数理解析研究所講究録
第832巻 1993年 127-136

疎行列解法概観

慶應義塾大学理工学部 野寺 隆 (Takashi Nodera)

1. はじめに

近年の計算機を使う環境の変化には,著しいものがある.従来,科学技術計算の多くは,メインフレームと呼ばれる大型計算機を使って行なうのが主流であった.しかし,現在ではunixをOS(オペレーティングシステム)とする低価格で高性能のワークステーションやパーソナルコンピュータが現れてきたので,科学技術計算を主流とする研究者の多くが,これらの計算機を主として利用するようになってきた.現在では,年間数百万円の費用をメインフレームの計算時間に使っていた大学の多くの研究室が,そのお金を利用してワークステーションの導入にふみきっている.

当然,OSとしてunixを使うようになると,計算を行なうためのプログラムも科学技術計算ではなくてはならないFortranを使うのではなく,C言語やC++などのオブジェクト指向型の言語で記述する傾向にある.例えば,従来,科学技術計算のサブルーチンとして定評を得ているIMSLはFortran言語で記述されているものを利用するのが主流であったが,近年ANSI準拠のC言語で記述されたものが販売されるようになった.そこで,序々にではあるがC言語のサブルーチンを使うような移行が行なわれている.また,Fortranも現在使われているFortran77からベクトル計算や並列計算を効率的に行なう命令を備えたFortran90がそろそろリリースされようとしている.Fortran90の基本的な命令体系は従来のFortran77と同じであるが,よりC言語に近い言語の様式を備えており,Fortran77とは全く別ものと考えるほうがよい.Fortran自体,これからはFortran77の体系のものと,Fortran90の体系の二つに分割されようとしている.

現在の科学技術計算の分野では,計算した結果をコンピュータ・グラフィックスのソフトウェアを使って視覚化することが重要な事柄の一つである.この点においても,ワークステーションやパーソナルコンピュータは小回りがきくのでおおいに利用できる.

いうなれば科学技術計算の分野においても,なかなか融通をきかせることが難しいメインフレームから小回りのきくワークステーションへとダウンサイジング(down sizing)が行なわれているのである.また,時代の流れというか,メインフレイムのOSもJCLコマンド形式のものからunixのコマンド形式のものへの移行が行なわれている.

大型の連立一次方程式の解を求める計算も,従来はメインフレームの独断場であったが,問題の性質にもよるのだが,今日ではワークステーションでも十分満足のゆく結果が得られることが実証されてきている.

本稿では,まず最初にパブリックドメインで提供されている疎行列解法のソフトウェアにどんなものがあるかについて述べる.特に,非対称な疎行列を係数とする連立一次

方程式の直接解法のソフトウェアについて述べる. 次に, 非対称な疎行列を係数とする連立一次方程式の解を求める反復解法として, 最近注目を集めている CGStab 法があまりよい収束特性を示さない例について報告する.

2. 疎行列解法

先日, 友人の一人から「行列計算の中で疎な非対称行列を係数とする連立一次方程式の解を求める直接解法のソフトウェアがどこかで入手できないか.」との問い合わせがあった.

行列計算のソフトウェアといえば, 何と行っても LINPACK や最近話題の LAPACK (主に, LINPACK と EISPACK の主なルーチンを持ち, スーパーコンピュータ用にも利用できるようにチューニングされている) をあげることができる. しかし, 疎行列計算が行なえるソフトウェアと考えてみると, これがなかなか思いあたらない. 当然, 前記の二つのソフトウェアには, 疎行列計算の機能はない. LINPACK の制作者の一人である J. Dongarra 教授 (テネシー大学とオークリッジ国立研究所) に尋ねたところ, 「疎行列解法のルーチンを LINPACK に入れようと考えたが, 疎行列解法のルーチンは解くべき問題に強く依存するので, あえて LINPACK のコードとしては取り入れなかった.」とのことであった.

偏微分方程式から得られる離散化行列を係数とする連立一次方程式の解を求めるものは直接解法, 反復解法を問わずいくつか存在するのだが, みんな有料 (それもかなり高額) のソフトウェアで, パブリックドメインのソフトウェア, 特に, 疎な非対称行列を係数とする連立一次方程式の解を求める直接解法のコードを捜すとなると, これがなかなか大変な作業である. 特に, 「お金をだして買いたくない.」という条件がつくと, 一段と難しい.

こんな時に, 役に立つのが Dongarra 教授と Grosse 氏 (AT&T のベル研究所) の 2 人によって始められた Netlib におうかがいを立ててみることである. そこで最初は, Netlib のインデックス (index) を取り寄せることにして,

```
netlib@research.att.com
```

というアドレス (IP address: 192.20.225.2) に

```
send index
```

という要求の入ったエレクトリックメールを送った. 当然, 折り返しに, Netlib が提供しているインデックスが送られてきた (特に, ネットワークが IP 接続されていれば, 数十秒で返事が返ってくる).

送られてきたインデックスの先頭には, Netlib の使い型が記述されているので, それを参考に Netlib に登録されているソフトウェアを入手すればよい. Netlib が始められた当時は, 十数個のソフトウェアが提供されていたののを覚えているが, 現在では数十個のソフトウェアが登録されている. 個々のソフトウェアの評価は, まちまちで非常に

クオリティの高いものもあるが，そうでないものもある．これは，ソフトウェアの制作者に評価がまかされており，だれでも提供を申しでればNetlibに登録できるからである(以前は，おおまかなソフトウェアの評価もインデックスに記述されていたが，現在は省略されている)．なお，Netlibのコードを使って得た解の値が間違っていても，Netlibはなにも保証はしてくれない．なお，現在ではNetlibのソフトウェアを入手したり，情報を得るコードとしてXウインドウシステムで作動するXNetlibと呼ばれるものがある．

インデックスを調べてみると，直接解法による対称，非対称の疎行列計算が行なえるソフトウェアには，次のものがあることがわかった．

- c/meshach
- MA28
- sparse
- sparspack
- Y12M

詳細は以下の通りである．

```
------------------------------------------------------------------------------
lib  c/meschach
by David E. Stewart des@thrain.anu.edu.au
for  numerical linear algebra, dense and sparse, with permutations,
 error handling, input/output
lang C
prec double
#  o dynamic allocation, de-allocation, re-sizing and copying of objects
#  o input and output routines for these objects, and MATLAB
#  save/load format
#  o error/exception handling
#  o basic numerical linear algebra -- linear combinations, inner
#  products, matrix-vector and matrix-matrix products
#  o dense matrix factorise and solve -- LU, Cholesky, LDL^T, QR,
#  QR with column pivoting, symmetric indefinite (BKP)
#  o dense matrix factorisation update routines -- LDL^T, QR
#  o eigenvector/eigenvalue routines -- symmetric, real Schur
#  decomposition, SVD, extract eigenvector
#  o sparse matrix "utility" routines
#  o sparse matrix factorise and solve -- Cholesky and LU
#  o sparse incomplete factorisations -- Cholesky and LU
#  o iterative techniques -- pre-conditioned conjugate gradients,
#  LSQR, CGS, Lanczos, Arnoldi
#  o allowance for "procedurally defined" matrices in the iterative
#  techniques
#  true master copy: ftp thrain.anu.edu.au:pub/meschach/
master research.att.com
------------------------------------------------------------------------------

lib  harwell
for  sparse unsymmetric matrix routine MA28 from the Harwell library
editor Iain Duff
lang fortran
master ornl.gov
------------------------------------------------------------------------------

lib  sparse
for  large sparse systems of linear equations using LU factorization.
#  real and complex square
#  Besides being able to solve linear systems,
```

```
#  it is solves transposed systems, find determinants, multiplies
#  a vector by a matrix, and estimate errors due to
#  ill-conditioning in the system of equations and instability in
#  the computations. Sparse does not require symmetry
#  and is able to perform numerical pivoting (either diagonal or
#  complete) to avoid unnecessary error in the solution.
by Ken Kundert, Alberto Sangiovanni-Vincentelli. (sparse@ic.berkeley.edu)
lang C
editor Jack Dongarra
master ornl.gov
------------------------------------------------------------------------------

lib  sparspak
ref  Computer Solution of Large Sparse Positive Definite Systems,
 Prentice Hall, 1981.
by George and Liu
master ornl.gov
------------------------------------------------------------------------------

lib  y12m
for  sparse linear systems
by Zahari Zlatev, Jerzy Wasniewski and Kjeld Schaumburg
# Comp Sci; Math Inst; Univ Aarhus; Ny Munkegade; DK 8000 Aarhus
ref  Springer LNCS
master ornl.gov
------------------------------------------------------------------------------
```

この中で, 疎な非対称行列計算ができるものというと, ウォータールー大学の Alan George (現在, テネシー大学) と Joseph W-H Liu [7] が制作した *sparspak*を除いてすべてのものが利用可能であることがわかった. さらに, プログラムが最近開発されたものは, C 言語で記述したものが多いことにもちょっと驚かされた. 時代の流れにはやはり逆らうことができないようだ. これも unix OS とワークステーションの与えた影響が大というところであろう.

個々のソフトウェアについては, Netlib からエレクトリックメールまたは ftp を利用してファイルを入手すればよい. 大きなファイルを入手するには, エレクトリックメールではなく, ftp を使うとよい. E. Grosse [1] もこれを勧めているので, 小生もこれを利用している. 特に, 大きなファイルは, 小さな数十個のファイルに分割されていることが多い. 例えば, マルチグリッド法のコードである *pltmg*は, ほぼ 100 個の小さなファイルによって構成されている.

1:	ftp research.att.com (または, 192.20.225.2) [改行]
2:	*login*: netlib [改行]
3:	*password*: <自分のメールアドレス> [改行]
4:	<ファイルの転送の処理>
5:	quit [改行]

非対称な疎行列を扱うソフトウェアには, *MA28*, *sparse* , *Y12M*などがある. *Y12M*は, このソフトウェアのなかではもっとも古くから提供されているもので, デンマークの Aarhus 大学の Z. Zlatev 氏たち [17] によって開発されたものである.

*Y12M*と同様にこれらのソフトウェアの中では, 英国 Hawell 研究所の Iain Duff 氏が中心となって開発した *MA28* と呼ばれるライブラリー (Fortran 言語で記述されている) は意外に定評があるようだ. 先日, 東芝の福井氏と疎行列解法について話した折りにも,

このコードが話題にあがった. また, カリホルニア大学のバークレー校の Ken Kundert 氏たちによって開発された *sparse* と呼ばれるルーチンは, C 言語で記述されたもので, コードの使用ドキュメントもかなりしっかり記述されており, ちょっとびっくりさせられたシステムである.

この他に, Netlib には反復解法に基づくソフトウェアとして, 次の二つのものが提供されている.

- itpack
- slap

ソフトウェアの詳細は以下の通りである.

```
---------------------------------------------------------------------------
lib  itpack
for  Iterative Linear System Solvers
#  Jacobi method, SOR, SSOR with conjugate gradient acceleration
#  or with Chebyshev (semi-iteration - SI) acceleration.
by Young and Kincaid and the group at U of Texas.
#  kincaid@cs.utexas.edu oppe@scri1.scri.fsu.edu joubert@cs.utexas.edu
#  Center for Numerical Analysis; (512) 471-1242
#  RLM Bldg. 13.150; University of Texas at Austin; Austin TX 78713-8510
editor Bill Coughran
master research.att.com
---------------------------------------------------------------------------

lib  slap
for  iterative symmetric and non-symmetric linear system solution
#  Sparse Linear Algebra Package.
#  Included in this package are core routines to do Iterative
#  Refinement iteration, Preconditioned Conjugate Gradient
#  iteration, Preconditioned Conjugate Gradient iteration on the
#  Normal Equations, Preconditioned BiConjugate Gradient iteration,
#  Preconditioned BiConjugate Gradient Squared iteration, Orthomin
#  iteration and Generalized Minimum Residual iteration. Core
#  routines require the user to supply "MATVEC" (Matrix Vector
#  Multiply) and "MSOLVE" (Preconditiong) routines.
by Mark K. Seager & Anne Greenbaum
editor Jack Dongarra
master ornl.gov
---------------------------------------------------------------------------
```

itpack はテキサス大学のオースティン校の David M. Young 教授を中心として開発された SOR 法の系統を主流とする反復解法ルーチンを集めたものである. 以前はなにがしかのお金 ($200 程度) を払って利用するような形態をとっていたが, 今ではパブリックドメインのソフトウェアで無料で利用できる.

slap と呼ばれるソフトウェアは, ローレンスリバモア国立研究所の Mark K. Seager 氏とニューヨーク大学のクーラン研究所の Anne Greenbaum 教授が中心となって共同開発したもので, 共役勾配法の系統の反復解法を集めたものである.

疎行列を係数とする連立一次方程式の反復解法の多くは, 小国氏の著書 [11] でも紹介され, 彼らの開発したコードは本の付録のフロッピーディスクでも提供されているので, 我が国でもおなじみのものが多いと思う.

Netlib で特に驚いたことは, Lanczos 法を用いた固有値解法のソフトウェアが三つ (*lanczos*, *lanso*, *lanz*) 提供されていることである. *lanczos* [4] と呼ばれるソフトウェ

アは, 米国 IBM のヨークトン研究所の Jane K. Cullum 氏と Ralph A. Willouhby 氏が開発したものである. 筆者の見解によれば, Lanczos 法のソフトウェアとしては, かなり大がかりなものである. また, Neltlib にはマルチグリッド法のソフトウェアとして, *madpack* と *pltmg* [3] も提供されている.

最後に, Netlib を使う上で注意することは, 必要なファイルがすべて備わっていないこともある点である. また, ソフトウェアのインストールには makefile が用意されているがほとんどが SunOS 用に作られている. 他の計算機 (HP, IBM, DEC etc.) へのインストールには, 多少 makefile を変更する必要がある.

3. Bi-CGStab 法の収束性について

科学技術計算で現れる大規模な疎行列を係数とする連立一次方程式の解を求める反復解法は, 現在では決定的なものといえるものは存在していない. 非対称問題の解法には, 対称問題に対する共役勾配法に似た算法がいくつか存在する. しかし, 算法の収束は解くべき問題に強く従属しており, ある問題ではとてつもなく速い収束性を示すが, 別の問題に適用すると非常に遅い収束性を示したりすることがある. よって, 一概に解法の良し悪しを判断することができない. 非対称問題の解法には, 決定的な方法が存在しないといわれるのもこの事実からである.

過去 10 年くらいの間, BCG 法 (Bi-Conjugate Gradient method) [6] の系統の算法が人気が高いことも事実であるが, GCR 法系 [5] の算法もやはり捨て難い. わが国では, パラメータの決定を必要としない算法が人気が高いので, BCG 法を改良した算法や収束性の研究が行なわれてきた ([9, 15, 16, 10]).

オランダのデルフト大学の Sonneveld 教授によって提案された CGS 法 (Conjugate Gradient Squared method) は, ある収束条件を満足すれば BCG 法の 2 倍の収束特性を持つ算法である. しかし, 反復を繰り返すに従って残差ベクトルのノルムは, 大きく振動する性質があり, 丸め誤差の影響を受けやく, 発散してしまう可能性もある.

そこで登場したのが, BCG 法を改良した Bi-CGStab 法 [14] である. これは, デルフト大学の van der Vorst 教授によって提案された. Bi-CGStab 法は, BCG 法や CGS 法と比較すると, 一見, 収束性も滑らかで速い. 張, 藤野氏による残差多項式による解析の結果 [16] を見ると, ついに非対称行列問題の反復解法の決定版の登場かと思ってしまったのだが, 彼らの解析した例題が対称問題であるので, どっこい非対称問題のなかにはやはりだめな問題も存在するのである. しかし, 一般的にいえば Bi-CGStab 法はかなりの好成績をあげることも確かな事実である. おまけに演算量の面からみても, 超大型の計算を行なわない限り, 他の算法と比較して見劣りするものではない.

例えば, 次の Convection Diffusion 問題 [8] を考えることにしよう.

$$
\begin{aligned}
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \beta \frac{\partial u}{\partial x} &= f, \quad (x, y) \in \Omega \\
u &= xy(1-x)(1-y), \quad (x, y) \in \partial\Omega
\end{aligned}
$$

表 3.1 標準的な 5 点差分を用いた場合の反復回数 (前処理なし)

β	BCG	CGS	Bi-CGStab	GCR
1	1795	—†	1488	3716
100	—†	—†	747	3565

($\epsilon \leq 10^{-14}$, †,,—" は 10000 回の反復で収束しない.)

表 3.2 修正風上差分を用いた場合の反復回数 (前処理なし)

β	BCG	CGS	Bi-CGStab	GCR
1	1914	—†	1613	3719
100	—†	—†	721	3603

($\epsilon \leq 10^{-14}$, †,,—" は 10000 回の反復で収束しない.)

ただし, Ωは正方形領域であり, $\partial\Omega$はその境界であるものとする. また, β は定数であり, f は次の条件を満足するものとする.

$$f = 2x(x-1) + y(y-1)[2 - \beta(1-2x)]$$

この正方形領域を $x-y$ 両軸方向ともに 399($h = 1/400$) 当分割し, 5 点差分で近似すると 159201 個の変数を持つ連立一次方程式

$$Ax = b$$

を得ることはおわかりのことと思う. ここで行なったすべての計算は, Sun 4/490, SPARC Station 2 および SPARC 10/31 を利用した.

おのおのの計算結果を表 3.1 と表 3.2 にあげることにする. ただし, この場合には行列の前処理を行なっていない. 得られた計算結果はほぼ良好で, 標準的な 5 点差分で離散化したものも, 修正風上差分で離散化したものも Bi-CGStab 法が好成績をあげている. 計算結果の詳細は, Nodera [10] を参照してほしい.

次に, Bi-CGStab 法があまりよい収束性をしない例をあげることにする. 例えば, 次のような分子構造を持つものを考える. これは修正風上差分の要素の一つに修正を加えたものである.

$$(A)_{ij} = \begin{pmatrix} & -1 & \\ -\left(1-\frac{\beta h}{2}\right)^{-1} & 2+\beta h+2\left(1+\frac{\beta h}{2}\right)^{-1} & -\beta h-\left(1+\frac{\beta h}{2}\right)^{-1} \\ & -1 & \end{pmatrix}$$

とりあえず, $\beta = 1$ と$\beta = 100$ について, 行列の前処理を行なわない計算結果 (残差ベクトルのノルムの収束性 v.s. 反復回数) を図 3.1 と図 3.2 に示すことにする.

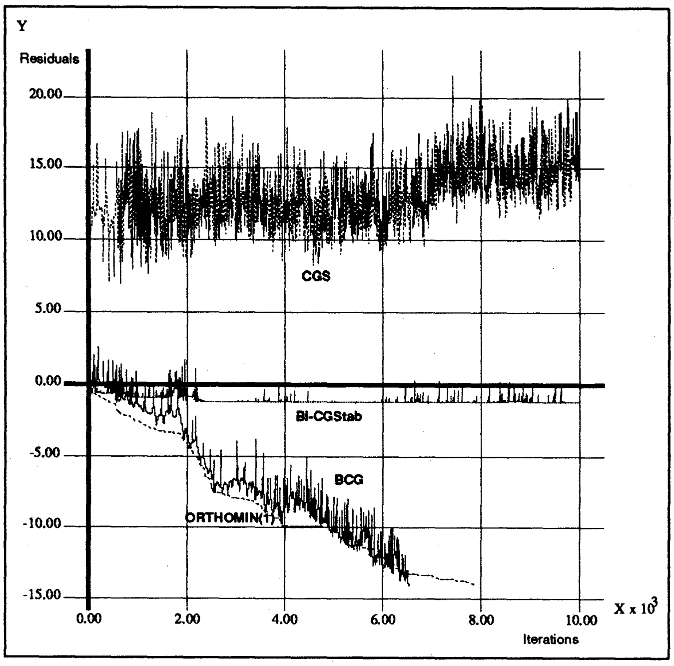

図 **3.1** 残差 vs. 反復回数, $h = 1/400$, $\beta = 1$, 前処理なし.

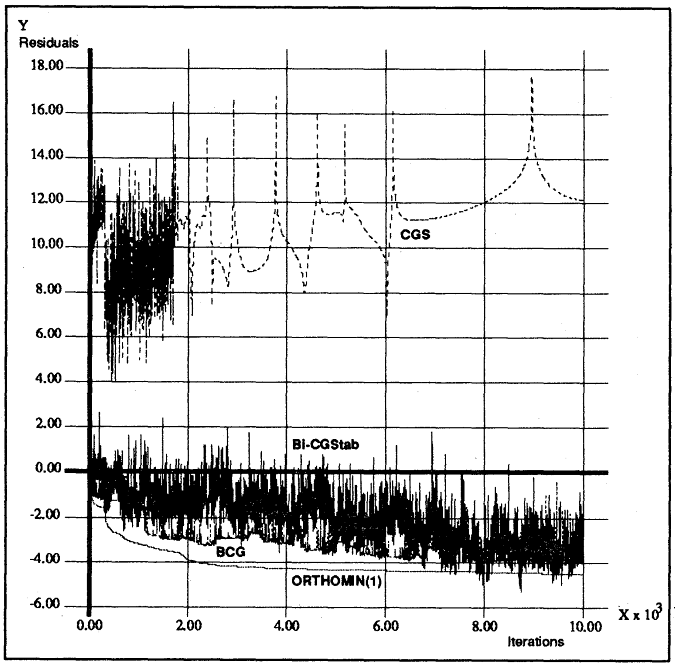

図 **3.2** 残差 vs. 反復回数, $h = 1/400$, $\beta = 100$, 前処理なし.

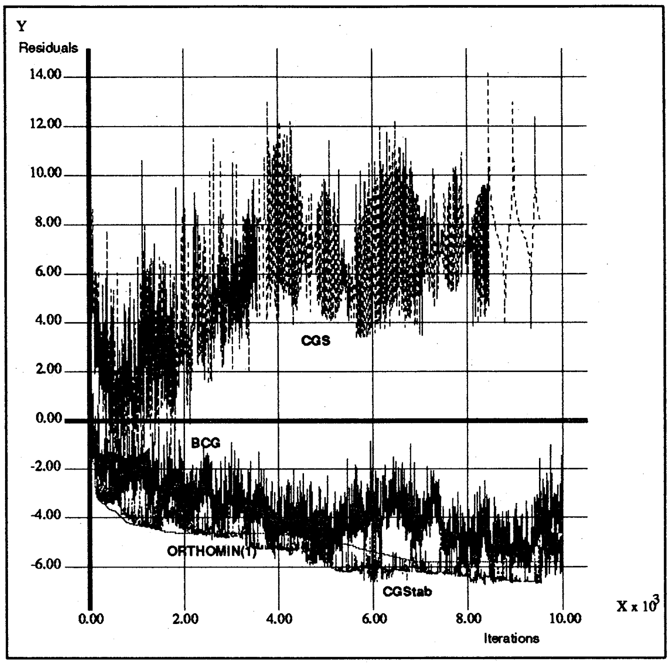

図 3.3 残差 vs. 反復回数, $h = 1/400$, $\beta = 100$, 前処理:ILU(不完全 LU) 分解.

図 3.1 と図 3.2 の収束性を比較してみると, Bi-CGStab 法はけして速い収束性を示しているとはいいがたい. へたをすると, Bi-CGStab 法は BCG 法や ORTHOMIN(1) 法にも負けてしまうことがある. 特に, ORTHOMIN(1) 法は, 反復 1 回に対する演算回数が他の算法に比べて少ないので, これは計算時間の割り合いからいっても一番速い例の一つである. また, CGS 法はこの二例についていえば, 発散して全然問題外といえる. CGS 法は算法の性質上, 丸め誤差の影響を受けやすく, 問題が大きくなればなるほど発散する傾向にある. しかし, 問題が小さければ, CGS 法は良好の収束性を示すこともある ([9]).

図 3.3 は, $\beta = 100$ の場合に ILU(不完全 LU) 分解を行列の前処理として用いた場合の収束性を示したものである. この例では不完全 LU 分解を用いても, すべての解法に対してあまり思わしい収束性を示してはいない. CGS 法は, この場合にも発散してしまう.

4. おわりに

近年の計算機の環境の変化に伴い, 科学技術計算の分野にもダウンサイジングが行なわれ, メインフレーム中心の計算環境からワークステーションやパーソナル計算機中心の分散型にかわろうとしている. 同時に, ソフトウェアもメインフレーム型のものではなく, かゆい所に手が届くとでもいうのか, 使い易さを重点においたものが現れてきている. 疎行列計算も序々にではあるが, 超大型の計算を除いてワークステーションでも

計算が行なえる環境ができあがりつつある.

計算機の環境の変化に伴い, 従来数値計算の分野ではプログラムを記述するといえば Fortran 言語一辺倒であったものが, 若い研究者を中心として C 言語やオブジェクト指向型の言語に変りつつある. しかし, 計算機環境のダウンサイジングにも問題がないわけではない. 最大のものは, ソフトウェアのメインテナンスの問題をあげることができるのではないか. また, 評価の高いソフトウェアは, お値段もそれなりの高額であるので, 1 つの CPU ごとに料金を払っていたらとんでもない高額の料金になることも事実である.

次々に生まれるニュー・テクノロジー. 目まぐるしく変化する計算機環境. かって日本の農家が経験したことであるが, 農業の機械化によって, ある年代以降の堕ちこぼれ (社会的な断絶) を作ったことがある. 新しい計算機の環境が生まれるに従って, ある年代以降の計算機環境への順応性が失われて, あれよあれよというまに堕ちこぼれになる時代は, すぐそこにあるように思われてならない. これも unix OS の弊害の一つかもしれない. 小生の一人ごとであればよいのだが.

参考文献

[1] E. Grosse, "*Netlib news: big files* ," AT&T Bell Laboratory, Aug. 24, 1991.

[2] J. Dongarra, and E. Grosse, „*Distribution of Mathematical Software via Electric mail*," Comm. ACM. 30(1987), pp. 403–407.

[3] R. Bank, "*PLTMG User's Guide*," SIAM publication, 1991.

[4] J. K. Cullum and R. A. Willouby, "*Lanczos Algorithm for Large Symmetric Eigenvalue Computations I and II*," Birkhauser, 1985.

[5] S. C. Eisenstat H. C. Elman and M. H. Schultz, "*Variational Iterative Methods for Nonsymmetric Systems of Linear Equations*," SIAM J. Numer. Anal., Vol. 20, pp. 345–357 (1983).

[6] R. Flecher, "*Conjugate Gradient Methods for Indefinite Systems*," Lecture Notes in Math., Vol. 506 (1976), pp. 73-891.

[7] A. George and J. W. Liu, "*Computer Solution of Large Sparse Positive Definite Systems*," Prentice-Hall, 1981.

[8] D. R. Kincaid and D. M. Young, "*Adaptive Iterative Algorithms Developed for Symmetric Systems to Nonsymmetric Systems*," Elliptic Problem Solvers, pp. 353–359 (1981).

[9] T. Nodera, "*The use of a CGS Method for Convective Diffusion Problem*," Computational Techniques and Applications (Noye, J. and Fletcher C. eds.), North-Holland, pp. 529-538 (1988).

[10] ———, "*A Note on CGStab Algorithm*," Proc. of 18th South African Sympo. on Numer. Math., 1992, pp. 157–167.

[11] 小国力編, 『行列計算ソフトウェア (WS, スーパーコン, 並列計算機)』, 丸善, 1991.

[12] 島崎真昭, 応用数理における電子メール/ ニュースの利用, 応用数理 Vol. 2, No. 4, 1992.

[13] P. S. Sonneveld, "*CGS, A Fast Lanczos Type Solver for Nonsymmetric Linear Systems*," Delft University, Report 84-16(1984), and SIAM J. Sci. Statist. Comput., Vol. 10(1989), pp. 36–52.

[14] H. A. Van der Vorst, "*Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems*," SIAM J. Sci. Stat. Comput., Vol. 13(1992), pp. 801-812.

[15] H. Watanabe, S. Doi, "*A Comparison of Bi-CGStab and CGS Methods for Convectioin-Diffusion Equations*," Advances in Numerical Methods for Large Sparse Sets of Linear Equations, No. 7(1991)1, pp. 19-26.

[16] 長紹良, 藤野清次, CGS 法と Bi-CGSTAB 法の残差多項式, 京都大学数理解析研究所講究録 No. 791, pp. 1–15 (1992).

[17] Z. Zlatev, J. Wasniewski and K. Schaumburg, "*Y12M: Solution of Large and Sparse Systems of Linear Algebraic Equations*," Lecture Notes in Computer Science No. 121, Springer-Verlag, 1981.