter six-ripple optimum error curve. ig. 5.4(a) Seventh-order system modelling example. 5-parameter five-ripple solution response. ig. 5.4(b) Seventh-order system modelling example. 5-parameter five-ripple solution response. 	ig. 5.2(a)Seventh-order system modelling example. 3- parameter optimum response87ig. 5.2(b)Seventh-order system modelling example. 3- parameter optimum error curve.88ig. 5.3(a)Seventh-order system modelling example. 5- parameter six-ripple optimum response.91ig. 5.3(b)Seventh-order system modelling example. 5- parameter six-ripple optimum error curve.92ig. 5.4(a)Seventh-order system modelling example. 5- parameter five-ripple solution response.95ig. 5.4(b)Seventh-order system modelling example. 5- parameter five-ripple solution response.96ig. 5.4(b)Seventh-order system modelling example. 5- parameter five-ripple solution error curve.96	ig. 5.2(a)Seventh-order system modelling example. 3- parameter optimum response87ig. 5.2(b)Seventh-order system modelling example. 3- parameter optimum error curve.88ig. 5.3(a)Seventh-order system modelling example. 5- parameter six-ripple optimum response.91ig. 5.3(b)Seventh-order system modelling example. 5- parameter six-ripple optimum error curve.92ig. 5.4(a)Seventh-order system modelling example. 5- parameter five-ripple solution response.95ig. 5.4(b)Seventh-order system modelling example. 5- parameter five-ripple solution error curve.96ig. 5.4(b)Seventh-order system modelling example. 5- parameter five-ripple solution error curve.96	ig. 5.2(a)Seventh-order system modelling example. 3- parameter optimum response87ig. 5.2(b)Seventh-order system modelling example. 3- parameter optimum error curve.88ig. 5.3(a)Seventh-order system modelling example. 5- parameter six-ripple optimum response.91ig. 5.3(b)Seventh-order system modelling example. 5- parameter six-ripple optimum error curve.92ig. 5.4(a)Seventh-order system modelling example. 5- parameter five-ripple solution response.95ig. 5.4(b)Seventh-order system modelling example. 5- parameter five-ripple solution response.96

		м,	
	*		
•	•		
Figure	· · · · ·	Page	
Fig. 5.5(a)	Seventh-order system modelling example. Optimal responses for a second-order model with no zeros.	110	
Fig. 5.5(b)	Seventh-order system modelling example. Optimal error curves for a second-order model with no zeros.	111	
Fig. 5.6(a)	Seventh-order system modelling example. Optimal responses for a second-order model with one zero.	112	
Fig. 5.6(b)	Seventh-order system modelling example. Optimal error curves for a second-order model with one zero.	113	
Fig. 5.7(a)	Seventh-order system modelling example. Optimal responses for a third-order model with two zeros.	114 .	
Fig. 5.7(b)	Seventh-order system modelling example. Optimal error responses for a third-order model with two zeros.	115	
Fig. A.1	Typical main program and analysis program for the grazor search package.	124	Y
Fig. A.2	(a) Typical printout if IDATA is .TRUE. (b) Typical printout if IPRINT is .TRUE.	125	
Fig. B.1	(Typical printout of results for the problem given in the text.	149	

X

· ·

CHAPTER I'

INTRODUCTION

Computer-aided design is now increasingly being accepted as a valuable tool whenever classical design techniques fail to achieve acceptable and realistic design criteria. This is especially true in electrical network analysis and synthesis where classical circuit theory restricts the network configuration and the degrees of freedom that may be demanded by the designer. Computer-aided network design has thus become a state-of-art which tries to accommodate the design specifications and constraints in a meaningful way so that design objectives, which would have been considered difficult by classical designers have now not only become feasible but are regularly being implemented on the digital computer. Many optimization algorithms have now been tested on a number of circuit design problems with the aim of improving circuit performance and convergence towards an optimal solution. The algorithms differ both in the way they generate downhill directions (directions of decreasing objective function value) and the computational effort involved.

It is thus apparent that there are two steps which are relevant to the circuit designer - the first one being that the design specifications, constraints involving the model parameters, and the objective function, have to be explicitly specified in advance, and the other being that a reliable and efficient algorithm has to be chosen for the optimization of the design variables. The exphasis of this work has been to bring both the system modelling and optimization techniques into the

foreground so that the advantages and pitfalls encountered in the area of computer-aided design can be well appreciated.

This thesis concentrates mainly on minimax objectives, and Chapter II gives a brief review of existing minimax optimization methods, such as those by Osborne and Watson (1969), Bandler and Macdonald (1969b), and Bandler and Charalambous (1972d).

A new algorithm called the <u>grazor search</u> method has been developed which is guaranteed to converge under certain conditions. See Bandler and Srinivasan (1971) and Bandler, Srinivasan and Charalambous (1972). The problem of function minimization subject to constraints can now be formulated as a minimax problem (Bandler and Charalambous 1972a). This approach can be extended to tackle minimax optimization problems subject to constraints (Bandler and Srinivasan 1973a). Once a minimax solution has been achieved by the systems designer, it may be required to investigate the solution for optimality, and suitable methods are available for this investigation (Bandler and Srinivasan 1973c). Chapter III considers the above mentioned approaches to the minimax problem.

Chapter IV deals with the area of computer-aided electrical circuit design for minimax objectives. The problems considered include the design of lumped LC transformers and cascaded transmission-line networks acting as transformers or filters. A critical comparison has been made between the grazor search method and other optimization schemes for reliability and efficiency in convergence towards the optima.

System modelling is an area which demands attention primarily because of the complexity and computational effort involved when

considering the original system, and the introduction of judíciously chosen models can not only reduce the complexity but also improve the computation time. It is now possible to model a high-order system and control this system on-line or off-line by dealing with the lower-order models_directly. Chapter V deals with lower-order modelling of highorder systems for a variety of objectives and design considerations. Minimax objectives subject to arbitrary transient and steady-state constraints have been considered, and a method suggested by means of which the whole modelling procedure can be automated. See Bandler, Markettos and Srinivasan (1972, 1973), and Bandler and Srinivasan (1973b, 1973e).

Discussions and conclusions on the proposed methods are included in Chapter VI, while the Appendices A and B provide two computer program descriptions for minimax objectives (Bandler and Srinivasan 1972, 1973d).

The adjoint network method of evaluating the first-order derivatives was used for network design problems (Director and Rohrer 1969, Bandler and Seviora 1970). The CDC 6400 computer was used for the numerical experiments.

The purpose of this work can be described as an attempt to fill some of the gaps existing in the areas of approximation, optimization and system modelling.

REVIEW OF MINIMAX METHODS

CHAPTER II

J

2.1 Introduction

Minimax optimization methods are assuming significance in the computer-aided system design area and much effort has gone into the development of suitable algorithms for minimax objectives. The methods have been used to optimize electrical networks where the objective is to minimize the maximum deviation of a network response from an ideal response specification. This chapter gives a brief review of minimax optimization techniques.

2.2 Function Minimization

The problem of unconstrained function minimization consists of minimizing with respect to ϕ a real function

f & f()

(2.1)

where

 $\stackrel{\bullet}{\underset{\sim}{\triangleq}} \stackrel{\bullet}{\underset{\sim}{\triangleq}} \stackrel{\bullet}{\underset{\sim}{1}} \stackrel{\bullet}{\underset{\sim}{2}} \cdots \stackrel{\bullet}{\underset{k}{}} \stackrel{T}{\underset{\sim}{}} \\$

(2.2)

(2.3)

is a column vector consisting of k independent parameter elements, T denotes the matrix transpose and f is the objective function.

The constrained version of the above problem, also known as the nonlinear programming problem, consists of minimizing $f(\phi)$ subject to

i = 1,2, ..., =

g₁(♦) ≥`0

where the g_i are, in general, nonlinear functions of the parameters.

2.3 Least pth Approximation for Single Specified Function

2.3.1 The Error Function

Define

$$o(\phi, \psi) \triangleq w(\psi) (F(\phi, \psi) - S(\psi))$$

where

 $S(\psi)$ is a specified function (real or complex)

 $F(\phi, \psi)$ is an approximating function (real or complex)

 $w(\psi)$ is a positive weighting function

 $e(\phi,\psi)$ is the weighted error or deviation between

S (ψ) and F(ϕ , ψ)

is an independent variable (e.g., frequency or time)

(2.4)

2.3.2 Continuous Approximation

Define the norm

$$||\mathbf{o}||_{p} \triangleq \left(\int_{\psi_{p}}^{\psi_{u}} |\mathbf{o}(\phi, \psi)|^{p} d\psi \right)^{1/p} \cdot 1 \leq p < \infty$$

$$(2.5)$$

where ψ_{\pm} and ψ_{\pm} are lower and upper bounds, respectively, on the interval of approximation. Minimization of $||e||_p$ is called least pth approximation. For p = 2, we have the well-known least squares approximation.

Assume, for example, that $|e(\phi, \phi)|$ is continuous on a finite closed interval $[\phi_{g}, \phi_{u}]$. The Chebyshev or uniform norm is given by

 $\| \bullet \|_{\infty} \stackrel{\text{frank}}{=} \| \bullet (\phi, \phi) \|$ (2.6)

The process of minimization of $||e||_{a}$ is called minimax or Chebyshev approximation.

It may be noted that

$$||\mathbf{e}||_{\infty} = \lim_{p \to \infty} \left\{ \frac{1}{\psi_{u} - \psi_{g}} \int_{\psi_{g}}^{\psi_{u}} |\mathbf{e}(\phi, \psi)|^{p} d\psi \right\}^{1/p}$$
(2.7)

The larger the value of p, the more emphasis will be given to the maximum absolute error, and the optimal least pth solution should be closer to the optimal minimax solution.

2.3.3 Discrete Approximation

In practice the various functions contained in (2.4) are usually evaluated at discrete values ψ_1 . It is thus appropriate to consider discrete approximation.

Define the norm

$$\left\| \bullet \right\|_{p} \triangleq \left(\begin{array}{c} \Sigma \\ i \in I \end{array} \right)^{p} \left(\begin{array}{c} 1 \\ \gamma \end{array} \right)^{1/p} \quad 1 \leq p^{< \bullet} \qquad (2.8)$$

where

$$(\phi) \triangleq \left[e_1(\phi) & e_2(\phi) & \dots & e_n(\phi) \right]^T$$

$$(2.9)$$

and

The process of minimization of ||e||_p is called discrete least pth approximation. The discrete minimax norm may be defined as (2.11),

ici ~

6

(2.10)

and minimization of $||e||_{\infty}$ is called discrete minimax approximation. As mentioned earlier,

 $\frac{||\mathbf{e}||_{\infty} = \lim_{p \to \infty} ||\mathbf{e}||_{p}}{\gamma}$

and the same comments hold as in the continuous case.

For a sufficiently large number of uniformly sampled values of ψ and with suitable weighting factors, the discrete approximation approach- \langle es the continuous approximation.

2.4 The Minimax Problem

Unless otherwise mentioned, the unconstrained discrete nonlinear minimax problem that is considered throughout this work consists of minimizing

 $U(\phi) \stackrel{\wedge}{=} \max_{i \in I} y_i(\phi)$

(2.13)

(2.12)

where I, as defined in (2.10), is an index set relating to discrete elements corresponding to the i, and the y_i are, in general, `nonlinear differentiable functions. It is desired to find a point ϕ such that

(2.14)

 $\check{U} \triangleq U(\check{\phi}) = \min \max_{n \neq i \in I} y_i(\phi)$, $\sim \phi \quad i \in I$

where ϕ is a local or global minimax optimum.

2.5 Minimax Methods

Many methods use the direct minimax formulation of (2.13) which,

in general, gives rise to discontinuous partial derivatives of the objective function with respect to the variable parameters. Otherwise efficient optimization methods may slow down or even fail to reach an optimum in such circumstances, particularly when the response hypersurface has a narrow curved valley along which the path of discontinuous derivatives lies.

In direct search strategies, the minimax problem has been explored using pattern search and razor search (Bandler and Macdonald 1969a, 1969b). Of the gradient strategies, there are methods involving the penalty function approach (Fiacco and McCormick 1964a, 1964b), linear programming (Osborne and Watson 1969, Ishizaki and Watanabe 1968), quadratic programming (Heller 1969), and a method proposed by Bandler and Lee-Chan (1971).

Whenever efficient methods of finding derivatives are not available, direct search methods are useful. For electrical networks, in particular, it is now possible to evaluate the derivatives of network responses with respect to network parameters rather easily using the adjoint network approach (Director and Rohrer 1969, Bandler and Seviora / 1970), and the gradient methods are thus more suited for such cases. The quadratic programming methods are usually more time-consuming than solution of linear programming problems, while penalty function methods rely on suitable function minimization algorithms.

2.5.1 The Razor Search Method

The razor search method of Bandler and Macdonald (1969b, 1971)

essentially begins with a modified version of the pattern search (Hooke and Jeeves 1961) until this fails. A random point is selected automatically in the neighbourhood and a second pattern search is initiated until this one fails. Using the two points where pattern search failed, a new pattern in the direction of the optimum is established and a pattern search strategy resumed until it too fails. This process is repeated until any of several possible terminating criteria is satisfied. Thus, the strategy tries to negotiate certain kinds of "razor sharp" valleys in multidimensional space. The method has been compared with other direct search methods on some test problems, and has been found to be reliable and computationally efficient in most of

the cases.

2.5.2 Sequential Unconstrained Minimization Technique

The nonlinear minimax optimization problem of Section 2/4 may be transformed into a nonlinear programming problem (Waren, Lasdon and Suchman 1967) of Section 2.2 as follows

Minimize .

(2.15)

(2.16)

subject to

 $\phi_{k+1} = y_1(\phi) \ge 0$

ieI

k+1

The nonlinear programming problem may, in turn, be solved by well-established methods such as the Sequential Unconstrained Minimization Technique (SUMT) due to Fiacco and McCormick (1964a, 1964b), which is a development of the Created Response Surface Technique (CRST) suggested by Carroll (1961). The problem of (2.15) and (2.16) may be reformulated as follows. Minimize

$$P(\phi,\phi_{k+1},\mathbf{r}) = \phi_{k+1} + \mathbf{r} \sum_{i \in I} \frac{w_i}{\phi_{k+1} - y_i(\phi)}$$
(2.17)

where

 ϕ_{k+1} is an independent variable, and r,w_i > 0 icI (2.18)

 $P(\phi, \phi_{k+1}, r)$ is an unconstrained objective where points close to the constraint boundaries are penalized.

Define the interior of the region of feasible points as

$$R^{0} \triangleq \{\phi, \phi_{k+1} | \phi_{k+1} - y_{i}(\phi) > 0, i \in I\}$$
 (2.19)

where the region of feasible points is

 $R \triangleq \{\phi, \phi_{k+1} | \phi_{k+1} = y_1(\phi) \ge 0, \quad i \in I\}$ (2.20)

Starting with a point $\phi_{,\phi_{k+1}}$ and a value of r, initially r_1 , such that $\phi_{,\phi_{k+1}} \in \mathbb{R}^0$ and $r_1 > 0$ the unconstrained function $P(\phi,\phi_{k+1},r_1)$ is minimized with respect to ϕ and ϕ_{k+1} . The form of (2.17) leads one to expect that a minimum will lie in \mathbb{R}^0 , since as any one of the $\phi_{k+1} - y_1(\phi)$ approaches 0, P approaches =. The location of the minimum will depend on the value of r_1 and is denoted by $\phi(r_m), \phi_{k+1}(r_1)$.

This procedure is repeated for a decreasing sequence of r values such that

(2.21)

(2.22)

$$r_1 > r_2 > \dots > r_j = 0$$

each minimization being started at the previous minimum. For example, the minimization of $P(\phi, \phi_{k+1}, r_2)$ would be started at $\phi(r_1)$ and $\phi_{k+1}(r_1)$. Every time r is reduced, the effect of the penalty is reduced, so that one would expect in the limit as j + = and $r_j + 0$ that $\phi(r_j) + \phi$ and, consequently, that $\phi_{k+1}(r_j) + U(\phi)$, the minimax optimum.

Conditions which guarantee convergence have been proved by Fiacco and McCormick. It is important that the initial value of r chosen is realistic, and r should be reduced systematically after each iterative cycle of minimization of P.

2.5.3 Algorithm due to Osborne and Watson

This minimax algorithm (Osborne and Matson 1969, Watson 1970). deals with minimax formulations by following two steps - a linear programming part that provides a given step in the parameter space, followed by a linear search along the direction of the step. This algorithm is very similar to the one proposed by Ishizaki and Watanabe (1968) and works very well for many minimax problems. In cases where the linear approximation is not very good in the vicinity of the optimum, the method may fail to converge toward the optimum for successive iterations.

Consider the problem of minimizing $||e(\phi)||_{in}$ in (2.11), where e consists of real elements. Linearizing $e_i(\phi)$ at some point ϕ^j the problem may be stated as

Minimize 🔶 🔶

subject to $\phi_{k+1} - \mathbf{e}_{i}(\phi^{j}) - \nabla^{T} \mathbf{e}_{i}(\phi^{j}) \Delta \phi^{j} \geq 0$ (2, 23)iel $\phi_{k+1} + e_{i}(\phi^{j}) + \nabla^{T} e_{i}(\phi^{j}) \Delta \phi^{j} \geq 0$ where $\Delta = \begin{bmatrix} \frac{9\phi^1}{9} & \frac{9\phi^2}{9} & \cdots & \frac{9\phi^r}{9} \end{bmatrix}_{\mathrm{L}}$ (2.24)(2:25) <u>n > k</u> ∇ is the first partial derivative operator with respect to the parameter vector \$, A denotes incremental changes, and n is the number of elements of I. Noting that the variables for linear programming should all be د م nonnegative, and imposing a rather practical constraint that the ele-_ ments of ϕ should not change sign we have the linear programming problem in (2.26) $\mathbf{x} \triangleq [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_{k+1}]^T$ as follows. Step 1 (2.27)Minimize x k+1 subject to (2.25) and $\pm \left(\mathbf{e}_{\mathbf{i}} \left(\mathbf{e}_{\mathbf{j}}^{\mathbf{j}} \right) + \nabla^{\mathsf{T}} \mathbf{e}_{\mathbf{i}} \left(\mathbf{e}_{\mathbf{j}}^{\mathbf{j}} \right) \right) = \left(\mathbf{e}_{\mathbf{i}}^{\mathbf{j}} \mathbf{x}_{1}^{\mathbf{j}} - \mathbf{e}_{1}^{\mathbf{j}} \mathbf{x}_{1}^{\mathbf{j}} - \mathbf{e}_{1}^{\mathbf{j}} \right) = \left(\mathbf{e}_{\mathbf{i}}^{\mathbf{j}} \mathbf{x}_{1}^{\mathbf{j}} - \mathbf{e}_{1}^{\mathbf{j}} \mathbf{x}_{1}^{\mathbf{j}} \right) = \left(\mathbf{e}_{\mathbf{i}}^{\mathbf{j}} \mathbf{x}_{1}^{\mathbf{j}} - \mathbf{e}_{1}^{\mathbf{j}} \mathbf{x}_{1}^{\mathbf{j}} \right)$ (2.28)i¢I

where

 $x \ge 0$

$$x_{\underline{k}} \stackrel{\Delta}{=} \frac{\Delta \phi_{\underline{k}}}{\phi_{\underline{k}}} + 1 \qquad \underline{k} = 1, 2, \dots, k \qquad (2.30)$$

The solution produces a direction given by $\Delta \phi^{J}$.

Step 2

Next we find y^{j^*} such that

 $\max_{i \in I} \left| e_i (\phi^j + \gamma^j) \phi^j \right|$

is a minimum with respect to y^j . Set

 $\phi^{j+1} = \phi^j + \gamma^{j*} \phi^{j}$

and return to Step 1.

The convergence of the method holds under certain conditions (Osborne and Watson 1969). This approach is directly applicable to linear functions such as polynomials, for which k+l equal extrema results at the optimum.

2.5.4 Method due to Bandler and Lee-Chan

The nonlinear minimax objective given by (2.13) is minimized here by exploiting the gradient information of the local discrete maxima of the functions $y_i(\phi)$ to get a downhill direction by solving a set of simultaneous equations. The method works very well, except that in the case of linear dependence of the equations, some problems may arise in the convergence toward is optimum. See Bandler and Lee-Chan (1971).

(2.29)

(2.31)

(2.32)

2.6 Near-Minimax Methods

by

As is well-known to network designers, least pth approximation for sufficiently large values of p can result in an optimal solution very close to the optimal minimax solution (Temes and Zai 1969, Temes 1969, Bandler 1969a, Seviora, Sablatash and Bandler 1970).

When appropriate error functions are raised to a power p given

 $\tilde{f}(\phi) = \sum_{i \in I} |e_i(\phi)|^p$ (2.33)

and $f(\phi)$ is minimized, ill-conditioning may result for nominal values of p (usually greater than or equal to about 10). The objective function of the form (2.33) has been used by a number of authors (Temes and Zai 1969, Temes 1969, Bandler 1969a, Bandler and Seviora 1970).

Bandler and Charalambous (1972c, 1972d) have given a unified approach to the Teast pth approximation problems, as encountered in network and system design, having upper and lower response specifications e.g., as in filter design. The ill-conditioning is removed by proper scaling, and least pth optimization has been carried out for extremely large values of p, typically 10³ to 10⁶. This approach has been used extensively in a variety of computer-aided network design problems (Bandler and Bardakjian 1973, Bandler and Charalambous 1972d, Bandler, Charalambous and Tam 1972, Bandler and Jha 1972, Popovic. 1972, Charalambous 1973).

The least pth approximation problem can effectively be tackled by efficient gradient minimization techniques such as the Fletcher -Powell method (1963), Jacobson - Oksman algorithm (1972), and a more recent method due to Fletcher (1970). These methods have been compared critically for near-minimax approximation problems in the area of lower-order modelling of high-order systems (Bandler, Markettos and Srinivasan 1972, 1973).

The discrete nonlinear minimax approximation problem of Section 2.4 can be formulated as a least pth approximation problem (Bandler 1972). Suppose at least one of the functions $y_i(\phi)$ is positive. Then, since $U(\phi) > 0$,

$$U(\phi) = \lim_{\tau \to \infty} U(\phi) \left\{ \sum_{i \in I} \left(\frac{w_i y_i(\phi)}{U(\phi)} \right)^p \right\}^{1/p}$$

where

 $\mathbf{w}_{i} = \begin{cases} 0 & \text{for } \mathbf{y}_{i} < 0 \\ 1 & \text{for } \mathbf{y}_{i} \ge 0 \end{cases}$

Suppose all the functions y_i are negative. Then, since $U(\phi) < 0$,

$$U(\phi) = \lim_{p \to -\infty} U(\phi) \left(\sum_{i \in I} \left(\frac{w_i y_i(\phi)}{U(\phi)} \right)^p \right)^{1/p}$$
(2.36)

where

 $x_i = 1$ for all $y_i < 0$ (2.37)

Therefore, the minimization function is chosen as

$$f(\phi) = U(\phi) \left[\sum_{\lambda} \left[\frac{w_{i}y_{i}(\phi)}{U(\phi)} \right]^{q} \right]^{1/q}$$
(2.38)

(2.34)

(2.35)

ð

where

A number of interesting features of $f(\phi)$ can be stated. For 1 < |q| < , q having the appropriate sign, and for appropriate values of w_i , in accordance with (2.35) for $U(\phi) > 0$ and (2.37) for $U(\phi) < 0$, we have a continuous function $f(\phi)$ with continuous derivatives with respect to ϕ so long as $U(\phi) \neq 0$. When $U(\phi) > 0$, $f(\phi)$ is like penalty term including violated constraints, in this case only positive y_i , which it is desired to make feasible (or acceptable). If min $f(\phi) > 0$, the constraints remain violated. In least pth approximation this indicates that the specifications have not been satisfied. When $U(\phi) < 0$ the specifications are satisfied and $f(\phi)$ is like a penalty term designed to move a solution as far from the boundary of the feasible region as possible.

16

(2.39)

CHAPTER III

NEW APPROACHES TO THE MINIMAX PROBLEM

3.1 Introduction

In this chapter a new gradient algorithm for minimax objectives called the grazor search (or gradient razor search) method is introduced (Bandler and Srinivasan 1971, Bandler, Srinivasan and Charalambous 1972). As the name suggests, the method attempts to follow the path of discontinuous derivatives when encountering razor-sharp valleys in multidimensional parameter space. The method is especially suitable for nonlinear minimax optimization of network and system responses. This algorithm uses the gradient information of one or more of the highest ripples in the error function to produce a downhill direction by solving a suitable linear programming problem. A linear search follows to find the minimum in that direction, and the procedure is repeated. This type of descent process is repeated with as many ripples as necessary until a minimax solution is reached to some desired accuracy. Unlike the razor search method due to Bandler and Macdonald (1969b), the present method overcomes the problem of discontinuous derivatives characteristic of minimax objectives without using random moves. It can fully exploit the advantages of the adjoint network method of evaluating partial derivatives of the response function with respect to the variable parameters (Director and Rohrer 1969, Bandler and Seviora 1970).

The problem of constrained minimax optimization is considered

next. This problem has been reformulated as an unconstrained minimax problem by two methods, one extending a recently proposed method due to Bandler and Charalambous (1972a, 1973b) and the other using weighting functions. The reformulated problem can then be tackled by efficient unconstrained minimax algorithms. The method has a number of applications, including high-order system modelling and control system designs, where constraints have to be imposed on the pole-zero locations of the models chosen. Appropriate constraints can also be imposed on the upper and lower bounds of the parameter values. See Bandler and Srinivasan (1973a, 1973e).

Investigation of optimality conditions of a proposed or a design solution is of great practical importance to the system designer wishing to approximate a desired response by a system response. Conditions for optimality in the minimax sense in conventional synthesis problems involving polynomials and rational functions are fairly widely appreciated. However, with the ever-increasing need for network designs containing elements not conducive to the rational function approach, e.g., a mixture of lumped and distributed elements, and the application of automatic optimization methods involving least pth and minimax objectives, some means of testing for convergence to an optimum for more arbitrary problems is highly desirable. Depending on the optimization method employed, a satisfactory minimax solution may be obtained for a problem after a number of iterations of the algorithm on the computer. It may then be required to investigate the solution for minimax optimality (Bandler 1971) so as to verify whether the solution is optimal or not. Though the necessary optimality conditions may seem to be straightforward

to verify, they are both tedious and difficult to implement in practice. A practical way of implementing them is considered in detail. See Bandler and Srinivasan (1973c, 1973d).

3.2 The Grazor Search Strategy

e.

3.2.1 Theoretical Considerations

The grazor search algorithm is a generalization of the method due to Bandler and Lee-Chan (1971), and is basically of the steepest descent type. The nonlinear minimax optimization problem is the one already stated in Section 2.4.

Define a subset JGI such that

$$J(\phi^{j},\epsilon^{j}) \triangleq \{i \mid U(\phi^{j}) - \gamma_{i}(\phi^{j}) \leq \epsilon^{j}, i\epsilon I\}$$
(3)

ε, > 0

where

 ϕ^{j} denotes a feasible point at the beginning of the jth iteration, and is the tolerance with respect to the current max $y_{i}(\phi^{j})$ within which the y_{i} for icJ lie. icI γ Linearizing y_{i} at ϕ^{j} , we can consider the first-order changes $\delta y_{i}(\phi^{j}) = \nabla^{T} y_{i}(\phi^{j}) \delta \phi^{j}$ icJ (ϕ^{j}, c^{j}) (3.3)

A sufficient condition for $\Delta \phi^{j}$ to define a descent direction

.1)

(3.2)



 $\nabla^{T} y_{i}(\phi^{j}) \Delta \phi^{j} < 0$ $i \in J(\phi^{j}, \varepsilon^{j})$ (3.4)

Consider $\Delta \phi^{j} = - \Sigma \alpha^{j} \nabla y_{i}(\phi^{j})$ $\sim i \epsilon J \gamma^{j} \gamma^$

> (3.6) $\sum_{i\in J} \alpha_i^{j} = 1$ (3.7) $a_i^j \ge 0$

(3:4) may now be written as

$$-\nabla^{T} \mathbf{y}_{i}(\phi^{j}) \Sigma \alpha_{i}^{j} \nabla \mathbf{y}_{i}(\phi^{j}) < 0 \qquad i \epsilon J(\phi^{j}, \epsilon^{j}) \qquad (3.8)$$

which suggests the linear program:

·

Maximize

$$a_{k_r+1}^{j}(\phi^{j},\epsilon^{j}) \geq 0$$
 (3.9)

subject to

.:

$$-\nabla^{T} y_{i}(\phi^{j}) \sum_{i \in J} \alpha_{i}^{j} \nabla y_{i}(\phi^{j}) \leq -\alpha_{k_{T}+1}^{j} \quad i \in J(\phi^{j}, e^{j})$$

$$(3.10)$$

plus: (3.6) and (3.7), where k_r denotes the number of elements of

$$(\phi^{J}, \varepsilon^{J})$$
. Note that if

$$\Delta \phi^{j} = 0 \qquad \text{for } s^{j} = 0$$

the necessary conditions for a minimax optimum are satisfied at ϕ^{j}

20

(3.5)

(Bandler 1971). Observe that J is non-empty and that if J has only one-element, we obtain the steepest descent direction for the corresponding maximum of the $y_i(\phi)$.

3.2.2 Proof of Convergence

Before proving the convergence of the algorithm it may be worth . restating the following lemma due to Farkas (Lasdon 1970).

Let $\{p_0, p_1, \dots, p_n\}$ be an arbitrary set of vectors. There exist $\beta_i \ge 0$ (3.11)

such that

 $p_0 = i \frac{\Sigma}{1} \beta_i \frac{\beta}{1} i_{\gamma}^{p_1}$

if and only if

 $p_0^T q \ge 0$ (3.13)

for all q satisfying

 $p_i^T q \ge 0$ i = 1, 2, ..., n (3.14)

It is, therefore, possible to find nonnegative values of a_j^j in the expression for (3.5) if and only if

(3.12)

(3.15).

for all $\Delta \phi^j$ satisfying

$$\nabla^{T} y_{i}(\phi^{j})(-\Delta \phi^{j}) \geq 0 \qquad i \varepsilon J(\phi^{j}, \varepsilon^{j}) \qquad (3.16)$$

where (3.13) and (3.14) correspond to (3.15) and (3.16), respectively, and $-\Delta\phi^{j}$, $\nabla y_{i}(\phi^{j})$, $-\Delta\phi^{j}$ take the place of p_{0} , p_{i} , q.

Now (3.15) is always satisfied, thrugh it may not be possible to satisfy (3.16) if ϵ^{j} is too large. By suitably decreasing ϵ^{j} , (3.16) may be forced to hold.

3.2.3 Practical Implementation

Fig. 3.1 illustrates how the different subroutines are called and their relative hierarchy. Flow charts of subroutines GRAZOR, SELEC and GOLDEN appear in Figs. 3.2 - 3.4. See Appendix A for further details and definitions. The objective function $U(\phi^{j})$ is calculated by subroutine LOCATE.

As given by linear programming (see, for example, Subroutine SIMPLE), $\Delta \phi^{j}$ is normalised to

$$\Delta \phi_{n}^{j} = \frac{\Delta \phi_{n}^{j}}{\frac{1}{2} \left[\Delta \phi_{n}^{j} \right] \left[\frac{1}{2} \right]}$$

(3.17)

by subroutine NORM. Starting at ϕ^j , a step $a^j \Delta \phi_n^j$ is taken for $a^j = a_0^j$; if no improvement in U results, a^j is reduced by factors of 8 until a better point is obtained or $a^j < a^j$. Let a^j^* produce the first improved point from ϕ^j . Then $\Delta \phi^0 = a^{j^*} \Delta \phi_n^j$ (3.18)







Fig. 3.2 ⁱ Mathematical flow diagram of subroutine

GRAZOR $(a_0, X, B, \varepsilon, \varepsilon', n, \phi^0, \psi_1, k)$, $(n, n_1, U_{\phi 0}, TERM)$

ž



. . .

٠,

•

.

ų.

7ete

٠.

•

Call WESDET(Toj-in Ay) to seet the roj is decreasing values and in identifies the sch highest of the ve

e-La,

5

ς,

ゝ

· _ _ _





.

•

Fig. 3.3 Mathematical flow diagram of subroutine

SELEC $(\phi^0, \psi_1, \hat{\psi}_m, k, n, n_T, \hat{y}_m)$


Ţ



GOLDEN $(\gamma^*, \eta, \phi, \phi^0, \Delta \phi^0, \psi_1, k, \eta, U_{\phi^0}, U_{\phi^0})$

, is defined.

Next a method based on golden section search (Temes 1969) is used to find γ^{j*} corresponding to the constrained minimum value of max $y_i(\phi^j + \gamma^j \Delta \phi^j)$. The jth iteration ends by setting icl

$$\phi^{j+1} = \phi^j + \gamma^{j*} \Delta \phi^0 \qquad (3.19)$$

and

up

$$a_{0'}^{j+1} = a^{j+1} \gamma^{j+1}$$
(3.20)

In Fig. 3.4,

$$\tau = \frac{1}{2} (1 + \sqrt{5})$$
 (3.21)

is the factor associated with the golden section. Subscripts t and u denote lower and upper limits, respectively, and a and b denote interior points of the interval of search. An attempt to bound the minimum is made. Then golden section search is used to locate the minimum to a desired accuracy. The search is terminated when the resolution between two interior points falls below a factor η of the initial interval.??

In Fig. 3.3 the maxima implied by the functions y_i, sampled in a certain order, are located and sorted out in decreasing magnitude (by, say, Subroutine TGSORT).

Fig. 3.2 shows the grazor search strategy. Note that in setting

Ax = b

generate a descent direction based on the gradient of the maximum

slack variables $(x_{k_r}^*+2, x_{k_r}^*+3, \dots, x_{2k_r}^*+1)$ are introduced. We try to

(3.22)

function $(k_r = 1)$, proceed to the minimum of U in that direction, and repeat the process. If, at any stage, this process or the linear program does not yield a direction of decreasing U, or does not provide an improvement greater than ϵ , the procedure is repeated after including the function corresponding to the next largest of the current n_r discrete local maxima (i.e., ripples) if one exists. When all local maxima have been included and U can still not be reduced or improved satisfactorily by a value greater than ϵ , we repeat the procedure with k_r functions corresponding to the first k_r largest of the candidates, beginning with $k_r = 1$, in another series of attempts to reduce U. The allocithm terminates only when there are no more suitable functions left and when there are either no improvements or improvements less than ϵ_r^1 over one complete cycle of k_r , starting from 1 and ending with n_r .

3.2.4 Example

The design of a two-section 100 to 10 quarter-wave transmissionline transformer network over a 100 percent bandwidth centred at 1GHz is considered (Matthaei, Young and Jones 1964) as an example for testing the grazor search strategy. This problem has already received attention from the optimization point of view (Bandler and Macdonald 1969a, 1969b). The lengths t_1, t_2 are fixed at t_q , the quarter-wavelength at centre frequency, and the impedances Z_1, Z_2 are varied.

Table 3.1, in association with Fig. 3.5, illustrates how the grazor search strategy effectively follows the path of discontinuous derivatives to locate the optimum in the course of minimax optimization

TABLE 3.1

SUMMARY OF IMPORTANT STEPS IN THE EXAMPLE ILLUSTRATING THE GRAZOR SEARCH STRATEGY

Iteration	Points of	Starting Point	Values o	of Scale Factors	
Number	lteration	or relation	Point	Scale Factor	K 1
<u> </u>		∳ ¹ =(1.0, 3.0)		· • · · · · · · · · · · · · · · · · · ·	-
		v	2 ²	a*=1.00	
· 1	1-5	U'(•1)=0.70954	2.ª	γ=1+τ	1
			* ⁵ =¢ ²	Y*=1.000	
		∳ ⁵ =(1.99996, 3.00893)			_
		•	2 ⁶	a=1.00	
2	5-12	U' (• ⁵)=0.63086	\$ ⁷	a*=0.10	.1
		, •	\$ ¹²	Y*=2+τ	
		♦ ¹² =(1.69865, 3.20921)			
		` v	2 ¹³	α=0.1(τ+2)	
			2 ¹⁴	α= 0.01(τ+2)	
3	12-20	U' (\$14)=0.48073	±15	α*=0.001(τ+2)	1
		· · · · · ·	¢20	γ*=τ+1	
- <u></u>		↓ ²⁰ =(1.70806, 3.20821)	·		
		•	\$ ²¹	a=9.472x10 ⁻³	
4	20-26	U' (\$ ²⁰)=0.47843	-+ ²²	a*=9.472x10 ⁻⁴	:
١		~	¢ ²⁶	γ*=1.000	
<u></u>		\$ ²⁶ =(1.70723, 3.20865)		· · · · · · · · · · · · · · · · · · ·	
	·	······································	\$ ³⁰	a*=1.0x10 ⁻⁶	
S	26-35		2 ³⁵	Υ*=τ+1	
		\$ ³⁵ =(1.70723, 3.20866)			
		W (435)-0 47704	2 ³⁶	a*=9.472x10 ⁻⁴	
6	.35-04		***	γ*=1.096x10 ³	

t

.

SUMMARY OF IMPORTANT. STEPS IN THE EXAMPLE ILLUSTRATING THE GRAZOR SEARCH STRATEGY

= (2,23605, 4,47210), U'() = 0.42857 Values of Scale Factors Starting Point Points of Iteration Iteration of Iteration k_T Number Point Scale Factor \$6⁶=(2.05489, 4.18669) **≜**64 64-72 U' (454)=0.44064 **7** 2 **γ*=τ+2** ∮⁷⁸ U¹(∮⁷²)=0.43199 8 72-78 Y*=1.000 2 \$⁷⁸=(2.09380, 4.17280) *⁹⁶ $U'(\phi^{78})=0.43146$ 2 78-96 Y*=60.69 9 ϕ^{96} =(2.18832, 4.38018) 1⁹⁸ a*=2.279x10-3 2 10 96-103 \$¹⁰³ U'(****)=0.42929 Υ***=1.000** 4103-(2.19040, 4.37924) U'((103)=0.42886 **117** 11 103-117 Y*=30.03 2 $\phi^{117}=(2.22029, 4.44082)$ 1²⁶ U'(4117)=0.42864 3 - γ*=10.47 12 117-126 $\phi^{126}=(2.23088, 4.46221)$ 13 126-132 U'(\$125)=0.42862 133 - 126 =2.279x10⁻⁵ ປາ(ອ¹³³) = ປ(ອ 2 -133-136 13 1 % '=**t+2** 4169=(2.23595, 4.47237) 2¹⁷⁶ U'(4169)=0.42861 Y *=1.000 3. 169-176 18 9

30

t





of the network (see Fig. 3.6). Let

$$y_{i}(\phi) = \frac{1}{2} |\rho(\phi, \psi_{i})|^{2}$$

and define

 $\begin{array}{c} U^{\dagger}(\phi) = \max \left| \rho(\phi, \psi_{\underline{i}}) \right| \\ \gamma & i \end{array}$

where $\phi = [Z_1 \ Z_2]^T$, and ρ is the reflection coefficient on 11 uniformly spaced frequencies ψ_i in the band 0.5-1.5 GHz.

The grazor search strategy starts at

$$\phi^{1} = [1.0 \quad 3.0]^{T}$$

 \sim
 $U^{*}(\phi^{1}) = 0.70954$

and the values of the parameters used are $a_0 = 1$ (at start), $\check{\alpha} = 10^{-6}$, $\beta = 10$, $\eta = 0.5$, $\varepsilon = 10^{-4}$ and $\varepsilon^{*} = 10^{-6}$.

The first iteration extends from ϕ^1 to ϕ^5 ; ϕ^2 is the new point obtained when taking a unit step along the direction suggested by the negative gradient. Since ϕ^2 is a satisfactory improvement, a golden section search is infliated, yielding $\phi^3(\gamma=1+\tau)$ which is not an improvement over ϕ^2_{μ} . The interval of search is thus found. $\phi^4(\gamma = \tau)$ is found to be no improvement over ϕ^2 . The golden section search is now terminated, since the current resolution between two interior points of search falls below the minimum allowed value. $\phi^5 = \phi^2$ is thus the best point attained at the end of iteration 1. At the end of iteration 5, $U(\phi^{26})-U(\phi^{35})<\epsilon$, so k_r is increased from 1 to 2 in the next

(3, 23)

(3.24)



Fig. 3.6 2-section 100 to 10 quarter-wave transmission-line transformer.



iteration. For a similar reason, k_r is increased from 2 to 3 for iteration 12, and reset to 1 from 3 for iteration 13. During iteration 18, the parameter values remain the same to 5 significant digits, and the improvement in U at the end is less than ϵ '; all successive attempts to achieve a better point with an improvement greater than ϵ ' (by considering 1, 2 and 3 ripples) fail, and the procedure is terminated.

3.3 Constrained Minimax Optimization

3.3.1 Statement of the Problem

The constrained minimax problem considered may be stated as follows.

. Minimize

$$U(\phi) = \max y_i(\phi)$$

$$\gamma_i(\phi) = i \epsilon I - \gamma_i(\phi)$$

subject to

$$g_j(\phi) \ge 0$$
 jeN (3.26)

.where

I
$$\Delta \{1, 2, ..., n\}$$
 (3.27)
M $\Delta \{1, 2, ..., m\}$ (3.28)

(see Sections 2.2 and 2.4)

It will be assumed that the functions y_i and g_j are continuous with continuous partial derivatives, and that the inequality constraints-(3.26) are such that a Kuhn-Tucker solution exists (Lasdon 1970, Zangwill 1969).

Let $\hat{y}_{\pm}(\phi)$ for £cL be the largest local discrete maxima (ripples)

34

(3.25)

 $L \triangleq \{1, 2, ..., n_r\}$

(3.29)

(3.30)

(3, 32)

ł

35

3.3.2 Formulation 1

The constrained minimax problem of (3.25) and (3.26) can be formulated as a non-linear programming problem as follows.

Minimize ϕ_{k+1} subject to (3.26) and

 $\phi_{k+1} - y_i(\phi) \ge 0 \qquad \text{icI} \qquad (3.31)$

The above problem can then be reformulated as an unconstrained minimax problem as follows.

Minimize with respect to ϕ and ϕ_{k+1}

$$\bigvee (\phi, \phi_{k+1}, \alpha) = \max_{\substack{i \in I \\ j \in \mathbb{N}}} \left[\phi_{k+1}, \phi_{k+1} - \alpha_1(\phi_{k+1} - y_1(\phi)), \phi_{k+1} - \alpha_{j+1}g_j(\phi) \right]$$

where

Ċ

$$\begin{bmatrix} \alpha & \Delta & [\alpha_1 & \alpha_2 & \dots & \alpha_{m+1}]^T \\ \ddots & & & & & \\ a_j > 0 & j = 1, 2, \dots, m+1 \end{bmatrix}$$
 (3.33
(3.34)

For a large enough value of a one can obtain, in principle, the

exact optimal solution for the original problem by minimizing this reformulated objective function.

2

When implementing this scheme one can, for the problem defined earlier, slightly modify the formulation in order to save on computational effort, so that the minimization function chosen is

$$\begin{array}{c} V'(\phi,\phi_{k+1},\alpha) & * \max \left[\phi_{k+1},\phi_{k+1}^{\dagger} - \alpha_{1}(\phi_{k+1},\hat{y}_{\ell}(\phi)), \\ j \in M \right] \\ \phi_{k+1} & - \alpha_{j+1} g_{j}(\phi) \\ \phi_{k+1} & - \alpha_{j+1} g_{j}(\phi) \\ \end{array} \right]$$
(3.35)

3.3.3 Formulation 2

In this formulation, weighting functions are used to convert the original problem into an unconstrained minimax problem as follows.

Minimize with respect to \$

$$N(\phi, w) = \max \left[y_{i}(\phi), -w_{j}g_{j}(\phi) \right]^{\prime}$$

$$\gamma \sim icI \qquad \gamma$$

$$icM \qquad (3.36)$$

where

٦.

. . . .

$$\mathbf{w} \triangleq \left[\mathbf{w}_{1} \ \mathbf{w}_{2} \ \cdots \ \mathbf{w}_{m}\right]^{\mathrm{T}} \qquad (3.37)$$

For purposes of practical implementation, as long as $U(\phi) > 0$ and one wishes to apply nonzero weights only to violated constraints of (3.26), the minimization function may be chosen as

$$f'(\phi, w') = \max \left[\hat{y}_{g}(\phi), -w'_{j} g_{j}(\phi) \right]$$

$$\gamma \neq teL \qquad \gamma \qquad jeN \qquad (3.39)$$

37

where

$$\mathbf{w}^{*} \triangleq \left[\mathbf{w}_{1}^{*} \mathbf{w}_{2}^{*} \cdots \mathbf{w}_{m}^{*}\right]^{\mathrm{T}}$$
(3.40)

j≘M

The advantage of this formulation is apparent when U > 0 implies that certain specifications are violated and U < 0 implies that they are satisfied. In this case, comparison with violated and satisfied constraints seems appropriate.

 $w_j' > 0$ for $g_j(\phi) < 0$

 $w_j' = 0$ for $g_j(\phi) \ge 0$

3.3.4 Comments

By proper choice of the elements of a, w, or w', the reformulated functions V, V', W or W' can be minimized by a suitable minimax or nearminimax algorithm. In case of parameter constraints, upper and lower specifications can be considered as follows.

 $\mathbf{g}_{2i-1}(\phi) = \phi_i - \phi_{ii} \ge 0$

(3.42) i = 1, 2, ... , k

$$\mathbf{s}_{2i}(\phi) = -(\phi_i - \phi_{iu}) \ge 0$$

(3.41)

 $g_{j}(\phi) \geq 0$ j = 2k+1, 2k+2, ...

(3.43)

3.4 Practical Investigation of Minimax Optimality Conditions

3.4.1 Introduction

In recent paper (Bandler 1971), the conditions for a minimax, optimum were derived for a general nonlinear minimax approximation problem from the Kuhn-Tucker (1950) conditions for a constrained optimum in nonlinear programming. See also Dem'yanov (1970), Medanic (1970). The minimax optimality conditions have also been derived from conditions for optimality in generalized least pth approximation problems for $p + \infty$ by Bandler and Charalambous (1971, 1972b, 1973a).

3.4.2 Conditions for a Minimax Optimum

The minimax problem considered is the unconstrained version of the problem stated in Section 3.3.1 (i.e., when (3.26) is ignored). The necessary (Theorem 1) and sufficient (Theorem 2) conditions for a minimax optimum are stated as follows.

Theorem 1

At an optimum point ϕ^0 for the minimax approximation problem there exist

 $t = 1, 2, ..., k_{r}$ (3.44)u > 0

such that



k,r (3.46)Σ t=1 = 1 '

where $\hat{y}_{t}(\phi^{0})$ for $t = 1, 2, ..., k_{T}$ are the equal maxima.

Theorem 2

If the relations in Theorem 1 are satisfied at a point ϕ^0 and all the functions $y_i(\phi)$ for it is convex, then ϕ^0 is optimal.

Theorems 1 and 2 have been proved by Bandler (1971), and the optimality conditions as derived by Curtis and Powell (1966) follow immediately from these theorems.

3.4.3 Practical Implementation

Once a proposed or a design solution is obtained for a minimax problem, it may be necessary to investigate the necessary optimality conditions. If the point ϕ , corresponding to a selution, is to be tested

for optimality, an attempt is made to solve

plus (3.44) and (3.46) for $k_r = 1, 2, ...$ until for a value of k_T^* (n_r), (3.44), (3.46) and (3.47) are satisfied. If this is not possible, the necessary conditions are not satisfied.

A computer program has been developed which can test a solution for the necessary conditions for a minimax optimum by two formulations. One uses a linear programming approach, and the other the solution of a set of linear independent equations. See Appendix B, Bandler and Srinivasan (1973c, 1973d).

3.4.4 Method 1

(3.44), (3.46) and (3.47) are solved here by minimizing

such that (3.44), (3.46) are satisfied and

 $\begin{vmatrix} k_{T} & \frac{\partial \hat{y}_{E}}{\Sigma} \\ \frac{\Sigma}{t=1} & u_{E} \frac{\partial \hat{y}_{E}}{\partial \phi_{1}} \end{vmatrix} \leq u_{k_{T}} + 1 \qquad i = 1, 2, \dots, k$

Linear programming ensures that

$$u_{1} \ge 0$$
 $t = 1, 2, \dots, k_{r} + 1$ (3.50)

(3.48)

(3.49)

3.4.5 Method 2

Here, we solve a set of linearly independent equations

$$\frac{k}{\Sigma^{T}} \frac{\partial y_{\ell}}{\partial \phi_{i}} = 0 \qquad i \in K^{\dagger} \qquad (3.51)$$

and (3.46), where K'is a suitable subset of $\{1, 2, \ldots, k\}$.

There is no guarantee, however, that (3.44) will hold. When k_r -1 is greater than the number of elements of K', the system of equations (3.46) and (3.51) have more unknowns than equations, and we use Method 1 to get the u.



Appendix B contains a program description incorporating the ideas of the previous two sections. The program package can be called from the user's main program and either of the two, or both the methods can be used to test the optimality conditions. The user can either specify the value of k_r or a tolerance ξ relative to \hat{y}_1 within which some of the $\hat{y}_2, \ldots, \hat{y}_{n_r}$ lie. The necessary conditions for optimality are satisfied when the norm ||r|| of the residual vector

falls within a user-specified value c, and (3.44), (3.46) hold, for a value of m_{T} starting with 1. If the conditions are not satisfied for m_{T} =1, m_{T} is incremented by 1 and the procedure is repeated. The investigation ends as soon as the conditions are satisfied for a value of $m_{T} \leq k_{T}$, or

(3.52)

the conditions are not satisfied for $m_r = 1, 2, ..., k_r$. The userspecified definitions of $||\cdot||'$ and the value of ε should be realistic so that the program may give meaningful results.

The importance of this investigation cannot be underestimated especially when there may be a number of solutions obtained by the same, or different optimization methods for a given problem and one wishes to test these solutions for optimality so as to be able to detect local optima, and to compare the methods for convergence towards the optima. This program may be used in such a way that it is possible to investigate the solutions after a certain number of iterations of the algorithm, or when a certain convergence criterion is reached, so that one may decide whether to carry on with further optimization, or to terminate altogether.

The program also makes it possible to find the maxima which are active in the vicinity of the optimum, so that the user may gain insight into the various scaling factors associated with the problem.

3.4.7 Example

The problem chosen was the lower-order modelling of a ninthorder nuclear reactor system when the operating reactor power level is in the 90-100 percent range of the full power (Bereznai 1971). A second-order model was chosen and the step-response of the system was approximated by that of the model for a minimax objective over a timeinterval of 0-10 seconds. A solution was obtained for this problem and the program described in Appendix B was used to test the solution

for optimality.

The relevant input parameters are: k = 2, $n_T = 4$, $c = 10^{-6}$, $\xi = 0.01$, and the norm chosen is given by :

$$||\mathbf{r}|| = \max_{\substack{n \leq i \leq k}} |\mathbf{r}_i|$$

 $\nabla \hat{y}$ is given by $\sim \nabla$

$$\nabla \hat{y}_{1} = \begin{bmatrix} .38711013 \times 10^{-3} \\ ..14208087 \times 10^{-3} \end{bmatrix}, \\ \nabla \hat{y}_{2} = \begin{bmatrix} -.29632883 \times 10^{-1} \\ .10876118 \times 10^{-1} \end{bmatrix}$$
$$\nabla \hat{y}_{3} = \begin{bmatrix} .79840875 \times 10^{-3} \\ .68487328 \times 10^{-2} \end{bmatrix}, \\ \nabla \hat{y}_{4} = \begin{bmatrix} .17968278 \times 10^{-2} \\ .14014776 \times 10^{-3} \end{bmatrix}$$

and
$$\hat{y}$$
 is given by
 $\hat{y}_1 = .29234162 \times 10^{-2}$, $\hat{y}_2 = .29234034 \times 10^{-2}$
 $\hat{y}_3 = .23141899 \times 10^{-2}$, $\hat{y}_4 = .62431057 \times 10^{-3}$

Corresponding to $\xi = 0.01$, the value of k_r is equal to 2. Both the methods were used to test the solution for optimality, and the results obtained are shown below.

(i)
$$n_{-} = 1$$

Both the methods give the same result as there is only one function under consideration.

7

ça T

$$r = [0.38711013 \times 10^{-3} - .14208087 \times 10^{-3}]^{T}$$

$$||r|| = 0.38711013 \times 10^{-3}$$

(3.44) and (3.46) are satisfied, while $||\mathbf{r}||$ is not less than ϵ . Thus the conditions are not satisfied for $\mathbf{m}_{\mathbf{r}} = 1$.

Method 1

$$u = [0.98710491 \quad 0.12895086 \times 10^{-1}]^{T}$$

$$x = [-0.25789922 \times 10^{-9} \quad 0.25789922 \times 10^{-9}]^{T}$$

$$|\mathbf{r}|| = 0.25789922 \times 10^{-9}$$

Method 2

$$u = [0.98710492 \quad 0.12895077 \times 10^{-1}]^{T}$$

$$x = [0. \qquad -0.35255563 \times 10^{-9}]^{T}$$

$$||r|| = 0.35255563 \times 10^{-9}$$

(3.44) and (3.46) are satisfied and $||\mathbf{r}|| < \varepsilon$ for both the methods. The necessary optimality conditions are thus satisfied for $\mathbf{m}_{\mathbf{r}} = 2$. It is also observed that due to the type of formulation of the problem in Method 1, the elements of \mathbf{r}' have equal magnitude.

44

<u>Э</u>,

3.5 Conclusions

A new minimax algorithm called grazor search has been proposed. Conditions which guarantee the convergence of the algorithm have also been stated. The spectrum of problems that can be accomodated has been extended to include constrained minimax objectives, and any efficient unconstrained minimax method can suitably be used for this purpose. The practical investigation of a solution for necessary optimality conditions has been implemented on the computer, so that it is now possible to check solutions at any stage of the optimization process. The subject matter of this chapter makes it possible to tackle. unconstrained and constrained minimax problems by a new gradient algorithm, and to test intermediate or final solutions for optimality, on line.

CHAPTER IV

COMPUTER-AIDED CIRCUIT DESIGN

41 Introduction

This chapter primarily concentrates on applying the ideas presented in Chapter III to computer-aided design of electrical networks. Minimax designs are of special interest to the designer mainly because they attempt to achieve an equiripple behaviour of the response error function, which is useful in many cases. The problems considered include the design of LC transformers and cascaded transmission-line transformers and filters. Appropriate constraints have been incorporated whenever necessary, and the grazor search algorithm has been compared with the Osborne and Natson method and razor search strategy for reliability and efficiency (See Bandler, Srinivasan and Charalambous 1972, Bandler and Srinivasan 1973a). Unless otherwise mentioned, the objective function to be minimized is chosen as (2.13).

4.2 Lumped LC Transformer

The problem considered (Hatley 1967) is the design of a 3section lumped-element LC transformer to match a 10 load to a 30 generator over the angular frequency range of 0.5 - 1.179 radians/sec. Fig. 4.1 shows the structure of the network, and the objective is to minimize

(4.1)

 $U(\phi) = \max_{i} |\rho_{i}(\phi)|$



where $\rho_i(\phi) = \rho(\phi, \psi_i)$ is the reflection coefficient over 21 uniformly spaced frequencies ψ_i in the passband, and

$$\phi = [L_1 C_2 L_3 C_4 L_5 C_6]^T$$
(4.2)

The six parameters were optimized by the grazor search strategy and the Osborne and Watson method, and Fig. 4.2 shows a typical graph of objective function against function evaluations for the two methods for identical starting points. As can be seen from the graph, the Osborne and Watson method fails to reach the vicinity of the optimum, while the grazor search algorithm achieves an optimal solution. Table 4.1 shows the number of function evaluations needed to get within 0.01 percent of the optimum for different values of n, the factor of resolution between two interior points of the golden section for the grazor search, and it is clear that the value of n chosen need not be very small.

4.3 Quarter-Wave Transmission-Line Transformer

The problem considered is the design of 2-section and 3-section 10Ω to 1Ω transmission-line transformers over a 100 percent relative bandwidth centred at 1 GHz(Matthaei, Young and Jones 1964, Bandler and Macdonald 1969a, 1969b). The objective is to minimize max $| 0 (\phi, \psi_i) |$ on 11 frequencies ψ_i in the band 0.5-1.5 GHz for the network shown in Fig. 4.3, where ρ_i is the reflection coefficient of the metwork at ψ_i .

The grazor search method and the Qsborne and Watson algorithm were used for minimax optimization. For both the methods, the objective \Rightarrow

48

ΰ



GRAZOR SEARCH METHOD TO REACH WITHIN 0.01 PERCENT OF THE OPTIMUM FOR DIFFERENT VALUES OF n FOR IDENTICAL STARTING POINTS • $L_1 = L_3 = L_5 = C_2 = C_4 = C_6 = 1$ 7 Function Evaluations η 0.01 1316 0.10 880 0.50 561

TABLE 4.1

COMPARISON OF THE NUMBER OF FUNCTION EVALUATIONS REQUIRED BY THE

50

ج.



J

function is given by (2.13) where

 $y_{i}(\phi) = \frac{1}{2}|p_{i}(\phi)|^{2}$

In the 2-section examples, the 11 frequencies were uniformly spaced. In the 3-section examples, the frequencies were 0.5, 0.6, 0.7, 0.77, 0.9, 1.0, 1.1, 1.23, 1.30, 1.40, and 1.50 GHz. The progress of the algorithms from identical starting points with respect to the number of function evaluations (one corresponding to 11 evaluations of ρ) is recorded in Figs. 4.4 and 4.5. The points shown mark the successful end of a linear search or the beginning of linear programming.

A comparison was made between the grazor search, Osborne and Watson, and razor search methods, as shown in Tables 4.2 and 4.3. From Table 4.2, it is clear that the grazor search algorithm is, in general, faster than the razor search technique for the 2-section case when the lengths are kept fixed and the impedances are varied. From Table 4.3, it is clear that the grazor search algorithm is the best. The Osborne and Watson algorithm, though fairly fast initially, may in some cases fail or slow down near the optimum.

The grazor search method and the Osborne and Watson algorithm were further compared on the 3-section transformer problem when the lengths were fixed at quarter-wavelength values and the impedances were varied. For a starting point of $Z_1 = 3.16228$, $Z_2 = 1.0$ and $Z_3 = 10.0$, the former took 184 and 218 function evaluations, while the latter consumed 151 and 219 function evaluations to reach within 0.01 and 0.001 percent of the optimum value of the maximum reflection coefficient,

52

(4.3)















TABLE 4.2

OPTIMIZATION OF A 2-SECTION 100 TO 10

TRANSMISSION-LINE TRANSFORMER OVER 100 PERCENT RELATIVE BANDWIDTH

Starting Poin	t .	Function Ev	aluations [†]
z ₁	z ₂	Razor Search	Grazor Search
			•
1.0	3.0	157	126
	U U	207	
1.0	6.0	34	83
	X	152	
7 5		223	52
5.5	.	100	
~ ~		210	* 29 ·
5.5	J. U	163	X
· .		105	,

+ Number of function evaluations required to bring the reflection

coefficient within 0.01 percent of its optimum value.

TABLE 3.3

3

OPTIMIZATION OF A 3-SECTION 100 TO 10 TRANSMISSION-LINE

5

TRANSFORMER OVER 100 PERCENT RELATIVE BANDWIDTH

	Fixed L	, engths		Variable	Lengths and	Impedances
° Parameters φ.	Starting Point	Maximum Reflection Coefficient at Start	Starting	Maximum Reflection Coefficient at Start	Starting Point	Maximum Reflection Coefficient at Start
z1 z1 z2/zq z2/zq z3/zq z3	1.0 1.0 1.0 3.16228 1.0 10.0	0.70930	1.0 1.0 1.0 5.16228 1.0 10.0	0.70930	0.8 1.5 3.0 6.0 6.0	0.38865
RAZOT	Final Maximum Reflection Coefficient	0.19729	0	0.19733		0,19731
Search Aigorithm	Number of Function Evaluations	406		1300		1250
Grazor	Final Maximum Reflection Coefficient	0.19729		0.19729		0.19729
Search Algorithm	Number of Function Evaluations	219		696		198

í.

ŷ
•	OPTIMIZATION OF TRANSFORMEF	TABLE 4.5 (cor F A 3-SECTION 106 TO 16 R OVER 100 PERCENT REL	atinued) a TRANSMISSION-LINE ATIVE BANDWIDTH	
	Fixed I	Lengths	Variable Lengt	hs and Impedances
Algorithms	Final Maximum Reflection Coefficient	0.19729	0.20831	0.19788
due to Osborne and Matson (1969)	Number of Function Evaluations	199	860	237
			¥ . —	`
*		,		
•	f ▼ -			x

respectively. This case illustrates how the two algorithms compare when both methods work efficiently.

 \mathcal{A}

4.4 Cascaded Transmission-Line Filters

In this section, the grazor search algorithm is used to achieve the minimax design of cascaded transmission-line filters with desired attenuation characteristics. Three examples are chosen, and the ideas presented in Chapter III are applied to the problems.

4:4.1 Problem 1

The design of a 7-section cascaded transmission-line filter with frequency-dependent terminations is considered here (see Fig. 4.6). This problem has been considered by Carlin and Gupta (1969). The frequency variation of the terminations is like that of rectangular waveguides operating in the H_{10} mode with cutoff frequency 2.077 GHz. All section lengths were kept fixed at 125cm so that the maximum stopband insertion loss would occur at about 5 GHz. The passband 2.16 to 3 GHz was selected, for which a maximum passband insertion loss of 0.4dB was specified.

Fig. 4.7 shows the response of Carlin and Gupta which was used as an initial design. The other responses in Fig. 4.7 are a least 10th optimum obtained by Bandler and Seviora (1970) and a minimax optimum obtained by the grazor search strategy. In both cases only the passband was optimized. The minimax response has a maximum passband insertion





loss of 0.086 dB. Table 4.4 gives the appropriate parameter values.

Fig. 4.8 shows the results of applying the grazor search method to optimize the sections in a filtering sense. Thus, it was desired to meet the 0.4 dB passband insertion loss while maximizing the stopband insertion loss at a single frequency (5GHz). Let

 $y_{i}(\phi) = \begin{cases} \frac{1}{2}(|\rho_{i}(\phi)|^{2} - r^{2}) & \text{in the passband} \\ \frac{1}{2}(|\rho_{i}(\phi)|^{2}) & \text{in the stopband} \end{cases}$ (4.4)

where

 $\phi = [z_1 \ z_2 \ \cdots \ z_7]^T$

and r is the reflection coefficient magnitude corresponding to an insertion loss of 0.4 dB. Here 22 uniformly-spaced points were selected from the passband. Table 4.4 gives the resulting parameter values. A similar response was attained by the grazor search technique when the section impedances were assumed symmetrical i.e., $Z_5 = Z_3$, $Z_6 = Z_2$, $Z_7 = Z_1$.

4.4.2 Problem 2

The problem chosen consists of a 5-section cascaded transmissionline low-pass filter design and has been previously considered by Brancher, Maffioli and Premoli (1970). The filter structure is the same as in Fig. 4.3 for R=1. The terminating impedances are real and normalised to be 1Ω . It is required to have a passband insertion loss of less than 0.01 dB from 0 to 1 GHz and as high a stopband insertion loss as possible at 5 GHz. Twenty-one uniformly spaced points were chosen in the passband and one point in the stopband (5 GHz). The length of each section is

(4.5)

TABLE 4.4

COMPARISON OF PARAMETER VALUES FOR THE 7-SECTION FILTER (PROBLEM 1)

haracteristic Impedances Normalized)	Carlin and Gupta (1969)	Minimax Design (Fig. 4.7)	Minimax Design (Fig. 4.8)
			. *
Ζ.	1476.5	1305.2	3069.4
- 1 7	733.6	607.8	2856.4
- 2 7	1963.6	1323.3	25871.2
~ 3	461.8	362.7	10573.3
-4	1963.6	1323.2	25874.0
⁺ 5	733.6	607.9	2856.7
² 6 7	1476.5	1305.2	3069.8

 \mathcal{O} .

67

١.



normalized with respect to $l_q = 1.49896$ cm, the quarter-wavelength at 5 GHz.

The $y_i(\phi)$ are given by (4.4) where r is the reflection coefficient magnitude corresponding to an insertion loss of 0.01dB, and

$$\phi = \left[\ell_{n1} \ Z_1 \ \ell_{n2} \ Z_2 \ \ell_{n3} \ Z_3 \ \ell_{n4} \ Z_4 \ \ell_{n5} \ Z_5 \right]^T$$
(4.6)
$$\ell_{ni} = \ell_i / \ell_q \qquad i = 1, 2, ..., 5$$
(4.7)

The lengths were initially fixed at t_q , and the impedances varied. Levy (1965) has derived an optimal solution to this problem analytically. The grazor search method was used on this problem for minimax optimization, and the result obtained was identical to the one derived by Levy and Fig. 4.9 shows the optimal response obtained.

Brancher, Maffioli and Premoli (1970) have achieved some results for the problem, and an observation of their responses leads one to suspect that the results are not optimal. The grazor search method was used to test whether an improvement on the results of Brancher, Maffioli and Premoli was possible, and improved results were obtained.

Fig. 4.10 and Table 4.5 show the results for the problem where the impedances are fixed at some practical values and only the lengths are allowed to vary. As the final values obtained by the grazor search method indicate, the response at finish represents a good improvement over the response at start, both from passband and stopband considerations.





TABLE 4.5



4.4.3 Problem 3

The design of a 5-section cascaded transmission-line filter subject to parameter constraints is considered here, and the ideas presented in Section 3.3 are used to tackle this problem. The filter structure is the same as the one considered in Section 4.4.2. The problem has been previously considered by Carlin (1971) for fixed lengths at a quarter-wavelength of $L_q = 2.5$ cm corresponding to 3 GHz, and for a required attenuation of 0.4 dB in the passband (0-1 GHz). Optimal values have been derived for characteristic impedance values when a stopband frequency of 3 GHz was chosen (Levy 1965). The objective function to be minimized was chosen as (2.13) where

$$y_{i}(\phi) = \begin{cases} |\rho_{i}(\phi)| - r & \psi_{i} \in 0 - 1 \text{ GHz} \\ \\ \\ 1 - |\rho_{i}(\phi)| & \psi_{i} = 3 \text{ GHz} \end{cases}$$

 ϕ corresponds to (4.6) and r corresponds to an attenuation of 0.4 dB. Twenty-one uniformly-spaced points were chosen in the passband.

Initially the lengths were fixed at L_q and the impedances Z_i were varied. The impedance constraints imposed were

$$0.5 \leq \frac{7}{i} \leq 2.0$$
 /i = 1,2,..., 5 (4.9)

and the minimization function was chosen as $W'(\phi, w')$ of (3.39) where W' is given by (3.40), n = 22, m= 10, and

$$g_{2i-1}(\phi) = Z_i - 0.5 \ge 0$$

 $i = 1, 2, ..., 5$ (4.10)

$$2_{2i}(\phi) = -(Z_i - 2.0) > 0$$

(4.8)

نز)

$$w_{j}' = \begin{cases} 1000 \text{ for } g_{j}(\phi) < 0 \\ & j = 1, 2, ..., m \end{cases}$$
(4.11)
0 for $g_{j}'(\phi) \ge 0$

The result of optimizing the impedances using the grazor search method is shown in Table 4.6 where U corresponds to $\max_{i} y_{i}(\phi)$, and y_{i} is given by (4.8). It is observed that some of the impedances of the constrained solution lie on constraint boundaries. Moreover, there are two distinct solutions, for which the impedances are reciprocals of each other.

As a further step, it was desired to investigate the possibility of improving the unconstrained optimal solution (for length fixed at t_q) of Table 4.6, by allowing both the lengths and impedances to vary, and imposing the following constraints:

 $0 \leq \ell_{ni} \leq 2$

1

$$i = 1, 2, \dots, 5$$
 (4.12)

 $0.4416 \le Z_i \le 4.419$ i = 1, 2, ..., 5 (4.13)

$$5 \qquad (4.14)$$

$$0 < \Sigma \quad t \quad < 5$$

$$- i z \quad nj \quad -$$

where the t_{ni} correspond to (4.7) and the upper and lower bounds of Z_i in (4.13) correspond to upper and lower values of the unconstrained optimal values of Table 4.6.

74

Ł.

5-SECTION TRANSMISSION-LINE LOWPASS FILTER DESIGN (PROBLEM 3)

FOR LENGTHS FIXED AT Lq

)	Unconstrained	Constrained	Solution
	Optimal Solution	(i)	(ii)
7	3.151	0.5683	1.760
21 7	0.4416	2.000	0.5000
² 2 7	4.419	0.5000	2.000
23 7	0.4416	2.000	0.5000
² 4 ² 5	3.151	0.5683	1.760
· U	3.951x10 ⁻⁵	3.255x10 ⁻³	3.255×10 ⁻³
	2.419x10 ³	3.255x10 ⁻³	3.255x10 ⁻³
n'		·	

75

The function to be minimized was chosen as $V(\phi, \phi_{k+1}, \alpha)$ in (3.32) where α is given by (3.33) and (3.34), n = 22, m = 22,

 $\alpha_{j} = 10$ j = 1, 2, ..., m+1 (4.15)

and $g_j(\phi)$, j = 1, 2, ..., m correspond to the constraints (4.12)-(4.14) It was observed that no improvement could be achieved from the starting value (corresponding to the unconstrained optimal solution of Table 4.6) and that the starting point satisfies the necessary conditions for a minimax optimum, as verified by the method described in Section 3.4 and Appendix B.

4.5 Conclusions

The results indicate that the grazor search algorithm is generally more reliable in reaching an optimal minimax solution than the Osborne and Watson algorithm, and is faster than the razor search technique. Typically 1 min is sufficient time to optimize a six-parameter design, and 2 to 3 min are sufficient to optimize a ten-parameter problem, depending on how far from the optimum one starts and how close one wishes to get, on a CDC 6400 computer. The grazor search algorithm is capable of handling, without any difficulty, filter design problems with upper and lower specifications over many frequency bands. The method should be very useful in design problems for which exact methods are not available.

CHAPTER V

SYSTEM MODELLING

5.1 Introduction

Lower-order modelling of complex high-order systems is now widely being used in the area of systems design and control both online and off-line. The modelling can be performed for a variety of performance criteria and objectives, using different model derivation techin. niques. Some of the techniques obtain a model by neglecting modes of the original system which contribute little to the overall response of the system (Davison 1966, Chidambara 1969, Mitra 1969, Marshall 1966). Other methods search for optimal coefficients of a set of differential or difference equations of a given order, the response of which is approximated as cosely as possible to that of the system, when both are driven by the same inputs (Anderson 1967, Sinha and Pille 1971, Sinha and Bereznai 1971, Markettos 1972). The search of these coefficients has been, in the past, carried out using both direct search and gradient methods of optimization for a least-squares or quadratic cost function, but for this work, the investigation is mainly on near-minimax and minimax objectives; and the input-output data of the system is assumed to be known. See also Chen and Shieh (1968) and Kokotović and Sannuti (1968).

77

Đ

5.2 Statement of the Problem

It is required to find a transfer function of a model of a given order, the response of which is the best approximation to the response of the actual system to a particular input for a specified error criterion.

Ŷ

 $^{-8}$

(5.1)

In general the transfer function of a given order n may be written as

$$H_{m,n}(s) = \frac{b_{m}s^{m}+b_{m-1}s^{m-1}+\ldots+b_{1}s+b_{0}}{s^{n}+a_{n-1}s^{n-1}+\ldots+a_{1}s+a_{0}}$$



where $m \leq n$ for physical systems. For this work the input is a unit step and the criterion chosen is to directly or indirectly minimize an error function over a specified time interval [0,T]. The problem, therefore, is the determination of the parameters ϕ , given by

$$\Rightarrow = [a_0 \ a_1 \ \dots \ a_{n-1} \ b_0 \ b_1 \ \dots \ b_m]^T$$
 (5.2

such that an error function is minimized. Optimization of model parameters for a least-squares error criterion has already received attention (Bandler, Markettos and Sinha 1973, Markettos 1972).

5.3 Minimax System Modelling

1

The error criterion chosen is to minimize the maximum error

between the system and model responses over [0,T], where ϕ is given by (5.2). The following notation is introduced.

$$t_{i} \qquad \text{is an i th time instant in [0,T]}$$

$$I \qquad \text{is an index set of i such that } t_{i} \in [0,T]$$

$$c_{i}^{S} \qquad \text{is the response of the system at } t_{i}$$

$$c_{i}^{m}(\phi) \qquad \text{is the response of the approximating}$$

$$model at t_{i}$$

$$c_{i}^{(\phi)} - c_{i}^{S} \qquad \text{is the error between the system and the model}$$

$$responses at t_{i}$$

$$c_{\infty}^{S} \qquad \text{is the steady-state value of the system}$$

$$c_{\infty}^{m} \qquad \text{is the steady-state value of the model}$$

In Section 5.4, the approximation problem considered assumes that c_{∞}^{m} is fixed at a convenient value (usually c_{∞}^{5} or c_{1}^{5} at t_{1} =T), so that the objective is to minimize

$$U(\phi) = \max_{i \in [0,T]} y_i(\phi)$$
(5.3)

where

 $y_i(\phi) = |e_i(\phi)|$

This problem can now be solved by an efficient minimax or nearminimax optimization method as suggested in Sections 2.5 and 2.6.

79

(5.4)

5.4 Example

ç

The problem considered is the modelling of a seventh-order system representing the control system for the pitch rate of a supersonic transport aircraft (Dorf 1967, Bandler, Markettos and Sinha 1973). The transfer function of the system is given by

$$G(s) = \begin{cases} \frac{375000(s+0.08333)}{s^{7}+83.64s^{6}+4097s^{5}+70342s^{4}+853703s^{3}+} \\ 2814271s^{2}+3310875s+281250 \end{cases}$$
(5.5)

with a steady-state value of 0.11111 for a unit step.

Minimax optimization of the model parameters as performed by the grazor search method consists of minimizing (5.3), while near-minimax optimization minimizes

 $f(\phi) = U(\phi) \left[\sum_{\substack{\tau_i \in [0,T] \\ \tau_i \in [0,T]}} \left| \frac{e_i(\phi)}{U(\phi)} \right|^p \right]^{1/p}$ (5.6)

for large values of p (Bandler and Charalambous1972d). Let $J \subset I$ be an index set relating only to the extrema of the error functions $y_i(\phi)$ given by (5.4). If I is replaced by J in (5.6), considerable economy in computing time results at a slight risk of creating false optima. The larger the value of p, the closer the solution gets to the minimax result, but the central processor time increases considerably. For this work, a value of p=1000 was considered suitable for optimization purposes. For least p th optimization, three gradient methods due to Fletcher and Powell (1963), Jacobson-Oksman(1972) and Fletcher (1970) have been used fur the modelling problem.



÷.

5.4.1 Second- and Third-Order Models

The time-interval over which the approximation was made was 0-8 seconds (T=8 sec). 101 uniformly-spaced sample points were chosen over the interval. The steady state value of the model for a unit step $(E=c_{\infty}^{m})$, was set at 0.11706, corresponding to the response of the system at the final sample point $(c_{i}^{S} \text{ for } t_{i}=T)$. See Bandler, Markettos and Srinivasan (1972, 1973).

Two second-order and one third-order models were considered for minimax approximation of the system. The transfer functions of the chosen models were

$H_{02}(s) = \frac{Ea_0}{s^2 + a_1 s + a_0}$	(5.7)
$H_{12}(s) = \frac{b_1 s + Ea_0}{s^2 + a_1 s + a_2}$	(5.8)
$H_{23}(s) = \frac{b_2 s^2 + b_1 s + Ea_0}{s^3 + c_2 s^2 + c_2 s + c_2 s + c_2}$	(5.9)
$= \frac{\frac{x_{5}s^{2} + x_{4}s + Ex_{1}x_{3}}{(s + x_{7})(s^{2} + x_{7}s + x_{1})}}$	(5.10)

where

ЕÅС

For this work, the response of the models in the time domain were obtained by using standard Laplace Transform Tables to invert from the s to the t domain.

(a) 2-Parameter Problem

The model transfer function chosen is (5.7) and the parameter

81

(5.11)

÷.

vector is given by

 $\oint_{\mathcal{D}} = [a_0 \ a_1]^T$ (5.12)

The optimum parameters using the grazor search method were

$$a_0 = 3.06472, a_1 = 2.38338$$

resulting in a four-ripple error curve with a maximum error value

 $U = 3.76347 \times 10^{-3}$

The response and error curves are shown in Figs. 5.1(a) and 5.1(b) respectively.

The optimum parameters using least pth approximation for p=1000 were

 $a_0 = 3.06549, a_1 = 2.38414$

resulting in a similar four-ripple curve with a maximum error value

 $U = 3.76510 \times 10^{-3}$

Table 5.1 shows the number of function evaluations required for each of the methods to reach a maximum error value of 3.76619×10^{-3} . For this problem the Fletcher method and Jacobson-Oksman method appeared to be the most efficient.

(b) 3-Parameter Problem

By allowing the model to have a zero, as indicated by (5.8) a 3-variable problem results, where

$$\phi = [a_0 a_1 b_1]^T$$

(5.13)





TABLE 5.1

.

SEVENTH-ORDER SYSTEM MODELLING EXAMPLE NUMBER OF FUNCTION EVALUATIONS REQUIRED TO REACH U = 3.76619×10^{-3} FOR THE 2-PARAMETER MODEL

		•	Minim	ization of $f(\phi)$	
tarting point	Minimization			Jacobson ·	Oksman
پ	of U(\$) ~ Grazor	Fletcher	Powell	Quadratic Step Prediction	Homogeneous Step Prediction
3.0	107	42	59	36	36
2.0					
1.0	130	78	334	91	127
1.0					
1.0	165	96	718	834	*
4.0	<i>A</i>				· · · · · · · · · · · · · · · · · · ·
4.0	129	64	false	41	45
रूरे 1.0			optimum		
and an and a second			·	. *	

.

Indicates an ARGUMENT TOO LARGE message was given by the computer.

The optimum parameters using the grazor search method were

$$a_0 = 3.83255$$
, $a_1 = 3.00365$, $b_1 = -.0176390$

giving a maximum error value

 $U = 2.48724 \times 10^{-3}$

The response and error curves are shown in Figs. 5.2(a) and 5.2(b) respectively.

For p=1000 the optimum parameters obtained were

 $a_0 = 3.83592$, $a_1 = 3.00605$, $b_1 = -.0177277$

giving similar response and error curves as in Figs. 5.2(a) and 5.2(b) and

 $11 = 2.48794 \times 10^{-3}$

The number of function evaluations needed for the three parameter problem to reach the value $U = 2.48794 \times 10^{-3}$ are shown in Table 5.2. The grazor search technique and the Fletcher method required a smaller number of function evaluations.

(c) 5-Parameter Problem

2

The third-order model of (5.9) is considered next. For computional efficiency, the transfer function of the form (5.10) is chosen. The model has five parameters given by

$$= [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T$$

(5.14)

The optimum parameters obtained using the grazor search method



đ



TABLE 5.2

			Minis	ization of	f f(g)	
Starting	Minimization		_	Jacob	oson - Oksman	
point ¢	of U(\$)	Fletcher	Fletcher-	Quac Pr	lratic Step rediction	Homogeneou
ý	Grazor		Powell	. p = 1	ρ = 0.5	Step Prediction
2.5 2.0 -2.0	149	339	- 500	279	•	339
1.0 1.0 -1.0	368	362	•	104	276	137
4.0 3.0 0.01	165	242	184	142	97	260
3.5 1.5 -1.0	358	280	342	217	151	•
5.0 1.0 -1.0	325	193	• • •	•	205	•
5.0- 1.0 3.0	406	245	•	159	119	•

*Indicates time limit of 64 seconds was reached.

4

"Indicates an ARGUMENT TOO LARGE message was given by the computer.

89

were

J

$$x_1 = 4.34547$$
, $x_2 = 3.36809$, $x_3 = .108248$

$$x_{i} = .514475, x_{c} = -.0356180$$

resulting in a six-ripple error curve with a maximum error value

 $11 = 1.02062 \times 10^{-3}$

The response and error curves are shown in Figs. 5.3(a) and 5.3(b) respectively.

The optimum parameters using p=1000 were

 $x_1 = 4.34682, x_2 = 3.36738, x_3 = .0996086$

 $x_4 = .514728, x_5 = -.0356154$

giving response and error curves similar to those of Figs. 5.3(a) and 5.3(b) and a maximum error

 $U = 1.02063 \times 10^{-3}$

Some runs with the Fletcher-Powell method, on the five-parameter problem, indicated that the method was the slowest and since this was already established in the previous models, as indicated in Tables 5.1 and 5.2, further runs with Fletcher-Powell.method were considered unnecessary. The results of optimization by the other three methods are shown in Table 5.3.

The Fletcher method reached a unique six-ripple solution in all the cases tried, although there was a large variation in the number of function evaluations required. The grazor search technique reached the





92_.

SEVENTH-ORDER SYSTEM MODELLING EXAMPLE

NUMBER OF FUNCTION EVALUATIONS REQUIRED TO REACH THE INDICATED VALUE OF 1000 U FOR THE 5-PARAMETER MODEL

Starting	Minimization	ЕМ,	nimization of $f(\phi)$	· .
point ¢	of U(\$)		Jacobson-Oksman (Predictio	uadratic Step m
~	~ Grazor	Fletcher	ρ = 1	ρ = 0.5
	· · · · · · · · · · · · · · · · · · ·		* ~	J
3.0	437	530	886	778
1.5 0.5 -0.1	1.2139	1.0207	1.0206	1.0206
1.5	782	768	931	325*
2.5 1.0 0.1	1.2473	1.0207	1.0206	45 086
4 .0 3.0	. 489	177	114*	108
0.1 0.5 -0.03	1.0206	1.0207	1.5061 {	. 1.0206
3.0	634	. 862	248	350
0.2 0.3 -0.1	1.1720	1.0207	1.0206	1.0207
5.0	817	484	17•	582
0.5 1.0 -0.5	1.0337	1.0207	19.660	1.0207
Least	537	799	263*	1208
Squares Optimum	1.2472	1.0206	1.8954	1.0283

"Indicates time limit of 128 seconds was reached.

Indicates an ARGUMENT TOO LARGE message was given by the computer.

six-ripple solution in one of the cases shown, while in some of the other cases it terminated in a five-ripple solution.

In some instances, the real pole of the model had the tendency to move to the right-hand side of the s-plane and since this would produce an unstable model, the last parameters giving stable results were taken as the final values. In all cases, however, the real pole seems to lie very close to the axis and any constraint, although easily implemented in the form of a square transformation, would have made the pole go to zero.

It was further noted that when the Fletcher method, used with p=1000, was started from one of the five-ripple solutions where the grazox search technique terminated, a direction was found which decreased $f(\phi)$ while temporarily increasing $U(\phi)$ and the method converged towards the six-ripple minimax solution, though slowly. When the same procedure was repeated with $p=10^6$, the algorithm failed to move from that point. Figs. 5.4(a) and 5.4(b) show the response and error curves for a five-ripple solution obtained by the grazor search method.

5.4.2 Optimality of Model Parameters

The conditions for minimax optimality, as mentioned in Section 3.4.2, were applied to the final parameter values arrived at through optimization of the grazor search method (the corresponding responses are shown in Figs. 5.1-5.4), and the results are indicated in Tables 5.4-5.7. The necessary conditions are satisfied in all the cases, as observed from the tables. The $\hat{\gamma}_g(\phi)$ for $L = 1, 2, ..., n_g$ are the local





error

TABLE 5.4.

VERIFICATION OF CONDITIONS FOR MINIMAX OPTIMALITY

2-PARAMETER SOLUTION CORRESPONDING TO FIG. 5.1

Ĺ	Time Instant	Error Maximum (1000ŷ _l)	Multiplier (u _l)
	0.24	3.76347	0.75047
3).	0.88	3,76347	0.16519
	ο 2.16	3.76347	8.4342×10^{-2}
VERIFICATION OF CONDITIONS FOR MINIMAX OPTIMALITY

3-PARAMETER SOLUTION CORRESPONDING TO FIG. 5.2

 $n_{r} = 4$, $k_{r}^{*} = 3$ Multiplier (u₁) Error Maximum $(1000\hat{y}_{1})$ Time Instant L 0.90758 2.48724 4.0 1 <u></u> 17 4.2744×10^{-2} 2.48724 0.24 2 4.9680×10^{-2} 2,48724 0.96 3 2.00700×10^{-1} 2.00 4 $[0.0 0.0 1.1 \times 10^{-5}]^{T}$ 1.0 u.

,

TABLE 5.6

VERIFICATION OF CONDITIONS FOR MINIMAX OPTIMALITY

5-PARAMETER, 6-RIPPLE SOLUTION CORRESPONDING, TO FIG. 5.3

 $n_{r} = 6, k_{r}^{*} = 6$ Error Maximum (1000ŷ_L) Multiplier Time Instant £ 👌 (u__)_ 3.6510×10^{-2} 1.020616 1.84 1 8.4333 x 10^{-2} 1.020616 0.72 2 0.51806 1.020616 Ó.08 3 1.020616 2.7915 x 3.76 0.32227 1.020616 0.24 5 1.0910×10^{-2} 1.016870 8.00 0.0]^T 0.0 0.0 0:0 [0.0 **t**=1

TABLE 5.7

VERIFICATION OF CONDITIONS FOR MINIMAX OPTIMALITY

5-PARAMETER, S-RIPPLE SOLUTION CORRESPONDING TO FIG. 5.4

 $n_{r} = 5, k_{r}^{*} = 5$

	Nultiplier (u _l)	Error Maximum (1000ŷ _t)	Time Instant	L
	0.23428	1.213988	0.32	 -
			ũ	ı
. 0	0.19815	1.213988	5.12	2
	0.39281	1.213988	0.08	5
	0.10217	1.213986	0.96	,
,	7.2598 x 10^{-2}	1.212651	0.32	,

1.0

t,

٤.

discrete maxima of $y_i(\phi)$, it as mentioned in Section 3.3.1, and Method 2 described in Section 3.4.5 is used for verifying the optimality conditions.

For the cases corresponding to Tables 5.5 and 5.7, k_T^* is equal to k and there are k_T^* +1 equations and k_T^* unknowns for the solution of (3.46) and (3.51). The non-zero values of the components of r for these cases correspond to the residuals of the dependent equations (refer to Sections 3.4.5, 3.4.6 and Appendix B).

In interpreting these results one may associate the results corresponding to Tables 5.4 and 5.6 in saying that the main criterion is how close to equal the ripples are and the results of Tables 5.5 and 5.7 in how small the size of the linear combination is in comparison with the sizes of the individual gradient vectors. In the first case we are satisfied with the criterion from a practical point of view, in the second the linear combination is about 2 to 4 orders of magnitude smaller than the gradient vectors.

5.4.3 Discussion

The grazor search algorithm is found to be more efficient than the Fletcher-Powell method on the problems chosen. The method proposed by Fletcher appears to be the most efficient of the methods used for nearminimax results in efficiency and consistency in reaching the vicinity of the optimum. The Jacobson-Oksman method; although giving good results,

appeared to be sensitive to scaling.

It has to be mentioned that the Fletcher-Powell package, as available in the IBM Scientific Subroutine Package, has a programming error. Appropriate corrections have been made and the Fletcher-Powell method has been applied to a number of test problems. The results have indicated that very little improvement is obtained for the corrected version. The Fletcher-Powell results, as shown in Tables 5.1-5.2, correspond to the uncorrected version, and it is expected that the corrected version might improve the function evaluations slightly.

5.5 New Approaches to Minimax System Modelling

In this section, some new ideas are presented so as to satisfy stringent design requirements (Bandler and Srinivasan 1973b, 1973e). In Section 5:4, c_m^m was assumed fixed. It may, however, be unacceptable to fix c_m^m at a certain value, in which case a realistic trade-off between transient and steady-state errors can be achieved. The design requirement may be such that arbitrary transient and steady-state response specifications need be imposed on the model for a desired performance criterion. It would also be realistic to expect the modelling procedure to be automated in such a way that it is possible to move from lowerorder models to high-order ones whenever, say, the solutions satisfy the necessary optimality conditions. 5.5.1 A Generalized Objective Function

It is possible to extend the ideas of constrained minimax optimization (discussed in Section 3.3) to system modelling so that a generalized objective function can be defined to take into account both the transient and steady-state response errors. The following additional notation is introduced.

is the upper bound of the system specifications at steady-state

S_{1.}

S.,,,,,

is the lower bound of the system specifications at steady-state

e....==c___S

is the error between upper steady-state specifications and model steady-state value

is the error between lower steady-state specifications and model steady-state value

The problem may now be formulated into two forms as follows. The first one minimizes with respect to ϕ and ϕ_{k+1}

$$\frac{V(\phi,\phi_{k+1},\alpha,\alpha_{\ell},\alpha_{u})}{\sqrt{t_{i}}} = \max_{\substack{t_{i} \in [0,T]}} [\phi_{k+1},\phi_{k+1}-\alpha(\phi_{k+1}-|\phi_{i}(\phi)|),$$

$$\phi_{k+1}-\alpha_{\ell},\phi_{\ell},\phi_{k+1}+\alpha_{u},\phi_{u},\phi_{u}]$$

where a, $a_{\underline{l}m}$, $a_{\underline{u}m}$ are positive. If $c_{\underline{m}}^{\underline{m}}$ is fixed such that $e_{\underline{l}m}$ and $-e_{\underline{u}m}$ are positive, the objective function (5.15) reduces essentially to $U(\phi)$ in (5.3). The second one minimizes with respect to ϕ

$$\frac{W(\phi, w_{\underline{t}}, w_{\underline{t}}, w_{\underline{t}})}{\sim} = \max_{\underline{t}_{\underline{t}} \in [0, T]} \left[\left[\phi_{\underline{t}}(\phi) \right] - w_{\underline{t}} \phi_{\underline{t}}, w_{\underline{t}} \phi_{\underline{t}} \right]$$

$$(5.16)$$

(5.15)

where

$$f_{L^{m}} \begin{cases} = 0 \quad \text{for } -e_{\underline{L}^{m}} < 0 \\ \\ > 0 \quad \text{for } -e_{\underline{L}^{m}} \ge 0 \\ \\ > 0 \quad \text{for } -e_{\underline{L}^{m}} \ge 0 \end{cases}$$
(5.17)

 $W_{um} \begin{cases} = 0 \quad \text{for } e_{um} < 0 \\ & & \\ > 0 \quad \text{for } e_{um} > 0 \\ & & \\ > 0 \quad \text{for } e_{um} = 0 \end{cases}$ (5.18)

If c_{\perp}^{m} is fixed within satisfied specifications the above objective. function reduces to U(ϕ) in (5.3).

In cases where suitable constraints - including parameter constraints - are imposed, the above procedure may be used to incorporate them in the objective function. In many cases, it is convenient to choose $S_{\underline{t}\underline{m}} = S_{\underline{u}\underline{m}} = c_{\underline{m}}^{S}$.

5.5.2 Automated Lower-order Models

One of the major problems that is encountered in modelling is to decide whether a certain lower-order model is acceptable or not. If the model is too simple so that computing time for optimizing model parameters is small, the approximation to the original system may be very bad, while if the model is complex, then the very need for system modelling is lost. If one were to strike a reasonable compromise between the speed with which the model is optimized, and the accuracy of the approximation, it would not be unreasonable to devise a scheme whereby one could increase

the complexity of the model in an automated fashion after a certain number of iterations or computer time. It is, however, important to keep in mind the desirability of making this increase in complexity as smooth as possible, so that the objective function value is not degraded. Thus, either the number of parameters could be increased for a model with a certain order, or the order of the model itself can be increased.

Let H_{m,n} denote an optimized model of the form (5.1). Three possibilities occur as follows.

(i) Increase in parameters only

 $H_{\mathbf{m},\mathbf{n}}^{*}(\mathbf{s}) + H_{\mathbf{m}+\mathbf{p},\mathbf{n}}(\mathbf{s})$

Here b_{m+p} , b_{m+p-1} , b_{m+1} are initially assumed to be zero so that $H_{m+p,n} = H_{m,n}^{*}$ in the first iteration.

(ii) Increase in order

$$H_{\mathbf{m},\mathbf{n}}^{*}(\mathbf{s}) + H_{\mathbf{m}+\mathbf{q},\mathbf{n}+\mathbf{q}}(\mathbf{s})$$

Here q poles of $H_{m+q,n+q}(s)$ are assumed to cancel with q zeros initially, so that $H_{m+q,n+q}=H_{m,n}^{*}$ in the first iteration. In this case, initial guesses for q poles (or zeros) are necessary.

(iii) Increase in order and parameters

Here $b_{m+q+p}, \dots, b_{m+q+1}$ are assumed to be zero initially and that there is a cancellation of q zeros and q poles at start, so that

 $H_{m+p+q,n+q} = H_{m,n}^{*}$ in the first iteration.

A careful choice of initial parameters can make the increase in model complexity smooth so that the whole modelling procedure can be automated on a small digital computer on-line.

5.5.3 Optimality Conditions

1 ~~

When a certain low-order model is being optimized, it may be useful to investigate intermediate or final solutions after a certain number of iterations of the modelling algorithm, or after a certain convergence criterion is reached, so that one may decide whether to carry on with further optimization, to increase the order of the model, or to terminate altogether. For minimum objectives, it is possible to test the optimality by the procedure outlined in Section 3.4.

5.5.4 Results

Two examples were considered, and two second-order models and a third-order model were chosen as follows.



(5.19)

(5.20)

(5.21)

The transition between the models can be made smooth by making the following substitutions at the start of the new model.

.

$$H_{02}^{*} + H_{12} : A_{0} = a_{0}^{*}, A_{1} = a_{1}^{*}, B_{0} = b_{0}^{*}, B_{1} = 0$$

$$H_{02}^{*} + H_{23} : x_{1} = a_{0}^{*}, x_{2} = a_{1}^{*}, x_{3} = \text{positive value}, x_{4} = x_{3}b_{0}^{*},$$

$$x_{5} = b_{0}^{*}, x_{6} = 0$$

$$H_{12}^{*} + H_{23} : x_{1} = A_{0}^{*}, x_{2} = A_{1}^{*}, x_{3} = \text{positive value}, x_{4} = B_{0}^{*}x_{3},$$

$$x_5 = B_1 x_3 + B_0, x_6 = B_1$$

Two cases were considered for both examples. In the first case, c_{a}^{R} is fixed, and

$$\begin{array}{c} \mathbf{w}_{i} = \mathbf{w}_{i} = \mathbf{0} \\ \mathbf{U}(\phi) = \max_{i \in [0,T]} |\mathbf{e}_{i}(\phi)| \\ \mathbf{t}_{i} \in [0,T] \\ \end{array}$$

In the second case, c_m^m is varied, and -

$$U(\phi) = \max \{ [\phi_1(\phi)], -W_{\alpha}\phi_{\beta\alpha}, W_{\alpha}\phi_{\alpha}] \}$$

$$(\phi) = \max \{ [\phi_1(\phi)], -W_{\alpha}\phi_{\beta\alpha}, W_{\alpha}\phi_{\alpha}] \}$$

A 9th-order nuclear reactor system was chosen for one example, where a step input is considered so that the power level of the reactor system changes from 90 to 100 percent of the full power (See Bereznai 1971 and Section 3.4.7). T was equal to 10 seconds.

The results, shown in Table 5.8, indicate that the increase in order of the model did not produce any large improvement in U, the minimum value of U, and in this case a model increase is quite wasteful from the computing viewpoint. On the other hand, an improvement in the transient error at a slight expense on the steady-state error is obtained.

Another system considered was the 7th-order control system problem mentioned in Section 5.4. T was equal to 8 seconds though the responses shown in Figs. 5.5-5.7 were taken up to 20 seconds. c_m^5 was equal to 0.11111. The results are summarized in Table 5.9.



5.5.5 Discussion

j.

The results indicate that when c_m^m is fixed increasing the order of the model does improve the transient errors, and it has been shown in Section 5.4.2 that for the third-order model both the 5-ripple and 6ripple solutions satisfy the necessary minimax optimality conditions. It is interesting to note that in all the cases considered, the thirdorder model gives the best result corresponding to the same transient error and three different steady-state errors. Some of the optimal parameters when c_m^m is fixed tend to have nearly zero real parts which may make the model oscillatory. Using appropriate parameter constraints (as indicated in an earlier section) satisfactory results can be obtained which would guarantee a minimum damping of the model for a step imput.

$H_{02} 2.9234 0$ $H_{12} 2.7018 0$ $H_{12} 2.7018 0$ $H_{23} 2.4040 0$ $C_{m}^{m} varied$	Case	Model	1000 Ŭ	1000 max [-ee]
$H_{12} = 2.7018 = 0$ $H_{23} = 2.4040 = 0$ $H_{23} = 0.0172 = 0.2166$	fixed	H ₀₂	2.9234	0
H ₂₃ 2.4040 0	at c ^s	H ₁₂	2.7018	0
c ^m varied		H ₂₃	2.4040	0 .
	varied			
	w_ = 1 S_ =S_ =c_ fu= c_		:	,

RESULTS FOR NUCLEAR REACTOR SYSTEM MODELLING





¥.





در ا





TABLE	5.	9
-------	----	---

Case	Mode1	1000 Ŭ	1000 max [-e ₁ , e ₁]	Fig.
•	H ₀₂	3.7635		5.5
c_{ω}^{m} fixed at	H ₁₂	2.4872		5.6
c [¶] for t _i =T	H ₂₃ (6 ripple) (5 ripple)	1.0207 1.2140	, ,	5.7
c varied	H ₀₂	4.1656	4.1656	5.5
w _{en} = 1	H ₁₂	4.1582	4.1582	5.6
S _t =S _u =c ^s	H ₂₃	1.0201	0.91785	5.7
c ^m varied	, H ₀₂	7.7657	7.6945	
$w_{m} = 10^{6}$	H ₁₂	7.8624	× 10 ⁻⁶	· - · ·
5 ₂ =0.11061 5 ₁ =0.11161	H ₂₃	L 0201	9.8483 x 10 ⁻⁷	- -

RESULTS FOR SEVENTH-ORDER SYSTEM MODELLING

.5.6 Conclusions

The lower-order modelling of high-order systems for minimax objectives has been considered in detail, and the grazor search method has been critically compared with efficient minimization methods for least pth objectives. The grazor search method is very reliable, and the Fletcher method has been observed to be both reliable and efficient. The ideas proposed in this chapter make it possible to automate the modelling procedure, and with the availability of efficient optimization techniques, on-line system modelling and control is entirely feasible. The suggested procedures can be effectively used to get desired optimal models in the minimax sense within user-specified computing times and error allowances.

CHAPTER VI

. DISCUSSION AND CONCLUSIONS

The thesis covers the areas of minimax approximation methods as applied to electrical network design and system modelling in great detail. A reliable algorithm has been proposed and applied to a variety of practical minimax design problems. The method has been critically compared with existing methods for efficiency and reliability, and works very well on most of the problems considered. The philosophy of system modelling is discussed at length, including various techniques involved in implementing the models. Automated modelling and design of highorder systems is shown to be feasible, and the present state of minimax circuit design is considered in detail.

The new ideas presented in the thesis have been verified and used in computer-aided design of a variety of electrical networks subject to different objectives and various constraint specifications. Filters can now easily be designed to meet upper and lower response specifications at predetermined frequencies, within reasonable computing time and desired accuracy. The choice of a circuit model and objective function are as important as the choice of a reliable and efficient optimization technique to give optimal model parameters. If suitable optimization techniques or modelling procedures do not exist for a particular system, the designer is confronted with the task of improving the modelling technique and developing an efficient algorithm to evolve a realistic design. This involves a great deal of system experience and

expertise in the state of the art methods of computer-aided design.

The contributions of this work may be listed as follows.

(1) A new method called the grazor search algorithm has been proposed for minimax objectives. This method has been tested extensively on a number of problems including electrical network design and system modelling.

(2) A practical way of accommodating constraints in the minimax optimization problem has been proposed and applied to some problems.

(3) Methods for investigating a solution for minimax optimality have been proposed and used to test the optimality conditions on a variety of design problems.

(4) The grazor search method has been critically compared on lowerorder minimax modelling of a high-order system with three efficient methods.

(5) Some ideas have been presented for automated system modelling, by means of which the order of the models can be increased in an automated fashion whenever certain criteria are satisfied, and optimality conditions can be directly implemented on the computer. Suitable transient and steady-state constraints can also be taken into account. The proposed approach makes it feasible to automate on-line modelling.

(6) The grazor search method and the method for investigating minimax optimality conditions have been programmed on a digital computer and user-oriented computer program packages have been developed.

It is folt that replacing the present linear search by a more efficient search technique will improve the efficiency of the grasor search algorithm. Further, the concept of automated modelling could be extended to include automated control so that it may be applicable to on-line system modelling and control.



. 119

ŝ

APPENDIX A

GRAZOR SEARCH PROGRAM FOR MINIMAX OPTIMIZATION

A.1 Introduction

The grazor search program is a package of subroutines that optimizes the designable parameters of networks or systems to meet minimax objectives. Full details of the method, including mathematical flow charts and a discussion of computational experience, have already been covered in Chapters III, IV and V. A computer program written in Fortran (Version 2.3 and Scope Version 3.3 for the CDC 6400 computer) is listed at the end of this Appendix.



A.2 Nomenclature

The following is a list of some of the arguments and important variables of the grazor search package as indicated in the flow charts of Figs. 3.2-3.4.

a scale factor for determining the magnitude of the parameter step to be taken at the end of linear program
 a initial specified value of a, previous value of a which gave a satisfactory improvement
 a minimum allowable a
 β reduction factor for a
 γ factor of the step Δφ⁰ which gives the best new point, when starting from φ⁰

number of discrete maxima under consideration, k_r , is increased by one (if $k_r \leq n_r$ -1) or set equal to one (if $k_r = n_r$) if the improvement of the objective function at the new best point as compared to the value at the previous point is less than this quantity

ε'

η

 $-\hat{\psi}_i$

U,

ε

main program is eventually terminated if the improvement of objective function at the new best point us compared to the value at the previous point is repeatedly less than this quantity specified factor of the initial interval of linear search which determines the final resolution between two internal points of the search

current point

starting point, current best point

increment from ϕ^0 which gives the first improved point obtained in each iteration on entering the linear search

ith sample point

sample points corresponding to the y

Ψ_{oi} sample points corresponding to the y_{oi}

DERIV logical variable; if .TRUE. the Vy are calculated, otherwise they are not calculated

ji identifies the ith highest of the yoi

k dimensionality of parameter space

n_r available number of discrete local maxima y_i

value of the objective function at (

 $U_{\phi O}$ value of the objective function at ϕ^{O} y_i function value at ϕ_i for a given ϕ y_i ith highest discrete local maximum ∇y_i gradient of y_i with respect to ϕ γ_{oj} discrete local maxima implied by the y_i TERMlogical variable, initially set to .FALSE., is reset to .TRUE.only if there are failures or improvements in objective function valuetion valueless than c^* after considering values of k_r froml to n_ in one complete cycle.

A.3 Program Description

The user may call the package from his own program as follows. CALL GRAZOR (ALPHAO, ALPMIN, BETA, EPS, EPS1, ETA, PHO, PSI, K, KR, N,

NR, UPHO; TERM)

The variables in the argument list are: FORTRAN Name Variable ALPHAO ao ALPMIN & BETA EPS c EPS1 c' ETA n PHO 60 FSI \$1



K k KR k N n NR n UPHO U_{\$0} TERM TERM

The input variables are a_0 , a_1° , β , ϵ , ϵ° , η , ϕ° , ψ_1 , k, k_r and n while the output variables are a_0 , ϕ° , k_r , n_r , $U_{\phi o}$ and TERN.

It was convenient to place the following user-specified variables

COMMON/GRZR/NCOUNT, IPRINT, UNIT, IOPT, IDATA

in

- NCOUNT number of function evaluations at any stage of the iterative cycle of grazor, is initially set to zero by the user.
- IPRINT logical variable which, if .TRUE., enables all intermediate and final results to be printed out, and no print-outs otherwise.
- UNIT integer variable specifying the data set reference number of the output unit.
- IOPT integer variable denoting the number of times grazor search package was called by the user, is set to zero initially by the user.
- IDATA logical variable which, if .TRUE., enables the input data to be printed out; otherwise not.

Fig. A.1 shows a typical main program for calling the package and the form of a typical analysis program while Fig. A.2 shows typical print-outs of the package.



124

.

U

C	*****
C	A TYPICAL MAIN PROGRAM FOR THE GRAZOR SEARCH ALGORITHM
C	
	DIMENSION PHO(15/ PSI(11)
	LUGICAL TERM, IPRINI, IDATA
	COMMON (GR7D COUNT IDDING UNDE IDDE
c	TYPICAL INDUT VALUES SULLOW
<u> </u>	ALDHAMET.
	ACTALIA.
,	NCOUNT=0
	10PT=0 .
с	
Ċ	INPUT VALUES FOR THE SPECIFIC PROBLEM FOLLOW
	IPRINT=.TRUE.
	IDATA=+TRUE+
	UNIT=6
	EPS=1+0E-03
	EPS1=1+0E-06
	K=2
	N=11
	PHO(1)=1.
	PHO(2)=3.
	PSI(1)=0+5
•	DO 1 I=2;N
Ċ	PSI(I)=PSI(I=1)+0.1
č	MINIMAX OPTIMIZATION STARTS
-	DO 2 1=1,100
•	CALL GRAZOR (ALPHAQ, ALPHIN, BETA, EPS, EPS1, ETA, PHU, PS1, KOKK,
	IN, NR, UPHO, TERM)
C	***************************************
	IF (TERM) GO TO 3
2	CONTINUE
3	STOP
-	END
C	
C	A TURTERI ANALMETE ROUTIAN FOR CRATTIC STATES A CONTRA
	A TYPICAL ANALTSIS PROGRAM FOR GRAZOR SEARCH ALGURITHM
Ļ	FULLUNG
	DINENSION DUC(1) CRADV(1)
	LOGICAL DEDIN
c	THE VALUE OF V AT A SINGLE SAMPLE DOINT F IS CALCULATED.
č	HERE
-	IF(.NOT_DERIV) RETURN
C (THE DERIVATIVES GRADY(1) GRADY(2) GRADY(K) OF THE
č	FUNCTION Y WITH RESPECT TO PARAMETERS PHO(1) PHO(2)
с	PHO(K) ARE CALCULATED HERE
	RETURN
	END

Fig. A.1 Typical main program and analysis program for the grazor search package

• •

IF IPRINT IS .TRUE.

Fig. A.2 (a) Typical printout if IDATA is .TRUE. (b) Typical printout

7

THE FOLLOWING IS A LIST OF IMPUT DATA

Ę

A.4 Subprograms

The subroutine ANAL(ϕ , ψ_i , DERIV, k, y_i , ∇y_i) is a user-supplied analysis program to evaluate y_i and/or ∇y_i at a given point ϕ . If DERIV is .TRUE., the ∇y_i are calculated, otherwise they are not calculated.

The following subroutine need not be written by the user, but is part of the grazor search package. The function subprogram $Y(\phi, \psi_i, k)$ calculates the y_i corresponding to the point ϕ by calling ANAL. The subroutine LOCATE $(\phi, \psi_i, k, n, U_{\phi})$ evaluates the objective function U_{ϕ} by calling $Y(\phi, \psi_i, k)$ for i = 1, 2, ..., n. The grazor search package also uses a linear program solving routine called SIMPLE (see Subroutine SIMPLE), which is a modified version of a program documented with the SHARE Distribution Agency, and written by R.J. Clasen (Reference No. SDA 3384). Section A.7 includes a listing of this subroutine.

A.5 Comments

1.cr

As it stands the package has been programmed to handle up to 15 variable parameters and 15 ripples. The choice of input parameters including scale factors may be critical to efficiency of the algorithm, and the grazor search strategy should be well-understood before the user attempts to use this program.

This program was run and tested on a CDC 6400 computer. The Fortran deck consists of 901 cards which includes detailed comments at appropriate places. The package requires roughly 20,000 octal units of computer memory.

A.6 Discussion

The grazor search algorithm has been programmed in such a way that it allows a certain amount of flexibility to the user. Thus, when GRAZOR is called once, one complete iterative step of the algorithm results, and by introducing GRAZOR in a DO loop, the user has the complete freedom to make his own decision about termination subject to his own convergence criteria, or printing out intermediate results according to a preferred format, or branching out to another optimization package if desired. Appropriate diagnostic messages are provided in the program wherever necessary.

As this is a gradient strategy, it is important that the grad-



A.7 Grazor Search Fortran Program Listing

ļ

•··· = ····-		
AUTHORS	J.W.BANDLER AND T.V.SRINIVASAN.COMMUNICATIONS RESEAR LABORATORY AND DEPARTMENT OF ELECTRICAL ENGINEERING. MCMASTER UNIVERSITY.HANILTON.ONTARIO.CANADA	514
THE GRATC	NO SEADCH ALCONTRACT	
USER SO T	HAT THE ALGORITHM HAS TO BE PLACED IN A DO LOOP BY TH	1E
NECESSARY	TO GET SATISFACTORY THOUSENED AS MANY TIMES AS IS	
FUNCTION	STATISTICS ACTORY INPROVENENTS IN THE OBJECTIVE	
A TYPICAL	MAIN PROGRAM FUR GRAZOR SEARCH ALGORITHM FOLLOWS-	
	PH0(15)+PSI(11)	
INTEGED //	ERM FIPEINT FIDATA	
COMMONIA	TO INCOMENT TOUTHE AND ADDR	
TYPICAL I	NPUT VALUES FOLLOW	
ALPHAO=1.		
ALPHIN=1.	0E-06	
RETA-10.		
ETA+0.01		
KR=1		
NCOUNTEO		
10PT=0	· · ·	
INPUT VAL	UES FOR THE SPECIFIC PROBLEM FOLLOW	
108181=+1		
LUAIATAIN		
EPS=1.0F=	01	
EP51+1.UF	-06	
K#2		
N=11		
PH0(1)=1.		
PH0(2)=3.		
PSI(1)=0.	5	
DO 1 1=2+	N ·	
PSI(1)=PS	[[]-1]+0,1	
	· · · ·	
MININAX O	PTIMIZATION STARTS	
LALL GRAZI	URIALPHAU+ALPMIN+BETA+EPS+EPS1+ETA+PHO+P51+K+KR+N+NK+U	Рно
IF(TFRM) (50 TO 1	
CONTINUE		
STOP	· .	
END		

A TYPICAL	ANALYSIS PRUGHAN FOR GRAZUR SEARCH ALGUNITHM FULLOWS-	
SUBROUTINE	ANAL IPHOSFODERIVSKAY, GRADY)	
COMICAL DE		

THE VALUE OF Y AT A GIVEN SAMPLE PUINT F IS CALCULATED HERE IF(-NOT-DERLY) RETURN 60 . THE DERIVATIVES GRADY (1) + GRADY (2) + + + + + GRADY (K) OF THE FUNCTION 61 WITH RESPECT TO PARAMETERS PHU(1) PHO(2) PHO(K) ARE 62 CALCULATED HERE A 63 RETURN 64 END 65 ٠ . 66 67 A ۸ 68 69 SUBROUTINE GRAZOR (ALPHAD . ALPHIN. BETA. EPS. EPSI. ETA. PHO. PSI. K. KR. N. 70 INR+UPHO+TERM1 71 72 73 THE USER HAS TO SPECIFY VALUES FOR ALPHAU, ALPHIN, BETA, EPS, LPS1, LTA 74 +PHO+PSI+K+KR+N 75 ۵ 76 STARTING VALUES -----٨ 77 ALPHA0=1. A 78 BETA=10. A 79 A 80 KR=) SUGGESTED STARTING VALUES ۸ 81 ALPMIN=1.0E-06 A 62 A 83 ETA=0.01 EPSIATHE MINIMUM IMPROVEMENT IN THE OBJECTIVE FUNCTION BETWEEN ۸ 85 SUCCESSIVE ITERATIONS+MUST BE SPECIFIED BY THE USER 86 EPS=EP51=1000. . 87 THE FOLLOWING COMMON STATEMENT IS TO BE SPECIFIED BY THE USER A 24 COMMON/GRZR/NCOUNT+IPHINT+UNIT+IOPT+IDATA ۸ YO A 91 NCOUNT-NUMBER OF FUNCTION EVALUATIONS AT ANY STAGE OF THE A 92 ITERATIVE CYCLE OF GRAZOR ۱A 93 NCOUNT IS INITIALLY SET TO ZERO BY THE USER . 94 TOPT CORRESPONDS TO AN ITERATIVE CYCLE OF THE GRAZOR SEARCH ALGORI THH AND IS THE NUMBER OF TIMES OPTIMIZATION PACKAGE GRAZOW HAS A 95 8 96 BEEN CALLEU. TOPT IS INITIALLY SET TO ZERO BY THE USER ۸ ¥7 IF IPRINT IS .TRUE. ALL INTERMEDIATE AND FINAL RESULTS ARE TO BE A ٧W PRINTED OUT OTHERWISE THERE ARE NO PRINT-OUTS 99 ۸ UNIT IS AN INTEGER VARIABLE SPECIFYING THE DATA SET REFERENCE ۸ 100 NUMBER OF THE OUTPUT UNIT 101 A IF IDATA IS .TRUE. THE INPUT DATA IS PRINTED OUT.OTHERWISE NUT A 102 THE USER HAS TO SPECIFY VALUES FOR IPRINT UNIT . IDATA 101 ¢ . 104 192 THE VARIABLES PSI AND PHU HAVE TO BE DIMENSIONED IN THE CALLING PH A 106 OGRAM CURRESPONDING TO NAXIMUM VALUES OF NOAND KIELD ARESPECTIVELY A 107 THE USER HAS TO INDICATE IN HIS MAIN PROGRAM THAT TERMOIPHINT. A 108 IDATA ARE LOGICAL VARIABLES AND THAT UNIT IS AN INTEGER VARIABLE A 104 IF TERM IS .TRUE. AT THE END OF AN ITERATIVE CYCLE OF GRALOR. THE USER MAS TO DECREASE THE VALUES OF ALPHIN AND ETA DEFORE A 110 A 111 GRAZOR CAN BE CALLED AGAIN IN THE MAIN PROGRAM . -112 THE USER HAS TO FURNISH SUBROUTINE ANAL FOR IMPLEMENTING THE 111 ۸ GRAZOR SEARCH STRATEGY A-114 A 115 ۸ 114

٠.

C

c

c

C

c c

С

¢

C

C

C

c

c

c

С

C

C

C

÷€

¢

C

C

¢

C

¢

¢

c

¢

C

C

C

C

ς

¢

C

C

C

¢

¢

C

C

¢

¢

γς

¢

¢

¢

C

C

C

C

C

C

С

٢.

THE FOLLOWING IS A BRIEF SUMMARY OF THE VARIABLES IN GRAZOR------PHO- THE PARAMETER VECTOR. IT IS EITHER THE STARTING POINT ON THE A 117 A 118 PHI= CURRENT PARAMETER VECTOR A 119 KENUMBER OF PARAMETERS PHO A 120 PSI- VECTOR OF SAMPLE POINTS A 121 Nº NUMBER OF SAMPLE POINTS PSI UPHO. OBJECTIVE FUNCTION AT PHO A 122 UPHIN ORJECTIVE FUNCTION AT PHI A 123 A 124 YMAX - VECTOR CONSISTING OF THE LOCAL DISCRETE MAXIMA INPLIED BY THE FUNCTIONS Y ARRANGED IN DECREASING MAUNITUGEOUVER & SAMPLE A 125 A 126 A 127 PSIMAX = VECTOR OF SAMPLE PUINTS CURRESPONDING TO THE VECTOR YMAX A 128 AR NUMBER OF DISCRETE LOCAL MAXIMA YMAX A 129 KR. NUMBER OF DISCRETE LOCAL MAXIMA YMAX UNDER CONSIDERATION-KR IS A 130 LESS THAN OR EQUAL TO NR A 131 GRAD . MATRIX OF FIRST DERIVATIVES OF VECTOR YMAK WITH RESPECT A 132 TO THE PARAMETERS PHO A 133 TERM- LUGICAL VARIABLE WHICH. IF THUE. INDICATES THE LUNVERGENCE OF A 134 THE GRAZOR SEARCH ALGORITHM A 135 ٨ 136 A 137 THE DIMENSION OF SUBSCRIPTED VARIABLES IN GRAZOR CORNESPOND TO A - 1-38 MAXIMUM VALUES OF K+15 AND NR+15 A 139 THE SUBSCRIPTED VARIABLES DUNNY PHI DELPHI DELPHNO DELP ANE A 140 DIMENSIONED CORRESPONDING TO A MAXIMUM VALUE OF K+15 THE SUBSCRIPTED VARIABLES YMAX, PSIMAX ARE DIMENSIONED ٨ 141 A 142 CORNESPONDING TO A MAXIMUM VALUE OF MR=15 A 143 MATRIX GRAD IS DIMENSIONED CURRESPUNDING TO MARINUM VALUES OF A 144 NR=15 AND K=15 A 145 A 146 A 147 THE USER HAS TO SUPPLY AN ANALYSIS PROGRAM AND THE FULLOWING IS A A 148 BRIEF DESCRIPTION OF ITS ARGUMENTS A 149 SUBRUUTINE ANAL (PHO +F + DERIV + K + Y + GRADY / CALCULATES THE VALUE OF A 150 FUNCTION Y AND ITS FIRST PARTIAL VENIVATIVES GRAUVEL G . 151 ۸ 152 ... PHOIR) FOR A GIVEN SAMPLE POINT F A 153 PHO AND GRADY ARE TO BE VARIABLE-DIMENSIONED IN ANAL OR A 154 DIMENSIONED CORRESPONDING TO THE MAXIMUM VALUE FOR KYALA A 155 DERIVALOGICAL VARIABLE WHICH. IF THUE ALLOWS THE GRAU TO BE A 156 EVALUATED. UTHERWISE GRADY ARE NOT EVALUATED A 157 A 154 A 159 DIMENSION PHO(1). PSILLI, DUMNY(15/, PHILS/, YMARLIS/, PSIMARLIS/ A 160 1. GRAD(15,15). DELPHI(15). DELPHN(15). DELP(15). X(31). A(16,31). A INI 28(16), C(32), KO(6), PS(16), JH(16), XK(16), YY(16), PE(16), E(16. A 162 A 163 LOGICAL TERNAIPRINTAIDATA A 164 INTEGER UNIT A 165 COMMON /GRZR/ NCOUNT+IPRINT+UNIT+IUPT+IDATA A 166 10PT+10PT+1 A 167 A IR, INCOUNT-EU-O) TERMA-FALSE-166 A 169 IF (TERM) GO TO 32 ALPHA#ALPHAO A 170 A 171 ALPHAT=ALPHAO ¥ 1,15 ICLOCK=0 A 175

۵

C

C

c

C

C

C

C

C

C

C

C

C

C

C

C

C

¢

C

C

c

C

C

C

C

C

c

C

c

C

¢

C

C

C

¢

C

C

C

C

C

C

C

¢

C

C



132

	ISTOP=0	
1	CALL SELEC (PHO.PSI.PSIMAX.K.N.MP. VMAY)	A 174
	UPHO=YMAX(1)	A 175
	NCOUNT=NCOUNT+1	A 176
	IF (NCOUNT+GE+2) GO TO 2	A 177
	IF (IDATA) WRITE (UNIT-38) ALPHADIAL PHIM. BETA FOR THAT	A 178
	ITERNO (IOPHORIJOINIK)	A 179
2	IF (IDATA) WRITE (UNIT-39) (I-PSI(I)-Int-M) - '	A 180
	IF (IPRINT) WRITE (UNIT+34) IOPT+NCOUNT+UPHO+(PHO+1+++++)	- ¥ 191
. 2\	IF (ICLOCK+GT+1) GO TO 3	- Y 195
	IF JKR-NE-1) GO TO 4	A 183
¥.	KR=1	A 184
1	DO 6 L+1+KR	A 185
- 1	CALL ANAL (PHO+PSIMAX(L)++TRUE++K+YMAX(L)+DURMY)	- 106
	DO 5 1=1,K	- A 187
1	GRADIC+II=DUMAY([)	A 100
2	CONTINUE	A 190
0		A 191
	17 1KK+CG+17 GO 10 22	A 192
		A 193
		A 194
		A 195
		A 196
		A 197
		A 198
	DO 7 Milelar	A 199
		A 200
7	CONTINUE	A 201
•		A 202
8	CONTINUE	A 203
9	CONTINUE	A- 204
	DO 10 I=1-KR	A 205
	A(1+KR))=1+0	A 206
10	CONTINUE	A 207
	DO 12 T=leKR	A 208
	DO 11 J=KR2+KR3	A 209
	0+0+{L+1}A	A 210
	1F (J.FQ.([+KR])) A([,]=1.0	A 211
11	CONTINUE	A 212
12	CONTINUE	. A 214
	DO-13 J=1+KR	A 215
	A(KR1+J)=1=0	A 216
13	CONTINUE	A 217
	DO 14 JUKRI SKRI	A 218
• •	A(KR1+J)+0+0	A 219
14	CONTINUE	A 220
	NO 15 T+1+KR	A 221
		A 222
• 7		N 573
		A 224
		A 225
		¥ 554
16	CUNTINNS CONTINNS	A 227
ċ	Z rush Tuff	A 224
è		A 229
-		A 230

Ś

:

SUBROUTINE SIMPLE IS NOW GUING TO BE CALLED-ANY ALTERNATIVE C CHOICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USER AS LONG AS IT C A 231 PERFORMS THE FOLLOWING OPERATION-C A 232 SUBROUTINE SIMPLE SOLVES A LINEAR PHOGRAMMING PROBLEM OF C A 233 MINIMIZING CAX SUBJECT TO ARX#8+WHERE X+C+B ARE VECTORS OF LENGTH C A 234 KR3+KR3+KR1 RESPECTIVELY+AND A IS A MATRIX OF SIZE KH1+KH3 C A 235 C A 236 SUBROUTINE SIMPLE ATTACHED TO THIS PACKAGE IS A MODIFIED VERSION ¢ A 237 OF A PROGRAM AVAILABLE WITH SHAKE DISTRIBUTION AGENCY, REFERENCE ¢ A 238 NUMBER SDA 3384 AND WRITTEN BY R.J.CLASEN C A 239 THE MODIFIED VERSION IS IN THE NCHASTER UNIVERSITY LATA PHUCESSING C A 240 AND COMPUTING CENTRE LIBRARY SINFORMATION SHEET MILLS 5+7+130 C A 241 A 242 C NA AND IFLAG ARE TO HE SPECIFIED BEFORE CALLING SIMPLE C A 243 IFLAG IS SET EQUAL TO ZERO A 244 ٤ NA IS THE FIRST DIMENSION OF THE ARKAY A AND IS SET EQUAL TO THE C A 245 MAXIBUN VALUE OF NR+11=161 C ** 246 X IS THE VECTOR OF DIMENSION 20NA-1 A 247 C THE FOLLOWING SUBSCRIPTED VARIABLES ARE PART OF THE ARGUMENT LIST C A 244 OF SIMPLE AND ARE TEMPORARY STORAGE SPACES TO BE DIMENSIONED IN A 249 ¢ A 250 THE CALLING PROGRAM (GRAZOR) C PS+JH+XX+YY AND PE ARE TEMPORARY STURAGE VECTORS OF DIMENSION NA C A 251 E IS A TEMPORARY STORAGE MATKIX OF UIMENSION (NA. NA. 2-1) A 252 С KU IS A VECTOR OF LENGTH 6. UPON COMPLETION OF THE EXECUTION OF A 233 ¢ SIMPLE. KOILIS IF THE LINEAR PROGRAMMING PROBLEM WAS FEASIBLE. C A 234 A 255 THE SULUTION LIES IN XIJI.J. J. KR3 C A 256 c · IFLAGED A 257 A 254 NA=16 A 259 CALL STMPLE (IFLAG+KR1+KH3+A+B+C+KU+X+P5+JH+XX+YY+PE+L+NA) A 260 C A 261 00 18 J=14K A 262 DELPHI(J)=0.0 A 263 DO 17 1+1+KR A 264 DELPHI(J)=DELPHI(J)=X(I)=GRAD(I+J) A 265 CONTINUE 17 A 266 18 CONTINUE A 267 C A 268 THE INCHEMENTAL PARAMETER STEP DELPHI IS NORMALIZED TO UNIT C A 269 LENGTH BY SUBROUTINE NORM C C A 271 19 CALL NORM (K.DELPHI.DELPHN) A 272 C A 273 THE LINEAR SEARCH BEGINS A 274 ALPHA IS A SCALE FACTOR FOR DETERMINING THE MAGNITUDE OF THE C A 275 NORMALIZED STEP DELPHIN TO BE TAKEN FOR THE LINEAR SEARCH C A 276 C ALPHAU IS THE INITIALLY SPECIFIED VALUE ME ALPHA OR THE PHEVIOUS A 277 VALUE OF ALPHA WHICH GAVE A SATISFACTURY IMPROVEMENT C A 278 ALPHIN IS THE MINIMUN ALLOWABLE ALPHA C A 214 IF (ALPHA.LT.ALPHIN) ALPHA-ALPHIN A 280 20 DO 21 1=1-K A 281 PHI(1)=PHO(1)+ALPHA+DELPHN(1) A 282 21 CONTENUE A 283 GO TO 24 A 1284 A STEP TAKEN IN THE NEGATIVE GRADIENT DIRECTION OF HIGHEST C NIPPLE A 285 DO 23 1=1.K 22 A 286 DELPHI(1)=-GRAD(1+1) A 207
*****?

	_
	-
	_
	- 3

23	CONTINUE			
	GO TO 19			
-24	CALL' LOCATE (PHI PSI at a Mandala A		200	
	NCOUNT=NCOUNT+1		269	
	IF TUPHIALTAUPHOL ON TO 28		290	
	IF LALPHA-EQ-ALPHINI GO TO TO	- 7	~~¥¥	
C		- 2	272	
C	ALPHA REDUCED BY FACTORS OF SEA		493	
C	THE THE THE THE THE THE	- 7	299	
	ALPHA#ALPHA/BETA	- 7	493	
	IF LALPHAALFAALPHINT ALDHAAAAAAAA	- 7	470	
	GO TO 20	- 7	204	
25	PO 26 1=1+K	- 2	200	
	DELP(I)=ALPHADDELPHAT(I)		100	
26	CONTINUE	Â	300	
٠.	. 8	Â	102	
c	DELP- INCREMENT FROM PHO WHITH CIVES DUE OF	- Â	101	
C	OBTAINED ON ENTERING THE LINE AN OVER THE FIRST IMPROVED POINT		304	`
C	ETA IS THE SPECIFIED FACTOR OF THE ALL SCARCH		305	
C	SEARCH WHICH DETERMINES THE THE INITIAL INTERVAL OF LINEAR	Â	50.6	
C	PUINTS OF THE SEARCH STILL THAL RESOLUTION BETWEEN TWO INTERNAL	Â	307	
C		Ä	JOB	
	FINTHETA	Å	309	
	CALL GOLDEN (GAMASEINT BHIL DWL DWL DWL DWL DWL DWL DWL	A	310	
C	the series of th	A	311	
C	TERM IS SET TO TRUE AND GRAZOH METURA TO THE		312	
C	UPHO-UPHI IS REPEATEDLY LESS THAN FOUND THE CALLING PROGRAM IF	A	313	
C	Contract Cross Lines En21		314	
	IF (LUPHO-UPH1)+LT+EPS1) GO TO NO		315	
	ISTOP+U		310	
	DO 27 1+1+K		317	
	PH0(1)=PHI(1)	•	314	
27	CONTINUE		71A	
	IF (IPRINT) WRITE (UNIT-37) IOPTANCIUMT-INNE		320	
C			321	
C	IF THE OBJECTIVE FUNCTION UPHI AT A NEW POINT WHILLS LEFT THESE	.	755	
C	VALUE UPHO AT THE PREVIOUS PUTAT PHE BY A VALUE GREATER THE		323	
Ç	EQUAL TO EPSOTHE NEW POINT IS CONSIDENTED A CATION OF	A	154	
C	INPROVEMENT IF NOT KR IS INCHEMENTED BY I FOR TO THE THE	A	142	
C	EQUAL TO NR-1) OR SET EQUAL TO 1 (Full EROME)	<u>.</u>	326	
C		A .	327	
	IF (10PH0-0PH1)+LT+EPS) 40 TO 31		328	
	HPHO=UPHI	A .	329	
	ALPHAO-ALPHAOGANA	A .	330	
	60 TO 33	A	<u>, , , , , , , , , , , , , , , , , , , </u>	
20		.		
	1F (1STOP+EU+0) GO TO 1	.		
	IF (ISTOP+LE-NR) GO TO 3	A .		
	TERN=.TRUE.			
	WRITE (MIT.75) EPS1			
	GO TO 13		///	
79	TP (KR+EQ+NR) GO TO 28		73 4 134	
			737 148	
		A . 1		
	10 115100-10-01 GO TO 1			
30		Â	43	
•••	AL FTA . AL FTA .	Â		

15TOP=1STOP+1 A 345 GO TO 29 A 346 UPHO=UPHI 31 A 347 ALPHA=ALPHAT A 348 GO TO 29 A 349 WRITE (UNIT+36) 32 A 350 11 RETURN A 351 C ۸ 352 A 353 A 354 FURMAT (#1 #/43X+# THE GRAZON SEANCH STRATEGY FOR MINIMAX OBJECTLY 34 A 355 1F5 +/43X++ ----*//6X. A 356 A 351 SINIMAX UBJECTIVE FUNCTION++10X++VARIABLE PARAMETER VECTUR+//15X++1 A 336 40PT++30X+*NCOUNT++30X+*UPHU#+32X+*PHO#////14X+15+30X+15+23X+14+8+ A 359 518X+F16+8/(111X+E16+8)) A 360 35 FORMAT (* TERM++TRUE++IMPROVEMENT IN UBJECTIVE FUNCTION LESS THAN A 361 1EP51=*+E16+8) A 362 FURMAL (* THE GRAZOR SEANCH RUUTINE CANNUT RESTART AS TERM IS EVUA 36 A 363 IL TO STRUES FROM THE PREVIOUS ITERATIONS AND CONVERGENCE CRITERIUM A 364 2 HAS BEENATA REACHED FOR A SPECIFIED EPSISTHE UNLY WAY TU RESTART OF JOD 315 TO DECREASE THE VALUES OF ALPHIN AND ETA. ++ A 366 FORMAT (//14X+15+30X+15+23X+E14+8+14X+E14+8/4111X+E16+8/ 37 A 367 FORMAT 149X+* THE FOLLOWING IS A LIST OF INPUT DATA*/49X+*---38 A 368 1-----*//55X+*ALPHAO ##+E16+#755X+#ALPH A 364 21N -++E16+8/55X+PBETA --+ 216-4/552+=225 +*+E16+#/55X+*E A 370 # =*+E16+#/55K+*LTA 3251 ----**+15/55X+*K A 371 =++15/55X++N 4H **+15/35#+#TENA **+L>//155X+*PHU(* A 312 5+13++1 +++E16+811 A 373 FURMAT (/(8X+*PS1(*+13+*)+*+16+8+5X+*P51(*+13+*+*+16+8+5X+*P51(39 A 314 1*+13+*1**+16+8+5X+*PS1(*+13+*)+*+L16+8)/ A 375 END A 376 A 377 A 376 A 374-14 1 ь 2 SUBRUUTINE SELLC (PHISPSISPSINARSKAR, MARYMAR) ы 8 4 в > IN THIS SUBROUTINE THE RIPPLES OF THE FUNCTIONS Y AT A PUINT PHE ы OVER N SAMPLE POINTS PSI ARE EUCATED AND SURTED GUT IN DECHERSING ь 1 MAGNITUDE 8 6 ITAGOMMAXOMAX ANE DIMENSIONED CONNESPONDING TO A MAXIMUM VALUE ø * OF NR=15 8 10 MAX & DISCRETE LOCAL MAXIMA IMPLIED BY THE FUNCTIONS Y AT A PUINT ۲ 11 PHI AS SAMPLING PROCEEDS FROM PSILLE TO PSILM? 8 14 PMAX+ SAMPLE POINTS CORRESPONDING TO MAX 13 . 16 . 15 DIMENSION PSILLIS PHILLS YMAXLIS ASIMAALIS ITAGLISS PHAALISS . 1.0 IMAKE151 11 -REAL MAX . 18 NR=1 19 PHAX(1)=P51(1) . 20 MAX(1)=Y(PH[=P51(1)=K) 21 . YZ=MAX(1) 22

C

C

C

C

C

C

C

C

۲

¢

C

C

C

C

C

C

C

C

C

KD=1 8 23 DO 4 1=2+N YXAY (PHI +PSI(1)+K) 8 24 ۲ 25 IF (YX-YZ) 1.1.2 8 26 1 KD=-1 В 27 GO TO 3 B 28 ÷ YMAXT=YX > Ð 29 LT+E 30 B IF (KD.EQ.-1) NR=NR+1 ы 31 KD=1 ы 2د PMAX(NR)=PSI(LT) B 33 MAX (NR) = YMAXT ы 34 ٩ YZ=YX ₿ 35 CONTINUE 4 C C H. 36 37 8 8 38 CALL TOSORT (MAX+ITAG, NK+1) 92 Ľ۵ 00 5 J=1+NH ø 4Ú LD=ITAG(J) B 41 MAX(J)=MAX(LD) 42 в PSTMAX(J)=PMAX(LD) 8 43 CONTINUE 5 ь 44 RETURN 45 B END B 46 ¢ 8 47 C 8 40 C 49-8 ¢ С 1 C ł ¢ SUBROUTINE GOLDEN (GANA, ETA, PHI, PHU, DELP, PSI, K. N. UPHI, UPHO) C 3 C C 4 ¢ C 5 THIS SUBRUUTINE USES THE GULDEN SECTION SEARCH TO FIND THE GAMA C L ٠ c CURRESPONDING TO THE MINIMUM OF THE OBJECTIVE FUNCTION AT THE 1 ς POINT PHO+GAMA+DELP C C, . C PHIA PHIB PHIU ARE DIMENSIONED CORRESPONDING TO A MAXIMUM VALUE C ¥ OF K=15 C ¢ 10 ¢ C 11 C C 12 COMMON /GRZR/ NCOUNT+IPHINT+UNIT+IUPT+IDATA C 13 LOGICAL IPHINT. IDATA C 14 INTEGER UNIT C 1> DIMENSION PHOEIS PHILLS ULLPELS PHIALSS PHIBLESS PHILLSS ċ 1.9 151(1) 11 C TAU=0-5+(1-0+(5-0)++0-5) C 14 TAUSO-TAUSTAU C 19 ETA+ETA+(TAU+1+) C 20 GAMAL+0+ C C 21 GAMAU=1+0 ٤Ż Č GAMAA+0+0 23 UPHIA=UPHO 24 1 00 2 1=1+K C 42 PHIUEII=PHOEII+GAMAU+DELPEII C 26 ¢ 2 CONTINUE 27 CALL LOCATE (PHEUsPSIsKsNaUPHEU) NCOUNT+NCOUNT+1 Ĉ 48 29 IF IUPHIU-LE-UPHIAI GO TO 4 C 30

	GAMAN-GAMAL+IGAMAU-GAMALI/TAU	_		
	DO 3 1=1+K	Ç	١٤	
	PHIA(E)=PHO(E)+UELP(E)+GANAA	ç	32	-
٦	CONTINUE	c	دد 34	
	GAMAA	č	35	
•		ē	34	
		č	37	•
		ē	34	
		č	24	
_		č	40	
•		Ē	41	
	PRIATISEPHOTIS+DELP(I)+GANAA	ē	<u>.</u>	•
6	CONTINUE	č	41	
	CALL LOCATE (PHIA+PSI+K+N+UPHIA)	è		-
	NCOUNT #ACOUNT + 1	- Z		
-	GO TO 9	ē	44	
'	DO 8 I=1+K	Ē	41	
	PH1B(LI=PHO(L)+DELP(LI=GAMAB	-ē	6 M	
8	CONTINUE	Ē		
	CALL LOCATE (PHIB+PSI+K+N+UPHIB)	è	50	
	NCOUNT=NCOUNT+1	2	51	
9	IF ((GAMAB-GAMAA)+LT+ETA/ GO TO 11	- Z	42	
	IF (UPHIA-UE-UPHID) GO TO LO	- 2	55	-
	GAMAU - GAMAB	Ē	24	
	GAMAR-GAMAA	- 2	55	
	UPH1B+UPH1A	- č	56	4
	GAMAA=GAMAL+(GAMAU-GAMAL)/TAUSU	č	\$7	12
	GO TO S CONTRACTOR STOLEN	č	58	
10	GAMAL=GAMAA	č	54	
	GAMAA=GAMAH	č	60	
	UPH LA=UPH [8	Ē		
	GAMAH=GAMAL+(GAMAU-GAMALI/TAU	Ē	6 4	
	GO TO 7	č	A 3	
11	1F (UPH1A+LT+UPH1B) GO TO 12	ē	64	
	GAMA=GAMAH	ē	65	
	UPH1=UPH18	č		
	GO TO 13	ē	67	
12	GAMA-GAMAA	č		
	UPHIAUPHIA	Ē.	64	
13	DO 14 I=1+K	č	10	
	PH1(1)=PH0(1)+GAMA+UELP(1)	÷Č	11	
14	CONTINUE	ċ	72	
C	· · · · · · · · · · · · · · · · · · ·	ć	73 -	
C	THIS VALUE OF GAMA IS THE FACTOR OF THE STEP DELP WHICH GIVES THE	ć	16	
C	AEST NEW POINT-WHEN STANTING FROM PHU	č	15	
C		Ē	7.	
	RETURN	Č	11	
	END	C	78	
C		¢	79	
C		ς	80	
C	·	C	81-	
Ç		D	1	
C	***************************************	Ð	4	
	SUBROUTINE LOCATE (PHI+PSI+K+N+UPHI)	Ų	3	
¢		5	•	,
¢		9	>	
C	LOCATE CALCULATES THE MINIMAX UNDECTIVE FUNCTION OF THE Y AT A		•	

POINT PHI UVER A GIVEN SET OF SAMPLE POINTS PSI D 7 D 8 D 9 DIMENSION PHI(1), PSI(1) D 10 DO 1 1=1+N D 11 YT=Y(PH1+PSI(I)+K) D 12 IF (1.EQ.1) UPHI=YT D 13 IF (YT.GT.UPHI) UPHI=YT D 14 CONTINUE 15 U RETURN D 16 FND D 17 Ď ۵đ ₽ 19 ų 20-E 1 É 2 FUNCTION Y (PHI+F+K) E 3 E 4 HERE THE FUNCTION VALUE Y AT A POINT PHI CURRESPONDING TO A SAMPLE 5 E POINT F IS CALCULATED £ 6 DUMMY HAS BEEN DINENSIONED CORRESPONDING TO A MAXINUM VALUE OF 7 Ł K=15 Ë 8 Ł ¥ DIMENSION PHILLI £ 10 CALL ANAL (PHI +F++FALSE++K+Y1+DUNHY) E 11 Ē Y=Y1 12 RETURN 13 È END 14 Ł 15 £ 14 17-E F Ł , 2 ۶ 3 SUBROUTINE NORM (K+W+WN) F ٠ DIMENSION WILLS ANILLS SUN=0. F 3 ş 6 00 1 1+1+K 7 F SUN=SUN+W(1)+W(1) F 8 CONTINUE F ¥ SUMRT+SURT(SUM) F 10 DO 2 1+1+K 11 P WHILL HELL FURT F 12 CONTINUE 13 ĸ RETURN ø 14 FND F 15 F 1. ø 17-L 6 G 2 6 3 SUBROUTINE TOSORT (A.I.N.M.) 6 4 5 6 SUBRUUTINE TOSURT EMAX+ITAGENIERNI FURINS A VECTOR OF TAGE ITAG SU 10 ٠ THAT ITAGELISITAGELISSON STALING ANE UNDERED SUBSCRIPTS UP 6 7 VECTOR MAX SUCH THAT MAXIITAGIL ! . MAXIITAGIZ ! MAXIITAGINA !! 6 6 6 ARE IN ALGEBRAIC ORDER

c

C

Ċ

1

C C C

c

C

C

C

۲

c

c

C

C C

¢

c c

1

2

C

¢

٢

¢

С

C

C

C

C

C

¢

138

Ø

MH IS POSITIVE FOR A HIGH TO LUW ORDERING AND NEGATIVE FOR LUW TO HIGH ORDERING 10 THIS SUBROUTINE LISTING WAS OBTAINED FROM THE DATA PROCESSING AND G 11 COMPUTING CENTRE + LIBRARY INFURMATION SHEET MILLS 5+3+34+ MCMASTER G 12 **.** 13 UNIVERSITY G 14 b 15 G DIMENSION ALL'S ILL 16 G LOGICAL HILO.TIME1 17 HILO-M.LT.O 6 10 Ģ 19 G 20 00 1 J=1+N ų, 21 G 22 G 23 G 24 DO 6 K=1+N2 G 25 TIME1=.TRUE. 4 20 DO 4 J=1+N G 27 IF II(J)+GT+D) GO TO 4 G IF (-NOT+TIME1+ GO TO 2 20 G 44 TIMEL=.FALSE. í, 30 SMALL=RIG=A(J) G 31 G 32 G 33 IF (ALJ)+GT+SHALL) GO TO 3 6 SHALL=A(J) G 35 ų, 36 IF (A(J)+LT+BIG) GO TO 4 Ġ 37 G 36 6 39 G 40 G 41 IUBI+IABS(I(JH)) G 42 TFJS7=TABSETEJS17 G 43 IF (HILO) GO TO S G -I(L)=ISIGH(JS+I(L)) 6 45 I(K)=ISEGN(JB+E(K)) G 44 G 47 I(L)=ISIGN(JB+I(L)) G - 44 1 (K)=151GN(JS+1(K)) 6 49 G 50 6 51 G 32

C C

c c

c c

C

l

2

3

5

6

¢

C

¢

C

C

c

\$

C

C

٠.

N1=N+1

N2=N1/2

[(J)=-1

JS=JR=J

GO TO 4

BIGHALJ

CONTINUE

L=N1+K

GO TO 6

CONTINUE

RETURN

END

JS=J

JB=J

CONTINUE

************************* SUBROUTINE SIMPLE (INFLAGORIANAAABACAKUAAUAPAJNAIAYAPEALARA) CDC 6400 1172 GLTAL WONDS ARE REQUIRED THETC SIMPLE REP AUTOHATIC SIMPLEX REDUNDANT EQUATIONS CAUSE INFEASIBILITY REAL AINA.1361 REAL 811,+C(1)+P(1)+X(1++Y(1++PE(1++E(1) INTEGER INFLAGOREONNOROLOFORBELLOUPICLE EQUIVALENCE (XX+LL) THE FULLOWING DIMENSION SHOULD BE THE SAME HERE AS LT IS IN

÷

ŝ,

139

ú >>

6 54

G

et

н

H

н

H

H

м

Ħ

н

Η, 10

H 11

>>-

A.

2

3

٠

>

.

7

•

.

C CALLER. 12 REAL AA+AIJI+88+COST+UT+RCUST+TEXP+THIV+TY+XULD+XX+X++1+YHAX н INTEGER I. I.A. INVC. IR. ITER. J. J. T. K. KUJ. L. L. L. M. M. C. INA. н 11 н 14 INTEGER NOUT , NPLY , NUNVR , NVER н 15 LOGICAL FEASAVERINEGATRIGAKQAABSC н 16 c н 17 SET INITIAL VALUES. SET CONSTANT VALUES C H 18 ITER=0 н 19 NUHVR=0 н <0 NUMPV=0 н ۷1 N=HX н 22 N=NN н 23 TEXP=+5++16 н 24 NCUT=4##+10 н 25 NVER+#/2+5 н 26 M2=H++2 H 27 FEAS=+FALSE+ н ... IF LINFLAG.NE.01 GO TU 3 н 44 INEW! C+ START PHASE ONE WITH SINGLETON BASIS н 30 DO 2 J=1+N H 31 KB(J)=0 н 32 KQ=.FALSE. н 33 DO 1 1-1+M н 34 IF (A(1+J)+E0+0+0) GO TO 1 H 35 IF IKU.OH.ALI.J.L. O.O. GO TO 2 н at KQ=.TRUE. 51 н CONTINUE 1 н 38 K8(J)=1 4 н 39 CONTINUE 2 а н 40 3 DO 4 [=1+M . н 41 JH(1)=+1 н 42 CONTINUE 4 H 43 C+ IVER! CREATE INVERSE FRUM 1KB1 AND 1JH1 (STEP 7) н 44 VER-TRUE. 5 Ħ 45 INVC+U н 44 NUMVR=NUMVR+1 н 47 TRIG++FALSE+ н 48 00 6 1=1+MZ Ħ **A** 9 E(1)=0.0 н 50 CONTINUE 6 н. 51 HH=1 н 52 ٦ 00 7 1+1+H н 53 E (PMI)=1.0 н 54 PE(1)=0.0 Ħ 55 X(1)=8(1) н 56 $\mathbf{H} \geqslant$ 51 IF (JH([]+NE+0) JH([)=-1 MH=MH+H+1 Ħ 58 CONTINUE н 59 7 ¢ FORM INVERSE н •0 DO 14 JT=1+N Ħ •1 н 62 IF (KB(JT)+EQ+0) GO TO 14 GO TO 30 H 43 н 44 C 30 CALL JMY 65 H C CHOOSE PIVOT TY=0.0 н 41 н KO-.FALSE. 00 13 1=1+M H ...

IF (JH(1)-NE--1-OR-AUS(Y(1))-LE-TPIV) GU TU 13 н H н H

IF (X(1)+EQ+0+) GO TO 9 IF (ABSIYII)/X(I)).LE.TY) GO TO 13 TY=ABS(Y(1))X(1)) GO TO 12 . . KQ=.TRUE. 2 . GO TO 11 IF (X(I).NE.O..OR.ABS(Y(I)).LE.TY) GO TO 13 TY=ABS(Y(1)) <u>،</u> IR=I CONTINUE KB(JT)=0 TEST "PIVÓT IF (TY.LE.0.) GO TO 14 PIVOT GO TO 43 43 CALL PIV CONTINUE RESET ARTIFICIALS DO 15 1=1.M IF (JH(I).EQ.-1) JH(I)=0 JF (JH(I)=EQ.0) FEAS=FALSE. CONTINUE -VER=.FALSE. PERFORM ONE ITERATION DETERMINE FEASIBILITY *XCK+ (STEP 14 NEG=.FALSF. IF (FEAS) GO TO 18 FEAS=+TRUE+ DO 17 I=1.H IF (X(1).LT.0.0) GO TO 20 IF (JHILI.EQ.OI FEAS=.FALSE. CONTINUE •GET • GET APPLICABLE PRICES ISTEP 21 IF (.NOT.FEAS) GO TO 21 DO 19 1=1.M P(1)=PE(1) IF (X(I)+LT+0+) X(1)=0+ CONTINUE ABSC=.FALSE. GO TO 27 FEAS=+FALSE+ NEG=.TRUE. D0 22 J=1+H P(J)=0. CONTINUE ARSC=.TRUE. DO 26 1+1.M HHtu [IF (X(1).GL.0.0) 60 TO 24 ARSC=+FALSE+ 00 23 J=1+H P(J)=P(J)+E(MM)

1

ŧ,

IF (KQ) GO TO 10

9

10

11

12

13

C

C

C

14

C

15

16

C

C+

17

18

19

20

21

22

23

-5

MH = MM + M

CONTINUE

GO TO 26

C# 1

141

69

70

71

72

74 н

н 73

н 75

H. 76

H 77

H 78

н 79

н 80

н 81

н 82

н 83

н 84

н 85

н Ş6

н 87

H 88

н 89

'n 90

Ή 91

н 92

н 94

H 96

н 97

H 99

93 H

95 н

98 H

H 100

H 101

H 102

H 103

H 104

H 105 H 106

2H 107

H 108

H 103

H 130

н 111

H 115

H 113

H 114 H 115

H 116 H 117

N 110

H 11A H 130

122 H H. 125

N 134

H 125

H 121

IF (JH(I).NE.0) GO TO 26 24 H 126 IF (X(1)+NE+0+) ABSC++FALSE+ H 127 DO 25 J=1+N H 128 P(J)=P(J)-E(MH) H 129 MN=MH+M H 130 CONTINUE 25 H 131 CONTINUE 26 н 132 FIND MINIMUM REDUCED COST C+ *MIN* ISTEP 31 H 133 27 JT=0 H 134 88=0.J H 135 DO 29. J=1+N H 136 IF (KB(J).NE.0) GO TO 29 H 137 DT=0.0 H 136 DO 28 1=1+M н 139 DT=DT+P(1)=A(1,J) H 140 Ð, CONTINUE 28 14.141 IF (FEAS) DT=DT+C(J) H_142 IF (ABSC) DT=-ABS(DT) H 143 IF IDT.GE.OBJ GO TO 29 H 144 BR=DT H 145 JT=J H 146 29 CONTINUE H 147 TEST FOR NO PIVOT COLUMN C H 148 IF (JT.LE.0) GO TO 50 N 149 TEST FOR ITERATION LINIT EXCEEDED ¢ H 150 IF (ITER.GE.NCUT) GO TO 49 H 151 ITER=ITER+1 H 152 C+ • JMX • MULTIPLY INVERSE TIMES AGAINTY ISTEP 41 H 153 DO 31 [=1+M 30 H 154 Y(1)=0+0 N 155 CONTINUE 31 H 156 LL=0 н 157 COST#CEUTE H 158 00.34 I=1.# H 159 • 1 (TL+1)A+TLIA H 100 IF (AIJT.EQ.O.) GO TO 33 COST-COST+AIJT+PE(1) H 161 H 162 DO 32 J=1+H H 163 LL=LL+1 H 164 YLJ)+YLJ)+AIJT+E(LL)/ H 165 CONTINUE H 166 32 GO TO 34 H 167 H 165 33 LLELL+M H 191 CONTINUE 34 H 110 C COMPUTE PIVOT TOLENANCE H 171 YMAX=0.0 H 172 DO 35 1=1+H H 173 YMAX=AMAX1(ABS(Y(1)).YMAX) CONTINUE H 174 35 H 175 TPIV=YMAX+TEXP H 176 HETURN TO INVERSION ROUTINE. IF INVERTING C IF (VER) GO TO B COST TOLLRANCE CONTHOL H 177 n 174 ¢ H 139 RCOST=YNAX/08 M 180 IF (TRIG.AND.BB.GE.-TPIVI GO TO SO H 181 TRIG*.FALSE. H 182 IF (BU.GE.-TPIV) TRIG+.TRUE.

)

f

: .

142

¢

Ľ

ROW! SELECT PIVOT ROW (STEP 5) H 183 AMONG EQS. WITH X+0. FIND MAXIMUN Y AMONG ARTIFICIALS. UR. IF H 184 NONE: H 185 GET MAX POSITIVE YILL AMONG REALS. H 186 1R=0 H 187 AA=0.0 H 188 KO=+FALSE+ H 189 DO 39 1=1+M H 190 IF (X(1)+NE+0+0+0R+Y(1)+LE+TPIV) GO TO 39 H 191 IF (JH(1).EQ.0) GO TO 37 H 192 1F (KQ) GO TO 39 H 193 IF LYLLI+LE+AA) GO TO 39 H 194 GO TO 38 H 195 IF (KQ) 60 TO 36 H 196 KQ#+TRUE+ H 197 AA=Y(I) H 198 IR=I H 199 CONTINUE. H 200 IF (IR-NE-0) GO TO 42 H 201 AA=1.0E+20 FIND MIN. PIVOT ANONG PUSITIVE EQUATIONS H 202 H 203 DO 40 [=1+M H 204 IF (Y(I)+LE+TPIV+OR+X(I)+LE+0+0+UR+Y(I)+AA+LE+X(I)) GU TO 40 H 205 AA=X(1)/Y(1) H 206 IR=I H 207 CONTINUE H 208 IF (.NOT.NEG) GO TO 42 H 209 FIND PIVOT ANONG NEGATIVE EQUATIONS. IN WHICH X/Y IS LESS THAN THE H 410. MININUM X/Y IN THE POSITIVE EQUATIONS. THAT HAS THE LARGEST H 211 ABSF(Y) H 212 AB=-TPIV H 213 00 41 I-1+M H 214 IF (XIA).GE.O..OR.YII).GE.BU.OR.YII) MA.GT.XIII GO TO 41 H 215 88=Y*y*(1) H 216 -18-7 H 217 CONTINUE H 218 TEST FOR NO PIVOT ROW H 219 IF LIR.LE.C. GO TO 48 H 220 PIV PIVOT ON (IR,JT) H 221 ISTEP 61 TA=JH(TR) H 222 IF (IA.GT.O) KB(IA)=0 H 223 NUMPV+NUMPV+1 H 224 JH(1R)=JT H 225 K8(JT)=1R H 226 Y[=+Y(]R) H 227 Y(IR)=-1+0 H.228 H 229 LL=0 TRANSFORM INVERSE H 230 00 46 J=1+H H 231

LL=LL+N GO TO 46 44 XY=E(L)/YI PE(J)=PE(J)+COST=XY E(L)=0 DO 45 1=1+N

L=LL+IR

IF (E(L).NE.U.D) GO TO 44

C+

Ċ

C

c

36

37

38

39

С

40

C

C

С

41

42

C+

43

C

C

H 232

H 233 H 234

H 235 H 236

н 237

H 238 H 239

	•	
	LL=LL+1	
	E(LL)=E(LL)+XY=Y(])	H 240
45	CONTINUE	H 241
46	CONTINUE	H 242
c	TRANSFORM X	H 243
	XY=X(IR)/Y]	H 244
	DO 47 I=1+M	H 245
	XQLD=X(1)	H 246
	X(1)=XOLD+XY+Y(1)	H 247
	IF (.NOT.VER.AND.XII).LT.D.AND.XULU.GE.D.1 XII).	H 240
47	CONTINUE	- H 249
	Y(1R)=+YI	M 250
	x(IR)=-XY ~	H 431
	IF (VER) GO TO 14	- H 252
	TF (NUMPV-LE-M) GO TO 16	H 233
c	TEST FOR INVERSION ON THIS ITERATION	PI 434
	INVC=INVC+1	M 253
	IF (INVC+EQ+NVER) GO TO S	M 253
	GO TO 16	N 257
C+	END OF ALGORITHM. SET EXIT VALUES	M 260
48	IF (-NOT-FEAS-OR-RCOST-LE-1000-) W TU 50	H 240
c	INFINITE SOLUTION	H 261
	¥ =7	M 747
	GO TO 51	/ · H 263
c	PROBLEM IS CYCLING	H 464
49	K=4	N 265
	GO TO 51	H 266
C	FEASIBLE OR INFEASIBLE SOLUTION	H 247
50	κ=Ο	H 268
51	IF (*NOT*FEAS) K=K+1	H 269
	DO 52 J=1+N	H 270
	XX=0.0	H 271
	KBJ=KB(J)	H 272
	[F (KUJ=NE=0) XX#X{KUJ}	H 273
	KB(J)=LL .	H 274
52	CONTINUE	H 275
	rol1)=K	H 276
	KO(2)=ITER	H 277
	K0(3)=INVC	H 278
	KOTA)=NUNVR	- H 279
	KU(5)=NUMPV	085 M
	KO(6)=JT -	H 281
	RETURN	.H 282
	END	H 283-

,

7

÷

CD TOT 0901

ţ

•

.

APPENDIX B

PROGRAM FOR INVESTIGATING MINIMAX OPTIMALITY CONDITIONS

B.1 Introduction

This program is a package of subprograms which investigates the optimality of a design or a proposed solution to an approximation problem in the minimax sense. The program is designed to test a solution for the necessary conditions for a minimax optimum by two different formulations. As indicated in Section 3.4, one uses linear programming, and the other the solution of a set of linear independent equations. A computer program written in Fortran (Version 2.3 and Scope Version 3.4 for the CDC 6400 computer) is listed at the end of the Appendix.

B.2 Program Description

The user may call the package from his main program as follows: CALL MINIMAX (K, KR, NR, YMAX, GRAD, NRMAX, DELTA, EPS, ICRIT, IDATA, IPRINT, MET, NORM, RELTOL, UNIT, K1, K3, MR3, MR1, MR2, X1, X2, X1SUM, X2SUM, R1, R2, RINORM, R2NORM, OPTIM1, OPTIM2, A, B, C, X, PS, JH, XX, YY, PE, E, D, H, Q, IROW, ICOL, LL, MM).

The variables in the argument list of the above subroutine are ordered as input, output and storage variables respectively, and are listed below in that order.

The input variables are k, k_r , n_r , $\hat{y}([\hat{y}_1 \dots \hat{y}_{n_r}]^T)$, $(\nabla \hat{y}^T)^T ([\nabla \hat{y}_1 \dots \nabla \hat{y}_{n_r}]^T)$, followed by

- n maximum possible number of the y.
 - numerical approximation to zero.

δ

ε

a user-specified factor; if $||\mathbf{r}_1||$ or $||\mathbf{r}_2|| < \varepsilon$ and the multiplier vector \mathbf{u}_1 or $\mathbf{u}_2 \ge 0$ the conditions are satisfied for Method 1 or 2; otherwise not.

146

- ICRIT for ICRIT = 1, the user specifies the value of RELTOL and considers $\hat{y}_{\underline{t}}$ for which $(1-\hat{y}_{\underline{t}}/\hat{y}_1) \leq \text{RELTOL}$ for $\underline{t}=2,\ldots,n_r$, to be active while when ICRIT = 2, the user specifies the value of $k_r (\leq n_r)$.
- IDATA logical variable which, if .TRUE., enables the input data to be printed out; otherwise not.
- IPRINT logical variable which if .TRUE., enables all intermediate and final results to be printed out, and no print-outs otherwise.
- MET when MET=1,2, or 3, the package uses Method 1, Method 2 or both the methods, respectively.
- NORM NORM=1 corresponds to the Euclidean vector norm and NORM=2 corresponds to the maximum absolute value of the elements of the vector.

RELTOL tolerance relative to \hat{y}_1 within which some of the $\hat{y}_2, \dots, \hat{y}_n$ lie.

. =1

integer variable specifying the data set refer-UNIT ence number of the output unit.

This is followed by $k_1(=k+1)$, $k_3(=2k+1)$ and $m_{r3}(=2k+1+n_r)$ For the output variables that follow, subscripts 1 and 2 correspond to methods 1 and 2, respectively, as shown below.

> number of \hat{y}_{t} (for t=1,...,n_r) considered when "r1,"r2 optimal conditions are reached.

> > vector of multipliers $[u_{11} \dots u_{1m_{-1}}]^T$, $[u_{21}...u_{2m_2}]^T$

r1,r2

^u1,^u2

||r₁||,||r₂|| norm of vectors r1, r2 logical variables; indicate that the necessary OPTIM1, OPTIM2

conditions for minimax optimum are satisfied if .TRUE., and not satisfied otherwise.

residual vectors $\sum_{k=1}^{m} u_{1k} \sqrt{y_k}$, $\sum_{k=1}^{m} 2k \sqrt{y_k}$

The above output variable list is followed by storage variables, which form the rest of the argument list. The size of the storage arrays and vectors is determined by n_r , k_1 , k_3 and m_{r3} . The values of ε and δ . as specified by the user are crucial for the verification of the optimality conditions, and should be carefully chosen. For further details, see Sections 5.4.4-5.4.6.

.<

B.3 Required Subprograms

The user has to have a subprogram by which the discrete values of the n_r functions \hat{y}_{l} (arranged in descending magnitude) and their derivatives $(\nabla \hat{y}^T)^T$ with respect to the parameters $\phi_1, \phi_2, \dots \phi_k$ are explicitly available. The package uses the following subroutines, the listings of which are available as indicated in the References (see Subroutine ARRAY, Subroutine MINV, Subroutine MFGR, Subroutine SIMPLE, Subroutine SOLVE).

ARRAY converts data arrays from single to double dimension or vice versa while MINV inverts a matrix and calculates its determinant. MFGR determines the rank and linearly independent rows and columns of a given matrix. SIMPLE is a linear-program solving subroutine (listing available in Section A.7) and SOLVE solves a set of linear simultaneous equations.



B.4 Comments

The program was used to test a solution on the problem of lowerorder modelling of a ninth-order nuclear reactor system as treated in Section 3.4.7. Fig. B.1 shows a typical printout of the package for this problem.

This program was run and tested on a CDC 6400 computer. The package requires roughly 40,000 octal units of memory for k=15 and n_r =15. A Fortran listing consisting of 721 cards (including comments) is included in Section B.S.

WUMBER OF HIGHEST HIGHEST HIGHEST HIGHEST HIGHEST HIGHEST MULTIPLIERSSUM OF OF MULTIPLIERSNORM OF FRA MINIAX OFTAUG SATISTED MAL SATISTED MAL OFTAUG SATISTED MAL OFTAUG SATISTED MAL SATISTED MAL OFTAUG SATISTED MAL OF MAL ARE RELTOL SATISTED MAL OFTAUG SATISTED MAL OFTAUG ARE RELTOL SATISTED MAL OFTAUG SATISTED MAL OFTAUG ARE RELTOL SATISTED MAL OFTAUG ARE RELTOL ARE OF SATISTED MAL OFTAUG ARE RELTOL ARE OF ARE NORMAN ARE OF ARE NORMAN ARE NO	NUMBER OF		Σ Ι	E1100 1		~
(w) (x1) (x15Uh) (R1) (R1MORH) (TE/N0) 1 .1000000E+01 .1000000E+01 .18711015E-03 .38711015E-03 NO 2 .99710491E+00 .1000000E+01 .25789922E-09 .25789922E-09 YES 2 .99710491E+00 .1000000E+01 .25789922E-09 .25789922E-09 YES 3 .12885066E-01 .1000000E+01 .25789922E-09 .25789922E-09 YES 3 .12885066E-01 .1000000E+01 .25789922E-09 .25789922E-09 YES 4 .112885066E-01 .1000000E+01 .25789922E-09 .25789922E-09 YES MMRER OF .1288506E-01 .1000000E+01 .25789922E-09 .25789922E-09 YES MAINUM .117PLIEAS MULTPLIEAS MULTPLIEAS MULTPLIEAS MULTPLIEAS MAINUM .123810115E-03 .387110115E-03 .387110115E-03 .00 .1 .1000000E+01 .1000000E+01 .14208087E-03 .387110115E-03 .00 .1 .10000000E+01 .142080	HIGHEST MAXIMA MU CONSIDERED	VECTOR OF JLTIPLIERS	SUM OF MULTIPLIERS	VECTOR OF RESIDUALS	NORM OF RESIDUAL VECTOR	ARE NECESSARY CONDITIO FOR A MINIMAX OPTIMUM SATISFIED FOR A USER- SPECIFIED VALUE OF KR OR RELTOL
1 .1000000E+01 .13711013E-03 .33711013E-03 .3011013E-03 .00 2 .99710491E+00 .1000000E+01 13289922E-09 .25789922E-09 .25789922E-09 YES 2 .99710491E+00 .1000000E+01 25789922E-09 .25789922E-09 YES 1.12395056E-01 .1000000E+01 .25789922E-09 .25789922E-09 YES MMER OF VECTOR SUM VECTOR SUM VECTOR MUTIPLIERS SUM VECTOR SUM VECTOR SUM MUTIPLIERS NULTIPLIERS RESIDUALS RESIDUAL SECTIFIED VALUE OF N MUTIPLIERS NULTIPLIERS NORM NORM NORM NORM MUTIPLIERS NULTIPLIERS RESIDUALS RESIDUAL SECTIFIED VALUE OF N MUTIPLIERS NO (R1) (R2) (R2) NO MUTIPLIERS NO (R2) (R2) (R2) NO 1 .1000000E+01 .14208087E-03 .38711013E-03 NO 2	(196)	(tx)	(XISUN)	(R1)	(RINORM)	(YES/NO)
2 .94710491E+00 .12895086E-01 .1000000E+01 25789922E-09 .25789922E-09 YES *	11	1000000E+01	.10000000E+01	.38711013E-03 14208087E-03	.38711013E-03	9
Method 2 Method 2 Number of Hiddest Maxima Maxima MutripLiens Method 2 MutripLiens Sum of Figure 5 Method 7 Maxima Maxima Maxima Maxima MutripLiens Norm of Figure 5 Method 7 Maxima Maxima Maxima MutripLiens Norm of Figure 5 Norm of Figure 5 Method 7 Maxima Maxima Maxima Maxima Maxima MutripLiens Sum of Figure 5 Norm 7 Are recessary contribute 5 Norm 7 Are recessary contribute 5 Maxima Maxim	2.9	12710491E+00 2895086E-01	.10000000E+01	25789922E-09 .25789922E-09	.25789922E-09	YES
METHOD 2NUMBER OF HIGHESTVECTOR OF MULTIPLIERSSUM OF OF FOR MULTIPLIERSMETHOD 2NUMBER OF HIGHESTOF OF MULTIPLIERSNORM OF FOR MULTIPLIERSMENA OF FOR A MINIMAX OPTIMU OF FOR A MINIMAX OPTIMU OF KI SECIFIED VALUE OF KI SRITISFIED FOR A USER.NALINA MULTIPLIERSOF OF MULTIPLIERSNORM OF FOR A MINIMAX OPTIMU OF SECIFIED VALUE OF KI OR RELTOL OR RELTOLNB(N2)(X2)(X2)UM)(R2)(R2NORM)(YEÉ/NO)1.1000000E+01.1000000E+01.38711013E-03.38711013E-03NO2.98710492E+00.1000000E+01035255563E-09YES2.98710492E+00.1000000E+01035255563E-09YES	,			ру		
NUMBER OF HIGHEST MXINAVECTOR OF OF MXINANORM OF OF MULTIPLIERSSUM OF OF OF MULTIPLIERSVECTOR A B A 			X	ETHOD 2		
(M2) (X2) (X2UM) (R2) (R2NDRM) (YES/NO) 1 .1000000E+01 .1000000E+01 .38711013E-03 .38711013E-03 NO 2 .98710492E+00 .1000000E+01 0. .5525563E-09 YES 2 .98710492E+00 .1000000E+01 0. .5525563E-09 YES	NUMBER OF HIGHEST MAXIMA CONSIDERED	VECTOR OF JLTIPLIERS	SUM OF MULTIPLIERS	VECTOR OF RESIDUALS	NORM OF RESIDUAL VECTOR	ARE NECESSARY CONDITI FOR A MINIMAX OPTIMUM SATISFIED FOR A USER- SPECIFIED VALUE OF KR OR RELTOL
1 .1000000E+01 .1000000E+01 .38711013E-03 .38711013E-03 NO 2 .98710492E+00 .1000000E+01 0. .35255563E-09 YES 12895077E-01 .35255563E-09 .35255563E-09 YES	(HR)	(22)	(X2SUM)	(R2)	(R2NORM)	(YES/NO)
2 .98710492E+00 .1000000E+01 035255563E-09 YES .12895077E-0135255563E-09	Ι.	10000000E+01 *	.10000000E+01	.38711013E-03 14208087E-03	.38711013E-03	Q.
	2	98710492E+00 12895077E-01	.10000000E+01	0. 35255563E-09	.35255563E-09	YES

·.

t

B.5 Fortran Listing for MINIMAX Program

ĩ

.

.-

. ª

۲

.

C	PROGRAM FOR INVESTIGATING MINIMAX OPTIMALITY CONDITIONS	A 1
ŕ		A 2
č	ANTHORS IN MANY LO INC. The State Concerns	A 3
2	AUTIONS STUDER AND I.V. SKINIVASAN DEPARTMENT OF ELECTRI	CAL A 4
Č,	ENGINEERING + MCMASTER UNIVERSITY + MANILTON + ONTARIO + CAN	IADA A 5
(A 6
C	THIS PROGRAM IS A PACKAGE OF SUBPROGRAMS WHICH INVESTIGATES TO	E A T
C	OPTIMALITY OF A DESIGN OR A PROPOSED SOLUTION TO AN APPROXIMAT	ION A B
C	PROBLEM IN THE MINIMAX SENSE	A Y
C		A 10
C	A TYPICAL MAIN PROGRAM FOR MINIMAX SOLUTION CHECK FOLLOWS	11
C	DIMENSION YMAX(15)+GRAD(15+10)+X1(15)+X2(15)+R1(10)+R2(10)	
C	DIMENSION A(21+36)+B(21)+C(36)+X(36)+PS(21)+XX(21)+YY(21)+PE(2	11. 1
Ç.	1 E (21+21)	A 14
C	DIMENSION D(15+10)+H(11+1)+H(11)+IROW(15)+ICUL(10)+L((15)+144	151
¢	LUGICAL IDATA+IPRINT+OPTIMI+OPTIM2	A 16
C	INTEGER UNIT	A 17
C	K=2	
C	NRMAX=15	
C	NR=4	· A 19
C	DELTA=.01	A 20
C	EPS=1+0E-04	A 21
C	ICRIT=2	A 22
C	K K=NR	A 23
c	IDATA=.T.	A 24
C	1PRINT=+T-	A 25
ċ	MET=3	A 26
c	NORMa 1	A 27
ċ	LINET #A	A 28
è	K I MARA	A 29
è.	N 1 - N - 1 Y 3 - 7 af x 1	_ A 30
è	NJ=C+N+1 MD3=7eFx1amDmay	A 31
~	BEADLESS IN FUNKMANTE FOR SAL	A 32
r i	PEACESSIT THRATISTICAS NRS	5 E 🔺
č	KCAU(3)2/ ((GRAU(1)J))J*J*I*K)*I=I*NR)	A 34
C C	[FURMAI()216-8)	A 35
ç	2 FORMAT(2E16.8)	A 36
ç	CALL MINIMAXIK+KR+NR+YMAX+GRAD+NRMAX+DELTA+EPS+ICRIT+IQATA+IPR	INT+ A 37
ć	IMEIONORNORELTOLOUNITOKIOKSOMRJOMRJOMRZOXIOX20XISUM0X2SUM0HIOH2	1KIN A 38
C .	ZNURM+RZNORM+OPTIM1+OPTIM2+A+B+C+X+P5+JH+XX+Y4+PE+E+U+H+Q+1RU++	LCOL A 39
C .	3 · LL · MM1	A 40
Ç	510P	A 41
Ç	END	* A _42
ç		A 43
Ç		A 44
C		A 45
	SUBROUTINE MINIMAX (K.KR. NR. YMAX. GRAD. NRMAX. UELTA. EPS. ICHIT. ID	ATA. A 46
	LIPRINT .MET .NORM.RELTOL.UNIT.K1.K3.MH3.MH1.MH2.K1.K1.K1.K1.K1.K1.K1.K1.K1.K1.K1.K1.K1.	N.K. A 47
	2 . RZ . RINORM . RZNURM . OPTIMI . UPTIMZ . A. B. C. K. PS . JH . XX . YY . Pt . t . U . H. U	INO A 48
	3W+ICOL+LL+NM)	A 44
C .		A 50
C		A 61
C	THE MINIMAX SOLUTION TESTING IS DONE BY TWO RETRODUCTION TO AND M	
C	METI CONSIDERS & LINEAR PROGRAMMING FUNMULATION	A 53
C	HETZ CONSIDERS & FORMA ATION CONSISTING OF A SET OF LINEAD	
c	EQUATIONS	A 24
c		A 33
ċ	· · · ·	A 24
ċ	INPUT-OUTPUT INFORMATION	A 37
ċ	THE USER HAS TO CRETERY VALUES FOR FULL TO DELIVER TO STREAM	
•	THE OUT THE TO PECTT THEORY FOR KINK TOR RELIDER ANNALING	A 37

Ó

151

J,

YMAX+GRAD+DELTA+EPS+ICRIT+IDATA+IPRINT+MET+NORM+UNIT+K1+K3 AND MK3 . 60 THE OUTPUT VARIABLES ARE MR1+R1+RINURN+X1+X1SUH+UPTIM1+MR2+R2+ . 61 R2NORM+X2+X2SUM AND OPTIM2 62 THE VARIABLES A.B.C.X.PS.JH.XX.YY.PE.C.D.H.W.IRGW.ICUL.L.ON ARE A 63 TEMPORARY STORAGE SPACES CURRESPONDING TO K AND NRMAX ITO BE A 64 DIMENSIONED BY THE USER! 65 . 66 . 67 APPENDIX OF WARIABLES 68 A **=NUMBER OF VARIABLE PARAMETERS** 69 NRMAX +MAXIMUM NUMBER OF FUNCTIONS YMAX THAT MAY BE ENCOUNTERED BY A 10 THE USER.FOR THE SAKE OF SAVING MEMORY SPACE INKHAX CAN BE 71 A PUT EQUAL TO NR IF NR IS KNOWN BEFOREHAND ٨ 12 ... NUMBER OF HIGHEST FUNCTIONS YMAXLID YMAXL2D WHICH ARE NR 73 . AVAILABLE FOR CHECKING A SOLUTION FOR THE NECESSARY 74 A CONDITIONS FOR A MINIMAX OPTIMUM. NR SHOULD NEVER BE 75 A GREATER THAN NRMAX A 76 KR -NUMBER OF HIGHEST YMAXII YMAXINR! THAT MAY BE A 77 CONSIDERED ACTIVE BY THE USER FOR CHECKING OPTIMALITY A 78 CONDITIONS. KR IS LESS THAN OR EQUAL TO NR. THE VALUE OF KR A 79 HAS TO BE SUPPLIED BY THE USER IF ICRIT=2 ۸ 80 YHAY =VECTOR OF FUNCTIONS YMAX(1) ++++ YMAX(NR) ARRANGED IN A 81 DECREASING MAGNITUDE. THESE FUNCTIONS ARE TO BE TESTED FOR A 82 THE NECESSARY CONDITIONS FOR A MINIMAX OPTIMUM. YMAXIII IS ۸ н 4 GREATER THAN OR EQUAL TO YMAX(1+1) FOR 1+1+++++R-1 A 84 GRAD **HATRIX OF FIRST DERIVATIVES OF VECTOR YMAX WITH RESPECT TO** 85 . THE & PARAMETERS. THE RUWS OF GRAD CORRESPOND TO THE . 86 GRADIENTS OF YMAXILI, YMAXIZI, YMAXINR! RESPECTIVELY. A 81 GRAD IS OF SIZE INRMAX KI 86 ۸ DELTA .TEST FACTOR FOR ZERU, AFFECTED BY ROUNDOFF NOISE. THE VALUE ۸ 49 OF DELTA DEPENDS UPON THE MAGNITUDE OF ELEMENTS OF GRAD ۸ ¥0 EPS *SCALE FACTOR. WHEN CONSIDERING METHOD 1 IF THE MULTIPLIERS A AΤ A 92 93 A RESIDUAL VECTOR RI IS COMPARED WITH LPS. IF RINORM IS LESS 94 A THAN OR EQUAL TO EPS THE NECESSARY CONDITIONS FOR A HINIMAX A 95 A 96 SITUATION HOLDS FOR METHOD 2 WHEN MR2 HIGHEST FUNCTIONS ARE 41 ٨ CONSIDERED. ٠ QB. ICRIT +THERE ARE TWO CRITERIA AVAILABLE FOR CHECKING THE NECESSARY 99 A CONDITIONS FOR A MINIMAX OPTIMUM. FOR ICHIT=1. THE USER HAS . 100 TO SPECIFY THE VALUE OF RELTUL AND FOR ICHIT=2. THE USER HAS TO SPECIFY THE VALUE OF KR. IF THE USER HAS NO IVEA OF HOW A 101 104 MANY OF THE HIGHEST FUNCTIONS TO CHOUSE OUT OF YMAXLIF..... A 103 THAXINRISHE COULD SPECIFY A VALUE OF KR EQUAL TO NES IFSUN A 104 THE OTHER HAND. THE USER WISHES TO SPECIFY A TOLERANCE BANU A 105 BELOW YMAXILL WITHIN WHICH HE CONSIDERS THE FUNCTIONS TO BE A 106 ACTIVE, HE COULD SPECIFY THE VALUE OF RELTOL A 10/ IDATA +LOGICAL VARIABLE-WHICH IF .TRUE. ENABLES INPUT DATA TO BE A 100 PHINTED OUT ... OTHERWISE NOT-A 107

C

C

C

C

C

C

C

C

C

C

C

C

c

¢

C

C

C

¢

C

¢

C

C

C

C

C

٢

C

c

C

C

C

ς

C

¢

C

¢

C

C

C

C

C

C

C

C

C

C

c

C

C

¢

C

C

C

IPRINT-LOGICAL VARIABLE-WHICH IF TRUES ENABLES ALL INTERMEDIATE AND FINAL RESULTS TO BE PRINTED OUT-AND NO PHINTOUTS OTHERWISES MET SINTEGER VARIABLE WHICH ENABLES THE USER TO CHECK THE NECESSARY CONDITIONS FOR A MINIMAX OPTIMUM BY HETHODS I DR

NECESSARY CONDITIONS FOR A MINIMAX OPTIMUM BY METMOUS 1 UN 2 OR BOTH FOR METHI ON 2 OR 3 -NURM - EVARIABLE WHICH ALLOWS TWO NURMS TO BE AVAILABLE FOR VESTORS

A 110

A 111

A 112

A 113

A 114

A 115

A 110

R1 AND R2. IF NORM=1. THE EUCLIDEAN NORM OF A VECTOR IS A 117 CALCULATED. IF NORM#2. THE VECTOR NURM IS EQUAL TO THE A 118 MAXIMUM ABSOLUTE VALUE OF THE VECTOR ELEMENTS A 119 RELTOL. TOLERANCE RELATIVE TO YMAX(1) WITHIN WHICH YMAX(2)..... A 120 YMAXINR) LIE. THIS FACTOR HAS TO BE SPECIFIED BY THE USER IF A 121 ICRIT-2. WHEN THE USER CONSIDERS THOSE FUNCTIONS FOR WHICH A 122 (1--YMAX(L)/YMAX(1))+L=1+2++++NR+ IS LESS THAN RELTOL+TO A 123 BE ACTIVE FOR OPTIMALITY CUNDITIONS A 124 -INTEGER VARIABLE SPECIFYING THE DATA SET REFERENCE NUMBER UNIT A 125 OF THE OUTPUT UNIT. A 126 A 127 IN THE FULLOWING SECTION SUBSCRIPTS 1 AND 2 DENUTE METHODS 1 AND 2 A 128 RESPECTIVELY. A 129 MR1+MR2 A 130 WHICH SATISFY THE THE NECESSARY CONDITIONS FOR A A 131 MINIMAX OPTIMUM AS VERIFIED BY METHODS 1 AND 2 . A 132 RESPECTIVELY A: 133 EVECTOR OF MULTIPLIERS OF LENGTH HAL AND HAR2 X1+X2 A 134 RESPECTIVELY+WHEN THE NECESSARY CONDITIONS FOR A A 135 2 MINIMAX OPTIMUM ARE SATISFIED AS VERIFIED BY NETHOUS A 136 1 AND 2. AT AN OPTINUM. THE ELEMENTS OF THE A 137 MULTIPLIERS ARE ALL NON-NEGATIVE A 138 X1SUM+X2SUM +SUM OF ELEMENTS OF VECTORS X1 AND X2 RESPECTIVELY. A 139 AT THE OPTIMUM. THE ELEMENTS OF THE VECTORS ARE ALL A 140 NUN-NEGATIVE AND ADD UP TO UNITY A 141 **R**1 -RESIDUAL VECTOR OF LENGTH & GENERATED BY LINEAR A 142 CONBINATION OF THE GRADIENTS OF YMAX(1)+YMAX(2)++++ A 143 YMAX(MR1) BY THE MULTIPLIERS X1(1)+X1(2)++++X1(MK1) A 144 GUT FROM METHOD 1. THUS KI IS A PRODUCT OF THE ROW-A 145 VECTOR X1 POST-MULTIPLIED BY THE MR1 ROWS OF GRAD A 146 R 2 RESIDUAL VECTOR OF LENGTH K GENERATED BY LINEAR A 147 COMBINATION OF THE GRADIENTS OF YMAX(1), YMAX(2)..... A 148 YMAX (MR2) BY THE MULTIPLIERS X2(1/+X2(2/++++X2(MH2) A 149 GOT FROM METHOD 2. THUS R2 IS A PRUDUCT OF THE HOW-A 150 VECTOR X2 PUST-NULTIPLIED BY THE MK2 KOWS OF GRAU A 121 RINORM+RZNORM=NURMS OF VECTORS RI AND RZ A 152 OPTIM1.0PT1M2=LOGICAL VARIABLES.IF .TRUE. INDICATE THAT NECESSARY A 153 CONDITIONS ARE MET FOR A USER-SPECIFIED VALUE OF EPS A 154 AS VERIFIED BY METHODS 1 AND 2 RESPECTIVELY. IF THEY A 155 ARE .FALSE. THE NECESSARY CONDITIONS ARE NOT A 156 SATISFIED A 157 A 154 K 1 =K+1 A 159 K N =24(+1)A 160 MRN =2*K+1+NRMAX A 161 K1+K3+MR3 ARE INTEGERS WHICH ARE NECESSARY FOR EFFICIENT USE OF A 162 COMPUTER CORE MEMORY FOR SOME TEMPORARY STORAGE VECTORS AND ARRAYS A 163 A 164 A 165 DIMENSIONING INFORMATION -----A 166 THE USER HAS TO DIMENSION IN HIS MAIN PROGRAM THE FOLLOWING A 167 ARRAYS AND VECTORS. A 164 YMAX =VECTOR OF DIMENSION NRMAX A 169 GRAD +ARRAY OF DIMENSION INRMAX+K A 170 X1+X2=VECTORS OF LENGTH NRMAX A 171 R1+R2=VECTORS OF LENGTH K A 172 . -ARRAY OF SIZE (K3.MR3) A 173

c

¢

٢

C

C

C

C

C

¢

C

C

¢

C

C

C

C

C

6

¢

C

C

C

C

C

C

Ć

с

C

¢

¢

C

C

C

C

C

C

c

C

C

C

¢

C

c

C

C

C

¢

C

C

¢

C

0

C

٢

¢

C B+PS+JH+XX+YY+PE=VECTORS OF LENGTH K3 A 174 C+X =VECTORS OF DIMENSION MR3 C · Ø A 175 *ARRAY OF SIZE (K3.K3) F C 176 =ARRAY OF DIMENSION (MRMAX+K) =ARRAY OF DIMENSION (NRMAX+K) C D 177 A C Ο. 178 . -SQUARE MATRIX OF SIZE (KISKI) C H. 179 ¢ 0 **=VECTOR OF LENGTH K1** 180 IROW +VECTOR OF LENGTH NRMAX . C 181 C ICOL =VECTOR OF LENGTH K A 182 LL+MM=VECTORS OF LENGTH NRMAX C 183 . C A 184 A 185 TYPE DECLARATION ---C ____ A 186 THE USER HAS TO DECLARE THE TYPE OF SOME OF THE VARIABLES AS A 187 FOLLOWS. A 188 INTEGER UNIT A 189 LOGICAL IDATA+IPRINT+OPTIN1+OPTIM2 A 190 A 191 C A 192 ¢ SUBROUTINE INFORMATION ---A 193 THE USER HAS TO SUPPLY THE FOLLOWING SUBROUTINES WITH THIS PACKAGE A 194 OR ENSURE THAT THESE SUBROUTINES ARE IN THE PERMANENT LIBHARY OF A 195 THE COMPUTER HE IS USING A IGA SUBROUTINE ARRAY -REFERENCE (2) 197 . SUBROUTINE MINY -REFERENCE (3) SUBROUTINE MEGR -REFERENCE (4) SUBROUTINE SIMPLE -REFERENCES (5)+16 A 198 A 199 A 200 A 201 IT IS IMPORTANT TO POINT OUT THAT THE VALUES OF EPS AND WELTA AS A 202 SPECIFIED BY THE USER ARE CRITICAL FOR TESTING A SOLUTION FOR THE -A 203 NECESSARY CONDITIONS FOR A MINIMAX UPTINUM, AND A GREAT UEAL OF A 204 CARE HAS TO BE EXCERCISED WHEN SPECIFYING VALUES FOR THEM. IN ADUITION.IT HAS TO BE PUINTED OUT THAT IN A FORMAT A 205 A ∡06 STATEMENT IS LIKE A HOLLERITH PARAMETER INCLUDING WHATEVER IS A 207 WITHIN THE TWO . SYMBOLS IN THE HOLLERITH FIELD. A 208 LUGICAL IPRINT. IDATA. OPTIMI. OPTIM2. ISP A 209 INTEGER UNIT A 210 DIMENSION YMAX(1), X1(1), X2(1), R1(1), R2(1), GRAD(NRMAX,1) DIMENSION A(K3,1), U(1), C(1), X(1), PS(1), JH(1), XX(1), Y(1), P A 211 A 212 161110 ELK30110 MIK10110 GELTO INOWELTO ICULETTO LLETTO MMETTO UNM A 213 2MAX+11 A 214 IF (NR.LE.NRMAX) GO TU 1 A 215 WRITE LUNIT-201 A 216 RETURN A 217 CONTINUE A 218 ISP=.F. A 219 OPTIMI .F. A 220 OPTIM2++F+ A 221 GO TO 12.61. ICRIT A 222 KR=1 A 223 IF INR.EQ.11 GO TO 4 A 224 DO 3 [=2+NR A 225 IF IEYMAXELI-YMAXEEII.eLE. (RELTOLOYMAXELI) KR=KR+1 A 226 CONTINUE A 227 IF (-NOT-IDATA) GO TO B A 228 WHITE (UNIT-21) K-KN-NRMAK-NR-DELTA-EPS-ICHIT-IDATA-IPHINT-MET-NUK A 244 1M+RELTUL+UNIT+K1+K3+HK3+(I+YMAX(I)+I+1+NK) A 2305

1

C

¢

c

C

C

C

C

C

C

C

¢

¢

¢

C

C

6

C

C

¢

¢

C

¥

7

٦

.

.

.

. .

.

t

DO 5 I=1+NR WRITE (UNIT+22) ((I+J+GRAD(I+J))+J=1+K) CONTINUF GO TO 8 CONTINUE RELTOL=1+-YMAX(KRJ/YMAX(1) IF (+NOT+IDATA) GO TO 8 WRITE (UNIT+23) K+KR+NRMAX+NR+DELTA+LPS+ICKIT+IUATA+IPRINT+M	
WRITE (UNIT+22) ((1+J+GRAD(1+J))+J=1+K) CONTINUF GO TO 8 CONTINUE RELTOL=1+-YMAX(KR1/YMAX(1) IF (+NOT+IDATA) GO TO 8 WRITE (UNIT+23) K+KR+NRMAX+NR+DELTA+LPS+ICHIT+IDATA+IPRINT+M	
CONTINUF GO TO 8 CONTINUE RELTOL=1YMAX(KRI/YMAX(1) IF (.NOT.IDATA) GO TO 8 WRITE (UNIT.23) K.KR.NRMAX.NR.DELTA.LPS.ICHIT.IUATA.IPRINT.M	
GO TO 8 CONTINUE RELTOL=1YMAX(KRI/YMAX(1) IF (.NOT.IDATA) GO TO 8 WRITE (UNIT.23) K.KR.NRMAX.NR.DELTA.LPS.ICHIT.IUATA.IPRINT.M	
CONTINUE RELTOL=1YMAX(KRI/YMAX(1) IF (.NOT.IDATA) GU TO 8 WRITE (UNIT.23) K.KR.NNMAX.NN.DELTA.LPS.ICHIT.IUATA.IPRINT.M	. ^
RELTOL=1YMAX(KRJ/YMAX(1) IF (.NOT.IDATA) GU TO 8 WRITE (UNIT.23) K.KR.NKMAX.NK.DELTA.LPS.ICKIT.IUATA.IPRINT.M	
IF (.NOT.IDATA) GU TO 8 WRITE (UNIT.23) K.KR.NNMAX.NN.DELTA.LPS.ICKIT.IUATA.IPRINT.M	
WRITE (UNIT+23) K+KR+NRMAX+NR+UELTA+LPS+ICHIT+IDATA+1PRINT+M	
• 1	LTONUH
IM+RELTOL+UNIT+K1+K3+NK3+(I+YMAX(I+1/2)+NR)	
DO 7 I=1+NR	
WRITE (UNIT+22) ((I+J+GRAD(I+J+)+J=1+K)	
CONTINUE	
GO TO (9+14+9)+ MET	
CONTINUE	
IF (.NOT. IPRINT) GO TO 10	,
WRITE (UNIT-24)	
WRITE (UNIT,25)	
WRITE (UNIT+26)	
CONTINUE	
MR=1	•
x1(1)=1+	
x1SUM=1.	
DO 11 J=ISK	
R1(J) = GRAD(1+J)	
CONTINUE	
CALL SOLCHK IK .MR .MR I. ICRIT, IPRINT .NURM .UNIT .XI .XI SUM .RI .RI	IÜRM #EF
15.0PT1M11	
1F (OPIIMI) GO TO 13	
F (xR-F0-1) 60 10 13	. . `
CALL METI (K+K3+MR+NR+NK3+YMAX+GRAQ+X1+X15UH+R1+A+B+C+K0+X+F	S.+JH+)
1X-YY-PE-F-NRMAX)	· •
CALL SOLCHK (K+MR+MR)+ICRIT+1PRINT+NORM+UNIT+X1+X15UM+R1+R1+	IORM + EF
15 OPTIM11	
16 (OPTIM1) GO TO 13	•
CONTINUE	
CONTINUE	
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE	
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE ME (-NOT_IPPINT) GO TO 15	,
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 HEITE (INT.27)	,
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WEITE (UNIT.27) WEITE (UNIT.27)	,
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WRITE (UNIT.27) WRITE (UNIT.27) WRITE (UNIT.27)	` •
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WRITE (UNIT.27) WPITE (UNIT.25) WRITE (UNIT.20) CONTINUE	
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WEITE (UNIT.27) WPITE (UNIT.25) WRITE (UNIT.28) CONTINUE MP.1	` •
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WEITE (UNIT.27) WPITE (UNIT.25) WRITE (UNIT.20) CONTINUE MR=1 23(1)=1	,
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WHITE (UNIT.27) WHITE (UNIT.25) WRITE (UNIT.20) CONTINUE MR=1 X2(1)=1. B	` •
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WHITE (UNIT.27) WHITE (UNIT.25) WRITE (UNIT.25) CONTINUE MR=1 X2(1)=1. X25UH=1. P	`
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE TF (.NOT.IPRINT) GO TO 15 WFITE (UNIT.27) WFITE (UNIT.25) WRITE (UNIT.25) WRITE (UNIT.20) CONTINUE MR=1 X2(1)=1. X2SUM=1. DQ 16 J=1.K DQ 16 J=1.K	
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WRITE (UNIT.27) WRITE (UNIT.25) WRITE (UNIT.25) WRITE (UNIT.20) CONTINUE MR=1 X2(1)=1. X2SUM=1. DO 16 J=1.K R2(J)=GRADE1.J) CONTINUE	、
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WHITE (UNIT.27) WHITE (UNIT.27) CONTINUE R2(J)=GRADE1.J) CONTINUE CONTINUE	KORM •E
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WHITE (UNIT.27) WHITE (UNIT.25) WRITE (UNIT.25) WRITE (UNIT.26) CONTINUE MR=1 X2(1)=1. X2(1)=1. R2(J)=GRADE1.J) CONTINUE CALL SOLCHK (K.MR.MR2.ICRIT.IPRINT.NURM.UNIT.X2.X25UM.R2.H2)	WORM • E
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WFITE (UNIT.27) WFITE (UNIT.25) WRITE (UNIT.20) CONTINUE MR=1 X2(1)=1. X2SUM=1. DO 16 J=1.K R7(J)=GRADE1.J) CONTINUE CALL SOLCHK (K.MR.MR2.ICRIT.IPRINT.NURM.UNIT.X2.X2SUM.R2.R2) IS OPTIM2)	WORM •E
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPRINT) GO TO 15 WEITE (UNIT.27) WPITE (UNIT.25) WRITE (UNIT.20) CONTINUE MR=1 X2(1)=1. X2SUM=1. DO 16 J=1.K R2(J)=GRAD[1.J] CONTINUE CALL SOLCHK (K.MR.MR2.ICRIT.IPRINT.NURM.UNIT.X2.X2SUM.R2.R2) IS.OPTIM2) IF (OPTIM2) GO TU 19 IF (OPTIM2) IS OTU 19	KORM • EI
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE TE (INOT.IPRINT) GO TO 15 WHITE (UNIT.27) WHITE (UNIT.25) WHITE (UNIT.20) CONTINUE MR=1 X2(1)=1. X2SUM=1. DO 16 J=1.K R7(J)=GRADT1.J) CONTINUE CALL SOLCHK (K.MR.MR2.ICRIT.IPRINT.NURM.UNIT.X2.X2SUM.R2.H2) IS +OPTIM2) IF (OPTIM2/ GO TU 19 IF (KR.E0.1/ GO TU 19) IF (KR.E0.1/ GO TU 19)	VORM • E
CONTINUE CONTINUE IF (HET.NE.3) RETURN CONTINUE IF (.NOT.IPR(NT) GO TO 15 WITE (UNIT.27) WHITE (UNIT.25) WRITE (UNIT.25) CONTINUE MR=1 X2(1)=1. X2SUM=1. PQ 16 J=1.K R2(J)=GRADE1.J) CONTINUE CALL SOLCHK (K.MR.MR2.ICRIT.IPRINT.NUKH.UNIT.X2.X2SUM.R2.R2) IS.OPTIM2) IF (OPTIM2/ GO TU 19 IF (KR.EQ.1/ GO TO 19 UO 18 II=2.KR	WORM . E

t,

o

--:4

÷

L

-	CALL METZ (K+K1+MR+NK+YMAX+GKAD+DELTA+IPRINT+ISP+UNIT+X2+X2SUM+H2+ 1D+H+G+IROW+ICOL+LL+MM+NRMAX) IF (+NOT+ISP% GO TO 17	Á	288 289 290	
ć		A	291	
č	WHEN 13- 13 - INC. SETTHER THE NUMBER OF UNKNOWN NULTIPLIERS IS		292	
č	DELTA IS TOO SMALL SO THAT HE DETTEN FOR THE VALUE OF		293	
è	THE CURRENT VALUE OF NO		294	
č		•	295	
-	CALL METT (Kakaama And and a guara guara a sa	A	296	
•	1X YY PE +E +NRMAX)		297	
	15P*•F•		298	
17	CONTINUE	<u>.</u>	299	
	CALL SULCHK (K.MR.MR.2.)CKIT.IPRINT.NURN.LUNIT.X2.X251M.H2.W/MOVM.LD		300	
	15.0PTIM2)	2	302	
	IF (OPTIM2) GO TO 19		302	
18	CONTINUE	7	304	
19]	CONTINUE		305	
_ /	RETURN	Â	306	
≤ 1		A	307	
\leq			308	
<u>c</u>		A	309	
51	-	A	310	
20	FURMAT (141) A MO IS SUCATED THAN ADDARD STORE AND A SUC	A	311	
20	I INCREASE IN NOMAY TO A VALUE A PRAVAX HERE AND THIS CALLS FOR AN	•	312	
21	FORMAT EINING A VALUE GREATER THAN OR EQUAL TO NROT		313	
••	1 IS/65X *KR=* IS-8 IS (WHE COMMING TO SET TO BE ADD AND AND A TO SET		314	
	2NE##15/62X #0H1 TAM#+1AAMAAMAAMAAMAAMAAMAAMAAMAAMAAMAAMAAMAA	•	312	
	3ATA=*165/61X=#IPRINT=*15/64X+#FT=515/637_64(**1CKII=*167624*ID	.	310	
	4***E10*8/63X**UNIT#**I5/65X***E1**15/65X****C10L	<u></u>	211	
	5x+#YMAX(#+12+#1##+F16+#%9x+#YMAX(#+12+0)+#+14+#04+#WMAY##+13/(#	•	370	
	6++E16+8+9X++YMAX(++12++)+++16+8)		330	
22	FORMAT (* GRAD(*,12,*,*,12,*)**,E16.8.6X.*GRAD(*,12,*,*,12,*)**.()	2	320	
	16-8+6X+*GRAD(*+12+*+*+12+*+*+E16-8+6X+*GRAD(*+12+*+*+12+*+*+16+	Ā	322	1
	28)		323	
23	FURMAT (1H1/60X+*INPUT DATA LIST#/61X+*	A	324	٠
	1+15/65X+#KR=#+15/62X+#NRMAX=#+15/65X+#NR=#+15/62X+#ULTA=++Ê16+8/6		325	
	24x+*EPS=*El6+8/62x+*ICKIT=*+15/62x+*IUATA=*+L5/61x+*IPKINT=*+L5/64	A	326	
	3X+*MET=*+15/63X+*NORM=*+15/61X+*RELTOL=*+E16+#+#(CORKESPONDING TO	•	327	
	4KR1+/63X++UNIT=++15/65X++K1=++15/65X++K3=++15/64X++HR3=++15/(4X++Y		328	
	5MAX[*+I2+*]=**E16+8+9X+*YMAX[*+I2+*]=*+E16+8+9X+*YMAX(*+I2+*)=++E1		329	
	60-0-74- THAX (+-12+010-616-611	A	330	
24	FORMAT LITTIGE AS THE THOU IN GAR	A	331	
23	TORMAL TOAT HOMDER OF STAAT TECTOR TAAT TEAT TO THE TAAT TAAT TAAT TAAT TAAT TAAT TAAT	•	336	
	1211 ADE 204 ADE 104 AFAI A HINTAR ANTIGEST 11/A. THE START		333	
	ALTIN IF IF A SALE AND A THE AND A ALTINGAL OF INCATANA ADDALANA A AND A	- Ĉ	334	
	AIISFIED FOR A USED AVANTAL CLEASE MELTOR AND	"	222	
	SALUE OF KR#//110X.+OR BEITOLE/	2	347	
26	FURNAT (9X++(MR)++17X++(X1)++17X++(X1SHM)++16X++(+1)++18X++(+1)+(+++)	2	مدد	
	11++12X++(YES/NO1+//)	À	334	
27	FURMAT (1H1/64%+METHOD 2#/64%+##///)	Å	340	
28	FURMAT (9X++ (MR)++17X++(X2)++17X++(X2SUN)++16X++(H2)++18X++(H2NUKM		341	
	1)++12X++(YES/NO)+//}	~	342	
	END		343	
C		A	344	

č			345
č			346
C		B	1
	SUBROUTINE SOLCHK (K+MR+MMR+ICRIT+IPRINT+NORM+UNIT+X+XSUM+K+RNORM+	́В	4
	1EPS+OPTIM)	B	٦
c	,	· 8	
C	THIS SUBROUTINE CHECKS THE SOLUTION FOR NECESSARY CONDITIONS FOR		ž
C	A MINIMAX OPTIMUM BY FIRST TESTING WHETHER THE MULTIPLIERS VIII-	6	1
C	X(2)++++X(MR) ARE NON-NEGATIVE AND THEN FINDING OUT IF THE MUDE		, ,
C	RNURM OF THE RESIDUAL VECTOR & IS LESS THAN UP EQUAL TO FUE		<u></u>
C		0	• •
	DIMENSION X(1), R(1)	0	, y
	INTEGR UNIT	8	10
	LOGICAL IPRINT.OPTIM	8	11
		В	12
		8	13
	60 TO (1-2), NORM	в	14
		ь	15
•		ы	16
		8	17
2	HNORM=ANORM2[K + R]	B	18
,	CONTINUE	8	19
	IF INNORM.GT.EPS) GO TO 4	ы	20
	OPTIM=.T.	в	21
	MMR = MR	8	22
4	CONTINUE	- B	23
	DO 5 J=1, MR	Ē	24
	1F (X(J)+GE+0+) GO TO 5	н	25
	OPTIM=_F.	R.	26
	GO TO 6	н	27
5	CONTINUE	н	<u>с</u> ,
6	CONTINUE	0 0	20
	IF (+NOT+IPRINT) RETURN		4.7
	IF (OPTIN) GO TO 7	5	10
	WRITE LUNIT+15/ MR-X(1)+XSUM-R(1)+HNDHA	0	21
	GO TO B	0	22
7	WRITE (UNIT-16) MR. XEIL.XSIN.DELL.DACOM	D	
A	IF (MR.EQ.1) GO TO 13	P	34
•		B	35
a		в	36
*		8	37
10		8	38
10		8	39
	IT IN EWORK' REIURN	В	40
	HRITE CONTINES (X(I))IR(I)MR)	Ð	-4 J
		B	42
זי	DO 12 L=2.4MR	B	43
	WRITE (UNIT-17) X(I)-R(I)	B	44
12	CONTINUE	в	45
13	IF (K.EQ.MR) RETURN	В	46
	WRITE (UNIT+19) (R(I)+I=MR1+K)	В	47
14	RETURN	6 0	40
C		В	49
15	FORMAT (//16x+12+11x+E16+8+6x+E16+8+6x+E16+8+6x+E16+8+11x++N0+1	в	50
16	FURMAT (//10X+12+11X+E16+8+6X+E16+8+6X+E16+8+6X+E16+8+11X++YES+	Þ	51
17	FORMAT (/23X+F)6-8+28X+E16+8)	8	52
18	FORMAT (/23X,E16.8)	8	53
19	FORMAT (/67X+E16+B)	B	54
	FND	8	55

-

SUBROUTINE METI (K.K.3.HWI.KR3.YHAX.GUAD.KI.XI.SUMHHIAABCKAUAS. SUBROUTINE METI (K.K.3.HWI.KR3.YHAX.GUAD.KI.XI.SUMHHIAABCKAUAS. DIMENSION AKS.YN.YHEE KINHAX. DIMENSION AKS.II. GAUKARAA.JI. XI(I). RI(I) DIMENSION AKS.II. GUI.C.CII. XI(I). FII). ATTII. YYII. HRI-HR-I HRI-HR-I HRI-HR-I HRI-KR-I C. 10 DO 2 1-11HR DO 2 1-11HR DO 2 1-11HR C. 11 DO 2 1-11HR DO 3 J-1.KZ C. 11 CONTINUE C. 12 CONTINUE C. 14 CONTINUE C. 14 CONTINUE C. 14 C. 14 C. 14 C. 15 C. 16 C. 16 C. 16 C. 16 C. 17 A.J.YHRI-1. C. 16 C. 17 DO 5 J-1.KZ DO 5 J-1.KZ C. 18 C. 19 DO 5 J-1.KZ C. 19 C. 19 DO 5 J-1.KZ C. 10 DO 5 J-1.KZ C. 10 DO 5 J-1.KZ C. 10 C. 10 C					
C B 57 SUBROUTINE METI ICKRSIMUM ANKARSIYMAX.GRAD.X1XXISUM.MIAABECKGYXP SUBROUTINE METI ICKRSIMUM ANAXSIYMAX.GRAD.X1XXISUM.MIAABECKGYXP DIMENSION ACKSIPH BILL GRAD.GRAD.X1XXISUM.MIAABECKGYXP DIMENSION ACKSIPH BILL GRAD.GRAD.X1XXIJ, X1(1), R1(1) DIMENSION ACKSIPH BILL GRAD.GRAD.X1X, X1(1), R1(1) DIMENSION ACKSIPH BILL GRAD.GRAD.X1, X1(1), R1(1) DIMENSION ACKSIPH BILL GRAD.GRAD.X1, X1(1), R1(1) MRI-MR+1 MRI-MR+1 MRI-MR+1 MRI-MR+1 C 1 DIMENSION ACKSIPH BILL C 1 DIMENSION ACKSIPH BILL C 1 DIMENSION ACKSIPH BILL C 1 MRI-MR+2 MRIMENTAL C 1 DIMENSION ACKSIPH C 2 DIMENSION ACKSIPH C 2 DIMENSIPH C 2 DIMENSION ACKSIPH C 2 DIMENSIPH C	c c		8	56	
C SUBROUTINE METI IS (K3)-MULANK K33, YMAX-UNAD-X1AXISUM-NILAAD-CAROUX (K)	C		8	57	
SUBBOUTINE METI (K.W3.MH.HAR, AK3.YWAX.GNAD.X1.X1SUM.W11A.HBF.K.V.X.C IS.JM.XX.YY.PE.C.MNHAX: DIMENSION A(K.S1): B(1/, C(1): K(1): PS(1): JH(1): XX(1): YY(1): P IE(1): F(KS1): HR]=HR+1 HR]=HR+1 HR]=HR+1 HR]=HR+2 K3KC2-1 DU 2 I=1.4R DU 2 I=1.4R C 1 DU 2 I=1.4R C 1 C 1 DU 2 I=1.4R C 1 DU 2 I=1.4R C 1 C 1 C 1 C 1 C 1 C 1 C 1 C 1	C		8	58-	
IS-JH-XX.YY-PC.E-NEMAX:) DIMENSION YMAXIL: GALORGHAX:], X1[], R1[], DIMENSION AKAS:], B[], C[], X[], PS[], JH[], XX[], YY[], P C MR]-MR-1 MR]-MR-1 MR]-MR-2 MR]-MR-2 MR]-MR-2 C MR]-MR]-C C C C C C C C C C C C C C		SUBROUTINE METL (K+K3+MR+KR3+YMAX+GRAD+X1+X1-X1+H+B1+++	C	1	
DIMENSION YMAX(1): GRADUNRHAR,10, X1(1), R1(1) DIMENSION A(XS): B(1), G(1): X(1): PS(1): JH(1): XX(1): YY(1): P (C) HR1-HR+1 HR2-HR+2 HR2-HR+2 K3-K2-1 DU 2 1-1;HR DU 1 J-1;K A(J): H(J) DU 1 J-1;K A(J): H(J): A(J): A(J		IS+JH+XX+YY+PE+E+NRMAX)	C	- 2	
DIMENSION AK(3):1) B(1)* C(1)* X(1)* PS(1)* JH(1)* XX(1)* YY(1)* P E(1)* F(X3):1) HR1=HR+1 HR2=HR+2 HR3=HR122*C C 2 K2*K2*1 C 3 C 4 K2*K2*1 C 10 C 11 D 2 1=1.MR C 12 C 10 C 11 D 1 J=1:K C 10 C 11 D 1 J=1:K C 10 C 11 C 1 J=0:A0(1).J) C 11 C 11 C 1 J=0:A0(1).J) C 11 C 11 C 1 J=0:A(1).J) C 10 C 1 J=0:K C 1 J C 1 J=0:K C 1 J C 1 J C 1 J=0:K C 1 J C 2 J		DIMENSION YMAX(1), GRAU(NRMAX,1), X1(1), R1(1)	C	3	
Itelli, Fiks, I) C MR1=MR+1 C MR2=MR+2 C MR3=MR1+2*K C K3=K2+1 C DU 2 1-1, MR C DU 1 -1, K C A(J+K)1=-KAD(1,J) C A(J+K)1=-KAD(1,J) C A(J+K)1=-KAD(1,J) C CONTINUE C CONTINUE C DO 3 J=1,K2 C A(J+K)1=-KAD(1,J) C CONTINUE C DO 5 J=1,K2 C A(J+K)1+=0. C CONTINUE C CONTINUE C CONTINUE C DO 6 1=+MR2+MR3 C A(X-S,1)=0. C CONTINUE C DO 7 1=+MR2+MR3 C A(X-S,1)=1. C CONTINUE C DO 7 1=+MR2+MR3 C A(X-S,1)=0. C CONTINUE C DO 7 1=+MR2+MR3 C A(X-S,1)=0. C CONTINUE C		DIMENSION A(K3+1), B(1), C(1), X(1), PS(1), JH(1), X(1), MALL	C	4	
MHI-MH-1 MR2-MR-2 MR3-MR1-2*C K2-2*C K3*K2-1 DU 2 1-1.MR DU 1 J-1:K A(J-K)1J-CRADU(JJ) A(J-K)1J-CRADU(JJ) A(J-K)1J-CRADU(JJ) A(J-K)1J-CRADU(JJ) CONTINUE C		1E(1), F(K3,1) C	ç	5	
MH2-MHR1+22K C 7 MH2-MHR1+22K C 6 K2-2-*K C 9 K2-2-*K C 10 D0 1 J=1:K C 11 D0 1 J=1:K C 12 A(J+K)1-ALD(1+J) C 13 A(J+K)1-ALD(1+J) C 14 CONTINUE C 15 CONTINUE C 16 D0 3 J=1:K2 C 16 CONTINUE C 18 D0 5 J=1:K2 C 18 CONTINUE C 22 D0 5 J=1:K2 C 17 CONTINUE C 22 D0 5 J=1:K2 C 22 IF (J=FG.(1-HR1)1 A(J+1)=1. C 22 CONTINUE C 22 D0 6 1=1:HR A(J+1)=0. C CONTINUE C 23 D0 7 1=KR1; MR3 C 24 A(K3:1)=1. C 23 CONTINUE C 23 D0 7 1=KR1; MR3 C 23			C	6	
MR3=MR1/242K C 8 K2=2+1 C 10 D0 1 J=1+K C 11 D0 1 J=1+K C 12 A(J=K+1)1=-A(J+1) C 13 CONTINUE C 15 CONTINUE C 16 CONTINUE C 16 CONTINUE C 16 CONTINUE C 17 A(J=K+1)=-A(J=1) C 18 CONTINUE C 19 D0 5 J=1+K2 C 11 A(J=HR1)=-1. C 18 CONTINUE C 19 D0 5 J=1+K2 C 20 A(J=1)=0. C 22 CONTINUE C 23 D0 6 J=1+MR C 24 A(Sa:1)=0. C 24 CONTINUE C 23 D0 7 I=MR1,MR3 C 23 B(I)=0. C 23 CONTINUE C 23 D0 7 I=MR1,MR3 C 23 B(K3)=1.		MR2 #MR+2	C	7	
K2N+K C 10 D0 2 1-11-K C 11 D0 1 J-12K C 11 A(J+L)-URAD(1,J) C 12 A(J+K)II-ALJ,I) C 13 CONTINUE C 15 CONTINUE C 16 D0 3 J=1+K2 C 17 A(J,MR1)1. C 18 CONTINUE C 19 D0 5 J-1+K2 C 19 D0 5 J-1+K2 C 10 CONTINUE C 22 D0 5 J-1+K2 C 21 CONTINUE C 22 D0 5 J-1+K2 C 21 CONTINUE C 22 D0 6 1+1+KR C 23 CONTINUE C 24 D0 7 1+KR1+KR3 C 25 D0 6 1+1+KR C 23 D0 7 1+KR1+KR3 C 33 GUTINUE C 33 D0 7 1+KR1+KR3 C 33 GUTINUE C 33 D0 7 1+KR1+KR3 C 33 GUTINUE C 33 D0 9 1+1,KR3 C 33 GUTINUE C 33 GUTINUE C 33 D0 9 1+1,KR3 C 34		MK3=MK1+2+K	ć	8	
CAREVAL C 10 DU 2 1-1+MR C 11 DU 1 J-1+K C 12 ALJSK-11ALJ,1) C 13 ALJSK-11ALJ,1) C 14 CONTINUE C 15 CONTINUE C 16 DO 3 J-1+K2 C 17 ALJSK-11ALJ,1) C 18 CONTINUE C 18 DO 5 J-1+K2 C 10 DO 6 J-1+K2 C 20 DO 7 J-1+K2 C 21 ALJSK-11+0, C 22 IF (J-FO-(1-HR1)) A(J+1)=1, C 22 CONTINUE C 22 DO 6 J-1+K2 C 22 CONTINUE C 22 CONTINUE C 22 DO 7 I-MR1, HR3 C 22 ALS3-11+0, C 22 CONTINUE C 22 DO 7 I-MR1, HR3 C 33 GONTINUE C 32 DO 7 I-MR1, HR3 C 33 GONTINUE C 34 DO 7 I-MR1, HR3 C 35 GONTINUE C 32 DO 7 I-MR1, HR3 C 33 GONTINUE C 34 DO 9 I-I-HR3, HR3		K 7= 2 * K	с -	9	
DO 2 1914K DO 1 JeleK ALJALIPORAD(1,J) ALJAKA ALJAKAL C 13 ALJAKAL C 13 CONTINUE C ONTINUE C ONTINUE		[K 4#K 2+1] D(() 1 − 1 → MD	ç	10	
D0 1 Jilkk C 12 A(JSIIAKA)(1,J) C 13 A(JSK11=-A(J)) C 14 CONTINUE C 16 D0 3 JSIK C 16 PCONTINUE C 16 CONTINUE C 16 D0 5 JSIK C 16 PCONTINUE C 16 D0 5 JSIK C 20 D0 5 JSIK C 21 D0 5 JSIK C 22 IF (JSCO.(I-HR1)) A(JSI) C 23 D0 6 ISIS C 24 CONTINUE C 26 D0 7 ISING C 27 CONTINUE C 26 D0 7 ISING C 27 CONTINUE C 28 D0 7 ISING C 28 D0 8 ISIS C 27 CONTINUE C 28 D0 9 ISING C 28 CONTINUE C 30 D0 9 ISING C 38 CONTINUE C 39 CONTINUE C 39 CONTINUE			ç	11	
ALJSK, IJ=ALJSID ALJSK, IJ=ALJSID CONTINUE			2	42	
A JOKE JIPA (JST) CONTINUE CONTINU			ç	13	
CONTINUE C 13 CONTINUE C 16 DO 3 J=1.K2 CONTINUE C 16 CONTINUE C 19 DO 5 J=1.K2 CONTINUE C 10 A(J,1)=0. IF (J,FO.(1-MR1)) A(J,1)=1. C 22 CONTINUE C 25 C 24 C 25 C 25 C 26 C 27 C 27	,		$\sum_{i=1}^{n}$	1.	
CONTINUEC 1600 3 J=1.K2C 17A(J+MR)I=-1.C 1910 0 5 J=1.K2C 2010 5 J=1.K2C 2011 0 0 5 J=1.K2C 2111 0 0 5 J=1.K2C 2111 0 0 5 J=1.K2C 2111 0 0 5 J=1.K2C 2211 1 0 0 5 J=1.K2C 2211 1 0 0 5 J=1.K2C 2311 1 0 0 5 J=1.K2C 2511 1 0 0 5 J=1.K2C 2511 1 0 0 5 J=1.K2C 2611 1 0 0 5 J=1.K2C 2811 1 0 0 5 J=1.K2C 2811 1 0 0 5 J=1.K2C 3011 1 0 0 5 J=1.K2C 3311 1 0 0 5 J=1.K2C 3311 1 0 0 5 J=1.K2C 3311 1 0 0 7 1 1 NUEC 3311 0 0 6 1 = 1.K2C 3311 0 0 7 1 1 NUEC 3311 0 0 7 1 NUEC 3311 0 0 7 1 1 NUEC 3311 0 0 7 1 1 NUEC 3311 0 0 7 1 1 NUEC 3312 0 9 1 1 - 0.C 3313 1 - 0.C 3414 1 - 0.C 3515 0 9 1 1 - 0.C 3616 1 - 0.C 3717 1 1 - 0.C 3618 1 1 - 0.C 3619 1 - 0.C 3710 9 1 1 - 0.C 3717 1 - 0.C 3618 1 - 0.C 3719 1 - 0.C 3710 2 1 - 0.C 3710 2 1 - 0.C 3610 3 1 - 0.C 3710 3 1 - 0.C 3710 4 2 1 - 0.C 3711 4 2 1 - 0.C 3711 5 2 1 1 - 0. <t< td=""><td>/ ></td><td></td><td>~</td><td>70</td><td></td></t<>	/ >		~	70	
00 + J J=1KZ C 11 A LJMRT1=-1. C 18 CONTINUE C 20 00 4 1=MR2+MR3 C 21 ALJ,11=0. C 22 1F LJ+FG.(1=MR1)1 A(J+1)=1. C 23 CONTINUE C 25 00 6 1=1+MR C 26 00 7 1=MR1+MR3 C 27 CONTINUE C 28 00 7 1=MR1+MR3 C 30 CONTINUE C 30 00 7 1=MR1+MR3 C 30 CONTINUE C 31 00 7 1=MR1+MR3 C 32 SUBROUTINE C 31 00 8 1=1+K2 C 33 B11=0. C 33 CONTINUE C 36 D0 9 1=1+MR3 C 33 CONTINUE C 36 GONTINUE C 37 D0 9 1=1+MR3 C 32 CONTINUE C 36 GUNROUTINE SIMPLE IS NOW GUING TO BE CALLED-ANY ALTERNATIVE CHUICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USEN AS LONG AS 11 CHUICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USEN AS LONG AS 11 CHUICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USEN AS LONG AS 11 CHUICE TO THIS SUBROUTINE IS AND COLO	5		~	10	
A USAWATIA-1. CONTINUE C			· •	10	
CONTINUE C 40 D0 5 J=1xZ2 C 40 D0 4 I=MRZ;MR3 C 21 ALJ;I]=0. C 43 CONTINUE C 44 CONTINUE C 25 D0 6 I=I.MR C 26 AK3,I]=1. C 28 CONTINUE C 28 D0 7 I=MR1,MR3 C 30 CONTINUE C 31 D0 7 I=MR1,MR3 C 30 CONTINUE C 33 CONTINUE C 33 D0 7 I=I.MR3 C 33 CONTINUE C 34 D0 9 I=I.MR3 C 36 C11=0. C 38 GONTINUE C 34 SUBROUTINE SIMPLE IS NOW GUING TO BE CALLED.ANY ALTERNATIVE C 42 CMUITE C TO THIS SUBRUTINE IS ALLOWABLE FUNCTINE SIMPLE C 35 C 36 SUBROUTINE SIMPLE SOLVES A LINEAR PROUKAMAINE PRUBLA OLINEARS IT C 44 C 42 MITH SIMPLE SOLVES A LINEAR PROUKAMAINE PRUBLA OLINEARS			č	10	
DU 9 JULIAZ DU 9 JULIAZ ALJ,1]=U. IF (JJ-FO.(I-MR1)) A(J,1)=1. C 22 CONTINUE C CONTINUE C C C CONTINUE C C C CONTINUE C C C CONTINUE C C C C CONTINUE C C C C CONTINUE C C C C C C C C C C C C C C C C C C C	,		č	19	
OU + INRESTRY C 21 ALJJIEU. C 22 IF (JFG.(I-MR1)) A(Jeliel. C 23 CONTINUE C 25 DO 6 I=LMR C 26 CONTINUE C 27 CONTINUE C 28 DO 7 I=MRI,MR3 C 27 CONTINUE C 28 DO 7 I=MRI,MR3 C 30 CONTINUE C 32 DO 8 I=1-K2 C 33 BIJ=0. C 34 CONTINUE C 35 CONTINUE C 35 CONTINUE C 35 CONTINUE C 36 CONTINUE C 37 C 37 C 37			~	20	
ALSTING. IF (J.FO.(1-MR1)) A(J+1)=1. CONTINUE CONTINU			è	22	
17 10-00-01 (1-RE1) 7 A(J+1)-1. C 23 CONTINUE C 23 D0 6 1=1+MR C 24 AIX3+13-1. C 27 CONTINUE C 28 D0 7 1=+RE1+MR3 C 30 AIX3+13-0. C 31 CONTINUE C 32 D0 7 1=+RE1+MR3 C 31 CONTINUE C 32 D0 6 1=1+K2 C 33 B(1)=0. C 33 CONTINUE C 35 D0 9 1=1+K2 C 35 B(1)=0. C 35 CONTINUE C 35 D0 9 1=1+MR3 C 35 CONTINUE C 35 D0 9 1=1+MR3 C 35 C11=0. C 35 D0 9 1=1+MR3 C 35 C11=0. C 35 D0 9 1=1+MR3 C 35 C11=0. C 35 CONTINUE C 36 SUBROUTINE SIMPLE IS NOW GUING TO BE CALLED-ANY ALTERNATIVE C 42 CMOICE TO THIS SUBROUTINE IS ALLOWABLE FOUR THE USEN AS LUNU AS 11 C 42 CMUICE TO THIS SUBROUTINE SUBROUTINE IS ALLOWABLE FOUR THE USEN AS LUNU AS 11 C 42			è	24	
CONTINUE CONTIN		$[\mathbf{A}_{\mathbf{J}}] = [\mathbf{A}_{\mathbf{J}}] + [\mathbf{A}_{\mathbf{J}}] = [\mathbf{A}_{\mathbf{J}}]$	č	14	
CONTINUE DO 6 I=I=HR A(K3,I)=I. CONTINUE C			è	25	
D0 6 1 = 1 + MR C 27 A(K3:1)=1. C 28 CONTINUE C 28 D0 7 1 = MR1, MR3 C 30 A(K3:1)=0. C 31 CONTINUE C 32 B(I)=0. C 33 CONTINUE C 33 B(K3)=1. C 33 CONTINUE C 34 D0 9 1=1.4R3 C 35 CONTINUE C 36 CONTINUE C 37 CONTINUE C 36 CONTINUE C 37 CONTINUE C 40 CONTINUE C 40 CONTONINE SUBROUTINE IS			č	22	
Altistip: Continue C 28 DO 7 I=MR1,MR3 C 29 Atts:::::::::::::::::::::::::::::::::::			č	20	
CONTINUE C 20 DO 7 I=MR1+MR3 C 30 A1K3+11=0. C 30 CONTINUE C 31 DO 8 1=1+K2 C 32 B(I)=0. C 33 CONTINUE C 35 B(K3)=1. C 36 DO 9 1=1+MR3 C 36 C(1)=0. C 37 C(1)=0. C 36 CONTINUE C 36 DO 9 1=1-MR3 C 36 CONTINUE C 37 C(1)=0. C 36 CONTINUE C 37 CONTINUE C 40 CONTINUE C 40 CONTINUE C 40			è	28	
DO J. HARLIARS C 30 AIK3.11=0. C 30 CONTINUE C 31 DO B 1=1.K2 C 33 B(I)=0. C 33 CONTINUE C 33 DO 9 1=1.MR3 C 35 DO 9 1=1.MR3 C 35 CONTINUE C 35 B(K3)=1. C 36 DO 9 1=1.MR3 C 35 CONTINUE C 35 SUBROUTINE SIMPLE IS NOW GOING TO BE CALLED.ANY ALTERNATIVE C 36 CONTINUE C 36 SUBROUTINE SUBROUTINE IS ALLOWADE FUR THE USEN AS LUNG AS IT C 42 CHUICE TO THIS SUBROUTINE IS ALLOWADE FUR THE USEN AS LUNG AS IT C 42 CHUICE TO THIS SUBROUTINE SUBPLEXATION. C 43 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 45 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 45 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 45 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 45 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 45 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 45 SUBROUTINE SIMPLE IS A NOUTFLED VENSION OF A PROURAM AVAILABLE <td></td> <td></td> <td>č</td> <td>24</td> <td></td>			č	24	
CONTINUE CONTINUE DO 8 I=1+K2 B(I)=0. CONTINUE B(K3)=1. CONTINUE B(K3)=1. CONTINUE CONT		DU T 1=mk(19MK) A(K),T)	č	30	
CONTINUE C 32 DO B I=1+K2 C 33 B(I)=0. C 34 CONTINUE C 35 B(K3)=1. C 36 DO 9 I=1+MR3 C 37 C(I)=0. C 37 IF (I.E0-MR1) C(I)=1. C 38 CONTINUE C 37 SUBROUTINE SIMPLE IS NOW GUING TO BE CALLED-ANY ALTERNATIVE C 42 CONTINUE C 42 SUBROUTINE SIMPLE IS NOW GUING TO BE CALLED-ANY ALTERNATIVE C 42 CHUICE TO THIS SUBROUTINE IS ALLOWABLE FUR THE USER AS LONG AS IT C 43 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 43 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 43 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 44 MINIMIZING (*X SUBJECT TO 4*3=B+MHERE X*6+D ARE VECTORS OF LENGTH C 44 SUBROUTINE SIMPLE IS A MOUIFIED VERSION OF A PROGRAM AVAILABLE C 44 WITH SHARE DISTRIBUTING AGENCY+REFERENCE NUMBER SOA 3384 AND C 50 WRITTEN HY R-J-CLASEN IREFERENCE NUMBER 151) C 51 C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 54			č	11	
000 6 11142 C 33 B(I)=0. C 34 CONTINUE C 35 B(K3)=1. C 36 D0 9 1=1.MR3 C 37 C(I)=0. C 36 IF (1.E0.MR1) C(I)=1. C 37 CONTINUE C 37 SUBROUTINE SIMPLE IS NOW GOING TO BE CALLED.ANY ALTERNATIVE C 40 C 40 C 41 CONTINUE C 42 CHUICE TO THIS SUBROUTINE IS ALLOWABLE FUR THE USER AS LONG AS IT C 43 PERFORMS THE FOLLOWING OPENATION. C 45 SUBROUTINE SIMPLE SOLVES A LIMEAR PROGRAMAING PROBLEM OF C 40 MINIMIZING C+X SUBJECTI TO A+X=B.WHERE X:6.0 ARE VECTORS OF LEMOTH C 45 SUBROUTINE SIMPLE IS A MODIFIED VENSION OF A PROGRAM AVAILABLE C 47 WITH SHARE DISTRIBUTING AGENCY.REFERENCE NUMBER SOA 3384 AND C 30 WRITTEN HY R.J.CLASEN IREFERENCE NUMBER (5) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 51 NUMBER (6). C 54			č	32	
CONTINUE CONTINUE B(K3)=1: DO 9 [=1;MR3 C(1)=0. IF (1:E0:MR1) C(1)=1: CONTINUE			č	33	No.
BIK31=1. C 35 D0 9 1=1+MR3 C 36 C11=0. C 37 IF 11-E0.MR1) C11=1. C 39 CONTINUE C 39 CONTINUE C 39 CUBROUTINE SIMPLE IS NOW GUING TO BE CALLED.ANY ALTERNATIVE C 40 C 40 C 40 CHUICE TO THIS SUBROUTINE IS ALLOWABLE FUR THE USER AS LUNG AS IT C 43 PERFORMS THE FOLLOWING OPENATION. C 43 SUBROUTINE SIMPLE SUBLECT TO A*X=5.WHERE X:C:O ARE VECTORS OF LEMOTH C 45 MINIMIZING C*X SUBJECT TO A*X=5.WHERE X:C:O ARE VECTORS OF LEMOTH C 47 MR3:MR3:K3 RESPECTIVELY.AND A IS A MATRIX OF SIZE (K3:MR3) C 40 SUBROUTINE SIMPLE IS A MOUIFIED VERSION OF A PROURAM AVAILABLE C 49 WITH SHARE DISTRIBUTING AGENCY.KEFENENCE NUMBER SOA 3384 AND C 50 WRITTEN HY R.J.CLASEN (KEFERENCE NUMBER (5)) C 51 C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 C 54			č	34	
DO 9 I=1+MR3 C(1)=U. IF (1-EU-MR1) C(1)=1. C 34 CONTINUE C 40 C 40 C 40 C 41 C 42 C 42 C 42 C 43 C 44 C 42 C 42 C 42 C 42 C 43 C 44 C 43 C 44 C 45 C 5 C 5 C 55 C 55			Č	د د	
CIII=U. CIII=I. CIII=I. IF (I.EU.MR1) C(II=I. CIII=I. CIII=I. CONTINUE CIII=I. CIII=I. SUBROUTINE SIMPLE IS NOW GOING TO BE CALLED.ANY ALTERNATIVE CIIII CHUICE TO THIS SUBROUTINE IS ALLOWABLE FUR THE USER AS LUNG AS II CIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII			С	36	'\
IF (I.EU.MR1) C(I)=1. C 38 CONTINUE C 40 SUBROUTINE SIMPLE IS NOW GUING TO BE CALLED.ANY ALTERNATIVE C 42 CHUICE TO THIS SUBROUTINE IS ALLOWABLE FUR THE USEN AS LONG AS IT C 43 PERFORMS THE FOLLOWING OPENATION. C 45 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMMING PROBLEM OF C 45 MINIMIZING C*X SUBJECT TO A*X=B*#MERE X*C*D ARE VECTORS OF LENGTH C 47 MR3*MR3*K3 RESPECTIVELY*AND A IS A MATRIX OF A PROURAM AVAILABLE C 49 WITH SHARE DISTRIBUTING AGENCY*REFERENCE NUMBER SOA 3384 AND C 50 WRITTEN HY R*J*CLASEN IREFERENCE NUMBER (57) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 53 K C 54			C	37	
CONTINUE CONTIN		IF (L-FO-MP1) CITIAN	C	36	
CONTINUE SUBROUTINE SIMPLE IS NOW GUING TO BE CALLED.ANY ALTERNATIVE C 41 C 42 CHUICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USER AS LONG AS IT PERFORMS THE FOLLOWING OPERATION. SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMMING PROBLEM OF MINIMIZING C+X SUBJECT TO A+X=B+WHERE X+C+D ARE VECTORS OF LENGTH C 45 C 55 C			C	39	
C 41 C 42 C 42 CHUICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USER AS LONG AS IT PERFORMS THE FOLLOWING OPERATION. SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMMING PROBLEM OF MINIMIZING COX SUBJECT TO AOX=BOOMERE XOCOD ARE VECTORS OF LENGTH C 45 MINIMIZING COX SUBJECT TO AOX=BOOMERE XOCOD ARE VECTORS OF LENGTH C 47 MR3,MR3,K3 RESPECTIVELY,AND A IS A MATRIX OF SIZE (K3,0R3) SUBROUTINE SIMPLE IS A MODIFIED VERSION OF A PROGRAM AVAILABLE VITH SHARE DISTRIBUTING AGENCY,HEFERENCE NUMBER SOA 3384 AND C 50 WRITTEN HY R.J.CLASEN (HEFERENCE NUMBER (57) A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE NUMBER (6).			C	40	
SUBROUTINE SIMPLE IS NOW GUING TO BE CALLED.ANY ALTERNATIVE C 43 CHUICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USER AS LONG AS IT C 44 PERFORMS THE FOLLOWING OPERATION. C 45 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMAING PROBLEM OF C 40 MINIMIZING C#X SUBJECT TO A#X=B+WHERE X+C+D ARE VECTORS OF LENGTH C 47 MR3+MR3+K3 RESPECTIVELY+AND A IS A MATRIX OF SIZE (K3+MR3) C 40 SUBROUTINE SIMPLE IS A MODIFIED VERSION OF A PROGRAM AVAILABLE C 49 WITH SHARE DISTRIBUTING AGENCY+REFERENCE NUMBER SOA 3384 AND C 50 WRITTEN HY R+J=CLASEN (REFERENCE NUMBER (57) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6). C 54		:	C	41	
CHUICE TO THIS SUBROUTINE IS ALLOWABLE FOR THE USER AS LONG AS IT C 43 PERFORMS THE FOLLOWING OPERATION. C 45 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMMING PROBLEM OF C 46 MINIMIZING C+X SUBJECT TO A+X=B+WHERE X+C+D ARE VECTORS OF LENGTH C 47 MR3+MR3+K3 RESPECTIVELY+AND A IS A MATRIX OF SIZE (K3+MR3) C 40 SUBROUTINE SIMPLE IS A MODIFIED VERSION OF A PROGRAM AVAILABLE C 49 WITH SHARE DISTRIBUTING AGENCY+REFERENCE NUMBER SOA 3384 AND C 50 WRITTEN HY R+J+CLASEN (REFERENCE NUMBER (57) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6)+ C 54		SUBROUTINE'S THREE TE MOVE COMPLETE TO THE PERSON	c	42	
PERFORMS THE FOLLOWING UPERATION. C 45 SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMMING PROBLEM OF C 45 MINIMIZING C+X SUBJECT TO A+XTH+WHERE X+C+D ARE VECTORS OF LENGTH C 47 MR3+MR3+K3 RESPECTIVELY+AND A IS A MATRIX OF SIZE (K3+MR3) C 40 SUBROUTINE SIMPLE IS A MODIFIED VERSION OF A PROGRAM AVAILABLE C 49 WITH SHARE DISTRIBUTING AGENCY+REFERENCE NUMBER SOA 3384 AND C 50 WRITTEN HY R+J+CLASEN (REFERENCE NUMBER (57) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6)+ C 54		CHUICE TO THIS SUBDUTTING TO BE CALLED. ANY ALTERNATIVE	C	43	
SUBROUTINE SIMPLE SOLVES A LINEAR PROGRAMMING PROBLEM OF C 45 MINIMIZING C*X SUBJECT TO A*X=B*WHERE X*C*D ARE VECTORS OF LENGTH C 47 MR3*MR3*K3 RESPECTIVELY*AND A IS A MATRIX OF SIZE (K3*MR3) C 40 SUBROUTINE SIMPLE IS A MODIFIED VERSION OF A PROGRAM AVAILABLE C 49 WITH SHARE DISTRIBUTING AGENCY*REFERENCE NUMBER SOA 3384 AND C 50 WRITTEN HY R*J*CLASEN (REFERENCE NUMBER (57) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6)* C 54		PERFORMS THE FOLLOWING OF MATTER ALLOWABLE FOR THE USER AS LONG AS IT	C	44.46	
MINIMIZING CAR SUBJECT TU AARABAMERE XACAD ARE VECTORS OF LENUTH C 4/ MR3+MR3+K3 RESPECTIVELY+AND A IS A MATRIX OF SIZE (K4+NR3) C 40 SUBROUTINE SIMPLE IS A MODIFIED VERSION OF A PROURAM AVAILABLE C 49 WITH SHARE DISTRIBUTING AGENCY+REFERENCE NUMBER SDA 3384 AND C 50 WRITTEN BY R+J+CLASEN (REFERENCE NUMBER (5/) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6)+ C 54		SUBROUTINE SIMPLE SOLVES A LINEAR DAMAGE AND ADDRESS	C	45	
MR3+MR3+K3 RESPECTIVELY+AND A IS A MATRIX OF SIZE (K3+MR3) C 40 SUBROUTINE SIMPLE IS A MODIFIED VERSION OF A PROGRAM AVAILABLE C 49 WITH SHARE DISTRIBUTING AGENCY+NEFERENCE NUMBER SDA 3384 AND C 50 WRITTEN HY R+J+CLASEN (REFERENCE NUMBER (5/) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6)+ C 54		MINIMIZING COX SUN INCT TO AAVEN WITH A AVEN AND AND AND AND AND AND AND AND AND AN	C.	40	
SUBROUTINE SIMPLE IS A MODIFIED VERSION OF A PROGRAM AVAILABLE C 49 WITH SHARE DISTRIBUTING AGENCY, REFERENCE NUMBER SDA 3384 AND C 50 WRITTEN BY R.J.CLASEN (REFERENCE NUMBER (5/) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6). C 53		MR3+MR3+K3 RESPECTIVELY AND A TE A MATTINE AND ARE VECTORS OF LENUTH	¢	41	
WITH SHARE DISTRIBUTING AGENCY, KEFEKENCE NUMBER SDA 3384 AND C 50 WRITTEN HY R.J.CLASEN IREFERENCE NUMBER (5/) C 51 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6). C 53 C 54		SUBROUTINE SIMPLE IS A MOUSELLO MAINTAIN OF SIZE (KS, MR3)	C	40	
WRITTEN HY R.J.CLASEN (REFERENCE NUMBER (5/) C 50 A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6). C 53 C 54		WITH SHARE DISTRIBUTING AGENCY, HELLURGE MENTER ANALLABLE	C	4 Y	
A COPY OF THE LISTING IS AVAILABLE AS INDICATED IN REFERENCE C 52 NUMBER (6). C 53 C 54		WRITTEN HY R.J.CLASEN THEFEDENCE NUMBER SDA 3384 AND	C	>0	
NUMBER (6). C 52		A COPY OF THE LISTING IS AVAILABLE AT INCLUDED IN THE AND A COPY OF THE LISTING IS AVAILABLE AT INCLUDED IN THE ADDRESS OF THE	C	51	
C 33 C 54		NUMBER (6).	C	52	
د ج د د د د د د د د د د د د د د د د د د			Ç.	>>	
•		*	C	54	

۲

,

źà

÷,

C NA AND IFLAG ARE TO BE SPECIFIED BEFORE CALLING SIMPLE C SS C THAG IS SET DUMENSION OF ARRAY A AND IS SET EQUAL TO K3 C SS C THE SET EQUAL TO ZERO C SS C THE SET EQUAL TO ZERO C SS C THE SET DIMENSION OF ARRAY A AND IS SET EQUAL TO K3 C SS C THE FOLLOWING SUBJECTION WAS STARGE SALES TO UE UNKENT U		*		Ň	
<pre>The Still Decay in the still of still of the still o</pre>	C	NA AND IFLAG ARE TO BE SPECIFIED BEFORE CALLING SIMPLE	c	••	
<pre>X IS A VECTOR OF DIMERSION INATIAN A AND IS SET EQUAL TO KS THE FOLLOWING SUBSCHIPTED VARIABLES ANE PART OF THE ANGUMENT LIST C 59 C THE FOLLOWING SUBSCHIPTED VARIABLES ANE PART OF THE ANGUMENT LIST C 59 C THE FOLLOWING SUBSCHIPTED VARIABLES ANE PART OF THE ANGUMENT LIST C 59 C THE CALLING PROGRAM (INITIAN) C 61 C E IS A TEMPOROGNAM (INITIAN) C 62 C E IS A TEMPOROGNAM (INITIAN) C 63 C C E IS A TEMPOROGNAM (INITIAN) C 63 C C E IS A TEMPOROGNAM (INITIAN) C 63 C C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 63 C E IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 73 C C IS A TEMPORATY STORAGE MATHING OF DIMENSION (INA,NA) C 73 C C IS A TEMPORATY STORAGE EXAMINATION (INA,NA) C 73 C C IS A TEMPORATY STORAGE EXAMINES A TEMPORATY STORAGE EXAMINES C E IS A TEMPORATY STORAGE EXAMINES A TEMPORATY STORAGE EXAMINES C E IS A TEMPORATY STORAGE EXAMINES A TEMPORATY STORAGE EXAMINES C E IS A TEMPORATY STORAGE EXAMINES A TEMPORATY STORAGE EXAMINES C E IS A TEMPORATY STORAGE EXAMINES A TEMPORATY STORAGE EXAMINES C E IS A TEMPORATY STORAGE EXAMINES A TEMPORATY STORAGE EXAMINES C E IS A TEMPORATY STORAGE EXAMINES A TEMPORATY STORAGE EXAMINES C E IS A TEMPORATY STORAGE EXAMINES A TEMPORATY STORAGE EXAMINES A</pre>	ſ	NA IS THE FIRST DIMENSION OF ADDAM A AND AS SEE	č	56	
C THE FOLLOWING SUBSCRIPTED VARIABLES ARE PART OF THE ARGUMENT LIST C 53 OF SIMPLE AND ARE TEMPORARY STURAGE SPACES TO DE VIMENSION AN C 64 C THE CALLING PROGRAM ININIMARY C 51 A VECTOR OF LENGTH CAMPORARY STURAGE VECTORS OF UMENSION AN C 64 C E IS A TEMPORARY STORAGE MATHIX OF DIMENSION INA.NA C 64 C E IS A TEMPORARY STORAGE MATHIX OF DIMENSION INA.NA C 64 C E IS A VECTOR OF LENGTH 6-UPON COMPLETION OF THE EXECUTION OF C 64 C SIMPLE, KOLIJ=O IF THE LIMEAR PROGRAMMING PROBLEM WAS FEASIBLE. C 65 C F C F C F C F C F C F C F C F	č	X IS A VECTOR OF DIMENSION MR3	¢	51	
C UF SIMPLE AND ANE TEMPORARY STURADE SPACES TO DE DIMENSIONED IN C.S. THE CALLING PROBERM ININIMAX: PSUJHIXXAYY AND PE ARE TEMPORARY STURADE VECTORS OF DIMENSION NA C E IS A TEMPORARY STORAGE MATHIX OF DIMENSION IN ANANA) C SIMPLE: ACTION OF LENGTH GUPON COMPLETION OF THE LAELUTION OF C SIMPLE: ACTION OF LENGTH GUPON COMPLETION OF THE LAELUTION OF C SIMPLE: ACTION OF LENGTH GUPON COMPLETION OF THE LAELUTION OF C SIMPLE: ACTION OF LENGTH GUPON COMPLETION OF THE LAELUTION OF C SIMPLE: ACTION OF LENGTH GUPON COMPLETION OF THE LAELUTION OF C SIMPLE: ACTION LIES IN XIJJ:J=1,4R3 C C ST C SIMPLE: ACTION COMPLETION OF THE LAELUTION OF C SUBROUTINE MET2 (KARIHAMANISTICH STATEMENT OF THE LAELUTION OF C SUBROUTINE MET2 (KARIHAMANISTALI): X2111: H2111 C ONTINUE C SUBROUTINE MET2 (KARIHAMANISTALI): X2111: H2111 C DIMENSION MAXALILE GRADINNAALI'S X2111: H2111 C DIMENSION MAXALILE GRADINNAALI'S X2111: H2111 C DIMENSION TO ANANCIES VARIABLE UNDER SIGNET OF DUBULE DIMENSION OF C SUBROUTINE ARRAY CONVERTS UATA FROM SINGLE TO DOUBLE DIMENSION OF C C C C C C C C C C C C C C C C C C C	C	THE FOLLOWING SUBSCRIPTED VARIABLES ARE PART OF THE ANGUMENT LIST	ç	58	
C THE CALLING PROGRAM (MINIMAX) PS-UN-LING PROGRAM (MINIMAX) C PS-UN-LING PROFAME TEMPORARY STURAGE VECTORS OF DIMENSION (MA) C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, KOIII-O IF THE LINEAK PROGRAMMING PROBLEM WAS FEASIBLE. C SIMPLE, C SIMPLE, C SIMPLE, SIMPLE, C SIMPLE, SIMPLE, SIMPLE, SIMPLE, C SIM	Ç	OF SIMPLE AND ARE TEMPURARY STURAGE SPACES TO BE DIMENSIONED IN	Ľ,	27	
<pre>C PSIJNIANTI AND PE ARE TEMPORARY STORAGE VECTORS OF DIRENSION NA C E IS A TEMPORARY STORAGE MATKIN OF DIRENSION (NA,NA) C CSI C KU IS A VECTOR OF LENGTH 6.UPON COMPLETION OF THE EXECUTION OF C SIMPLE; KOITION LIES IN X(J)+J=1+NR3 C C C THE SOLUTION LIES IN X(J)+J=1+NR3 C C C C C C C C C C C C C C C C C C C</pre>	Ç	THE CALLING PROGRAM IMINIMAXI	ر د	61	
<pre>c bis A vector of is Johange MARIX OF DIMENSION (MANA) c c) SIMPLE, KOLLIPO IF THE LINEAK PROGRAMMING PROBLEN WAS FEASIBLE. c c) c sIMPLE, KOLLIPO IF THE LINEAK PROGRAMMING PROBLEN WAS FEASIBLE. c c) c the Solution lies in X(j),j=1,NR3 c c) c file(a=0</pre>	ć	E IS A TEMPORARY STORAGE VECTORS OF DIMENSION NA	Ę	62	
<pre>SIMPLE: KOTITED OF THE LIMEAN PROBLEM WAS FEASIBLE. C 66 THE SOLUTION LIES IN X(J)+J=1+NR3 C 66 IFLAG=0</pre>	č	KU IS A VECTOR OF LENGTH A HUGH COMPLETIVE OF THE AND	C	63	
C THE SOLUTION LIES IN XIJ)+J+1+NR3 C THE SOLUTION LIES IN XIJ)+J+1+NR3 C C C C C C C C C C C C C C C C C C C	è	SIMPLE' KOILISO IF THE LINEAR PROGRAMMING PROPIEN WAS STAND	C	64	
C IFLAG-U C 60 IFLAG-U C 67 CALL SIMPLE (IFLAG.K3:MH3,A.B.C.KU.K.PS.JH.XX.YY.PL.E.NA) C 70 DO 10 J=1.MR C 72 CALL SIMPLE (IFLAG.K3:MH3,A.B.C.KU.K.PS.JH.XX.YY.PL.E.NA) C 71 DO 10 J=1.MR C 72 IO CONTINUE C 75 DO 12 I=1.K C 75 R1(1)=R1(1)=R1(1)=X1(J) C 76 R1(1)=R1(1)=R1(1)=X1(J) C 76 C 77 C CONTINUE C 79 C CONTINUE C 79 C CONTINUE C 79 C CONTINUE C 80 C 80	c	THE SOLUTION LIES IN XIJI +J=1+NR3	Ç	65	
C IFLAG=0 C 68 -NA=K3 C 69 CALL SIMPLE (IFLAG,K3,MK3,A,B,C,KU,X,PS,JH,XX,YY,PE,E,NAI C 70 D0 10 J=1,MR C 72 10 CONTINUE C 73 D0 12 I=1+K C 73 D0 12 I=1+K C 75 R1(1)=0. C 75 D0 11 J=1,MR C 75 D0 11 J=1,MR C 77 11 CONTINUE C 76 12 CONTINUE C 76 13 CONTINUE C 79 XISUM=N. C 80 KISUM=XISUM=XI(J) C 80 C 80	C		Č	66	
IPLAG=0 C 60 CALL SIMPLE (IFLAG=K3=MR3=A+B+C+KU+X=P5=JH=XX=YY=PE=E=NA) C 70 DO 10 J=1=MR C 72 X1(J)=X(J) C 73 DO 12 I=1=K C 74 R1(I)=0 C 75 DO 12 I=1=K C 74 R1(I)=0. C 75 DO 12 I=1=K C 76 R1(I)=0. C 77 DO 12 I=1=K C 77 R1(I)=0. C 77 DO 12 J=1=MR C 77 11 CONTINUE C 78 12 CONTINUE C 78 13 CONTINUE C 80 X1SUM=x1SUM=X1SUM=X1(J) C 83 13 CONTINUE C 84 FUURN=2-0.5-MULTROW-ICOL=LEM=MAX=MAX=MAX=MAX=MAX=MAX=MAX=MAX=MAX=MA	C		ι ι	67	•
CALL SIMPLE (IFLAG.K3.MR3.A.B.C.KU.X.PS.JH.XX.YY.PE.E.NA) CALL SIMPLE (IFLAG.K3.MR3.A.B.C.KU.X.PS.JH.XX.YY.PE.E.NA) C 70 D 10 J=1.MR C 77 D CONTINUE C 77 D CONTINUE C 77 D CONTINUE C 77 R 111J=R1(1)+A(1+J)+X1(J) C 77 C 76 C 77 C 77			č	69	
D0 10 J=1+RR C 71 X1(J)=X(J) C 73 D0 12 1=1+K C 73 D0 12 1=1+K C 75 R1(1)=0. C 75 D0 11 J=1+RR C 76 R1(1)=A(L,J)=X1(J) C 76 D0 11 J=1+RR C 77 11 CONTINUE C 77 12 CONTINUE C 78 13 CONTINUE C 80 14 CONTINUE C 81 15 CONTINUE C 81 16 CONTINUE C 82 17 CONTINUE C 81 18 CONTINUE C 82 19 CONTINUE C 83 10 MR2-0H-MOR C 83 11 CONTINUE C 83 12 CONTINUE C 83 13 CONTINUE C 83 14 UMR2-0H-MOR C 83 15 C C 88 16 C C 87 17 UMRA-10 MUE C 83 18 C C 88 19 UMRA-10 MUE C 80 10 UMRA-20-H-MOR C 88 10 UMRA-10 MUE C 10++++++++++++++++++++++++++++++++++++		- DAHAJ CALL STADLE (TELAG VA MUDIA V CIMUM VALVE V	c	70	
X1(1)=X(J) C 72 10 CONTINUE C 73 10 CONTINUE C 74 11 CONTINUE C 76 12 CONTINUE C 77 11 CONTINUE C 76 12 CONTINUE C 76 13 CONTINUE C 78 14 CONTINUE C 78 15 CONTINUE C 78 16 CONTINUE C 78 17 CONTINUE C 78 18 CONTINUE C 78 19 CONTINUE C 78 10 CONTINUE C 88 13 CONTINUE C 88 14 C 88 C 88 15 CONTINUE C 89 0 10 INTEGER UNIT GANDINHNAX:11, KANDINHNAX:11, K2(11) U 4 10 INTEGER UNIT GANDINHNA:11, H(K1+1), G(11), H(K1+1), H(L1), H(L1), H(L1), H(L1), H(L1), H(L1		DO 10 JEL AND	ć	11	
10 CONTINUE C 73 11 CONTINUE C 74 11 Polizialisk C 76 11 CONTINUE C 77 11 CONTINUE C 77 12 CONTINUE C 77 12 CONTINUE C 77 12 CONTINUE C 77 13 CONTINUE C 78 14 CONTINUE C 81 15 CONTINUE C 81 16 CONTINUE C 81 17 CONTINUE C 81 18 CONTINUE C 83 19 CONTINUE C 83 10 CONTINUE C 83 10 CONTINUE C 84 11 CONTINUE C 85 12 CONTINUE C 84 13 CONTINUE C 85 14 DIMENSION TAXILINGARUALINALINALINALINALINALINALINALINALINALIN		{L}X={L}X={L}X={L}X={L}X={L}X={L}X={L}X=	C	72	
DO 12 1-1+K R1(1)=0. C 75 C 76 DO 11 J=1+MR R1(1)=R1(1)=A(1,J)=X1(J) C 77 R1(1)=R1(1)=A(1,J)=X1(J) C 77 C 80 C	10	CONTINUE	ç	73	
R1(1)=0. C 75 00 11 J=1.HR C 77 11 CONTINUE C 77 12 CONTINUE C 79 12 CONTINUE C 79 12 CONTINUE C 80 13 CONTINUE C 80 14 CONTINUE C 80 15 CONTINUE C 80 16 CONTINUE C 83 17 CONTINUE C 83 18 CONTINUE C 85 19 CONTINUE C 85 10 CONTINE HET2 (K+x1+M+NN+YAAX-OHAD+UELTA+IPRINI+ISP+UNIT+X2+X25 U 10 C B5 C 89- U U 10 HR HET2 (K+x1+M+N+YAAX-OHAD+UELTA+IPRINI+ISP+UNIT+X2+X25 U U 10 HET2 K+x1+M+N+N+YAAX-OHAD+UELTA+IPRINI+ISP+UNIT+X2+X25 U U U 10 HET2 IK+x1+M+N+N+YAAX-OHAD+UELTA+IPRINI+ISP+UNIT+X2+X25 U <t< td=""><td></td><td>DO 12 I=1+K</td><td>Ċ</td><td>76</td><td></td></t<>		DO 12 I=1+K	Ċ	76	
DO 11 J=1+MR R111)=R1(1)+A11,J)*X1(J) 11 CONTINUE C 77 12 CONTINUE X15UM+N, C 80 C 13 J=1+MR C 83 CONTINUE C 85 C 86 C 90 C 86 C 90 C 9		R1(1)=0.	č	76	
A TITITATI(1)*AT(1)*AT(1) C 78 11 CONTINUE C 79 12 CONTINUE C 80 D0 13 J14MR C 81 13 CONTINUE C 83 13 CONTINUE C 83 13 CONTINUE C 85 14 CONTINUE C 85 15 CONTINUE C 85 16 C 67 C 17 C 85 C 86 18 CUBROUTINE MET2 (K*K1*MMCNN+YAAX3GHADO*UELTA*IPRINT+ISP*UNIT*X2*X4S U 2 19 DIMENSION YMAX(1)* GRAD(1)***MAX1** U 3 10 DIMENSION YMAX(1)* GRAD(1)************************************		DO 11 J=1.MR	č	17	
11 CONTINUE C 79 12 CONTINUE C 80 13 CONTINUE C 81 14 CONTINUE C 81 15 CONTINUE C 81 13 CONTINUE C 83 14 CONTINUE C 83 15 CONTINUE C 83 16 C 84 C 84 17 CONTINUE C 84 18 CONTINUE C 84 19 C 84 C 84 10 RETURN C 84 11 CONTINUE C 84 12 CONTINUE C 84 13 CONTINUE C 84 14 CONTINUE C 84 15 C 84 C 94 16 C 84 C 94 17 DIMENSION DINMAX+11+ GRADINAX+11+ X211+ K211+ K211 U 4 18 CONTINUE C 10 </td <td>11</td> <td>CONTINUE CONTINUE</td> <td>C</td> <td>78</td> <td></td>	11	CONTINUE CONTINUE	C	78	
XISUM=0. C 80 D0 13 J=1.MR C 81 XISUM=XISUM=X1(J) C 83 13 CONTINUE C 83 RETURN C 85 END C 86 C C 85 C C 86 C C 85 C C 87 C C 87 C C 87 DIMENSION YMAX(1) GRADUNNHAX; DIMENSION YMAX(1) GRADUNNHAX; DIMENSION YMAX(1) GRADUNNHAX; DIMENSION DINRMAX: D 10 D J MR1=MR+1 D 3 D0 2 1=1;MR D 10 D0 2 1=1;MR D 10 D0 1 J=1;K D 10 D1 1 C CONTINUE D 12 C CONTINUE D 14 D 15 D 15 D 16 D 16 D 17 D 16 C CONTINUE D 16 C CONTINUE D 16	12		¢	79	
D0 13 J=1+MR x1SUM=x1SUM=x1SUM=x1(J) 13 CONTINUE RETURN END C 85 C 86 C 87 C 87 C 87 C 88 C 87 C 87 C 88 C 88 C 89- D 1 LUM:R2+D.+M.U.FIROW.FICUL:LL:MM:NRMAX:GNAD:DELTA-IPRINI:JSP:UNIT:X2+X2S U 2 LUM:R2+D.+M.U.FIROW.FICUL:LL:MM:NRMAX:GNAD:DELTA-IPRINI:JSP:UNIT:X2+X2S U 2 D 14ENSION YMAX(1): GRADINNAX:J1, X2(1), K2(1) LOGICAL 1PRINT:ISP INTEGER UNIT DIMENSION D (MRMAX:1): H(K1:1): U(1): IRDM(1): ICOL(1): LL(1): MH(1) U 9 D0 2 I=1.MR D0 2 I=1.MR D0 2 I=1.MR D0 1 J=1:K D 14 C 00TINUE C CONTINUE C CONTINUE C CONTINUE C CONTINUE C CONTINUE C CALL ARHAY (2:MR:K:NRMAX:K+D:D) C C C C C C C C C C C C C C C C C C C	• •	X15UM+0.	C	80	
x1SUM-x1SUM+x1(J) C 83 13 CONTINUE C 83 rETURN C 85 END C 86 C Subroutine SUBROUTINE MET2 (K-KLI-MK+NN+YMAX+GHAD+DELTA+IPRINI+ISP+UNIT+X2+X25 D 1 SUBROUTINE MET2 (K-KLI-MK+NN+YMAX+GHAD+DELTA+IPRINI+ISP+UNIT+X2+X25 D 1 SUBROUTINE MET2 (K-KLI-MK+NN+YMAX+GHAD+DELTA+IPRINI+ISP+UNIT+X2+X25 D 2 SUBROUTINE MET2 (K-KLI-MK+NN+YMAX+GHAD+DELTA+IPRINI+ISP+UNIT+X2+X25 U 2 IUM+R2+D+H+0+IROW+ICOL+L+MH+NN+YMAX+GHAD+DELTA+IPRINI+ISP+UNIT+X2+X25 U 2 SUBROUTINE MET2 (K-KLI-MK+NN+YMAX+GHAD+DELTA+IPRINI+ISP+UNIT+X2+X25 U 2 IUM+R2+D+H+0+IROW+ICOL+L+MH+NN+YMAX+GHAD+DELTA+IPRINI+ISP+UNIT+X2+X25 U 2 IUM+R2+D+H+0+IROW+ICOL+L+MH+NH+XX+ U 2 IUM+R2+D+H+0+IROW+ICOL+R U 2 IUM+R2+D+1 U 4 UCMITHE MAX+1+ INTEGER U 14 U 14 U 14		DO 13 J=1+MR	ç	81	
13 CONTINUE RETURN END C BB C C C B7 C B8 C B8 C B9- U BROUTINE MET2 (K+K1+MK+NN+YMAX5GHAD+UELTA+IPRINI+ISP+UNIT+X2+X2S U C B8 C B9- U SUBROUTINE MET2 (K+K1+MK+NN+YMAX5GHAD+UELTA+IPRINI+ISP+UNIT+X2+X2S U C B8 C B9- U S U C B9- U S U C B9- U S U C C B8 C B9- U S U C C B9- U S U C C B9- U S U C C C C C C C C C C C C C C C C C		X1SUH=X1SUH+X1(J)	č	0∠ ∺ 4	
RETURN END C B5 END C B6 C B7 C B8 C B8	13	CONTINUE	č	84	
END C BB C C BC C C BT SUBROUTINE MET2 (K-K1+NK+NN+YAXX+UHAD+UELTA+IPRINI+ISP+UNIT+X2+X2S U 2 IUM+R2+D+H+U+IROW+ICUL+LL+MM+NKMAX* DIMENSION YMAX(1)+ GRAU(NNMAX+1)+ X2(1)+ K2(1) LOGICAL IPRINT+ISP U 4 INTEGER UNIT DIMENSION D(NRMAX+1)+ H(K1+1)+ U(1)+ IROW(1)+ ICOL(1)+ LL(1)+ MM(1 U 4 U 6 DIMENSION D(NRMAX+1)+ H(K1+1)+ U(1)+ IROW(1)+ ICOL(1)+ LL(1)+ MM(1 U 7 1) MRI=MR+1 U 6 DO 2 I=1+MR U 9 DO 2 I=1+MR U 9 IO 2 I=1+K U 10 CONTINUE C CONTINUE C CONTINUE C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION UN U 117 VICE VENSA+IT ENABLES VARIABLE UIMENSIONING UF DATA MATHICES Lift U 10 C ALL ARHAY (2+MR+K+NRMAX+K+D+D) U 221 CALL ARHAY (2+MR+K+NRMAX+K+D+D) U 221		RETURN	č	85	
C 87 C 88 C 89- SUBROUTINE MET2 (K+K1+MK+NR+YMAX+GHAD+UELTA+IPRINI+ISP+UNIT+X2+X2S U 2 IUM+R2+D+H+GFIROW+ICOL+L+MH+NHMAX+ DIMENSION YMAX(1)+ GRAU(NRNAX+1)+ X2(1)+ K2(1) LOGICAL IPRINT+ISP INTEGER UNIT DIMENSION DINRMAX+1)+ H(K1+1)+ G(1)+ IROW(1)+ ICOL(1)+ LL(1)+ MM(1) U 4 U 5 U 4 U 5 U 4 U 5 U 6 D 10 U 7 1) MR1=MR+1 D 7 1) MR1=MR+1 D 7 1) CONTINUE C CONTINUE C CONTINUE C CONTINUE C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION ON U 14 C 14 C 14 C 14 C 2 C CALL ARHAY (2+MR+K+NRMAX+K+D+D) C C C C C C C C C C C C C C C C C C C	r		C	86	
C 88 C 89- U 1 IUM+R2+D+H+U FROW+ICOL+LL+MM+NR+YMAX+GHAD+UELTA+IPRINT+ISP+UMIT+X2+X25 U 2 IUM+R2+D+H+U FROW+ICOL+LL+MM+NRHAX+ DIMENSION YMAX(1)+ GRAU(NNHAX+1)+ X2(1)+ K2(1) U 3 U 4 LOGICAL IPRINT+ISP U 5 INTEGER UNIT DIMENSION DINRMAX+1)+ H(K1+1)+ U(1)+ IROW(1)+ ICOL(1)+ LL(1)+ MM(1) U 6 DIMENSION DINRMAX+1)+ H(K1+1)+ U(1)+ IROW(1)+ ICOL(1)+ LL(1)+ MM(1) U 7 1) MR1=MR+1 U 9 D0 2 I=1+MR D 10 U 9 D0 2 I=1+MR D 10 U 14 CONTINUE C ONTINUE C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION UN U 17 VICE VENSA+IT ENABLES VARIABLE UIMENSIONING UP DATA MATHICES Line U 19 C CALL ARHAY (2+MR+K+NRMAX+K+D+D) C C CALL ARHAY (2+MR+K+NRMAX+K+D+D)	ċ		C	87	
C SUBROUTINE MET2 (K+K1+MK+NR+YHAX+GHAD+UELTA+IPRINI+ISP+UMIT+X2+X2S U 2 IUM+R2+D+H+U+IROW+ICOL+LL+MM+NRHAX+ DIMENSION YMAX(1)+ GRADINNHAX+1), X2(1)+ H2(1) U 4 LOGICAL IPRINT+ISP INTEGER UNIT DIMENSION D(NRMAX+1)+ H(K1+1)+ U(1)+ IHOW(1)+ ICOL(1)+ LL(1)+ MM(1) U 7 1) MR1=MR+1 D0 2 I=1+MR D0 2 I=1+MR D0 2 I=1+MR D1 = 1+K D1 = 1+K D1 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 =	ċ		¢	88	— .
SUBROUTINE MET2 (K+K1+MR+NR+YMAX+GRAD=JUELTA+IPRINI+ISP+UNIT+X2+X25 U IUM+R2+D+H5U+IROW+ICOL+LL+MR+NRHAX+ U DIMENSION YMAX(1)+ GRADINNAX+1+, X2(1)+ R2(1) U LOGICAL IPRINT+ISP U INTEGER UNIT U DIMENSION DINRMAX+1+ H(K1+1)+ U(1)+ IROW(1)+ ICOL(1)+ LL(1)+ MN(1) U NR1=MR+1 U D0 2 I=1+MR U D0 1 J=1+K U D11 CONTINUE U CONTINUE U 13 CONTINUE U 14 CONTINUE U 15 C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION ON U C If CALL ING PROGRAM-REFERENCE NUMBER (2)+ U C CALL ARHAY (2+MR+K+NRMAX+K+D+D) U	C	* * * * * * * * * * * * * * * * * * * *	C	89-	
IUM+R2+D+H+U+IROW+ICUL+LL+MM+NRMAX; U J DIMENSION YMAX(1)+ GRAU(NRMAX+1)+ X2(1)+ H2(1) U 4 LOGICAL IPRINT+ISP U 4 INTEGER UNIT U 6 DIMENSION D(NRMAX+1)+ H(K1+1)+ Q(1)+ IROW(1)+ ICOL(1)+ LL(1)+ MM(1) 0 1) U 8 MR1=MR+1 U 9 D0 ? I=1+MR D 10 D0 1 J=1+K U 11 CONTINUE U 14 CONTINUE U 14 CONTINUE U 14 C U 15 C U 16 C U 17 VICE VENSA-IT ENABLES VARIABLE UIMENSIONING UF UATA MATHICES Line U C U 17 VICE VENSA-IT ENABLES VARIABLE UIMENSIONING UF UATA MATHICES Line U C U 17 VICE VENSA-IT ENABLES VARIABLE UIMENSIONING UF UATA MATHICES Line U C U U C U U U C C U U C <td< td=""><td></td><td>SUBROUTINE METZ (K+K1+MK+NK+YAAX+GHAD+UELTA+IPRINT+ISP+UNIT+X++X+S</td><td>ц Ц</td><td>5</td><td></td></td<>		SUBROUTINE METZ (K+K1+MK+NK+YAAX+GHAD+UELTA+IPRINT+ISP+UNIT+X++X+S	ц Ц	5	
DIMENSION YMAX(1), GRAD(NRNAX,1), X2(1), H2(1) LOGICAL IPRINT,ISP INTEGER UNIT DIMENSION D(NRMAX,1), H(K1,1), U(1), IROW(1), ICOL(1), LL(1), MM(1) U HR1=MR+1 U D0 2 I=1,MR D0 2 I=1,MR D0 1 J=1,K U(1,J)=GRAD(1,J) CONTINUE CON		1UH+R2+D+H+U+IROW+ICUL+LL+MH+NRHAX	Ū	َ وَ	·
LUGICAL IPRINTISS INTEGER UNIT DIMENSION D(NRMAX+1)+ H(K1+1)+ Q(1)+ IROW(1)+ ICOL(1)+ LL(1)+ MM(1) D 7 1) MR1=MR+1 D 2 D 2 I=1+MR D 3 D 0 2 I=1+MR D 0 9 D 0 1 J=1+K D 10 U 4 D 12 CONTINUE C CONTINUE C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION OR C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION OR C 16 C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION OR C 17 C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION OR C 17 C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION OR C 17 C C LL ARRAY (2+MR+K+NRMAX+K+D+D) C C C C LL ARRAY (2+MR+K+NRMAX+K+D+D) D 22 C C C C C C C C C C C C C C C C C C C		DIMENSION YMAX(1), GRADINHNAX,1), X2(1), R2(1)	υ	4	
DIMENSION D(NRMAX+1)+ H(K1+1)+ Q(1)+ IROW(1)+ ICOL(1)+ LL(1)+ MM(1) D 7 1) MR1=MR+1 D 9 D0 2 I=1+MR D 10 D0 1 J=1+K D 10 D0 1 J=1+K D 12 CONTINUE D 12 CONTINUE D 12 CONTINUE D 14 C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION UN U 17 C VICE VENSA+IT ENABLES VARIABLE DIMENSIONING OF DATA MATHICES LH U 10 C THE CALLING PROGRAM+REFERENCE NUMBER (2)+ C C U 21 C C C C C C U 21 C C C C C C U 21 C C C C C U 21 C C C C C C U 21 C C C C C C U 21 C C C C C C C U 21 C C C C C C U 21 C C C C C C U 21 C C C C C C U 2-MR+K+NRMAX+K+D+D V U 21 C C C C C C C U 21 C C C C C C U 21 C C C C C C U 2-MR+K+NRMAX+K+D+D V U 21 C C C C C C C U 21 C C C C C C C C C C C C C C C C C C C		LUGICAL IPRINISISP	υ	5	
1) MR1=MR+1 U B D0 2 I=1+MR D 10 D0 1 J=1+K U 11 D1+J=GRAD(I+J) D 12 1 CONTINUE U 13 2 CONTINUE U 14 0 15 D 16 1 CUTINE U 14 2 CONTINUE U 13 4 C D 16 5 UBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION UN U 17 4 C U 14 17 5 CALL ARKAY (2+MR+K+NRMAX+K+D+D) U 17 U 21 4 CALL ARKAY (2+MR+K+NRMAX+K+D+D) U 22 U 22		DIMENSION DENRMAXALIA MEXIALA (VELA, TROVELA, TROVELA, AND	υ	6	
MR1=MR+1 D 0 9 D0 2 I=1+MR D 10 0 10 D0 1 J=1+K D 10 11 0 12 D(I+J)=GRAD(I+J) D 12 0 13 1 CONTINUE D 14 0 14 2 CONTINUE D 14 0 15 0 SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION ON D 16 0 VICE VERSA+IT ENABLES VARIABLE DIMENSIONING OF DATA MATHICES Ling D 16 0 THE CALLING PROGRAM+REFERENCE NUMBER (2)+ D 20 19 0 C CALL ARHAY (2+MR+K+NRMAX+K+D+D) D 21 D 22		1)	5	1	
DO 2 I=1+MR DO 1 J=1+K D(1+J)=GRAD(I+J) CONTINUE CON		MR1=MR+1	0	0 0	
DO 1 J=1+K U 11 D(1+J)=GRAD(1+J) D 12 1 CONTINUE U 13 2 CONTINUE U 13 C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION UN U 14 C VICE VERSA+IT ENABLES VARIABLE DIMENSIONING UP DATA MATHICES Lin U 10 C THE CALLING PROGRAM+REFERENCE NUMBER (21+ C C U 21 C CALL ARKAY (2+MR+K+NRMAX+K+D+D) U 22		DO 2 I=1+MR	Ď	10	
D(1+J)=GRAD([+J) D 12 1 CONTINUE U 13 2 CONTINUE U 14 2 CONTINUE U 15 C D 16 D C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION ON U 17 C VICE VERSA+IT ENABLES VARIABLE DIMENSIONING OF DATA MATHICES Lind U 10 C THE CALLING PROGRAM+REFERENCE NUMBER (21+ U 19 C C U 21 C CALL ARHAY (2+MR+K+NRMAX+K+D+D) U 22		DO 1 J=1+K	ū	11	
1 CONTINUE 0 13 2 CONTINUE 0 14 0 15 0 15 0 10 15 0 16 0 14 0 15 0 16 0 14 0 15 0 16 0 14 0 17 0 17 0 VICE VERSA-IT ENABLES VARIABLE UIMENSIONING UP DATA MATHICES Lin 0 10 17 0 THE CALLING PROGRAM-REFERENCE NUMBER (21+ 0 19 0 20 0 C 0 21 0 21 0 22 0 CALL ARHAY (2+MR+K+NRMAX+K+D+D) 0 22 0 22	,		Ð	12	
C URROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION ON U 17 C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION ON U 17 C VICE VERSA-IT ENABLES VARIABLE DIMENSIONING OF DATA MATHICES LO U 17 C THE CALLING PROGRAM-REFERENCE NUMBER (2). C C U 21 C CALL ARRAY (2+MR+K+NRMAX+K+D+D) U 22	2		Ú	و لا	•
C D 15 D 16 D 16 D 16 D 16 D 16 D 16 D 16 D 17 C VICE VERSA.IT ENABLES VARIABLE DIMENSIONING OF DATA MATHICES Liv U 17 D 17 D 17 D 17 D 17 D 17 D 15 D 16 D 16 D 16 D 16 D 16 D 16 D 16 D 16	r		U U	14	
C SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION ON U U 17 C VICE VERSA.IT ENABLES VARIABLE DIMENSIONING OF DATA MATHICES Line U 10 C THE CALLING PROGRAM.REFERENCE NUMBER (21) U 19 C C U 21 C CALL ARRAY (2+MR+K+NRMAX+K+D+D) U 22	٢	· · · · · · · · · · · · · · · · · · ·	U n	15	
C VICE VERSA.IT ENABLES VARIABLE UIMENSIUNING UP DATA MATHICES L. U LU C THE CALLING PROGRAM.REFERENCE NUMBER (2). C C U U 20 C CALL ARKAY (2,MR,K,NRMAX,K,D,D) C C U 21 C C C C C C C C C C C C C C C C C C C	C	SUBROUTINE ARRAY CONVERTS DATA FROM SINGLE TO DOUBLE DIMENSION OF	4	17	
C THE CALLING PROGRAM-REFERENCE NUMBER (2). U 19 C U 20 C CALL ARKAY (2+MR+K+NRMAX+K+D+D) U 21 C D 22	C	VICE VENSA. IT ENABLES VARIABLE UIMENSIUNING OF DATA MATHICES LA	ũ	10	-
C CALL ARHAY (2+MR+K+NRMAX+K+D+D) D 21 CALL ARHAY (2+MR+K+NRMAX+K+D+D)	(THE CALLING PROGRAM.REFERENCE NUMBER (2).	Ũ	19	
CALL ARHAY (2+MR+K+NRMAX+K+D+D) D 21	č		U	٤U	
	`	CALL ARKAY (2.MP.K. MONAY F. D. D.	U	21	
			U	4 2	

HERE D IS CONVERTED FROM AN ARRAY OF SIZE (NRMAX+K) TO AN ARRAY OF C υ 23 C SIZE (MR.K) D 24 C υ 25 C Ð 26 SUBROUTINE NEGR DETERMINES RANK AND LINEARLY INDEPENDENT ROWS AND C <u>د</u> ا ν COLUMNS OF A GIVEN MATRIX D OF SIZE INK+K1-REFERENCE (4). C ω 20 DELTA IS A TEST VALUE FOR ZERO AFFECTED BY ROUNDUFF NUISE D 29 IRANK IS THE RESULTANT HANK OF D Ð ۵د TROW IS AN INTEGER VECTOR OF LENGTH MR CUNTAINING THE SUBSCRIPTS D 31 OF BASIC ROWS IN IROW(1/++++IROW(IRANK) υ 32 ICOL IS AN INTEGER VECTOR OF LENGTH & CONTAINING THE SUBSCRIPTS OF C 33 U. BASIC COLUMNS IN ICOLLEY UPTO ICOLLIRANNI C Ð 54 D 35 C 36 υ CALL MEGR ID. NR. K. DELTA, IRANK, IROW, ICULI D 37 D 38 CALL ARRAY (1+MR+K+NRMAX+K+D+D) 39 μ HERE D IS RECONVERTED FROM AN ARRAY OF SIZE (NR+K) TO AN ARRAY OF υ 40 SIZE (NRMAX .K) D 41 υ 42 IF (IRANK+NL+MR) GO TO 6 ь 43 DO 4 1=1+NR υ 44 DO 3 J=1+MR 45 LL I D(J+1)=GRAD(IROW(1)+ICOL(J)) D 46 CONTINUE 47 D CONTINUE D 48 D 49 CALL ARRAY 12+ MR+MR+NKMAX+K+U+U+ >0 υ HERE D IS CONVERTED FROM AN ARRAY OF SIZE INRMAXANT TO AN ARRAY OF υ 51 SIZE (MR+K) D 52 53 D D 54 SUBROUTINE MINV INVERTS SQUARE MATRIX D OF SIZE (MR+MR) AND STORES THE RESULT IN D+ DET IS THE DETERMINANT OF THE ORIGINAL MATRIX D+ 55 υ Ð 56 WHILE LE AND MM ARE WORK VECTORS OF SIZE MR. REFERENCE NUMBER (3) L 51 D 58 L, 59 CALL MINV (D.MR.DET.LL.MA) υ 60 D 61 CALL ARRAY, (1+MR+MR+NRMAX+K+D+D) 62 υ HERE D IS RECONVERTED FROM AN ARRAY OF SIZE (NR+K) TO AN ARRAY OF υ 63 SIZE (NRMAX+K) D 64 D 65 ISP=.T. υ 66 IF (ABS(DET/+LE+1+0E-10) GO' TO 5 υ 61 RETURN v 68 CONTINUE Ø 69 IF (IPRINT) WRITE (UNIT.16) MR Þ 70 RETURN υ 71 CONTINUE D 72 IRANKI=IRANK+1 Ð 13 IF (IRANKI.GE.MR) GU TU 7 Ð 74 ISP+.T. 75 Ð RETURN υ 76 CONTINUE υ 77 DO 9,1+1+1RANK1 D 78 DO 8 J=1+1RANK D 79

C

C

C

C

C

C

C

C

C

3

4

¢

C

٢

C

¢

C

C

¢

C

C

C

C

¢

٢

4

٨

7

JJ=1C0L(J) H(J+I)=GRAD(I+JJ) D 80 8 CONTINUE D 81 9 CONTINUE υ 62 DO 10 J=1+IRANK1 υ 83 H(IRANK1+J)=1+ υ 84 10 CONTINUE Ð 85 DG 11 J=1. IRANK D 86 0(1)=0. Ð 87 11 CONTINUE μ 88 Q(IRANK1)=1. υ 89 C . 0 90 C Ŀ 91 SUBROUTINE SOLVE SOLVES A SET OF LINEAR SIMULTANEOUS EQUATIONS. ¢ D - 92 H IS A MATRIX OF ROW SIZE TRANKI IN ALL ARRAY ROW DIMENSION KI. C U 93 A SQUARE SUBMATRIX OF H OF SIZE (IRANK1+IRANK1) IS PART OF THE 94 C μ THE MATRIX EQUATION ON THE LEFTHAND SIDE ANTILE & IS INITIALLY THE VECTOR ON THE RIGHTHAND SIDE & IS FINALLY THE SOLUTION VECTOR. C D 95 L. ¢ 76 IDET IS DEFINED BY 20010 .LT. AUS(DET) .LT. 200(10+1) WHERE DET IS C U 97 THE DÉTERMINANT OF THE SUBMATRIX. D 98 C D 99 с. REFFRENCE NUMBER (7) A LISTING OF THIS SUBROUTINE IS ATTACHED TO THE PACKAGE C D 100 0 101 C U 102 ٢ CALL SOLVE (H.Q.IDET.IRANKI.KI) D 103 D 104 DU 12 J=1+IRANK1 U 105 X2131=Q[J] D 106 12 CONTINUE 00 14 J=1+K U 107 D 108 JJ=ICOL(J) D'109 Q(J)=0. D 110 DO 13 1-1.MR 01J)=Q(J)+X2([)+GRAD([+JJ) D 111 D 112 R2(JJ)+Q(J) 0.113 13 CONTINUE D 114 14 CONTINUE D 115 X2SUN=Ga DO 15 I=1.MR D 116 ٩. U 117 X25IM#X25UN+X2(1) 15 CONTINUE D 118 D 119^b RETURN D 120 c 0 121 FURMAT (10X+12+98X++USER IS ADVISED TU+/110X++INCREASE VALUE OF DE 16 U 122 ILTA#//E 0 123 END D 124 C 0 125 r D 126 C 0 127-C *********** Ł 1 FUNCTION ANORM1 (K+b1) THE EUCLIDEAN NORM OF VECTOR BI IS CALCULATED HERE Ł ¢ 2 Ε 3 DIMENSION BILLI Ł 4 ANORM LEC. 5 Ε . ' 00 1 1+1+K Ε ANORMI-ANORMI+H1(1)+B1(1) 6 ε 7 CONTINUE 8 È ANORM1=SURT(ANORM1) Ł y

. :

RETURN E, 10 END E 11 C £ 12 ٢ Ē 13 c c 14-F T FUNCTION ANORH2 (K+81+ F 2 c MAX(ABS(01(1))+ABS(01(2))++++ABS(01(K))) IS CALCULATED HERE F з DIMENSION BIELD F 4 ANORH2=ABS(B1(1)) F 5 IF (K.LT.2) GO TO 2 F 6 7 DO 1 1-2+K F ABSB1=ABS(B1(I)) F 8 IF (ABSB1.GT.ANORH2) ANORH2=ABSB1 F 9 1 CONTINUE F 10 RETURN 2 F 11 END F 12 C F 13 FR C 14 Ċ F 15-6 1 SUBROUTINE SOLVE (A+X+ID+N+NA) G 2 DIMENSION A(NA.1). X111 G 3 D=0. G 4 DATA DIV/-693147181/ 5 G 00 6 I=1+N 6 6 AA=0. G 1 DO 1 J=1+N G 8 AB+ABSTALJ+III G 9 IF TAB-LE-AAT GO TO 1 G 10 K=J G 11 AA=AB G 12 CONTINUE 1 G 13 D=D+ALOGIAAI G 14 IF LI-EQ-NJ GO TO 7 IF (K-EQ-14 GO TO 3 15 G G 79 00 2 J=1+N G 17 AB-AII+JI G 18 A11+J}=A(K+J) ----G 19. A(K+J)=Ab ú 20 CONTINUE 2 G 21 A8=X(1) G 22 X(1)=X(K) 6 23 X(K)=AB G 24 25 11=1+1 . G DO 5 J=11+N G 26 AA=-A[J+1}/A(1+1) G 27 A(J+1)=0. í, 28 00 4 K-11.N G 29 ALJ.KI-ALJ.KI+AA+ALI.KI G 30 CONTINUE 4 6 31 (1)X#AA+(L)X=(L)X G 32 CONTINUE 4 G 33 CONTINUE Ġ 6 34 7 ID=D/DIV ما 35 XIN7=XIN1/AIN+N1 G 36 DU 9 11+2+N 37 L.

٥.

Æ.

162

无了

ا سر

I=N+1-11		6	Зн
<u>[]=[+1</u>		č	10
AA=0.	· · ·	· č	5,7 6,0
DO 8 J#11+N		Ğ	40
LL, LJA+AA=AA	(L)X+	Ğ	42
CONTINUE		ŭ	
X([)=(X([)-W	\A}/A[[+]}	Ğ	44
CONTINUE		6	45
RETURN		Ğ	46
FND		č	A7
		Ğ	48
		6	40
		Ğ	50-
REFERENCES	(1) J+W+BANDLER, CONDITIONS FOR A MINIMAX OPTIMUM',	ĥ	
	IEEE TRANS CIRCUIT THEORY ICORRESP. F. VUL. CT-18.	h	ž
	PP 476-4/9+JULY 1971		3
	(2) SUBROUTINE ARRAY + PYB + SYSTEM/360 SCIENTIFIC	h	. 4
	SUBROUTINE PACKAGE.VERSION 3. IBN.PROGRAM NUMBER	n	5
	36UA-CM-U3X	н	6
	(3) SUBROUTINE MINV PI18 SYSTEM/360 SCIENTIFIC	н	7
	SUBROUTINE PACKAGE VERSION 3.10M.PRUGHAM NUMBER	н	6
	360A-CM-03X	н	У
	(4) SUBROUTINE MEGR. P127. SYSTEM/360 SCIENTIFIC	n	10
	SUBROUTINE PACKAGE.VERSION 3, IDM, PROGRAM NUMBER	н	11
	360A-CM-03X	н	12
	(5) SUBROUTINE SIMPLE, DATA PROCESSING AND COMPUTING	н	15
	CENTRE+LIBRARY INFURMATION SHEET MILLS 15+02+01+	н	14
	P130+MCHASTER UNIVERSITY+HAMILTUN+ONTARIO+CANAUA	н	15
	(6) J.W.BANULER AND T.V.SKINIVASAN, THE URAZOR SEARCH	n	16
	PROGRAM FOR MINIMAX UBJECTIVES + LEE TRANS	н	17
	MICROWAVE THEORY TECH(PROGRAM DESCRIPTION).	n	10
	VOL. MTT-20, PP 784-785, NOVEMBER 1972	н	19
	171SUBROUTINE SULVE, DATA PROCESSING AND COMPUTING	н	∠0
	CENTREOLIBRARY INFURMATION SHEET MILIS 45.04.16.	н	21
	P 27+MCHASTER UNIVERSITY HAMILTUN UNTAHIO CANADA	**	22-

٦.

Ł

8 9

с С



REFERENCES

- J. H. Anderson (1967), "Geometrical approach to reduction of dynamic systems", Proc. IEE, vol. 114, pp.1014-1018.
- J. W. Bandler (1969m), "Optimization methods for computer-aided design", <u>IEEE Trans. Microwave Theory Tech.</u>, vol. MTT-17, pp.533-552.
- J. W. Bandler (1969b), "Computer optimization of inhomogeneous waveguide transformers", <u>IEEE Trans. Microwave Theory Tech.</u>, vol. MTT-17, pp.563-571.
- J. W. Bandler (1971), "Conditions for a minimax optimum", <u>IEEE Trans</u>. <u>Circuit Theory</u>, vol. CT-18, pp.476-479.
- J. W. Bandler (1972), "Optimization methods and microwave circuit design", Proc. 6th Princeton Conf. Information Sciences and Systems (Princeton, N.J.), pp.10-14.
- J. W. Bandler and B. L. Bardakjian (1973), "Least pth optimization of recursive digital filters", <u>IEEE Int. Symp. Circuit Theory</u> (Toronto, Canada), pp.377-380.
- J. W. Bandler and C. Charalambous (1971), "On conditions for optimality in least pth approximation with p+=", Proc. 9th Allerton Conf. <u>Circuit and System Theory</u> (Urbana, Ill.), pp.404-413.
- J. W. Bandler and C. Charalambous (1972a), "A new approach to nonlinear programming", <u>Proc. 5th Hawaii Int. Conf. System Sciences</u> (Honolulu, Hawaii), pp.127-129.
- J. W. Bandler and C. Charalambous (1972b), "Conditions for minimax approximation obtained from the 1 norm", <u>IEEE Trans. Automatic</u> <u>Control,vol. AC-17, pp.257-258.</u>

- J. W. Bandler and C. Charalambous (1972c), "Theory of generalized least pth approximation", <u>IEEE Trans. Circuit Theory</u>, vol. CT-19, pp.287-289.
- J. W. Bandler and C. Charalambous (1972d), "Practical least pth optimization of networks", <u>IEEE Trans. Microwave Theory Tech.</u>, vol. MTT-20, pp.834-840.
- J. W. Bandler and C. Charalambous (1973a), "On conditions for optimality in least pth approximation with p+∞", <u>J. Optimization Theory and</u> Applications, vol. 11.
 - J. W. Bandler and C. Charalambous (1973b), "Nonlinear programming using minimax techniques", <u>J. Optimization Theory and Applications</u>, accepted for publication.
 - J. W. Bandler, C. Charalambous and S. K. Tam (1972), "Computer-aided equal ripple design of lumped-distributed-active filters", Int. Filter Symp. (Santa Monica, Calif.), pp.79-80.
 - J. W. Bandler and V. K. Jha (1972), "Network optimization computer program package", CRL Internal Report Series No. CRL-5, Communications Research Laboratory, Department of Electrical Engineering, McMaster Univ., Hamilton, Canada.
 - J. W. Bandler and A. G. Lee-Chan (1971), "Gradient razor search method for optimization ", <u>IEEE Int. Nicrowave Symp. Dig.</u> (Washington, D.C.), pp.118-119.
 - J. W. Bandler and P. A. Macdonald (1969a), "Cascaded noncommensurate transmission-line networks as optimization problems", <u>IEEE</u> <u>Trans. Circuit Theory</u>, vol. CT-16, pp.391-394.

- J. W. Bandler and P. A. Macdonald (1969b), "Optimization of microwave networks by razor search", <u>IEEE Trans. Microwave Theory Tech.</u>, vol. MTT-17, pp.522-562.
- J. W. Bandler and P. A. Macdonald (1971), "The razor search program", <u>IEEE Trans. Microwave Theory Tech.</u> (Comput. Program Descr.), vol. MTT-19, p.667.
- J. W. Bandler, N. D. Markettos and N. K. Sinha (1973), "Optimum system modelling using recent gradient methods", <u>Int. J. Systems Science</u>, vol. 4, pp.33-44.
- J. W. Bandler, N. D. Markettos and T. V. Srinivasan (1972), "A comparison of recent minimax techniques on optimum system modelling", <u>Proc. 6th Princeton Conf.</u> Information Sciences and Systems (Princeton, N.J.), pp.540-544.
- J. W. Bandler, N. D. Markettos and T. V. Srinivasan (1973), "Gradient minimax techniques for system modelling", <u>Int. J. Systems Science</u>, vol. 4, pp.317-331.
- J. W. Bandler and R. E. Seviora (1970), "Current trends in network optimization", <u>IEEE Trans. Microwave Theory Tech.</u>, vol. MTT-18, pp.1159-1170.
- J. W. Bandler and T. V. Srinivasan (1971), "A new gradient algorithm for minimax optimization of networks and systems", <u>Proc. 14th</u> <u>Midwest Symp. Circuit Theory</u> (Denver, Colo.), pp.16.5.1-16.5.11.
- J. W. Bandler and T. V. Srinivasan (1972), "The grazor search program for minimax objectives", <u>IEEE Trans. Microwave Theory Tech.</u>, vol. MTT-20, pp.784-785.

- J. W. Bandler and T. V. Srinivasan (1973a), "Constrained minimax optimization by grazor search", <u>Proc. 6th Hawaii Int. Conf.</u> <u>System Sciences</u> (Honolulu, Hawaii), pp.125-128.
- J. W. Bandler and T. V. Sriniva'san (1973b), "On realistic minimax modelling of high-order systems", <u>Proc. 7th Princeton Conf.</u> <u>Information</u> <u>Sciences and Systems</u> (Princeton, N.J.).
- J. W. Bandler and T. V. Srinivasan (1973c), "Practical investigation of conditions for minimax optimality", Proc. 16th Midwest Symp. <u>Circuit Theory</u> (Waterloo, Canada), pp.XX.4.1-XX.4.10.
- J. W. Bandler and T. V. Srinivasan (1973d), "Program for investigating minimax optimality conditions", <u>IEEE Trans. Microwave Theory</u> <u>Tech.</u>, vol. MTT-21, p.433.
- J. W. Bandler and T. V. Srinivasan (1973e), "Computer-aided system modelling and design", <u>Proc. Int. Conf.</u> Systems and Control (Coimbatore, India).
- J. W. Bandler, T. V. Srinivasan and C. Charalambous (1972), "Minimax optimization of networks by grazor search", <u>IEEE Trans. Micro-</u> <u>wave Theory Tech.</u>, vol. MIT-20, pp.596-604.
- G. T. Bereznai (1971), "Adaptive nuclear control based on optimal loworder linear models", Ph.D. Dissertation, McMaster Univ., Hamilton, Canada.
- C. Brancher, F. Maffioli and A. Premoli (1970), "Computer optimisation of cascaded noncommensurable-line lowpass filters", <u>Electronics</u> <u>Letters</u>, vol. 6, pp.513-515.
- H. J. Carlin (1971), "Distributed circuit design with transmission line elements", Proc. IEEE, vol. 59, pp.1059-1081.



- H. J. Carlin and O. P. Gupta (1969), "Computer design of filters with lumped-distributed elements or frequency variable terminations", <u>IEEE Trans. Microwave Theory Tech.</u>, vol. MTT-17, pp.598-604.
- C. W. Carroll (1961), "The created response surface technique for optimizing nonlinear, restrained systems", <u>Operations Research</u>, vol. 9, pp.169-185.
- C. Charalambous (1973), "Nonlinear least pth approximation with applications", Ph.D. Dissertation, McMaster Univ., Hamilton, Canada.
- C. F. Chen and L. S. Shieh (1968), "A novel approach to linear model simplifications", Proc. JACC (Ann Arbor, Mich.), pp.454-460.
- M. R. Chidambara (1969), "Two simple techniques for the simplification of large dynamic systems", <u>Proc. JACC</u>, pp.669-674.
- A. R. Curtis and M. J. D. Powell (1966), "Necessary conditions for a minimax approximation", <u>Computer J.</u>, vol. 8, pp.358-361.
- E. J. Davison (1966), "A method for simplifying linear dynamic systems", <u>IEEE Trans. Automatic Control</u>, vol. AC-11, pp.93-101.
- V. F. Dem'yanov (1970), "Sufficient conditions for minimax problems", <u>Zh. Uychisl. Mat. Fiz.</u>, vol. 10, pp.1107-1115.
- S. W. Director and R. A. Rohrer (1969), "The generalized adjoint network and network sensitivities", <u>IEEE Trans. Circuit Theory</u>, vol. CT-10, pp.318-323.
- R. C. Dorf (1967), Modern Control Systems. Reading, Mass. : Addison-Wesley, pp.167-168.
- A. V. Fiacco and G. P. McCormick (1964a), "The sequential unconstrained minimization technique for nonlinear programming, a primal-dual method", Management Science, vol. 10, pp.360-366.

 A. V. Fiacco and G. P. McCormick (1964b), "Computational algorithm for the sequential unconstrained minimization technique for nonlinear programming", <u>Management Science</u>, vol. 10, pp.601-617.
 R. Fletcher (1970), "A new approach to variable metric algorithms",

Computer J., vol. 13, pp.317-322.

R. Fletcher and M. J. D. Powell (1963), "A rapidly convergent descent method for minimization", Computer J., vol. 6, pp.163-168.

- W. T. Hatley, Jr. (1967), "Computer synthesis of wide-band impedancematching networks", Tech. Rept. 6657-2, Stanford Electronics Laboratories, Stanford Univ., Stanford, Calif.
- J. E. Heller (1969), "A gradient algorithm for minimax design",

Report R-406, Coordinated Science Lab., Univ. of Illinois,

Urbana.

- R. Hooke and T. A. Jeeves (1961), " 'Direct search' solution of numerical and statistical problems", J. Ass. Comput. Mach., vol. 8, pp.212-229.
- Y. Ishizaki and H. Watanabe (1968), "An iterative Chebyshev approximation method for network design", IEEE Trans. Circuit Theory, vol. CF-15, pp. 326-336
 - D. H. Jacobson and W. Oksman (1972), "An algorithm that minimizes homogeneous functions of N variables in N+2 iterations and rapidly minimizes general functions", <u>J. Math. Anal. Applics.</u>, vol. 38, pp.535-552.
 - P. Kokotović and P. Sannuti (1968), "Single perturbation method for reducing the model order in optimal design", <u>Proc. JACC</u> (Ann Arbor, *... Mich.), pp.468-476.
H. W. Kuhn and A. W. Tucker (1951), "Nonlinear programming",

Proc. 2nd Berkeley Symp. Mathematical Statistics and Probability (Berkeley, Calif.), pp.481-492.

- L. S. Lasdon (1970), Optimization Theory for Large Systems. New York : Macmillan.
- R. Levy (1965), "Tables of element values for the distributed lowpass prototype filter", <u>IEEE Trans. Microwave Theory Tech.</u>, vol. MTT-13, pp.514-536.
- N. D. Markettos (1972), "Optimum system modelling using recent gradient methods", M. Eng. Dissertation, McMaster Univ., Hamilton, Canada.
- S. A. Marshall (1966), "An approximate method for reducing the order of linear systems", Control, vol. 10, pp.642-643.
- G. L. Matthaei, L. Young and E. M. T. Jones (1964), <u>Microwave Filters</u>, <u>Impedance Matching Networks and Coupling Structures</u>. New York : <u>McGraw-Hill</u>.
- J. Medanic (1970), "Solution of the convex minimax problem by the Newton-Raphson method", Proc. 8th Allerton Conf. Circuit and System Theory (Urbana, Ill.), pp.13-22.
- D. Mitra (1969), "The reduction of complexity of linear time-invariant dynamic systems", Proc. 4th IFAC Congress (Warsaw, Poland), pp.19-33.
- M. R. Osborne and G. A. Watson (1969), "An algorithm for minimax approximation in the non-linear case", <u>Computer J.</u>, vol.12, pp.63-68.

170

- J. R. Popović (1972), "General programs for least pth and near minimax approximation", M. Eng. Dissertation, McMaster Univ., Hamilton, Canada.
- R.E.Seviora, M.Sablatash and J. W. Bandler (1970), "Least pth and minimax objectives for automated network design", <u>Electronics Letters</u>, vol. 6, pp.14-15.
- N. K. Sinha and G. T. Bereznai (1971), "Optimum approximation of highorder systems by low-order models", <u>Int. J. Control</u>, vol. 14, pp.951-959.
- N. K. Sinha and W. Pille (1971), "A new method for reduction of dynamic . systems", <u>Int. J. Control</u>, vol. 14, pp.111-118.

Subroutine ARRAY, System/360 Scientific Subroutine Package, Version III, IBM Programmer's Manual Number 360A-CM-03x, p.98.

Subroutine MFGR, System/360 Scientific Subroutine Package, Version III, IBM Programmer's Manual Number 360A-CM-03x, p.127.

Subroutine MINV, System/360 Scientific Subroutine Package, Version III,

IBM Programmer's Manual Number 360A-CM-03x, p.118.

Subroutine SIMPLE, Data Processing and Computing Centre, Library Information Sheet MILIS 15.02.01, McMaster Univ., Hamilton, Canada,

p.130.

 $\sum_{i=1}^{n}$

Subroutine SOLVE, Data Processing and Computing Centre, Library Information Sheet MILIS 45.04.16, McMaster Univ., Hamilton, Canada,

1-

p.27.

Subroutine TGSORT, Data Processing and Computing Centre, Library Information Sheet MILIS 5.3.34, McMaster Univ., Hamilton, Canada.

- G. C. Temes (1969), "Optimization methods in circuit design", in <u>Computer Oriented Circuit Design</u>, F. F. Kuo and W. G. Magnuson Jr., Eds. Englewood Cliffs, N.J. : Prentice Hall.
- G. C. Temes and D. Y. F. Zai (1969), "Least pth approximation", IEEE Trans. Circuit Theory, vol. CT-16, pp. 335-237.
- A. D. Waren, L. S. Lasdon and D. F. Suchman (1967), "Optimization in engineering design", Proc. IEEE, vol. 55, pp.1885-1897.
- G. A. Watson (1970), "On an algorithm for nonlinear minimax approximation", Commun. Ass. Comput. Mach., vol. 13, pp.160-162.
- W. I. Zangwill (1969), Nonlinear Programming : A Unified Approach. Englewood Cliffs, N.J., : Prentice Hall.



	AUTHOR INDEX
J. H. Anderson	77,164
J. W. Bandler	2,3,8,13,14,15,17,18,19,21,28,38,39,40,46,48, 63,65,78,80,81,102,164-167,171
B. L. Bardakjian	14,164
G. T. Bereznai	42,77,107,167,171
C. Brancher	66,69,71,167
H. J. Carlin	63,65,67,73,167,168
C. W. Carroll	10,168
C. Charalambous	2,14,17,18,38,46,80,164,165,167,168
C. F. Chen	77,168
M. R. Chidambara	77,168
A. R. Curtis	39,168
E. J. Davison	77,168
V. F. Dem'yanov	38,168
S. W. Director	3,8,17,168
R. C. Dorf	80,168
A. V. Fiacco	8,9,11,168,169
R. Fletcher	14,15,80,82,85,86,89,90,93,94,101,102,117,169
0. P. Gupta	63,65,67,168
W. T. Hatley, Jr.	46,169
J. E. Heller	8,169
R. Hooke	9,169
Y. Ishizaki	8,11,169
D. H. Jacobson	14,80,82,85,89,93,101,169

173

-

- *-* ·

T. A. Jeeves	9,169
V. K. Jha	14,165
E. M. T. Jones	28,48,170
P. Kokotović	77,169
H. W. Kuhn	34,38,170
L. S. Lasdon	9,21,34,170,172
A. G. Lee-Chan	8,13,19,165
R. Levy	69,73,170
P. A. Macdonald	2,8,17,28,48,165,166
F. Maffioli	66,69,71,167
N. D. Markettos	3,15,77,78,80,81,166,170
S. A. Marshall	77,170
G. L. Matthaei	28,48,170
G. P. McCormick	8,9,11,168,169
J. Medanic	38,170
D. Mitra	77,170
W. Oksman	14,80,82,85,89,93,101,169
M. R. Osborne	2,8,11,13,46,48,49,52-59,62,76,170
W. Pille	77,171
J. R. Popović	14,171
M. J. D. Powell	14,39,80,85,89,90,101,102,168,169
A. Premoli	66,69,71,167
R. A. Rohrer	3,8,17,168
M. Sablatash	14,171
P. Sannuti	77,169
R. E. Seviora	3,8,14,17,63,65,166,171
•	٩

. . .

ø

L. S. Shieh	77,168
N. K. Sinha	77,78,80,166,171
T. V. Srinivasan	2,3,15,17,18,19,40,46,81,102,166,167
D. F. Suchman	9,172
S. K. Tam	14,165
G. C. Temes	14,27,172
A. W. Tucker	34,38,170
A. D. Waren	9,172
H. Watanabe	8,11,169
G. A. Natson	2,8,11,13,46,48,49,52-59,62,76,170,172
L. Young	28,48,170
D. Y. F. Zai	14,172
W. I. Zangwill	34,172



175

J