General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

NASA Contractor Report 177968

ALGEBRAIC GRID GENERATION USING TENSOR PRODUCT B-SPLINES

ALGEBRAIC GRID GENERATION USING TENSOR N86-11908 PLODUCT E-SELINES Ph.D. Thesis (Oklahoma Old Dominica Univ.) Univ.) 152 p HC A08/MF A01 CSCL 12A

> Unclas G3/64 27608



Bonita V. Saunders

-

.

.

OLD DOMINION UNIVERSITY Norfolk, Virginia

Grant NGT 47-003-802 September 1985



Space Administration
Langley Research Center

Hampton, Virginia 23665

ABSTRACT

ALGEBRAIC GRID GENERATION USING TENSOR PRODUCT B-SPLINES

Bonita Valerie Saunders Old Dominion University, 1985 Director: Dr. Philip W. Smith

In general, finite difference methods are more successful if the accompanying grid has lines which are smooth and nearly orthogonal. This thesis discusses the development of an algorithm which produces such a grid when given the boundary description.

Topological considerations in structuring the grid generation mapping are discussed. In particular, this thesis examines the concept of the degree of a mapping and how it can be used to determine what requirements are necessary if a mapping is to produce a suitable grid.

The grid generation algorithm uses a mapping composed of bicubic B-splines. Boundary coefficients are chosen so that the splines produce Schoenberg's variation diminishing spline approximation to the boundary. Interior coefficients are initially chosen to give a variation diminishing approximation to the transfinite bilinear interpolant of the function mapping the boundary of the unit square onto

i

the boundary of the grid.

The practicality of optimizing the grid by minimizing a functional involving the Jacobian of the grid generation mapping at each interior grid point and the dot product of vectors tangent to the grid lines is investigated.

Grids generated by using the algorithm are presented.

ACKNOWLEDGEMENTS

I would like to thank my dissertation director, Dr. Philip W. Smith, for his encouragement, patience and assistance during all stages of the development of this dissertation. I also thank Dr. Robert Smith of NASA Langley Research Center for his guidance and instruction during my past three years at NASA. Dr. Smith offered invaluable suggestions and advice to me during my research. I would also like to thank others who helped in the reviewing of this thesis. The comments and suggestions of Dr. John Tweed, Dr. James Schwing and Dr. John Swetits are greatly appreciated.

I must also express my gratitude to the entire Computer Science and Applications Branch in the Analysis and Computation Division at Langley Research Center. Everyone always willingly answered my questions or found others who could assist me.

Finally, my sincere thanks to Mrs. Harriet Sims whose patience and perseverance made the typing of this dissertation possible.

iii

TABLE OF CONTENTS

										Page
LIS	T OF	IGURES	• • • •	•••	•••		•		•	i v
Cha	pter									
1.	INTR	DUCTION		•••	••		•	• •	•	1
2.	APPL	CATIONS OF DE	GREE THE	DRY	•••		•	• •	•	6
	2.1	Defining the	Degree of	fa Ma	opping]	•	• •	•	7
	2.2	Properties of	the Degi	ree	•••	• •	•	• •	•	8
	2.3	A Topological	Definit	ion of	the	Deg	ree	• •	•	11
	2.4	Applications	to Grid (Genera	ition		•	• •	•	19
	2.5	Additional Top	pological	l Ques	tions	5	•	• •	•	22
3.	AN A	GEBRAIC GRID	GENERATI	ON MAP	PING		•		•	27
	3.1	A First Order	Example				•		•	27
	3.2	B-splines								30
		3.2.1 Defini	ng B-spli	ines.	•••		•	• •	•	30
		3.2.2 Proper	ties of I	3-spli	nes.	•••	•		•	31
		3.2.3 Spline	Function	ns			•	• •	•	34
		3.2.4 Variat:	ion Dimir	nishin	ıg Spl	ine	s.	• •	•	36
		3.2.5 Tensor	Product	B-spl	ines		•		•	38
	3.3	A Smoothing F	unctional	l			•		•	40
		3.3.1 Charac	teristic	s of t	he Fu	inct	ion	al.	•	42
		3.3.2 Converg	gence of	the S	mooth	ning				
		Fund	ctional.	•••	• •	••	•	•••	•	46

۷

PRECEDING PAGE BLANK NOT FILMED

CARRIER CONTRACTOR STATE STATE

>

.

PAGE_____INTERTIONALLY WARDS

																					Page
4.	PROGR	RAM 1	FENT	EST	•••	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	50
	4.1	The	Alg	ori	thm	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	50
	4.2	Comp	outi	ng	the	T	ens	501	r	Pr	odı	ıct	B	3 - 2	ip l	ir	ies	5.	•	•	51
	4.3	Choc	osin	g t	he	In	iti	ia	1	Co	eft	ic	i e	nt	s	•	•	•	•	•	53
	4.4	Mini	imiz	ing	th	е	Smo	0.0	th	in	g F	un	ct	ic	na	1	•	•	•	•	56
	4.5	Dist	trib	uti	on	Fu	nci	ti	o n	s.	•	•	•	•	•	•	•	•	•	•	60
5.	RESUL	.ts #	AND	DIS	CUS	SI	ON	•	•	•	•	•	•	•	•	•	•	•	•	•	66
	5.1	Conv	/ex	Dom	ain	S .	•	•	•	•	•	•	•	•	•	•	•	•	•	•	68
		5.1.	. 1	Tra	pez	o i (d.	•	•	•	•	•	•	•	•	•	•	•	•	•	68
		5.1.	. 2	Qua	dri	la	ter	^ a	1	wi'	th	Un	eq	ua	1	Si	de	es	•	•	70
		5.1.	. 3	Tri	ang	le	•	•	•	•	•	•	•	•	•	•	•	•	•	•	74
		5.1.	, 4	Cir	cle	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	74
	5.2	Nond	conv	еx	Dom	ai	n s	•	•	•	•	•	•	•	•	•	•	•	•	•	77
		5.2.	. 1	Non	con	ve	х (⊋u	ad	ri	lat	er	al	•	•	•	•	•	•	•	77
		5.2.	. 2	Puz	zle	Ρ	ied	:e	s.	•	•	•	•	•	•	•	•	•	•	•	81
	5.3	Gric	ls f	or	Spe	сi	fid	:	ΟЪ,	je	cts	i .	•	•	•	•	•	•	•	•	85
		5.3.	. 1	Air	foi	1.	•	•	•	•	•	•	•	•	•	•	•	•	•	•	91
		5.3.	2	Spi	ke-	No	sec	1	Bo	dy	•	•	•	•	•	•	•	•	•	٠	95
		5.3.	, 3	Sħu	ttl	e.	٠	•	•	•	•	•	•	•	•	•	•	•	•	•	95
6.	CONCL	USIC	DNS.	•	•••	•	•	•	•		•	•	•	•	•	•	•	•	•	•	102
APPE	ENDIX		• •	•		•	•	•	•		•	•			•				•		106

ALL THE MAKE

CTURNER ADDRESS STATES

vi

LIST OF FIGURES

291 NR/122

>

161

FIGURE		Page
1.	Invariance of the degree when points connected by a continuous path avoiding F(aC)	10
2.	Mapping the unit square onto a nonconvex domain	24
3.	Initial grid on nonconvex domain	25
4.	Optimized grid on nonconvex domain	26
5.	Support of tensor product B-spline B_{ij}	4 1
6.	Mapping from computational domain to physical domain	55
7.	Trapezoid Grid	57
8.	Support for tensor product B-spline $B_{6,5}$	59
9.	Obtaining a concentration of grid points	62
10.	Concentrating grid points on trapezoid domain	64
11.	Grids on trapezoid domain	69
12.	Effect of weights, wj and wd	71
13.	Grids on quadrilateral with unequal sides	72
14.	Concentrating grid points on quadrilateral	73
15.	Division of triangular boundary into four sections	75
16.	Grids on triangular domain	76
17.	Grids on circular domain	78
18.	Grids on circular domain after optimization .	79

vii

FIGURE		Page
19.	Nonconvex quadrilateral	. 80
20.	Grids on nonconvex quadrilateral domain	. 82
21.	Optimized grids on nonconvex quadrilateral domain	. 83
22.	Enlarged corner of optimized grid	. 84
23.	Puzzle shaped domains	. 86
24.	Grids on first puzzle shaped domain	. 87
25.	Grids obtained after various iterations	. 88
26.	Grids on second puzzle shaped domain	. 89
27.	Exponential distributions on puzzle shaped grids	. 90
28.	Domain around Kárman - Trefftz airfoil	. 92
29.	Grids for Karman - Trefftz airfoil	. 93
30.	Optimized grids for Karman - Trefftz airfoil	. 94
31.	Grids for spike-nosea body	. 96
32.	Optimized grids for spike-nosed body	. 97
33.	Original grid for shuttle	. 99
34.	Optimized grid for shuttle	. 100
35.	Optimized grid concen†rated near shuttle boundary	. 101

viii

•

1. INTRODUCTION

Grid generation is the numerical development of curvilinear coordinate systems. In recent years grid generation has been the key to solving partial differential equations on arbitrarily shaped regions by finite difference methods. Although much of the motivation for grid generation has come from fluid dynamics, the techniques apply to any area, such as electromagnetics and heat transfer, which involves the solving of partial differential equations on a physical domain.

Inherent in grid generation techniques is a mapping T from some canonical domain such as a square or rectangle in two dimensions, or cube in three dimensions, onto the physical domain on which the partial differential equations are to be solved. The image of a mesh on the canonical, or computational, domain will be a grid on the physical domain. When the grid boundary coincides with the boundary of the physical domain, the system generated is called a boundary fitted coordinate system.

A boundary fitted coordinate system allows one to apply boundary conditions exactly, thus avoiding interpolation errors. However, such a system may make the equations to be solved more complex [Sm].

The distribution of the coordinate lines, or grid

lines, should be smooth, but concentrated in areas where a large gradient occurs in the physical solution. As stated by Thompson, Warsi and Mastin [TWM], "the grid points may be thought of as a finite set of observers of the physical solution, stationed to be most effective in covering all of the action on the field." Ideally, the grid should be adaptive, that is, coupled with the physical solution so that it automatically redistributes its grid lines to obtain the desired regions of concentration as the solution evolves. However, the interior lines should not cross the physical boundary and should be nearly orthogonal at the intersection points to avoid large truncation errors in the finite difference approximations.

Grid generation is based on the observation that finite difference computations are much easier to make on a uniform mesh over a canonical domain such as a square or cube than on a grid over an irregularly shaped region. Therefore, the partial differential equations to be solved must first be transformed so that the computational coordinates become the independent coordinates. The resulting equations may then be expressed as finite difference equations on the computational domain.

Grid generation techniques may be divided into two general types: partial differential equation methods and algebraic methods. P.d.e. methods include elliptic, hyperbolic and conformal mapping techniques. All of these methods involve the solving of partial differential equations to

2

CHICKEN STREET, STREET

A State of the State of the second

obtain the grid coordinates. The simplest elliptic method for grid generation uses the Laplace equations

 $\Delta^{a} \xi = \frac{\partial^{a} \xi}{\partial x^{2}} + \frac{\partial^{a} \xi}{\partial y^{2}} = 0$ $\Delta^{a} \eta = \frac{\partial^{a} \eta}{\partial x^{2}} + \frac{\partial^{a} \eta}{\partial y^{2}} = 0$

where ξ and η are the computational coordinates and x and y are the physical coordinates in two dimensions. The equations are first transformed so that the independent and dependent variables are interchanged. Then the new equations are solved for x and y in terms of ξ and η . Some control over the grid cell spacing can be accomplished by introducing control functions $P(\xi,\eta)$, $Q(\xi,\eta)$ and solving the Poisson equations [TWM, p. 39]

 $\Delta^{a} \xi = P(\xi, \eta)$ $\Delta^{a} \eta = Q(\xi, \eta).$ Solving the Laplace equations

 $\Delta^{a} \xi = 0$ $\Delta^{a} \eta = 0$

with boundary conditions

produces a conformal transformation [TWM, p. 11]

Starius [St, p. 27] shows that solving an initial value problem satisfying

 $x_{\eta} = -y_{\xi}F$ $y_{\eta} = x_{\xi}F$ where F is chosen so that the system is hyperbolic produces a hyperbolic grid generating system. Grids generated from elliptic equations are generally smooth regardless of the type of boundary, but slope discontinuities propagate through hyperbolically generated grids [St]. Generating a grid using conformal mapping techniques requires careful selection of the boundary data, making it difficult to structure the grid to obtain a high concentration of grid points in areas of large gradients in the physical solution. More grid points may have to be added in order to capture regions of rapid change such as shocks and boundary layers. Also in p.d.e. generated systems the Jacobian information needed for the transformation of the equations being solved must be computed numerically.

In algebraic methods an explicit functional relationship between the computational and physical domains is defined. Therefore, no p.d.e. need be solved to obtain the grid coordinates and the Jacobian matrix can be computed analytically. Such methods allow more precise controls of the grid structure making it easier to concentrate grid points in large gradient areas. However, algebraically generated grids are more sensitive to point distributions on the boundary and, in general, may not be as smooth as those generated by elliptic techniques [Sm]. Slope discontinuities on the boundary may propagate into the field. Nevertheless, a variety of techniques have been used to produce acceptable smoothness in algebraically generated grids.

This thesis discusses an algebraic grid generation technique for creating boundary fitted coordinate systems. This technique uses a mapping which is a sum of tensor product B-splines. Chapter 2 discusses degree theory, explaining how the degree of a mapping can be used to determine what conditions must be met if an algebraic transformation is to produce a suitable grid. Chapter 3 presents the tensor product grid generation mapping and discusses the properties of B-splines to show their suitability for use in such a mapping. Chapter 3 also introduces a functional which can be used to change the coefficients in the mapping in order to enhance the smoothness and orthogonality in the generated grid.

Chapter 4 discusses the computer program TENTEST which uses the techniques presented in Chapter 3 to generate grids on arbitrarily shaped two-dimensional domains. Some of the grids created using TENTEST are illustrated and discussed in Chapter 5. Conclusions and suggestions for further study are presented in Chapter 6.

2. APPLICATIONS OF DEGREE THEORY

This chapter discusses degree thoery and shows how the degree of a mapping can be used to help determine what requirements are necessary if a transformation T is to produce a suitable grid.

Since the distribution of grid lines should be smooth with concentration in areas of large gradients in the physical solution, the image of T should cover the entire physical domain, that is, T should be onto. Also, the transformation should be one to one. In terms of the grid, this means that the grid lines should not overlap the physical boundary and should intersect only at points corresponding to intersection points on the mesh in the computation domain.

Requiring T to be one to one and onto is equivalent to saying that the system T(s)=p must have one and only one solution in the computational domain for each point p in the physical domain. This provides the motivation for looking at the following general problem:

Pick an open set $D \subset \mathbb{R}^n$, where \mathbb{R}^n is euclidean n-space, and let C be an open bounded set such that $\overline{C} \subset D$. If $F: D \subset \mathbb{R}^n \to \mathbb{R}^n$ is a continuous mapping and $y \in \mathbb{R}^n$ is given, how many solutions of F(x) = y exist in C?

The difficulty in solving this problem lies in the fact that in general the solutions do not vary continuously

with F or y. This difficulty may be resolved by looking instead at the difference between the number of solutions for which the Jacobian of F is positive and the number of solutions for which the Jacobian of F is negative. Loosely, this is what is called the degree of F at y with respect to C.

2.1 Defining the Degree of a Mapping

A more precise definition of the degree of a mapping F takes on different forms depending on what restrictions are placed on F. What follows are essentially the definitions presented in references [S] and [O].

<u>2.1-1</u> <u>Definition</u>. Let $C \sim p^n$ be an open bounded set and let $F:\overline{C} \subset \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable on C. Pick $y \notin F(aC)$ and let $\Gamma = \{x \in C \mid F(x) = y\}$. If F'(x) is nonsingular for all $x \in \Gamma$ then one defines the degree of F'at y with respect to C by

 $deg(F,C,y) = \sum_{x \in r} sign det F'(x).$

In [0], Ortega and Rheinboldt actually define the degree in terms of an integral and then show that it has the equivalent form given above.

On removing the restriction that det $F'(x) \neq 0$ for xer the definition becomes

 $deg(F,C,y) = \lim_{k \to \infty} deg(F,C,y_k)$

where $\lim_{k\to\infty} y_k = y$ and each element of $\{y_k\}$

satisfies $y_k \notin F(aC)$ and det $F'(x) \neq 0$ whenever $F(x) = y_k$.

Actually, one can make the stronger statement that for any such sequence $\{y_k\}$ there is a k_0 such that $deg(F,C,y) = deg(F,C,y_k)$ for $k \ge k_0$ [0, p. 159].

The Weierstrass approximation theorem makes it possible to extend the definition of the degree of a mapping to a continuous function.

<u>2.1-2</u> <u>Definition</u>. Let $F:\overline{C} \subset \mathbb{R}^n \to \mathbb{R}^n$ be continuous on the bounded open set C. Define $||F|| = \sup_{\overline{C} \to \mathscr{A}} ||F(x)||$ where $|\cdot|$ is the Euclidean norm. Then for $y \notin F(aC)$ one defines the degree of F at y with respect to C by

 $deg(F,C,y) = \lim_{j \to \infty} deg(F_j,C,y)$

where $\{F_j\}$ is a sequence of maps which are continuously differentiable on an open set D> \overline{C} and which satisfy lim $||F_j - F||_C = 0$ $j \rightarrow \infty$

2.2 Properties of the Degree

The principal properties of the degree are given below. Excellent proofs may be found in [S], [O] and [H].

<u>2.2-1</u> Theorem. Let $F:\overline{C} \subset \mathbb{R}^n \to \mathbb{R}^n \to \mathbb{C}$ continuous on the open bounded set C and let $\Gamma = \{x \in C \mid F(x) = y\}$. For any $y \notin F(aC)$ there exists a quantity, deg(F,C,y), which has the properties listed below. It is:

- 1. Integer valued
- 2. Invariant under homotopy

If $W: \tilde{C}x[0,1] = \mathbb{R}^{n+1} \to \mathbb{R}^n$ is continuous, then for any $z \in \mathbb{R}^n$ satisfying $W(x,t) \neq z$ whenever $(x,t) \in aCx [0,1], deg(W(\cdot,t),C,z)$ is constant for all $t \in [0,1]$.

- 3. Dependent only on boundary values If $G:\tilde{C} \in \mathbb{R}^n \to \mathbb{R}^n$ is continuous and $G|_{\partial C} = F|_{\partial C}$, then deg(F,C,y)=deg(G,C,y).
- Invariant under translation
 For any zεRⁿ.
 deg(F-z,C,y-z)=deg(F,C,y).
- 5. <u>Invariant for points which can be connected by a continuous path avoiding F(aC)</u> See Figure 1.
- 6. Invariant under the excision from C of any closed set Q satisfying $QAr = \emptyset$

In other words, if $Q\Lambda\Gamma=\emptyset$, then deg(F,C,y) = deg(F,C-Q,y). In particular, if Q= \overline{C} , deg(F,C-Q,y)=0. This property will be called the Excision Property.

The Excision Property can be used to prove a very important result which is called the Kronecker Theorem in [0, p. 161].

<u>2.2-2</u> <u>Theorem</u> (Kronecker). If $F:\overline{C} = \mathbb{R}^n \to \mathbb{R}^n$ is continuous on the bounded open set C, $y \notin F(aC)$ and $deg(F,C,y) \neq 0$, then the equation F(x)=y has a solution in C. <u>Proof</u>: Suppose F has no solutions in C. Let $Q=\overline{C}$. Since $y \notin F(Q)$, the Excision Property implies deg(F,C,y)=0.

Q.E.D.



 $deg(F,C,y_0) = deg(F,C,y_1)$

Figure 1. Invariance of the degree when points connected by a continuous path avoiding F(aC).

2.3 A Topological Definition of the Degree

Dugundji [D] presents an alternate formulation for the degree of a mapping. He defines the degree of a mapping $f:S \rightarrow S$ where S is the unit n-sphere in \mathbb{R}^{n} , that is,

 $S = \{x \in \mathbb{R}^n | |x| = 1\}.$

This degree can be shown to be equivalent to the analytically defined degree in the previous sections.

Before defining this degree, several terms must be discussed.

<u>2.3-1</u> <u>Definition</u>. A set $E \in \mathbb{R}^n$ is called a <u>linear variety</u> if $x_1, x_2 \in E$ implies $\lambda x_1 + (1-\lambda) x_2 \in E$ for all real λ .

2.3-2 Definition. A <u>hyperplane</u> in Rⁿ is an (n-1) dimensional linear variety. If n=1 then a hyperplane will be a point. For n=2 it will be a line, and for n=3 it is a plane.

2.3-3 Definition. If $\{x_0, x_1, \dots, x_n\}$ is a set of n+1 points in Rⁿ, then the convex hull is called an <u>n-simplex</u>. It will be denoted by $\delta = (x_0, x_1, \dots, x_n)$.

The points x_0, x_1, \ldots, x_n are called the <u>vertices</u> of the n-simplex. If the vertices lie on a hyperplane in \mathbb{R}^n , then the n-simplex is said to be <u>degenerate</u>. Now if (x_1^1, \ldots, x_n^n) are the coordinates of point x_i , then the volume of an n-simplex [F, p. 208] is given by

11

$$\frac{1}{n!} \det(x_{1} - x_{0}, x_{2} - x_{0}, \dots, x_{n} - x_{0})$$

$$= \frac{1}{n!} \det \begin{bmatrix} x_{1}^{1} - x_{0}^{1} & x_{2}^{1} - x_{0}^{1} & \dots & x_{n}^{1} - x_{0}^{1} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ x_{1}^{n} - x_{0}^{n} & x_{2}^{n} - x_{0}^{n} & \dots & x_{n}^{n} - x_{0}^{n} \end{bmatrix}$$

An n-simplex is degenerate if and only if $det(x_1-x_0, x_2-x_0, \dots, x_n-x_0)=0.$

The next three definitions will be used to explain the term "ordered n-simplex."

<u>2.3-4</u> Definition. A binary relation Λ in a set A is a subset $\Lambda \subset A \times A$.

<u>2.3-5 Definition</u>. If Λ is a binary relation in a set A, then Λ is trichotomous if exactly one of the following is true for each x,y $\in A$:

 $x \wedge y$, x = y, $y \wedge x$.

<u>2.3-6 Definition</u>. Let Λ be a binary relation in a set A. Then Λ is a total order if it is transitive and trichotomous [G, p. 2].

<u>2.3-7 Definition</u>. An <u>ordered</u> <u>n-simplex</u> [D, p. 336] is an n-simplex together with a total ordering on its vertices.

Therefore, if the vertices x_0, x_1, \ldots, x_n of an n-simplex satisfy $x_0 < x_1 < \ldots < x_n$, then "<" totally orders the set $\{x_0, x_1, \ldots, x_n\}$. Therefore, the n-simplex $\delta = (x_0, x_1, \ldots, x_n)$

is an ordered n-simplex. Such a simplex will be denoted [s] = $[x_0, x_1, ..., x_n]$. The sign of the ordered simplex is the sign of det $(x_1 - x_0, x_2 - x_0, ..., x_n - x_0)$.

Now suppose $x_0, x_1, \ldots, x_{n-1}$ is a set of n points on S having a diameter less than 1 so that the convex hull of the set does not contain the origin. Then the convex hull can be projected onto S by choosing the points on S lying on the directed rays which start at the origin and pass through the convex hull. The points on S form what will be called the <u>spherical (n-1)</u> - <u>simplex</u> $\delta = (x_0, \ldots, x_{n-1})$. The spherical simplex δ is degenerate if and only if $x_0, x_1, \ldots, x_{n-1}$, lie on a hyperplane in \mathbb{R}^n passing through the origin, that is, if and only if $(x_0, x_1, \ldots, x_{n-1}, 0)$ is a degenerate n-simplex in \mathbb{R}^n . An <u>ordered spherical (n-1)</u> -<u>simplex</u> is a spherical (n-1)-simplex with a total order on its vertices. The <u>sign</u> of an ordered spherical (n-1)-simplex [δ] = $[x_0, \ldots, x_{n-1}, 0]$ in \mathbb{R}^n [D, p. 337].

The next two definitions, which can be found in [D, p. 337], complete the terminology needed to define the Dugundji degree.

<u>2.3-8 Definition</u>. A <u>triangulation</u> \triangle of S is a decomposition of S into a finite number of nonoverlapping, nondegenerate spherical (n-1)-simplexes such that each face of an (n-1)simplex is the common face of exactly two (n-1)-simplexes.

13

2.3-9 Definition. Suppose S and Σ are unit n-spheres in \mathbb{R}^n . (Different symbols are used to make the concepts more clear.) Let Δ be a triangulation of S. A <u>proper vertex</u> <u>map</u> $\varphi: \Delta \rightarrow \Sigma$ is a map defined only on the vertices of the spherical (n-1)-simplexes in Δ and is such that whenever $x_0, x_1, \ldots, x_{n-1}$ are vertices of a simplex in Δ , the set $\{\varphi(x_0), \varphi(x_1), \ldots, \varphi(x_{n-1})\} \in \Sigma$ has diameter less than 1.

Under the proper vertex map $\varphi: \Delta \rightarrow \Sigma$ there will be a unique simplex $\varphi(\sigma)$ lying on Σ corresponding to each simplex $\sigma \epsilon \Delta$. There will be a unique ordered (n-1)-simplex $\varphi[\sigma] = [\varphi(x_0), \varphi(x_1), \ldots, \varphi(x_{n-1})]$ on Σ corresponding to each ordered (n-1)-spherical simplex $[\sigma]$. The sign of $[\sigma]$ may differ from that of $\varphi[\sigma]$, and the family of sets { $\varphi(\sigma) | \sigma \epsilon \Delta$ } may not form a triangulation of Σ since it may contain overlapping simplexes and degenerate simplexes. However, the family does have the fundamental property presented in the following theorem which Dugundji proves [D, p. 237].

<u>2.3-10</u> Theorem. Suppose Δ is a triangulation of S and $\varphi: \Delta \rightarrow \Sigma$ a proper vertex map. Let y be any point not on the boundary of any set $\varphi(\sigma)$. If $p(y, \Delta, \varphi)$ is the number of positive $\varphi[\sigma]$ containing y and $n(y, \Delta, \varphi)$ is the number of negative, then the number $D(y, \Delta, \varphi)=p(y, \Delta, \varphi)-n(y, \Delta, \varphi)$ is the same for all $y_{\varepsilon}\Sigma$ not on the boundary of any $\varphi(\sigma)$.

Since $D(y, \Delta, \phi)$ is independent of y it can be denoted $D(\Delta, \phi)$.

Now if $F:S \rightarrow \Sigma$ is continuous then the compactness of S makes it possible to find a triangulation Δ of S such that the diameter of $F(\sigma)$ is less than 1 for each $\sigma \epsilon \Delta$. Then if $\varphi_{\Gamma}: \Delta \rightarrow \Sigma$ is the proper vertex map defined by $\varphi_c(x) = F(x)$ for each vertex x of Δ , Dugundji [D, p. 339] shows that the number $D(\Delta, \phi_F)$, where ϕ_F is the proper vertex map associated with Δ , is independent of the triangulation He calls the quantity $D(\Delta, \varphi_F)$ the degree of F. of S. Since Δ and ϕ_F actually depend only on F, D($\Delta,\phi_F)$ can be denoted D(F).

14

Like the analytically defined degree, this degree is invariant under homotopy [D, p. 239].

2.3-11 Theorem. If $F: S \rightarrow \Sigma$ is homotopic to $\tilde{F}: S \rightarrow \Sigma$, then $D(F) = D(\tilde{F}).$

Now let V be the unit n-ball in Rⁿ, that is, $V = \{x \in \mathbb{R}^n \mid |x| < 1\}$. Dugundji's degree can be extended to a continuous map $H: V \rightarrow V$ provided $H|_S$ maps S into S. S is clearly the boundary of V. Dugundji calls such maps regular. The technique for determining the degree of H is analogous to what is done to obtain D(F) for $F: S \rightarrow S$. V is triangulated into n-simplexes such that each face not on S is the face of exactly two n-simplexes. Then a regular vertex map is defined on the triangulation. φ is a regular vertex map of a triangulation Δ of V if ϕ maps each vertex on S to a point on S and $\varphi|_{S}$ is a proper vertex map. To calculate

the degree of H, which will be denoted $D_{reg}(H)$, one chooses the regular vertex map $\varphi_H: \Delta \rightarrow V$ defined by $\varphi_H(x) = H(x)$ for any vertex x $\epsilon \Delta$. Then choosing $y_{\epsilon}V-S$ such that y is not on the boundary of any $\varphi_H(\sigma)$ one computes

> $D_{reg}(H) = (number of positive \varphi_H[\sigma] containing y)$ - (number of negative $\varphi_H[\sigma]$ containing y).

 $D_{reg}(H)$ depends only on H, and if H and \tilde{H} are homotopic in such a way that the image of S remains on S throughout the entire deformation, then $D_{reg}(H) = D_{reg}(\tilde{H})$. Furthermore, Dugundji proves the following very useful result.

<u>2.3-12</u> <u>Theorem</u>. Suppose $H: V \rightarrow V$ is a regular map. Let $F=H|_{S}: S \rightarrow S$. Then $D(F) = D_{reg}(H)$.

This theorem provides the information needed to show that Dugundji's degree is equivalent to the analytically defined degree. The following !emma will be used in the proof.

2.3-13 Lemma. Suppose the following hypotheses are given:

- H:V→V is a continuously differentiable regular
 map
- 2. $y \in V S$ and $\Gamma = \{x \in V | H(x) = y\}$

3. H'(x) is nonsingular for all $x \in \Gamma$

Then there exists a triangulation Δ of V with associated regular vertex map $\varphi_{\rm H}$ such that whenever $\sigma \epsilon \Delta$ contains $x \epsilon r$ and $\varphi_{\rm H}[\sigma]$ is nondegenerate,

sign $\varphi_{H}[\sigma] = \text{sign det } H'(x)$.

16

and denote that a first set of

<u>Proof</u>: For each $x \in \Gamma$ choose a neighborhood N_x of x so that either sign det H'(p)>0 for all points $p \in N_x$ or sign det H'(p) < 0 for all points $p \in N_x$. Choose the neighborhoods small enough so that the family of sets $\{N_x | x \in r\}$ is disjoint. Then triangulate V so that each N_{χ} contains a nondegenerate equilateral simplex $\sigma_{\mathbf{x}}$ in which x lies, that is, a nondegenerate simplex in which the distance between any two vertices is the same. Call this triangulation Δ . Now suppose $\sigma_x \in \Delta$ and $\sigma_x = (x_0, x_1, \dots, x_n)$ with the vertices labeled so that $[\sigma_x] = [x_0, x_1, \dots, x_n]$ is positive. Since $\varphi_{H}(x_{i}) = H(x_{i}), i = 0, 1, \dots, n, \text{ the sign of } \varphi_{H} [\sigma_{x}]$ = sign det $[H(x_1) - H(x_0), \dots, H(x_n) - H(x_0)].$ However, $H(x_i) - H(x_0) = H'(x_0) (x_i - x_0) + o(|x_i - x_0|)$ for i=1,...,n. Therefore, the sign of φ_{μ} [σ_{ν}] = sign det $[H'(x_0)(x_1-x_0) + o(|x_1-x_0|), \dots, H'(x_0)(x_n-x_0) + o(x_n-x_0)].$ Now if $|x_i - x_0| = \varepsilon$ for i=0,1,...,n then expanding the determinant above yields det $[H'(x_0)(x_1-x_0,...,x_n-x_0)]$ plus terms of order $o(\epsilon^n)$. Therefore, det $[H'(x_0)(x_1-x_0,...,x_n-x_0)]$, which has order $O(\epsilon^n)$, is the dominant term, and the other terms can be neglected. Consequently, the sign of $\varphi_{H}[\sigma_{x}] = \text{sign det } [H'(x_{0})(x_{1}-x_{0},\ldots,x_{n}-x_{0})]$ = sign [(det $H'(x_0)$) · (det $(x_1 - x_0, ..., x_n - x_0)$] = sign det $H'(x_0)$ since $[\sigma_x]$ is positive. However, since $x_0 \in N_x$, sign det $H'(x_0) = sign det H'(x)$.

Q.E.D.

<u>2.3-14</u> Theorem. Suppose $H: V \rightarrow V$ is a continuously differentiable regular map. Pick $y_{\varepsilon}V-S$ and let $\Gamma = \{x_{\varepsilon}V|H(x) = y\}$. Let $F=H|_{S}: S \rightarrow S$. If H'(x) is nonsingular for all $x_{\varepsilon}\Gamma$ then

D(F) = deg (H, V, y).<u>Proof</u>: Since 2.3-12 says $D(F) = D_{reg}(H)$, it suffices to show that $D_{reg}(H) = deg (H, V, y).$

Choose a triangulation Δ of V as specified in 2.3-13 and let φ_H be the regular vertex map associated with Δ . Without loss of generality, one can assume that $\varphi_H(\sigma)$ is nondegenerate for all $\sigma \epsilon \Delta$ because any degenerate $\varphi_H(\sigma)$ can be approximated by a nondegenerate simplex $\varphi(\sigma)$ where φ is defined on all vertices p in Δ so that $|\varphi(p)-\varphi_H(p)| < \epsilon$ for a given ϵ . According to Dugundji [D, p. 338], $D(\Delta, \varphi) =$ $D(\Delta, \varphi_H) = D(H)$ if ϵ is sufficiently small.

Furthermore, one can also assume that y does not lie on the boundary of any $\varphi_{H}(\sigma)$ for $\sigma \in \Delta$ since property number 5 of Section 2.2 implies deg(H,V,y) = deg(H,V,P) for all $p \in V-S$.

Therefore, it follows from 2.3-13 that $D_{reg}(H) =$ number of positive $\varphi_H(\sigma)$ containing y

 number of negative φ_H(σ) containing y
 Σ sign det H(x). xεΓ

Q.E.D.

The following corollary shows that the restriction that H'(x) be nonsingular for all xer can be removed.

2.3-15 <u>Corollary</u>. Suppose H: V - V is a continuously differentiable regular map. Pick $y \in V - S$ and let $\Gamma = \{x \in V | H(x) = y\}$. Let $F = H|_{S}: S \to S$. Then D(F) = deg(H, V, y). <u>Proof</u>: By Ortega and Rheinboldt [0, p. 159], there exists a sequence $\{y_k\}$ which converges to y and has the following properties:

Each y_k ∉H(S)

The Angeland strength of the st

- 2. For each y_k , det $H'(x) \neq 0$ for all x such that $H(x)=y_k$
- 3. For some k_0 , deg(H,V,y)=deg(H,V,y_k) for all $k \ge k_0$

So pick k* so that $k* \ge k_0$ and whenever $k \ge k*$, $y_k \in Y-S$. Then by 2.3-14 D(F)=deg(H,V, y_k*)=deg(H,V,y).

Q.E.D.

ł

It should be noted that this corollary still holds if H maps some of the interior points outside of V. The points outside of V can be projected onto S so that one obtains a mapping from V into V.

2.4 Applications to Grid Generation

The usefulness of degree theory in grid generation surfaces when one studies a grid generating transformation T. One might immediately note from the Kronecker theorem that determining the degree at every point in the physical domain would show whether or not T were onto. Unfortunately, the degree is not always easy to compute in practice.

One therefore looks instead at how the degree can be used to prove some things about those quantities, such as the Jacobian of T, which can be easily computed.

Recall that if $A \subset \mathbb{R}^n$, $B \subset \mathbb{R}^n$, $t^+ \in n$ A homeomorphic to B means there is a continuous one to one, onto mapping from A to B whose inverse is also continuous. It is clear that T should be a homeomorphism from the computational domain onto the physical domain.

The following result shows that if T is a homeomorphism, its Jacobian does not change sign.

In all of the theorems which follow $I_n = [0,1]^n$, JT = Jacobian of T, $C^0 = interior of C$, $C^C = complement of C, aC = boundary of C and <math>\alpha < R^n$ is homeomorphic to I_n .

<u>2.4-1</u> Theorem. If is a homeomorphism from I_n to Ω and T is continuously differentiable, then the Jacobian, JT, of T has one sign in I_n^0 , i.e., either $JT(x) \ge 0$ for all $x \in I_n^0$ or $JT(x) \le 0$ for all $x \in I_n^0$.

<u>Proof</u>: Suppose by way of contradiction that $JT(x_0) > 0$ while $JT(x_1) < 0$ for some $x_0 \cdot x_1 \in I_n^0$. Let $y_c = T(x_0)$ and $y_1 = T(x_1)$. Define $p:[0,1] \rightarrow R^n$ by

 $p(t) = T ((1-t)x_0 + tx_1).$

Then $p(0)=y_0$, $p(1)=y_1$, and $p(t)\notin T(aI_n)$ for $t_{\varepsilon}[0,1]$. Hence, by property 5, $1=deg(T, I_n^0, y_0)=deg(T, I_n^0, y_1) = -1$. Therefore, either $JT(x) \ge 0$ or $JT(x) \le 0$ for all $x_{\varepsilon} I_n^0$.

Q.E.D.

In an algebraic grid generation algorithm, the construction of T will be based on boundary information. The next theorem shows that requiring T to be a homemorphism from the boundary of the computational domain to the boundary of the physical domain will insure that the image of T covers all of the physical domain.

2.4-2 Theorem. If $T:I_n \rightarrow R^n$ is continuously differentiable and T maps aI_n homeomorphically onto a_n , then $T(I_n) \supset \alpha$. <u>Proof</u>: Let S be the unit n-sphere in R^n . By Dugundji [D, p. 353], the Dugundji degree D of a map which is a homeomorphism from S to S is +1 or -1. But aI_n and a_n are homeomorphic to S. Therefore, from 2.3-15 it follows that for any $y \in \Omega^0$, $deg(T, I_n^c, y) = \pm 1$. Therefore, by the Kronecker Theorem (2.2-2) α lies in the image of T. Q.E.D.

Smith and Sritharan [SS] show that if an additional hypothesis is added, one can obtain a much stronger conclusion:

<u>2.4-3</u> <u>Theorem</u>. If $T:I_n \rightarrow R^n$ is continuously differentiable, T maps aI_n homeomorphically onto a_Ω and $JT(x) \neq 0$ for all $x \in I_n^0$, then T is a homeomorphism from I_n to Ω .

The next theorem shows that the Jacobian changes sign when the image of T overlaps the physical boundary. Theorem 2.4-3 and Theorem 2.4-4 show that it is important that T be constructed so that its Jacobian does not change sign.

<u>2.4-4</u> Theorem. Suppose T: $I_n \rightarrow R^n$ has the following properties:

- 1. T is continuously differentiable
- 2. T maps al homeomorphically onto an
- 3. $m(T(I_n)-\Omega)>0$

Then JT has a sign change.

Q.E.D.

2.5 Additional Topological Questions

Section 2.4 suggests other questions which should be asked. Can a continuously differentiable homeomorphism from aI_n to an always be extended to a continuously differentiable homeomorphism from I_n to n? If not, under what conditions is such an extension possible? How can one guarantee that a mapping from I_n to R^n will be a diffeomorphism?

The answers to these questions will provide valuable information for creating an algebraic grid generation mapping. Although this paper does not answer all of these questions, partial answers were presented in the previous section. Also, the following example shows that continuously differentiable boundary homeomorphisms cannot always be extended.

<u>2.5-1</u> Example. Suppose $T:I_2 \rightarrow R^2$ is continuously differentiable and maps the boundary of the square homeomorphically onto the boundary of the nonconvex region Ω shown in figure 2. Let p be the point indicated and $\Delta p = \begin{pmatrix} \Delta \xi \\ -\Delta \eta \end{pmatrix}$. If $T_1(p) = \frac{\partial T}{\partial \xi}(p)$ and $T_2(p) = \frac{\partial T}{\partial \eta}(p)$ then $T(p+\Delta p) = T(p) + T'(p)(\Delta p) + o(|\Delta p|)$

= $T(p) + \Delta \xi T_1(p) - \Delta \eta T_2(p) + o(|\Delta p|)$

When $|\Delta p|$ is small, the terms of order $o(|\Delta p|)$ are negligible in size when compared to $\Delta g T_1(p)$ and $\Delta n T_2(p)$. Therefore, those terms may be neglected from the equation above. However, then it is clear that $T(p+\Delta p)$ must lie outside the boundary of Ω . Consequently, T cannot be a homeomorphism from I_2 to Ω . This is illustrated in figures 3 and 4 which show the result of attempts to construct a tensor product spline transformation that maps the square onto Ω . In each case points overlap the boundary near the "V" shaped corner.

The first grid was obtained by choosing the B-spline coefficients so that the transformation approximated a transfinite bilinear interpolation mapping. This is discussed in Chapter 4. The second grid was obtained by changing some of the coefficients in order to minimize a functional which is described in the next chapter.









3. AN ALGEBRAIC GRID GENERATION MAPPING

In this chapter an algebraic grid generation technique which uses a transformation consisting of tensor product B-splines is discussed. In the first section, finite difference approximations to the transformed derivatives of a first order partial differential equation are examined. The effect of the size of the Jacobian on smoothness and orthogonality is discussed, and its influence on local truncation error is examined. The next section defines the particular transformation of interest in this paper and discusses the properties of the building blocks for this transformation: kth order B-splines. The final section discusses a functional which can be used to modify the transformation so that the grid lines are distributed more smoothly and are nearly orthogonal at points of intersection.

3.1 A First Order Example

If ξ and n are the computational coordinates, satisfying $0 \le \xi \le 1$ and $0 \le n \le 1$, and x and y are the physical coordinates, then the grid on the physical domain will consist of coordinate lines produced by a mapping

 $T(\xi, \eta) = \begin{pmatrix} x(\xi, \eta) \\ y(\xi, \eta) \end{pmatrix} .$
If $u_t = F(x, y, u, u_x, u_y)$ is a first order partial differential equation defined on the physical domain, then the chain rule yields $(u_{\xi} \ u_{\eta}) = (u_x \ u_y) \times J$ where $J = \begin{bmatrix} x_{\xi} \ x_{\eta} \\ y_{\xi} \ y_{\eta} \end{bmatrix}$, the Jacobian matrix for the transformation T. Hence $(u_x \ u_y) = (u_{\xi} \ u_{\eta}) \times J^{-1}$

$$= (u_{\xi} u_{\eta}) \times \begin{bmatrix} y_{\eta} - x_{\eta} \\ -y_{\xi} x_{\xi} \end{bmatrix} / JT$$

where $JT = |J| = x \xi y_n - x_n y \xi$. It is clear that the partial differential equation can be transformed once the elements of J are computed. These elements may be approximated by differences when explicit formulas are not available. The transformed expressions for u_X and u_y show immediately that the grid must be structured so that $JT \neq 0$ at all mesh points (ξ, n) .

Once the partial differential equations are transformed, difference approximations can be written for u_{ξ} and u_{η} . Large truncation errors in the approximations will affect the solution of the partial differential equations. One can obtain an expression for the truncation error at mesh point (ξ_i, n_j) by doing a Taylor series expansion at (ξ_i, n_i) . If $u_{ij} = u(\xi_i, n_j)$, then

$$u_{i+i,j} = u_{ij} + u_{\xi \Delta \xi} + u_{\xi \xi} \frac{(\Delta \varepsilon)^2}{2!} + u_{\xi \xi \xi} \frac{(\Delta \varepsilon)^3}{3!} + HOT$$
$$u_{i-1,j} = u_{ij} - u_{\xi \Delta \xi} + u_{\xi \xi} \frac{(\Delta \varepsilon)^2}{2!} - u_{\xi \xi \xi} \frac{(\Delta \varepsilon)^3}{3!} + HOT$$

where HOT = higher order terms. Subtracting these two equations and solving for u_E yields

$$u_{\xi} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta\xi} - u_{\xi\xi\xi} \frac{(\Delta\xi)^2}{6} + HOT$$

Similarly,

$$u_{\eta} = \frac{u_{i,j+1} - u_{i,j-1}}{2 \Delta \eta} - u_{\eta \eta \eta} \frac{(\Delta \eta)^2}{6} + HOT$$

Therefore

$$u_{X} = \frac{1}{JT} \left(y_{\eta} \delta_{\xi} u - y_{\xi} \delta_{\eta} u \right) - \frac{1}{6JT} \left(y_{\eta} u_{\xi\xi\xi} \left(\Delta_{\xi} \right)^{2} - y_{\xi} u_{\eta\eta\eta} \left(\Delta_{\eta} \right)^{2} \right) + \dots$$

where $\delta_{\xi} u$ and $\delta_{\eta} u$ are the central difference approximations for u_E and u, respectively. The truncation error is

$$\frac{-1}{6JT} \left(y_{\eta} u_{\xi\xi\xi} (\Delta_{\xi})^{2} - y_{\xi} u_{\eta\eta\eta} (\Delta_{\eta})^{2} \right) + \dots$$
Now if $r = \binom{x}{y}$, then
$$JT = x_{\xi} y_{\eta} - x_{\eta} y_{\xi}$$

$$= \left(r_{\xi} x r_{\eta} \right) + \left(0, 0, 1 \right)^{T}$$

$$= \left| r_{\xi} \right| \left| r_{\eta} \right| \sin \theta$$

where Θ is the angle of intersection of the grid lines at (ξ ,n). Again, the importance of JT \neq 0 is evident, but one can also see why the grid lines should be as orthogonal as possible. The expression for JT implies that the truncation error is inversely proportional to sin Θ . However, according to Thompson, Warsi and Mastin [TWM, p. 82] a departure from orthogonality of up to 45⁰ is usually tolerable.

3.2 B-splines

The mapping T discussed in this paper has the form

$$T(\xi, \eta) = \begin{bmatrix} x(\xi, \eta) \\ y(\xi, \eta) \end{bmatrix} = \begin{bmatrix} m & n \\ \Sigma & \Sigma \\ i=1 & j=1^{\alpha} i j^{\beta} i j(\xi, \eta) \\ m & n \\ \Sigma & \Sigma \\ i=1 & j=1^{\beta} i j^{\beta} i j(\xi, \eta) \end{bmatrix} \xrightarrow{0 \le \eta \le 1}$$

where the B_{ij} , i=1,...m; j=1,...,n are tensor products of B-splines and the coefficients α_{ij} , B_{ij} , i=1,...,m; j=1,...,n are real numbers. In this section, the terms B-spline, spline function and tensor product B-spline are defined, and some of the important properties of these functions are discussed.

3.2-1 Defining B-splines

The following definition is from <u>A Practical Guide</u> to Splines by Carl de Boor [de B, p. 108].

<u>3.2-1-1</u> <u>Definition</u>. If $t = \{t_i\}$ is a nondecreasing sequence, then the i-th normalized B-spline of order k for knot sequence t is defined by

 $B_{i,k,t}(x) = (t_{i+k}-t_i)[t_i, \dots, t_{i+k}](\cdot - x)_+^{k-1} \text{ where } x \in \mathbb{R}.$ The sequence t may be finite, infinite or biinfinite. The expression $[t_i, \dots, t_{i+k}](\cdot - x)_+^{k-1}$ denotes the kth divided difference of $(\cdot - x)_+^{k-1}$, or the leading coefficient of the polynomial of degree k which interpolates $(\cdot - x)_+^{k-1}$ at t_1, \ldots, t_{i+k} . The notation $(\cdot - x)_+^{k-1}$ represents the truncated power function $(\tau - x)_+^{k-1}$ which is defined by

$$(\tau - x)_{+}^{k-1} = \begin{cases} (\tau - x)^{k-1} & \text{for } \tau > x \\ 0 & \text{for } \tau \le x \end{cases}$$

The -indicates that the kth divided difference above should be evaluated by holding x fixed and considering $(\tau - x)_{+}^{k-1}$ as a function of τ only. Nevertheless, since $B_{i,k,t}(x)$ changes as one chooses different values for x, it is clearly a function of x.

The definition above differs slightly from the original definition given by Curry and Schoenberg. Their B-spline $M_{i,k,t}$ is related to $B_{i,k,t}$ by the equation

 $M_{i,k,t} = [k/(t_{i+k}-t_i)] B_{i,k,t}$ [de B, p. 109].

3.2.2 Properties of B-splines

A kth order B-spline $B_{i,k,t}$ is a piecewise polynomial of degree k-1 with breakpoints at t_i, \ldots, t_{i+k} . On each interval (t_j, t_{j+1}) , $B_{i,k,t}$ is a polynomial of degree k-1 or less. For convenience it will be assumed that $B_{i,k,t}$ is continuous from the right at breakpoints.

B-splines have many properties which make them convenient for applications involving computers. One important property is their small support. If $x \neq [t_i, t_{i+k}]$, then $(\tau - x)_{+}^{k-1}$ will be a polynomial of degree k-1 or less on $[t_i, t_{i+k}]$. Hence $[t_i, ..., t_{i+k}] (\tau - x)_+^{k-1} = 0$. Therefore, $B_{i,k,t}(x) = 0$ for $x \notin [t_i, t_{i+k}]$.

This implies that the support of $B_{i,k,t}$ can lie in at most k intervals of the form $[t_j,t_{j+1}]$. Therefore, if $\{B_i\}$ represents the sequence of B-splines of order k for the knot sequence $t = \{t_i\}$, it follows that only the k B-splines B_{j-k+1} , B_{j-k+2} ,..., B_j can have support in any given interval $[t_j, t_{j+1}]$.

The next two results, which are proved in [de B, p. 110] and [de B, p. 130], respectively, show that B-splines form a partition of unity, i.e., the sequence $\{B_i\}$ consists of nonnegative functions which sum up to 1.

<u>3.2.2-1</u> Theorem. If $\{B_i\}$ is the sequence of B-splines of order k for a nondecreasing sequence $t = \{t_i\}$, then

$$\sum_{i} B_{i}(x) = \frac{q-1}{\Sigma} B_{i}(x) = 1$$
$$i=p-k+1$$

for any $x \in (t_p t_q)$ where p and q are such that p-k+1 and q+k-1 lie in the index set for t.

<u>3.2.2-2</u> Theorem. If B_i is the ith element of the sequence of B-splines of order k for a nondecreasing sequence $t = \{t_i\}$, then $B_i(x) > 0$ for $t_i < x < t_{i+k}$.

One can think of the "B" in B-splines as representing the word "basis," for when the knot sequence t is chosen

appropriately, the kth order B-spliner for t form a basis for the piecewise polynomial space $P_{k,\xi,v}$. $P_{k,\xi,v}$ is the notation used by de Boor [de B, p. 100] to represent the space of piecewise polynomials of degree k-1 which have breakpoint sequence ξ and which satisfy smoothness conditions specified by v. If $\xi = \{\xi_i\}_1^{m+1}$, then the nonnegative sequence $v = \{v_i\}_2^m$ gives the number of smoothness conditions at each ξ_i , i = 2, ..., m. For example, if $v_g = 3$ then any $f \in P_{k,\xi,v}$ must have at least 3 smoothness conditions at ξ_g , that is, the function, its derivative and second derivative must be continuous at ξ_g . The dimension of $P_{k,\xi,v}$

The following theorem of Curry and Schoenberg [de B,C] shows how the knot sequence t should be chosen so that the corresponding B-spline sequence forms a basis for $P_{k,E,v}$.

3.2.2-3 Theorem (Curry and Schoenberg).

Let $\xi = \{\xi_i\}_1^{m+1}$ be a strictly increasing sequence and $v = \{v_i\}_2^m$ be a nonnegative integer sequence such that $v_i \leq k$ for all i. Set $n = k + \frac{m}{2} (k - v_i) = km - \frac{m}{2} v_i$ and let i = 2 $t = \{t_i\}_1^{n+k}$ be a nondecreasing sequence such that (i) $t_1 \leq t_2 \leq \cdots \leq t_k \leq \xi_1$ and $\xi_{m+1} \leq t_{n+1} \leq \cdots \leq t_{n+k}$ (ii) for i=2,...,m, the number ξ_i occurs exactly $k-\nu_i$ times in t.

Then the sequence B_1, \ldots, B_n of B-splines of order k for the knot sequence t is a basis for $P_{k,\xi,\nu}$, viewed as functions on $[t_k, t_{n+1}]$.

This theorem shows how the number of knots at a breakpoint translates into the amount of smoothness there. Since the number ξ_i occurs exactly k-v_i times in t and v_i represents the number of smoothness conditions at ξ_i , the number of smoothness conditions at ξ_i equals k minus the number of knots at ξ_i . Hence if k=4 and ξ_j , $2 \le j \le m$, occurs exactly once in t then the piecewise polynomials generated by B_1, \ldots, B_n will satisfy three smoothness conditions at ξ_j , i.e., the piecewise polynomials, their first derivative and their second derivative will be continuous at ξ_i .

3.2.3 Spline Functions

In early studies of splines, a spline function of order k was defined to be a piecewise polynomial of degree k-1 with k-2 continuous derivatives. However, in this paper the more general definition in [de B] is used.

3.2.3-1 Definition. If $t = \{t_i\}$ is a nondecreasing sequence, then a <u>spline function</u> of order k with knot sequence t is any linear combination of the B-splines of order k for the knot sequence t. If one denotes the collection of all such functions by $S_{k,t}$ then

 $S_{k,t} = \{ \sum_{i=1}^{\infty} B_{i,k,t} : \alpha_i \text{ real for all } i \}.$

It is clear that when t has the form described in the Curry and Schoenberg theorem 3.2.2-3, $S_{k,t}=P_{k,E,v}$ on $[t_k,t_{n+1}]$.

The first derivative of a spline function $\sum_{i=1}^{\infty} B_{i,k,t}$ can be found by using the differences between successive coefficients. The following result, proved in [de B, p. 138], shows that the derivative of a spline function of order k will be a spline function of order k-1.

<u>3.2.3-2</u> <u>Theorem</u>. Let $\sum_{i=1}^{\infty} B_{i,k,t}$ be a kth order spline function constructed with B-splines $B_{i,k,t}$ corresponding to a nondecreasing sequence $t = \{t_i\}$. Then the first derivative of $\sum_{i=1}^{\infty} B_{i,k,t}$ is given by

$$\frac{d(\sum_{i=1}^{\infty} a_i B_{i,k,t}) = \sum_{i=1}^{\infty} (k-1) \frac{\alpha_i - \alpha_i}{t_{i+k-1} - t_i} B_{i,k-1,t}$$

The value of a spline function $f = \sum_{j=1}^{\infty} a_j B_{j,k,t}$ at a point x satisfying $t_i < x < t_{i+1}$ is a convex combination of the k coefficients a_{i+1-k}, \dots, a_i . For if $t_i < x < t_{i+1}$, then $f(x) = \sum_{j=1}^{\infty} B_{j,k,t}(x) = \sum_{j=1-k+1}^{i} a_j B_{j,k,t}(x)$ with the $B_{j,k,t}$ satisfying $\sum_{j=1}^{\infty} B_{j,k,t}(x) = 1$ and $B_k(x) \ge 0$ for all j. B-spline coefficients model the functions that they represent. In other words, the coefficients are approximately equal to the value of the function at certain points. This is illustrated in the next section.

Carl de Boor [de B] proves the following result concerning the relationship between a spline function and its B-spline coefficients. The notation $||f||_{[a,b]}$ denotes max $|f(x)|_{.xe[a,b]}$

<u>3.2.3-3 Theorem</u>. Let $\sum_{i=1}^{\infty} B_{i,k,t}$ be a kth order spline function constructed with B-splines $B_{i,k,t}$ corresponding to a nondecreasing sequence $t = \{t_i\}$. Then there exists a positive constant D_k , depending only on k, so that for all i,

$$|\alpha_i| \leq D_k ||_j^{\Sigma} \alpha_j \beta_{j,k,t} ||_{[t_{i+1},t_{i+k-1}]}$$

3.2.4 Variation Diminishing Splines

Given an f known to lie in $P_{k,\xi,v}$ one can write it in the form f = $\prod_{\Sigma}^{n} \alpha_i B_i$. The Curry and Schoenberg Theorem i=1

(3.2.2-3) shows how one obtains the B-spline basis and the following lemma suggests how one might obtain the coefficients. Its proof may be found in [de B, p. 116].

3.2.4-1 Lemma (de Boor and Fix). Let B_i be the sequence of B-splines of order k for a nondecreasing sequence $t = \{t_i\}$. Let λ_i be the linear functional defined for all f by $\lambda_i f = \sum_{r=1}^{k-1} (-) \sum_{r=1}^{k-1-r} e^{(k-1-r)}(\tau_i) f^{(r)}(\tau_i)$ where $\phi(t) = (t_{i+1}-t) \dots (t_{i+k}-t) / (k-1)!$ and τ_i is some arbitrary point in the open interval (t_i, t_{i+k}) . Then $\lambda_i B_i = \sigma_{ij}$ for all j. Hence, if $f = \prod_{k=1}^{n} \alpha_{i} B_{i}$ it follows that α_{k} , $1 \le k \le n$ may be found by computing $\lambda_k f = \lambda_k (\sum_{i=1}^{\infty} B_i) = \alpha_k$. By explicitly writing out the expression for λ_k f one can easily show [de B, p. 159] that $\alpha_i = f(\tau_i) + O(|t|)$ if τ_i is any point in (t_i, t_{i+k}) and $|t| = \max_{i} \{t_{i+1} - t_i\}$. However, if $\tau_i = t_i^*, 1 \le i \le n$ where $t_i^* = (t_{i+1} + ... + t_{i+k-1}) / (k-1)$ then $\alpha_i = f(t_i^*) + O(|t|^2)$. Choosing $\alpha_i = f(t_i)$ for $1 \le i \le n$ yields a shape preserving approximation called Schoenberg's variation diminishing spline approximation [de B, p. 159]. So if $t \neq \{t_i^*\}$ h the variation diminishing spline approximation to f, vf, is defined by

 $vf = \sum_{i=1}^{n} f(t_i^*) B_i.$

This spline reproduces polynomials of degree one, i.e., if f is a straight line then vf = f. For any f the number of

37

times the spline approximation crosses a given line will be less than or equal to the number of times f crosses the line. From this it follows that if f is nonnegative, then vf is nonnegative and if f is convex then vf is convex. However, since v has these shape preserving properties, it is not a very high order approximation. In fact, if g is a function defined on [a,b] and g has m continuous derivatives for some $m \ge 2$, then de Boor [de B, p. 161] states that $||g-vg||_{[a,b]} \leq g_k |t|^2$, where $c_{g,k}$ is a constant depending on the order of the spline function k and the function q. No matter how large m is, no exponent larger than 2 can be put in the inequality. De Boor shows that it is possible to obtain other spline approximations which are more accurate, but variation diminishing splines are convenient for applications such as computer-aided design and grid generation where shape preservation is important.

3.2.5 Tensor Product B-splines

<u>3.2.5-1</u> <u>Definition</u>. Let R be the set of real numbers. If V is a linear space of functions mapping some set X into R and W is a linear space of functions mapping some set Y into R, then for each veV and weW the <u>tensor product</u>, vew of v and w is defined by

 $v \blacksquare w(x,y) = v(x)w(y)$ for $(x,y) \in XxY$. Furthermore, the set of all finite linear combinations of the form v∎w for some v∈V and w∈W is called the tensor

product, VBW of V with W.

A typical element u of VBW has the form

 $u = \frac{n}{\sum} \alpha_j (v_j \mathbf{P} w_j)$ j=1

where $\alpha_j \in \mathbb{R}$, $v_j \in V$, $w_j \in W$ for j = 1, ..., n.

If V and W are the linear spaces of spline functions $S_{h,s}$ and $S_{k,t}$, respectively, then the elements of VBW are linear combinations of tensor product B-splines. A tensor product B-spline B_{ij} is defined by $B_{ij}(x,y)=B_{i,h,s}(x)B_{j,k,t}(y)$ where $B_{i,h,s}$ is the ith B-spline of order h for the knot sequence $s = \{s_i\}$ and $B_{j,k,t}$ is the jth B-spline of order k for the knot sequence $t = \{t_j\}$. An element u of VBW will be called a tensor product spline and will have the form $u = \sum_{i=1}^{r} \sum_{\alpha_{ij}} B_{ij}$

where $\alpha_{ij} \in \mathbb{R}$ for all i,j. When h=k=4 u may also be called a bicubic spline.

Many of the properties of tensor product B-splines follow trivially from B-spline properties. For example, the tensor product B-spline B_{ij} will be positive on its support since both $B_{i,h,s}$ and $B_{j,k,t}$ are positive on their support. Furthermore, the support of B_{ij} is small. Since $B_{i,h,s}(x) = 0$ for $x \notin [s_i, s_{i+h}]$ and $B_{j,k,t}(y) = 0$ for $y \notin [t_j, t_{j+k}]$ it is clear that $B_{ij}(x, y) = 0$ if either $x \notin [s_i, s_{i+h}]$ or $y \notin [t_j, t_{j+k}]$. Hence the support of B_{ij} lies in the shaded

39

Ľ,

area shown in figure 5.

Tensor product B-splines also form a partition of unity. It follows from 3.2.2-1 that $\Sigma\Sigma B(x,y) = ij ij$ $\Sigma B(x)\Sigma B(y) = 1$ for any $(x,y) \in (s_p,s_q)x(t_r,t_m)$ where p i i j j and q are such that p-h+l and q+h-l lie in the index set for sequence s and r-k+l and m+k-l lie in the index set for sequence t.

Partial derivatives of tensor product splines are easy to compute since they reduce to derivatives of spline functions.

 $\frac{\partial}{\partial x} \left(\begin{array}{c} \Sigma\Sigma & \alpha & B \\ ij & ij & ij \end{array} \right) = \begin{array}{c} \frac{\partial}{\partial x} & \Sigma\Sigma & B \\ \frac{\partial}{\partial x} & ij^{\alpha} & ij \end{array} \left(\begin{array}{c} x & B \\ i & j \end{array} \right) \\ = \begin{array}{c} \SigmaB \\ j & j \end{array} \left(\begin{array}{c} y \\ \frac{\partial}{\partial x} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial x} \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} \Sigma\alpha & B \\ \frac{\partial}{\partial y} \end{array} \right) \left(\begin{array}{c} z \end{array} \right) \left(\begin{array}{c} z \\ \frac{\partial}{\partial$

3.3 A Smoothing Functional

The mapping T described in this paper uses tensor product B-splines to map the unit square onto a physical domain of arbitrary shape. This section shows that choosing the coefficients of the tensor product B-splines so that they minimize a certain functional can improve the quality of the physical grid produced by T. This functional is described and conditions under which it will





41

1

have a minimum are examined.

3.3.1 Characteristics of the Functional

The coefficients of the mapping defined by

$$T(\xi,\eta) = \begin{bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{bmatrix} = \begin{bmatrix} m & n \\ \Sigma & \Sigma \\ i=j & j=l^{\alpha} i j^{\beta} i j(\xi,\eta) \\ m & n \\ \Sigma & \Sigma \\ i=l & j=l^{\beta} i j^{\beta} i j(\xi,\eta) \end{bmatrix} \xrightarrow{0 \le \eta \le l} 0$$

can be divided into two groups: boundary coefficients and interior coefficients. T uses the boundary coefficients, α_{ij} , β_{ij} , j=1,...,m and α_{im} , β_{im} , i=1,...,n to map the boundary of the square onto the boundary of the physical domain. Hence, the flexibility of their values is limited. The rest of the coefficients, the interior coefficients, can be moved around in order to change the characteristics of the physical grid. To produce orthogonality in the grid 'ines and maximize the smoothness of the distribution of grid lines one can choose the interior coefficients to minimize the functional

$$F = \int_{I_2} w_1 \left(\left(\frac{\partial JT}{\partial \xi} \right)^2 + \left(\frac{\partial JT}{\partial \eta} \right)^2 \right) dA + \int_{I_2} w_2 (Dot)^2 dA$$

where

$$JT(\xi,\eta) = Jacobian \text{ of } T \text{ at } (\xi,\eta)$$
$$= \begin{vmatrix} \frac{\partial x}{\partial \xi} & (\xi,\eta) & \frac{\partial x}{\partial \eta} & (\xi,\eta) \\ \frac{\partial y}{\partial \xi} & (\xi,\eta) & \frac{\partial y}{\partial \xi} & (\xi,\eta) \\ \frac{\partial y}{\partial \xi} & 0 & \eta \end{vmatrix}$$

$$= \frac{\partial x}{\partial \xi} (\xi, \eta) \frac{\partial y}{\partial \eta} (\xi, \eta) - \frac{\partial y}{\partial \xi} (\xi, \eta) \frac{\partial x}{\partial \eta} (\xi, \eta),$$

Dot $(\xi, \eta) = \frac{\partial T}{\partial \xi} (\xi, \eta) \cdot \frac{\partial T}{\partial \eta} (\xi, \eta)$

$$= \left[\frac{\partial x}{\partial \xi} (\xi, \eta) \\ \frac{\partial y}{\partial \xi} (\xi, \eta) \\ \frac{\partial y}{\partial \eta} (\xi, \eta) \right] \cdot \left[\frac{\partial x}{\partial \eta} (\xi, \eta) \\ \frac{\partial y}{\partial \eta} (\xi, \eta) \\ \frac{\partial y}{\partial \xi} (\xi, \eta) - \frac{\partial x}{\partial \eta} (\xi, \eta) + \frac{\partial y}{\partial \xi} (\xi, \eta) \\ \frac{\partial y}{\partial \eta} (\xi, \eta) - \frac{\partial y}{\partial \eta} (\xi, \eta) \right]$$

and $w_1(\xi, n)$, $w_2(\xi, n) =$ weight functions evaluated at (ξ, n) . After the minimization of F is completed, where w_1 is large the variation of the Jacobian values at nearby points will be small. Hence, w_1 can be used to decrease skewness in a grid. Where w_2 is large, Dot will be small causing the grid lines to approach orthogonality.

To avoid the tedious differentiation and integration of tensor product B-splines, the following discrete approximation to F can be implemented in computer algorithms:

$$G = \sum_{i=1}^{p} \sum_{j=1}^{q} \left(\frac{\left(JT_{i+1,j} - JT_{ij}\right)^{2}}{\left(\Delta\xi\right)^{2}} + \left(\frac{JT_{i,j+1} - JT_{ij}}{\left(\Delta\eta\right)^{2}}\right)^{2} \right) \Delta\xi\Delta\eta$$

+
$$\sum_{i=1}^{p} \sum_{j=1}^{q} w_{2} \left(Dot_{ij}\right)^{2} \Delta\xi\Delta\eta$$

$$i = 1 \quad j = 1$$

where

 $0 = \xi_1 < \xi_2 < \ldots < \xi_p = 1$,

$$0 = n_1 < n_2 < \dots < n_q = 1,$$

$$JT_{ij} = JT (\xi_i n_j), Dot_{ij} = Dot (\xi_i \cdot n_j),$$

$$\Delta \xi = 1/(p-1), \Delta n = 1/(q-1), \text{ and}$$

the parameters w_1 and w_2 are weight functions. Both F and G depend only on the coefficients of the tensor product B-splines which compose T.

Now

$$\frac{\partial x}{\partial \xi} (\xi, \eta) = \prod_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} \frac{\partial}{\partial \xi} (B_{ij}(\xi, \eta))$$

$$\frac{\partial x}{\partial \eta} (\xi, \eta) = \prod_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} \frac{\partial}{\partial \eta} (B_{ij}(\xi, \eta))$$

$$\frac{\partial y}{\partial \xi} (\xi, \eta) = \prod_{i=1}^{m} \sum_{j=1}^{n} \beta_{ij} \frac{\partial}{\partial \xi} (B_{ij}(\xi, \eta))$$

$$\frac{\partial y}{\partial \eta} (\xi, \eta) = \prod_{i=1}^{m} \sum_{j=1}^{n} \beta_{ij} \frac{\partial}{\partial \eta} (B_{ij}(\xi, \eta)).$$

Thus, for ξ , n fixed, $JT(\xi, n)$ is a linear function in each coefficient α_{ij} , β_{ij} , $i=1,\ldots,m$, $j=1,\ldots,n$ and Dot (ξ,n) is a quadratic polynomial in each coefficient. Since the terms involving Dot (ξ,n) and $JT(\xi,n)$ are squared in G, one can see that G is actually a quartic polynomial in each coefficient. This suggests an elementary iteration method for finding the minimum of G: the cyclic coordinate method [B, p. 271].

The cyclic coordinate method is a multidimensional search technique for minimizing a function of several variables without using derivatives. It searches for a minimum along each coordinate direction. This method, when applied to a differentiable function, converges to a point where the gradient is zero [B, p. 273]. It can be applied to G if one treats each coefficient α_{ij} , β_{ij} , $i=1,\ldots,m$, $j=1,\ldots,n$ as a variable representing a particular coordinate direction. This technique is discussed further in the next chapter.

The importance of requiring that the Jacobian of T be of one sign was illustrated in Chapter 2. For this reason, if possible, the feasible region for the minimization problem is chosen to be a region where the Jacobian of T is nonnegative. Now since B-splines have small support, any given coefficient α_{rs} or β_{rs} only affects the Jacobian of T at a small number of points on the unit square mesh. By solving the inequality $JT \ge 0$ for α_{rs} at each of these points one can determine on what interval α_{rs} must lie so that the Jacobian values at the points it affects are nonnegative. This inequality is easy to solve since JT is linear in a_{re} . Repeating this procedure for each coefficient will, in most cases, produce a satisfactory approximation to the desired feasible region. However, since the boundary coefficients are fixed, there may sometimes be problems near the boundary. This is the case with the nonconvex

region examined in Section 2.5. The Jacobian will remain negative at one of its corner points even after the domain for the coefficients is restricted by using the procedure above. This is because the boundary points are fixed and not affected by the procedure. The Jacobian will also remain negative near this corner because of continuity.

3.3.2 Convergence of the Umoothing Functional

Under what conditions will the discrete smoothing functional G converge to a minimum value? Is it important that G be restricted to a region where the Jacobian of T is nonnegative? What happens if one of the tensor product coefficients becomes large?

These are some of the questions which might be asked about G. The notation defined below will be used to discuss these problems:

 $\{A_r\}$ is a sequence in which each term represents a set of coefficients for the mapping T:I₂ R² defined by

 $T(\xi,\eta) = \begin{bmatrix} m & n \\ \Sigma & \Sigma & \alpha_{ij} B_{ij}(\xi,\eta) \\ i=l & j=l \end{bmatrix}, 0 \le \xi \le l, 0 \le \eta \le l.$ $m & n \\ \Sigma & \Sigma & \alpha_{ij} B_{ij}(\xi,\eta) \\ i=l & j=l \end{bmatrix}$

by

Each A_r can be considered a discrete function defined

$$A_{r}(s,i,j) = \begin{bmatrix} \alpha_{ij}^{r}, & \text{if } s = 1 \\ \beta_{ij}^{r} & \text{if } s = 2 \end{bmatrix}, \quad i=1,\ldots,m; \quad j=1,\ldots,n.$$

T denotes the mapping obtained when the coefficients

47

 $T_{\rm r}$ denotes the mapping obtained when the coefficients given by $A_{\rm r}$ are used for T.

 JT_r denotes the Jacobian of T_r .

 $|A_r|_{\max} = \max_{s,i,j} |A_r(s,i,j)|.$

It follows that if the sequence $\{A_r\}$ of coefficients converges to a single point then the corresponding values of G also converge. Hence, it is important to determine conditions which guarantee the convergence of the coefficient sequence. Well, since the elements of $\{A_r\}$ can be viewed as points in \mathbb{R}^{2mn} , the sequence converges if and only if it is a Cauchy sequence; however, a necessary condition for the convergence of $\{A_r\}$ is that the sequence be bounded. The following theorem and corollary show how the Jacobian affects the boundedness of the sequence.

<u>3.3.2-1</u> Theorem. Suppose for all $r T_r$ maps al₂ homeomorphically onto an. If $JT_r(\xi, n) \ge 0$ for all r and all points $(\xi, n) \in I_2^0$, then either $\{A_r\}$ is bounded or $JT_r(\xi_0, n_0) = 0$ for some point $(\xi_0, n_0) \in I_2^0$.

<u>Proof</u>: By way of contradiction, suppose $\{A_r\}$ is not bounded and $JT_r(\xi,n)>0$ for all r and all points $(\xi,n) \in I_2^0$. For any integer N there exists an $A_r \in \{A_r\}$ such that $|A_r|_{\max} > N$. But this implies that either $a_N^{r_N}|>N$ or $|\beta_{ij}^{r_N}|>N$ for some i,j. Now since α is bounded there exists M>0 such that $|\vec{p}| \leq M$ for all $\vec{p} \in \Omega$. From 3.2.3-3 it follows that for large enough N, $\max_{\substack{0 \leq \xi < 1 \\ 0 \leq n \leq 1}} |T_r|_N (\xi,n)| > 2M$. Hence T_{r_N} maps some point $(\xi,n_1) \in I_2^0$ outside a Ω . Now since $JT_{r_N}>0$ on I_2^0 , $m(T_r(I_2)-\alpha)>0$. But then 2.4.4 says that JT_{r_N} has a sign change.

Q.E.D.

The corollary below follows immediately.

<u>3.3.2-2</u> <u>Corollary</u>. Suppose for all r T_r maps I_2 homeomorphically onto an. If $JT_r(\xi,\eta)>0$ for all r and all points $(\xi,\eta) \in I_2^0$, then $\{A_r\}$ is bounded.

One would like to show that the requirement $JT_r(\xi,n) \ge 0$ for all r and all points $(\xi,n) \in I_2^0$ is sufficient to guarantee the boundedness of $\{A_r\}$. As indicated in 3.3.2-1, it is clear that if the magnitude of a coefficient is large enough, then the mapping T_r associated with the coefficient will map some point in I_2^0 outside an. However, it is no longer clear that $m(T_r(I_2)-\alpha)>0$ since $JT_r(\xi,n)$ may be 0 outside $\alpha\alpha$. Thus 2.4.4 cannot be used to obtain the contradiction that JT_r must have a sign change as was done in Theorem 3.3.2-1. Although the writer has been unable to devise an acceptable proof to date, further study may show that the inequality, $m(T_r(I_2)-\alpha)>0$, is actually true.

4. PROGRAM TENTEST

This chapter discusses the computer program TENTEST which algebraically generates grids using tensor product cubic B-splines. A listing of TENTEST is given in the appendix at the end of this paper.

The first section of this chapter presents the major steps involved in the computer algorithm. Sections 2 through 5 examine the important features of the program, briefly discussing the subroutines involved.

4.1 The Algorithm

Although TENTEST contains almost a thousand lines of code, it is based on the following eight step algorithm:

- i. Input knot sequences {s $_i$ } and {t $_j$ } consisting of values from [0,1].
- ii. Compute the tensor product cubic B-splines corresponding to the knot sequences.
- iii. Choose initial coefficients to form a bicubic spline mapping from the square to a physical domain.
 - iv. Use the mapping to plot a grid on the physical domain.
 - v. If grid satisfactory, stop. If grid unsatisfactory, continue.
 - vi. Input weights for smoothing functional.
- vii. Complete one iteration of minimization routine to obtain new coefficients.

viii. Go to step iv.

There also exists a batch version of TENTEST which allows the user to request several iterations of the minimization routine at a time. All the information needed to plot the initial and final grids is stored in files which can be interactively accessed after the execution of the program is completed.

The programs were run on a PRIME 750 computer. The PRIMOS operating system, coupled with a PLOT 10 graphics package, was used to interactively draw the grids on a Tektronics 4014 terminal. The PRIME 750 can communicate at a baud rate of up to 9600 thus making it satisfactory for interactive graphics.

4.2 Computing the Tensor Product B-splines

Since B-splines are determined by the knots with which they are associated, the first concern of the user is to choose appropriate knot sequences. The user must pick two _equences $s=\{s_i\}$ and $t=\{t_j\}$, placing them in file TENSORDAT. The user actually picks only the "interior" knots for each sequence. In other words, he constructs two increasing sequences of numbers between 0 and 1. After reading the numbers from file TENSORDAT, TENTEST places four 0's at the beginning of each sequence and four 1's at the end of each sequence. By 3.2.2-3 (Curry and Schoenberg) and 3.2.3-1, the cubic B-splines associated with s and t

form bases for spline spaces $S_{4,s}$ and $S_{4,t}$. The functions in each of these spaces will have three continuity conditions at each interior knot. The products of the B-splines will form a basis for the tensor product of $S_{4,s}$ and $S_{4,t}$. The tensor product B-splines can be used to construct a transformation T on the square which maps the boundary of the square onto the boundary of a physical domain as described in Chapter 3. The user may obtain a better approximation to the boundary of the physical domain by increasing the number of interior knots in s and t or by redistributing the knots. This is discussed in more detail in Section 4.5.

On a given pxq mesh on the square with mesh points (ξ_u, n_v) , u=1,...,p, v=1,...,q, the values of the tensor product B-splines which compose T are fixed. Since these tensor product B-splines are the products of B-splines $B_i, i=1,...,m$ and $B_j, j=1,...,n$ for some m and n, it is convenient to store the function values and first derivatives of these B-splines at each ξ_u and n_v . Subroutine COMSPLINE uses the de Boor routine BSPLVD [de B, p. 288] to compute these values. BSPLVD calculates the function value and derivatives of all the nonvanishing B-splines at a given point. COMSPLINE stores the function values and first derivatives in two arrays: XSPLINE and YSPLINE. Therefore, after a call to COMSPLINE is completed, XSPLINE will contain the function value and first derivative of each B-

spline in $\{B_i\}_{i=1}^m$ at ξ_u , u=1,...,p and YSPLINE contains the function value and first derivative of each B-spline in $\{B_j\}_{j=1}^n$ at $n_V, v=1,...,q$. Computing T or its partial derivatives at a mesh point becomes a matter of calculating the sum of the products of the tensor product coefficients with the appropriate elements of XSPLINE and YSPLINE. This computation is done in subroutine TENVALF.

The next section explains how the coefficients are chosen initially.

4.3 Choosing the Initial Coefficients

Many different methods can be used to choose the coefficients initially. Since B-spline coefficients model the function they represent, one might simply choose the boundary coefficients to equal points along the boundary of the physical domain, and choose the interior coefficients to equal points known to lie in the interior of the physical domain. However, this creates the problem of deciding which interior points should be chosen as coefficients. Ideally, the original coefficients should produce a grid which is somewhat smooth so that only a few iterations are needed to obtain an acceptable degree of smoothness and orthogonality.

For this reason, the computer program described in this paper initially selects coefficients which produce an approximation to the transfinite bilinear interpolant of a mapping $V:I_2 \rightarrow R^2$ satisfying $V: \partial I_2 \rightarrow \partial \Omega$. In reality

one need only define V on aI₂. The user may provide parametric equations which map the boundary of the square onto the boundary of the physical domain, or simply input a set of boundary points for the physical domain. In the first instance V is defined by using the parametric equations. In the latter case V is obtained by linearly interpolating between successive boundary points. The parametric equations below map the four sides of the unit square onto the four sides of the trapezoid as shown in figure 6.

$$V(\xi, 0) = g_{1}(\xi) = \begin{pmatrix} 2\xi+1\\ 0 \end{pmatrix}$$

$$V(1, \eta) = g_{2}(\eta) = \begin{pmatrix} 3+\eta\\ 2\eta \end{pmatrix}$$

$$V(\xi, 1) = g_{3}(\xi) = \begin{pmatrix} 4\xi\\ 2 \end{pmatrix}$$

$$V(0, \eta) = g_{4}(\eta) = \begin{pmatrix} 1-\eta\\ 2\eta \end{pmatrix}$$

by

The tra.sfinite bilinear interpolant U of V is defined

$$J(\xi, n) = (1 - n)V(\xi, 0) + nV(\xi, 1) + \xi V(1, n) + (1 - \xi)V(0, n) - (1 - \xi)(1 - n)V(0, 0) - \xi(1 - n)V(1, 0) - (1 - \xi) nV(0, 1) - \xi nV(1, 1).$$

U agrees with V on the boundary of the square and hence interpolates V at an infinite number of points. Transfinite interpolants are discussed by William J. Gordon and Charles A. Hall in [G].

The program selects initial coefficients which produce a variation diminishing spline approximation to U.



Figure 6. Mapping from computational domain to physical domain.

đ

Hence, if T is constructed from tensor products of B-splines $B_i = B_{i,4,s}$, i=1,...,m and $B_j = B_{j,4,t}$, j=1,...,n, which correspond to knot sequences $s = \{s_i\} \frac{m+4}{i=1}$ and $t = \{t_j\}_{j=1}^{n+4}$, respectively, then the initial coefficients of the tensor product splines are $\left\{ \alpha_{i,j} \atop l = U(s_i^*, t_j^*), i=1,...,m; j=1,...,n \text{ where} \right.$ $\left\{ a_{i,j} \atop l = U(s_i^*, t_j^*), i=1,...,m; j=1,...,n \text{ where} \right.$ $s_i = (s_{i+1}, +...+s_{i+3})/3$, i=1,...,m and $t_j = (t_{j+1}+...+t_{j+3})/3$, j=1,...,n. Since variation diminishing splines yield exact approximations to linear polynomials, T will reproduce the boundary of any physical domain which can be divided into four line segments. Arbitrarily shaped boundaries can be approximated as accurately as desired by increasing the number of knots used to define the tensor product splines or by changing the placement of knots to increase the concentration in complex shaped areas of the boundary.

The initial tensor product coefficients are constructed in subroutines BOUNCOEF and INNERCOEF. Figure 7 shows a grid on a trapezoid domain constructed with a mapping T having coefficients as described above. The grid is the image of T over an equally spaced mesh on the square.

4.4 Minimizing the Smoothing Functional

In TENTEST, the cyclic coordinate method is used to find the minimum of the smoothing functional G described in Section 3.3. As the name suggests, this method attempts to find the minimum of a multivariable function by cyclicly searching in the direction of each coordinate axis. For



W C C K

and in

Figure 7. Trapezoid Grid



G, the coordinate directions are represented by the tensor product coefficients α_{ij} , β_{ij} , $i=1,\ldots,m$; $j=1,\ldots,n$.

The user must first decide what size mesh should be used to obtain a grid with acceptable smoothness and orthogonality. G is a function of 2mn coefficients, however, since the boundary coefficients are fixed only 2(m-2)(n-2)coefficients are free. Therefore, in general, the mesh used for the minimization technique should contain at least 2(m-2)(n-2)points.

The user must also decide on the size of the weights w_1, w_2 for G. One can choose constant weights for both JT and Dot, or choose a weight function for Dot which produces more orthogonality near the boundary of the grid than in the interior. Small constant weights of values between 1 and 10 can be used initially to determine how they affect the smoothness and orthogonality of the grid.

Changing coefficient α_{ij} (or β_{ij}) changes the value of the mapping T only on the support of the tensor product B-spline B_{ij} . Therefore, in order to locate the minimum of G in the direction represented by α_{ij} one need only consider the sum over those terms in G which contain the value of JT or Dot at mesh points (ξ , n) lying on the support of B_{ij} . Subroutine CORANGE determines the range of summation associated with each tensor product coefficient for a given mesh on the square, and function GF computes the sum over the range indicated by CORANGE. Figure 8 shows the support of a tensor product B-spline associated with





knot sequences $s = \{s_i\}_{i=1}^{m+4}$ and $t = \{t_j\}_{j=1}^{n+4}$. The shaded section represents the support of tensor product B-spline B_{6,5}. In order to minimize in the direction of coefficient $\alpha_{6,5}$ it would be sufficient to look at the sum

$$GF = \frac{6}{2} \sum_{i=3}^{7} w_{1} (JT_{i+1,j} - JT_{ij})^{2} \frac{\Delta n}{\Delta t} + \frac{6}{2} \sum_{i=4}^{7} w_{1} (JT_{i,j+1} - JT_{ij})^{2} \frac{\Delta E}{\Delta n}$$

i=3 j=3 $\Delta E_{i=4} j=2$

+ 6 7

$$\Sigma \Sigma W_2(\text{Dot}_{ij})^2 \Delta \xi \Delta \eta$$
.
 $i=4 j=3$

Like G, the partial sum, GF, will be a quartic polynomial in each coefficient.

All of this information is used by the minimization routine FFMIN. Each call to FFMIN produces one complete iteration of the cyclic coordinate method. For each coefficient, the routine first determines the interval on which the coefficient must lie if JT is to be nonnegative at most of the mesh points affected by the coefficient. Then it calls eitner TESTMINØ, TESTMINL, TESTMINR, or TESTMINB depending on whether the interval is biinfinite, has a left endpoint, a right endpoint, or two endpoints. The chosen subroutine finds the location of the minimum of GF on the interval and changes the value of the appropriate coefficient accordingly.

4.5 Distribution Functions

If solutions of partial differential equations on a domain are to be accurate, the grid on the domain must be concentrated in areas of rapid change such as boundary layers and shocks. In most cases concentration near the boundary of the domain can be easily accomplished through the use of distribution functions.

Rearranging the points on the square mesh changes the distribution of grid points on the physical domain. A nonuniform distribution of points on the square mesh can be viewed as the image of functions $\varphi_1:I_1 \rightarrow I_1$, and $\varphi_2:I_1 \rightarrow \overline{I}_1$ defined on ξ and η , respectively. The grid is then generated by the mapping T defined by

 $\tilde{T}(\xi,\eta) = To_{\varphi}(\xi,\eta)$ where $\varphi: I_2 \rightarrow I_2$ satisfies

 $\varphi(\xi,\eta) = \left[\varphi_1(\xi) \\ \varphi_2(\eta) \right].$

This is graphically illustrated in figure 9. The grid on the physical domain is the image under \tilde{T} of an equally spaced mesh on the square.

In the current version of TENTEST, the user may request one of three distributions for ξ and r_1 : uniform, exponential, cr arctangent. Selecting the uniform option produces an equally spaced distribution. The distribution function is simply the identity function on I_1 . If the exponential option is selected, TENTEST calls routine EXPONENTIAL which maps $\xi \in I_1$ into $\psi(\xi) = \frac{e^{\zeta \xi} - 1}{e^{\zeta} - 1}$



Figure 9. Obtaining a Concentration of Gridpoints

where c is a nonzero constant. If c>0, ϕ concentrates the grid lines near the line corresponding to $\zeta=0$. If c<0, ϕ concentrates the grid lines closer to the line corresponding to $\zeta=1$. The grid in figure 10a was produced with $\phi_1(\xi) = \xi$ and $\phi_2(n) = \phi(n)$. The constant c is 4. In figure 10b, $\phi_1(\xi) = \phi(\xi)$ with c=5 and $\phi_2(n) = n$. The degree of concentration increases or decreases as |c| is increased or decreased. In figure 10c, $\phi_1(\xi) = \phi(\xi)$ with c=2 and $\phi_2(n) = n$.

TENTEST calls ARCTANGENT when the user selects the arctangent distribution option. ARCTANGENT maps $\zeta \in I_1$ into

$$\gamma(\zeta) = \frac{\arctangent (2c\zeta - c)}{\arctangent (c)} - \frac{\arctangent (-c)}{\arctangent (-c)}$$

where c is a positive constant. This function concentrates grid lines near points corresponding to $\zeta = 0$ and $\zeta = 1$ simultaneously. This is shown in figure 10d with $\varphi_1(\xi) = \xi$, $\varphi_2(n) = \gamma(n)$ and c=5.

Future improvements to TENTEST might include the addition of more distribution functions and the creation of a routine which allows the user to create his own distribution function by interactively digitizing points on the unit square. The routine would then create a variation diminishing spline approximation to the points to form the distribution function.

Since the distribution functions described in this section are defined on I_1 , they can also be used to

19 A




Figure 10c. Exponential distribution on g with c= -2 Figure 10d. Arctangent distribution on n with c=5.

Figure 10. Concentrating grid points on trapezoid domain.

redistribute the knots which define the tensor product B-splines that form T. This will permit the user to concentrate more knots in areas mapped to complex portions of the physical boundary so that T produces a better boundary approximation. Presently the user can choose to keep the original distribution on the knots or choose to redistribute the knots to obtain an exponential or arctangent distribution.

5. RESULTS AND DISCUSSION

This chapter examines some of the grids produced by TENTEST. Physical domains of various shapes are illustrated. Some of the grids are for actual objects, such as an airfoil or part of the space shuttle, but most are simply grids on domains of various shapes and sizes chosen to illustrate the range of the program.

The user's chief concern is the creation of an acceptable grid on a given physical domain in the shortest amount of time possible. Since the grid will be the image of a continuous mapping on the square, the best technique is to minimize the smoothing functional by using a grid generated from a coarse mesh. Then, once the new coefficients are obtained, the user can request that the grid be plotted using a much finer mesh. This technique is illustrated in the examples which follow. Most of the examples contain at least four grids: The image under T, with its initial coefficients, of a coarse square mesh; the image of a finer mesh; the image of the coarse mesh after several iterations of the minimization procedure; and the image of a finer mesh after application of the minimization procedure. Any other grids shown are chosen to illustrate grid concentration or other points of interest. In all the examples shown, only constant weight functions were used in the smoothing functional.

÷.

The first four examples show grids on domains with common geometric shapes: a trapezoid, a quadrilateral with unequal, nonparallel sides, a triangle and a circle. Since the domains are simply connected and convex, only a few interior points are needed for the sequences s and t which determine the tensor product B-splines that compose T.

The next three examples show grids on domains which are not convex. The major concern with such grids is the overlapping of grid lines near the boundary.

The last examples deal with grids around concrute objects such as an airfoil or part of the space shuttle. The irregular boundaries of some of these grids make it necessary to use more knots to define T.

For convenience, the following notation is used in this chapter.

- N_{ξ} = number of B-splines B_i in the sequence corresponding to knot sequence s, or 4 + number of interior knots in s.
- N_{η} = number of B-splines B_j in the sequence corresponding to knot sequence t, or 4 + number of interior knots in t.
- wj = constant weight multiplied times the terms in the smoothing functional involving the Jacobian, JT, of T.
- wd = constant weight multiplied times the terms in the smoothing functional containing Dot.

 $N_{\xi} \times N_{\eta}$ will be the dimension of the tensor product spline space generated by $B_{ij} = B_i \times B_j$, $i=1,...,N_{\xi}$; $j = 1,...N_{\xi}$. cpu = central processing unit - main control section of a computer.

5.1 Convex Domains

The first three examples, which have linear boundaries, require only one interior knot for each of the knot sequences s and t. The simplicity of the domains also means that a very coarse grid can be used to minimize the smoothing functional. Four or five iterations produce good results. The circular grids in the fourth example require more interior knots.

5.1.1 Trapezoid

In this example $N_{\xi} = N_{\eta} = 5$ and wj = wd = 1. The first picture in figure 11 is the grid obtained using the initial coefficients in Section 4.3. It is the image under T of an equally spaced 5x5 mesh on the square. This is the grid on which the minimization procedure was applied. Note that the number of grid points is 25, while the number of free coefficients is given by $2(N_{\xi}-2)(N_{\eta}-2) = 18$. Figure 11b is a finer grid constructed using the same coefficients. Figure 11c shows how the initial 5x5 grid changes after five iterations of the minimization procedure. The new coefficients produce grid lines that appear to be nearly orthogonal at most grid points. The image under



Original grid refined.



Figure llc Optimized grid. Figure lld



Optimized grid refined.

Figure 11 Grids on trapezoid domain.

the new T of a 20x20 mesh is given in figure 11d. The amount of cpu time used in the optimization process was 1 minute and 25 seconds.

In figure 12 the weights wj and wd have been changed to show what effect they have in the minimization process. Figure 12a shows how the initial 5x5 grid is changed after only three iterations when wj=0 and wd=1. Orthogonality is more pronounced, but the grid spacing is no longer as smooth. In the refined grid in 12b the spacing is very skewed near the top boundary. Figure 12c shows the 5x5 grid after five iterations with wj=1 and wd=0. The spacing is smoother but the grid lines are not orthogonal. Figure 12d shows a finer grid.

5.1.2 Quadrilateral with Unequal Sides

Again, in this example $N_{\xi} = N_{\eta} = 5$ which means sequences s and t each contain one interior knot. Also, wj=wd=1. The minimization procedure was applied on the 5x5 grid shown in figure 13a. Five iterations of the technique produced the grid in 13c. Figures 13b and 13d show refined versions of the grids in 13a and 13b, respectively. The five iterations of the minimization procedure required 2 minutes and 16 seconds of cpu time. In figure 14 the optimized grids are concentrated near different parts of the boundary. In 14a an exponential distribution with parameter c=4 has been put on n. Figure 14b shows an exponential distribution on ξ and n with c=4 in each







71

1

÷

. **.**



.



Figure 13a Original grid

Figure 13b Original grid refined



Figure 13c Optimized grid



Figure 13d Optimized grid refined

Figure 13 Grids on quadrilateral with unequal sides.

ORIGINAL PAGE 13 OF POOR QUALITY



Figure 14a Exponential distribution of η



Figure 14b Exponential distribution on g and η.



Figure 14c Arctangent distribution on g

Figure 14d Arctangent distribution on n

Figure 14 Concentrating gridpoints on quadrilateral.

case. In figures 14c and 14d, an arctangent distribution with c=5 has been placed on ξ and n, respectively.

5.1.3 Triangle

In the previous examples, it was clear that each side of the unit square should be mapped to a side of the four-sided physical domain, but in the case of a triangle, which has three sides, this cannot be done. The boundary must be divided into four sections. The simplest thing to do is to divide one of the sides of the triangle into two parts so that two sides of the unit square are mapped onto one side of the triangle as shown in figure 15. Figure 16a shows the initial 5x5 grid constructed with Ng = Nn = 5 and wj=wd=1. Figure 16b shows a 20x20 grid constructed using the same coefficients. After five iterations of the minimization procedure, the initial 5x5 grid is transformed into figure 16c. Figure 16d shows a finer grid. Optimization required 2 minutes and 22 seconds of cpu time.

5.1.4 Circle

Variation diminishing splines reproduce straight lines exactly, but the same cannot be said about their approximation of nonlinear curves. For such curves the accuracy of the approximation depends on the number of knots used to define the spline function. For this reason more knots are needed to obtain a satisfactory mapping of the unit square onto a circular physical domain. For the







Figure 16a Original grid

Figure 16b Original grid refined







Figure 16c Optimized grid Figure 16d Optimized grid refined

Figure 16 Grids on triangular domain.

grids shown in figure 17, $N_{\xi} = N_{\eta} = 9$ and wj=wd=1. Hence, there are five interior knots in both sequence s and sequence t.

Note that $2(N_{\xi}-2)(N_{\eta}-2) = 98$. Although this number indicates that a mesh of at least 98 points should be used for the minimization routine, the 8x8 grid shown in figure 17a seems to produce an acceptable grid. One reason for this might be that the initial grid in 17a already appears to be quite smooth and orthogonal at most points. The major problems with orthogonality occur near the areas to which the corners of the square are mapped. These areas are indicated by the arrows in 17a. Figure 18a shows how the initial grid is changed after fifteen iterations of the minimization procedure. Figures 17b and 18b show finer grids. The fifteen iterations of the minimization procedure required 20 minutes and 14 seconds of cpu time.

5.2 Nonconvex Domains

The grids in this section show some of the difficulties in creating grids on domains which are not convex sets.

5.2.1 Nonconvex Quadrilateral

Figure 19 shows the shape of the domain. This example was first mentioned in Section 2.5. The boundary of the unit square is mapped onto the boundary of the domain as indicated in figure 2. Example 2.5-1 shows that T will not map the square homeomorphically onto the domain even



Figure 17a Original grid



Figure 17b Original grid refined

Figure 17 Grids on circular domain.







Figure 18b Optimized grid refined

Figure 18 Grids on circular domain after optimization.



Figure 19 Nonconvex quadrilateral.

after the coefficients are changed. This fact is supported by the negative Jacobian present at one of the corners of the square. The negative sign suggests that points near that corner will be mapped outside of the physical domain. This is confirmed by the grids illustrated. In this example, $N\xi = N\eta = 5$ and wj=wd=1. The 5x5 initial grid shown in figure 20a was used for the minimization procedure. The enlarged picture in figure 20b shows a finer grid. Figure 21a shows the result of four iterations of the minimization procedure. The nonnegative Jacobian requirement pulls the grid lines into the interior of the domain. However, figure 22 shows an enlarged version of the corner which indicates that part of the grid still overlaps the boundary. This means that the minimization routine was unable to restrict all of the coefficients to intervals where the Jacobian of T is nonnegative.

This is further indicated in figure 21b which shows a finer version of the grid in figure 21a. The four iterations of the minimization procedure required 1 minute and 1 second of cpu time.

5.2.2 Puzzle Pieces

The next two domains, illustrated in figure 23, look like pieces from a puzzle. In each case N ξ = 19, N η = 5, wj=l and wd=10.

Grids on the first domain are shown in figures 24 and 25. The minimization procedure was performed on the

¥. \$







Figure 20b Original grid refined

Figure 20. Grids on nonconvex quadrilateral domain.



Figure 21a Optimized grid



Figure 21b Optimized grid refined

Figure 21 Optimized grids on nonconvex quadrilateral domain.



Figure 22 Enlarged corner of optimized grid.

22x8 grid in figure 24a. Figure 24b shows a finer grid. Figure 24c shows the grid obtained after forty iterations and figure 24d shows a finer grid. The grids in figure 25 show how the initial grid changes after two, five and fifteen iterations. The grid obtained after forty iterations is shown again for comparison. On this domain Tentest is able to pull all of the grid lines into the interior of the domain.

Grids on the second domain are shown in figure 26. The initial 22x8 grid is shown in figure 26a and figure 26b shows a finer grid. After forty iterations, the initial grid is transformed into 26c and a finer grid is shown in 26d. Figure 27a shows a grid on the first domain concentrated near the bottom boundary by using an exponential distribution on η with c=4. Figure 27b shows a grid on the second domain concentrated near the top by using an exponential distribution on η with c=-4.

The forty iterations used for the first domain required 1 hour, 42 minutes and 23 seconds of cpu time, but the second domain required 2 hours, 4 minutes and 46 seconds for forty iterations.

5.3 Grids for Specific Objects

This section deals with grids about particular objects such as an airfoil. The boundaries often have peculiarities which make it difficult to obtain satisfactory grids. In many cases it may be difficult to maintain smoothness in



Figure 23 Puzzle shaped domains.

ORIGINAL PAGE 13 OF POOR QUALITY





Figure 24a Original grid Figure 24b Original grid refined





Figure 24c Optimized grid Figure 24d

Optimized grid refined

Figure 24 Grids on first puzzle shaped domain.



Figure 25a After two iterations

Figure 25b After five iterations





Figure 25c After fifteen iterations

Figure 25d After forty iterations

Figure 25 Grids obtained after various iterations.

ORIGINAL PACE 13 OF POOR QUALITY



Figure 26a Original grid Figure 26b



Figure 26b Original grid refined



Figure 26c Optimized grid Figure 26d



Figure 26d Optimized grid refined

Figure 26 Grids on second puzzle shaped domain.



Figure 27 Exponential distributions on puzzle shaped grids.

$$(-2)$$

the grid while increasing orthogonality. Often the user must try to find an acceptable balance. He must also attempt to concentrate the grids in areas where rapid changes are likely to occur when partial differential equations are solved on the domain.

5.3.1 Airfoil

The grids in this example are for the Kármán -Trefftz airfoil. The parameters $N_{\xi} = 19$, $N_{\eta} = 9$, wj=l and wd=.5. Hence, there are 15 knots in the s sequence and 5 knots in the t sequence. Figure 28 shows how the domain can be viewed as having a boundary consisting of four parts. The minimization procedure was performed on the 21x12 grid in figure 29a. The grid lines appear to be orthogonal everywhere except near boundaries 1, 2 and 4. Note the sharp corners behind the airfoil. After one iteration the corners have been eliminated and the angles of the lines near the airfoil are not as acute. This is shown in figure 29b and in the finer grid in figure 30a.

Solutions on a grid about an airfoil are usually more accurate if a higher concentration of points is placed near the airfoil boundary since this is the area most affected as air moves over the airfoil. Figure 30b shows a 30x30 grid concentrated near the airfoil boundary by using an exponential distribution on n with constant c=4.

The minimization procedure required 6 minutes and 36 seconds of cpu time.



Figure 28 Domain around Kármán - Trefftz airfoil.



Figure 29a Original grid

±1⁸

3



Figure 29b Optimized grid

Figure 29 Grids for Kármán - Trefftz airfoil.







Figure 30b Optimized grid concentrated near airfoil boundary. Figure 30 Optimized grids for Kármán - Trefftz airfoil.

5.3.2 Spike-Nosed Body

According to [Sm, p. 130], the spike-nosed configuration occurs frequently in supersonic flow. R. E. Smith states that supersonic flow about such bodies is unsteady, with separation occurring near the nose-shoulder region. Therefore, the grids must be concentrated in that area [Sm, p. 48]. The boundary data for the grids shown in this section can be found in [Sm, p. 60]. Rotating the bottom boundary around a horizontal axis of symmetry produces a clearer picture of the actual body. The ratio of the length of the nose to the height of the shoulder is 2.14.

As in the previous example, $N_{\xi} = 19$, $N_{\eta} = 9$, wj=1 and wd=0.5. The 21x12 initial grid in figure 31a was used for the minimization procedure. Two iterations produce a small amount of orthogonality near the bottom boundary as shown in figure 31b. Additional iterations produce an undesirable wiggle in the grid lines near the shoulder. Figure 32a shows a finer grid and figure 32b shows a grid concentrated near the bottom boundary by using an exponential distribution on η with c=4.

The two iterations of the minimization procedure required 13 minutes and 1 second of cpu time.

5.3.3 Shuttle

The grids in this example are for a model of the space shuttle. The optimized grids are the result of ten iterations on the 32x12 grid shown in figure 33. Para-



Figure 31b Optimized grid

Figure 31 Grids for spike-nosed body.



- Figure 32b Optimized grid concentrated near boundary of spike-nose body.
- Figure 32 Optimized grids for spike-nosed body.

meters $N_{\xi} = 29$, $N_{\eta} = 9$, wj=wd=1. Ten iterations of the minimization procedure produce a small amount of orthogonality near the boundary of the shuttle as shown in figure 34. Figure 35 shows an optimized 32x20 grid concentrated near the shuttle boundary using an exponential distribution on n with c=4. The ten iterations of the minimization procedure required 1 hour and 43 minutes of cpu time. The 32x12 grid in figure 33 is the largest grid on which the minimization procedure has been applied.

ORIGINAL PAGE IS DE POOR QUALITY



Figure 33 Original grid for shuttle.


Figure 34 Optimized grid for shuttle.





6. CONCLUSIONS

This paper has examined an effective algebraic method for creating boundary fitted coordinate systems. The method, which involves a mapping T composed of tensor product B-splines allows one to regulate grid characteristics by adjusting the coefficients of the splines. Modifying the coefficients so that they minimize a smoothing functional enhances the smoothness and orthogonality of the grids generated by T.

The method is implemented in the program TENTEST which gives the user control over the number and concentration of grid points. The user can also regulate the amount of smoothness and orthogonality in the grids by the selection of weight functions for the smoothing functional.

Suggestions for future revisions of TENTEST include the addition of more distribution functions to allow greater control over grid concentration. One might also investigate the possibility of adjusting the boundary coefficients during the optimization process so that the boundary points of the grid are affected by the minimization procedure.

Ultimately, the true test for a grid comes when it is actually used to solve partial differential equations. Therefore, the next stage of research must include solving problems on several grids produced by TENTEST. Then it may be possible to change the program into an adaptive technique which rearranges the grid points in response to gradient information from the evolving solution.

Once these things are accomplished, one may attempt to use the technique to generate grids on more complicated multiconnected domains. This may involve the study of techniques for grid patching.

Also, the Prime 750 computer is an excellent machine for graphics, but not very fast in solving problems involving a large amount of computations. Hence, the possibility of creating a version of TENTEST which operates efficiently on a vector computer such as the VPS 32 at NASA Langley Research Center should be investigated. This will permit the user to run much larger and more complicated problems.

Finally, once this grid generation technique has been thoroughly developed for two dimensional domains, a three dimensional technique can be attempted. T would become a mapping from the unit cube to the desired physical domain, composed of the tensor product of B-splines in the three coordinate directions. As in the two dimensional case, characteristics of the grid would be changed by changing the coefficients of the tensor product B-splines.

BIBLIOGRAPHY

- [B] Bazaraa, Mokhtar and Shetty, C. M. Nonlinear Programming: Theory and Algorithms. New York: John Wiley and Sons, 1979.
- [de B] De Boor, Carl. <u>A Practical Guide to Splines</u>. New York: Springer-Verlag, 1978.
- [C] Curry, H. B. and Schoenberg, I, J. "On Spline Distributions and Their Limits: The Polya Distribution Functions," Abstract 380t, Bulletin of the American Mathematical Society, Vol. 53, 1947, p. 109.
- [D] Dugundji, James. <u>Topology</u>. Boston: Allyn and Bacon, Inc., 1968.
- [F] Fleming, Wendell. Functions of Several Variables. New York: Springer-Verlag, 1977.
- [G] Goldhaber, Jacob K. and Ehrlich, Gertrude. <u>Algebra</u>. London: The MacMillan Company, 1970.
- [H] Heinz, Erhard. "An Elementary Analytic Theory of the Degree of Mapping in n-Dimensional Space." Journal of Mathematics and Mechanics, Vol. 8, No. 2, 1959, p. 231.
- [0] Ortega, J. M. and Rheinboldt, W. C. <u>Iterative</u> <u>Solution of Nonlinear Equations in Several</u> <u>Variables</u>. New York: Academic Press, 1970.
- [S] Schwartz, J. T. <u>Nonlinear Functional Analysis</u>. New York: Courant Institute of Mathematical Sciences, New York University, 1965.
- [Sm] Smith, R. E. "Two-boundary Grid Generation for the Solution of the Three-dimensional Compressible Navier-Stokes Equations," NASA TM 83123, May 1981.
- [St] Starius, G. "Constructing Orthogonal Curvilinear Meshes by Solving Initial Value Problems." Numerische Mathematik, Vol. 28, 1977, p. 25.

- [SS] Smith, Philip W. and Sritharan, S. S. "On the Grid Generation Methods by Harmonic Mapping on Plane and Curved Surfaces." ICASE Report 84-12, 1984.
- [TWM] Thompson, J. E., Warsi, Z. U. A. and Mastin, C. W. "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review," Journal of Computational Physics, Vol. 47, No. 1, July 1982, p. 1.

.

•

APPENDIX

	00001104	***************************************
	00002:	PROGRAM TENTEST
	00003:C	
•	000041C	
	00005:C	TENTEST MAPS A SQUARE GRID (0,1)X(0,1) ONTO
	00006:C	A PHYSICAL DOMAIN OF ARBITRARY SHAPE THROUGH THE USE OF
	00007:C	TENSOR PRODUCT B-SPLINES. THE ORIGINAL KNOT SEQUENCES
	00008:0	MAY BE CHOSEN TO HAVE AN EQUALLY SPACED DISTRIBUTION,
	00009:C	EXPONENTIAL DISTRIBUTION, OR ARCTANGENT DISTRIBUTION.
	00010:C	SIMILAR CHOICES CAN BE MADE FOR THE DISTRIBUTION OF
	00011:C	GRIDPOINTS ON THE SQUARE.
	00012:C	TENTEST CONSTRUCTS AN INITIAL GRID GENERATION MAPPING
	00013:C	CONSISTING OF A LINEAR COMBINATION OF TENSOR PRODUCT
	00014:C	B-SPLINES WITH THE COEFFICIENTS CHOSEN SO THAT THE MAPPING
	00015:C	YIELDS A VARIATION DIMINISHING SPLINE APPROXIMATION
	00016:C	TO THE TRANSFINITE BILINEAR INTERPOLANT OF A
	00017:C	FUNCTION WHICH MAPS THE BOUNDARY OF THE UNIT SQUARE
	00018:C	ONTO THE BOUNDARY OF THE PHYSICAL DOMAIN.
	00019:C	IF THE USER REQUESTS A NEW GRID, TENTEST REARRANGES
	00020:C	THE COEFFICIENTS IN AN ATTEMPT TO MINIMIZE A FUNCTIONAL
	00021:C	G INVOLVING THE DIFFERENCE IN THE JACOBIAN OF THE GRID
	00022:C	GENERATION MAPPING AT ADJACENT MESH POINTS AND THE DOT
	00023:C	PRODUCT OF VECTORS TANGENT TO THE GRID LINES ON THE
	00024:C	PHYSICAL DOMAIN.
	00025:C	
	00026:C	
	00027:C	ROUTINES
	00028:C	
	00029:C	EXPONENTIAL
	00030:C	ARCTANGENT
	00031:C	FIXKNOTS
	00032:C	BOUNCOEF
	00033:C	INNERCOEF
	00034:C	CONSPLINE
	00035:C	TENVALF
	00036:0	I ENSUKVAL
	00037:0	JACUB
	000381C	CURANGE
	0003910	
	0004010	r F MIN
	00041:0	
	0004210	IEDINING
	0004316	IEDINIAK Tectmini
	0004416	IEJINIRL Tectmind
	0004410	1631U140
	0004710	EVTDENER UDDIL
	0004/iL	
	VVV771L	·

000511C THE FOLLOWING SUBROUTINES ARE ALSO REQUIRED. 00052:C THEY MAY BE FOUND IN "A PRACTICAL GUIDE TO SPLINES" 00053:C BY CARL DE BOOR, SPRINGER-VERLAG, 1978. 00054:C 000551C BSPLVB... CONPUTES THE VALUE OF ALL POSSIBLE 000561C NONZERO B-SPLINES OF A GIVEN ORDER AT 000571C A GIVEN POINT. 000581C 00059:C BSPLVD.... COMPUTES THE VALUE AND DERIVATIVES OF ALL B-SPLINES WHICH DO NOT VANISH AT 000401C 00061:C A GIVEN POINT 000621C 00063:C INTERV... DETERMINES THE KNOT INTERVAL ON WHICH A 00064:C GIVEN POINT LIES. ITS OUTPUT IS THE SUBSCRIPT IDENTIFYING THE KNOT WHICH IS 000651C 01:0000 INMEDIATELY LEFT OF THE POINT. 00067:C 00068:C BVALUE... CALCULATES THE JDERIV-TH DERIVATIVE 000691C OF A SPLINE FUNCTION WHOSE COEFFICIENTS 000701C ARE STORED IN ARRAY BCOEF. THE VALUE OF 00071:C JDERIV IS PROVIDED BY THE USER. 00072:C 00073:C 000741C 00075:C TENTEST USES ROUTINES FROM A PLOTIO GRAPHICS 00076:C PACKAGE TO PLOT THE GRIDS. 000771C 00078:C 00079:C VARIABLES 00080:C 00081:C NKNOTX, NKNOTY 00082:C AND 00083:C NEWNOTX, NEWNOTY... DIMENSIONS FOR SQUARE MESH. 00084:C NX,NY... DIMENSION OF SPLINE SPACE IN X 00085:C DIRECTION, Y DIRECTION. 00086:C KX... QUANTITY OF NUMBERS TO BE ADDED TO THE FRONT 00087:C AND BACK OF THE INTERIOR X KNOT SEQUENCE. 00088:C ORDER OF B-SPLINES IN X DIRECTION. 00089:C KY ... QUANTITY OF NUMBERS TO BE ADDED TO THE FRONT AND BACK OF THE INTERIOR Y KNOT SEQUENCE. 00090:C 00091:C ORDER OF B-SPLINES IN Y DIRECTION. 00092:C FNX,FNY... NUMBERS TO BE PLACED AT THE FRONT OF 00093:C THE X AND Y KNOT SEQUENCES, RESPECTIVELY. 00094:C BNX, BNY... NUMBERS TO BE PLACED AT THE BACK OF THE 00095:C X AND Y KNOT SEQUENCES. RESPECTIVELY. 00096:C INX, INY... DIMENSIONS OF INTERIOR X AND Y KNOT SEQUENCES, 000971C RESPECTIVELY. 00098:C INTX.INTY.. INTERIOR X KNOT SEQUENCE, INTERIOR Y KNOT SEQ. 00099:C X KNOT SEQUENCE, Y KNOT SEQUENCE. TX.TY... ARRAYS CONTAINING COEFFICIENTS OF OO100:C ALPHA.BETA... 00101:C TENSOR PRODUCT SPLINE MAPPING.

Ĩ

t

0010210	A.B		
0010310	AND	1	
0010410	AP.	PP	APPAVE CONTAINING COOPDINATES FOR
0010510	THE P		
0010410	1 66	TY FETY	ADBAVE TRENTTEVING KNOT INTEDIALS
	LEF	INTELLION	nnnig igentiftiing nngt intenvneg
001071			UN WHICH SUUNKE MESH CUUKUINATES LIE.
00108:C			(LEFTX(I)=J IMPLIES TX(J)<=A(I) <tx(j+1))< td=""></tx(j+1))<>
00109:C	W1,	WZ	WEIGHTS FOR JACUBIAN, DOT PRODUCT TO
00110:C			BE USED IN SMOOTHING FUNCTIONAL.
00111:C	X, Y	• • •	TWO-DIMENSIONAL ARRAYS CONTAINING COORDINATES
00112:C			OF GRID POINTS TO BE PLOTTED.
00113:C	KOU	NTE	VARIABLE USED TO COUNT ITERATIONS OF
00114:C			MINIMIZATION PROCEDURE, OR
00115:C			NUMBER OF CALLS TO ROUTINE FFMIN.
0011610			
00117:0			
0011910			TA UALERTE CAUNTERS
0011010	רו ה	ATE: HNV	100E
0012010	D.	TIEN JULI	1790
0012010			
00121.0			
00122:L			
001231C			
00124:C4	****	**********	***********************************
00125:C			
00126:C			
00127:		COMMON/COEF	/ALPHA(100,100),BETA(100,100)
00128:		COMMON/KNOT	S/TX(100),TY(100)
00129:		COMMON/PARA	M/FKOUNT
00130:		COMMON/PARA	M2/A(100).B(100).NX.NY.KX.KY
00131:	*	-LEETX(100)	-1 FETY(100)
00132:	-	COMMON/KNOT	/NKNOTY NKNOTY
001371		COMMON/UETG	
001331			NEC/VEDITNE/EA 100 2) VEDITNE/EA 100 2)
00134.			RED/ADFLINE(JV)IVV)2/)IDFLINE(JV)IVV)2/
00133.			E/1F1R51(100),1LH51(100),JF1R51(100)
00136;	#	,JLH31(100)	
00137:		REAL X(100,	100), BCUEF(100), INTX(100), INTY(100)
00138:	*	,Y(100,100)),AP(100),BP(100)
00139:		CHARACTER#1	O NAME
00140:		INTEGER#2 S	TRING(28)
00141:		INTEGER#2 N	UMB,DATE(3)
00142:		INTEGER#2 T	IME, TIME1, TIME2
00143:		EQUIVALENCE	(STRING(1),DATE)
00144:		EQUIVALENCE	(STRING(4),TIME)
00145:		EQUIVALENCE	(STRING(5).TIME1)
001441		EQUIVALENCE	(STRING(7).TIME2)
001471		NUMB=28	
0014R!		CALL TIMBAT	(STRING.NUMR)
001401		UDITE/1_111	N RATE
001501	111	CODMAT/7471	
001511	***	UDITE(1.202)	N TIME.TIME1.TIME2
VVIJI-		WD116\19444	/ *************************************
VV1221	222 -	- ruknni(16,1 (D + 10 /

001531 PI=3.14159 00154: OPEN(12.FILE='TENSORDAT') OPEN(13.FILE='NEWDATA') 001551 OPEN(14,FILE='ORGRID') 00156: 001571 OPEN(16,FILE='ORIG2') 00158: ¥1=0 00159: W2=0 KOUNTE=0 001601 PRINT*, 'INPUT NKNOTX.NKNOTY' 00161: READ(1, *) NKNOTX, NKNOTY 00162: 00163: NSAVEX=NKNOTX 001641 NSAVEY=NKNOTY 00165: PRINT*, WHAT IS KX' 001661 READ(1,*) KX PRINT .. WHAT IS KY' 00167: READ(1,*) KY 00168: READ(12,*) FNX, BNX, FNY, BNY 00169: READ(12,*) INX, INY 00170: 001711 READ(12.1) (INTX(I).I=1.INX) 00172; READ(12,*) (INTY(I),I=1,INY) 00173: PRINT*, 'DISTRIBUTION FOR X KNOTS' PRINT*, '1=EQUALLY SPACED, 2=EXPONENTIAL, 3=ARCTANGENT' 00174: 00175: READ(1.*) KODEX 00176: IF(KODEX-2) 5,10,15 10 CALL EXPONENTIAL (INTX, INX, 2.) 00177: GOTO 5 00178: 00179: 15 CALL ARCTANGENT(INTX, INX, 5.) PRINT*, 'DISTRIBUTION FOR Y KNOTS' 00180: 5 PRINT*, '1=EQUALLY SPACED, 2=EXPONENTIAL, 3=ARCTANGENT' 00181: READ(1.*) KODEY 00182: 00193: IF(KODEY-2) 16,18,19 18 CALL EXPONENTIAL (INTY, INY, 2.) 00184: GO TO 16 00185: CALL ARCTANGENT(INTY, INY, 5.) 19 00186: 00187: 16 CONTINUE 00188: NX=INX+KX 00189: NY=INY+KY CALL FIXKNOTS(KX,KY,FNX,FNY,BNX,BNY, 001901 # INX, INY, INTX, INTY) 00191: CALL BOUNCOEF(KX,KY,NX,NY) 00192: CALL INNERCOEF(KX,KY,NX,NY) 00193: WRITE(16, *) ((ALPHA(I, J), J=1, NY), I=1, NX) 00174: WRITE(16,*) ((BETA(1,J),J=1,NY),I=1,NX) 00195: DO 50 I=1.NKNOTX 00196: A(I)=FLOAT(I-1)/(NKNOTX-1) 001971 DO 20 J=1,NKNOTY 00198; 00199: B(J)=FLOAT(J-1)/(NKNOTY-1) IF(A(I).GE.1.0) A(I)=.99999 00200: 00201: IF(B(J).GE.1.0) B(J)=.99999 00202: 20 CONTINUE 00203: 50 CONTINUE

14 . Jog

ŧ

ġ.,

00204:		PRINT*, 'DISTRIBUTION FOR COMPUTATIONAL X'
00205:		PRINT#, '1=EQUALLY SPACED, 2=EXPONENTIAL, 3=ARCTANGENT'
002061		READ(1,#) KODEXX
00207:	•••	IF(KODEXX-2) 22,24,26
00208:	24	CALL EXPONENTIAL(A, NKNUTX, 2,)
002091	•••	GOTO 22
00210:	26	CALL ARCTANGENT(A, NKNUTX, 5.)
00211:	22	CONTINUE
00212:		PRINT&, DISTRIBUTION FOR COMPUTATIONAL Y
00213:		PRINT#, 'I=EQUALLY SPACED, 2=EXPUNENTIAL, 3=ARCTANGENT'
002147		KEAD(1,#) KUDETT
00215:		IF (KUDEYY-2) 30, 32, 34
00216;	32	CALL EXPONENTIAL (B, NKNOTY, 2.)
002171		GOTO 30
00218:	- 34	CALL ARCTANGENT(B,NKNOTY,5.)
00219:	30	CALL CONSPLINE
00220:		CALL CORANGE (NKNOTX, NKNOTY)
00221:		DO 70 I=1,NKNOTX
00222:		DO 60 J=1,NKNDTY
002231		CALL TENVALF(ALPHA,LEFTX(I),LEFTY(J),KX,KY,I,J,
00224:	*	X(I,J),0,0)
00225:		CALL TENVALF(BETA,LEFTX(I),LEFTY(J),KX,KY,I,J
00226:	*	,Y(I,J),0,0)
00227:	60	CONTINUE
00228:	70	CONTINUE
00229:		PRINT#, 'JACOBIAN YES(1) OR NO(0)'
00230:		READ(1,*) JCODE
00231:		IF(JCODE.EQ.1) GOTO 700
002321		PRINT#, COMPUTE DERIVATIVES YES(1) OR NO(0)
00233:		READ(1,*) KCODE
00234:		IF(KCODE.E0.0) GOTO BO
00235:		PRINT*, 'INPUT DERIVATIVES DESIRED FOR X, Y DIRECT'
00236:		READ(1,*) JDX, JDY
00237:		PRINT*, ' X Y X COMP DERIV Y COMP DE
00238:		DO 600 II=1, NKNOTX
00239:		DO 500 JJ=1,NKNOTY
00240:	_	CALL TENVALF(ALPHA,LEFTX(II),LEFTY(JJ),KX,K",II,JJ,
00241;	*	XD, JDX, JDY)
00242:		CALL TENVALF(BETA,LEFTX(II),LEFTY(JJ),KX,KY,II,JJ,
00243:	#	YD, JDX, JDY)
00244;		PRINT#,A(II),B(JJ),XD,YD
00245:	500	CONTINUE
00246:	600	CONTINUE
00247:		GOTO 80
00248:	700	CALL JACOB(NX,NY,KX,KY,A,B)
00249:	80	CONTINUE
00250:		CALL EXTREMES(X,Y,TMAX,TMIN,NKNOTX,NKNOTY)
00251:		CALL NORH(X.Y.THAX.THIN.NKNOTX.NKNOTY)

- 00251:00252: PAUSE
- 00253: NN=2 00254:

•

.

4

.

00255: WRITE(13, *) NAME 002561 WRITE(13,#)NKNOTX,NKNOTY,NN,NN 002571 WRITE(13, *) (X(I.1), I=1, NKNOTX) 002581 WRITE(13, *) (Y(I,1), I=1, NKNOTY) 00259: WRITE(13, *) (X(I, NKNOTY), I=1, NKNOTX) 002601 WRITE(13, *) (Y(1, NKNOTY), I=1, NKNOTX) WRITE(13,*) X(1,1),X(1,NKNOTY) 002611 00262: WRITE(13, *) Y(1, 1), Y(1, NKNOTY) 002631 WRITE(13, *) X(NKNOTX, 1), X(NKNOTX, NKNOTY) 00264: WRITE(13,*) Y(NKNOTX.1),Y(NKNOTX.NKNOTY) 00265: CALL INITT(960) 00266: CALL TWINDO(0,760,0,760) 00267: CALL DWINDO(-.07,1.07,-.07,1.07) 00268: 338 DO 200 I=1,NKNOTX 00269: CALL MOVEA(X(I,1),Y(I,1)) 002701 DO 100 J=1.NKNOTY 00271: CALL DRAWA(X(I,J),Y(I,J)) 00272: 100 CONTINUE 00273: 200 CONTINUE 00274: DO 400 J=1, NKNOTY 00275: CALL HOVEA(X(1,J),Y(1,J)) 00276: DO 300 I=1.NKNOTX 00277: CALL DRAWA(X(I,J),Y(I,J)) 00278: 300 CONTINUE 00279: 400 CONTINUE 002801 WRITE(14,*) NKNOTY,NKNOTX 00281: WRITE(14, *) ((X(I, J), I=1, NKNOTX), J=1, NKNOTY) 00282: WRITE(14, *) ((Y(I, J), I=1, NKNOTX), J=1, NKNOTY) 00283: CALL MOVABS(0,760) 00284: CALL ANMODE 00285; PRINT*, 'ITERATION', KOUNTE 00286: PRINT*, NX=',NX,' NY=',NY 00287: IF(KOUNTE.EQ.0) GOTO 410 00288: PRINT*, 'JACOBIAN WEIGHT=', W1 00289: PRINT*, 'ORTHOG WEIGHT=';W2 00290: PRINT*, 'OPTIMIZED ON', NSAVEX, ' BY'.NSAVEY.' GRID' 00291: 410 KOUNTE=KOUNTE+1 00292: PRINT#, 'DO YOU WANT TO CHANGE THE GRID, YES OR NO(0)' 00293; READ(1,*) KODE 002941 IF(KODE.EQ.0) GOTO 339 00295: PRINT*, 'CURRENT WEIGHTS ARE', W1, W2 002961 PRINT*, 'NEW WEIGHTS, YES(1) OR NO(0)' 00297: READ(1,*) KW 00298: IF(KW.EQ.0) GOTO 401 00299: PRINT*, 'INPUT WEIGHTS FOR JACOB, ORTHOG' 00300: READ(1,*) W1,W2 00301: 401 CONTINUE 00302: NKNOTX=NSAVEX 00303; NKNOTY=NSAVEY 00304: CALL FFMIN(ERMAX) 00305: PRINT*, 'OUTPUT JACOBIAN, YES(1) OR NO(0)'

*.

1902001 READ(1.*) JKODE 003071 IF(JKODE.EQ.0) GOTO 399 003081 CALL JACOB(NX,NY,KX,KY,A,B) 00309: 399 CONTINUE PRINT#, 'CHANGE NUM OF GRIDPOINTS, YES(1) OR NO(0)' 00310: 00311: READ(1,*) KODE 00312: IF(KODE.EQ.0) GOTO 501 00313: PRINT#, 'ENTER NUMBER OF GRIDPOINTS FOR X DIRECTION, 00314: *** Y DIRECTION'** 00315: READ(1,*) NEWNOTX, NEWNOTY 00316: NKNOTX=NEUNOTX 00317: NKNOTY=NEWNOTY 00318: GOTO 502 00329: 501 CONTINUE 003201 NEWNOTX=NKNOTX 00321: NEWNOTY=NKNOTY 00322: 502 CONTINUE 00323: CALL ERASE 00324: DO 450 I=1,NEWNOTX 00325: DO 425 J=1, NEWNOTY 00326: AP(I)=FLOAT(I-1)/(NEWNOTX-1) 00327: BP(J)=FLOAT(J-1)/(NEWNOTY-1) 00328: IF(AP(I).GE.1.0) AP(I)=.99999 60329: IF(BP(J).GE.1.0) BP(J)=.9999 00330: 425 CONTINUE 00331: 450 CONTINUE PRINT*, 'DISTRIBUTION FOR COMPUTATIONAL X' 00332: 00333: PRINT#, '1=EQUALLY SPACED, 2=EXPONENTIAL, 3=ARCTAN' 00334: READ(1,*) KODE3X 00335: IF(KODE3X-2) 452,454,456 00336: 454 CALL EXPONENTIAL (AP, NEWNOTX, 2.) 00337: GOTO 452 00338: 456 CALL ARCTANGENT(AP, NEWNOTX, 5.) 00339: 452 CONTINUE PRINT*, 'DISTRIBUTION FOR COMPUTATIONAL Y' 00340: READ(1,*) KODE3Y 00341: 00342: IF(KODE3Y-2) 458,460,462 00343: 460 CALL EXPONENTIAL (BP, NEWNOTY, 2.) 00344: GOTO 458 00345: 462 CALL ARCTANGENT (BP, NEWNOTY, 5.) 00346: 458 CONTINUE 00347: DO 480 I=1,NEWNOTX 00348: DO 470 J=1, NEWNOTY 00349: CALL TENSORVAL (ALPHA, NX, NY, KX, KY, AP(I), 00350: ***** BP(J),X(I,J),0,0) 00351: CALL TENSORVAL(BETA, NX, NY, KX, KY, AP(I), BP(J), 00352: ***** Y(I,J),0,0) 00353: 470 CONTINUE 00354: 480 CONTINUE CALL EXTREME3(X,Y,TMAX,TMIN,NEWNOTX,NEWNOTY) 00355: 00356: CALL NORH(X,Y,TMAX,THIN,NEWNOTX,NEWNOTY)

00357: 00358: 00359: 00340: 00341: 00342: 00343: 00344: 00345: 00345: 00346: 00346: 00347: 00348: 00349: 00370:	PAUSE GOTO 338 339 CONTINUE CALL FINITT(0,760) PRINTS,'KX IS ',KX PRINTS,'KY IS ',KY PRINTS,'DISTRIBUTIONS' PRINTS,'I=EQUALLY SPACED,2=EXPONENTIAL,3=ARCTANGENT' PRINTS,'I=EQUALLY SPACED,2=EXPONENTIAL,3=ARCTANGENT' PRINTS,'I=EQUALLY SPACED,2=EXPONENTIAL,3=ARCTANGENT' PRINTS,'I=EQUALLY SPACED,2=EXPONENTIAL,3=ARCTANGENT' PRINTS,'I=EQUALLY SPACED,2=EXPONENTIAL,3=ARCTANGENT' PRINTS,'I=EQUALLY SPACED,2=EXPONENTIAL,3=ARCTANGENT' PRINTS,'I=EQUALLY SPACED,2=EXPONENTIAL,3=ARCTANGENT' PRINTS,'I=EQUALLY SPACED,2=EXPONENTIAL,3=ARCTANGENT' PRINTS,'KNOT DIST. IS',KODEX PRINTS,'Y KNOT DIST. IS',KODEY PRINTS,'COMPUTATIONAL Y DIST IS',KODEXX PRINTS,'COMPUTATIONAL Y DIST IS',KODEYY CLOSE(12) CLOSE(13)
003717	
003721	CALL TIMUNT(STRING, NUNB)
003731	WRITE(1,222) (INE, (INE) (INE)
003741	STOP
00376:C 00377:C	
00378: 00379:C	SUBROUTINE EXPONENTIAL(X,N,AC)
00380:C	THIS ROUTINE PRODUCES AN EXPONENTIAL DISTRIBUTION OF
00381:C	POINTS BY SUBSTITUTING AN ORIGINAL SET OF NUMBERS
00382:C	U LYING BETWEEN O AND 1 INTO THE EXPONENTIAL
00383:C	FUNCTION (EXP(A#U)-1.)/(EXP(AC)-1) WHERE AC IS A
00384:C	PARAMETER WHOSE VALUE IS SUPPLIED BY THE USER.
00385:C	
00386:C	VARIABLES
00387:C	
00388:C	XTHIS AN ARRAY WHICH ON INPUT CONTAINS THE ORIGINAL
00389:C	SET OF NUMBERS AND ON OUTPUT CONTAINS THE EXPONENTIA
00390:C	DISTRIBUTION OF NUMBERS.
00391:C	NSIZE OF ARRAY X
00392:C	ACPARAMETER IN EXPONENTIAL FUNCTION
00393:C	
00394:	REAL X(100)
00395:	DO 10 I=1,N
00396:	U=X(I)
00397:	X(I)=(EXP(AC#U)-1.)/(EXP(AC)-1.)
00398:	IF (X(I).GE.1.0) X(I)=.99999
00399:	10 CONTINUE
00400:	RETURN
00401:	END
00402:C	
00403:C	
00404:C	
00405:	SUBROUTINE ARCTANGENT(X,N,AC)

004041C 004071C THIS ROUTINE PRODUCES AN ARCTANGENT DISTRIBUTION 004081C OF POINTS BY SUBSTITUTING AN ORIGINAL SET OF NUMBERS 004091C U LYING BETWEEN O AND 1 INTO THE ARCTANGENT FUNCTION 004101C (ATAN(AC)-ATAN(-ATAN(-AC))/(ATAN(AC)-ATAN(-AC)) 004111C WHERE AC IS A PARAMETER WHOSE VALUE IS 004121C SUPPLIED BY THE USER. 004131C 00414:C VARIABLES 004151C X ... THIS AN ARRAY WHICH ON INPUT CONTAINS THE ORIGINAL SET 00414:C OF NUMBERS AND ON OUTPUT CONTAINS THE ARCTANGENT DISTRIBUT 004171C 00418:C OF NUMBERS. 004191C N...SIZE OF ARRAY X 004201C AC.. PARAMETER IN ARCTANGENT FUNCTION 004211C 004221C 004231 REAL X(100) 00424: DO 10 I=1.N U=X(I) 004251 004261 X(1)=(ATAN(2.*AC*U-AC)-ATAN(-AC))004271 # /(ATAN(AC)-ATAN(-AC)) IF(X(I).GE.1.0) X(I)=.99999 00428: 004291 10 CONTINUE 00430: RETURN 00431: END 004321C 004331C 004341C 00435: SUBROUTINE FIXKNOTS(KX,KY,FNX,FNY,BNX,BNY,INX, 00436: INY, INTX, INTY) 004371C THIS ROUTINE PLACES KX COPIES OF FNX AT THE 00438:C 00439:C BEGINNING OF THE INTERIOR X KNOT SEQUENCE, 00440:C KY COPIES OF FNY AT THE BEGINNING OF THE 00441:C INTERIOR Y KNOT SEQUENCE, KX COPIES OF BNX AT THE 00442:C END OF THE INTERIOR X KNOT SEQUENCE AND KY COPIES 00443:C OF BNY AT THE END OF THE INTERIOR Y KNOT SEQUENCE. 004441C 004451C INPUT 004461C 004471C 004481C 00449:C KX ... QUANTITY OF NUMBERS TO BE ADDED TO THE FRONT AND BACK OF THE INTERIOR X KNOT SEQUENCE. 004501C 00451:C ORDER OF B-SPLINES IN X DIRECTION. QUANTITY OF NUMBERS TO BE ADDED TO THE FRONT 00452:C KY... 004531C AND BACK OF THE INTERIOR Y KNOT SEQUENCE. ORDER OF B-SPLINES IN Y DIRECTION. 00454:C 00455:C FNX, FNY.... NUMBERS TO BE PLACED AT THE FRONT OF THE X AND Y KNOT SEQUENCES, RESPECTIVELY. 00456:C

004571C BNX, BNY NUMBERS TO BE PLACED AT THE BACK OF THE 004581C X AND Y KNOT SEQUENCES, RESPECTIVELY. 0045910 INX, INY... DIMENSIONS OF INTERIOR X AND Y KNOT SEQUENCES. 004601C RESPECTIVELY. 004611C INTERIOR X KNOT SEQUENCE, INTERIOR Y KNCT SEQ. INTX.INTY... 004421C 004431C 004641C OUTPUT(IN COMMON) 004651C 004661C TX.TY ... X KNOT SEQUENCE, Y KNOT SEQUENCE 004471 004681C 004691C 004701 COMMON/KNOTS/TX(100), TY(100) 004711 REAL INTX(1), INTY(1) 004721 NX=INX+KX 00473: NY=INY+KY 004741 DO 100 I=KX+1.NX 004751 J=I-KX 004761 TX(I)=INTX(J) 00477: 100 CONTINUE 00478: DO 200 I=KY+1.NY 004791 J=I-KY 00480: TY(I)=INTY(J) 00481: 200 CONTINUE 004821 DO 5 I=1.KX 004831 TX(I)=FNX 004841 INDEX=I+NX 00485: TX(INDEX)=BNX 5 CONTINUE 004861 00487: DO 6 I=1,KY 00488: TY(I)=FNY 004891 INDEX=I+NY 004901 TY(INDEX)=BNY 00491: 6 CONTINUE 004921 RETURN 00493: END 004941C 004951C 00496:C 004971 SUBROUTINE BOUNCOEF (KX.KY.NX.NY) 004981C 004991C THIS ROUTINE COMPUTES THE BOUNDARY COEFFICIENTS 00500:C FOR TWO TENSOR PRODUCT B-SPLINES (X.Y COMPONENTS). 00501:C SPECIFICALLY, IT COMPUTES ALPHA(I,1), BETA(I,1) AND 005021C ALPHA(I,NY), BETA(I,NY) FOR I=1 TO NX; 005031C ALPHA(1,J), BETA(1,J) AND ALPHA(NX,J), BETA(NX,J) 00504:C FOR J=1 TO NY. COEFFICIENTS ARE CHOSEN SO THAT THERE IS A 00505:C 005061C VARIATION DIMINISHING APPROXIMATION ALONG THE 005071C BOUNDARY.

.

•

UNDAR!F		
00509:C	1	NPUT is deeted for the second s
00510:C	с. 1941 г.	
00511:C	KX,	KYORDER OF SPLINE IN X DIRECTION, Y DIRECTION
00512:C	NX,	NYDIMENSION OF SPLINE IN X DIRECT, Y DIRECT.
00513:C		
00514:C	BOT	H COEFFICIENT SEQUENCES WILL HAVE DIMENSION
005151C	NX	NY
00514:0		•
00517:C		
00518:0		NITPUT
00519:0		
00520:0	(TN	CONNON)
00521:0	ROU	INDARY COFFETCIENTS PLACED IN ALPHA.BETA ARRAYS.
0052210		
005231		CONMON/COFE/ALPHA(100.100).BETA(100.100)
005241		CONNON/COLT/THETHICLOOFICO/FEETICLOOFICO/
005251		TIMENCIAN TYCTAD/ 100) TYCTAD/ 100)
00323+		DINEWSION (VS(UV(IVO)))(2(UV(IVO)
VVJ201		
00327.		
		62X(1)=3.11
00329:		021(1)=2+#1
00530:		637(1)=4.*1
00531:		G3Y(T)=2.
00532:		G4X(T)=1T
00533:		G4Y(T)=2.*T
00534:		PI=3.14159
00535:		DO 100 I=1,NX
00536:		SUM=0.
00537;		. DO 50 J=1,KX-1
00538:		SUM=SUM+TX(I+J)
00539:	50	CONTINUE
00540:		TXSTAR(I)=SUM/(KX-1)
00541:	100	CONTINUE
00542:		DO 200 J=1,NY
00543:		SUM=0.
00544:		DO 150 K=1,KY-1
00545:		SUN=SUH+TY(J+K)
00546:	150	CONTINUE
00547:		TYSTAR(J)=SUM/(KY-1)
00548:	200	CONTINUE
00549:		00 300 I=1.NX
00550:		A=TXSTAR(I)
00551:		ALPHA(I.1)=G1X(A)
00552:		BETA(I.1)=GIY(A)
00553:		ALPHA(I.NY)=G3X(A)
005541		BETA(I.NY)=G3Y(A)
00555:	300	CONTINUE
00556:		DD 400 J=1.NY
005571		B=TYSTAR(J)
005581		$AIPHA(NX_{*}I) = G2X(R)$
		1981-101107797, APU/P.(

005591 BETA(NX,J)=G2Y(B) 005401 ALPHA(1, J)=G4X(B) 00561: BETA(1.J)=84Y(B) 005621 400 CONTINUE 005631 RETURN 005641 END 005451C 005661C 005671C 005681 SUBROUTINE INNERCOEF(KX,KY,NX,NY) 005691C 00570:C THIS ROUTINE COMPUTES THE INNER COEFFICIENTS 00571:C FOR TWO TENSOR PRODUCT B-SPLINES (X,Y COMPONENTS) 00572:C SPECIFICALLY, IT COMPUTES ALPHA(I, J), BETA(I, J) FOR 00573:C I=2 TO NX-1, J=2 TO NY-1. 00574:C THE COEFFICIENTS ARE COMPUTED THROUGH THE USE OF 005751C TRANSFINITE BILINEAR INTERPOLATION. THE INTERPOLANTS ARE EVALUATED AT POINTS SO THAT THE 00576:C 00577:C RESULTING COEFFICIENTS PRODUCE A VARIATION 00578:C DININISHING SPLINE APPROXIMATION TO THE TRANSFINITE BILINEAR INTERPOLANT. 00579:C 00580:C 00581:C INPUT 00582:C KX, KY... ORDER OF B-SPLINES IN X DIRECTION, Y DIRECTION 00583:6 00584:C NX,NY...DIMENSION OF SPLINE SPACE IN X DIRECT, Y DIRECT 005851C BOTH COEFFICIENT SEQUENCES WILL HAVE DIMENSION 00586:C 00587:C NX ZNY 00588:C TX, TY(IN COMMON)...KNOT SEQUENCE FOR X DIRECT, Y 00589:C DIRECTION 005901C 00591:C 00592:C OUTPUT(IN COMMON) 00593:C INTERIOR COEFFICIENTS PLACED IN ALPHA, BETA ARRAYS 00594:C 005951C 00596:C COMMON/COEF/ALPHA(100,100), BETA(100,100) 005971 00598: COMMON/KNOTS/TX(100), TY(100) 00599: DIMENSION TXSTAR(100), TYSTAR(100) 00600: G1X(T)=2.*T+1. 00601: G1Y(T)=0. 00602: G2X(T)=3.+T00603: G2Y(T)=2.#T 00604: G3X(T) = 4.1T006051 G3Y(T)=2. 00606: G4X(T)=1.-T. 00607: G4Y(T)=2,*T 00608: FX(X,Y)=G1X(X)*(1,-Y)+G3X(X)*Y 00609: * +G2X(Y)*X+(1.-X)*G4X(Y)

118

营养会

- 7

	A REVIEWARD VIEWARD VIEWA
00610:	X-G1X(Q)X(1X)X(1Y)-G2X(U)XXX(1Y) *_G1Y(A)*(1Y)*Y_G2Y(1.)*Y*Y
004121	6-00A\V/+\1+-X/+ -U2A\1+/+A+ EV/Y_V\+C1V/V\+/1V\4RTY/Y\#Y
004171	<pre>FILA\$[/=011(X)*(1)=1/7001(X/#) # LC2Y(V)\$Y1(1,=Y)\$CAY(V)</pre>
VV013+	#
VV014+	\$ "DIT(V)#(I;"/)#V_COV(1 \#V#V
00613;	
00616;	F1=3.14137
0061/;	
00618:	SUM=0.
00619:	10 30 J#1,KX-1
00620:	SUN=SUN+IX(1+J)
00621:	50 CUNTINUE
00622:	TXSTAR(I)=SUM/(KX+1)
00623:	100 CUNTINUE
00624:	DO 200 J=1,NY
00625:	SUM=0.
00626:	DO 150 K=1,KY-1
00627:	SUN=SUN+TY(J+K)
00 628:	150 CONTINUE
00629:	TYSTAR(J)=SUM/(KY-1)
00630:	200 CONTINUE
00631;	DO 400 I=2,NX-1
00632:	DO 300 J=2,NY-1
00633:	A=TXSTAR(I)
00634:	B=TYSTAR(J)
00635:	ALPHA(I,J)=FX(A,B)
00636:	BETA(I.J)=FY(A.B)
00637:	300 CONTINUE
00638:	400 CONTINUE
00639:	RETURN
00640:	END
00641:C	
0064210	
0064310	
00644:	SUBROUTINE COMSPLINE
00645:0	
0044410	THIS ROUTINE COMPLITES AND STORES THE FUNCTION
0064710	UALIES AND ETRST DERIVATIVES DE ALL THE NON-
0044910	HANTCHING BLODI THES AT FACH DOTHT OF A COHADE
0044010	AUNIDUTIO D_DEFITIED UL EUCU LOTILI OL U DROUVE
00047.0	THE ADDITION IT DETERMINED THE MART INTERNAL
	IN ADDITION, IT DETERMINES THE NUT INTERVAL
00651;0	UN WHICH EACH RESH COURDINATE LIES.
00652:0	9.12.17
0065310	INPUT
0065410	
0065510	AKKATS CUNIAINING COUKUINATES FOR SQUARE
0065610	MESH, FOINTS OF EVALUATION FOR B-SPLINES.
00657:0	NKNOTX, NKNOTY NUMBER OF ELEMENTS IN A, B.
00658:0	KX,KY, ORDER OF B-SPLINES IN X DIRECTION, Y
00659:C	DIRECTION.
00660:C	TX,TY X KNOT SEQUENCE FOR B-SPLINES,Y KNOT

00661:C	•	SEQUENCE FOR B-SPLINES.
00662:C	NX,NY	DINENSION OF SPLINE SPACE IN X DIRECTION,
00663:C		Y DIRECTION
00664:C		
00665:C	OUTPUT	
00666:C	(IN COMMON)	
006671C		
3:8660	XSPLINE.YSPLINE.	.THREE DIMENSIONAL ARRAYS CONTAINING
00669:C		FUNCTION VALUES AND FIRST DERIVATIVES OF
00670:C		B-SPLINES IN X DIRECTION, Y DIRECTION AT
00671:C		EACH ELEMENT OF A, B. THE FIRST SUBSCRIPT
00672:C		IDENTIFIES THE B-SPLINE, THE SECOND
00673:C		SUBSCRIPT REPRESENTS THE POINT OF EVALUATION
00674:C		AND THE THIRD SUBSCRIPT (1 OR 2) INDICATES
00675:C		WHETHER THE VALUE REPRESENTS A FUNCTION
00676:C		EVALUATION OR DERIVATIVE EVALUATION. HENCE
00677:C		XSPLINE(3,2,1) WILL CONTAIN THE VALUE OF THE
00678:C		B-SPLINE IN THE X DIRECTION. B(3). AT A(2).
00679:C		
00680:C	LEFTX.LEFTY	ARRAYS IDENTIFYING KNOT INTERVALS ON
00681 C		WHICH MESH COORDINATES LIE. LEFTX(3)=4 WOULD
00682:C		MEAN THAT A(3) LIES BETWEEN TX(4) AND TX(4+1)
00683:C		
00684:C	REQUIRED ROUTINES:	
00685:C	BSPLVD	
00686:0	BSPLVB	
00687:C	INTERV	
00688:C		
00689:	COMMON/PARAM2/	A(100),B(100),NX,NY,KX,KY,LEFTX(100)
00690:	<pre>* ,LEFTY(100)</pre>	
00691:	COMMON/SPLINES	/XSPLINE(50,100,2),YSPLINE(50,100,2)
00692:	COMMON/KNOTS/T	X(100),TY(100)
00693:	COMMON/KNOT/NK	NOTX,NKNOTY
00694:	REAL DBIATX(4,	2),WORK(4,4)
00695:	IDERIV=2	
00696:	JDERIV=2	· · · · · · · · · · · · · · · · · · ·
00697:C		
00698:C	INITIALIZATION	
00699:C		
00700:	NUMX=NX+KX	
00701:	NUMY=NY+KY	
00702:	DO 3 I=1,NX	
00703:	DO 2 J=1,NKNOT	X
00704:	DO 1 KK=1,2	
00705:	XSPLINE (I, J, KK	()= (,
00706:	1 CONTINUE	
00707:	2 CONTINUE	
00708:	3 CONTINUE	
00709:	DO 9 I=1,MY	
00710:	DO 8 J=1,NKNOT	Y
00711:	DO 7 KK=1,2	

.

.

00712: YSPLINE(I, J, KK)=0. 7 CONTINUE 00713: 007141 8 CONTINUE 00715: **9 CONTINUE** DO 25 I=1.NKNOTX 00716: CALL INTERV(TX, NUMX, A(I), LEFTX(I), MFLAG) 00717: 00718: CALL BSPLVD(TX,KX,A(I),LEFTX(I),WORK,DBIATX,IDERIV) 00719: DO 252 JJ=1.2 00720: DO 251 II=1.KX 00721: IB=LEFTX(I)-KX+II 00722: IF(IB.LE.0) GOTO 251 XSPLINE(IB,I,JJ)=DBIATX(II,JJ) 00723: 30724: 251 CONTINUE 00725: 252 CONTINUE 00726: 25 CONTINUE 00727: PRINT#, 'DO ALL X SED EQUAL ALL Y SEG YES(1) OR NO(0)' READ(1,*) KODE 00728: 00729: IF(KODE.EQ.0) GOTO 28 00730: DO 27 JJ=1,NKNOTY 00731: LEFTY(JJ)=LEFTX(JJ) 00732: DO 272 KK=1,2 00733: DO 271 II=1.NY 00734: YSPLINE(II, JJ, KK) = XSPLINE(II, JJ, KK) 00735: 271 CONTINUE 00736: 272 CONTINUE 00737: 27 CONTINUE 00738: GOTO 293 00739: 28 CONTINUE 00740: DO 29 J=1,NKNOTY 00741: CALL INTERV(TY,NUMY,B(J),LEFTY(J),MFLAG) 00742: CALL BSPLVD(TY,KY,B(J),LEFTY(J),WORK,DBIATX,JDERIV) 00743: DO 292 JJ=1.2 DO 291 II=1,KY 00744: 00745: JB=LEFTY(J)-KY+II 00746: IF(JB.LE.0) GOTO 291 00747: YSPLINE(JB, J, JJ)=DBIATX(II, JJ) 00748: 291 CONTINUE 00749: 292 CONTINUE 00750: 29 CONTINUE 00751: 293 CONTINUE 00752: RETURN 00753: END 007541C 00755:C 00756:C SUBROUTINE TENVALF(ARR,LEFTX,LEFTY,KX,KY 00757: 00758: * ,I,J,VALUE,IDERIV,JDERIV) 00759:C 00760:C TENVALF COMPUTES PARTIAL DERIVATIVES FOR A TENSOR 00761:C PRODUCT SPLINE FUNCTION AS INDICATED BY THE PARAMETERS

1.2111年7月夏夏夏夏夏夏月月月月月月月

法遵守性

00762:C IDERIV, JDERIV.

120

00763:C 00764:C INPUT 00765:C ARR... 00766:C ARRAY OF COEFFICIENTS FOR SPLINE 00767:C FUNCTION 00768:C INDICIES FOR POINT OF EVALUATION I.J... 00769:C (A(I), B(J)), WHERE A, B ARE ARRAYS 007701C WHICH CONTAIN COORDINATES OF A 00771:C SQUARE MESH. 00772:C LEFTX... VALUE INDICATING KNOT INTERVAL 00773:C ON WHICH A(I) LIES VALUE INDICATING KNOT INTERVAL 00774:C LEFTY 00775:C ON WHICH B(J) LIES 00776:C KX,KY... ORDER OF B-SPLINES IN X-DIRECTION, 00777:C **Y DIRECTION** 00778:C IDERIV... ORDER OF DERIVATIVE DESIRED FOR X 00779:C DIRECTION 00780:C JDERIV... ORDER OF DERIVATIVE DESIRED FOR Y 00781:C DIRECTION 00782:C (IN COMMON) 00783:C XSPLIME, YSPLINE, ARRAYS CONTAINING FUNCTION 00784:C VALUES AND FIRST DERIVATIVES OF B-SPLINES IN 00785:C X DIRECTION, Y DIRECTION AT EACH POINT 00786:C GIVEN IN ARRAYS A.B 00787:C 00788:C OUTPUT 00789:C 00790:C VALUE ... VALUE OF TENSOR PRODUCT SPLINE 00791:C OR DERIVATIVE 007921C 00793: COMMON/SPLINES/XSPLINE(50,100,2), YSPLINE(50,100,2) 00794: DIMENSION ARR(100,100) 00795: VALUE=0. 00796: DO 13 JJ=1,KY 00797: JB=LEFTY-KY+JJ 00798; IF(JB.LE.0) GOTO 13 DO 12 II=1,KX 00799: 00800: IB=LEFTX-KX+II 00801: IF(IB.LE.0) GOTO 12 00802: VALUE=VALUE+ARR(IB, JB) #XSPLINE(IB, I, IDERIV+1) 00803: # #YSPLINE(JB,J,JDERIV+1) 00804: 12 CONTINUE 00805: 13 CONTINUE 00806: RETURN 008071 END 00808:C 00809:C SUBROUTINE TENSORVAL(ARR, NX, NY, KX, KY, A, B, VALUE 00810: * ,JDX,JDY) 00811: THIS ROUTINE CALCULATES THE VALUE OF A TENSOR PRODUCT 00812:C 00813:C SPLINE AT THE POINT (A,B).

1.549

121

008141C 00815:C INPUT 00814:C ARR... ARRAY OF COEFFICIENTS 00817:C OOG18:C NX, NY...DIMENSION OF SPLINE SPACE IN X DIRECT., Y DIRECTION. ARRAY ARR WILL HAVE 00819:C 00820:C DIMENSION NXXNY. 00821:C KX, KY... ORDER OF B-SPLINES IN X DIRECT, Y DIRECT 00822:C A,B... POINT OF EVALUATION 00823:C 00824:C (IN COMMON) 00825:C TX, TY... KNOT SEQUENCE FOR X DIRECT, 008261C Y DIRECT. 008271C 008281C 00829:C OUTPUT 00830:C 00831:C VALUE...VALUE OF TENSOR PRODUCT SPLINE AT (A,B) 00832: COMMON/KNOTS/TX(100), TY(100) 00833: DIMENSION BCDEF(100), ARR(100,100) 00834: CALL INTERV(TY,NY,B,LEFTY,MFLAG) 00835: VALUE=0. 00836: DO 10 J=1,KY 00837: 10 BCDEF(J)=BVALUE(TX,ARR(1,LEFTY-KY+J),NX,KX,A,JDX) 00838: VALUE=BVALUE(TY(LEFTY-KY+1), BCOEF,KY,KY,B,JDY) 00839: RETURN 00840: END 00841:C 00842:C 00843:C 00844: SUBROUTINE JACOB(NX,NY,KX,KY,A,B) 00845:C 00846:C THIS ROUTINE COMPUTES THE JACOBIAN OF A 00847:C TENSOR PRODUCT B-SPLINE MAPPING. 00848:C 00849:C INPUT 00850:C 00851:C NX.NY... DIMENSION OF SPLINE IN X 00852:C DIRECTION, Y DIRECTION 00853:C ORDER OF B-SPLINES IN X DIRECTION KX,KY... 00854:C Y DIRECTION 00855:0 ARRAYS CONTAINING EVALUATION A.B... 00854:C POINTS 008571C (IN COMMON) 00858:C ALPHA, BETA, ... COEFFICIENTS OF B-SPLINES 00859:C NKNOTX, NKNOTY, NUMBER OF ELEMENTS IN 00860:C ARRAYS A.B 00861:C 00862:C JUTPUT(TO TERMINAL) 008631C ARRAYS CONTAINING EVALUATION 00864:C A.B...

00865:C		POINTS		
008661C	AJCOBIAN	JACOBIAN	AT EACH POIN	T
00867:C				
00868;	COMMON/COEF/AL	PHA(100,1	00), BETA(100	,100)
00869:	COMMON/KNOT/NK	NOTX, NKNO	TY	
00870:	REAL A(100),B(100)		
00871:	PRINT#,	X	Y	JACOBIAN'
00872:	PRINT*, '			
00873:	DO 20 I=1.NKNO	TX		
00874:	DO 10 J=1.NKNO	TY		
00875:	II=I			
00876:	ل=زز			
00877:	CALL TENSORVAL	(ALPHA.NX	NY.KX.KY.AC	II).B(JJ).
00878:	* XDFIRST.1.0)			
008791	CALL TENSORVAL	(ALPHA.NX	NY.KX.KY.AC	II).B(_().
00880:	# YDFIRST.0.1)			
00881:	CALL TENSORVAL	(BETA.NX.	NY . KX . KY . A (I)	I).B(JJ).XDSEC.
00882:	¥ 1.0)			•
00883:	CALL TENSORVAL	(BETA.NX.	NY . KX . KY . A (I	I).B(JJ).YDSEC.
00884:	* 0.1)			
00885:	AJCOBIAN=XDFIR	STXYDSEC-		т
00886:	PRINTE.A(II).B	OJLA. (LL)	BIAN	•
00887:	10 CONTINUE			•
00888:	20 CONTINUE			
00889:	RETURN			
00890:	END			
00891:0				
0089210				
0089310				
0007010	SUBBOUTTNE COB		TY_NKNOTY)	
0089510	SOPHODI THE COM			
0089410	CORANGE DETERMI	NES THE B	ANGES DE SUM	MATTON
00897:0	NEEDED TO MINIMIZE	THE SHOO	THING FUNCTI	NAL G TN
0089810	FACH COORDINATE DI	RECTION.		
00899:0	SPECIFICALLY.	FOR FACH	COFFETCIENT	ALPHA(T.I).
0000000	OR RETA(T. I). IT D	ETERMINES	THE PANCE OF	ב זאחזרבץ בחפ
00901:0	THE MESH POINTS I Y	TNG ON TH	F SUPPORT OF	THE TENSOR
00902:0	PRODUCT B-SPI INF H	AUTNG SUR	SCRIPTS T.J.	IT PLACES
0090310	THE QUALLET TH TE	TOQT (T) A	NR HETPET(1)	AND THE
0090410	LARGEST INDICES IN	TIAGT(T)	AND HACT/I	THERE UALLES
0090510	RETERMINE WITCH TE	DMC TN C	CHAIN D DE CHI	MACH LUCN
0000410	MINIMITING THE SMO	NHU IN U	UCTION IN TH	F RIDECTION
0090310	DEDDECENTED BY THE	COSSETCT	ENT ALDUATT.	IN OR RETACT. IN.
0070710	NETNESENTED DI THE	GUEFFICI		J/ UN DEIN(193/1
14000+L	TNOIT			
4474746 0091010	117F U 1			
0091110	NKNOTY NKNOTY	TITMENETON		MESH
AA0121C		UD NUMBED	OF FIFMENTE	TN ARRAYC A.P
0001210	(TH COMMON)		AL EFENCHIO	
0091410	A.R.		NTAINING COO	RDINATES FOR
1001E+0	ПТВТТТ	COLLADE ME		TERNILLE IVIN
VV71J+L		SAAUUUUE UE	JII .	

.

.

٠

•

00916:C	NX,NY	DIMENSION OF SPLINE SPACE IN X DIRECT.
0091736		TATAL WE OF 5 ON THESE IN A STREET OF
0071810		TUTAL NUE DESPLINES IN X DIRECTION,
00919:0		T DIRECTION
0092010	KX = K T + + +	URDER OF S-SPLINES IN X DIRECTION,
00921:C		Y DIRECTION
00922;C	LEFTX,LEFTY	ARRAYS IDENTIFYING KNOT INTERVALS
00923:C		ON WHICH SQUARE MESH COORDINATES
009241C		LIE. (LEFTX(I)=J IMPLIES
00925:C		TX(J)<=A(I) <tx(j+1)< td=""></tx(j+1)<>
00926:C		
00927:C	OUTPUT	
00928:C		
00929:C	IFIRST, JFIRST	ARRAYS CONTAINING STARTING
00930:C		POINTS FOR THE RANGES OF SUMMATION
00931:C		CORRESPONDING TO EACH COEFFICIENT
00932:C		ALPHA(I,J),BETA(I,J).
00933:C	ILAST, JLAST	ARRAYS CONTAINING FINAL
00934:C		PCINTS FOR THE RANGES OF SUMMATION
00935:C		CORRESPONDING TO EACH COEFFICIENT
00936:C		ALPHA(I,J),BETA(I,J)
00937:C		· · · · · · · · · · · · · · · · · · ·
00938:	COMMON/PARAM2	/A(100),B(100),NX,NY,KX,KY,
00939:	# LEFTX(100).LEF	FTY(100)
00940:	COMMON/RANGE/	IFIRST(100).ILAST(100).JFIRST(100).
00941:	# JLAST(100)	
00942:	BO 50 T=1-NX	
00943:	IF(I,FQ,1) TH	TN
00944:	TT=1	B *1
00945:	FISE	
009441	TTETETET/T-1	
00947!	FNDIE	
00040+	IN TEXT OF LEFTY	(TT)_/KY_1) AND T (E (SETV(TT))
007401	* COTO 20	(11)-((//_1)+UMD+1+FC+FEL(//(11))
000501	4 0010 20	
00730,		
00931;		
VV7J21	20 1F1K31(1)=[] 70 TT=TT14	
007531		
007341	IF (II.GT.NKNO)	(X) GUIU 40
00955:	IF(I.LI.LEFIX)	(11) - (KX - 1)) G''(0 40)
009561	GOTO 30	
00937:	40 ILAST(I)=II-1	
00758:	50 CONTINUE	
009591	DU 100 J=1,NY	
00960:	IF(J.EQ.1) THE	EN
00961:	JJ=1	
00962;	ELSE	
00963:	JJ=JFIRST(J-1)	
00964:	ENDIF	•
00965:	60 IF(J.GE.LEFTY)	(JJ)-(KY-1).AND.J.LE.LEFTY(JJ))
00966:	* GOTO 70	

00947: JJ=JJ+1 009681 GOTO 60 009691 70 JFIRST(J)=JJ 009701 80 JJ=JJ+1 IF(JJ.GT.NKNOTY) GOTO 90 00971: 00972: IF(J.LT.LEFTY(JJ)-(KY-1)) GOTD 90 00973: GOTO 80 00974: 90 JLAST(J)=JJ-1 00975: 100 CONTINUE 00976: RETURN END 00977: 00978:C 00979:C 00980:C 00981: REAL FUNCTION GF(II, JJ) 009821C 00983:C 00984:C FUNCTION OF COMPUTES THE SUN OF THE TERMS IN 00985:C THE SMOOTHING FUNCTIONAL G OVER THE RANGES INDICATED 00986:C BY IFIRST(II), JFIRST(JJ) AND ILAST(II), JLAST(JJ). 00987:C 00988:C INPUT 00989:C 00990:C II, JJ... INDICES FOR COEFFICIENT INVOLVED 00991:C IN MINIMIZATION. 00992:C (IN COMMON) 60993:C NKNOTX, NKNOTY... DIMENSIONS FOR SQUARE MESH 00994:C OR NUMBER OF ELEMENTS IN ARRAYS A.B 00995:C ARRAYS CONTAINING COORDINATES FOR A.B... 00996:C SQUARE MESH DIMENSION OF SPLINE SPACE IN X DIRECT. 00997:C NX,NY... 00998:C ,Y DIRECTION OR 00999:C TOTAL NO. OF B-SPLINES IN X DIRECTION. 01000:C Y DIRECTION ORDER OF B-SPLINES IN X DIRECTION, 01001:C **KX**,**KY**... 01002:C Y DIRECTION LEFTX, LEFTY... ARRAYS IDENTIFYING KNOT INTERVALS 01003:C 01004:C ON WHICH SQUARE MESH COORDINATES 01005:C LIE. (LEFTX(I)=J IMPLIES 01006:C $TX(J) \leq A(I) \leq TX(J+1)$ 01007:C 01008:C IFIRST, JFIRST.. ARRAYS CONTAINING STARTING 01009:C POINTS FOR THE RANGES OF SUMMATION 01010:C CORRESPONDING TO EACH COEFFICIENT 01011:C ALPHA(I,J), BETA(I,J). 01012:C ILAST, JLAST... ARRAYS CONTAINING FINAL POINTS FOR THE RANGES OF SUMMATION -01013:C CORRESPONDING TO EACH COEFFICIENT 01014:C 01015;C ALPHA(I,J), BETA(I,J) 01016:C W1,W2... WEIGHTS FOR JACOBIAN, DOT PRODUCT 01017:C

01018:C	OUT	PUT			
01019:5					
01020:C	G	iF	PARTIAL SUM	OF TERMS IN G	OVER THE
01021 C			APPROPRIATE	RANGE .	
0102210					
01023:		REAL A 1/30.30	- DOT (70 - 70)		
010231		COMMON /DADAMO	//////////////////////////////////////		
010275		LEETY/1001 LEE		// • • • • • • • • • • • • • • • • • •	,
01023.	-				
01026;			PHA(100,100)	, BEIA(100,100	•
01027:			KNUTX NKNUTT		
01028;		CUMMUN/WEIGHT	5/W1,W2		
01029:		COMMON/RANGE/	[FIRST(100),]	(LAST(100),JFI	(RST(100),
01030:	¥	JLAST(100)			
01031:		XUMX=NKNOTX			
01032:		NUMY=NKNOTY			
01033:		DELX=1./(NUMX-	-1.)		
01034:		DELY=1./(NUMY-	-1.)		
01035:		SDELX=DELX*DEL	X		
01036:		SDELY=DELY*DEL	.Υ		
01037:		SUM=0.0			
01038;		IF=IFIRST(II)			
01039:		JF=JFIRST(JJ)			
01040:		IL=ILAST(II)			
01041:		JL=JLAST(JJ)			
01042:		IF(IF.GT.1) IF	=IF-1		
01043:		IF(JF.GT.1) JF	=JF-1		
01044:		IF(IL.LT.NUMX)	IL=IL+1		
01045:		IF(JL.LT.NUMY)	JL=JL+1		
01046:		DO 200 J=JF.JL			
01047:		10 100 I=IF.IL			
01048:		CALL TENVALE (- L:HA.LEFTX(I	.LEFTY(.J).KX	.KY.I.J.
01049:	x	F1X.1.0)			,,_,_,
01050:	-	CALL TENVALE (F	ETA. EFTX(I)	-LEETYCD.KX.	KY.T.I.
01051:	x	F1Y.1.0)			,,,,,,,,
01052:		CALL TENVALE (LPHA.LEFTX(1).LEFTY().KX	.KY.T.J.
01053:	¥	F2X-0-1)		· • • • • • • • • • • • • • • • • • • •	,,,,,,,,,
01054:	-	CALL TENUALE(E	ETAL FETY(T)	JEETY(D.KY.	KY. T. I.
01055:	×	E2Y.0.1)			,,,,,,,,,
010541		A ((T - 1)=E1Y#ET	V_67Y#61V		
01057:		TIOT(T, I)=E1Y#F	771E17#E77		
010591		CHN+CHMADOT/T.	11447		
010501	100	CONTINUE	J/##2		
010571	200				
01060.	200				
010011		5UN1-V+			
010021			_1		
010631		DU 400 J=JF;JL DU 300 J=JF;JL	1		
VIV044 A1A4E4		10 JVV 1=17;1L	.T1 1 co 4) coto	700	
01044+		IF (I+EU+I+ANU+	J.EW.17 GUIO	500 6070 700	
010001		IF (I + EU+ I + ANU+	JIEUINUMT-1)	0010 300	
V1V0/1		IF (I.EU.NUMX-]		UIU 300	
01068:		SUM1=SUM1+(AJ(I,J)-AJ(I,J+	1))##2	

.

010691 SUH2=SUH2+(AJ(I+1,J)-AJ(I,J))**2 01070: 300 CONTINUE 400 01071: CONTINUE 01072: SUM1=SUM1*DELX/DELY 01073: SUM2=SUM2*DELY/DELX GF=W1#(SUM1+SUM2)+W2#DELX#DELY#SUM 01074: 01075: RETURN 01076: END 01077:C 010781C 01079:C 01080: SUBROUTINE FFMIN(ERMAX) 01081:C FFMIN SEARCHES FOR THE MINIMUM OF THE SMOOTHING 01082:C 01083:C FUNCTIONAL G. EACH CALL TO FFMIN PRODUCES ONE COMPLETE ITERATION OF THE CYCLIC COORDINATE HETHOD. 01084:C 01085:C A MULTIDIMENSIONAL SEARCH TECHNIQUE FOR MINIMIZING 01086:C A FUNCTION OF SEVERAL VARIABLES WITHOUT USING 01087:C DERIVATIVES. THE ROUTINE SEARCHES FOR A MININUM 01088:C ALONG EACH COORDINATE DIRECTION. IN FFMIN THE COORDINATE DIRECTIONS ARE REPRE-01089:C 01090:C SENTED BY THE TENSOR PRODUCT COEFFICIENTS .FOR EACH 01091:C COEFFICIENT THE ROUTINE FIRST DETERMINES THE 01092:C INTERVAL ON WHICH THE COEFFICIENT MUST LIE IF THE 01093:C JACOBIAN OF THE TENSOR PRODUCT MAPPING IS TO BE 01094:C NONNEGATIVE AT ALL MESH POINTS AFFECTED BY THE 01095:C COEFFICIENT. IT THEN CALLS EITHER TESTMINO, TEST-010961C MINL, TESTMINR, OR TESTMINB DEPENDING ON WHETHER 01097:C THE INTERVAL IS BIINFINITE, HAS A LEFT ENDPOINT. 01098:C A RIGHT ENDPOINT, OR TWO ENDPOINTS. THE CHOSEN SUBROUTINE FINDS THE LOCATION OF THE MINIMUM OF 01099:C 01100:C GF ON THE INTERVAL AND CHANGES THE APPROPRIATE COEFFICIENT ACCORDINGLY. 01101:C 01102:C 01103:C INPUT 01104:C 01105:C (IN COMMON) 01106:C ALPHA, BETA ... ARRAYS CONTAINING COEFFICIENTS OF TENSOR PRODUCT SPLINE MAPPING. 01107:C 01108:C ARRAYS CONTAINING COORDINATES FOR A, B... SQUARE MESH. 01109:C DIMENSION OF SPLINE SPACE IN X DIRECT. 01110:C NX, NY... 01111:C .Y DIRECTION OR 01112:C TOTAL NO. OF B-SPLINES IN X DIRECTION, 01113:C Y DIRECTION. ORDER OF B-SPLINES IN X DIRECTION. 01114:C KX.KY... 01115;C Y DIRECTION. ARRAYS IDENTIFYING KNOT INTERVALS 01116:C LEFTX, LEFTY... ON WHICH SQUARE MESH COORDINATES 01117:C LIE. (LEFTX(I)=J IMPLIES 01118:C TX(J)<=A(1)<TX(J+1)) 01117:C

01120:0	
01121:C	IFIRST, JFIRST ARRAYS CUNTAINING STARTING
01122:C	POINTS FOR THE RANGES OF SUMMATION
01123:C	CORRESPONDING TO EACH COEFFICIENT
0112410	ALPHA(I,J),BETA(I,J),
01125:C	ILAST, JLAST ARRAYS CONTAINING FINAL
01126:C	POINTS FOR THE RANGES OF SUMMATION
01127:C	CORRESPONDING TO EACH COEFFICIENT
01128:C	ALPHA(I,J),BETA(I,J).
01129:C	W1,W2 WEIGHTS FOR JACOBIAN, DOT PRODUCT
01130:C	TO BE USED IN GF.
01131:C	NKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH
01132:C	OR NUMBER OF ELEMENTS IN ARRAYS A,B.
01133:C	
01134:C	OUTPUT
01135:C	
01135:C	ERMAX MAXIMUM CHANGE IN THE COEFFICIENTS
01137:C	AFTER A COMPLETE ITERATION.
01138:C	(IN COMMON)
01139:C	ALPHA, BETA ARRAYS CONTAINING NEW COEFFICIENTS
01140:C	FOR TENSOR PRODUCT SPLINE MAPPING.
01141:C	
01142:	COMMON/COEF/ALPHA(100,100),BETA(100,100)
01143:	COMMON/PARAM2/A(100), B(100), NX, NY, KX, KY,
01144:	* LEFTX(100).LEFTY(100)
01145:	COMMON/RANGE/IFIRST(100), ILAST(100), JFIRST(100),
01146:	* JLAST(100)
01147:	COMMON/PARAM/FKOUNT
01148:	COMMON/WEIGHTS/W1.W2
01149:	COMMON/KNOT/NKNOTX.NKNOTY
01150:	REAL LEND.LINT.AJ(2)
01151:	INTEGER CTEST
01152:	FKOUNT=0
01153:	ERMAX=0.
01154:	BD 500 I=2.NX-1
01155:	DQ 400 J=2.NY-1
01156:	IF=IFIRST(1)
01157:	IL=ILAST(I)
01158:	JF=JFIRST(J)
01159:	
0:160:	MKQUNT=0
011612	5 CONTINUE
01167:	HOL D=GF(T, J)
01163:	HTEST=HKOUNT/2#2
011641	TE(MTEST.EQ.MKOUNT) THEN
011651	HOCO=ALPHA(I.J)
011661	ELSE
01167:	HOCO=BETA(I.J)
01168:	ENDIF
01169:	LEND=-10.0E8
011701	REND=10.0EB

01171:		KFLAG=0
01172:	•	IFLAG=0
01173:		DO 200 11=1F.IL
01174:		
01175:		IF (MKOUNT/2#2.ED.MKOUNT) THEN
011761		AL PHA(1.J)=0.
01177:		FLSF
011781		RETA (1.1)=0.
01179!		ENDIE
01180:		
01181:		CALL TENUALE (ALPHA.) FETX(TT).) FETY(11).KY.KY.TT.
011821	*	
01123:	-	CALL TENUALE(RETALLEETY(TT),LEETY(TT),KY,KY,TT
0118A!	*	
A11951	-	CALL TENUALE(ALDHA.LEETY(TT).LEETY(TT).KY.KY.TT
V11071	•	LLEOY.A.1)
A11971	-	CALL TÊNHALE(BETA,LEETY(TT),LEETY/LE) KY KY TT
V110/1	•	THE TERMETORING CETAVILIAGE TAVILIA
V1100+	•	JJFF279V927 A 1/K _E1V4E9V_E9V#E1V
VI1071		TJ/N/7=FIA4F21=F2A4F11 15/NKOUNT/343 EO NKOUNTA TUEN
011701		
01191:		nLrHn(1,J)=1.
01192;		
01193:		SCIO(1)J/=1.
01194:	10	
011731	10	
01196;		9=nJ(1)
01197:		L#AJ(2)-U
01198;		IF(U, U), I, UE-/) IMEN
01197;		
01200:		IF (LINT, GT.LENU) THEN
01201;		IF(LINT+LE+KEND) IMEN
01202;		
01203:		ELSE
01204;		IFLAG#-1
01205:		ENUIF
01206:		ENDIF
01207:		ELSE IF(C.LT1.0E-7) THEN
01208:		KINI=-D/C
01209:		IF (RINT.LT.REND) THEN
01210:		IF (RINT.GE.LENU) THEN
01211;		RENDERINI
01212:		ELSE
01213:		IFLAG=-1
01214:		ENDIF
01215:		ENDIF
01216:		ELSE
01217:		KFLAG=KFLAG+1
01218:		ENDIF
01219:	100	CONTINUE
01220:	200	CONTINUE
01221:		CTEST=(ILAST(I)-IFIRST(I)+1)*(JLAST(J)-JFIRST(J)+1)

01222: IF(KFLAG.EQ.CTEST.OR.IFLAG.EQ.-1) THEN 01223: IF(MKOUNT/2#2.EQ.MKOUNT) THEN 01224: ALPHA(I,J)=(LEND+REND)/2. 01225: ELSE 01226: BETA(I.J)=(LEND+REND)/2. 01227: ENDIF ELSE IF(LEND.LT.-1.0E7.AND.REND.GT.1.0E7) THEN 01228: 01229: CENTER=0.0 01230: CALL TESTMINO(MKOUNT,I,J) 01231: ELSE IF(LEND.LT.-10.0E7) THEN 01232: CENTER=0.0 CALL TESTMINR(MKOUNT, I, J, LEND, REND) 01233: ELSE IF (REND.GT.10.E7) THEN 01234: 01235: CENTER=0.0 01236: CALL TESTMINL(MKOUNT, I, J, LEND, REND) 01237: ELSE 01238: CENTER=(LEND+REND)/2. 01239: CALL TESTMINB(MKOUNT, I, J, LEND, REND) ENDIF 01240: 01241: S2≠GF(I,J) 01242: DIFF=HOLD-S2 01243: IF(HOLD.LT.S2) THEN IF (MTEST.EQ.MKOUNT) THEN 01244: 01245: ALPHA(I,J)=HOCO 01246: ELSE BETA(I,J)=HOCO 01247: 01248: ENDIF 01249: S2=HOLD 01250; DIFF=0. 01251: ENDIF 01252: IF(ABS(DIFF).GT.ERMAX) ERMAX=ABS(DIFF) 01253; PRINT*, FUNCTION VALUE IS',52 01254: PRINT*, 'COUNT IS', FKOUNT 01255: KFLAG=0 01256: MKOUNT=MKOUNT+1 01257: IF(MKOUNT.NE.MKOUNT/2*2) GOTO 5 01258: 400 CONTINUE 01259; 500 CONTINUE 01260: RETURN 01261: END 01262:C 01263:0 01264:C 01265; SUBROUTINE CRIT(CENTER.C1.C2.C3.C4.C5.NROOTS.R.MKOUNT. 01266; * I,J) 01267:C 01268:C CRIT FINDS THE COEFFICIENTS OF THE 4TH DEGREE 01269:C POLYNOMIAL REPRESENTING OF AND COMPUTES ITS CRITICAL 01270:C POINTS, I.E., IT FINDS THE POINTS FOR WHICH THE DERIVATIVE 01271:C OF THE POLYNOMIAL IS 0. 01272:C



01273;C	INPUT	
01274:C		
01275:C	CENTER	NUMBER AT CENTER OF INTERVAL TO BE
01276:C		CONSIDERED. IF INTERVAL IS INFINITE THEN
01277:C		CENTER ASSIGNED A VALUE OF 0.
01278:C	NKOUNT	NKOUNT EVEN MEANS THE COEFFICIENT
01279 : C		INVOLVED IN MINIMIZATION IS IN THE ALPHA
01280:C		ARRAY. MKOUNT ODD MEANS THE COEFFICIENT
01281:C		IS IN THE BETA ARRAY.
01282:C	I.J	SUBSCRIPTS FOR COEFFICIENT INVOLVED
01283;C		IN MINIMIZATION
01284:C	(IN COMMON)	
01285:C	ALPHA.BETA	ARRAYS CONTAINING COEFFICIENTS OF
01286:C		TENSOR PRODUCT SPLINE MAPPING.
0128710	A.B	ARRAYS CONTAINING COORDINATES FOR
01288:C		SDIARE MESH.
01289:C	NX . NY	DIMENSION OF SPLINE IN X DIRECTION
0129010		Y DIRECTION OR
01291:0		TOTAL NO. OF B-SPI INES IN X DIRECTION.
0129210		Y DIRECTION.
01293:0	KX.KY	DREER OF R-SPITNES IN Y DIRECTION.
0129410	***	Y DIRECTION.
01295:0	I FFTY .I FFTY	ARRAYS TRENTTEYING KNOT INTERUALS
0127510		IN SUTCH COMADE MEGH COCODINATES
012701C		ITE. (LEETY/T)= I THELTES
012771C		TY(1)/+A(T)/TY(L1))
		1X(J)/=H(1)/(1X(J+1))
0127716	ut up	HETCHTE FOR IACORTAN FOT PRODUCT
0130010	W1;;W2;++	TO BE HEED TH CE
	NENOTY NENOTY	TU DE UJED IN UF.
0130210	NKNUIX,NKNUIT	DE NUMBER OF ELEMENTS IN ADDAMS A D
01303:0	EXOUNT	DADAMETER CONTAINING NUMBER OF CALLS IN
01304:0	P NUUN I + + +	PARAMETER CUNTAINING NUMBER OF CALLS TO
01303:6		
01306:L	XN+++	ARKAT CUNIMININ PUINIS -2,-1,V,1,2
0130/10		WHICH ARE USED AS LEST PUINTS IN
01308:C		DETERMINING THE COEFFICIENTS OF THE
01309:C		4TH DEGREE POLYNOMIAL WHICH APPROXI-
01310:C		MATES GF.
01311:C		
01312:C	OUTPUT	
01313 : C		
01314:C	C1,C2,C3,C4,C5.	, COEFFICIENTS OF 4TH DEGREE POLYNOMIAL.
01315:C		C1 IS THE COEFFICIENT OF THE 4TH DEGREE TERM
01316:C	NROOTS	NUMBER OF CRITICAL POINTS
01317:C	R	ARRAY CONTAINING CRITICAL POINTS
01318:C		
01319:	REAL R(3),BK(5)
01320:	COMMON/COEF/AL	_PHA(100,100),BETA(100,100)
01321:	COMMON/PARAM2	/A(100),B(100),NX,NY,KX,KY,LEFTX(100)
01322:	<pre>* ,LEFTY(100)</pre>	
01323:	COMMON/KNOT/NI	KNOTX,NKNOTY

01324:		COMMON/WEIGHTS/W1,W2	
01325:		CONKON/PARAM/FKOUNT	
01326:		COMMON/XKS/XK(5)	
01327:		IF(MKOUNT/2#2.EQ.MKOUNT) THEN	
01328:		DO 100 IK=1,5	
01329:		ALPHA(I,J)=XK(IK)+CENTER	
01330:		BK(IK)=GF(I,J)	
01331:		FKOUNT=FKOUNT+1	
01332:	100	CONTINUE	
01333:		ELSE	
01334:		DO 200 IK=1,5	
01335:		BETA(I,J)=XK(IK)+CENTER	
01336:		BK(IK)=GF(I,J)	
01337:		FKOUNT=FKOUNT+1	
01338:	200	CONTINUE	
01339:		ENDIF	
01340:		D=XK(4)	
01341:		B1=BK(1)	
01342:		B2=BK(2)	
01343:		B3=BK(3)	
01344:		B4=BK(4)	
01345:		B5=BK(5)	
01346:		C5=B3	
01347:		SUM=-85+8.*(84-82)+81	
.01348:		C4=1./(12.*D)*SUN	
01349:		SUM=-B5+16.*(B4+B2)-30.*B3-B1	
01350:		C3=1./(24.*D*D)*SUM	
01351:		SUN=85-2.*(84-82)-81	
01352:		C2=1./(12.*D*D*D)*SUM	
01353:		SUM=B5-4.*(B4+B2)+6.*B3+B1	
01354:		C1=1./(24.*D**4)*SUM	
01355:		IF (ABS(C1).LT.1.0E-06) GDT0 300	
01356:		CALL CUBIC(4.*C1.3.*C2.2.*C3.C4.NRODTS.R)	
01357:		RETURN	
01358:	300	CONTINUE	
01359:		NROOTS=-1	
01360:		RETURN	
01361:		END	
01362:C			
01363 : C			
01364:C			
01365:		SUBROUTINE TESTMING(MKOUNT,I,J)	
01366:C			
01367:C		FOR A GIVEN COEFFICIENT ALPHA(I,J) OR BETA(I,J)	
01368:C	TES	TMINO FINDS AND TESTS THE CRITICAL POINTS OF	
01369:C	THE	4TH DEGREE POLYNOMIAL REPRESENTING GF TO	
01370:C	DETERMINE WHICH POINT YIELDS THE SMALLEST VALUE FOR GF		
01371 : C	WHEN GF IS VIEWED AS A FUNCTION OF THAT COEFFICIENT.		
01372:C	THE NUMBER CHOSEN BECOMES THE NEW VALUE FOR		
01373:C	ALPI	HA(I,J) OR BETA(I,J) .	
01374:C			

01375:C	INPUT	
01376:C		
01377:C	MKOUNT	MKOUNT EVEN MEANS THE COEFFICIENT
01378:C		INVOLVED IN MINIMIZATION IS IN THE ALPHA
01379:C		ARRAY. MKOUNT ODD MEANS THE COEFFICIENT
01380:C		IS IN THE BETA ARRAY.
01381:C	I,J	SUBSCRIPTS FOR COEFFICIENT INVOLVED
01382:C		IN MINIMIZATION
01383:C	(IN COMMON)	
01384:C	ALPHA,BETA	ARRAYS CONTAINING COEFFICIENTS OF
01385:C	-	TENSOR PRODUCT SPLINE MAPPING.
01386:0	A.B	ARRAYS CONTAINING COORDINATES FOR
01387:0		SQUARE MESH.
01388:C	NX . NY	DIMENSION OF SPLINE IN X DIRECTION
01389:0		Y DIRECTION OR
0139010		TOTAL NO. OF R-SPLINES IN X DIRECTION.
0139110		Y DIRECTION.
0139210	KY-KY-	ORDER OF R-SPITNES IN Y DIRECTION.
0139310		Y DIRECTION.
01394:0	LEETX JEETY	ARRAYS IDENTIFYING KNOT INTERVALS
0139510		TN WHICH SOURCE MESH COORDINATES
0139410		ITE. (IFETX(I)=1 IMPLIES
0139710		TY(1) = A(T) + TY(1+1)
01377 ·C		18(3)(=1(1)(18(3)1))
01399:0	M1.M2	WEIGHTS FOR MACORTAN, DOT PRODUCT
01400:0	~	TO BE USED IN GE.
0140110	NENOTY NENOTY	TTNENSTONS FOR SOMARE MESH
0140210		OR NUMBER OF ELEMENTS IN ARRAYS A.R.
01403*C	FROUNT	PARAMETER CONTAINING NUMBER DE CALLS TO
0140410		GE
0140510	YK	ARRAY CONTAINING POINTS -21.0.1.2
0140410	~~~~	WHICH ARE USED AS TEST POINTS IN
01407*C		DETERMINING THE COEFFICIENTS OF THE
01407.0		
01408+0		MATES GE.
01407.0		
01410.0	01170117	
01411.0	001201	
0141260		
01413:0	nLrnn(1,J) un p	EIN(1,J)NEW VALUE FOR COEFFICIENT
0141416		DUA/100 100) DETA/100 100)
01415;		(A/100) B/100) BEIN(100,100)
01416;	CUMMUN/PARAM2	7A(100),8(100),8X,81,8X,81,EF(X(100))
01417:	* LEFTY(100)	
01418:	COMMON/KNOT/N	
01419:	COMMON/WEIGHT	5/W1,W2
01420:	COMMON/PARAM/	
01421:	COMMON/XKS/XK	(3)
01422;	REAL R(3)	
01423:	FM(R)=C1*R**4	+U2#K#K#K+U3#K#K+U4#K+U3
01424:	DO 50 IK=1,5	·
014251	XK(IK)=FLOAT(IK)-3.

133

.

01426:	50	CONTINUE		
01427:		CALL CRIT(0,C)	L,C2,C3,C4,C5,NROOTS,R,MKOUNT,I,J)	
01428:		IF (NROOTS.NE	-1) GOTO 55	
014291		RETURN		
01430:	55	CONTINUE		
014311		THIN=10.0F10		
014321		00 400 TP=1.NE	DOTS	
014771		ENTNN-EN(D(TD)		
014331		TELEMINN IT TH	YY NA THEN	
014341 014751		THTN-CHINNELISIC		
014334		THIN-FOIRN		
014771		ENDIE		
0143/+	400			
01430+	800	TE / MKOUNT / 747	CO MKOUNTITUEN	
014374				
014401		NLFNA(1,J)=K(1		
01441;				
01442;		BEIN(1,J)=K(1)	11N)	
01443:		ENUIP		
01444;		RETURN		
01445:		ENU		
01446:C				
01447:C				
01448:C				
01449:		SUBROUTINE TES	STMINR(MKOUNT, I, J, LEND, REND)	
01450:C				
01451:C		FOR A GIVEN C	COEFFICIENT ALPHA(I,J) OR BETA(I,J)	
01452:C	TES1	MINR FINDS AND	TESTS THE CRITICAL POINTS OF	
01453:C	THE	4TH DEGREE POL	YNOMIAL REPRESENTING GF TO	
01454;C	DETE	RMINE WHICH PO	DINT YIELDS THE SMALLEST VALUE FOR GF	
01455:C	WHEN	GF IS VIEWED	AS A FUNCTION OF THAT COEFFICIENT.	
01456:C	THE	SMALLEST VALUE	IS COMPARED WITH THE VALUE AT THE	
01457:C	RIG	IT ENDPOINT OF	THE INTERVAL (LEND, REND)	
01458:C	TO I	TO DETERNINE AT WHAT NUMBER THE MINIMUM VALUE		
01459:C	OF GF DCCURS. THE NUMBER CHOSEN BECOMES THE NEW			
01460:C	VALL	E FOR ALPHA(I,	J) OR BETA(I,J) .	
01461:C				
01462:C	INPL	IT		
01463:C				
01464:C	MH	OUNT	MKOUNT EVEN MEANS THE COEFFICIENT	
01465:C			INVOLVED IN MINIMIZATION IS IN THE ALPHA	
01466:C			ARRAY. MKOUNT ODD MEANS THE COEFFICIENT	
01467:C			IS IN THE BETA ARRAY.	
01468:C	1.	J	SUBSCRIPTS FOR COEFFICIENT INVOLVED	
01469:C	-,		IN MINIMIZATION	
01470:C	LE	ND.REND.	LEFT AND RIGHT ENDPOINTS FOR	
01471 C			INTERVAL. LEND IS A NEGATIVE NO. WITH VERY	
0147210			LARGE MAGNITUDE INDICATING THAT THE LEFT	
0147310			ENDPOINT IS INFINITE.	
0147410	0			
0147510	Al	PHA.BETA.	ARRAYS CONTAINING COEFFICIENTS OF	
0147410	,		TENSOR PRODUCT SPI THE MAPPING.	
V 1 7 / U 1 U			TRUCAL FILATORS OF REAL FILLS THE FILLS	
01478:CSQUARE MESH.01479:CNX,NYDIMENSION OF SPLINE IN X DIRECTION01480:C,Y DIRECTION OR01481:CTOTAL NO. OF B-SPLINES IN X DIRECTION01482:CY DIRECTION.01482:CY DIRECTION.01483:CKX,KY01484:CY DIRECTION.01485:CLEFTX,LEFTY01485:CLEFTX,LEFTY01485:CLEFTX,LEFTY01485:CLEFTX,LEFTY01486:CIN WHICH SOUARE MESH COORDINATES01487:CLIE. (LEFTX(I)=J IMPLIES01488:CTX(J)<=A(I) <tx(j+1))< td="">01489:COR NUMBER OF JACOBIAN, DOT PRODUCT0149:COR NUMBER OF SOUARE MESH0149:COR NUMBER OF ELEMENTS IN ARRAYS A,B.0149:CGF0149:CGF0149:CKK0149:CWHICH ARE USED AS TEST POINTS IN0149:CDETERMINING THE COEFFICIENTS OF THE0149:CMHICH ARE USED AS TEST POINTS IN0149:COUTPUT0150:COUTPUT0150:COUTPUT0150:CCDHMON/CDEF/ALPHA(100,1C0),BETA(100,100)0150:CCDHMON/CDEF/ALPHA(100,1C0),BETA(100,100)0150:CCDHMON/PARAM/FKOUNT0150:CCOMMON/PARAM/FKOUNT0150:CCOMMON/PARAM/FKOUNT</tx(j+1))<>	E MESH. SION OF SPLINE IN X DIRECTION RECTION OR NO. OF B-SPLINES IN X DIRECTION, ECTION. OF B-SPLINES IN X DIRECTION, ECTION. S IDENTIFYING KNOT INTERVALS ICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I) <tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BCTA(100,100)</tx(j+1)) 	01477:C	A.B	ARRAYS CONTAINING COORDINATES FOR
---	--	---------	--------------------	---
01479:CNX,NYDIMENSION OF SPLINE IN X DIRECTION01480:C,Y DIRECTION OR01481:CTOTAL NO. OF B-SPLINES IN X DIRECTION01482:CY DIRECTION.01483:CKX,KY01484:CY DIRECTION.01484:CY DIRECTION.01484:CY DIRECTION.01484:CY DIRECTION.01484:CY DIRECTION.01484:CY DIRECTION.01484:CY DIRECTION.01484:CIEFTX,LEFTY01485:CIEFTX,LEFTY01486:CIN WHICH SQUARE MESH COORDINATES01487:CLIE. (LEFTX(I)=J IMPLIES01488:CTX(J)<=A(I) <tx(j+1))< td="">01489:COI01490:CW1,W201490:CW1,W201490:CNKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH01490:COR NUMBER OF ELEMENTS IN ARRAYS A,B.01490:CNKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH01490:COR NUMBER OF ELEMENTS IN ARRAYS A,B.01491:COR NUMBER OF ELEMENTS IN ARRAYS A,B.01494:CFKOUNT01495:COF01495:COF01495:COF01496:CXK01497:CWHICH ARE USED AS TEST POINTS IN01498:CDETERMINING THE COEFFICIENTS OF THE01499:CATH DEGREE POLYNOMIAL WHICH APPROXI-01501:COUTPUT01503:COUTPUT01503:CCOMMON/COEF/ALPHA(100,1C0), BETA(100,100)01507:COMMON/PARAM/FKOUNT01508:COMMON/PARAM/FKOUNT<tr <td="">C</tr></tx(j+1))<>	SION OF SPLINE IN X DIRECTION RECTION OR NO. OF B-SPLINES IN X DIRECTION, ECTION. OF B-SPLINES IN X DIRECTION, ECTION. S IDENTIFYING KNOT INTERVALS ICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I) <tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BCTA(100,100)</tx(j+1)) 	01478:C		SQUARE MESH.
014801C,Y DIRECTION OR014811CTOTAL NO. OF B-SPLINES IN X DIRECTION014831CKX,KY014831CY DIRECTION.014831CKX,KY014831CY DIRECTION.014831CLEFTX,LEFTY014841CY DIRECTION.014851CLEFTX,LEFTY014851CLEFTX,LEFTY014851CIN WHICH SQUARE MESH COORDINATES014851CLIE. (LEFTX(I)=J IMPLIES014851CTX(J)<=A(I) <tx(j+1))< td="">014851CTX(J)<=A(I)<tx(j+1))< td="">014851CTX(J)<=A(I)<tx(j+1))< td="">014851CTX(J)<=A(I)<tx(j+1))< td="">014851CTX(J)<=A(I)<tx(j+1))< td="">014851CTX(J)<=A(I)<tx(j+1))< td="">014851CTX(J)<=A(I)<tx(j+1))< td="">014851CTX(J)<=A(I)<tx(j+1))< td="">014951COR NUHBER OF LEMENTS IN ARRAYS A,B.014951CGF014951CGF014951CGF014951CGF014951CGF014951CDETERMINING THE COEFFICIENTS OF THE014951CDETERMINING THE COEFFICIENTS OF THE014951CDETERMINING THE COEFFICIENTS OF THE014951COUTPUT01501CMATES GF.015021COUTPUT015031CCOMHON/COEF/ALPHA(100,1C0), BETA(100,100)015031CCOMHON/COEF/ALPHA(100,1C0), BETA(100,100)015031CCOMHON/PARAM/FKOUNT015081COMHON/PARAM/FKOUNT015081COMHON/PARAM/FKOUNT015081COMHON/PARAM/FKOUNT015081COMHON/PARAM/FKOUNT<td< td=""><td>RECTION OR NO. OF B-SPLINES IN X DIRECTION, ECTION. OF B-SPLINES IN X DIRECTION, ECTION. S IDENTIFYING KNOT INTERVALS ICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I)<tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BCTA(100,100)</tx(j+1)) </td><td>01479:C</td><td>NX . NY</td><td>DIMENSION OF SPLINE IN X DIRECTION</td></td<></tx(j+1))<></tx(j+1))<></tx(j+1))<></tx(j+1))<></tx(j+1))<></tx(j+1))<></tx(j+1))<></tx(j+1))<>	RECTION OR NO. OF B-SPLINES IN X DIRECTION, ECTION. OF B-SPLINES IN X DIRECTION, ECTION. S IDENTIFYING KNOT INTERVALS ICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I) <tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BCTA(100,100)</tx(j+1)) 	01479:C	NX . NY	DIMENSION OF SPLINE IN X DIRECTION
01481:CTOTAL NO. OF B-SPLINES IN X DIRECTION01482:CY DIRECTION.01483:CKX,KYORDER OF B-SPLINES IN X DIRECTION,01484:CY DIRECTION.01485:CLEFTX,LEFTY01485:CLEFTX,LEFTY01486:CIN WHICH SQUARE MESH COORDINATES01486:CIN WHICH SQUARE MESH COORDINATES01487:CLIE. (LEFTX(I)=J IMPLIES01488:CTX(J)<=A(I) <tx(j+1))< td="">01489:CIT01490:CW1,W201490:CW1,W201491:CTO BE USED IN GF.01472:CNKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH01473:COR NUHBER OF ELEMENTS IN ARRAYS A,B.01474:CFKOUNT01475:CGF01475:CGF01476:CXK01477:CWHICH ARE USED AS TEST POINTS IN01479:CDETERNINING THE COEFFICIENTS OF THE01479:COUTPUT01500:CMATES GF.01501:COUTPUT01503:CCOMMON/COEF/ALPHA(100,1C0),BETA(100,100)01507:COMMON/CDEF/ALPHA(100,1C0),BETA(100,100)01507:COMMON/PARAM/FKOUNT01508:COMMON/PARAM/FKOUNT01508:COMMON/PARAM/FKOUNT</tx(j+1))<>	NO. OF B-SPLINES IN X DIRECTION, ECTION. OF B-SPLINES IN X DIRECTION, ECTION. S IDENTIFYING KNOT INTERVALS ICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I) <tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BCTA(100,100)</tx(j+1)) 	01480:C		Y DIRECTION OR
01482:CY DJRECTION.01483:CKX,KYORDER OF B-SPLINES IN X DIRECTION,01484:CY DIRECTION.01485:CLEFTX,LEFTYARRAYS IDENTIFYING KNOT INTERVALS01486:CIN WHICH SQUARE MESH COORDINATES01487:CLIE. (LEFTX(I)=J IMPLIES01488:CTX(J)<=A(I) <tx(j+1))< td="">01489:C001490:CW1,W201497:CNKNOTX,NKNOTYDIMENSIONS FOR JACOBIAN, DOT PRODUCT01491:CTO BE USED IN GF.01472:CNKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH01473:COR NUMBER OF ELEMENTS IN ARRAYS A,B.01475:CGF01494:CXK01475:CGF01476:CXK01477:CWHICH ARE USED AS TEST POINTS IN01478:CDETERMINING THE COEFFICIENTS OF THE01479:CATH DEGREE POLYNOMIAL WHICH APPROXI-01501:C01501:C01502:COUTPUT01503:CCOMMON/CDEF/ALPHA(100,1C0),BETA(100,100)01507:COMMON/PARAM/FKOUNT01508:COMMON/PARAM/FKOUNT01508:COMMON/PARAM/FKOUNT</tx(j+1))<>	ECTION. OF B-SPLINES IN X DIRECTION, ECTION. S IDENTIFYING KNOT INTERVALS ICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I) <tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)</tx(j+1)) 	01481:C		TOTAL NO. OF B-SPLINES IN X DIRECTION,
01483:C KX,KY ORDER OF B-SPLINES IN X DIRECTION, 01484:C Y DIRECTION. 01485:C LEFTX,LEFTY ARRAYS IDENTIFYING KNOT INTERVALS 01486:C IN WHICH SQUARE MESH COORDINATES 01487:C LIE. (LEFTX(I)=J IMPLIES 01488:C TX(J)<=A(I) <tx(j+1)) 01489:C V1,W2 WEIGHTS FOR JACOBIAN, DOT PRODUCT 01491:C TO BE USED IN GF. 01492:C NKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH 01493:C OR NUMBER OF ELEMENTS IN ARRAYS A,B. 01494:C FKOUNT PARAMETER CONTAINING NUMBER OF CALLS 01495:C GF 01496:C XK ARRAY CONTAININ POINTS -2,-1,0,1,2 01497:C WHICH ARE USED AS TEST POINTS IN 01498:C DETERMINING THE COEFFICIENTS OF THE 01499:C ATH DEGREE POLYNOMIAL WHICH APPROXI- 01501:C OUTPUT 01503:C OUTPUT 01503:C COMMON/COEF/ALPHA(100,1C0),BL⁻A(100,100) 01507: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM/FKOUNT</tx(j+1)) 	OF B-SPLINES IN X DIRECTION, ECTION. S IDENTIFYING KNOT INTERVALS NICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I) <tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)</tx(j+1)) 	01482:C		Y DIRECTION.
01484:CY DIRECTION.01485:CLEFTX,LEFTYARRAYS IDENTIFYING KNOT INTERVALS01486:CIN WHICH SQUARE MESH COORDINATES01487:CLIE. (LEFTX(I)=J IMPLIES01487:CNKNOTX,JUSCA(I) <tx(j+1))< td="">01489:CTX(J)<=A(I)<tx(j+1))< td="">01489:CTO BE USED IN GF.01490:CW1,W2WEIGHTS FOR JACOBIAN, DOT PRODUCT01491:CTO BE USED IN GF.01492:CNKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH01493:COR NUMBER OF ELEMENTS IN ARRAYS A,B.01494:CFKOUNT01495:CGF01496:CXK01497:CWHICH ARE USED AS TEST POINTS IN01498:CDETERMINING THE COEFFICIENTS OF THE01499:CATH DEGREE POLYNOMIAL WHICH APPROXI-01501:C0UTPUT01503:COUTPUT01503:CCOMHON/COEF/ALPHA(100,1C0),BETA(100,100)01507:COMHON/COEF/ALPHA(100,1C0),BETA(100,100)01507:COMHON/PARAM/FKOUNTCDMMON/PARAM/FKOUNTCOMHON/PARAM/FKOUNT</tx(j+1))<></tx(j+1))<>	ECTION. S IDENTIFYING KNOT INTERVALS NICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I) <tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A.B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)</tx(j+1)) 	01483:C	KX,KY	ORDER OF B-SPLINES IN X DIRECTION,
01485:C LEFTX,LEFTY ARRAYS IDENTIFYING KNOT INTERVALS 01486:C IN WHICH SQUARE MESH COORDINATES 01487:C LIE. (LEFTX(I)=J IMPLIES 01488:C TX(J)<=A(I) <tx(j+1)) 01489:C 01490:C W1,W2 WEIGHTS FOR JACOBIAN, DOT PRODUCT 01491:C TO BE USED IN GF. 01492:C NKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH 01493:C OR NUMBER OF ELEMENTS IN ARRAYS A,B. 01494:C FKOUNT PARAMETER CONTAINING NUMBER OF CALLS 01495:C GF 01496:C XK ARRAY CONTAININ PDINTS -2,-1,0,1,2 01497:C WHICH ARE USED AS TEST PDINTS IN 01498:C DETERMINING THE COEFFICIENTS OF THE 01499:C ATH DEGREE POLYNOMIAL WHICH APPROXI- 01501:C MATES GF. 01504:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C COMMON/COEF/ALPHA(100,1C0),BETA(100,100) 01507: COMMON/PARAM/FKOUNT 01508:C COMMON/PARAM/FKOUNT 01508:C COMMON/PARAM/FKOUNT</tx(j+1)) 	S IDENTIFYING KNOT INTERVALS ICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I) <tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)</tx(j+1)) 	01484:C		Y DIRECTION.
01486:CIN WHICH SQUARE MESH COORDINATES01487:CLIE. (LEFTX(I)=J IMPLIES01488:CTX(J)<=A(I) <tx(j+1))< td="">01489:CTO BE USED IN GF.01490:CW1,W201490:CW1,W201491:CTO BE USED IN GF.01472:CNKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH01473:COR NUMBER OF ELEMENTS IN ARRAYS A,B.01494:CFKOUNT01495:CGF01495:CGF01496:CXK01497:CWHICH ARE USED AS TEST POINTS IN01498:CDETERMINING THE COEFFICIENTS OF THE01497:CATH DEGREE POLYNOMIAL WHICH APPROXI-01500:CMATES GF.01501:C0UTPUT01503:COUTPUT01504:CALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT01506:COMMON/CDEF/ALPHA(100,1C0),BETA(100,100)01507:COMMON/PARAM/FKOUNT01508:COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY</tx(j+1))<>	ICH SQUARE MESH COORDINATES (LEFTX(I)=J IMPLIES <=A(I) <tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BCTA(100,100)</tx(j+1)) 	01485:C	LEFTX,LEFTY	ARRAYS IDENTIFYING KNOT INTERVALS
01487:C LIE. (LEFTX(I)=J IMPLIES 01488:C TX(J)<=A(I) <tx(j+1)) 01489:C 01490:C W1,W2 WEIGHTS FOR JACOBIAN, DOT PRODUCT 01491:C TO BE USED IN GF. 01492:C NKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH 01493:C OR NUMBER OF ELEMENTS IN ARRAYS A,B. 01494:C FKOUNT PARAMETER CONTAINING NUMBER OF CALLS 01495:C GF 01496:C XK ARRAY CONTAININ POINTS -2,-1,0,1,2 01497:C WHICH ARE USED AS TEST POINTS IN 01498:C DETERMINING THE COEFFICIENTS OF THE 01499:C ATH DEGREE POLYNOMIAL WHICH APPROXI- 01500:C MATES GF. 01504:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C 01506: COMMON/CDEF/ALPHA(100,1C0),BETA(100,100) 01507: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY</tx(j+1)) 	<pre>(LEFTX(I)=J IMPLIES <=A(I)<tx(j+1)) TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BCTA(100,100)</tx(j+1)) </pre>	01486:C		IN WHICH SQUARE MESH COORDINATES
01488:C TX(J)<=A(I) <tx(j+1)) 01489:C 01490:C W1,W2 WEIGHTS FOR JACOBIAN, DOT PRODUCT 01491:C TO BE USED IN GF. 01492:C NKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH 01493:C OR NUMBER OF ELEMENTS IN ARRAYS A,B. 01494:C FKOUNT PARAMETER CONTAINING NUMBER OF CALLS 01495:C GF 01496:C XK ARRAY CONTAININ POINTS -2,-1,0,1,2 01497:C WHICH ARE USED AS TEST POINTS IN 01498:C DETERMINING THE COEFFICIENTS OF THE 01499:C ATH DEGREE POLYNOMIAL WHICH APPROXI- 01500:C MATES GF. 01504:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C 01506: CDHHON/COEF/ALPHA(100,1C0),BETA(100,100) 01507: COMMON/PARAM/FKOUNT 01508: CDHHON/PARAM/FKOUNT</tx(j+1)) 	<pre><=A(I)<tx(j+1)) -2,-1,0,1,2="" 00,1c0),bcta(100,100)<="" a,b.="" approxi-="" are="" arrays="" as="" calls="" coefficient="" coefficients="" containin="" containing="" dot="" egree="" elements="" eter="" for="" gf.="" in="" j)new="" jacobian,="" mber="" mesh="" mining="" number="" of="" points="" polynomial="" pre="" product="" sions="" square="" test="" the="" to="" ts="" used="" value="" which=""></tx(j+1))></pre>	01487:C		LIE. (LEFTX(I)=J IMPLIES
01489:C 01490:C 01490:C 01491:C 01491:C 01491:C 01472:C 01472:C 01472:C 01473:C 01493:C 01494:C FKOUNT PARAMETER CONTAINING NUMBER OF CALLS 01494:C 01494:C 01494:C 01494:C 01494:C 01494:C 01494:C 01494:C 01494:C 01494:C 01494:C 01494:C 01494:C 01497:C 01497:C 01497:C 01497:C 01497:C 01498:C 01498:C 01498:C 01498:C 01498:C 01501:C 01501:C 01502:C 01502:C 01504:C 01504:C 01504:C 01504:C 01504:C 01504:C 01504:C 01504:C 01506: 01506: 01506: 01506: 01506: 01506: 01507: 01507: 01507: 01507: 01508:	TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BCTA(100,100)	01488:C		TX(J)<=A(I) <tx(j+1))< td=""></tx(j+1))<>
01490:CW1,W2WEIGHTS FOR JACOBIAN, DOT PRODUCT01491:CTO BE USED IN GF.01491:CNKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH01493:COR NUMBER OF ELEMENTS IN ARRAYS A,B.01494:CFKOUNT01495:CGF01496:CXK01497:CWHICH ARE USED AS TEST POINTS IN01498:CDETERMINING THE COEFFICIENTS OF THE01499:CATH DEGREE POLYNOMIAL WHICH APPROXI-01501:C01502:C01502:COUTPUT01503:CCDMMON/COEF/ALPHA(100,1C0), BETA(100,100)01507:COMMON/PARAM/FKOUNT01508:CDMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	TS FOR JACOBIAN, DOT PRODUCT USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)	01489:C		
01491:CTO BE USED IN GF.01491:CNKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH01493:COR NUMBER OF ELEMENTS IN ARRAYS A,B.01494:CFKOUNT01495:CGF01496:CXK01497:CWHICH ARE USED AS TEST POINTS IN01498:CDETERMINING THE COEFFICIENTS OF THE01499:CATH DEGREE POLYNOMIAL WHICH APPROXI-01500:CMATES GF.01501:C01502:C01504:CALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT01505:CCOMMON/COEF/ALPHA(100,1C0),BETA(100,100)01507:COMMON/PARAM/FKOUNT01508:COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	USED IN GF. SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A.B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)	01490:C	W1.W2	WEIGHTS FOR JACOBIAN, DOT PRODUCT
01472:CNKNOTX,NKNOTYDIMENSIONS FOR SQUARE MESH01473:COR NUMBER OF ELEMENTS IN ARRAYS A,B.01474:CFKOUNT01475:COF01475:COF01476:CXK01477:COR RAMETER CONTAINING NUMBER OF CALLS01477:COHICH ARE USED AS TEST POINTS IN01478:CDETERMINING THE COEFFICIENTS OF THE01479:COTTOL01479:COUTPUT01501:COUTPUT01502:COUTPUT01503:CCOMMON/COEF/ALPHA(100,1C0),BETA(100,100)01507:COMMON/PARAM/FKOUNT01508:COMMON/PARAM/FKOUNT01508:COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	SIONS FOR SQUARE MESH MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)	01491:C		TO BE USED IN GE.
O1493:C OR NUMBER OF ELEMENTS IN ARRAYS A,B. O1494:C FKOUNT PARAMETER CONTAINING NUMBER OF CALLS O1495:C O1496:C XK ARRAY CONTAININ POINTS -2,-1,0,1,2 WHICH ARE USED AS TEST POINTS IN DETERMINING THE COEFFICIENTS OF THE O1497:C UTPUT O1504:C OUTPUT O1504:C OUTPUT O1505:C O1504:C CDMMON/COEF/ALPHA(100,1C0),BETA(100,100) O1507: CDMMON/PARAM/FKOUNT O1508: CDMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	MBER OF ELEMENTS IN ARRAYS A,B. ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)	01492:0	NKNOTX NKNOTY	DIMENSIONS FOR SQUARE MESH
O1494:C FKOUNT PARAMETER CONTAINING NUMBER OF CALLS O1495:C GF O1496:C XK ARRAY CONTAININ POINTS -2,-1,0,1,2 O1497:C WHICH ARE USED AS TEST POINTS IN DETERMINING THE COEFFICIENTS OF THE O1497:C ATH DEGREE POLYNOMIAL WHICH APPROXI- 01497:C MATES GF. O1501:C OUTPUT O1503:C OUTPUT O1503:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT O1505:C COMMON/COEF/ALPHA(100,1C0),BETA(100,100) O1507: COMMON/PARAM/FKOUNT O1508: COMMON/PARAM/FKOUNT	ETER CONTAINING NUMBER OF CALLS TO CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)	01493:C		OR NUMBER OF ELEMENTS IN ARRAYS A.B.
GF 01495:C 01496:C 01496:C 01496:C 01497:C 01497:C 01498:C 01498:C 01498:C 01499:C 01500:C 01500:C 01501:C 01502:C 01502:C 01504:C 01504:C 01504:C 01504:C 01504:C 01504:C 01504:C 01504:C 01506: CDMMON/CDEF/ALPHA(100,1C0),BETA(100,100) 01507: CDMMON/PARAM/FKOUNT 01508: CDMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,100),BETA(100,100)	0149410	FKOUNT	PARAMETER CONTAINING NUMBER OF CALLS TO
O1496:CXKARRAY CONTAININ POINTS -2,-1,0,1,2O1497:CWHICH ARE USED AS TEST POINTS INO1498:CDETERMINING THE COEFFICIENTS OF THEO1499:C4TH DEGREE POLYNOMIAL WHICH APPROXI-O1500:CMATES GF.O1501:C01502:COUTPUT01503:CO1504:CALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENTO1505:CCDMMON/COEF/ALPHA(100,1C0), BETA(100,100)O1507:CDMMON/PARAM/FKOUNTO1508:CDMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	CONTAININ POINTS -2,-1,0,1,2 ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,100),BCTA(100,100)	0149510		GF
01497:C WHICH ARE USED AS TEST POINTS IN 01498:C DETERMINING THE COEFFICIENTS OF THE 01497:C 4TH DEGREE POLYNOMIAL WHICH APPROXI- 01500:C MATES GF. 01501:C 01502:C OUTPUT 01503:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C COMMON/COEF/ALPHA(100,1C0),BETA(100,100) 01507: COMMON/COEF/ALPHA(100).B(100).NX.NY.KX.KY	ARE USED AS TEST POINTS IN MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,100),BETA(100,100)	0149610	XK	ARRAY CONTAININ POINTS -21.0.1.2
01498:C DETERMINING THE COEFFICIENTS OF THE 01499:C ATH DEGREE POLYNOMIAL WHICH APPROXI- 01500:C MATES GF. 01501:C 0UTPUT 01503:C 0UTPUT 01504:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C 01506: COMMON/COEF/ALPHA(100,1C0),BETA(100,100) 01507: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	MINING THE COEFFICIENTS OF THE EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)	01497:C		WHICH ARE USED AS TEST POINTS IN
01499:C4TH DEGREE POLYNOMIAL WHICH APPROXI- 01500:C01500:CMATES GF.01501:C01502:C01502:COUTPUT01503:C01504:C01504:CALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT01505:C01506:01506:COMMON/COEF/ALPHA(100,1C0),BETA(100,100)01507:COMMON/PARAM/FKOUNT01508:COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	EGREE POLYNOMIAL WHICH APPROXI- GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)	01478:C		DETERMINING THE COEFFICIENTS OF THE
01500:C MATES GF. 01501:C 01502:C DUTPUT 01503:C 01504:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C 01506: COMMON/COEF/ALPHA(100,1C0),BETA(100,100) 01507: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	GF. J)NEW VALUE FOR COEFFICIENT 00,1C0),BETA(100,100)	01499:C		ATH DEGREE POLYNOMIAL WHICH APPROXI-
01501:C 01502:C OUTPUT 01503:C 01504:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C 01506: COMMON/COEF/ALPHA(100,1C0),BETA(100,100) 01507: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	J)NEW VALUE FOR COEFFICIENT 00,100),BETA(100,100)	01500:C		MATES GF.
01502:C DUTPUT 01503:C 01504:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C 01506: CDMMON/CDEF/ALPHA(100,1C0),BETA(100,100) 01507: CDMMON/PARAM/FKOUNT 01508: CDMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	J)NEW VALUE FOR COEFFICIENT 00,100),BETA(100,100)	01501:C		
01503:C 01504:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C 01506: CDMMON/COEF/ALPHA(100,1C0),BETA(100,100) 01507: CDMMON/PARAM/FKOUNT 01508: CDMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	J)NEW VALUE FOR COEFFICIENT 00,100),BETA(100,100)	01502:0	OUTPUT	
01504:C ALPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT 01505:C 01506: COMMON/COEF/ALPHA(100,1C0),BETA(100,100) 01507: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	J)NEW VALUE FOR COEFFICIENT 00,100),BETA(100,100)	0150310		
01505:C 01506: COMMON/COEF/ALPHA(100,1C0),BETA(100,100) 01507: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	00,100),BETA(100,100)	0150410	ALPHA(T.I) OR BE	TA(T.I). NEW VALUE FOR COFFETCIENT
01506: COMMON/COEF/ALPHA(100,100),BETA(100,100) 01507: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY	00,100),BETA(100,100)	0150510		
01507: COMMON/PARAM/FKOUNT 01508: COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY		01504:	COMMON/COFE/AL	PHA(100-100)-BETA(100-100)
01508: COMMON/PARAM2/A(100).B(100).NX.NY.KX.KY		01507:	COMMON/PARAM/P	KOUNT
).B(100).NX.NY.KX.KY	01508:	COMMON/PARAM2	A(100).B(100).NX.NY.KX.KY
01509: * .! FFTY(100).! FFTY(100)	00)	01509:	* .1 FFTX(100).1 F	FTY(100)
	NKNOTY	01510:		(NOTX.NKNOTY
01511: COMMON/WEIGHTS/W1-W2	n	01511:	COMMON/WEIGHTS	5/41.42
01512: COMMON/XKS/XK(5)	2	01512:	COMMON/XKS/XK	(5)
015131 REAL R(3).LEND	2	01513:	REAL RU3)+LENI	
01514: FM(R)=C1\$R\$\$\$4+C2\$R\$R\$R+C3\$R\$R+C4\$R+C5	2	01514:	FM(R)=C1\$R\$\$4	
01515: XK(1)=REND	∠ R#R+C3#R#R+C4#R+C5	015151	XK(1)=REND	
01516: DO 50 IK=2.5	∠ R#R+C3#R#R+C4#R+C5	01516:	DO 50 IK=2.5	
01517: XK(IK)=REND-FLOAT(IK)+1.	∠ R#R+C3#R#R+C4#R+C5	01517:	XK(TK)=RENR-FL	OAT(IK)+1.
01518: 50 CONTINUE	∠ R#R+C3#R#R+C4#R+C5 K)+1.	01518:	50 CONTINUE	
01519: CALL CRIT(0.C1.C2.C3.C4.C5.NR00TS.R.MK0UNT.T.U)	∠ R#R+C3#R#R+C4#R+C5 K)+1.	01519:	CALL CRIT(0.C)	-C2+C3+C4+C5+NROOTS+R+MKOUNT+T+U)
015201 IF(NR00TS.NF1) R0T0 55	<pre>∠ R#R+C3#R#R+C4#R+C5 K)+1. 3.C4.C5.NR00T5.R.MK0UNT.T.1)</pre>	01520:	IF (NROOTS, NE.	-1) GOTO 55
01521: PRINTX. /WARNING #1 COFF IS 0/	<pre>2 R*R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) TD 55</pre>	01521:	PRINTE UARNIN	IG #1 COFF IS O'
	<pre>4 R*R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) T0 55 C0EF 15 0/</pre>	015221	RETURN	A IT ARE AR A
01522: RETURN	Z R#R+C3#R#R+C4#R+C5 K)+1. 3,C4,C5,NROOTS,R,MKOUNT,I,J) TO 55 COEF IS 0'	01523:	55 CONTINUE	
01522: RETURN 01523: 55 CONTINUE	<pre>2 R#R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) T0 55 C0EF IS 0'</pre>	01524:	TMIN=10.F10	
01522: RETURN 01523: 55 CONTINUE 01524: THIN=10.E10	<pre></pre>	01525:	IR=0	i
01522: RETURN 01523: 55 CONTINUE 01524: TMIN=10.E10 01525: IR=0	2 R#R+C3#R#R+C4#R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) TO 55 COEF IS 0'	01526:	DO 600 IRONT#1	LANROOTS
01522: RETURN 01523: 55 CONTINUE 01524: TMIN=10.E10 01525: IR=0 01526: DO 600 IROOT=1.NROOTS	<pre>2 R*R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) TD 55 C0EF IS 0' TS</pre>	01527:	IF(R(IROOT).LE	REND) THEN
01509: * .LEFTX(100).LEFTY(100)	00)	01509:	* .LEFTX(100).LE	FTY(100)
01510: COMMON/KNOT/NKNOTX,NKNOTY	NKNOTY	01510:	COMMON/KNOT/N	(NOTX, NKNOTY
01511: COMMON/WEIGHTS/W1,W2		01511:	COMMON/WEIGHTS	5/W1,W2
01512: COMMON/XKS/XK(5)	2	01512:	COMMON/XKS/XK	(5)
01513: REAL R(3), LEND	2	01513:	REAL RIJILENI	
	2	015141	EM(D)-C1+D++A	FLJ4D4D4D7LJ4D4D7LV4D7LL2
		01514:	FM(K)=0146444	ru2akakaktujakaktu4aktuj
01515; XK(1)=REND	∠ R#R+C3#R#R+C4#R+C5	01515;	XK(1)=REND	
01516: DO 50 IK=2,5	∠ R#R+C3#R#R+C4#R+C5	01516:	DO 50 IK=2,5	
01517: XK(IK)=REND-FLOAT(IK)+1.	∠ R#R+C3#R#R+C4#R+C5	01517:	XK(IK)=REND-FL	OAT(IK)+1.
01518: 50 CONTINUE	∠ R#R+C3#R#R+C4#R+C5 K)+1.	01518:	50 CONTINUE	
01519: CALL CRIT(0,C1,C2,C3,C4,C5,NR00TS,R,MK0UNT,I,J)	∠ R#R+C3#R#R+C4#R+C5 K)+1.	01519:	CALL CRIT(0,C)	,C2,C3,C4,C5,NROOTS,R,MKOUNT,I,J)
01520; IF(NROOTS.NE1) GOTO 55	<pre>∠ R*R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J)</pre>	01520:	IF(NROOTS.NE	-1) GOTO 55
01521: PRINT*.'WARNING #1 COFF IS 0'	<pre>∠ R#R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) T0 55</pre>	01521:	PRINT*,'WARNIN	NG #1 COEF IS 0'
	<pre>2 R*R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) T0 55 C0EF IS 0'</pre>	01522:	RETURN	
01522: RETURN	<pre>2 R*R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) T0 55 C0EF IS 0'</pre>	01523;	DD CONTINUE	
01522: RETURN 01523: 55 CONTINUE	Z R#R+C3#R#R+C4#R+C5 K)+1. 3,C4,C5,NROOTS,R,MKOUNT,I,J) TO 55 COEF IS 0'	V1327i	ININ=IV+EIV	
01522: RETURN 01523: 55 CONTINUE 01524: TMIN=10.E10 01525: IR=0	<pre>4 R#R+C3#R#R+C4#R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) T0 55 C0EF IS 0'</pre>	VIJZJi		
01522: RETURN 01523: 55 CONTINUE 01524: TMIN=10.E10 01525: IR=0	<pre>4 R#R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) T0 55 C0EF IS 0' </pre>	VIJZO	DO 900 IKUU!=	
01522: RETURN 01523: 55 CONTINUE 01524: TMIN=10.E10 01525: IR=0 01526: DO 600 IROOT=1,NROOTS	<pre>4 R#R+C3*R*R+C4*R+C5 K)+1. 3,C4,C5,NR00TS,R,MK0UNT,I,J) T0 55 C0EF IS 0' TS </pre>	01527:	IF(R(IROOT),LE	REND) THEN

01528:	IR=IR+1
01529:	R(IR)=R(IROOT)
01530:	ENDIF
01531:	600 CONTINUE
01532:	NROOTS≠1R
01533:	IF (NROOTS.EQ.O) THEN
01534:	TE(MKOUNT/2#2.EQ.MKOUNT) THEN
01535:	
015741	
01530,	
01537.	
013381	DE N (1 + J) = NEMU THIN - EN (DE ND)
01537:	
01540:	ENUIF
01541;	ELSE
01542;	DO 700 IROOT=1,NROOTS
01543:	FMINN=FM(R(IROOT))
01544:	IF(FMINN.LT.TMIN) THEN
01545:	TM1N=FMINN -
01546:	IMIN=IROOT
01547:	ENDIF
01548:	700 CONTINUE
01549:	FMINN=FM(RENI)
01550:	IF(FMINN.IT.TMIN) R(IMIN)=REND
01551:	TE(MKOUNT/2±2.EQ.MKOUNT) THEN
015521	
015521	THTN-CH(D(THTN))
015335	(nin-rn(n(inin))
01554:	
01555:	BEIN(1,J)=R(ININ)
01556:	(MIN=FM(K(IMIN))
01557:	ENDIF
01558:	ENDIF
01557:	RETURN
01560:	END
01561:C	
01562:C	
01563:C	
01564:	SUBROUTINE TESTMINL(MKOUNT, I, J, LEND, REND)
01565:C	
01566:0	FOR A GIVEN COEFFICIENT ALPHA(I.J) OR BETA(I.J)
0156710	TESTMINE FINDS AND TESTS THE CRITICAL POINTS OF
0156810	THE ATH DEGREE POLYNOMIAL REPRESENTING GE TO
0154010	RETERNINE WHICH POINT YTELDS THE SMALLEST VALUE FOR GE
0157010	WHEN GE IS UTENED AS A FUNCTION OF THAT COFFETCIENT.
0157110	THE CHARLEST HALLE TO COMPARE WITH THE UALLE AT THE
015/1+0	THE GUNGELEGT ANGUE TO COMPANED WITH THE ANGUE OF THE
0157210	LEFT ERUFUINT OF THE INTERVAL (LERUTREAD) To determine at luar number the minimum hathe
010/310	THE PERSONNE AT WARE CHOSEN RECOVER THE NEW
	UP OF ULLUKS, THE NUMBER LHUSEN BELUMES THE NEW
015/5:C	VALUE FUR ALFHA(1,J) UK RE(A(1,J) .
01576 : C	
01577 : C	INPUT
01578:C	

01579:C	MKOUNT	MKOUNT EVEN MEANS THE COEFFICIENT
01580:C		INVOLVED IN MINIMIZATION IS IN THE ALPHA
01581:C		ARRAY, MKOUNT ODD MEANS THE COEFFICIENT
01582:C		IS IN THE BETA ARRAY.
01583:C	I,J.,	SUBSCRIPTS FOR COEFFICIENT INVOLVED
01584:C		IN MINIMIZATION
01585:C	LEND,REND	LEFT AND RIGHT ENDPOINTS FOR
01586:C		INTERVAL. REND IS A VERY LARGE NUMBER,
01587:C		INDICATING THAT THE RIGHT ENDPOINT IS
01588:C		INFINITE.
01587;C	(IN COMMON)	
01590:C	ALPHA, BETA	ARRAYS CONTAINING COEFFICIENTS OF
01591:C		TENSOR PRODUCT SPLINE MAPPING.
01592:C	A,B	ARRAYS CONTAINING COORDINATES FOR
01593:C		SQUARE MESH.
01594:C	NX, NY	DIMENSION OF SPLINE IN X DIRECTION
01595:C		,Y DIRECTION OR
01596:C		TOTAL NO. OF B-SPLINES IN X DIRECTION,
01597:C		Y DIRECTION.
01598:C	KX,KY	ORDER OF B-SPLINES IN X DIRECTION,
01599:C		Y DIRECTION.
01600:C	LEFTX,LEFTY	ARRAYS IDENTIFYING KNOT INTERVALS
01601:C		IN WHICH SQUARE MESH COORDINATES
01602:C		LIE. (LEFTX(I)=J IMPLIES
01603:C		TX(J)<=A(I) <tx(j+1))< td=""></tx(j+1))<>
01604:C		
01605;C	W1,W2	WEIGHTS FOR JACOBIAN, DOT PRODUCT
01606:C		TO BE USED IN GF.
01607:C	NKNOTX,NKNOTY	DIMENSIONS FOR SQUARE MESH
01608:C		OR NUMBER OF ELEMENTS IN ARRAYS A.B.
01609:C	FKOUNT	PARAMETER CONTAINING NUMBER OF CALLS TO
01610:C		GF
01611;C	XK	ARRAY CONTAININ POINTS -2,-1,0,1,2
01612:C		WHICH ARE USED AS TEST POINTS IN
01613:C		DETERMINING THE COEFFICIENTS OF THE
01614:C		ATH DEGREE POLYNOMIAL WHICH APPROXI-
01615:C		MATES GF.
01615:C		
01617:C	OUTPUT	
01618 : C		
01619:C	ALPHA(I,J) OR B	ETA(I,J)NEW VALUT FOR COEFFICIENT
01620:C		
01621;	COMMON/COEF/A	LPHA(100,100),BETA(100,100)
01622:	COMMON/PARAM/	FKOUNT
01623:	COMMON/PARAM2	/A(100),B(100),NX,NY,KX,KY
01624:	<pre># ,LEFTX(100),L</pre>	EFTY(100)
01625:	COMMON/KNOT/N	KNOTX, NKNOTY
01626:	COMMON/WEIGHT	5/W1,W2
01627:	COMMON/XKS/XK	(5)
01628:	REAL R(3),LEN	0
014291	2440413-4043	

•

01630:		XK(1)=LEND
01631:		DO 50 IK=2,5
01632:		XK(IK)=LEND+FLOAT(IK)-1.
01633:	50	CONTINUE
01634:		CALL CRIT(0,C1,C2,C3,C4,C5,NROOTS,R,MKOUNT,
01635:	*	I,J)
01636:		IF(NROOTS.NE1) GOTO 55
01337:		PRINT*, WARNING #1 COEF IS 0'
01638:		RETURN
01639:	55	CONTINUE
01640:		TMIN=10.0E10
01641:		IL=0
01642:		DO 600 IR=1,NROOTS
01643:		IF(R(IR).GE.LEND)THEN
01644:		IL=IL+1
01645:		R(IL)=R(IR)
01646:		ENDIF
01647:	600	CONTINUE
01648:		NROOTS=IL
01649:		IF(NROOTS.EQ.O) THEN
01650:		IF(MKOUNT/2#2.EQ.MKOUNT) THEN
01651:		ALPHA(I,J)=LEND
01652:		TMIN=FM(LEND)
01653:		ELSE
01654:		BETA(I,J)=LEND
01655:		TMIN=FM(LEND)
01656:		ENDIF
01657:		ELSE
01658:		DO 700 IR=1,NROOTS
01659:		FMINN=FM(R(IR))
01660;		IF(FMINN.LT.TMIN) THEN
01661:		TMIN=FMINN
01662:		IMIN=IR
01663:		ENDIF
01664:	700	CONTINUE
01665:		FMINN=FM(LEND)
01666;		IF(FMINN.LT.TMIN) R(IMIN)=LEND
01667:		IF(MKOUNT/2#2.EQ.MKOUNT) THEN
01668:		ALPHA(I,J)=R(IMIN)
01669:		TMIN=FM(R(IMIN))
01670:		ELSE
01671:		BETA(I,J)=R(IMIN)
01672:		TMIN=FM(R(IMIN))
01673;		ENDIF
01674:		ENDIF
01675:		RETURN
01676:		END
01677:C		
01678:C		
01679:C		
01680;		SUBROUTINE TESTMINB(MKOUNT,I,J,LEND,REND)

01681 IC		
01682:C	FOR A GIVEN	COEFFICIENT ALPHA(I,J) OR BETA(I,J)
01683:C	TESTMINB FINDS AN	D TESTS THE CRITICAL POINTS OF
01684:C	THE 4TH DEGREE PO	LYNOMIAL REPRESENTING GF TO
01685:C	DETERMINE WHICH P	DINT YIELDS THE SMALLEST VALUE FOR GF
0148610	WHEN OF IS VIEWED	AS A FUNCTION OF THAT COFFETCIENT.
0168710	THE SMALLEST VALU	E IS COMPARED WITH THE VALUE
0148810	AT THE ENTIPOINTS	OF THE INTERVAL (LEND.REND)
0143910	TO DETERMINE AT W	HAT NURFR THE NINIMUM VALUE
0140010	OF BE OCCURS. THE	NUMBER FURGEN BEFOMER THE NEL
0149110	UALUE ERR ALPHACT	I) OR RETA(T. I)
0140310		JUN DETINITION
0107210	TNOUT	
0107310	INFUI	
0167416	MKOUNIT	
01695:0	MKUUNI	MAUUNT EVEN MEANS THE COEFFICIENT
01696:C		INVOLVED IN MINIMIZATION IS IN THE ALPHA
01697 : C		ARRAY. MKOUNT ODD MEANS THE COEFFICIENT
01698:C		IS IN THE BETA ARRAY.
01699:C	I,J	SUBSCRIPTS FOR COEFFICIENT INVOLVED
01700:C		IN MINIMIZATION
01701:C	LEND,REND	LEFT AND RIGHT ENDPOINTS FOR
01702:C	•	INTERVAL
01703:C	(IN COMMON)	· · · ·
01704:0	AL PHA. BETA.	ARRAYS CONTAINING COEFFICIENTS OF
01705:0		TENSOR PRODUCT SPLINE MAPPING.
01704:0	A.B	ARRAYS CONTAINING COORDINATES FOR
0170710		CONADE MECH
01707.0	NY NY	DIVENETAN AF SELTNE IN Y AIDECTION
01708.0	MA # M + + +	V DIDECTION OF SPLINE IN A DIRECTION
01/07:0		I DIRECTION UR
01/10:0		TUTAL NU. OF B-SPLINES IN X DIRECTION,
01711:0		Y DIRECTION.
01712:C	КХ,КҮ	ORDER OF B-SPLINES IN X DIRECTION,
01713:C		Y DIRECTION.
01714:C	LEFTX,LEFTY	ARRAYS IDENTIFYING KNOT INTERVALS
01715:C		IN WHICH SQUARE MESH COORDINATES
01716:C		LIE. (LEFTX(I)=J IMPLIES
01717:C		$TX(J) \leq = A(I) \leq TX(J+1)$
01718:C		
01719:0	H1.H2	WEIGHTS FOR JACOBIAN, DOT PRODUCT
0172010		TO BE USED IN GE.
01721:0	NKNOTY NKNOTY	DINENSIONS FOR SOMARE MESH
0172710		OD NUMBER OF ELEMENTS IN ADRAVE A.B.
0170710	EKONNI	DADANETED CONTAINING NUMBED OF CALLS TO
V1/2310	FNUURI + + +	CONTRACTOR CONTRINUTION NUMBER OF CALLS IN
V1/291U		
0172510	XK	AKKAT CUNTAININ POINTS -2,-1,0,1,2
U1726:C		WHICH ARE USED AS TEST POINTS IN
01727:0		DETERMINING THE COEFFICIENTS OF THE
01728:C		ATH DEGREE POLYNOMIAL WHICH APPROXI-
01729:C		MATES GF.
01730:C		
01731:C	OUTPUT	

01/3210		
01733:C	A	LPHA(I,J) OR BETA(I,J)NEW VALUE FOR COEFFICIENT
01734:C		
01735:		COMMON/CDEF/ALPHA(100,100),BETA(100,100)
01736:		CONMON/PARAM/FKOUNT
01737:		COMMON/PARAM2/A(100),B(100),NX,NY,KX,KY
01738:	*	,LEFTX(100),LEFTY(100)
01739:		COMMON/KNOT/NKNOTX,NKNOTY
01740:		COMMON/WEIGHTS/W1.W2
01741:		COMMON/XKS/XK(5)
01742:		REAL R(3), LEND
01743:		FM(R)=C1*R**4+C2*R*R*R+C3*R*R+C4*R+C5
01744:		CENTER=(LEND+REND)/2.
01745:		XK(1)=LEND-CENTER
01746:		XK(2)=(END-CENTER)/2.
01747:		XK(3)=0.
01748:		XK(A)=(REND-CENTER)/2.
Δ174Q!		XK(S)=REND-CENTER
01750:		CALL CRIT(CENTER.C1.C2.C3.C4.C5.NROOTS.R.MKOUNT.
017511	T	
01752:		TE(NROOTS,NE1) GOTO 55
01753:		RETIIRN
017541	55	CONTINUE
017551		TNIN=10.0F10
01756:		TREO
01757:		DD A00 TR=1.NRDATS
01758!		TE (R (TR) + CENTER, RE. I END. AND. R (TR) + CENTER, I E. REND) THEN
01759:		TRETRAN
01760:		P(IR)=R(IR)
017411		
017621	600	
01763:		NBOOTSETB
01764		TE (NROOTS, EQ. A) THEN
01765:		TE(MKOUNT/2#2.ED.MKOUNT) THEN
01766		AI PHA(1, 1)=CENTER
01767:		TMTN=FM(CFNTFR)
01768:		FISE
01769:		BETA(1.1)=CENTER
01770:		
01771:		ENDIE
01772:		FISE
01773:		10 700 IR=1-NROOTS
01774!		ENTNN=EN(R(TR))
01775:		TE (EMTNN.I T. TMTN) THEN
01774!		
017771		THINE TO
01778:		FNDIF
017791	700	CONTINUE
01780!	/ • •	
01781:		FNINNR=FM(RENI)
01782:		TE(ENTINULLITEMINNR) THEN

01783: IF(FMINNL.LT.TMIN) THEN R(IMIN)=LEND 01784: TMIN=FM(LEND) 01785: ENDIF 01786: 01787: ELSE IF (FMINNR.LT.TMIN) THEN R(IMIN)=REND 01788: 01789: TMIN=FM(REND) 01790: ENDIF 01791: IF (HKOUNT/2#2.EQ. HKOUNT) THEN 01792: ALPHA(1, J)=R(IMIN)+CENTER 01793: ELSE 01794: BETA(I,J)=R(IMIN)+CENTER 01795: ENDIF 01796: ENDIF 01797: RETURN 01798: END 01799:C 01800:C 01801:C 01802: SUBROUTINE CUBIC(A3,A2,A1,A0,NROOTS,RR) 01803:C CUBIC CONPUTES THE ROOTS OF A CUBIC POLY-01804:C 01805:C NOMIAL USING FORMULAS FROM 'HANDBOOK OF 01806:C MATHEMATICAL TABLES AND FORMULAS' BY RICHARD 01807:C STEVENS BURINGTON, PH.D., MCGRAW-HILL , NEW YORK, 01808:C 1962. 01809:C 01810:C INPUT COEFFICIENTS OF CUBIC POLY-01811:C A3.A2.A1.A0.. 01812:C NOMIAL 01813:C 01814:C OUTPUT 01815:C NROOTS... NUMBER OF DIFFERENT REAL ROOTS 01816:C ARRAY CONTAINING REAL ROOTS 01817;C **RR...** 01818:C REAL RR(3) 01817: 01820: PI=3.1415 01821: P=A2/A3 01822: Q=A1/A3 01823: R=A0/A3 01824: A=1./3.*(3.*Q-P*P) B=1./27.#(2.#P#P#P-9.#P#Q+27.#R) 01825: IF (ABS(B).LT.1.0E-06) THEN 01826: 01827: SIGNB=0. 01828: ELSE SIGNB=B/ABS(B) 01829: 01830: ENDIF 01831: BB=B*B/4. AAA=A*A/27. 01832: 01833: TEST=BB+AAA

.

01834:	IF(TEST.LT.O) THEN
01835:	NROOTS=3
01836:	PHI=ACOS(-SIGNB#SQRT(BB/(-AAA)))
01837:	SRT=SQRT(-A/3,)
01838:	RR(1)=2.*SRT*COS(PHI/3.)
01839:	RR(2)=2.#SRT#COS(PH1/3.+2.#P1/3.)
01840:	RR(3)=2.#SRT#COS(PH1/3.#A.#PT/3.)
01941	ELEE TETTERT AT AN THEN
V10711	NDOATC-1
018421	
018431	31=-,J##T3WKI(1231) 03=
018441	32=-, J4B-54KI(1231)
01845;	IF (ABS(S1).LT.1.0E-06) THEN
01846:	SIGNS1=0.
01847:	ELSE
01848:	SIGNS1=S1/./BS(S1)
01849:	ENDIF
01850:	IF (ABS(S2).LT.1.0E-06) THEN
01851:	SIGNS2=0.
01852:	ELSE
01853:	SIGNS2#52/ARS(52)
018541	ENTITE
012551	DD/1)=CTGNC1#/ADC/C1)##/1./7.))
019541	* TCICNCJA(VDC/CJ)AA(1/2//)
V10J0.	# TOIDHOL#(NEO(04/##\1+/0+/)
0185/:	
01858:	
01859:	RR(1)=-51GNB#2.#5UR1(-A/3.)
01860:	RR(2)=SIGNB#SORT(-A/3.)
01861:	END IF
01862:	DO 10 I=1,3
01863:	RR(I)=RR(I)-P/3.
01864:	10 CONTINUE
01865:	RETURN
01866:	END
01867:C	
01868:C	
01869:0	
01870:	SUBROUTINE EXTREMES(X.Y.TMGX.THIN.NR.NC)
01871:C	
01872:0	
01873:0	EXTREMES FINDS THE MAYTMUM AND MINIMUM UNLIES
0107410	AMONG THE ELEMENTS OF THE THE THE THENSY ON A DECA
V10/4.C	HORD THE ELENERIS OF TWO INO-DIMENSIONAL HARMIST
01073+0	7 10117
018/016	INFUI
018//:L	
018/8:0	X X CUMPONENT
018/910	TT CONFUNENT
01880:C	NRDIMENSION OF X ARRAY
01881:C	NCDIMENSION OF Y ARRAY
01882:C	
01883:C	OUTPUT
01884:C	

THAX ... HAXINUH VALUE IN ARRAYS 01885:C 01886:C THIN ... MINIMUM VALUE IN ARRAYS 018871C 01888: REAL X(100,100),Y(100,100) TMAX=X(1,1) 01889: 01890: TMIN=X(1.1) 01891: DO 20 I=1,NR 01892: 10 10 J=1,NC 01893: IF(X(I,J).GT.TNAX) TMAX=X(I,J) 01894: IF(X(I,J).LT.TMIN) TMIN=X(I,J) 01895: IF(Y(I,J).GT.TMAX) TMAX=Y(I,J) IF(Y(I,J),LT,TMIN) THIN=Y(I,J) 01896: 01897: 10 CONTINUE 01898; 20 CONTINUE 01899; RETURN END 01900: 01901:C 01902:C 01903:C SUBROUTINE NORM(X,Y,TMAX,TMIN,NR,NC) 01904: 01905:C 01906:C THIS ROUTINE NORMALIZES THE VALUES OF TWO TWO DIMENSIONAL ARRAYS SO THAT THEY LIE 01907:C BETWEEN O AND 1 INCLUSIVE. 01908:C 01909:C INPUT 01910:C 01911:C 01912:C X...X COMPONENT ARRAY ON INPUT AND 01913:C NORMALIZED X COMPONENT ARRAY ON OUTPUT Y...Y COMPONENT ARRAY ON INPUT AND 01914:C NORMALIZED Y COMPONENT ARRAY ON OUTPUT 01915:C NR...DIMENSION OF X ARRAY 01916:C 01917:C NC...DIMENSION OF Y ARRAY TMAX... HAXIMUM VALUE IN ARRAYS 01918:C THIN ... HINIHUM VALUE IN ARRAYS 01919;C 01920:C OUTPUT 01921:C 01922:0 01923:C X...NORMALIZED X ARRAY Y...NORMALIZED Y ARRAY 01924:C 01925:C 01926: REAL X(100,100), Y(100,100) 01927: DO 20 I=1,NR 01928: DO 10 J=1,NC 01929: X(I,J)=(X(I,J)-THIN)/(THAX-THIN)Y(I,J)=(Y(I,J)-TMIN)/(TMAX-TMIN)01930: 01931: 10 CONTINUE CONTINUE 01932: 20 RETURN 01933: 01934: END BOTTOM

143

	2 Courses Anno		Destate Catalog	Ato .
NASA CR-177968	2. Ournment Access	041 NO.	J. Necipient's Catalog	10
4. Title and Subtitle			5. Report Date	
Algebraic Grid Generation U	sing Tensor Prod	luct	September 19	85
B-Splines			p. renorming organiza	
7. Author(s)			8. Performing Organiza	tion Report No.
Bonita V. Saunders				
			10. Work Unit No.	
9. Performing Organization Name and Address				
Old Dominion University		1	1. Contract or Grant NGT 47-003-8	No. 02
Nortolk, Virginia				
12 Section Agency Name and Address			3. Type of Report an	d Perioc Covered
National Aeronautics and Spa	ace Administrati	lon	Contractor R	eport
Washington, DC 20546			14. Sponsoring Agency 505=31=83=02	Code
15. Supplementary Notes This report is a dissertation fulfillment of the requirement Computational and Applied M. Langley Technical Monitor:	on submitted to ents for the Deg athematics. R. E. Smith	Old Dominion Univ gree of Doctor and	versity for pa f Philosphy in	rtial
In general, finite dif grid has lines which a the development of an boundary description. generation mapping are concept of the degree requirements are necess grid generation algori Boundary coefficients variation diminishing coefficients are initia approximation to the the the boundary of the un practicality of optimiz Jacobian of the grid generated by using the 17. Key Words (Suggested by Author(h)) grid generation, Computation Dynamics, Optimization	ference methods re smooth and ne algorithm which Topological cor discussed. In of a mapping and sary if a mappin thm uses a mappin are chosen so the spline approxima ally chosen to g ransfinite bilir it square onto t zing the grid by eneration mappin tangent to the algorithm are p	are more success early orthogonal. produces such a siderations in si particular, this thow it can be us ing composed of bi- nat the splines plation to the bound give a variation of the boundary of the minimizing a fur og at each interior grid lines is involved.	ful if the acc This thesis grid when give tructuring the thesis examin sed to determi a suitable gri icubic B-splin roduce Schoenb dary. Interio diminishing of the functio he grid. The nctional invol or grid point vestigated. G	Ompanying discusses n the grid es the ne what d. The es. erg's r n mapping ving the and the rids
19. Security Classif. (of this report)	20. Security Classif. (c	of this page)	21. No. of Pages	22 Fra.e*

1

. ...

· •

*For sale by the Pational Technical Information Service, Springfield, Virginia 22161

.