

数値計算ライブラリの自動選択システム

和田 武

愛媛大学総合情報処理センター

松山市文京町 3

TEL 089-927-8801, wada@dpc.ehime-u.ac.jp

概要

数値計算ライブラリを使用する際、浮動小数計算による誤差や解法アルゴリズムの特性のため、場合によっては全く信頼できない結果を生じる可能性がある。そこで、複数の使用可能なアルゴリズムの中から、問題に適合し、安定した結果を生じるアルゴリズムを自動的に選択するしくみが必要になる。本論では、与えた問題の性質を自動的に判定し、適切なアルゴリズムを選択し、かつその計算実行までも行い得るような利用者支援システムの開発を行う。具体的には解きにくい問題の例として硬い常微分方程式の数値解法を考え、硬さの度合い（ステイフ率）を数式処理システムを利用して計算し、データベースに格納されたアルゴリズムの中から適切なものを選択・実行するようにした。ユーザインターフェースを簡便にするため、Netscape Navigator などの Web ブラウザの利用を可能とした。また、実際のシステム動作を広く使用されている数値計算アルゴリズム SSL-II を対象として示した。

キーワード

利用者支援、アルゴリズム選択、数値計算ライブラリ

1. はじめに

幅広い分野における計算の要求と問題の多様化が進む中で、「問題の性質を把握しなければ安定した結果が得れない」ような計算が出現している。その多くは数値計算に出現する。数値計算では常にある種の近似解を求めるため、浮動小数計算による誤差が重要になったり、計算精度との関連で信じられないような結果を生んだりすることは一般に知られている。これらは広く悪条件問題と呼ばれている。本論では「硬い」(ステイフ:stiff) 常微分方程式を例にとって、その振舞について実験的に述べる。硬い常微分方程式の場合、一般的な数値解法を用いると、与えられた方程式の性質によって、非常に小さな刻み幅で数値解を求めない限り、正確な解の近似値を求められない場合がある。当然、大量の計算時間を消費し、かつ計算回数が増大とともに結果に含まれる数値誤差も無視できなくなる。しかし、このような場合にも硬い微分方程式に適する数値解法を用いると、上の困難を回避しつつ満足な結果を得ることができる。このように問題に適合した最適の解法(アルゴリズム)を自動的に選択するシステムをアルゴリズム選択システムという。計算機センターにこのようなシステムが構築され、学内外から自由にネットワークを通じて利用することが可能になれば、多くの利用者の便をはかり、かつ新しい学内センターの方向を示すこともできると思われる。

Table 1: 硬い常微分方程式の数値解と真の解の比較

| t | x (RKG) | x (ODGE) | x (真の解) |
|----------|-----------------|----------------|-----------------|
| 0.00E+00 | 0.00000000E+00 | 0.00000000E+00 | 0.000000000000 |
| 0.10E-05 | -0.55133384E+04 | 0.99999511E+00 | 0.9999949979513 |
| 0.30E-04 | -0.16767962E+12 | 0.99998522E+00 | 0.9999850001124 |
| 0.50E-04 | -0.50987906E+19 | 0.99997509E+00 | 0.9999750003124 |
| 0.80E-04 | -0.85496337E+30 | 0.99996012E+00 | 0.9999600007999 |
| 0.90E-04 | overflow | 0.99995500E+00 | 0.9999550010124 |
| 0.10E-03 | overflow | 0.99994993E+00 | 0.9999500012499 |

2. アルゴリズム選択と硬い常微分方程式

アルゴリズム選択の例として、次の一階連立常微分方程式の数値解を得る場合を考える。

$$\begin{cases} x' = -2\mu x + \mu y, & x(0) = 0 \\ y' = x - y, & y(0) = 2 \end{cases} \quad (1)$$

$$\mu = 10^6$$

この式は近似的に

$$\begin{cases} x(t) \simeq e^{-(1/2)t} - e^{-2\mu t} \\ y(t) \simeq 2e^{-(1/2)t} \end{cases} \quad (2)$$

となる。また、(2) をオイラー法で解くと、刻み幅 h に

$$h < \frac{4}{2\mu + 1} \simeq 2 \times 10^{-6}$$

という制約がつくことがわかる。 $t=1$ までの数値解を得ようとする、 10^6 ステップもの計算が必要になる。この状況はオイラー法より少し複雑なルンゲ-クッタ法やアダムス法(Adams-Bashforth 法など)を適用しても同様である。これは、行列 A の固有値の大きさが極端に異なることに起因している。このような微分方程式を硬い微分方程式と呼び、電気回路網、化学反応、原子炉の問題、制御系、偏微分方程式に対する「線の方法」などによくあらわれる。

(1) の連立常微分方程式を、適当な刻み幅 h によって、ルンゲ-クッタ-ギル法(RKG)で解いた場合と、硬い連立常微分方程式に適したギア法(ODGE)で解いた場合の結果を Table 1 に示す。overflow は数値結果が浮動小数の表現範囲を越えたために演算が不可能になったことを意味する。なお、計算には富士通が開発した数値計算ライブラリ、SSL-II に含まれるアルゴリズムを用いた。このような刻み幅では通常用いるルンゲ-クッタ-ギル法(RKG)による結果は全く信用をおくことができない。もちろん、刻み幅を上述のように極端に小さくすると、それなりの結果を得るが、計算回数が飛躍的に増大し、その間の誤差の累積も増大するので、数値結果の安定性と正確性の両面から、この方法の利用は不適切であることがわかる。もちろん、上のような悪条件性を持たない方程式に対しては、敢えて計算量の増すギア法等を採用しなくとも信頼のおける結果を得ることが出来る。このように硬い常微分方程式を解く際にはアルゴリズム選択が極めて重要になる。しかし、常微分方程式がスティッフであるか否かは一般に計算前にはわからないので、誤って不安定な結果を生じるアルゴリズムによる結果を盲信する危険がある。スティッフ率の大きさを事前に求めれば、正しいアルゴリズムを選択できることは明らかである。

3. アルゴリズム選択システム

図1 にアルゴリズム選択システムの構成を示す。点線の左側はユーザのマシン(クライアント)、右側はサーバマシン(ホスト)のシステムである。ユーザは Netscape Navigator 等の Web ブラウザを起動しサーバ

マシン上の Illustra データベースにアクセスしアルゴリズム検索を行う。システムの動作環境は Windows95 上の Web サーバ、WindowsNT4.0 上のデータベースシステム Illustra を中核とし、UNIX 環境下の数式処理システム Maple と結合する。

次に、アルゴリズム選択システムの動作例を示す。本システムを起動し、パッケージ選択後、問題の種類を与え、目的とするアルゴリズムの種類まで順次検索を行なっていく。図2は、SSL-II の常微分方程式入力画面である。システムは、入力された問題に対するステップ率を計算し、ステップな常微分方程式に適したアルゴリズムを呼び出し、実行可能な Fortran プログラムを自動的に作成する。図3がプログラムを実行した画面である。

4. むすび

本論では、アルゴリズム選択システムの開発について述べた。現在開発できている部分の概要は次のようにまとめることができる。

1. 数式として入力された常微分方程式の特徴抽出に数式処理システム Maple を活用し、アルゴリズム選択を行う。
2. ユーザインターフェースの向上を目指し、Netscape Navigator などの Web ブラウザでのシステムの利用を可能とした。このことにより、ネットワークを通じて、ユーザの場所、計算機の種類にとらわれない利用が可能となった。
3. 膨大なアルゴリズムの情報や検索システムを格納するため、オブジェクトリレーショナルデータベースシステム Illustra を活用した。
4. アルゴリズム名の選択のみでなく、実際の数値計算を行い結果を得ることを可能とした。

今後は、このようなシステムの一層の充実を図るため、以下の事項を早急実現する必要がある。

1. 最近の関連する論文[1,2]に見られるようなアルゴリズム選択システムの知識ベース化により、引数の数や型を意識せずに数値計算ライブラリを使用できるシステムの構成。
2. 数値計算ライブラリだけでなく、同様のシステムを統計計算ライブラリの使用に拡張し、多くのパッケージ間の引数や命令の差異を意識せずに計算可能な知的システムの作成。

参考文献

- [1] B.J.Dupee and J.H.Davenport, An Intelligent Interface to Numerical Routines, In Design and Implementation of Symbolic Computation Systems Eds. J.Calmet and C.Limongelli, Lecture Notes in Computer Science 1128, pp.252-262, Springer, 1996.
- [2] S.Weerawarana, E.N.Houstis, J.H.Rice, A.Joshi and C.E.Houstis, PYTHIA: A Knowledge-Based System to Select Scientific Algorithms, ACM Tran. Math. Soft., Vol.22, pp.447-468, 1996.

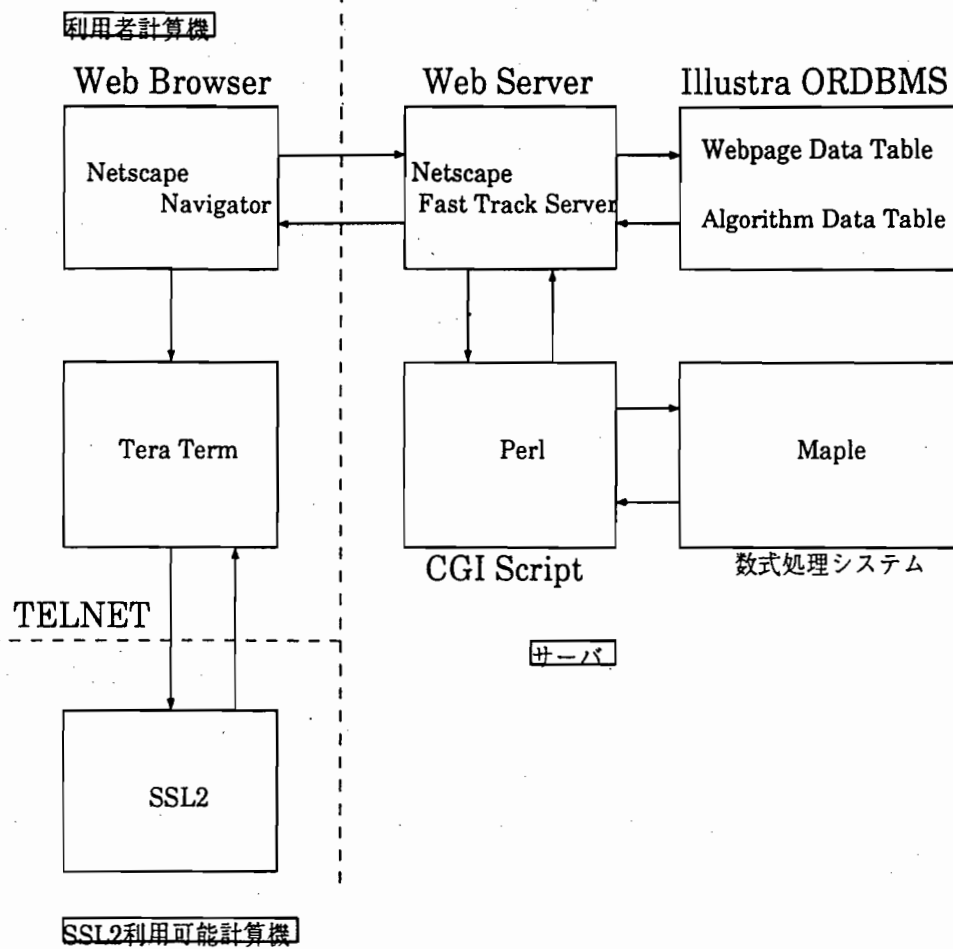


Fig.1 システム構成

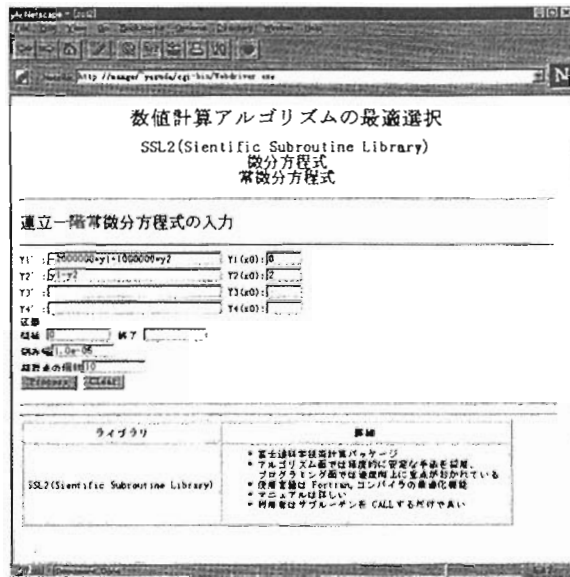


Fig.2 常微分方程式の入力



Fig.3 SSL-II の実行