

JAERI-M

7419

内挿法の数値解法プログラム
(SSLの拡充とベンチマーク・テスト No.3)

1977年12月

伊勢 武治・筒井 恒夫

日本原子力研究所
Japan Atomic Energy Research Institute

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問合せは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

内挿法の数値解法プログラム
(SSLの拡充とベンチマーク・テスト No.3)

日本原子力研究所東海研究所原子炉工学部
伊勢 武治・筒井 恒夫

(1977年11月7日受理)

最近の内挿法の数値解法に対して、実際的なベンチマーク問題を通してのテストが行われ、それらの相互評価がなされた。対象とされた方法は、1次元内挿では、Lagrange 多項式法、Aitken-Lagrange 法、連分数法、スプライン法、および Akima の方法であり、2次元内挿では、スプライン法および Akima の方法である。このテストの結果、Akima の方法が全ての点で優れており、現在よく用いられているスプライン法にとてかわるべきであることが示された。

付録には、用いられた全プログラムのジョブ・コントロール・カード付き入力例と、Anderson の作成した3次元内挿法プログラムを標準FORTRANに書き直したプログラム・リストが載せてある。

JAERI-M 7419

Computer Programs for Interpolations
(Development and Benchmark Test of SSL, No.3)

Takeharu ISE and Tsuneo TSUTSUI

Division of Reactor Engineering, Tokai Research Establishment, JAERI

(Received November 7, 1977)

Numerical evaluations have been performed on recent solution algorithms for interpolations through practical benchmark testings. Computer programs adopted for the present tests are those based on the Lagrange polynomial method, Aitken-Lagrange iterative method, the continued fraction iterative method, the spline method, and Akima's method, for uni-variate interpolations. In addition, programs using the spline method and Akima's method are tested for bi-variate interpolations. The results have shown that Akima's method is the best in every respect and hence it should take the place of the spline method used most often at present.

In Appendix, input examples with job control cards are given for all tested programs and Anderson's program list rewritten in the standard FORTRAN for ter-variate interpolations.

Keywords:

Interpolation, Benchmark test, Comparative Evaluation, Computer Program, Lagrange Method, Aitken-Lagrange Method, Continued Fraction Method, Spline Method, Akima's Method, Uni-Variate Function, Bi-Variate Function, Ter-Variate Function

總論

こゝ数年来、数値解析理論や新しいアルゴリズムばかりでなくSSL(Scientific Subroutine Library, 科学用サブルーチン・ライブラリー)や、特定の問題により深く関係した数値解法プログラム(例えば、Comp. Phys. Comm.)の発表件数は、膨大なものがある。従って、これらのルーチンを隅なくサーベイし、且つ整備しておくことは不可能であるのは勿論、効率的利用という立場から見ても、労多くして功少なしの感がある。そこで、実用上は各ルーチンの特徴の分析やベンチマーク・テスト，“stiff”な問題への使用経験など、集約された情報が要求され、これらが充分蓄積された時に、ルーチン相互の位置づけや体系化が可能となる。

我々の研究室では、数値解析の広い分野に亘る、これまでのアルゴリズム調査を土台として、ベンチマーク・テストなどの情報集約と評価を行った。対象としたのは主に最近発表されたプログラムとアルゴリズムで、先に行なったアルゴリズム調査の延長という形でまとめたので御利用いただきたい。

(西田雄彦、藤村統一郎)

目 次

1. はじめに.....	1
2. 内挿法の最近の発展.....	2
3. Akima の内挿法の数値解法アルゴリズム.....	3
4. Akima の内挿法プログラムの使い方.....	6
4.1 $Y = F(X)$ 型の内挿プログラム INTRPL.....	6
4.2 $Y = F(X)$ 型の曲線あてはめプログラム CURVFT	6
4.3 $Z = F(X, Y)$ 型の内挿プログラム ITPLBV	10
4.4 $Z = F(X, Y)$ 型の曲面あてはめプログラム SFCFIT	14
5. ベンチマーク・テストによる内挿法プログラムの評価.....	24
5.1 $Y = F(X)$ 型の内挿法プログラム.....	24
5.2 $Z = F(X, Y)$ 型の内挿法プログラム.....	43
6. 結 論.....	51
謝 辞.....	51
文 献.....	52
付 錄.....	54
A.1 ベンチマーク・テストで用いられた内挿法プログラムのジョブ・コントロール・カ ード付き入力例	
A.2 標準FORTRANに書き直されたAndersonの3次元内挿プログラムのリスト	

Contents

1. Introduction	1
2. Recent Development of Interpolation Method	2
3. Solution Algorithm of Akima's Method	3
4. How to Use Akima's Computer Programs	6
4.1 INTRPL, Interpolating Program of Uni-Variate Functions	6
4.2 CURVFT, Curve-Fitting Program of Uni-Variate Functions	6
4.3 ITPLBV, Interpolating Program of Bi-Variate Functions	10
4.4 SFCFIT, Surface-Fitting Program of Bi-Variate Functions	14
5. Numerical Evaluation of Interpolation Computer Programs through Benchmark Tests	24
5.1 Uni-Variate Computer Programs	24
5.2 Bi-Variate Computer Programs	43
6. Conclusion	51
Acknowledgement	51
References	52
Appendix	54
A.1 Input Examples with Job Control Cards of Interpolation Programs Used in Benchmark Tests	
A.2 Program List of Anderson's Ter-Variate Interpolation Program Rewritten in Standard FORTRAN	

1. はじめに

内挿法（或いは補間法, interpolation method^{1)~6)}が直接用いられるときとしては、次の3通りが考えられる。

(i) 正しい値とされているデータ点群から、必要とする値を内挿で求める。正しい値のデータ点とは、データ点が誤差を持っていないか、或いは誤差を持っていてもそれが非常に小さい値であるとする。^{*1)}

(ii) データ点群を構成している関数式が複雑な表現をしていて、必要な全ての値を求めるのは、経済的な意味でも容易でないとき、予め幾つかの値をできるだけ正確に求めておいて、後でこれらの値を用いて必要な値を内挿で求める。

(iii) プロッターなどで図形表示するとき、できるだけ滑らかな曲線を画くために、データ点間を更に細分化するが、これらの細分化点をできるだけ多く求めておく。

従来は、(i)および(ii)の意味で内挿法が用いられることが多かったが、電子計算機およびそれに伴う周辺機器が高度に発展しつつある現在では、(iii)の場合も、CAD(Computer Aided Design)における図形作成などのように、重要な役割を果している。^{6, 7)}いずれにせよ、用いられるべき内挿法は、データ点群が意味している関数系に、できるだけ高い精度で一致している値を求めるようなものでなければならない。

内挿法は、上記に述べた、内挿それ自身に用いること以外に、現在では他方面への応用がいろいろ行われている。まづ第一に挙げられるのは特殊関数、一般関数などの関数近似への応用であろう。^{8)~10)}最小自乗法では、平滑化操作を含めて、あてはめ関数として常に用いられる。^{11, 12)}数値積分では被積分関数の近似としてよく用いられる。^{13, 14)}微積分方程式の応用では、有限要素法に於ける基底関数として用いられるので特に重要である。¹⁵⁾

現在までのところ、多項式系のLagrange関数法およびChebyshev関数法、反復型に属する連分数法、Aitken-Lagrange法およびAitken-Hermite法、区分多項式を連ねて用いるスプライン関数法、^{*2)}などの基本的な1次元内挿法はどの計算センターでも整備しているようである。^{4, 5)}これらのうち、スプライン関数法は従来のものより局所的な性質をよく表わすものとして、最近特に注目されてきている。しかしながら、このスプライン法も不自然な曲線を与えることが知られている。これらの事柄に鑑みて、本報ではスプライン関数法の欠点を克服したAkimaの方法を中心に、又、実用上の観点からも、1次元と2次元の内挿法プログラムの紹介と実際的なベンチマーク・テストを通して、プログラムの評価検討を行った。

第2章以下の記述では、数値解析の初步の知識は前提としているので、必要に応じて適当なテキストを参照されたい。^{1), 2)}尚、内挿法のときとプロッターなどの図形処理のときとでは、次元の定

*1) データ点が正確でなかったり、或いは統計的分布を持っているときは、普通、最小自乗法や統計処理などの別の方法が採られる。

*2) スプライン(spline)とは、我々が曲線を手書きするときに用いる柔い定規のこと。内挿法のときは、普通3次の自然スプライン(関数)を指す。

義が異なるようなので注意されたい。例えば、 $Z = F(X, Y)$ 型の内挿に対して、プロッター関係では 3 次元と呼んでいるのに対し、¹⁶⁾ 数値解析関係では 2 次元と呼んでいる。我々は後者の習慣に従うが、紛らわしいので、できるだけ、 $Z = F(X, Y)$ 型という呼称をとることにする。又、滑らかな曲線 (smooth curve) と呼んだときは、最小自乗法による平滑化操作 (smoothing) で得られた曲線のことではなく、データ点群から手書きで曲線を画くとき、熟練した製図家が画くような自然な (natural) 曲線を指すこととする。

このリポートは、大体どの章も独立して読めるように配慮したつもりであるが、読者の便宜の事も考えて、全体の構成について述べておく。最新の内挿法については第 2 章を、プログラムとしての評価結果については第 5 章を参照されたい。第 5 章ではできるだけ視覚的に理解できるように図形表示が多く用いられている。第 3 章と第 4 章は、現在最も良い内挿法と思われる Akima の方法 (1 次元および 2 次元の内挿) の説明とプログラムの用法が記されている。付録には、ベンチャーマーク・テストで用いられた全プログラムに対する入力例を参考の為に付けておいた。また、Anderson の 3 次元 ($T = F(X, Y, Z)$) の内挿プログラムは有用と思われる所以、標準 FORTRAN で書き直したプログラム・リストを原著者の了解を得て載せておいた。

2. 内挿法の最近の発展

内挿法のアルゴリズムには、第 1 章でもふれたように、多項式型 (Lagrange 内挿, Chebyshev 内挿、など) と反復型 (Newton 法^{*3)} とも呼ばれる。Aitken-Neville 法, Aitken-Lagrange 法, Aitken-Hermite 法, 重心 (barycentric) 法, 接線 (osculatory) 法、など。連分数、有理関数、などを用いる方法も含まれる) とがあるが、従来では計算労力の関係で、比較的多項式型が用いられていた。

Lagrange 内挿では、Runge の現象 (等間隔の引数点の生データに対し、高次の Lagrange 多項式を適用したとき生ずる異常振動。⁶⁾ 数値例については第 5.1 節参照) を起すことがあるが、データ点の配列の順序は問題としない長所を持っている。Chebyshev 内挿でも、Gibbs の現象 (Fourier 級数の有限項までの部分和では不連続点の両側で常に強い振動を示すこと。数値例については第 5 章参照) のような欠点を示すときがあるが、全区間に亘って最良近似 (ミニマックスの意味で) を与えるので、内挿法よりもむしろ関数近似の際用いられることが多い。⁸⁾

最近発展している内挿法の多くは反復型に属するもので、データ点の継ぎ足しで精度を上げられるという長所を持っている。また、反復型という性質から、電子計算機向きの方法であろう。一般には手間を喰うが、多項式系に比べて 1 行から数行高精度であり、特に特異点の近くでの問題 ($e^x, \tan x$ など) に有用である。

最近よく使われるスプライン関数^{17,18)} (特に 3 次の自然スプライン関数) 法は、その平滑性から局所的な振る舞いを正確に表わし、また上記の分類に従えば、どちらかと云えば反復型の方に近い方法であるので、計算機向きの方法である。しかしながら、この方法でも、スプライン関数を計算する際、差分商を用いるので、著しい不等間隔のデータ点群 (これらはスプライン関数の節点を与える) からの内挿や、高次のスプライン関数を用いたときには誤差を生じ、その結果不

*3) 非線形方程式の解法の中にも同名の方法があるが区別されたい。

義が異なるようなので注意されたい。例えば、 $Z = F(X, Y)$ 型の内挿に対して、プロッター関係では 3 次元と呼んでいるのに対し、¹⁶⁾ 数値解析関係では 2 次元と呼んでいる。我々は後者の習慣に従うが、紛らわしいので、できるだけ、 $Z = F(X, Y)$ 型という呼称をとることにする。又、滑らかな曲線 (smooth curve) と呼んだときは、最小自乗法による平滑化操作 (smoothing) で得られた曲線のことではなく、データ点群から手書きで曲線を画くとき、熟練した製図家が画くような自然な (natural) 曲線を指すこととする。

このリポートは、大体どの章も独立して読めるように配慮したつもりであるが、読者の便宜の事も考えて、全体の構成について述べておく。最新の内挿法については第 2 章を、プログラムとしての評価結果については第 5 章を参照されたい。第 5 章ではできるだけ視覚的に理解できるように図形表示が多く用いられている。第 3 章と第 4 章は、現在最も良い内挿法と思われる Akima の方法 (1 次元および 2 次元の内挿) の説明とプログラムの用法が記されている。付録には、ベンチマーク・テストで用いられた全プログラムに対する入力例を参考の為に付けておいた。また、Anderson の 3 次元 ($T = F(X, Y, Z)$) の内挿プログラムは有用と思われるので、標準 FORTRAN で書き直したプログラム・リストを原著者の了解を得て載せておいた。

2. 内挿法の最近の発展

内挿法のアルゴリズムには、第 1 章でもふれたように、多項式型 (Lagrange 内挿, Chebyshev 内挿、など) と反復型 (Newton 法^{*3)} とも呼ばれる。Aitken-Neville 法, Aitken-Lagrange 法, Aitken-Hermite 法, 重心 (barycentric) 法, 接線 (osculatory) 法、など。連分数、有理関数、などを用いる方法も含まれる) とがあるが、従来では計算労力の関係で、比較的多項式型が用いられていた。

Lagrange 内挿では、Runge の現象 (等間隔の引数点の生データに対し、高次の Lagrange 多項式を適用したとき生ずる異常振動。⁶⁾ 数値例については第 5.1 節参照) を起すことがあるが、データ点の配列の順序は問題としない長所を持っている。Chebyshev 内挿でも、Gibbs の現象 (Fourier 級数の有限項までの部分和では不連続点の両側で常に強い振動を示すこと。数値例については第 5 章参照) のような欠点を示すときがあるが、全区間に亘って最良近似 (ミニマックスの意味で) を与えるので、内挿法よりもむしろ関数近似の際用いられることが多い。⁸⁾

最近発展している内挿法の多くは反復型に属するもので、データ点の継ぎ足しで精度を上げられるという長所を持っている。また、反復型という性質から、電子計算機向きの方法であろう。一般には手間を喰うが、多項式系に比べて 1 行から数行高精度であり、特に特異点の近くでの問題 ($e^x, \tan x$ など) に有用である。

最近よく使われるスプライン関数^{17,18)} (特に 3 次の自然スプライン関数) 法は、その平滑性から局所的な振る舞いを正確に表わし、また上記の分類に従えば、どちらかと云えば反復型の方に近い方法であるので、計算機向きの方法である。しかしながら、この方法でも、スプライン関数を計算する際、差分商を用いるので、著しい不等間隔のデータ点群 (これらはスプライン関数の節点を与える) からの内挿や、高次のスプライン関数を用いたときには誤差を生じ、その結果不

*3) 非線形方程式の解法の中にも同名の方法があるが区別されたい。

然な異常屈曲 (unnatural wiggle) を示すことが知られている。¹⁹⁾一方、スプライン (3次の自然スプライン) 関数法のこの欠点を克服したアルゴリズムも幾つか発表されている³⁾。そのひとつである Akima の方法は、^{20~23)} 3次の区分多項式の連結を解析的に行ってゆこうと云う方法で、本報の主題ともなっているので、第3章以下で詳述される。もうひとつは、張力スプライン (spline under tension) 関数法とも呼べるもので、^{24, 25)} この内挿式 (3次の区分多項式) を糸に見立て、異常屈曲線に対して、この糸を引っ掛け引張り (under tension)，自然な曲線にしようと云うもので、この引張り強さに関係するパラメータの値を外部からコントロールする方法である。従って、グラフィック・ディスプレイ上などで会話的に用いるように整備すると便利であろう。Akima の方法、張力スプライン法、いづれも、 $Y = F(X)$ 型と $Z = F(X, Y)$ 型の両方の計算プログラムが公表されている。

パラメータ・コントロール型としてもうひとつ、 $Y = F(X)$ 型であるが、スプライン関数に似た rounded ramp 関数を用いる方法がある。²⁶⁾ 張力スプライン法のような性質を持っているかどうか不明であるが、周期関数にも応用でき、そのときは Fourier 級数法 (Chebyshev 法が同類) よりも優れていると云う。 $Z = F(X, Y)$ 型の式も与えられている。以上の議論については、別に詳しく報告してあるので、こゝではこれ以上は立ち入らないことにする。³⁾

今までの話は1次元 ($Y = F(X)$) 内挿を中心に述べてきたが、理論的にはn次元内挿も可能である。²⁷⁾ しかしながら、実用上からは、2次元 ($Z = F(X, Y)$) および、3次元 ($T = F(X, Y, Z)$) 内挿ができれば充分であろうし、図形処理も可能である。^{*4)}

Günther は2次元にまで拡張した多項式型内挿プログラムを発表している。²⁸⁾ 更に Küfner は2次元および3次元内挿にまで拡張した反復型内挿法 (Aitken 法) を発表している。²⁹⁾ Anderson が発表した3次元内挿法 (3次元をBーススプライン関数のテンソル積として定義した) プログラムは、³⁰⁾ スプライン関数法の拡張でもあり、また使用実績もあるが、(特にプラズマ物理の方面で)³¹⁾ 非標準のFORTRAN で書かれており不便なので、標準のFORTRAN に書き直した。使用例を含むプログラム・リストが付録A.2 に示されている。

3. Akimaの内挿法の数値解法アルゴリズム

前章で紹介した様に、普通の（比較的滑らかな関数系の）問題に対する1次元内挿法の計算プログラムは、どの計算センターでも用意されている。しかしながら、特種な問題を扱う内挿プログラムについては、1次元および2次元内挿法共未整備のようである。3次元内挿に到っては何も準備されてないようである。

これらの現状に鑑みて、緊急に用意されるべきものとして、1次元および2次元内挿に対する Akima の内挿法を探り上げた。計算プログラムの使い方および種々な問題に対する計算例については、次章以下を参照されたい。3次元内挿に対しては、ベンチマーク問題の作成の困難もあって、今回のテストの対象に含まれていないが、利用者の希望もあるうと思うので、使用例付き

*4) 3次元の図表処理は2次元 (実際は $z = f(x, y)$) で立体表示し、残る次元は色付けをする方法であるが、未だ一般的でない。

然な異常屈曲 (unnatural wiggle) を示すことが知られている。¹⁹⁾一方、スプライン (3次の自然スプライン) 関数法のこの欠点を克服したアルゴリズムも幾つか発表されている³⁾。そのひとつである Akima の方法は、^{20~23)} 3次の区分多項式の連結を解析的に行ってゆこうと云う方法で、本報の主題ともなっているので、第3章以下で詳述される。もうひとつは、張力スプライン (spline under tension) 関数法とも呼べるもので、^{24, 25)} この内挿式 (3次の区分多項式) を糸に見立て、異常屈曲線に対して、この糸を引っ掛け引張り (under tension)，自然な曲線にしようと云うもので、この引張り強さに関係するパラメータの値を外部からコントロールする方法である。従って、グラフィック・ディスプレイ上などで会話的に用いるように整備すると便利であろう。Akima の方法、張力スプライン法、いづれも、 $Y = F(X)$ 型と $Z = F(X, Y)$ 型の両方の計算プログラムが公表されている。

パラメータ・コントロール型としてもうひとつ、 $Y = F(X)$ 型であるが、スプライン関数に似た rounded ramp 関数を用いる方法がある。²⁶⁾ 張力スプライン法のような性質を持っているかどうか不明であるが、周期関数にも応用でき、そのときは Fourier 級数法 (Chebyshev 法が同類) よりも優れていると云う。 $Z = F(X, Y)$ 型の式も与えられている。以上の議論については、別に詳しく報告してあるので、こゝではこれ以上は立ち入らないことにする。³⁾

今までの話は1次元 ($Y = F(X)$) 内挿を中心に述べてきたが、理論的にはn次元内挿も可能である。²⁷⁾ しかしながら、実用上からは、2次元 ($Z = F(X, Y)$) および、3次元 ($T = F(X, Y, Z)$) 内挿ができれば充分であろうし、図形処理も可能である。^{*4)}

Günther は2次元にまで拡張した多項式型内挿プログラムを発表している。²⁸⁾ 更に Küfner は2次元および3次元内挿にまで拡張した反復型内挿法 (Aitken 法) を発表している。²⁹⁾ Anderson が発表した3次元内挿法 (3次元をBーススプライン関数のテンソル積として定義した) プログラムは、³⁰⁾ スプライン関数法の拡張でもあり、また使用実績もあるが、(特にプラズマ物理の方面で)³¹⁾、非標準のFORTRAN で書かれており不便なので、標準のFORTRAN に書き直した。使用例を含むプログラム・リストが付録A.2 に示されている。

3. Akimaの内挿法の数値解法アルゴリズム

前章で紹介した様に、普通の（比較的滑らかな関数系の）問題に対する1次元内挿法の計算プログラムは、どの計算センターでも用意されている。しかしながら、特種な問題を扱う内挿プログラムについては、1次元および2次元内挿法共未整備のようである。3次元内挿に到っては何も準備されてないようである。

これらの現状に鑑みて、緊急に用意されるべきものとして、1次元および2次元内挿に対する Akima の内挿法を探り上げた。計算プログラムの使い方および種々な問題に対する計算例については、次章以下を参照されたい。3次元内挿に対しては、ベンチマーク問題の作成の困難もあって、今回のテストの対象に含まれていないが、利用者の希望もあるうと思うので、使用例付き^プ

*4) 3次元の図表処理は2次元 (実際は $z = f(x, y)$) で立体表示し、残る次元は色付けをする方法であるが、未だ一般的でない。

ログラム・リストを付録に示してある。

Akima の方法²⁰⁾は高々 3 次の区分多項式をデータ点に逐次適用してゆくのであるが、その際、各々の点での勾配をその点を中心とした 5 点を以って定めてゆくのが基本的な考え方である。今、5 点に対する番号を左から 1, 2, 3, 4, 5 と付け、点 3 における勾配を、線分 $\overline{12}$ の勾配が線分 $\overline{23}$ の勾配に近づいて行ったときの線分 $\overline{23}$ の勾配と考える。このようにして、点 3 での勾配 t は次式によって定義できる。

$$t = \frac{|m_4 - m_3| m_2 + |m_2 - m_1| m_3}{|m_4 - m_3| + |m_2 - m_1|} \quad (1)$$

こゝに、 m_1, m_2, m_3, m_4 はそれぞれ線分 $\overline{12}, \overline{23}, \overline{34}, \overline{45}$ の勾配である。これらの値を用いて、両端を除いたデータ点間における内挿式は、次のようにして決められる。

今、2 点 $(x_1, y_1), (x_2, y_2)$ に対する次の 4 個の条件：

$$\begin{aligned} &x = x_1 \text{ のとき, } y = y_1 \text{ および } \frac{dy}{dx} = t_1 \\ &x = x_2 \text{ のとき, } y = y_2 \text{ および } \frac{dy}{dx} = t_2 \end{aligned} \quad (2)$$

があれば、一意的に 3 次多項式を決めることができるので、あてはめ曲線（内挿式）として次式が得られる。

$$y = p_0 + p_1(x - x_1) + p_2(x - x_2)^2 + p_3(x - x_3)^3 \quad (3)$$

こゝに

$$\begin{aligned} p_0 &= y_1 \\ p_1 &= t_1 \\ p_2 &= \frac{3 \frac{y_2 - y_1}{x_2 - x_1} - 2t_1 - t_2}{x_2 - x_1} \\ p_3 &= \frac{t_1 + t_2 - 2 \frac{y_2 - y_1}{x_2 - x_1}}{(x_2 - x_1)^2} \end{aligned}$$

データ点群の両端に対しては、今までの話では勾配が決められないので、2 点以上の値を用いて端側の内挿を行う。即ち、端点 (x_3, y_3) とその内側の 2 点 $(x_2, y_2), (x_1, y_1)$ を取り、次の 2 次式をあてはめる。

$$y = g_0 + g_1(x - x_3) + g_2(x - x_3)^2 \quad (4)$$

その際、この定数 $g_i (i=0,1,2)$ を求める為に、仮想的に

$$\begin{aligned} x_5 - x_3 &= x_4 - x_2 = x_3 - x_1 \\ \frac{y_5 - y_4}{x_5 - x_4} - \frac{y_4 - y_3}{x_4 - x_3} &= \frac{y_4 - y_3}{x_4 - x_3} - \frac{y_3 - y_2}{x_3 - x_2} \end{aligned} \quad (5)$$

を満足する 2 点 $(x_4, y_4), (x_5, y_5)$ が採られる。

このようにして、3 次区分多項式を技巧的に用いてゆく Akima の方法によって、今迄最も滑らかな曲線を描くと云われていた 3 次スプライン内挿よりも自然な曲線が得られる。また一般的に

云つて、データ点間隔の不揃いは内挿結果に重要な影響を及ぼすが、Akima の方法は、この点でも滑らかな内挿を行う。更に、多値関数への拡張も、若干の変更で簡単にできるので、地図の等高線などの様なものも滑らかに画くことができる。²¹⁾

Akima は、このアルゴリズムをデータ点が $\{x_i, y_i ; i=1, 2, \dots\}$ の長方形格子から成る。 $z = f(x, y)$ 型関数系の内挿にも拡張している。^{22), 23)} 即ち、あてはめ関数としては双3次多項式：

$$z = f(x, y) = \sum_{\alpha, \beta} x^\alpha y^\beta, \quad \alpha = 0, 1, 2, 3; \beta = 0, 1, 2, 3 \quad (6)$$

を仮定し、 $(x_i, x_{i+1}, y_j, y_{j+1})$ から構成される4本の直線で囲まれる $x-y$ 平面の長方形に適用する。この多項式は、4隅の点における $z=f(x, y)$ の関数値およびその偏微分値 $z_x = (\partial z / \partial x)$, $z_y = (\partial z / \partial y)$, $z_{xy} = (\partial^2 z / \partial x \partial y)$ によって決められる。これらの微分値は、 $y=f(x)$ 型内挿のときに示したように、局所的に基本的には以下のようにして決められる。尚、添字の意味は $y=f(x)$ 型内挿のときと同様である。

今、点 (x_3, y_3) での偏微分値 z_x, z_y が次式で与えられるとする。

$$\begin{aligned} (z_x)_{33} &= \frac{w_{x2} c_{23} + w_{x3} c_{33}}{w_{x2} + w_{x3}} \\ (z_y)_{33} &= \frac{w_{y2} d_{32} + w_{y3} d_{33}}{w_{y2} + w_{y3}} \end{aligned} \quad (7)$$

こゝに重み係数 w_{xi}, w_{yi} は次式で与えられる。

$$\begin{aligned} w_{x2} &= |c_{43} - c_{33}| \\ w_{x3} &= |c_{23} - c_{13}| \\ w_{y2} &= |d_{34} - d_{33}| \\ w_{y3} &= |d_{32} - d_{31}| \end{aligned} \quad (8)$$

また1次の差分商 c_{ij}, d_{ij} は次式で与えられる。

$$\begin{aligned} c_{ij} &= \frac{(z)_{i+1,j} - (z)_{ij}}{x_{i+1} - x_i} \\ d_{ij} &= \frac{(z)_{i,j+1} - (z)_{ij}}{y_{j+1} - y_j} \end{aligned} \quad (9)$$

これらの定義を双3次形関数に適用すれば、点 (x_3, y_3) の混合偏微分 z_{xy} は次式によって求められる。

$$(z_{xy})_{33} = \frac{w_{x2} (w_{y2} e_{22} + w_{y3} e_{33}) + w_{x3} (w_{y2} e_{32} + w_{y3} e_{33})}{(w_{x2} + w_{x3})(w_{y2} + w_{y3})} \quad (10)$$

こゝに2次の差分商 e_{ij} は次式で与えられる。

$$e_{ij} = \frac{c_{i,j+1} - c_{ij}}{y_{j+1} - y_j} - \frac{d_{i+1,j} - d_{ij}}{x_{i+1} - x_i} \quad (11)$$

この方法によって得られた曲線は、双3次スプライン内挿より滑らかであるが、多値関数のときは良い結果が得られていない。²²⁾

4. Akimaの内挿法プログラムの使い方

最初に $Y = F(X)$ 型の内挿法プログラムとして、「INTRPL」と「CURVFT（原版名CRVFIT）」が用意されている。INTRPLは、内挿点のみを求めるとき向きで、CURVFTはプロッター向きである。

$Z = F(X, Y)$ 型の内挿法プログラムとしては、「ITPLBV」と「SFCFIT」が用意されていて、 $Y = F(X)$ 型内挿法のときと同様、前者が内挿点のみを求める向きで、後者がプロッター向きである。

いづれのプログラムもサブルーチン形式で用いられるが、現在、日本原子力研究所の科学用サブルーチン・ライブラリー（窗口は計算センター）に登録されていて、一般ユーザーに公開されている。尚、用語の説明は、断りなしにFORTRAN言語の規則（整数、実数の英文字、掛算の記号など）に従ってなされている。

4.1 $Y = F(X)$ 型の内挿プログラム INTRPL

用い方は、サブルーチン形式なので以下のようにする。

CALL INTRPL (IU, L, X, Y, N, U, V)

ここで、入力パラメーターは

IU = 標準出力プリントのファイル・ユニットで、普通は6とする。

L = 入力データ点の数 (≥ 2)。

X = 入力データ点の x 座標値。ディメンジョン L の配列。増加する順序で入力。

Y = 入力データ点の y 座標値。ディメンジョン L の配列。

N = 求めたい内挿値 $y = f(x)$ の数 (≥ 1)。

U = 求めたい内挿値 $y = f(x)$ の x 座標値。ディメンジョン N の配列。

出力パラメーターは

V = 求めたい内挿値、 $y = f(x)$ の結果。ディメンジョン N の配列。

用法の注意としては、

(1) このサブルーチンでは、データ点が意味する関数系（内挿関数系）が一価関数であること。

(2) 内挿関数系が周期関数であるときは、両方の端点に 2 個づつのデータ点、合計 $L+4$ 個を入力する。

Fig. 4.1 は、使用例をメイン・プログラムと共に示したものである。計算結果も一緒に示してあるが、一番最初にプリントされるブロックは入力データのカード入力そのまゝの形式のプリント（メイン・プログラムで CALL DTLIST で呼び出されている。既にライブラリー³⁴⁾に登録されている）で、その後に内挿結果が示されている。この例題での実行時間は 0.065 秒であった。

4.2 $Y = F(X)$ 型の曲線あてはめプログラム CURVFT

```

*****,1.....*,2.....*,3.....*,4.....*,5.....*,6.....*,7.....*,8

      A111.

      T,0,TIME 105
      C,1,CORE 64
      W,O/PAGE 40
      P,O/PCH 0
      /00331

      *C JOB 8881223,T,ISE+431,12,SHINER
      *C B=MAP
      *C PROGRAM INTRPL
      *C INTERPOLATION,H,AKIMA,CACH,15,10,914-918(1972),A433
      *C Y=F(X)
      *C DIMENSION A(18),X(100),Y(100),U(200),V(200)

      C REWIND 1
      C REWIND 2
      CALL DT1LIST
      READ (5,1000) A(1),L=1,L8
      WRITE(6,1000)(A(I),I=1,18)
      READ (5,1010) IU,LMAX,NMAX
      WRITE(6,1015) IU,LMAX,NMAX
      READ (5,1020) (XL,I),L=1,LMAX
      WRITE(6,1017) (I,I)=1,10,
      WRITE(6,1023) (XC(L),L=1,LMAX)
      WRITE(6,1025) (XX(L),L=1,LMAX)
      READ (5,1020) (YC(L),L=1,LMAX)
      WRITE(6,1030) (YL,I),I=1,LMAX
      READ (5,1025) (Y(L),L=1,LMAX)
      READ (5,1020) (VN,N),N=1,NMAX
      WRITE(6,1017) (I,I)=1,10,
      WRITE(6,1033) (VN,N),N=1,NMAX
      IF(IU,EQ,0) IU=6

      C CALL CLOCK(I1)
      CALL INTRPL(IU,MAX,X,Y,NMAX,U,V)
      CALL CLOCK(I1)
      CALL CLOCK(I2)
      WRITE(6,1052) (VN,N),N=1,NMAX
      IT = I12,I1
      WRITE(6,1025) (VN,N),N=1,NMAX
      WRITE(1) LMAX,(XL,I),L=1,LMAX,(YC(L),L=1,LMAX)
      WRITE(1) NMAX,(VN,N),N=1,NMAX,(V(N),N=1,NMAX)
      5 FORMAT(5X,3H TIME DURING INTRPL-ROUTINE (MSEC),=16)
      1000 FORMAT(1.8A4)
      1010 FORMAT(3I6)
      1015 FORMAT(5X,4H IU=13,3X,6H LMAX=13,3X,6H NMAX=13)
      1017 FORMAT(10(5X,12,5X))
      1020 FORMAT(16E12,5)
      1023 FORMAT(1/58X,2H X)
      1025 FORMAT(10(1PE12,4))
      1033 FORMAT(1/58X,2H Y)
      1043 FORMAT(1/58X,2H U)
      1053 FORMAT(1/58X,2H V)
      STOP

```

Fig. 4.1 Input example for interpolating program INTRPL
of uni-variate function

```

*****1.....*....2.....*....3.....*....4.....*....5.....*....6.....*....7.....*....8

*DISK F01
*DISK F02
*DATA
AN EXAMPLE OF CRVFIT(ND=1)
      10   46
      0.0   1.0   2.0   3.0   4.0   5.0
      6.0   7.0   8.0   9.0
      0.0   0.0   0.0   0.0
     10.0   80.0  100.0 150.0
      0.00   0.20   0.40   0.60   0.80   1.00
      1.20   1.40   1.60   1.80   2.00   2.20
      2.40   2.60   2.80   3.00   3.20   3.40
      3.60   3.80   4.00   4.20   4.40   4.60
      4.80   5.00   5.20   5.40   5.60   5.80
      6.00   6.20   6.40   6.60   6.80   7.00
      7.20   7.40   7.60   7.80   8.00   8.20
      8.40   8.60   8.80   9.00
*JEND

```

Fig. 4.1 (b)

用い方は、

CALL CURVFT (IU, MD, L, X, Y, M, N, U, V)

入力パラメーターは、

IU = 標準出力プリントのファイル・ユニットで、普通は 6 とする。

MD = 内挿曲線の多値性を示す。

= 1 : 一価関数のとき、

= 2 : 多値関数のとき、

L = 入力データ点の数 (≥ 2)。

X = 入力データ点の x 座標値。ディメンジョン L の配列。MD = 1 のときは、増加する順序か、或いは減少する順序で入力。

Y = 入力データ点の y 座標値。ディメンジョン L の配列。

M = 入力データ点間を内挿するときの等分割数 (≥ 2)。

N = 入力データ点も含めた内挿結果の値としての出力点の数 (x 座標値を与える)。

= $(L - 1) * M + 1$

出力パラメータは

U = 内挿結果の出力点の x 座標値。ディメンジョン N の配列。

V = 内挿結果の出力点の y 座標値。ディメンジョン N の配列。

このプログラムの特徴および用法の注意としては、

- (1) 入力データ点間の等分割数を入力とするだけで滑らかな曲線が得られる。従って、プロッタなどの図形処理向きである。
- (2) 一価関数でも多値関数でも扱える。
- (3) CALL CURVFT (IU, MD, L, X, Y, M, N, X, Y) としても用いることができるが、この場合は入力データ (X, Y) は保存されない。
- (4) 周期関数 (閉曲線も含めて) に対しては、両方の端点に、2 個づつのデータ点を余分に入力する。

Fig. 4.2 に使用例を計算結果と共に示してある。

4.3 Z = F(X, Y) 型の内挿プログラム ITPLBV

用い方は

CALL ITPLBV (IU, LX, LY, X, Y, Z, N, U, V, W, LL)

入力パラメーターは、

IU = 標準出力プリントのファイル・ユニットで、普通は 6 とする。

LX = 入力格子点の x 座標点の数 (≥ 2)

LY = 入力格子点の y 座標点の数 (≥ 2)

X = 入力格子点の x 座標値。ディメンジョン LX の配列。増加する順序で入力。

Y = 入力格子点の y 座標値。ディメンジョン LY の配列。増加する順序で入力。

```

***.*.....1.....*,....2.....*,....3.....*,....4.....*,....5.....*,....6.....*,....7.....*,....8

      B111,          / T,0/TIME 10S
      *HFOR T,1SE,431,12,SHINER
      *MAP C,1/CORE 64
      MAIN OF CURVFT W,0/PAGE 40
      INTERPOLATION,M,AKIMA,CACM,15,10,914-918(1972),A433
      Y=F(X) P,0/PCN 0
      FITS A SMOOTH CURVE TO A GIVEN DATA POINTS
      DIMENSION A(18),X(100),Y(100),U(300),V(200)
      REWIND 1
      REWIND 2
      CALL DTLIST
      READ C$,(1,000)(AC1),I=1,18)
      WRITE C$,(1,000)(AC1),I=1,18)
      READ C$(,1010) IUM,ND,LMAX,M
      NMAX=(LMAX-1)*M+1
      WRITE C$(,1015) IUM,ND,LMAX,M,NMAX
      READ C$(,1020) (X(L)),L=1,LMAX)
      WRITE C$(,1017) (I,I=1,10)
      WRITE C$(,1023) (Y(L)),L=1,LMAX)
      WRITE C$(,1025) (X(L2),L=1,LMAX)
      READ C$(,1020) (Y(L2),L=1,LMAX)
      WRITE C$(,1033)
      WRITE C$(,1025) (Y(L2),L=1,LMAX)
      WRITE C$(,1025) (Y(L2),L=1,LMAX)
      IF(IU,.0,0) IUM=6
      CALL CLOCK(MIT1)
      CALL CURVFT(IUM,MD,LMAX,X,Y,M,NMAX,U,V)
      CALL CLOCK(MIT2)
      WRITE C$(,1017) (I,I=1,10)
      WRITE C$(,1043) (VN(N),N=1,NMAX)
      WRITE C$(,1025) (VN(N),N=1,NMAX)
      WRITE C$(,1053) (VN(N),N=1,NMAX)
      WRITE C$(,1025) (VN(N),N=1,NMAX)
      IT=IT2-IT1
      WRITE (6,5) IT
      WRITE C$(,1) LMAX,(X(L)),L=1,LMAX),(Y(L)),L=1,LMAX)
      WRITE C$(,1) NMAX,LUM,VN(N),N=1,NMAX)
      WRITE C$(,1) VN(N),N=1,NMAX)
      5 FORMAT(5X,35H TIME DURING CURVFT=ROUTINE(MSEC),)=16)
      1000 FORMAT(1H)
      1010 FORMAT(1H)
      1015 FORMAT(5X,4H IUM,13,3X,4H MD,13,3X,6H LMAX,13,3X,3H N,13,3X,6H NMAX
      1*13)
      1017 FORMAT(10L,12,2X)
      1020 FORMAT (CE12,5)
      1023 FORMAT (L,5BX,2H X)
      1025 FORMAT (LOC1PE12,4))
      1033 FORMAT (L,5BX,2H )
      1043 FORMAT (L,5BX,2H U)
      1053 FORMAT (L,5BX,2H V)
      STOP

```

Fig. 4.2 Input example for curve-fitting program CURVFT
of uni-variate function

```

*****1*****2*****3*****4*****5*****6*****7*****8

      END
      *HLEDRUN   F01
      *DISK     F02
      *DATA
      AN EXAMPLE OF CRVFIT(MD=2)
      2    10      5      7
      -30,0  -30,0  -30,0  -16,0  0,0  18,0  xx
      30,0  30,0  30,0  30,0  0,0  4,0  yy
      70,0  40,0  20,0  4,0  0,0  4,0  yy
      20,0  40,0  50,0  70,0  0,0  0,0  yy
      -30,0  -30,0  -30,0  -30,0  -30,0  -30,0  cccc
      -30,0  -30,0  -30,0  -30,0  -30,0  -30,0  cccc
      -30,0  -30,0  -30,0  -30,0  -30,0  -30,0  cccc
      -27,66  -24,158  -21,001  -18,000  -14,641  -11,097
      -7,433  -3,713  0,0  3,713  7,333  11,997
      14,641  18,000  21,501  24,758  27,466  29,215
      30,000  30,000  30,000  30,000  30,000  30,000
      30,000  30,000  30,000  30,000  30,000  30,000
      30,000  30,000  30,000  30,000  30,000  30,000
      *END

```

Fig. 4.2 (b)

PAGE=0001

```

***** INPUT DATA LIST ****
AN EXAMPLE OF CRVFIT(MD=2)
  1  2   10      5      7          5          0,0        0,0        18,0      X
  2  -30,0    -30,0    -30,0     30,0      30,0      20,0      4,0        4,0
  3  30,0     30,0     30,0      40,0      40,0      50,0     70,0      Y
  4  70,0     40,0     40,0     40,0      50,0      50,0     70,0      X
  5  20,0    -30,0    -30,0     30,0      30,0     -30,0     30,0      X
  6  -30,0    -30,0    -30,0     30,0      30,0     -30,0     30,0      Y
  7  -27,466   -24,758   -24,758   -21,501   -18,000  -18,000 -14,641   -11,097
  8  -7,433    -3,713    -3,713    -0,0      0,0      3,713    24,758   -27,466
  9  14,641   16,000   16,000   21,501   21,501   21,501   24,758   29,355
 10  30,000   30,000   30,000   30,000   30,000   30,000   30,000  30,000
 11  30,000   30,000   30,000   30,000   30,000   30,000   30,000  30,000
 12  30,000   30,000   30,000   30,000   30,000   30,000   30,000  30,000
 13  30,000   30,000   30,000   30,000   30,000   30,000   30,000  30,000
 14  30,000   30,000   30,000   30,000   30,000   30,000   30,000  30,000
***** INPUT DATA END ****
AN EXAMPLE OF CRVFIT(MD=2)
  1  0   2   LMAX= 10   M= 5   NMAX= 46
  2  3       6      7      8      9      10
-3,0000E+01 -3,0000E+01 -3,0000E+01 -1,6000E+01 0,0        X  1,8000E+01  3,0000E+01  3,0000E+01  3,0000E+01
 7,0000E+01 4,0000E+01 2,0000E+01 4,0000E+00 0,0        Y  4,0000E+00  2,0000E+01  4,0000E+01  5,0000E+01  7,0000E+01
 1  2   3   4      5          6          7          8          9          10
-3,0000E+01 -3,0000E+01 -3,0000E+01 -3,0000E+01 -3,0000E+01 -3,0000E+01 -3,0000E+01 -3,0000E+01 -3,0000E+01 -3,0000E+01
-3,0000E+01 -2,9319E+01 -2,4759E+01 -2,1501E+01 -1,8000E+01 -1,4641E+01 -1,1097E+01 -80000E+01 -4,6441E+01 -1,1097E+01 -3,0000E+01
 0,0  3,7129E+00 7,4339E+00 1,1097E+01 1,4641E+01 1,8000E+01 2,1501E+01 2,4759E+01 2,7456E+01 2,9319E+01 2,9319E+00
 3,0000E+01 3,0000E+01
 3,0000E+01 3,0000E+01 3,0000E+01 3,0000E+01 3,0000E+01 3,0000E+01 3,0000E+01 3,0000E+01 3,0000E+01 3,0000E+01 3,0000E+01
 7,0000E+01 6,4000E+01 5,8000E+01 5,2000E+01 4,6000E+01 4,0000E+01 3,6000E+01 3,2000E+01 2,8000E+01 2,4000E+01 2,4000E+01
 2,0000E+01 1,6080E+01 1,2400E+01 9,0798E+00 6,2398E+00 4,6631E+00 4,0000E+00 3,3309E+00 2,4631E+00 2,4631E+00 2,4631E+00
 0,0  1,3577E-01 5,6730E-01 1,3309E+00 2,4631E+00 4,0000E+00 6,2398E+00 9,0798E+00 1,3309E+00 5,6730E-01 1,3577E-01
 2,0000E+01 2,4000E+01 2,8000E+01 3,2000E+01 3,6000E+01 4,0000E+01 4,4000E+01 4,2000E+01 4,2000E+01 4,0000E+01 4,0000E+01
 5,0000E+01 5,4000E+01 5,8000E+01 6,2000E+01 6,6000E+01 7,0000E+01
TIME DURING CURVFT ROUTINE (MSEC.) = 1

```

Fig. 4.2 (c)

Z = 入力格子点の z 座標値。ディメンジョン [LX, LY] の配列。

N = 求めたい内挿値 $z = f(x, y)$ の数 (≥ 1)。

U = 求めたい内挿値 $z = f(x, y)$ の x 座標値。ディメンジョン N の配列。

V = 求めたい内挿値 $z = f(x, y)$ の y 座標値。ディメンジョン N の配列。

LL = 入力格子点の z 座標値の配列に対する整合寸法 ($\geq \max(LX, LY)$)。

出力パラメーターは、

W = 内挿点の値 (z 座標値)。ディメンジョン N の配列。

用法の注意としては

(1) 内挿関数系が一価関数であること。

(2) 内挿関数系が、x 座標、或いは y 座標の周期関数のときは、各々の座標の端点に、2 個づつの余分のデータ点を入力する。

Fig. 4.3 に使用例を計算結果と共に示してある。又、この例での実行時間は 0.1 秒であった。

4.4 $Z = F(X, Y)$ 型の曲面あてはめプログラム SFCFIT

用い方は、

CALL SFCFIT(IU, LX, LY, X, Y, Z, MX, MY, NU, NV, U, V, W, LL, NN)

入力パラメーターは

IU = 標準出力プリントのファイル・ユニットで、普通は 6 とする。

LX = 入力格子点の x 座標点の数 (≥ 2)

LY = 入力格子点の y 座標点の数 (≥ 2)

X = 入力格子点の x 座標値。ディメンジョン LX の配列。増加する順序か減少する順序で入力。

Y = 入力格子点の y 座標値。ディメンジョン LY の配列。増加する順序か減少する順序で入力。

Z = 入力格子点の z 座標値。ディメンジョン [LX, LY] の配列。

MX = x 座標に対する、入力データ点間を内挿するときの等分割数 (≥ 2)。

MY = y 座標に対する、入力データ点間を内挿するときの等分割数 (≥ 2)。

NU = x 座標に対する、入力データ点も含めた出力点の数。

$$= (LX - 1) * MX + 1.$$

NV = y 座標に対する、入力データ点も含めた出力点の数。

$$= (LY - 1) * MY + 1.$$

LL = 入力格子点の z 座標値の配列に対する整合寸法 ($\geq \max(LX, LY)$)。

NN = 出力格子点 (内挿結果) の z 座標値の配列に対する整合寸法 ($\geq \max(NU, NV)$)。

出力パラメーターは、

U = 内挿結果である出力点 $z = f(x, y)$ の x 座標値。ディメンジョン NU の配列。

V = 内挿結果である出力点 $z = f(x, y)$ の y 座標値。ディメンジョン NV の配列。

W = 内挿点 $z = f(x, y)$ の値。ディメンジョン [NU, NV] の配列。

```

*****.*****.1.*****.2.*****.3.*****.4.*****.5.*****.6.*****.7.*****.8

      *NO    A222,          /
      *NFOB   8881223,T,1SE,431,12,SHINER
      *NMAP
      PROGRAM ITPLBV
      C   INTERPOLATION,H,AKIMA,CACM,17,1,26=31(1974),A474
      C   Z=X,Y
      C   DIMENSION X(30),Y(30),Z(30,30),U(50),V(50),W(50),A(18)
      C
      READ (5,1000) (A(I,J),I=1,18)
      WRITE(6,1000) (A(I,J),I=1,18)
      READ (5,1010) IU,LXMAX,LYMAX,NMAX
      WRITE(6,1015) IU,LXMAX,LYMAX,NMAX
      READ (5,1020) (X(I,J),I=1,LXMAX)
      WRITE(6,1020) (X(I,J),I=1,LXMAX)
      READ (5,1020) (Y(I,J),I=1,LYMAX)
      WRITE(6,1020) (Y(I,J),I=1,LYMAX)
      DO 10 LX=1,LXMAX
      READ (5,22) (Z(LX,LY),LY=1,LYMAX)
      WRITE(6,22) (Z(LX,LY),LY=1,LYMAX)
10  CONTINUE
      READ (5,22) CU(ND),N=1,NMAX
      WRITE(6,22) (CU(ND),N=1,NMAX)
      READ (5,22) CV(ND),N=1,NMAX
      WRITE(6,22) (CV(ND),N=1,NMAX)
      C
      WRITE(6,40)
      WRITE(6,42) (1,I=1,10)
      WRITE(6,44) (Y(I,L),L=1,LYMAX)
      WRITE(6,46) (Y(I,L),L=1,LYMAX)
      C
      DO 50 LX=1,LXMAX
      WRITE(6,52) LX,X(LX),(Z(LX,LY),LY=1,LYMAX)
50  CONTINUE
      IF (IU,EQ,0) IU=6
      C
      CALL CLOCK(MIT1)
      CALL ITPLBV(IU,LXMAX,LYMAX,X,Y,Z,NMAX,U,V,W,.30)
      CALL CLOCK(MIT2)
      IT=IT2-IT1
      C
      WRITE(6,60)
      WRITE(6,62)
      WRITE(6,64) CU(ND),V(ND),W(ND),N=1,NMAX
      WRITE(6,5) IT
      C
      1000 FORMAT(1.8A4,.)
      1010 FORMAT(4I6)
      1015 FORMAT(5X,4H IU=13,3X,7H LXMAX=13,3X,7H LYMAX=13,3X,6H NMAX=13)
      1020 FORMAT(6E12.5)
      22  FORMAT(6E12.5)
      40  FORMAT(43X,9H Z(LX,LY))

```

Fig. 4.3 Input example for interpolating program ITPLBV
of bi-variate function

3.00000E+01 3.00000E+01 0.0
 3.25000E+01 3.25000E+01 0.0
 3.50000E+01 3.50000E+01 0.0
 3.75000E+01 3.75000E+01 0.0
 4.00000E+01 4.00000E+01 0.0
 TIME DURING ITP_LBV=ROUTINE (MSEC.) = 4

Fig. 4.3 (d)

このプログラムの特徴および用法の注意としては、

- (1) 入力データ点間の等分割数を入力するだけで滑らかな曲面が得られるので、プロッターなどの図形処理向きである。
- (2) 内挿関数系が多価関数的でも扱える（原著者は多価関数に対しては否定的な表現をしている。この件に関しては次章のベンマーク・テストの結果を参照されたい）。
- (3) 内挿関数系が x 座標、或いは y 座標の周期関数のときは、各々の座標の端点に、2個づつの余分のデータ点を入力する。

Fig. 4.4 に使用例を計算結果と共に示してある。又、この例での実行時間は 0.2 秒であった。

```

* * * * * 1 * * * * 2 * * * * 3 * * * * 4 * * * * 5 * * * * 6 * * * * 7 * * * * 8
* * * * * 1 * * * * 2 * * * * 3 * * * * 4 * * * * 5 * * * * 6 * * * * 7 * * * * 8

MNO      B222,
          /TIME_10S_
          /T+O/TIME_10S_
          /C_1/CORE_64
          /W_0/PAGE_40
          /P_0/PCH_0
          /F0331

*6.JOB  8881223,T,1SE,431,12,SHINER
*6.FORT  BMAP
*6.PGM  PROGRAM SFCFIT
*6.C    INTERPOLATION,HAKIMA,CACM,17,1,26-31(1974),A474
*6.C    Z=Z(X,Y)
*6.C    DIMENSION X(30),Y(30),Z(30,30),U(50),V(50),W(50,50)
*6.C    DIMENSION A(18)

C    READ (5,1000) (A(I),I=1,18)
C    WRITE(6,1000) (A(I),I=1,18)
C    READ (5,1010) LU,LXMAX,LYMAX,MX,MY
C    WRITE(6,1015) LU,LXMAX,LYMAX,MX,MY
C    READ (5,1020) (XL,X),LX=1,LXMAXJ
C    WRITE(6,1020) (XL,X),LX=1,LXMAXJ
C    READ (5,1020) (XL,X),LX=1,LXMAXJ
C    READ (5,1020) (YL,Y),LY=1,LYMAXJ
C    WRITE(6,1020) (YL,Y),LY=1,LYMAXJ
DO 10 LX=1,LXMAX
  READ (5,1020) (Z(L,X,Y),LY=1,LYMAX)
  WRITE(6,1020) (Z(L,X,Y),LY=1,LYMAX)
10 CONTINUE

C    NUMAX=(LXMAX-1)*MX+1
C    NVMAX=(LYMAX-1)*MY+1

C    WRITE(6,40)
C    WRITE(6,42) (I,I=1,10)
C    WRITE(6,44)
C    WRITE(6,46) (Y(I),I=1,LYMAX)

C    DO 50 LX=1,LXMAX
      WRITE(6,52) LX,X(LX), (Z(L,X,Y),LY=1,LYMAX)
50  CONTINUE
  IF (LU.EQ.0) LU=6

C    CALL CLOCK(LT1)
CALL SFCFIT(LU,LXMAX,LYMAX,X,Y,Z,NX,MY,NUMAX,NUMAX,W,V,W,30,50)
CALL CLOCK(LT2)
  LT=LT2-LT1

C    WRITE(6,54)
C    WRITE(6,60)
C    WRITE(6,42) (I,I=1,10)
C    WRITE(6,62)
C    WRITE(6,64) (V(N),N=1,10)
DO 70 N=1,NUMAX
  WRITE(6,52) N,U(N),(W(N,K),K=1,10)
70 CONTINUE
  WRITE(6,62)
  WRITE(6,64) (V(N),N=11,NUMAX)

```

Fig. 4.4 Input example for surface-fitting program
SFCFIT of bi-variate function


```

***.*****1.....*....2.....*....3.....*....4.....*....5.....*....6.....*....7.....*....8

DO 71 N=1,NUMAX
    WRITE(6,52) N,U(N),W(N+K),K=L1,NUMAX
71 CONTINUE
    WRITE(6,5) IT
    WRITE(6,5) IT

C 1000 FORMAT(18A4)
1010 FORMAT(5I6)
1015 FORMAT(5X,4H IUF=13,3X,7H LXMAX=13, 3X,7H LYMAX=13,3X,4H MX=13,3X,4
*H MY=13)
1020 FORMAT(6E12.5)
40 FORMAT(743X,9H Z(X,X,Y))
42 FORMAT(6X,10(3X,13.2X))
44 FORMAT(1X,2HUX,6H X(LX),7H Y(Y)=)
46 FORMAT(18X,10(2X,F5.2,1X))
52 FORMAT(1X,12.1X,F5.2,10(2X,F5.2,1X))
54 FORMAT(//21H INTERPOLATED RESULTS/)
60 FORMAT(743X,9H W(KX,KY))
62 FORMAT(4H UVW)
64 FORMAT(8X,10(2X,F5.2,1X))
5 FORMAT(5X,35H TIME DURING SFCFIT-ROUTINE(MSEC.),=16)
    STOP
END

*HIEDRUN
*DATA
AN EXAMPLE SET OF INPUT DATA Z((IX,IY) FOR SFCFIT
   6      11      9      2
   0.0      5.0     10.0     15.0     20.0     25.0     X
   30.0     35.0     40.0     45.0     50.0
   0.0      5.0     10.0     15.0     20.0     25.0     Y
   30.0     35.0     40.0
   58.2     61.5     67.9     73.3     79.7     85.5     Z(X,Y)
   38.2     41.2     47.7     54.2     60.7     67.2     Z(X,Y)
   37.2     40.0     47.0     54.0     61.0     68.0     Z(X,Y)
   27.3     30.2     37.0     44.0     51.0     58.0     Z(X,Y)
   27.4     22.5     29.0     36.0     43.0     50.0     Z(X,Y)
   21.8     26.5     32.1     38.6     45.1     51.6     Z(X,Y)
   0.8      0.6      0.6      0.6      0.6      0.6      Z(X,Y)
   16.8     14.4     12.0     10.0     8.0      6.0      Z(X,Y)
   0.1      0.0      0.0      0.0      0.0      0.0      Z(X,Y)
   12.0     8.0      5.3      2.9      0.6      0.0      Z(X,Y)
   0.0      0.0      0.0      0.0      0.0      0.0      Z(X,Y)
   7.4      4.8      1.4      0.1      0.0      0.0      Z(X,Y)
   0.0      0.0      0.0      0.0      0.0      0.0      Z(X,Y)
   3.2      0.7      0.0      0.0      0.0      0.0      Z(X,Y)
   0.0      0.0      0.0      0.0      0.0      0.0      Z(X,Y)
*JEND

```

Fig. 4.4 (c)

14	32.50	5.24	3.86	2.57	1.34	0.35	0.04	0.01	0.00	0.00	0.00
15	35.00	3.20	1.68	0.70	0.20	0.00	0.00	0.00	0.00	0.00	0.00
16	37.50	1.09	0.41	0.08	-0.01	0.00	0.00	0.00	0.00	0.00	0.00
17	40.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	42.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	45.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	47.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	50.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
UVW	25.00	27.50	30.00	32.50	35.00	37.50	40.00				
1	0.0	45.50	41.20	38.20	39.80	41.20	41.67	41.70			
2	2.50	34.99	30.29	27.23	28.95	30.46	30.99	31.08			
3	5.00	24.50	20.25	17.30	18.84	20.70	20.70	20.80			
4	7.50	14.73	10.48	7.34	8.35	9.26	9.58	9.68			
5	10.00	7.20	4.93	1.80	1.96	2.10	2.12	2.10			
6	12.50	6.60	3.41	1.17	0.93	0.75	0.68	0.62			
7	15.00	7.60	3.74	0.80	0.66	0.60	0.59	0.60			
8	17.50	3.61	1.72	0.39	0.28	0.22	0.21	0.22			
9	20.00	0.60	0.25	0.10	0.04	0.00	-0.01	0.00			
10	22.50	0.16	0.06	0.02	0.01	0.00	-0.00	0.00			
11	25.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
12	27.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
13	30.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
14	32.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
15	35.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
16	37.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
17	40.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
18	42.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
19	45.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
20	47.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
21	50.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
		TIME DURING SFCFIT-ROUTINE (MSEC.)	=					17			

Fig. 4.4 (d)

5. ベンチマーク・テストによる内挿法プログラムの評価

ベンチマーク・テストによるプログラムの評価は、大きなプログラムでは、プログラムの有用性の実証（verification）を行うのが主目的であろう。一方、我々がこゝで対象とした内挿法のような小さいプログラムでは、その解法アルゴリズムも把握しやすいので、プログラムの実証も比較的簡単である。更に、大勢の利用者が、これらのプログラムを用いるし、又、必ずしも各プログラムに精通してからプログラムを用いるとは限らないであろう（プログラム・使用手引書も、全すべての利用者にわかりやすく書かれているとは限らない）。そこで、こゝでは、アカデミックな興味からのみではなく、プログラム利用者にも良き手助けとなるように、ベンチマーク問題を設定し、各プログラムの評価を行った。

尚、この章で用いられた全てのプログラムについての使用例を附録A.1に示しておいたので合わせて利用されたい（但し、第4章で既に掲げた例は除いてある）。

5.1 $\mathbf{Y} = \mathbf{F}(\mathbf{X})$ 型の内挿法プログラム

この節でベンチマーク・テストの対象として採り上げた内挿法のプログラムは、以下に示した如くで、いづれもサブルーチン形式（CALL □□□）で使用できるものばかりである。

LAGS⁴⁾ は、全データ区間にひとつの多項式をあてはめる、Lagrange 補間法プログラムである。このように一度に高次の式を作らずに、データ間の1次内挿をくり返し行う、いわゆる反復（線形内挿）型にしたのがAKLAGプログラムである。

CONFRS プログラムは、連分数（continued fraction）をあてはめ関数とするが、この関数をPadé型の有理関数に変換し、その係数を逆差分商を用いて反復的に解いていく、反復型に属する内挿法である。SPLINS も反復型に属するプログラムであるが⁴⁾、あてはめ関数としては、2階微分まで連続微分可能な3次区分多項式、即ち、3次自然スプライン、を用いた方法で、途中で差分商が使われている。INTRPL およびCURVFT プログラムは、既に、第3章で説明がなされているように、連続1階微分可能な3次区分多項式を解析幾何的に逐次定めて行く方法である。

この他によく計算センターなどに登録されているプログラムとしては、Aitken-Hermite内挿法（AKHER⁵⁾）およびChebyshev 内挿法（CHBCFS⁴⁾）があるが、Aitken-Hermite 内挿法はデータ点の微係数値も必要であり、また、Chebyshev 内挿法は、データ点のとり方が予め定められているので、どちらも一般的な内挿法としては、使い難いので、敢えて省略した。

尚、反復型（AKLAG, CONFRS, SPLINS）のプログラムは、入力データとして、関数値の他に、反復解法における収束判定因子（EPS, convergence criterion）が必要である。最後に、こゝで採り上げた内挿法プログラムの一覧表を、以下の説明の便宜上、Table 5.1に掲げておく。

1次元内挿法のベンチマーク問題としては、Runge の問題が有名である。これは対称な $y = 1 / (1 + 25x^2)$ (Gauss 曲線或いはLorentz 曲線でもよい) に対して、等間隔にデータ点を与えたときで、我々はこゝでは、 $-1.0 \leq x \leq 1.0$ の範囲の曲線に対して議論をしてゆく。

Table 5.1 Interpolation programs used for benchmark tests

	Subroutine name	Method	Essence of the method	Library or author
Y=F(X)	LAGS	Lagrange	directly interpolate through a Lagrange polynomial	SSL (Fujitsu Ltd.) ⁴⁾
	AKLAG	Aitken-Lagrange	repeat linear interpolations iteratively using the divided difference	SSL-II (Fujitsu Ltd.) ⁵⁾
	CONFRS	Continued fraction	interpolate iteratively through the continued fraction analogously to Aitken-Lagrange method	SSL (Fujitsu Ltd.) ⁴⁾
	SPLINS	Spline	interpolate through the natural cubic spline function by solving a tridiagonal linear system resulting from the continuity of spline	SSL (Fujitsu Ltd.) ⁴⁾
Z=F(X,Y)	INTRPL	Akima's interpolation	interpolate analytically and geometrically through a cubic piecewise polynomial using 5 data points basically	Akima ^{20, 21)}
	CURVFT	Akima's curve-fitting	smoothing subroutine (bivariate spline interpolation) in "THREE-D: perspective drawing program"	CALCOMP Co. ¹⁶⁾
	SMTH	Bi-variate spline	extension of Akima's method to bivariate interpolation	Akima ^{22, 23)}
	SFCFIT	Akima's surface-fitting		

Fig. 5.1 は、厳密解の曲線 ($-1.0 \leq x \leq 1.0$ で 101 点のデータを直線で結んだもの) を与えたもので以下の種々の内挿法のテストに対する参考曲線である。尚、以下の曲線の図に現われてくる記号で、□印は入力データ点を表わし、△印は内挿結果の点を表わしている。

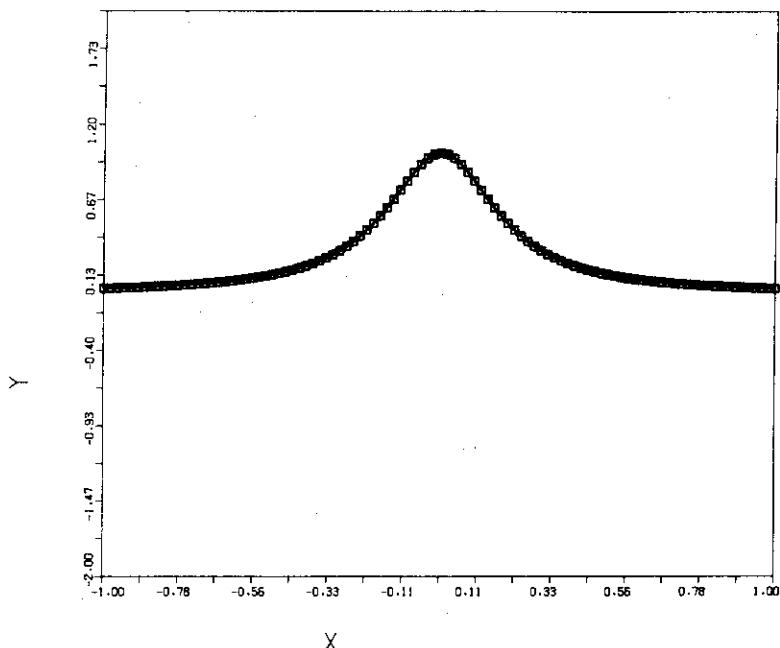


Fig. 5.1 Exact curve of $y=1/(1+25x^2)$ for Runge's tests

Fig. 5.2.1 ~ Fig. 5.2.4 のシリーズは、Lagrange 内挿法の、それぞれ、データ点が 5 点、9 点、21 点および 41 点が与えられたときの結果で、曲線もそれぞれ、5 次、9 次、21 次および 41 次の多項式で示されたものである。Runge が指摘した如く、高次多項式ひとつであてはめを行ったときの異常振動 (Runge の現象と云う) が現われている。

Fig. 5.3.1 ~ Fig. 5.3.4 (EPS = 1×10^{-6} のとき) および Fig. 5.4.1 ~ Fig. 5.4.4 (EPS = 0.1 のとき) は Aitken-Lagrange 法の結果であるが、EPS の値に関係なく、全く同様の結果を示している。1 次内挿のくり返しであるので高次 (データ点の多いもの) の方が良い内挿値を示していることがわかる。対称性も良く、Table 5.2 に示す如く、データ点数 21 以上では、実用上充分な精度を保証する。

次に、連分数内挿法の結果について述べる。Fig. 5.5.1 ~ Fig. 5.5.4 は EPS が 1×10^{-6} のオーダー附近のときの結果であるが、データ点が 41 点のときは、大変良い内挿結果を示している。データ点が 41 点のときは、引数値の大きい値の方で内挿の結果が悪くなっているが、これはこの方法が、インデックスの小さい方 (引数値の小さい方) から、逆差分をとりながら反復的に解いてゆくからで、インデックスの大きい方 (引数値の大きい方) へ向って、逆差分からの誤差が集積してゆくからである。従って、データ点が多くなると、この傾向が早めに現われてくる。図からは判定し難いが Table 5.3 に示す如く、上記の傾向が、21 データ点 (の引数値 0.68 周りから) から現われていることがわかる。Table 中の ILL で示されるコンディション・コード番号 (resultant condition code number) は、内挿結果の状態を示すものであるが、必ずしも ILL =

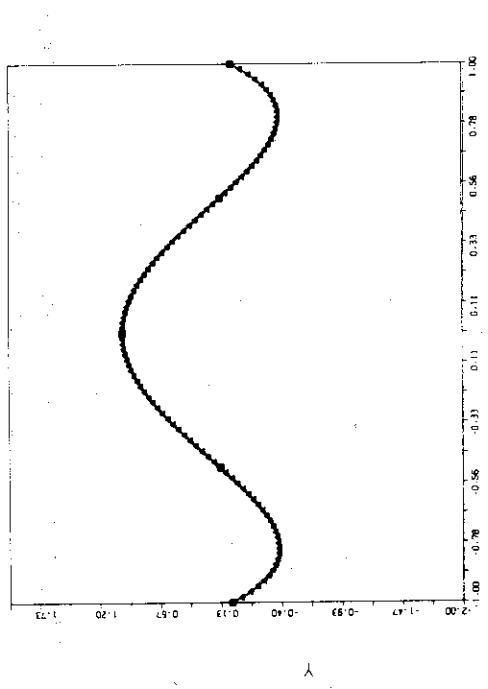


Fig. 5.2.1 5 point curve by Lagrange method (\square : given data points, Δ : interpolated points, the same to the following figures)

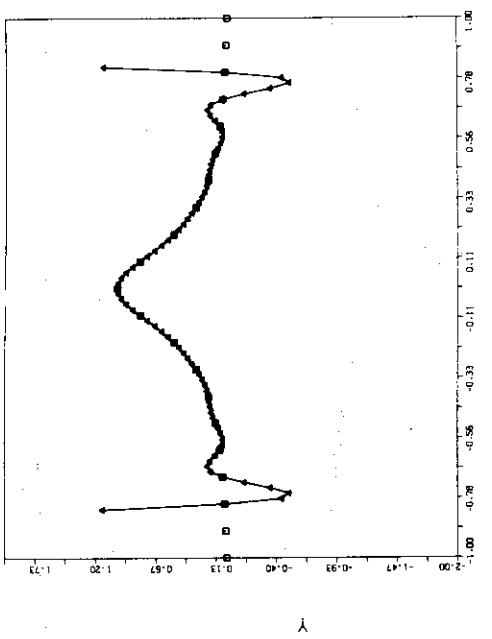


Fig. 5.2.3 21 point curve by Lagrange method

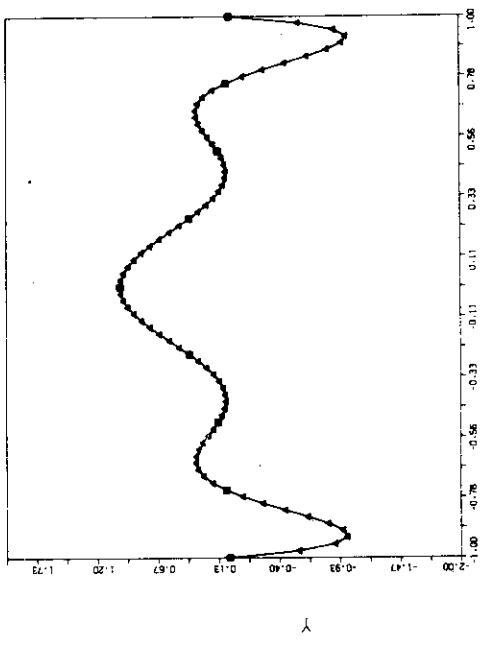


Fig. 5.2.2 9 point curve by Lagrange method

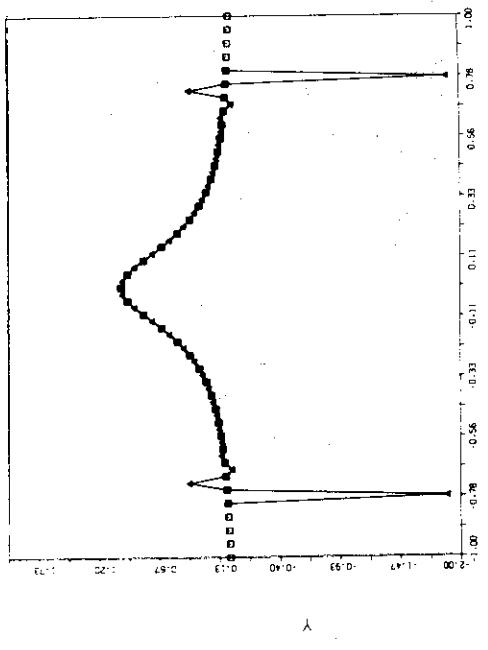


Fig. 5.2.4 41 point curve by Lagrange method

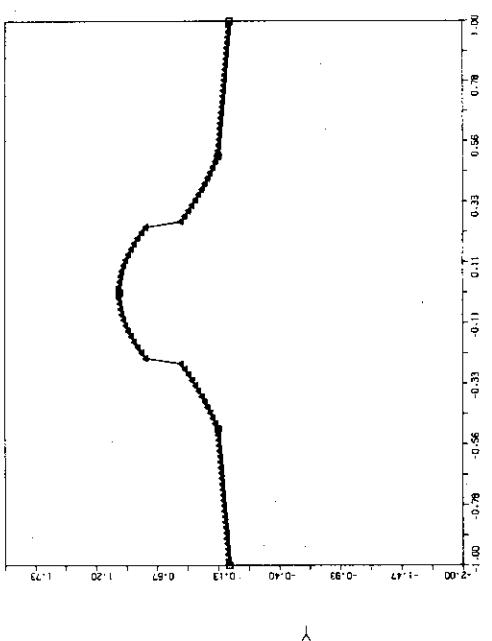


Fig. 5.3.1 5 point curve by Aitken-Lagrange method with convergence criterion of 1.0×10^{-6}

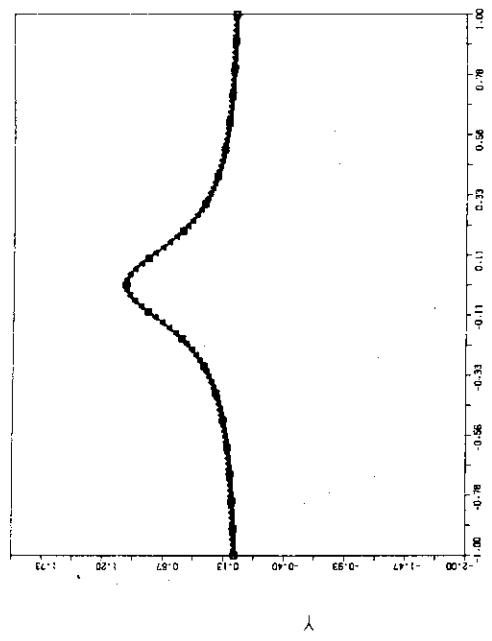


Fig. 5.3.3 21 point curve by Aitken-Lagrange method with convergence criterion of 1.0×10^{-6}

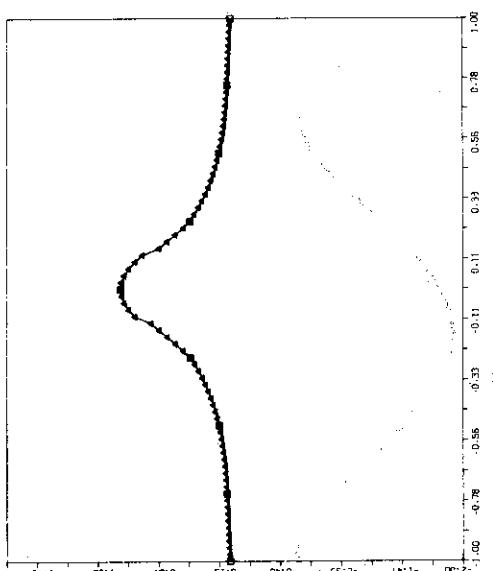


Fig. 5.3.2 9 point curve by Aitken-Lagrange method with convergence criterion of 1.0×10^{-6}

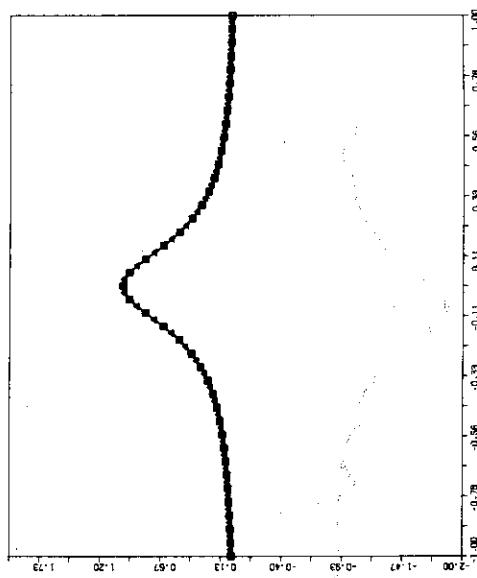


Fig. 5.3.4 41 point curve by Aitken-Lagrange method with convergence criterion of 1.0×10^{-6}

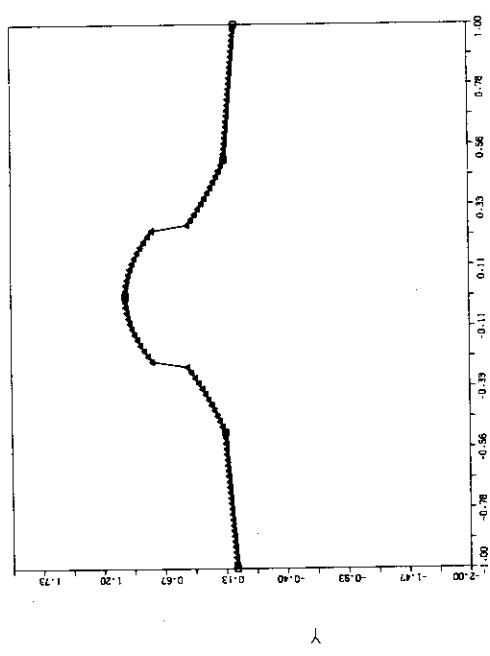


Fig. 5.4.1 5 point curve by Aitken-Lagrange method with convergence criterion of 0.1

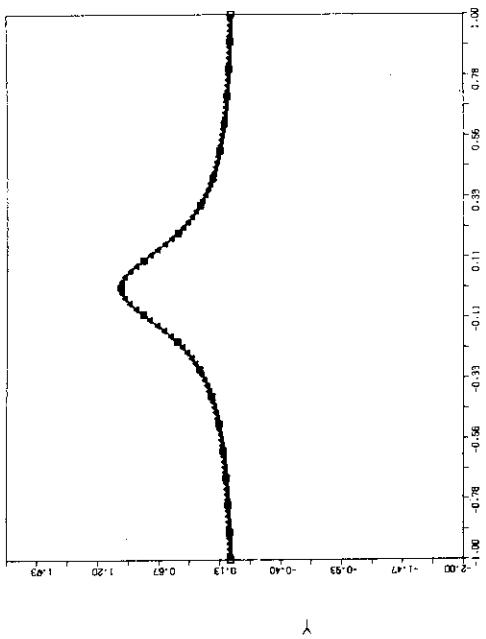


Fig. 5.4.3 21 point curve by Aitken-Lagrange method with convergence criterion of 0.1

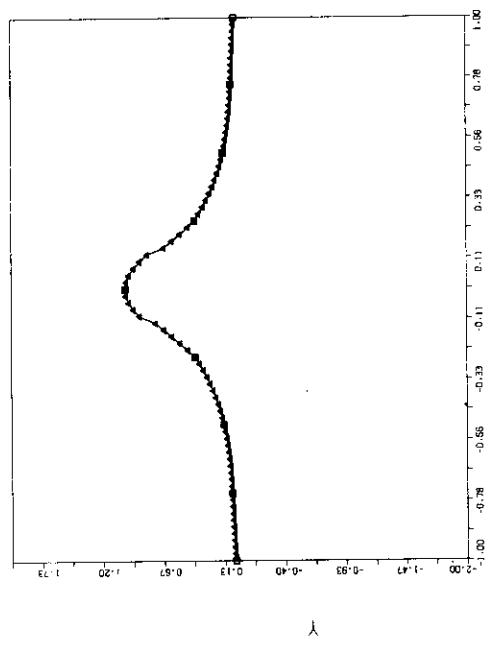


Fig. 5.4.2 9 point curve by Aitken-Lagrange method with convergence criterion of 0.1

Fig. 5.4.4 41 point curve by Aitken-Lagrange method with convergence criterion of 0.1

Table 5.2 Preciseness of the Aitken-Lagrange method as a function of data points for Runge's problem

Number of data points	Arguments and their interpolated results					
	-0.68	-0.18	0.18	0.68	-0.675	-0.175
Exact	7.9618×10^{-2}	5.5249×10^{-1}	5.5249×10^{-1}	7.9618×10^{-2}	8.0706×10^{-2}	5.6637×10^{-1}
	1.0212×10^{-1}	8.6490×10^{-1}	8.6490×10^{-1}	1.0212×10^{-1}		
5	1.0212×10^{-1}	8.6490×10^{-1}	8.6490×10^{-1}	1.0212×10^{-1}		
					7.7796×10^{-2}	5.9031×10^{-1}
9					7.7796×10^{-2}	5.9031×10^{-1}
						7.7796×10^{-2}
21	7.9618×10^{-2}	5.5363×10^{-1}	5.5363×10^{-1}	7.9618×10^{-2}	7.7796×10^{-2}	5.9031×10^{-1}
	7.9618×10^{-2}	5.5363×10^{-1}	5.5363×10^{-1}	7.9618×10^{-2}		
41					8.0706×10^{-2}	5.6638×10^{-1}
					8.0706×10^{-2}	5.6638×10^{-1}

Upper row = value for EPS = 1×10^{-6}

Lower row = value for EPS = 0.1

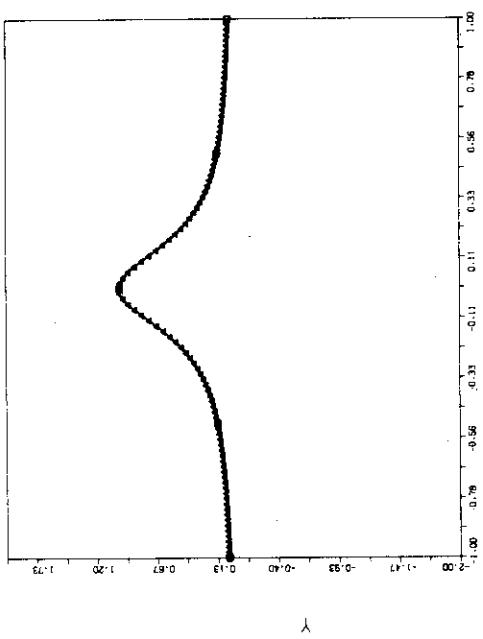


Fig. 5.5.1 5 point curve by the continued fraction method with convergence criterion of 1×10^{-5}

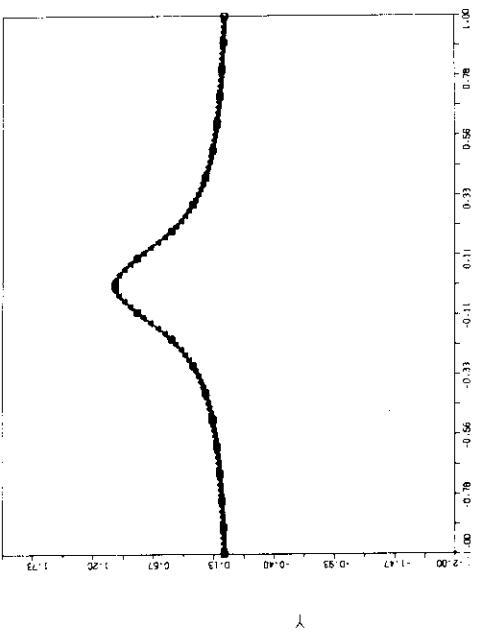


Fig. 5.5.3 21 point curve by the continued fraction method with convergence criterion of 1×10^{-5}

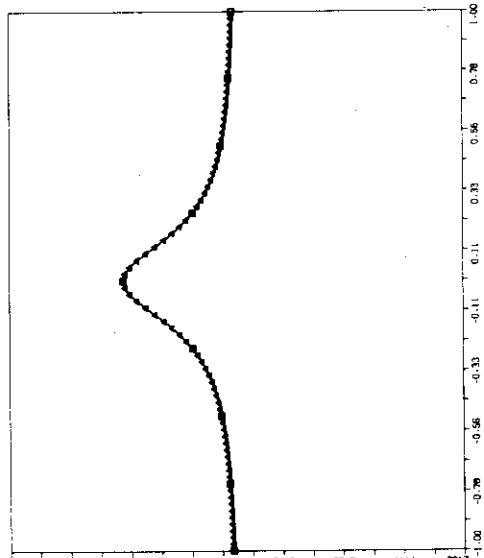


Fig. 5.5.2 9 point curve by the continued fraction method with convergence criterion of 1×10^{-5}

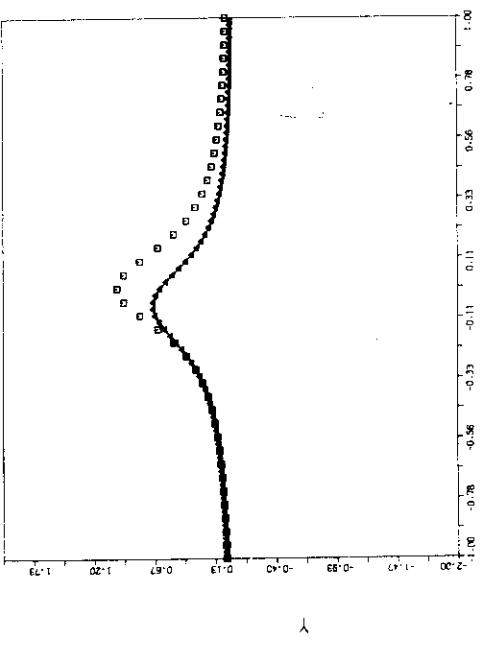


Fig. 5.5.4 41 point curve by the continued fraction method with convergence criterion of 1×10^{-5}

Table 5.3 Preciseness of the continued fraction method with convergence criterion (EPS) of 1×10^{-5} as a function of data points for Runge's problem

Number of data points	Arguments and their interpolated results						Resultant condition code; ILL
	-0.68	-0.18	0.18	0.68	-0.675	-0.175	
Exact	7.9618×10^{-2}	5.5249×10^{-1}	5.5249×10^{-1}	7.9618×10^{-2}	8.0706×10^{-2}	5.6637×10^{-1}	8.0706×10^{-2}
5	7.9618×10^{-2}	5.5248×10^{-1}	5.5248×10^{-1}	7.9618×10^{-2}			29000
9					8.0706×10^{-2}	5.6637×10^{-1}	
21	7.9618×10^{-2}	5.5249×10^{-1}	5.5249×10^{-1}	7.9629×10^{-2}			0
41					8.0706×10^{-2}	5.2831×10^{-1}	1.2477×10^{-2}
							28000

ILL is the resulting condition code number which is edited out of the program.

- ILL = 0 : normal end,
- = 28000 : not reached to the required precision because of computation error,
- = 29000 : not reached to the required precision due to lack of data points.

0(正常終了)でなくても、良い結果を与えるときがあることを注意しておく。*)

Fig. 5.6.1～Fig. 5.6.4は、上記の問題で、収束判定因子を0.00001から0.1に緩めたときの結果であるが、いづれも連分数の次数の不足から明らかにおかしな結果を示しており、このことは、収束判定因子を小さめにとるべきであることを示唆している。尚、収束判定因子を0.00001の上下1桁づつ変動させても、内挿結果に変りはなかった。

Fig. 5.7.1～Fig. 5.7.4はスプライン内挿法の結果($\text{EPS} = 1 \times 10^{-6}$)である。この方法はデータ点間を3次区分多項式で結び付けていくので、図から明らかなように、データ点の多い方が良い内挿結果を与えていている。Table 5.4に示す如く、21データ点数以上から、実用上の精度が保証されるようになり、又、結果の対称性も良く再現されている。

Fig. 5.8.1～Fig. 5.8.4は、Akimaの方法の内挿ルーチンの方(INTRPL)の結果であるが、大体スプライン法と同様の結果を示している。両方法を比較すると、データ点数が少ないとときは、Akimaの方法の方が厳密解に近い数値を示し、データ点数が充分に多いときは、略同じ値であるが、若干の差(Table 5.5参照; 引数值=0.175に対し、スプライン法では0.008%の誤差があるのに対し、Akimaの方法では、0.09%の誤差がある)は存在する。しかしながら、この程度の高精度ならば、充分実用上の精度は保証されていると云えよう。尚、Akimaの方法の曲線あてはめルーチン(CURVFIT)のMD=1(一価関数)のときは、全ての内挿値が内挿ルーチンの結果と全く的一致を示した。

Akimaの方法のCURVFITルーチンは、多価関数系のとき(このときはMD=2を入力)に有力な手段となるが、このMD=2のときに対して、一価関数系のRungeの問題のテストの結果がTable 5.6に示されている。大体、INTRPLルーチンと同程度の精度を示しているが、局所的には若干精度が落ちるようである:

Table 5.7は、こゝでのテストに対する各プログラムの計算時間(msec; サブルーチンを呼び出す前後の間の時刻の差)を示したものである。上段が内挿に要した計算時間であり、下段は、内挿点の数とデータ点の数の和を示している。LAGS, AKLAG, CONFRSとその他のプログラムとでは計算時間に大きな差があるが、これはプログラミング上の事で、前者では内挿点を求める毎にメイン・プログラムから内挿サブルーチンを呼び出しているのに反し、後者では一度の内挿サブルーチンの呼び出しで、サブルーチン内で全部処理されるからである。

最初の3つの方法の中では、Lagrange法は計算時間は、データ点が少ないとときは中程度であるが、Fig. 5.2のシリーズで明らかなように、データ点間を広くとれないし、又、データ点が多くなると時間がかかるだけでなく、Rungeの現象を起して異常振動してしまう。総じて、他の方法より長所は少ない。Aitken-Lagrange法は、計算時間はLagrange法に次いで長い。Fig. 5.3シリーズに示した如く異常振動も起さないが、精度を保つには、沢山のデータ点が必要である。尚、EPSの0.00001と0.1のときは計算時間も略同じで精度も同程度(Table 5.2)であった。連分数法は、比較的計算時間は喰うが、Fig. 5.5シリーズに示した如く、データ点の数の広い範囲

*) 一般的に、プログラム用法解説書のコンディション・コードの説明は、少し言葉不足のように思われる。

**) しかしながら、計算時間は内挿の際に用いたデータ点の個数(このプログラムでは入力M)に比例するので、経済性を考えて入力する必要がある。

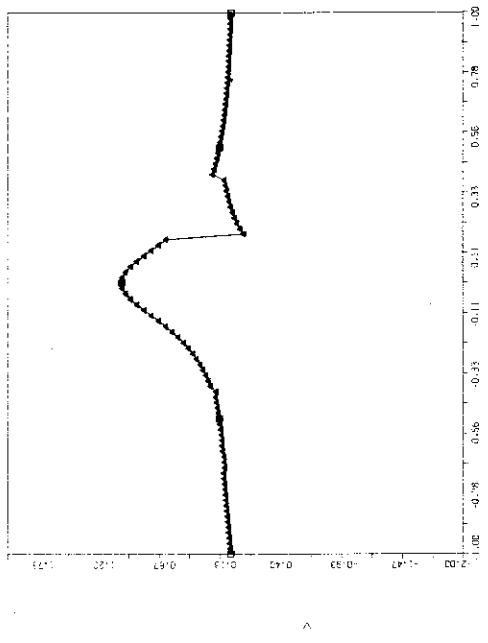


Fig. 5.6.1 5 point curve by the continued fraction method with convergence criterion of 0.1

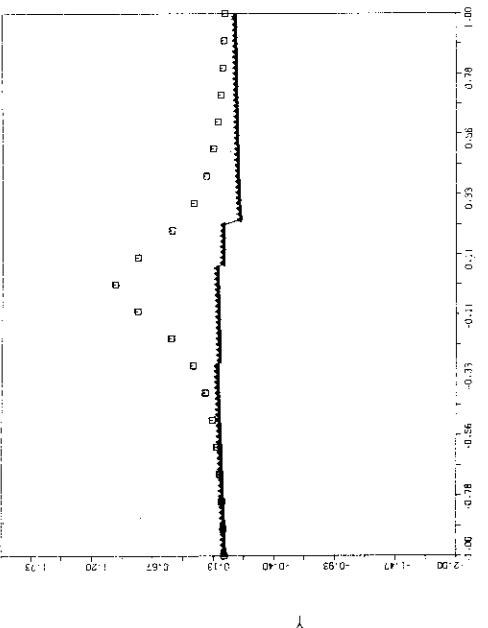


Fig. 5.6.3 21 point curve by the continued fraction method with convergence criterion of 0.1

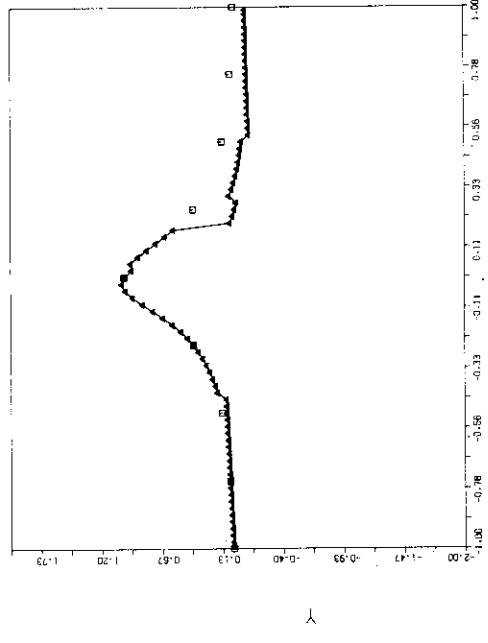


Fig. 5.6.2 9 point curve by the continued fraction method with convergence criterion of 0.1

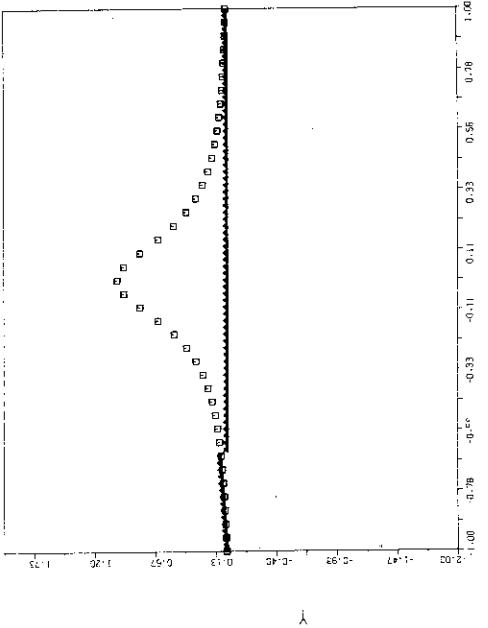


Fig. 5.6.4 41 point curve by the continued fraction method with convergence criterion of 0.1

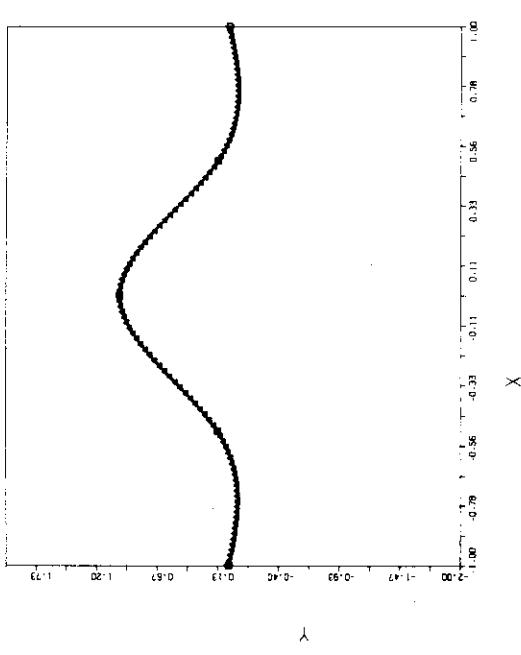


Fig. 5.7.1 5 point curve by the spline method

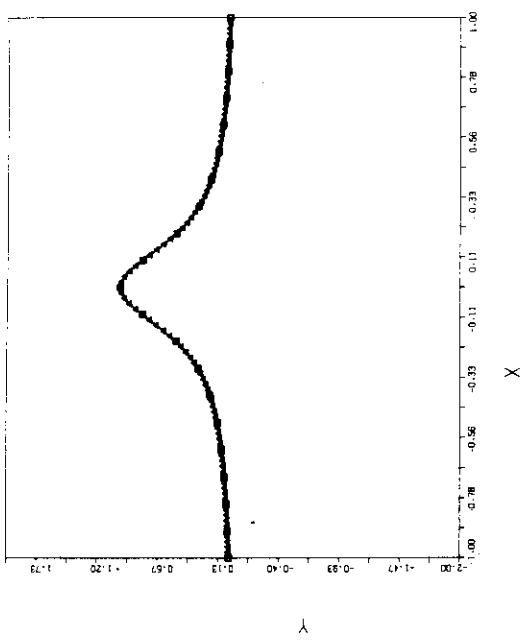


Fig. 5.7.3 21 point curve by the spline method

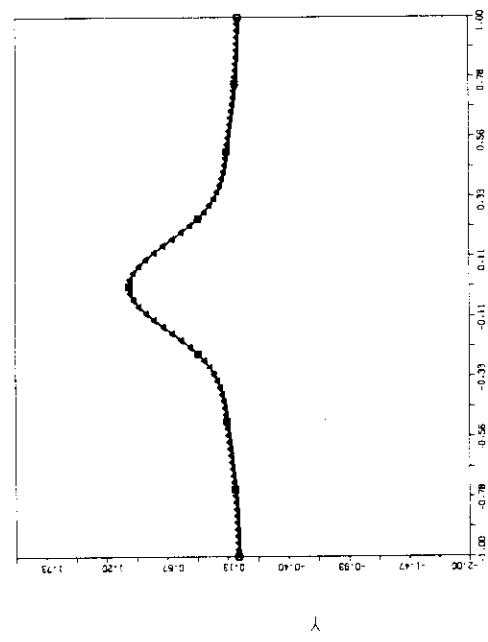


Fig. 5.7.2 9 point curve by the spline method

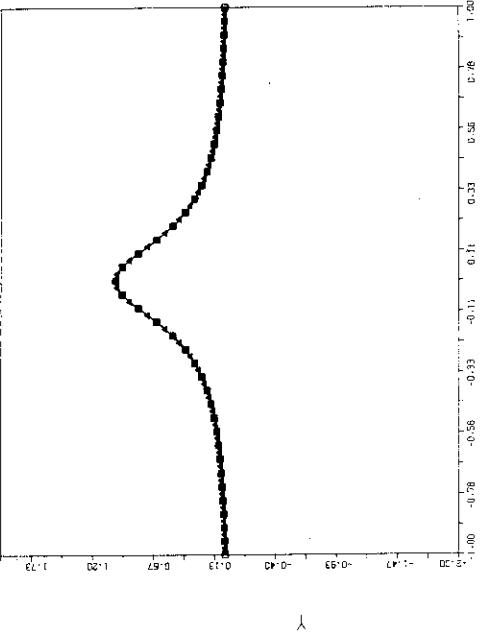


Fig. 5.7.4 41 point curve by the spline method

Table 5.4 Preciseness of the spline method as a function of data points for Runge's problem with convergence criterion (EPS) of 1×10^{-6}

Number of data points	Arguments and their interpolated results					Resultant condition code; IIL		
	-0.68	-0.18	0.18	0.68	-0.675	-0.175	0.175	0.675
Exact	7.9618×10^{-2}	5.5249×10^{-1}	5.5249×10^{-1}	7.9618×10^{-2}	8.0706×10^{-2}	5.6637×10^{-1}	5.6637×10^{-1}	8.0706×10^{-2}
5	-2.6742×10^{-2}	8.1009×10^{-1}	8.1009×10^{-1}	-2.6742×10^{-2}				0
9					8.4987×10^{-2}	6.1432×10^{-1}	6.1432×10^{-1}	8.4987×10^{-2}
21	7.9611×10^{-2}	5.5405×10^{-1}	5.5405×10^{-1}	7.9611×10^{-2}				0
41					8.0706×10^{-2}	5.6642×10^{-1}	5.6642×10^{-1}	8.0706×10^{-2}

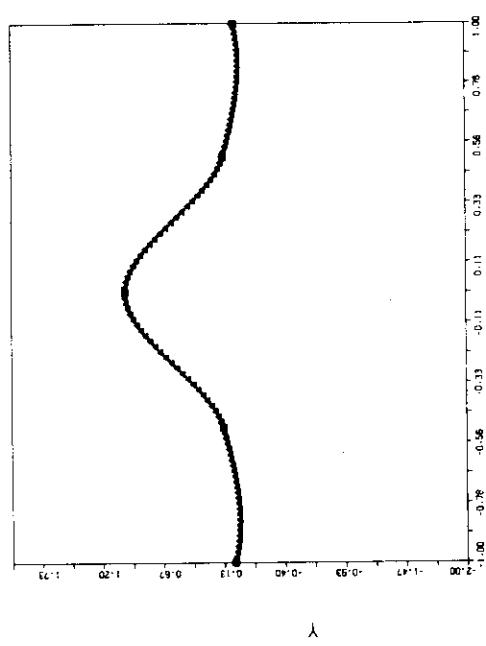


Fig. 5.8.1 5 point curve by Akima's method

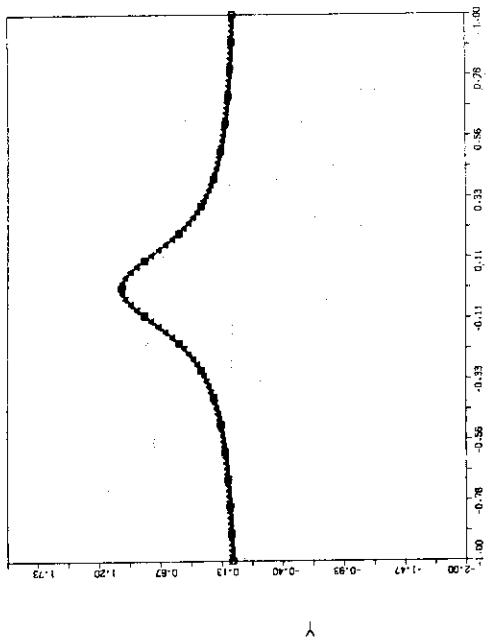


Fig. 5.8.3 21 point curve by Akima's method

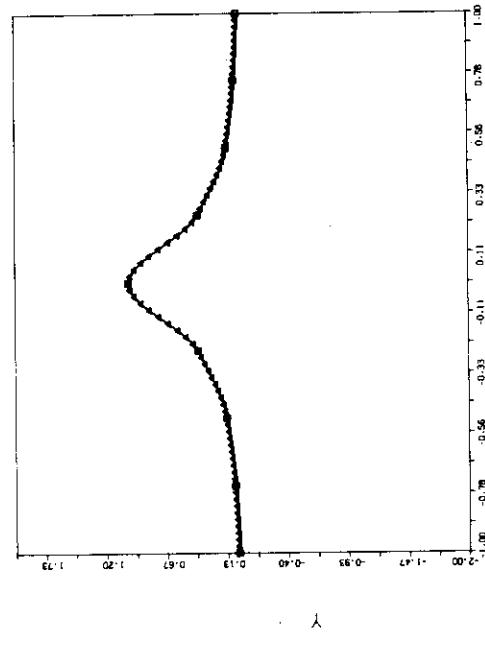


Fig. 5.8.2 9 point curve by Akima's method

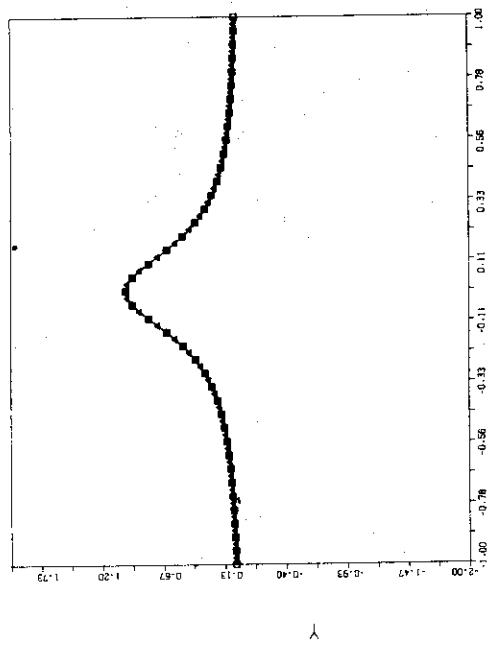


Fig. 5.8.4 41 point curve by Akima's method

Table 5.5 Preciseness of the Akima's method (subroutine INTRPL) as a function of data points for Runge's problem

Arguments and their interpolated results							
Number of data points	-0.68	-0.18	0.18	0.68	-0.675	-0.175	0.675
Exact	7.9618×10^{-2}	5.5249×10^{-1}	5.5249×10^{-1}	7.9618×10^{-2}	8.0706×10^{-2}	5.6637×10^{-1}	8.0706×10^{-2}
5	3.6010×10^{-2}	7.7292×10^{-1}	7.7292×10^{-1}	3.6010×10^{-2}			
9					8.1449×10^{-2}	5.6582×10^{-1}	8.1449×10^{-2}
21	7.9533×10^{-2}	5.5321×10^{-1}	5.5321×10^{-1}	7.9533×10^{-2}			
41					8.0713×10^{-2}	5.6686×10^{-1}	8.0713×10^{-2}

Table 5.6 Preciseness of the Akima's curve-fitting routine (CURVFT) with MD=2 as a function of data points for Runge's problem

Arguments and their interpolated results							
Number of data points	-0.69445	-0.17940	-0.67356	-0.17531	-0.67982	-0.17754	-0.67494
Exact	7.6590×10^{-2}	5.5414×10^{-1}	8.1024×10^{-2}	5.6550×10^{-1}	7.9657×10^{-2}	5.5928×10^{-1}	8.0719×10^{-2}
5	3.8165×10^{-2}	8.9121×10^{-1}					5.6918×10^{-1}
9			8.1737×10^{-2}	5.9621×10^{-1}			
21					7.9569×10^{-2}	5.5908×10^{-1}	
41						8.0725×10^{-2}	5.6964×10^{-1}

Table 5.7 Computing time of each subroutine for Runge's test
 (upper row=time in msec, lower row=data points + interpolating points)

Number of data points	LAGS	AKLAG	CONFRS	SPLINS	INTRPL	CURVFT	
						MD=1	MD=2
5	15	21	23	3	2	0	1
	101	101	101	101	101	81	81
9	29	32	41	3	2	0	1
	81	81	81	81	81	81	81
21	158	118	99	7	4	1	2
	101	101	101	101	101	101	101
41	370	177	69	11	5	2	4
	81	81	81	81	81	81	81
Characteristics			EPS=1. $\times 10^{-6}$	EPS=1. $\times 10^{-5}$	EPS=1. $\times 10^{-6}$	Interpolate within the subroutine because of curve-fitting program	
	Call subroutine for each interpolation		Interpolate within the subroutine				

に対して高精度を与え安定である。但し、必要以上余り多くのデータ点数をとると計算誤差を生ずる。注意としては、EPSを小さく（ 10^{-5} のオーダー前後）とらないと、おかしな値を算出してしまう（Fig. 5.6 のシリーズ参照）。

後半の方法は、いづれも3次区分多項式系を用いた方法であるが、内挿点のみを求める方法は、SPLINS（スプライン法）とINTRPL（Akima の方法）である。スプライン法は、Akima の方法より計算時間を喰うが、Fig. 5.7 のシリーズで示した如く、データ点数が極端に少ないと除いては、比較的データ点数の広い範囲に亘って良い内挿値を与えており、最近よくプログラム内に組まれたりしているのも頷ける。Akima の方法は、スプライン法よりも更に計算時間が短かく（3分の1から半分程度）、尚良いことには、スプライン法よりも広い範囲に亘って高精度で且つ丈夫である。

Akima の方法でCURVFTルーチンは、データ点間を何等分割して内挿するプログラムであるので、プロッター向きであるが、計算時間は内挿ルーチンの半分以下で（等分割しているのでプログラムが簡単），同じ精度を与える（Table 5.5 およびTable 5.6 を比較されたい）。一般に困難である多価関数系のデータ・フィッティングに強力な手段（このときはMD = 2として用いる。第4.1章参照）となる。このテスト例については後述する。

次に、データ点が非対称な分布をしている問題に対するテストを行う。問題としては、指標積分関数 $E_1(x)$ の $0.01 \leq x \leq 1.4$ の範囲が選ばれた。Fig. 5.9 は厳密解の曲線で、この問題での参考曲線として用いられる。Fig. 5.10～Fig. 5.14 は内挿法ルーチン（LAGS, AKLAG, CONFRS, SPLINS, INTRPL）による計算結果（データ点は9個）であるが、こゝでも、Lagrange 法は、一般には比較的内挿しやすい領域に入ってから振動発散している（データ点は□印、内挿点は△印などを思い出されたい）。他の方法は全領域に対して大体良い内挿を行っている。更に詳細に数値的検討を行ってみると、Table 5.8 に示す如く、連分数法が最も精度が高く、次にAkima の方法、スプライン法と続く。スプライン法は、Aitken-Lagrange 法よりは精度が高いが、Akima の方法よりは低くなっている。これらのことから、連分数法は計算時間は喰うが高精度であり（従って、特殊関数などの関数近似の際の基底関数としてよく用いられている），同系統に属するスプライン法とAkima の方法とでは、Akima の方法が勧められる。

以上で基本的なベンチマーク問題によるテストが終ったので、次に科学技術方面での実用的な応用例によるテストを行おう。上で述べたように、Akima の方法は、精度の点からも計算時間の点からも、バランスの良くとれた実用的な内挿法と云えよう。従って、こゝではできるだけ多くの問題によるテストを、Akima の方法にしぼって行いたいので、図形表示による検討にする。

Fig. 5.15 は液体金属カリウムのイオン・イオンポテンシャルのCURVFITルーチンによる曲線である。実に自然な曲線が描かれている。スプライン法の結果（Fig. 5.16）も略同様の結果を示すが、Akima の方法より若干不自然な処があることがわかる。現在この方面ではスプライン法の応用が最新と思われるが³³⁾、再考の余地があろう。

Fig. 5.17 は不飽和蒸気の臨界状態（圧力対密度）におけるCURVFITの結果である。Fig. 5.18 は、スプライン法の結果である。両者を比較してみると、 $T > T_c$ および $T = T_c$ の曲線では、曲

*³³⁾ 数値計算は Handbook of Mathematical Functions⁹⁾ の 5.1.53 の数値公式によって行われた。

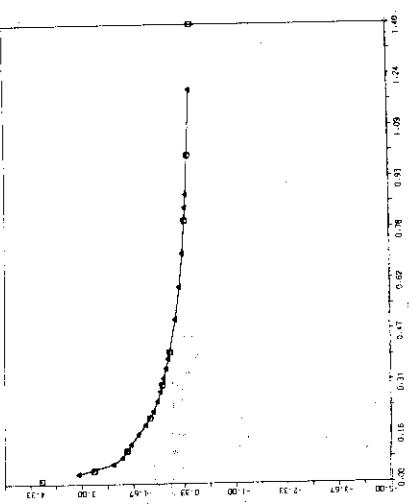


Fig. 5.9 Exact curve of the exponential integral function $E_1(x)$

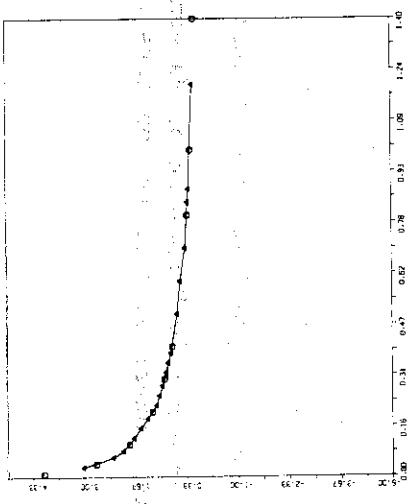


Fig. 5.11 Curve by Aitken-Lagrange method

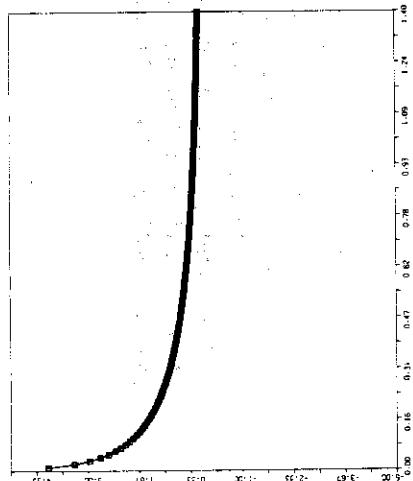


Fig. 5.12 Curve by the continued fraction method

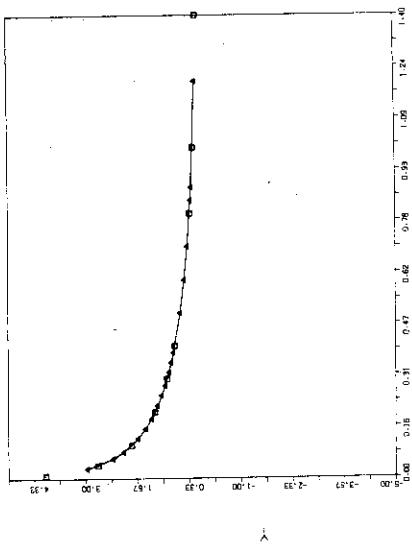


Fig. 5.13 Curve by the spline method

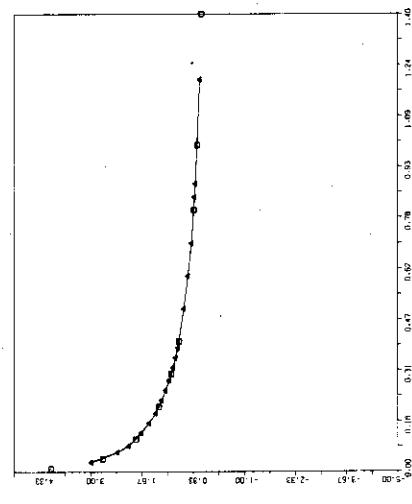


Fig. 5.14 Curve by Akima's method

Table 5.8 Preciseness of five interpolation methods for the exponential integral function $E_1(x)$

Method	Arguments and their interpolated results						Computing time (msec)	ILL EPS	
	0.03	0.08	0.18	0.25	0.35	0.60	0.88	1.20	
Exact (*1)	2.9591	2.0269	1.3098	1.0443	7.9423×10 ⁻¹	4.5445×10 ⁻¹	2.6963×10 ⁻¹	1.5840×10 ⁻¹	
LAGS	3.0169	1.9837	1.3410	9.9489×10 ⁻¹	8.7507×10 ⁻¹	-3.5553	6.1196	-1.3979×10 ²	6
AKLAG	3.0169	1.9837	1.3410	1.0394	8.0043×10 ⁻¹	4.6863×10 ⁻¹	2.6847×10 ⁻¹	1.6358×10 ⁻¹	7
CONFRS	2.9614	2.0267	1.3098	1.0443	7.9424×10 ⁻¹	4.5440×10 ⁻¹	2.6965×10 ⁻¹	1.5870×10 ⁻¹	10
SPLINS	3.0751	1.9354	1.3343	1.0278	7.9947×10 ⁻¹	4.4187×10 ⁻¹	2.7112×10 ⁻¹	1.5891×10 ⁻¹	1
INTRPL	2.9908	2.0123	1.3086	1.0495	7.9117×10 ⁻¹	4.6789×10 ⁻¹	2.6922×10 ⁻¹	1.6036×10 ⁻¹	1

ILL is resultant condition code number edited out of the program.

ILL=0 : normal end,

=29000: not reached to the required precision due to lack of data points

EPS is the convergence criterion.

(*1) Calculation by numerical formula from "Handbook of Mathematical Functions"9)

(*2) SPLINS doesn't converge with convergence criterion less than 1×10^{-4} .

線自身が素直なので差が見られないが、 $T < T_c$ のときは密度（x 軸方向）が 1.7 と 6.2 附近で急激に曲線の方向が変るので、Akima の方法が自然な曲線を表わしているのに反し、スプライン法は不自然な搖ぎを起している。

Fig. 5.19 は或る関数系の CURVFT の結果である。直線部分も曲線部分も自然な曲線を示している。一方、スプライン法 (Fig. 5.20) では、不連続点の $x = 0$ 附近で、Fourier 展開における Gibbs の現象^{*2)} と似た不自然さを示している。

以上で大体 Akima の方法の有用性が示されたので、以下では、Akima の方法 (CURVFT) のみを用いて、一般に難しいとされている問題に対するテスト結果を示そう。Fig. 5.21 は $y = \text{sign}(x)$ と呼ばれている階段関数であるが、実に綺麗に再現されている（多値関数なので、MD = 2 として用いることに注意されたい）。不連続点の $x = 0$ 附近では、異常振動を防ぐ為に半分づつ分割点（データ点）を細かくして行っている。Fig. 5.22 は飛行機のデザインに応用した例であるが、Akima の方法は、こゝでも実に自然に曲線を示している。

核断面積などのデータ処理でも内挿法が使われることが多い。この場合のように、関数系の変化がデカルト座標上での表示に適さない場合は、他の座標系に変換して内挿する方が適当である。例えば、²³⁵U の吸収断面積について考えてみよう。Fig. 5.23 は厳密解と考えてよい自然な曲線である。Fig. 5.24 は、対数変換して内挿した結果で、Fig. 5.25 は、対数変換せず元の座標系のまゝで内挿した結果である。両者を比較すると、座標変換した方が自然な曲線を与えており、座標変換の必要性が認識される。

最後に、データ点の与え方によって、内挿結果にどのような影響を与えるかについて調べてみよう。Fig. 5.26 は最適なデータ点を与えたときの少女のプロフィールであり、少女らしい感じが良く画かれている。一方、Fig. 5.27 は、少しデータ点数をサボったプロフィールであるが、少し硬い表情となっている。どっちが好きかって？ それは読者の嗜好の問題であろう。

5.2 $Z = F(X, Y)$ 型の内挿法プログラム

$Z = F(X, Y)$ 型の内挿法ルーチンとなると、 $Y = F(X)$ 型に比べるとデータ処理の上からも、数値表の比較よりも視覚的な図形を用いての比較の方がわかりやすい。又、 $Y = F(X)$ 型内挿のベンチマーク・テストの結果も貴重な参考情報となる。

プログラム化されているのも幾つか見受けられるが、1 次元内挿のテスト結果から考慮して、Akima の方法 (SFCFIT)^{22, 23)} とスプライン法 (SMTH；立体プロッター・プログラム THREE-D の平滑化（内挿でデータ点間の分割点を求める）ルーチン¹⁶⁾ を比較するのが妥当であろう。

Fig. 5.28 は、生のデータ点のみから構成された箱形の図形であるが、不連続的な曲線を有し、且つ多値関数的でもあるので、内挿法の問題としては難しい方に属する。Fig. 5.29 は Akima の方法による結果であるが元の図形に忠実であることが認められる。一方、スプライン法の Fig. 5.30 を観ると、不連続点附近でかなりの異常振動を起しており、このような問題に対しては実用上

*2) Fourier 展開で無限項までとれば正しい曲線を示すが、有限項まででは、不連続附近で異常振動をする現象。文献 6) の p. 69 参照。

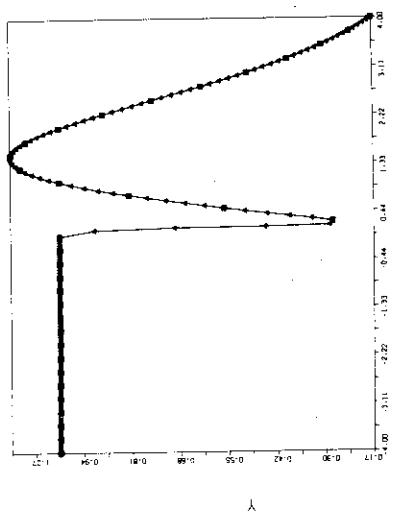


Fig. 5.15 Curve of the ion-ion potential of liquid potassium by Akima's method

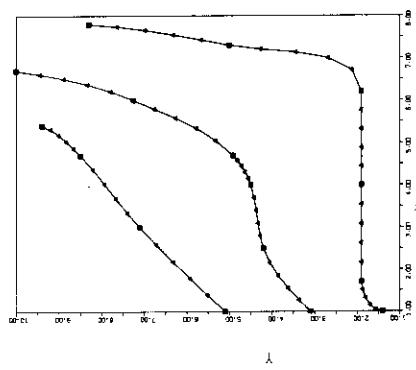


Fig. 5.16 Curve of the ion-ion potential of liquid potassium by the spline method

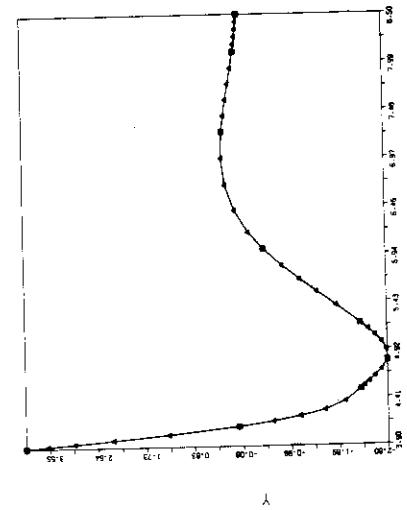


Fig. 5.17 Critical state curve (pressure-density as parameter of temperature) of unsaturated vapor by Akima's method

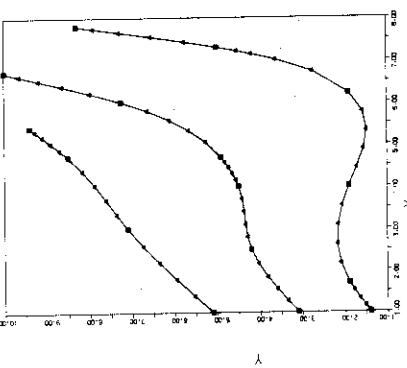


Fig. 5.18 Critical state curve (pressure-density as parameter of temperature) of unsaturated vapor by the spline method

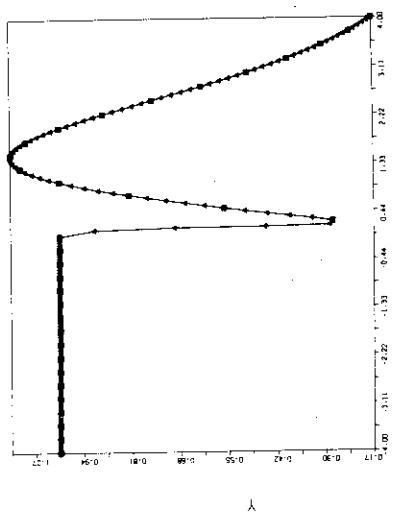


Fig. 5.19 Curve of a mathematical function by Akima's method

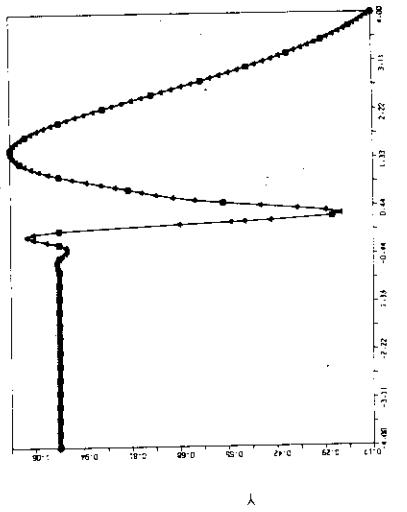


Fig. 5.20 Curve of a mathematical function by the spline method

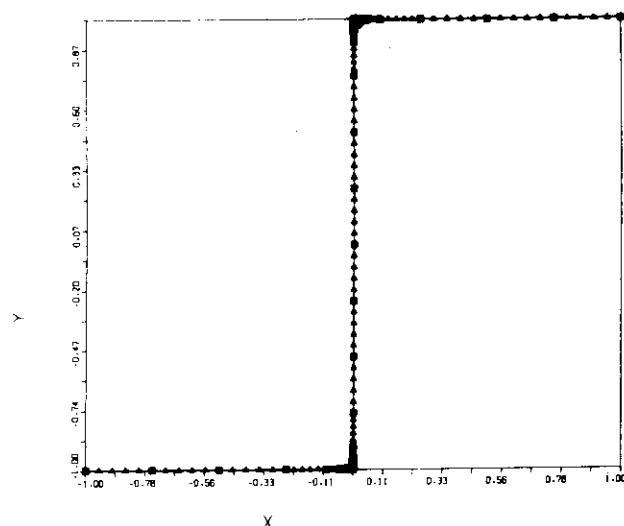


Fig. 5.21 Curve of $y=\text{sign}(x)$ by Akima's method

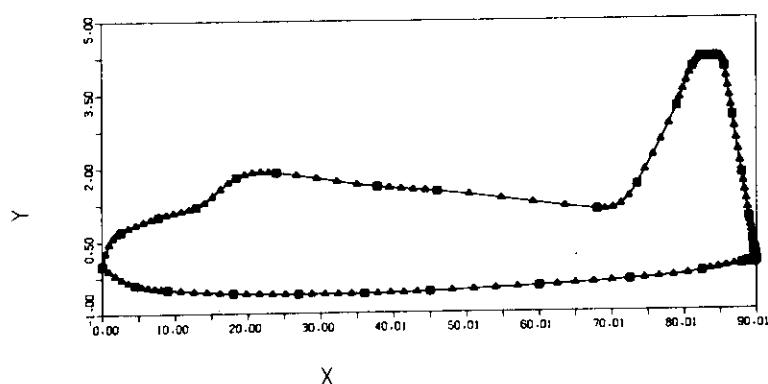
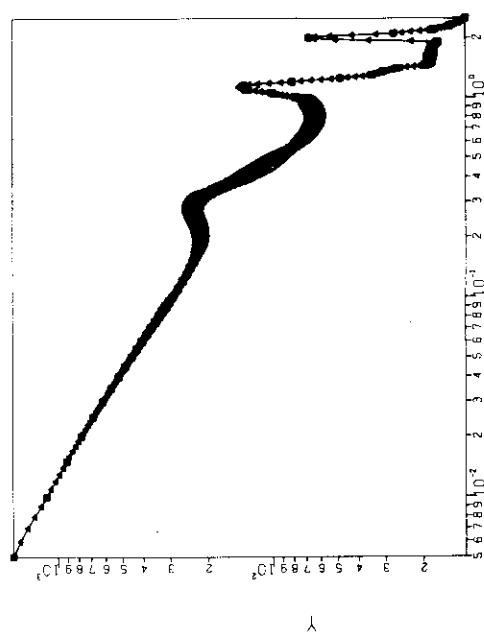
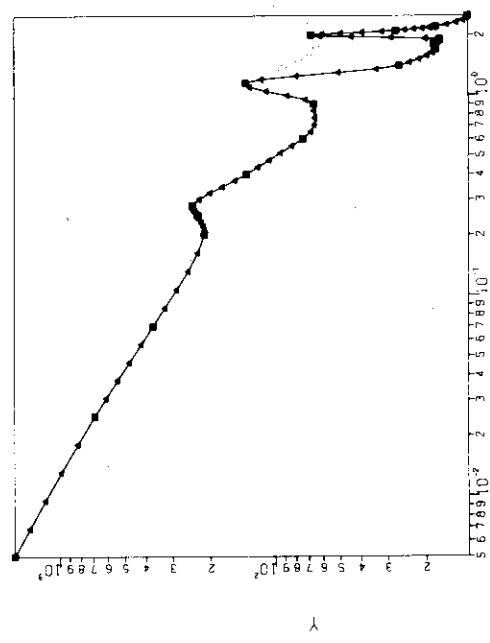
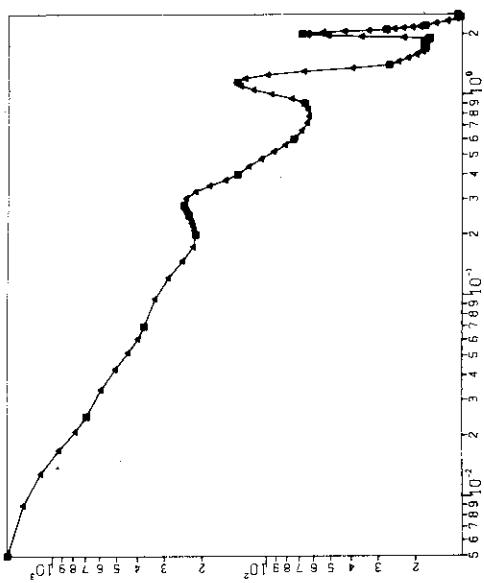


Fig. 5.22 Airplane's silhouette by Akima's method

Fig. 5.23 Exact curve of $\sigma_a(E)$ of ^{235}U Fig. 5.24 Curve of $\sigma_a(E)$ of ^{235}U interpolated logarithmically by Akima's methodFig. 5.25 Curve of $\sigma_a(E)$ of ^{235}U interpolated normally by Akima's method

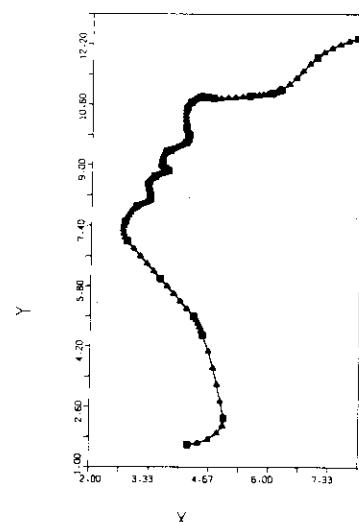


Fig. 5.26 Girl's profile with optimum 29 data points by Akima's method

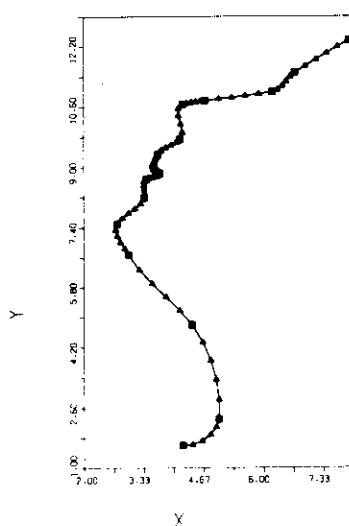


Fig. 5.27 Girl's profile with 16 data points by Akima's method

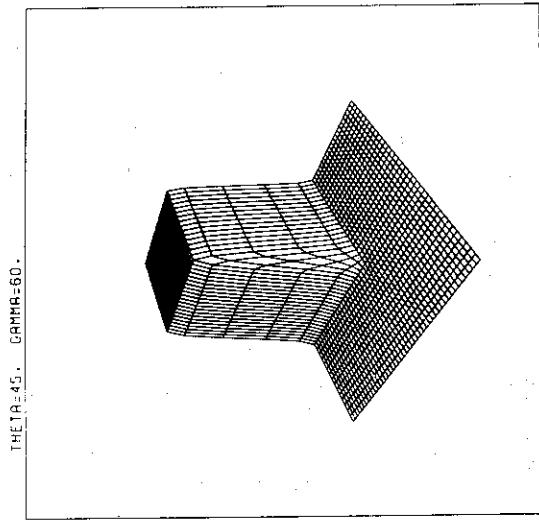


Fig. 5.29 Box-shaped surface fitted Akima's method

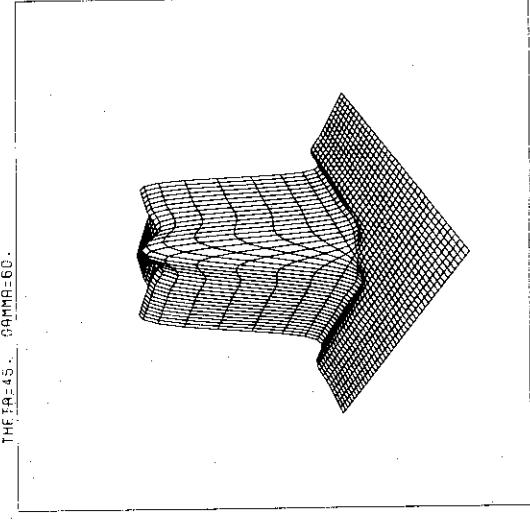


Fig. 5.30 Box-shaped surface fitted by the spline method

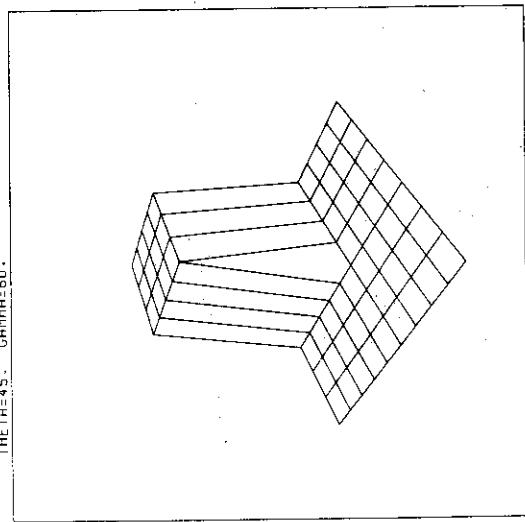


Fig. 5.28 Box-shaped surface (original)

使えないと考えてよいであろう。尚、Akima の图形で、箱の表面をより忠実に再現しようと考へるならば、不連續点（箱の稜）附近のデータ点を少し細かく与えればよい（前節のテスト結果、特に Fig.5.21 を参照）。

Fig. 5.31.1 および Fig. 5.31.2 は、鞍点 (saddle point) を持つ曲面 $y = \exp(-x^3 + 3xy^2)$ ($-1 \leq x \leq 1, -1 \leq y \leq 1, 0 \leq z \leq 2$) の図で、それぞれ正面図および側面図を示していて、以下の検討の参考図となるものである。Fig. 5.32.1 および Fig. 5.32.2 は、それぞれ Akima の方法による結果であるが、元の図の曲面を忠実に再現している。一方、Fig. 5.33.1 および Fig. 5.33.2 は、スプライン法の結果であるが、滑かに変化している等の曲線が凸凹していて、不自然な結果を示している。

上記の代表的な 2 つのテストから判断して、プロッター・ルーチン THREE-D の平滑化ルーチンは、特異点を含む問題への適用には注意が必要である。このような場合は、この平滑化ルーチンを Akima の方法に置き換えるか、或は、予め Akima の方法によって内挿点を求めておいて、プロッター・ルーチンを用いるのが賢明であろう。

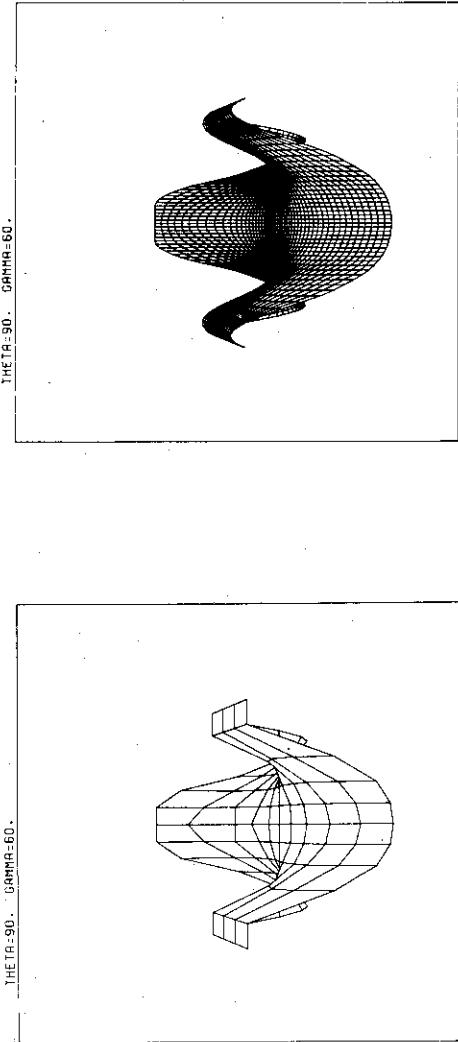


Fig. 5.31.1 Front view (original) of $Z = e^{-x^3+3xy^2}$ of $Z = e^{-x^3+3xy^2}$

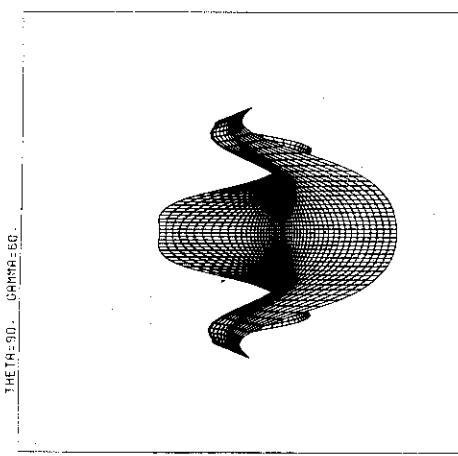


Fig. 5.33.1 Front view of $Z = e^{-x^3+3xy^2}$ surface fitted by the spline method

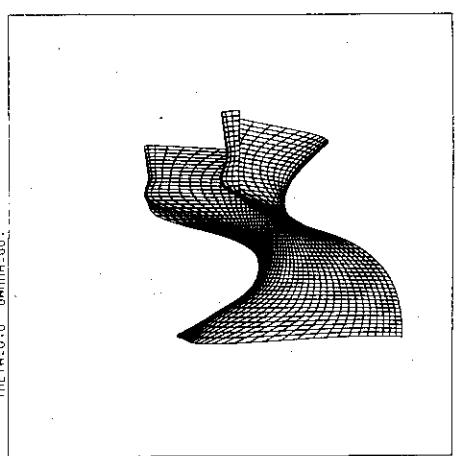


Fig. 5.33.2 Side view of $Z = e^{-x^3+3xy^2}$ surface fitted by the spline method

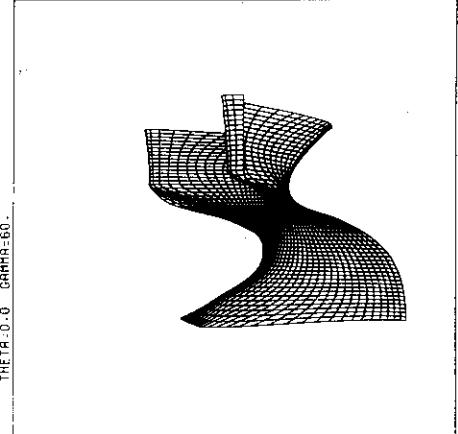


Fig. 5.32.2 Side view of $Z = e^{-x^3+3xy^2}$ surface fitted by the Akima's method

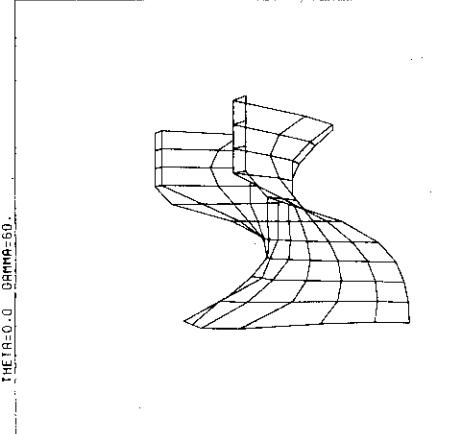


Fig. 5.31.2 Side view (original) of $Z = e^{-x^3+3xy^2}$

6. 結 論

内挿法の数値解法プログラムのベンチマーク・テストの結果をまとめると、以下の事が結論できよう。1次元内挿では、Lagrange 内挿法は如何なる用途でもお勧めできかねる。高精度と云う点では連分数法が優れているが計算時間の点で難がある。Akima の方法は全ての点で優れており、今後、現在よく用いられているスプライン法にとって換わるものであろう。更に、最小自乗法の平滑化における基底関数や数値積分の被積分関数の近似など、多方面での応用が期待される。2次元内挿における結論でも同じことが云えよう。

現在、内挿法が最もよく用いられているのはプロッター・ルーチンの平滑オプション（データ点間の分割点の算出）であるが（1次元では低次のLagrange 多項式法、2次元ではスプライン法が用いられているようである）、できるだけ早い時期に Akima の方法と交換すべきであろう。

謝 辞

Anderson 氏 (UCRL) は、我々に、彼のプログラムの標準FORTRANへの書き換えを勧めてくれ、合わせて、この為にプログラムの再テストを行い、プログラムの実証をしてくれました。又、藤村統一郎氏 (核設計研究室) には討論に参加していただきました。こゝに謝意を表わす次第です。

6. 結 論

内挿法の数値解法プログラムのベンチマーク・テストの結果をまとめると、以下の事が結論できよう。1次元内挿では、Lagrange 内挿法は如何なる用途でもお勧めできかねる。高精度と云う点では連分数法が優れているが計算時間の点で難がある。Akima の方法は全ての点で優れており、今後、現在よく用いられているスプライン法にとって換わるものであろう。更に、最小自乗法の平滑化における基底関数や数値積分の被積分関数の近似など、多方面での応用が期待される。2次元内挿における結論でも同じことが云えよう。

現在、内挿法が最もよく用いられているのはプロッター・ルーチンの平滑オプション（データ点間の分割点の算出）であるが（1次元では低次のLagrange 多項式法、2次元ではスプライン法が用いられているようである）、できるだけ早い時期に Akima の方法と交換すべきであろう。

謝 辞

Anderson 氏 (UCRL) は、我々に、彼のプログラムの標準FORTRANへの書き換えを勧めてくれ、合わせて、この為にプログラムの再テストを行い、プログラムの実証をしてくれました。又、藤村統一郎氏 (核設計研究室) には討論に参加していただきました。こゝに謝意を表わす次第です。

文 献

- 1) Wendorff B.: "Theoretical Numerical Analysis", Academic Press (1966);
- 2) 戸川隼人："数値計算", サイエンス (1977)
- 3) 伊勢武治, 藤村統一郎："最近の内插法のアルゴリズムと計算プログラム," 情報処理, 17, 417 (1976)
- 4) 富士通(株)："SSL(科学用サブルーチン・ライブラリー)解法解説書", 000-301~309-003-5 (1972)
- 5) idem.: "SSL II使用手引書", 99 SP-0050-1 (1976)
- 6) 森正武："曲線と曲面——計算機による作図と追跡", 教育出版 (1973)
- 7) 穂坂衛："コンピュータ・グラフィックス", 産業図書 (1974)
- 8) Olver F.W.J.: "Asymptotics and Special Functions", Academic Press (1974)
- 9) Abramowitz M. & Stegun I.A. (eds.): "Handbook of Mathematical Functions", Dover Pub. (1965)
- 10) Luke Y.L.: "Mathematical Functions and their Approximations", Academic Press (1975)
- 11) Hayes J.G. (ed.): "Numerical Approximation to Function and Data", Athlone Press (1970)
- 12) 伊勢武治, 西田雄彦, 鈴木忠和："最近の最小自乗法の計算コード", 日本原子力学会誌, 18, 89 (1976)
- 13) Stroud A.H.: "Numerical Quadrature and Solution of Ordinary Differential Equations - a textbook for a beginning course in numerical analysis", Springer-Verlag (1974)
- 14) Davis P.J. & Rabinowitz P.: "Method of Numerical Integration", Academic Press (1975)
- 15) Strang G. & Fix G.J.: "An Analysis of the Finite Element Method", Prentice-Hall (1973)
- 16) CALCOMP (株) : "THREE-D, a perspective drawing software system", California Computer Products, Inc. (1969)
- 17) Greville T.N.E. (ed.): "Theory and Applications of Spline Functions", Academic Press (1969)
- 18) Schultz M.H.: "Spline Analysis", Prentice-Hall (1973)
- 19) Schweikert D.G.: "An interpolation curve using a spline in tension", J. Math. Physics, 45, 312 (1966)
- 20) Akima H.: "New method of interpolation and smooth curve fitting based on local procedures", J. ACM, 17, 589 (1972)

- 21) idem.: "A433, Interpolation and smooth curve fitting based on local procedures", Comm. ACM, 15, 914 (1972)
- 22) idem.: "Method of bivariate interpolation and smooth surface fitting based on local procedures", Comm. ACM, 17, 18 (1974)
- 23) idem.: "A474, Bivariate interpolation and smooth surface fitting based on local procedures", Comm. ACM, 17, 26 (1974)
- 24) Cline A.K.: "Scalar - and planer-valued curve fitting using splines under tension", Comm. ACM, 17, 218 (1974)
- 25) idem.: "A476, Six subprograms for curve fitting using splines under tension", Comm. ACM, 17, 220 (1974)
- 26) Lagerlöf R.O.E.: "Interpolation with rounded ramp functions", Comm. ACM, 17, 476 (1974)
- 27) Havránek M.: "Spline interpolation in a N-dimensional data-base for nuclear reactor calculations", Atomkernenergie, 26, 190 (1975)
- 28) Günther C.: "IPOL - Ein FORTRAN-Program zur zwei-dimensionalen Interpolation", KFK 2175 (1976)
- 29) Küfner K.: "MODINT - Ein FORTRAN IV-Program zur zwei- und dreidimensionalen Interpolation in Gittern", KFK 2303 (1976)
- 30) Anderson D.V.: "The three-dimensional tensor-product, spline interpolation code (TPSIC)", UCRL-51958 (1976)
- 31) idem.: (private communication)
- 32) de Boor C.: "Package for calculating with B-splines", SIAM J. Numer. Anal. 14, 441 (1977)
- 33) Larsen R.D., Smith P.W. & Voges P.: "Spline function representation of the radial distribution function", Computers & Chemistry, 1, 23 (1974)
- 34) 藤村統一郎, 西田雄彦, 浅井清 (編) : " JSSL (原研版・科学用サブルーチン・ライブラリー) マニュアル" , JAERI-M 7102 (1977)

付 錄

付録 A.1 には、プログラム利用者の便宜を考えて、本文のベンチマーク・テストに用いられた全すべての内挿法プログラムの使用例をジョブ・コントロール・カード付きで載せておく（Akima のプログラムについては本文中に記してあるので省略されている）。プログラムの使用方法については、各々の使用法解説書を参照されたい。

Anderson の 3 次元内挿法プログラム（1 次元 B スプラインのテンソル積で表示）は、非標準のFORTRAN 言語で書かれてあったが、氏の勧めもあって、標準FORTRAN に書き直した。このプログラム・リストが付録 A.2 に示されている。現在プラズマ物理関係で用いられているそうで、良い結果を得ているとの事である。リストの公開については、既に了解をされている。

付録A.1 ベンチマーク・テストで用いられた内挿法プログラムのジョブ・
コントロール・カード付き入力例

```

....*...1...*...2...*...3...*...4...*...5...*...6...*...7...*...8...

*ND      A111.
*JOB     8881223,T,ISE+431,12,SHINER
*DELETE  J1223,LAGSE1A
*DELETE  J1223,LAGSE1B,NAME#2
*IMPORT
C   PROGRAM LAGS.
C   MAIN FOR LAGS IN SSL
C   Y=F(X)
C   DIMENSION A(18),X(100),Y(100),U(201),V(201),L(201)
C
C   REWIND 1
C   REWIND 2
C   CALL DTLIST
C   READ (5,1000)(A()),I=1,18
C   WRITE(6,1000)(A()),I=1,18
C   READ (5,1010) IU,LMAX,NMAX
C   WRITE(6,1015) IU,LMAX,NMAX
C   READ (5,1020)(X(L),L=1,LMAX)
C   WRITE(6,1017) (I,I=1,10)
C   WRITE(6,1023)
C   WRITE(6,1025)(X(L),L=1,LMAX)
C   DO 3  L=1,LMAX
C     AX=X(L)
C     Y(L)=E1(AX)
C
3  CONTINUE
C   WRITE(6,1033)
C   WRITE(6,1025) (Y(L),L=1,LMAX)
C   READ (5,1020) (U(N),N=1,NMAX)
C   WRITE(6,1017) (I,I=1,10)
C   WRITE(6,1043)
C   WRITE(6,1025) (V(N),N=1,NMAX)
C   IF(U,>0.0) IU=6
C
C   CALL CLOCKM(IT1)
C   DO 10 N=1,NMAX
C   PU(N)
C   CALL LAGS(X+Y+LMAX,P,0,ILL)
C   V(N)=0
C   IL(N)=ILL
C
10 CONTINUE
C   CALL CLOCKM(IT2)
C   WRITE(6,1052)
C   WRITE(6,1024)(IL(N),N=1,NMAX)
C
1  WRITE(6,1053)
C   WRITE(6,1025) (V(N),N=1,NMAX)
C   IT=IT2-IT1
C   WRITE(6,5) IT
C   WRITE(1) LMAX,(X(L)+L=1,LMAX),(Y(L),L=1,LMAX)
C   WRITE(2) NMAX,(U(N),N=1,NMAX),(V(N),N=1,NMAX)
5  FORMAT(5X,3H TIME DURING INTRPL-ROUTINE(MSEC.))=(6)
1000 FORMAT(1BA4)
1010 FORMAT(3I6)
1015 FORMAT(5X,4H IU=(3,3X,6H LMAX=13,3X,6H NMAX=13)
1017 FORMAT(10(5X,12,5X))
1020 FORMAT(6E12.5)
1023 FORMAT(/58X,2H X)
1024 FORMAT(10(3X,15,4X))
1025 FORMAT(10(1PE12.4))
1033 FORMAT(/58X,2H Y)
1043 FORMAT(/58X,2H U)
1052 FORMAT(/58X,4H ILL)
1053 FORMAT(/58X,2H V)
C
C   STOP
C   END
C   FUNCTION E1(X)
C   DIMENSION A(10)
C   DATA A/-5.7721566E-01,9.999193E-01,-2.4991055E-01,5.5519968E-02,
*=-9.76004E-03,1.07857E-03,2.334733.2,50621E-01,3.330657,1.681534/
C
C   HANDBOOK MATHEMATICAL FUNCTIONS 5.1,53-FORMULA
C
C   IF(X,LT,0.0) GO TO 999
C   IF(X,GT,1.0) GO TO 10
C   E1 = A(1)+X*(A(2)+X*(A(3)+X*(A(4)+X*(A(5)+A(6)*X))))-ALOG(X)
C   RETURN
C   GO TO 1
C
10 CONTINUE
C   E1 = ((X*(X+A(7))+A(8))/(X*(X+A(9))+A(10)))+(1.0/X)*EXP(-X)
C   RETURN
C   GO TO 1
999 WRITE(6,1000)
1000 FORMAT(26H ERROR = NEGATIVE ARGUMENT)
C
C   STOP
C   1 CONTINUE
C   END
C
*HLLIEDRUN
*FD  HLLIED1-SSL,FILE=(CATLG,SYST,SSLH)                                SSLH
*DJSKTN F01,J1223,LAGSE1A
*DJSKTN F02,J1223,LAGSE1B
*DATA
C   E1(X) TEST 10 POINTS FOR LAGS
      9    18
      0.01      0.04      0.10      0.20      0.30      0.40      X1
      0.00      1.00      1.40
      0.03      0.06      0.08      0.12      0.15      0.18      U1
      0.22      0.25      0.28      0.32      0.35      0.38      U2
      0.50      0.60      0.70      0.84      0.88      1.2       U3
*
*JEND

```

Fig. A.1 Input example for LAGS

```

.....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

*NO      A111,
*HPORT   /T.O/TIME 105
          C1/CORE 64
          W,O/PAGE 40
          P,O/PCH 0
          /00331

*JOB     2111223,T,ISE,431,12,SHINER
*C       PROGRAM AKLAG
*C       MAIN FOR AKLAG IN SSL2
*C       Y=F(X)
*C       DIMENSION A(18),X(100),Y(100),U(200),V(200),W(400),IL(200)
*C
*C       REWIND 1
*C       REWIND 2
*C       CALL DTLIST
*C       READ (5,1000)(A(I),I=1,18)
*C       WRITE(6,1000)(A(I),I=1,18)
*C       READ (5,1010) IU,LMAX,NMAX,EPS,KMAX
*C       WRITE(6,1015) IU,LMAX,NMAX
*C       READ (5,1020) (XL),L=1,LMAX
*C       WRITE(6,1017) (I,I=1,10)
*C       WRITE(6,1023)
*C       WRITE(6,1025) (X(L),L=1,LMAX)
*C       DO 3 L=1,LMAX
*C       AX=X(L)
*C       Y(L)=E1(AX)
*C       3  CONTINUE
*C       WRITE(6,1033)
*C       WRITE(6,1025) (Y(L),L=1,LMAX)
*C       READ (5,1020) (U(N),N=1,NMAX)
*C       WRITE(6,1017) (I,I=1,10)
*C       WRITE(6,1043)
*C       WRITE(6,1025) (U(N),N=1,NMAX)
*C       IF(U,UEQ,0) IU=5
*C
*C       CALL CLOCKM(IT1)
*C       DO 10 N=1,NMAX
*C       P=U(N)
*C       KMAX=9
*C       CALL AKLAG(X,Y,LMAX,P,KMAX,EPS,Q,VW,ILL)
*C       V(N)=Q
*C       IL(N)=ILL
*C       10 CONTINUE
*C       CALL CLOCKM(IT2)
*C       WRITE(6,1052)
*C       WRITE(6,1024)(IL(N),N=1,NMAX)
*C       1  WRITE(6,1053)
*C       WRITE(6,1025) (V(N),N=1,NMAX)
*C       IT=IT2-IT1
*C       WRITE(6, 5) IT
*C       WRITE(1) LMAX,(X(L),L=1,LMAX),(Y(L),L=1,LMAX)
*C       WRITE(2) NMAX,(U(N),N=1,NMAX),(V(N),N=1,NMAX)
*      5 FORMAT(5X,35H TIME DURING INTRL=ROUTINE(MSEC.),=16)
1000 FORMAT(18A4)
1010 FORMAT(316,E12.5,16)
*
1015 FORMAT(5X,4H (U=13,3X,6H LMAX=13,3X,6H NMAX=13)
1017 FORMAT(10(5X,12X))
1020 FORMAT(6E12.5)
1023 FORMAT(3BX,2H X)
1024 FORMAT(10(3X,154X))
1025 FORMAT(10(1PE12.4))
1033 FORMAT(/5BX,2H Y)
1043 FORMAT(/5BX,2H U)
1052 FORMAT(/5EX,4H ILL)
1053 FORMAT(/5BX,2H V)
STOP
END
FUNCTION E1(X)
DIMENSION A(10)
DATA A/-5.7721566E-01,9.9999193E-01,-2.4991055E-01,5.5519968E-02,
*-9.76004E-03,1.07857E-03,2.334733,2.50621E-01,3.330657,1.681934/
C
C       HANDBOOK MATHEMATICAL FUNCTIONS 5.1.53-FORMULA
C
IF(X,LT,0.0) GO TO 999
IF(X,GT,1.0) GO TO 10
E1 = A(1)+X*(A(2)+X*(A(3)+X*(A(4)+X*(A(5)+A(6)*X))))+ ALOG(X)
RETURN
10 CONTINUE
E1 = ((X*(X+A(7))+A(8))/(X*(X+A(9))+A(10)))*(1.0/X)*EXP(-X)
RETURN
999 WRITE(6,1000)
1000 FORMAT(26H ERROR = NEGATIVE ARGUMENT)
STOP
END
*HLLIEDRUN
#FD HLLIED1-SSL,FILE=(CATLG,SYST,SSL2)                         SSL2
#DISK F01
#DISK F02
*DATA
E1(X) TEST 10 POINTS FOR AKLAG
      9    18    1.0 -06    9
      0.01    0.04    0.10    0.20    0.30    0.40    X1
      0.80    1.0     1.4
      0.03    0.06    0.08    0.12    0.15    0.18    X2
      0.22    0.25    0.28    0.32    0.35    0.38    U1
      0.50    0.60    0.70    0.84    0.88    1.2    U2
      1.00    1.10    1.20    1.30    1.40    1.50    U3
*
*JEND

```

Fig. A.2 Input example for AKLAG

```

....*,...,1,...*,...,2,...*,...,3,...*,...,4,...*,...,5,...*,...,6,...*,...,7,...*,...,8

MNO      A111.
                                                 /
T.O/TIME 10S
C,1/CORE 64
W,O/PAGE 40
P,O/PCH 0
/00331

*DELETE    J1223,SPIONA
*DELETE    J1223,SPIONB,NAME=2
*GJOB     2111223,T,ISE,431.12,SHINER
*HFORTRAN
C   PROGRAM SPLINS
C   MAIN FOR SPLINS IN SSL
C   Y=F(X)
C   DIMENSION A(18),X(100),Y(100),U(200),V(200),V1(200),V2(200)
C
C   REWIND 1
C   REWIND 2
C   CALL DTLIST
C   HEAD (5,1000)(A(I),I=1,18)
C   WRITE(6,1000)(A(I),I=1,18)
C   READ (5,1010) IU,LMAX,NMAX,EPS
C   WRITE(6,1015) IU,LMAX,NMAX
C   READ (5,1020) (X(L),L=1,LMAX)
C   WRITE(6,1017) (I,I=1,10)
C   WRITE(6,1023)
C   WRITE(6,1025) (X(L),L=1,LMAX)
C   READ (5,1020) (Y(L),L=1,LMAX)
C   WRITE(6,1033)
C   WRITE(6,1025) (Y(L),L=1,LMAX)
C   READ (5,1020) (U(N),N=1,NMAX)
C   WRITE(6,1017) (I,I=1,10)
C   WRITE(6,1043)
C   WRITE(6,1025) (U(N),N=1,NMAX)
C   IF(U.EQ.0.) IU=6
C
C   CALL CLOCKM(IT1)
C   S=100.0
C   CALL SPLINS(LMAX,X,Y,NMAX,U,V,V1,V2,S,EPS,IU)
C   CALL CLOCKM(IT2)
2000 FORMAT(5H ILL=16)
1  WRITE(6,2000) ILL
WRITE(6,2010)
WRITE(6,1025) (V(I),I=1,NMAX)
WRITE(6,2030)
WRITE(6,1025) (V2(I),I=1,NMAX)
2010 FORMAT(16H 1-ST DERIVATIVE)
2030 FORMAT(16H 2ND DERIVATIVE)
WRITE(6,1053)
WRITE(6,1025) (V(N),N=1,NMAX)
IT = IT2-IT1
WRITE(6, 5) IT
WRITE(1) LMAX,(X(L),L=1,LMAX),(Y(L),L=1,LMAX)
WRITE(2) NMAX,(U(N),N=1,NMAX),(V(N),N=1,NMAX)
5  FORMAT(5X,3H TIME DURING INTRPL-ROUTINE(MSEC.)*16)
1000 FORMAT(18A4)
1010 FORMAT(316,E12.5)

1015 FORMAT(5X,4H IU=13,3X,6H LMAX=13,3X,6H NMAX=13)
1017 FORMAT(10(5X,12.5X))
1020 FORMAT(6E12.5)
1023 FORMAT(/58X,2H X)
1025 FORMAT(10(1E12.4))
1033 FORMAT(/58X,2H Y)
1043 FORMAT(/58X,2H U)
1053 FORMAT(/58X,2H V)
STOP
END
*HLEDRUN
*DISKTN F01,J1223,SPIONA
*DISKTN F02,J1223,SPIONB
*DATA
ION-ION POTENTIAL FOR LIQUID POTASSIUM---AKIMA INTERPOLATION WITH 9 KN,
   9   41   1,0 -06
      4.11      4.5      4.8      5.2      6.0      X1
      7.25      8.1      8.5
      4.0      0.0     -2.3     -2.8     -2.3     -0.5      Y1
      0.24      0.0     -0.08
      3.9001E+00  3.9196E+00  3.9433E+00  3.9782E+00  4.0315E+00  4.1100E+00
      4.1613E+00  4.2140E+00  4.2801E+00  4.3715E+00  4.5000E+00  4.5375E+00
      4.5791E+00  4.6320E+00  4.7033E+00  4.8000E+00  4.9090E+00  4.9958E+00
      5.0681E+00  5.1335E+00  5.2000E+00  5.3877E+00  5.5384E+00  5.6753E+00
      5.8215E+00  6.0000E+00  6.1806E+00  6.4123E+00  6.6793E+00  6.9840E+00
      7.2500E+00  7.4201E+00  7.5892E+00  7.7582E+00  7.9282E+00  8.1000E+00
      8.1797E+00  8.2592E+00  8.3388E+00  8.4190E+00  8.499 E+00
*JEND

```

Fig. A.3 Input example for SPLINS

```

      ,,,*,,,1,,,*,,,2,,,*,,,3,,,*,,,4,,,*,,,5,,,*,,,6,,,*,,,7,,,*,,,8

*END      A111:
T,0/TIME 10S
C,1/CORE 64
W,0/PAGE 40
P,0/PCH 0
/00391

*GJOB 8881223,T,[SE,431,12,SHINER
*HFORTRAN
C PROGRAM CONFRS
C MAIN FOR CONFRS IN SSL
C Y=F(X)
C DIMENSION A(18),X(100),Y(100),U(200),V(200),F(50),G(75),IL(200)
C
C REWIND 1
REWIND 2
CALL DTLIST
READ (5,1000)(A(I),I=1,18)
WRITE(6,1000)(A(I),I=1,18)
READ (5,1010) (U,LMAX,NMAX,EPS
WRITE(6,1015) (U,LMAX,NMAX,EPS
READ (5,1020) (X(L),L=1,LMAX)
WRITE(6,1017) (I,I=1,10)
WRITE(6,1023)
WRITE(6,1025) (X(L),L=1,LMAX)
DO 3 L=1,LMAX
AX=X(L)
Y(L)=E1(AX)
3 CONTINUE
WRITE(6,1033)
WRITE(6,1025) (Y(L),L=1,LMAX)
READ (5,1020) (U(N),N=1,NMAX)
WRITE(6,1017) (I,I=1,10)
WRITE(6,1043)
WRITE(6,1025) (U(N),N=1,NMAX)
IF(U,E0,O) IU=6
CALL CLOCKM(IT1)
DO 10 N=1,NMAX
P=U(N)
CALL CONFRS(X,Y,LMAX,P,EPS,O,LP,M0,F,G,ILL)
V(N)=0
IL(N)=ILL
10 CONTINUE
CALL CLOCKM(IT2)
WRITE(6,1052)
WRITE(6,1024)(IL(N),N=1,NMAX)
WRITE(6,1053)
WRITE(6,1025) (V(N),N=1,NMAX)
IT =IT2-IT1
WRITE(6, 5) IT
WRITE(1) LMAX,(X(L),L=1,LMAX),(Y(L),L=1,LMAX)
WRITE(2) NMAX,(U(N),N=1,NMAX),(V(N),N=1,NMAX)
5 FORMAT(5X,3H TIME DURING INTRPL-ROUTINE(MSEC.)=16)
1000 FORMAT(18A4)
1010 FORMAT(3I6,E12.5)
1015 FORMAT(5X,4H IU=I3,3X,6H LMAX=I3,3X,6H NMAX=I3,6H EPS=E12.5)
1017 FORMAT(105X,I2,5X))
1020 FORMAT(6E12.5)
1023 FORMAT(/58X,2H X)
1024 FORMAT(10(3X,I5.4X))
1025 FORMAT(10(IPE12,4))
1033 FORMAT(/58X,2H Y)
1043 FORMAT(/58X,2H U)
1052 FORMAT(/58X,4H ILL)
1053 FORMAT(/58X,2H V)
STOP
END
FUNCTION E1(X)
DIMENSION A(10)
DATA A/-5.7721566E-01,9,9999193E-01,-2,4991055E-01,5,5519968E-02,
*-9,76004E-03,1,07857E-03,2,334733,2,50621E-01,3,330657+1,681534/
C HANDBOOK MATHEMATICAL FUNCTIONS 5,1,53=FORMULA
C
IF(X,LT,0,O) GO TO 999
IF(X,GT,1,O) GO TO 10
E1 =A(1)+X*(A(2)+X*(A(3)+X*(A(4)+X*(A(5)+A(6)*X))))-ALOG(X)
RETURN
GO TO 1
10 CONTINUE
E1 =((X*(X+A(7))+A(8))/(X*(X+A(9))+A(10)))+(1,0/X)*EXP(-X)
RETURN
GO TO 1
999 WRITE(6,1000)
1000 FORMAT(26H ERROR * NEGATIVE ARGUMENT)
STOP
1 CONTINUE
END
#HL1|EDRUN
#DISK F01
#DISK F02
#DATA
E1(X) TEST 10 POINTS FOR CONFRS.
   9    18   1,6 -06
  0,01    0,04    0,10    0,20    0,30    0,40    X1
  0,00    1,0     1,4
  0,03    0,06    0,08    0,12    0,15    0,18    X2
  0,22    0,25    0,28    0,32    0,35    0,38    U1
  0,50    0,60    0,70    0,84    0,88    1,2    U2
                                         U3
*END

```

Fig. A.4 Input example for CONFRS

付録A.2 標準FORTRANに書き直されたAndersonの3次元内挿プログラムのリスト

```

      *,*,*,*,1,*,*,2,*,*,3,*,*,4,*,*,5,*,*,6,*,*,7,*,*,8

      COMMON/BSPC/AC(25,25,25),IP(25),RHS(25),A(25,25),B(25,25),
1          C(25,25),X(25),BLKS(4),F(12),Y(25),Z(25),IDER(25),
2          XBND(2),IBND(2),JDER(25),YBND(2),JBND(2),
3          FDAT(25,25,25),KDER(25),ZBND(2),KBND(2),RTX,RTY,RTZ,
4          XO,YO,ZO,TX(30),TY(30),TZ(30),DX,DY,DZ,KTX,KTY,KTZ,
5          TIX,TIY,TIZ
      COMMON/POLYCO/COEF(130),IOR,JOR,KOR
      COMMON/XYZ/IMIN,JMIN,KMIN,IMAX,JMAX,KMAX,KX,KY,KZ,IDIM,JDIM,KDIM,
1          NTX,NTY,NTZ,N2,KXM2,KYM2,KZM2,LDIM

C READ TEST FUNCTION PARAMETERS
      IMIN=1
      JMIN=1
      KMIN=1
      IMAX=21
      JMAX=21
      KMAX=21
      KX=4
      KY=4
      KZ=4
      IDIM=IMAX-IMIN+KX-1
      JDIM=JMAX-JMIN+KY-1
      KDIM=KMAX-KMIN+KZ-1
      NTX=IMAX-IMIN+2*KX-1
      NTY=JMAX-JMIN+2*KY-1
      NTZ=KMAX-KMIN+2*KZ-1
      N2=KDIM
      KXM2=KX-2
      KYM2=KY-2
      KZM2=KZ-2
      LDIM=12

C
      READ (5,12) IOR,JOR,KOR,VX,VY,VZ
      READ(5,13) NDX,NDY,NDZ,XB,YB,ZB,DX,DY,DZ,DELTAT
13 FORMAT(3I2,7F10.2)
      READ (5,11) (COEF(NDAT),NDAT=1,130)
11 FORMAT(10F8,1)
      WRITE(6,14)
      WRITE(6,12) IOR,JOR,KOR,VX,VY,VZ
      WRITE(6,15)
      WRITE(6,13) NDX,NDY,NDZ,XB,YB,ZB,DX,DY,DZ,DELTAT
      WRITE(6,10)
      WRITE(6,11) (COEF(NDAT),NDAT = 1,130)
      READ(5, 16) (XBND(I),IBND(I),I=1,KXM2)
      READ(5, 16) (YBND(J),JBND(J),J=1,KYM2)
      READ(5, 16) (ZBND(K),KBND(K),K=1,KZM2)
      WRITE(6, 17) (XBND(I),IBND(I),I=1,KXM2)
      WRITE(6, 18) (YBND(J),JBND(J),J=1,KYM2)
      WRITE(6, 19) (ZBND(K),KBND(K),K=1,KZM2)
10 FORMAT('1 POLYNOMIAL COEFFICIENTS FOLLOW',//)
15 FORMAT('NDX',//,'NDY',//,'NDZ' ,XB ,YB ,
         'ZB' ,DX ,DY ,DZ ,DELTAT',//)
14 FORMAT('!' ,IOR ,JOR ,KOR ,VX' ,
         'VY ,VZ',//)
12 FORMAT(3I10,3F10.2)
16 FORMAT(2(F10.2,I10))

```

```

.....*....1.....*....2.....*....3.....*....4.....*....5.....*....6.....*....7.....*....8

17 FORMAT(1X,'XBND,IBND = ',2(F10.2,I10))
18 FORMAT(1X,'YBND,JBND = ',2(F10.2,I10))
19 FORMAT(1X,'ZBND,KBND = ',2(F10.2,I10))
C SET UP THE GRID
C
DO 100 I = IMIN,IMAX
IDER(I) = 1
100 X(I) = (I-IMIN)*DX+XB
DO 101 I = IMAX+1,IMAX+KX-2
X(I) = XBND(I-IMAX)+XB
101 IDER(I) = IBND(I-IMAX)
DO 102 J = JMIN,JMAX
JDER(J) = 1
102 Y(J) = (J-JMIN)*DY+YB
DO 103 J = JMAX+1,JMAX+KY-2
Y(J) = YBND(J-JMAX)+YB
103 JDER(J) = JBND(J-JMAX)
DO 104 K = KMIN,KMAX
KDER(K) = 1
104 Z(K) = (K-KMIN)*DZ+ZB
DO 105 K = KMAX+1,KMAX+KZ-2
Z(K) = ZBND(K-KMAX)+ZB
105 KDER(K) = KBND(K-KMAX)
TIX = XB
T(Y = YB
TIZ = ZB
KTX = IMAX-IMIN
KTY = JMAX-JMIN
KTZ = KMAX-KMIN
C PRINT OUT X , Y , AND Z GRIDS
WRITE(6,20)
20 FORMAT('0 DATA OR OBSERVER GRID FOLLOWS!//,')
  , 'X  IDER  INDEX',//)
  , WRITE(6,21) (X(I),IDER(I),I,I=IMIN,IMAX+KX-2)
  , WRITE(6,22)
22 FORMAT('0 DATA OR OBSERVER GRID FOLLOWS!//,')
  , 'Y  JDER  INDEX',//)
  , WRITE(6,21) (Y(J),JDER(J),J,J=JMIN,JMAX+KY-2)
  , WRITE(6,23)
23 FORMAT('0 DATA OR OBSERVER GRID FOLLOWS!//,')
  , 'Z  KDER  INDEX',//)
  , WRITE(6,21) (Z(K),KDER(K),K,K=KMIN,KMAX+KZ-2)
21 FORMAT(F10.4,2I10)
C SET UP DATA FROM FUNCTION FUN FOR THIS TEST RUN
DO 1 I=1,1DIM
DO 1 J=1,JDIM
DO 1 K=1,KDIM
IDAT = IMIN-1+I
JDAT = JMIN-1+J
KDAT = KMIN-1+K
FDAT(IDAT,JDAT,KDAT)=FUN(X(IDAT),Y(JDAT),Z(KDAT),0,
,IDER(IDAT),JDER(JDAT),KDER(KDAT))
1  CONTINUE
C NORMALLY FDAT WOULD BE READ IN AS DATA

```

,,*,1,,,*,2,,,*,3,,,*,4,,,*,5,,,*,6,,,*,7,,,*,,,*

C THE FOLLOWING CALL WILL GENERATE ALL THE SPLINE
C COEFFICIENTS FOR THE 3D INTERPOLATION.

CALL TENPBS

C START AT XO = YO = ZO = ??? AND LET OBSERVER GO THRU THE GRID

XO=0,

YO=0,

ZO=0,

DELTAT=1,

LPRINT = 1

IF(LPRINT,EQ,1) WRITE(6,24)

24 FORMAT('1',//,' POLYNOMIAL = ',//)

25 FORMAT('1 COMPARISON OF ANALYTIC AND SPLINE RESULTS',//

' ,3X,1STEP FEXACT FSPLINE RFERROR,5X,'XO' YO' ZO',

' ,5X ,1 FDEX FDSPL RFDER',//)

DO 106 ISTEP = 1,500

FEXACT = FUN(XO, YO, ZO, LPRINT, 1, 1, 1)

FDEX = FUN(XO, YO, ZO, 0, NDX, NDY, NDZ)

IF(ISTEP,EQ,1) WRITE (6,25)

LPRINT = 0

FSPLIN = FUNT(XO, YO, ZO, 1, 1, 1)

FEERR = FEXACT - FSPLIN

RFERR = FEERR/(ABS(FEXACT)+1.E-20)

FDSPL = FUNT(XO, YO, ZO, NDX, NDY, NDZ)

FDERR = FDEX - FDSPL

RFDER = FDERR /(ABS(FDEX)+1.E-20)

WRITE(6, 26) ISTEP,FEXACT,FSPLIN ,RFERR ,XO,YO,ZO

,FDEX,FDSPL,RFDER

26 FORMAT(3X,14,1P3E12,3,0P3F8,2,1P3E12,3)

C PUSH OBSERVER UP TO NEXT OBSERVATION POINT

XO = XO + VX*DELTAT

YO = YO + VY*DELTAT

ZO = ZO + VZ*DELTAT

IF((XO,LE,X(IMAX)),AND,(XO,GE,X(IMIN))) GO TO 107

XO = XO - VX*DELTAT

VX = -VX

107 IF((YO,LE,Y(JMAX)),AND,(YO,GE,Y(JMIN))) GO TO 108

YO = YO - VY*DELTAT

VY = -VY

108 IF((ZO,LE,Z(KMAX)),AND,(ZO,GE,Z(KMIN))) GO TO 106

ZO = ZO - VZ*DELTAT

VZ = -VZ

106 CONTINUE

CALL EXIT

STOP

END

SUBROUTINE TENMAT

COMMON/BSPC/AC(25,25,25),IP(25),RHS(25),A(25,25),B(25,25),

C(25,25),X(25),BLKS(4),F(12),Y(25),Z(25),IDER(25),

XBND(2),IBND(2),JDER(25),YBND(2),JBND(2),

FDAT(25,25,25),KDER(25),ZBND(2),KBND(2),RTX,RTY,RTZ,

XO,YO,ZO,TX(30),TY(30),TZ(30),DX,DY,DZ,KTX,KTY,KTZ,

TIX,TIY,TIZ

COMMON/XYZ/IMIN,JMIN,KMIN,IMAX,JMAX,KMAX,KX,KY,KZ,LDIM,JDIM,KDIM,

NTX,NTY,NTZ,N2,KXM2,KYM2,KZM2,LDIM

,,...*,...,1,...*,...,2,...*,...,3,...*,...,4,...*,...,5,...*,...,6,...*,...,7,...*,...,8

```

L1=1
LXT=IDIM
LYT=JDIM
IF (IDIM,NE,1) CALL AMATZ(A,IMIN,KX,X,TX,IMAX,F,LDIM,IDER)
IF (JDIM,NE,1) CALL AMATZ(B,JMIN,KY,Y,TY,JMAX,F,LDIM,JDER)
IF (KDIM,NE,1) CALL AMATZ(C,KMIN,KZ,Z,TZ,KMAX,F,LDIM,KDER)
DO 1 I=1,IDIM
DO 1 J=1,JDIM
DO 1 K=1,KDIM
IDAT = IMIN+I
JDAT = JMIN+J
KDAT = KMIN+K
XDU = X(IDAT)
YDU = Y(JDAT)
ZDU = Z(KDAT)
KXD = KX
KYD = KY
KZD = KZ
IF (IDIM,NE,1) GO TO 100
XDU=XO
KXD=1
100 IF (JDIM,NE,1) GO TO 101
YDU=Y0
KYD=1
101 IF (KDIM,NE,1) GO TO 102
ZDU=ZO
KZD=1
102 CONTINUE
AC(I,J,K)=FDAT(IDAT,JDAT,KDAT)
1 CONTINUE
INDIAG = 1
GO TO 103
DO 104 K = 1,KDIM
WRITE(6,21) K,INDIAG
21 FORMAT('0 MATRIX AC AT ',I5,' TH LEVEL',I5,'TH FLAG POINT',//)
CALL PRNT(AC(1,1,K),LXT,LYT,L1,LXT,L1,LYT)
104 CONTINUE
103 IF (KDIM,EQ,1) GO TO 105
CALL DECOMP(KDIM,KDIM,C,IP,IER)
WRITE(6,21) IER
2 FORMAT(//,5H IER=I3,///)
IF (IER,NE,0) CALL EXIT
DO 6 I=1, IDIM
DO 6 J=1, JDIM
DO 4 K=1, KDIM
4 RHS(K)=AC(I,J,K)
CALL SOLVE(KDIM,KDIM,C,RHS,IP)
DO 5 K=1, KDIM
5 AC(I,J,K)=RHS(K)
6 CONTINUE
INDIAG = 2
GO TO 105
DO 106 K = 1, KDIM
WRITE(6,21) K,INDIAG
CALL PRNT(AC(1,1,K),LXT,LYT,L1,LXT,L1,LYT)

```

.....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

```

106 CONTINUE
105 IF(JDIM.EQ.1) GO TO 110
    CALL DECOMP(JDIM,JDIM,B,IP,IER)
    WRITE(6,2) IER
    IF(IER.NE.0) CALL EXIT
    DO 9 I=1,JDIM
    DO 9 K=1,KDIM
    DO 7 J=1,JDIM
7   RHS(J)=AC(I,J,K)
    CALL SOLVE(JDIM,JDIM,B,RHS,IP)
    DO 8 J=1,JDIM
8   AC(I,J,K)=RHS(J)
9   CONTINUE
    INDIAG = 3
    GO TO 110
    DO 107 K = 1,KDIM
    WRITE(6,21) K,INDIAG
    CALL PRNT(AC(1,1,K),LXT,LYT,L1,LXT,L1,LYT)
107 CONTINUE
110 IF(IDIM.EQ.1) GO TO 108
    CALL DECOMP(IDIM,1DIM,A,IP,IER)
    WRITE(6,2) IER
    IF(IER.NE.0) CALL EXIT
    DO 12 J=1,JDIM
    DO 12 K=1,KDIM
    DO 10 I=1,1DIM
10   RHS(I)=AC(I,J,K)
    CALL SOLVE(1DIM,1DIM,A,RHS,IP)
    DO 11 I=1,1DIM
11   AC(I,J,K)=RHS(I)
12   CONTINUE
    GO TO 108
    DO 109 K = 1,KDIM
    WRITE(6,22) K
22   FORMAT('0 MATRIX AC AT ',15,' TH LEVEL',//)
    CALL PRNT(AC(1,1,K),LXT,LYT,L1,LXT,L1,LYT)
109 CONTINUE
108 CONTINUE
    RETURN
    END
    SUBROUTINE SOLVE (N, NDIM, A, B, IP)           SOLVE E2,V1 07/18/74
C*NMS;MATHLIB;LLL;F4;SOURCE;SOLVE
C* THE FOLLOWING CARD IS FOR OPTIMIZED COMPIRATION UNDER CHAT,
C
C      DIMENSION A( 25,25), B(N), IP(N)
C-----
C
C      SOLUTION OF LINEAR SYSTEM, A*X = B .
C
C      ASSUMES DECOMP HAS BEEN CALLED FIRST TO TRIANGULARIZE THE MATRIX,
C      DO NOT USE IF DECOMP HAS SET IER ,NE, 0.
C
C      REFERENCE.,
C          C, B, MOLER, MATRIX COMPUTATIONS WITH FORTRAN AND PAGING.

```

.....*....1,...*....2....*....3,...*....4,...*....5,...*....6,...*....7,...*....8

C STAN-CS-71-196, COMPUTER SCIENCES DEPT., STANFORD UNIVERSITY,
C JANUARY 1971.

C-----
C INPUT..
C N = ORDER OF MATRIX,
C NDIM = DECLARED DIMENSION OF ARRAY A ,
C A = TRIANGULARIZED MATRIX OBTAINED FROM DECOMP,
C B = RIGHT HAND SIDE VECTOR,
C IP = PIVOT VECTOR OBTAINED FROM DECOMP.

C-----
C OUTPUT..
C B = SOLUTION VECTOR, X ,

C-----
C WRITTEN BY..... C, B, MOLER, UNIV, OF NEW MEXICO,
C MODIFIED BY..... A. C. HINDMARSH, LLL,
C NMG CONSULTANT.... A, B, HINDMARSH
C DATE LAST CHANGED.. JULY 18, 1974 (FNF)

C-----
C
IF (N .EQ. 1) GO TO 50
NM1 = N - 1
DO 20 K = 1,NM1
 KP1 = K + 1
 M = IP(K)
 T = B(M)
 B(M) = B(K)
 B(K) = T
 DO 10 I = KP1,N
10 B(I) = B(I) + A(I,K)*T
20 CONTINUE
DO 40 KB = 1,NM1
 KM1 = N - KB
 K = KM1 + 1
 B(K) = B(K)/A(K,K)
 T = -B(K)
 DO 30 I = 1,KM1
30 B(I) = B(I) + A(I,K)*T
40 CONTINUE
50 B(1) = B(1)/A(1,1)
RETURN

C *** END OF SOLVE ***
END
SUBROUTINE DECOMP (N, NDIM, A, IP, IER)
C*NMS;MATHLIB:LLL:F4;SOURCE:DECOMP

DECOMP E2,V1 07/18/74

C DIMENSION AC (25,25), IP(N)

C-----
C MATRIX TRIANGULARIZATION BY GAUSSIAN ELIMINATION,

.....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

```

C USE SOLVE TO OBTAIN SOLUTION OF LINEAR SYSTEM.
C
C REFERENCE.,
C. B. MOLER, MATRIX COMPUTATIONS WITH FORTRAN AND PAGING,
C STAN-CS-71-196, COMPUTER SCIENCES DEPT., STANFORD UNIVERSITY,
C JANUARY 1971.
C
C-----.
C INPUT.,
C N = ORDER OF MATRIX,
C NDIM = DECLARED DIMENSION OF ARRAY A ,
C A = MATRIX TO BE TRIANGULARIZED.
C
C OUTPUT.,
C A(I,J), I,LE,J = UPPER TRIZNGULAR FACTOR, U ,
C A(I,J), I,GT,J = MULTIPLIERS = LOWER TRIANGULAR FACTOR, I = L,
C IP(K), K,LT,N = INDEX OF K-TH PIVOT ROW,
C IP(N) = (-1)**(NUMBER OF INTERCHANGES) OR 0 ,
C IER = 0 IF A NONSINGULAR, OR K IF A FOUND TO BE
C SINGULAR AT STAGE K,
C
C DETERM(A) = IP(N)*A(1,1)*A(2,2)*,,,*A(N,N),
C
C IF IP(N)=0, A IS SINGULAR. SOLVE WILL DIVIDE BY ZERO,
C INTERCHANGES FINISHED IN U + ONLY PARTLY IN L .
C
C-----.
C
C WRITTEN BY..... C. B. MOLER, UNIV. OF NEW MEXICO.
C MODIFIED BY..... A. C. HINDMARSH, LLL.
C NMG CONSULTANT..... A. C. HINDMARSH
C DATE LAST CHANGED.. JULY 18, 1974 (FNF)
C
C-----.
C
C IER = 0
C IP(N) = 1
C IF (N .EQ. 1) GO TO 70
C NM1 = N - 1
C DO 60 K = 1,NM1
C   KP1 = K + 1
C   M = K
C   DO 10 I = KP1,N
C     IF (ABS(A(I,K)) .GT. ABS(A(M,K))) M = I
C     IP(K) = M
C     T = A(M,K)
C     IF (M .EQ. K) GO TO 20
C     IP(N) = -IP(N)
C     A(M,K) = A(K,K)
C     A(K,K) = T
C 20   IF (T .EQ. 0,) GO TO 80
C     DO 30 I = KP1,N
C       A(I,K) = -A(I,K)/T
C 30

```

.....*,...,1,...,*...,2,...,*...,3,...,*...,4,...,*...,5,...,*...,6,...,*...,7,...,*...,8

```

DO DO J = KP1,N
    T = A(M,J)
    A(M,J) = A(K,J)
    ACK,J) = T
    IF (T .EQ. 0.) GO TO 50
    DO 40 I = KP1,N
        AC(I,J) = AC(I,J) + AC(I,K)*T
40      CONTINUE
50      CONTINUE
70      K = N
    IF (AC(N,N) .EQ. 0.) GO TO 80
    RETURN
80      IER = K
    IP(N) = 0
    RETURN
C           *** END OF DECOMP ***
END
SUBROUTINE AMATZ(A,NMIN,K,X,T,NMAX,F,LDM,IDER)
DIMENSION A(25,25),T(1),F(12),X(25),IDER(25)
DO 111 I=1,25
DO 111 J=1,25
111 AC(I,J)=0.
KNX=NMAX-NMIN+2*K-1
C DIAGNOSTIC PRINT BELOW
    WRITE(6,16) KNX,(T(I),I,I=1,KNX)
16 FORMAT('0 KNX,T(I),I ARE = ',15,/,(' ',F10,4,15))
15 FORMAT(' X AND I AND ILEFT = ',F10,4,2I10)
DO 2 I=NMIN,NMAX+K-2
ISTR = I - NMIN+1
    CALL INTERV(T,KNX,X(I),ILEFT,MFLAG)
C DIAGNOSTIC PRINT FOLLOWS
    WRITE(6,15) X(I),I,ILEFT
15 IF(MFLAG,E0,0) GO TO 10
    WRITE(6,11)
11 FORMAT('//,15H TROUBLE MFLAG1,///')
    CALL EXIT
10 CALL BSPLVD(T,K,X(I),ILEFT,F,IDER(I))
    J1 = ILEFT-K+1
    J2 = MINOC(ILEFT,NMAX-NMIN+K-1)
    DO 1 J=J1,J2
        J3=J-ILEFT+K+(IDER(I)-1)*K
        AC(ISTR,J)=F(J3)
1     CONTINUE
2     CONTINUE
    L1=1
    LT=NMAX-NMIN+K+1
    CALL PRNT(A,LT,LT,L1,LT,L1,LT)
    RETURN
END
SUBROUTINE TENPBS
COMMON/BSPC/AC(25,25,25),IP(25),RHS(25),A(25,25),B(25,25),
1             C(25,25),X(25),BLKS(4),F(12),Y(25),Z(25),IDER(25),
2             XBND(2),IBND(2),JDER(25),YBND(2),JBND(2),
3             FDAT(25,25,25),KDER(25),ZBND(2),KBND(2),RTX,RTY,RTZ,
4             XO,YO,ZO,TX(30),TY(30),TZ(30),DX,DY,DZ,KTX,KTY,KTZ

```

.....*,...,1.,.,*,...,2.,.,*,...,3.,.,*,...,4.,.,*,...,5.,.,*,...,6.,.,*,...,7.,.,*,...,8

```

5      TIX,TIY,TIZ
COMMON/XYZ/IMIN,JMIN,KMIN,IMAX,JMAX,KMAX,KX,KY,KZ,LDIM,JDIM,KDIM,
1      NTX,NTY,NTZ,N2,KXM2,KYM2,KZM2,LDIM
CALL GENS(KX,TX,RTX,KTX,TIX,DX)
CALL GENS(KY,TY,RTY,KTY,TIY,DY)
CALL GENS(KZ,TZ,RTZ,KTZ,TIZ,DZ)
CALL TENMAT
RETURN
END
SUBROUTINE GENS(K,T,RDT,KT,T1,DT)
DIMENSION T(1)
WRITE(6,2)K,KT,T1,DT
2 FORMAT(//,.3H K=I6.4H KT=I6.4H TI=E12.5,4H DT=E12.5,///)
RDT=1./DT
T2=T1+KT*DT
NB=KT+K-1
C THIS CODE USES GUARD DNOTS ON THE T GRID
C RATHER THAN K-1 MULTIPLE KNOTS AT BOUNDARIES,
DO 3 I=1,K
  T(I)=T1-(K-I)*DT
3 T(I+NB)=T2+(I-1)*DT
DO 4 I=1,KT
  4 T(I+K)=T1+I*DT
RETURN
END
SUBROUTINE BSPLVD ( T, K, X, ILEFT, VNIKX, NDERIV )
CALCULATES VALUE AND DERIV,S OF ALL B-SPLINES WHICH DO NOT VANISH AT X
DIMENSION T(1),VNIKX(K,NDERIV)
DIMENSION A(20,20)
C
C FILL VNIKX(J,IDERIV), J=IDERIV, ., ,K WITH NONZERO VALUES OF
C B-SPLINES OF ORDER K+1-IDERIV + IDERIV=NDERIV, ., ,1, BY REPEATED
C CALLS TO BSPLVN
  CALL BSPLVN(T,K+1-NDERIV+1,X,ILEFT,VNIKX(NDERIV,NDERIV))
  IF (NDERIV .LE. 1)          GO TO 99
  IDERIV = NDERIV
  DO 15 I=2,NDERIV
    IDERVM = IDERIV-1
    DO 11 J=IDERIV,K
      VNIKX(J-1,IDERVM) = VNIKX(J,IDERIV)
11   IDERIV = IDERVM
    CALL BSPLVN(T,0.2*X,ILEFT,VNIKX(IDERIV,IDERIV))
15   CONTINUE
C
  DO 20 I=1,K
    DO 19 J=1,K
19    A(I,J) = 0,
20    A(I,I) = 1,
    KMD = K
    DO 40 M=2,NDERIV
      KMD = KMD-1
      FKMD = FLOAT(KMD)
      I = ILEFT
      J = K
21    JM1 = J-1

```

.....*....1.....*....2.....*....3.....*....4.....*....5.....*....6.....*....7.....*....8

```

IPKMD = I + KMD
DIFF = T(IPKMD) - T(I)
IF (JM1 .EQ. 0) GO TO 26
IF (DIFF .EQ. 0.) GO TO 25
DO 24 L=1,J
    A(L,J) = (A(L,J) - A(L,J-1))/DIFF*FKMD
24
25    J = JM1
    I = I - 1
        GO TO 21
26    IF (DIFF .EQ. 0.) GO TO 30
        A(1,1) = A(1,1)/DIFF*FKMD
C
30    DO 40 I=1,K
        V = 0,
        JLLOW = MAX0(I,M)
        DO 35 J=JLOW,K
            V = A(I,J)*VNIXX(J,M) + V
35
40    VNIXX(I,M) = V
99    RETURN
END
SUBROUTINE INTERV ( XT, LXT, X, ILEFT, MFLAG )
COMPUTES LARGEST ILEFT IN (1,LXT) SUCH THAT XT(ILEFT) ,LE, X.
DIMENSION XT(LXT)
DATA ILO /1/
IHI = ILO + 1
IF (IHI ,LT, LXT) GO TO 20
    IF (X ,GE, XT(LXT)) GO TO 110
    IF (LXT ,LE, 1) GO TO 90
    ILO = LXT - 1
        GO TO 21
20 IF (X ,GE, XT(IHI)) GO TO 40
21 IF (X ,GE, XT(ILO)) GO TO 100
C**** NOW X ,LT, XT(IHI) , FIND LOWER BOUND
30 ISTEP = 1
31 IHI = ILO
    ILO = IHI - ISTEP
    IF (ILO ,LE, 1) GO TO 35
    IF (X ,GE, XT(ILO)) GO TO 50
    ISTEP = ISTEP*2
        GO TO 31
35 ILO = 1
    IF (X ,LT, XT(1)) GO TO 90
        GO TO 50
C**** NOW X ,GE, XT(IL3) , FIND UPPER BOUND
40 ISTEP = 1
41 ILO = IHI
    IHI = ILO + ISTEP
    IF (IHI ,GE, LXT) GO TO 45
    IF (X ,LT, XT(IHI)) GO TO 50
    ISTEP = ISTEP*2
        GO TO 41
45 IF (X ,GE, XT(LXT)) GO TO 110
    IHI=LXT-1
C**** NOW XT(IL0) ,LE, X ,LT, XT(IHI) , NARROW THE INTERVAL
50 MIDDLE = (ILO + IHI)/2

```

```

,.,*,...,1.,.,*,.,2.,.,*,.,3.,.,*,.,4.,.,*,.,5.,.,*,.,6.,.,*,.,7.,.,*,.,8

      IF (MIDDLE .EQ. ILO)          GO TO 100
C     NOTE, IT IS ASSUMED THAT MIDDLE = ILO IN CASE IHI = ILO+1
      IF (X .LT. XT(MIDDLE))       GO TO 53
      ILO = MIDDLE                GO TO 50
100   IHI = MIDDLE               GO TO 50
C**** SET OUTPUT AND RETURN
90    MFLAG = -1
      ILEFT = 1                   RETURN
100   MFLAG = 0
      ILEFT = ILO                RETURN
110   MFLAG = 1
      ILEFT = LXT                RETURN
END
FUNCTION FUN(X,Y,Z,LPRINT,NDX,NDY,NDZ)
COMMON/POLYCO/COEF(130),IOR,JOR,KOR
F = 0.0
N = 1
XX=X
YY=Y
ZZ=Z
IF(X,EQ.0.) XX=1.E-30
IF(Y,EQ.0.) YY=1.E-30
IF(Z,EQ.0.) ZZ=1.E-30
DO 100 I = 0,IOR
DO 100 J = 0,JOR
DO 100 K = 0,KOR
IF(I+J+K,GT,5) GO TO 100
C FIND COEFFICIENTS ACCORDING TO DERIVATIVES INDEX
CX=1.0
CY=1.0
CZ=1.0
IF(NDX,EQ.1) GO TO 12
DO 10 L = 1,NDX-1
10 CX = CX*(I-L+1)
12 CONTINUE
IF(NDY,EQ.1) GO TO 21
DO 20 L = 1,NDY-1
20 CY = CY*(J-L+1)
21 CONTINUE
IF(NDZ,EQ.1) GO TO 31
DO 30 L = 1,NDZ-1
30 CZ = CZ*(K-L+1)
31 CONTINUE
ISTR = I-NDX+1
JSTR = J-NDY+1
KSTR = K-NDZ+1
IF(ISTR,LT,0) GO TO 200
IF(JSTR,LT,0) GO TO 200
IF(KSTR,LT,0) GO TO 200
F = F + CX*CY*CZ*COEF(N)*(XX**ISTR)*(YY**JSTR)*(ZZ**KSTR)

```

,.,*,.,.1.,.,*,.,2.,.,*,.,3.,.,*,.,4.,.,*,.,5.,.,*,.,6.,.,*,.,7.,.,*,.,8.

```

200 IF(LPRINT,E0,1) WRITE(6,11) N,CDEF(N),I,J,K
    N = N + 1
11 FORMAT(' +(CDEF(''13,'') = ''E10.5,'') * X**',
    ,I2,'* Y**',I2,'* Z**',I2)
100 CONTINUE
    FUN = F
    RETURN
    END
FUNCTION FUNT(XVAL,YVAL,ZVAL,NDX,NDY,NDZ)
COMMON/BSPC/AC(25,25,25),IP(25),RHS(25),A(25,25),B(25,25),
1           C(25,25),X(25),BLKS(4),F(12),Y(25),Z(25),IDER(25),
2           XBND(2),IBND(2),JDER(25),YBND(2),JBND(2),
3           FDAT(25,25,25),KDER(25),ZBND(2),KBND(2),RTX,RTY,RTZ,
4           XO,YO,ZO,TX(30),TY(30),TZ(30),DX,DY,DZ,KTX,KTY,KTZ,
5           TIX,TIY,TIZ
DIMENSION FX(12),FY(12),FZ(12)
COMMON/XYZ/IMIN,JMIN,KMIN,IMAX,JMAX,KMAX,KX,KY,KZ,IDIM,JDIM,KDIM,
1           NTX,NTY,NTZ,N2,KXM2,KYM2,KZM2,LDIM
I1=(XVAL-TX(KX))*RTX+1,
J1=(YVAL-TY(KY))*RTY+1,
K1=(ZVAL-TZ(KZ))*RTZ+1,
IUP=I1+KX-1
JUP=J1+KY-1
KUP=K1+KZ-1
CALL BSPLVD(TX,KX,XVAL,IUP,FX,NDX)
CALL BSPLVD(TY,KY,YVAL,JUP,FY,NDY)
CALL BSPLVD(TZ,KZ,ZVAL,KUP,FZ,NDZ)
IF(IDIM.NE.1) GO TO 100
I1=1
IUP=1
FX(1+(NDX-1)*KX) = 1.0
100 IF(JDIM.NE.1) GO TO 101
J1=1
JUP=1
FY(1+(NDY-1)*KY) = 1.0
101 IF(KDIM.NE.1) GO TO 102
K1=1
KUP=1
FZ(1+(NDZ-1)*KZ) = 1.0
102 SUM=0,
I3=0
DO 3 I=I1,IUP
I3 = I3 + 1
J3 = 0
DO 2 J=J1,JUP
J3 = J3 + 1
K3=0
DO 1 K=K1,KUP
K3=K3+1
I4 = I3 +(NDX-1)*KX
J4 = J3 +(NDY-1)*KY
K4 = K3 +(NDZ-1)*KZ
SUM=SUM+ACC(I,J,K)*FX(I4)*FY(J4)*FZ(K4)
1 CONTINUE
2 CONTINUE

```

,,,,*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

```

3 CONTINUE
  FUNT=SUM
  RETURN
END

SUBROUTINE BSPLVN ( T, JHIGH, INDEX, X, ILEFT, VNIKX )
CALCULATES THE VALUE OF ALL POSSIBLY NONZERO B- SPLINES AT *X* OF
C ORDER MAX(JHIGH,(J+I)(INDEX-1)) ON *T*.
  DIMENSION T(1),VNIKX(1)
  DIMENSION DELTAM(20),DELTAP(20)
  DATA J/1/, (DELTAM(I),I=1,20),(DELTAP(I),I=1,20)/40*0,/
          GO TO (10,20),INDEX

10 J = 1
  VNIKX(1) = 1,
  IF (J .GE. JHIGH)           GO TO 99
C
20  IPJ = ILEFT+J
  DELTAP(J) = T(IPJ) - X
  IMJP1 = ILEFT-J+1
  DELTAM(J) = X - T(IMJP1)
  VMPREV = 0,
  JP1 = J+1
  DO 26 L=1,J
    JPIML = JP1-L
    VM = VNIKX(L)/(DELTAP(L) + DELTAM(JPIML))
    VNIKX(L) = VM*DELTAP(L) + VMPREV
26  VMPREV = VM*DELTAM(JPIML)
  VNIKX(JP1) = VMPREV
  J = JP1
  IF (J .LT. JHIGH)           GO TO 20
C
99
      RETURN
END

SUBROUTINE PRNT(TFU,IT,JT,LD1,LD2,LD3,LD4)
DIMENSION TFU(25,25)
DOUBLE PRECISION HOL
DATA HOL/8H I =   /
MA=1
MB=10
23062 MC=MIN0(MB+1T)
  WRITE(6,23064)
23064 FORMAT(1H )
  WRITE(6,23063) (HOL,L,L=MA,MC)
23063 FORMAT(8X,10(1X,A6,13,2X))
  WRITE(6,23064)
  DO 23065 M=1, JT
23065 WRITE(6,23066) M , (TFU(L,M ),L=MA,MC)
23066 FORMAT(16,1P10E12,3)
  MA=MA+10
  MB=MB+10
  IF(MA .LE. IT) GO TO 23062
  RETURN
END
*HLIEDRUN
*DATA
      5      5      5      0,10      0,20      0,05

```

.....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....*

2	1	1	0,	0,	0,	1,0	1,0	1,0	1,0
0,	0,	0,	0,	0,	0,	0,1	0,1	0,1	0,1
0.1	0,	0,	0,	0,	0,1	0,1	0,1	0,	0,
0.1	0,	0,	0,	0,	0,	0,	0,	0,	0,
0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
0.1	0.1	0.1	0,	0,	0.1	0,	0,	0,	0,
0,	0,	0,	0,	0.1	0,	0.1	0.1	0.1	0.1
0,	0,	0,	0,	0.1	0,	0,	0,	0,	0,
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.	2	20,			3				
0.	3	20,			3				
0.	3	20,			3				