ハイパフォーマンス
コンピューティング　48－9
（1993．8．20）

マルチグリッド前処理付き自乗共役勾配法

襲田　勉　建部 修見　小柳 義夫
東京大学理学部情報科学科

概要

自乗共役勾配法の前処理として修正不完全LU分解が良く使われているが、この前処理は収束率の改善が不十分であり、しかも並列性が低く並列計算機上での効率的な実装が難しい。この論文では非対称係数行列の問題に対するマルチグリッド前処理付き自乗共役勾配法を提案する。マルチグリッド法は多くの格子点を使うことで収束を加速させる方法であるが、粗い格子点を使うことで非対称性が増し余り粗い格子点まで使用することができない。またこの方法は様々な問題に対し効率的な収束を行なわせるためには restriction や prolongation をより正確なものにしたり、緩和法をより強力にする必要があるがその場合並列性という点で難がある。この研究では非対称係数行列のときにも自乗共役勾配法の前処理としてこのマルチグリッド法を使うことでそれらの欠点を解消することができしかも収束が速くなることを示す。

The Multigrid Preconditioned Conjugate Gradient Squared Method

Tsutomu Osoda, Osamu Tatebe and Yoshio Oyanagi
Department of Information Science, Faculty of Science, the University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113, JAPAN

Abstract

The modified incomplete LU decomposition preconditioner, which is usually applied to the conjugate gradient squared method, does not improve convergence rate sufficiently and cannot be implemented efficiently on parallel computers. This paper proposes the multigrid preconditioned conjugate gradient squared method. The multigrid method shows the rapid convergence for symmetric problem exploiting grids of difference sizes but in the unsymmetric case, for too coarse grid, M-matrix property is broken. When unsymmetry of the matrix is strong, the parallelism is spoiled because it is necessary to make the restriction and the prolongation precise or to make smoother strong. This paper shows that when the multigrid method is used as the preconditioner of the conjugate gradient squared method, the faults of the multigrid method disappear and this preconditioner shows fast convergence rate.

1 マルチグリッド前処理付き自乗共役勾配法

非対称な係数行列を持つ問題の解法として、前処理付き自乗共役勾配法がよく用いられている。この研究では自乗共役勾配法の前処理として、マルチグリッド前処理を提案する。マルチグリッド前処理付き自乗共役勾配法とはマルチグリッド法を前処理として施した自乗共役勾配法で、問題を $A\boldsymbol{x} = \boldsymbol{b}$ とすると表1のように表される。

表 1. 前処理付き自乗共役勾配法

$\boldsymbol{r}_0 = C^{-1}(\boldsymbol{b} - A\boldsymbol{x});$
$\boldsymbol{p} = \boldsymbol{e} = \boldsymbol{r} = \boldsymbol{r}_0;$
$\mu_1 = (\boldsymbol{r}_0, \boldsymbol{r});$
while $||\boldsymbol{r}|| > eps||\boldsymbol{b}||$ do
{
  $\boldsymbol{q} = C^{-1}A\boldsymbol{p};$
  $\alpha = \frac{\mu_1}{(\boldsymbol{r}_0, \boldsymbol{q})};$
  $\boldsymbol{h} = \boldsymbol{e} - \alpha\boldsymbol{q};$
  $\boldsymbol{e} = \boldsymbol{e} + \boldsymbol{h};$
  $\boldsymbol{q} = C^{-1}A\boldsymbol{e};$
  $\boldsymbol{x} = \boldsymbol{x} + \alpha\boldsymbol{e};$
  $\boldsymbol{r} = \boldsymbol{r} - \alpha\boldsymbol{q};$
  $\mu_2 = \mu_1;\ \mu_1 = (\boldsymbol{r}_0, \boldsymbol{r});\ \beta = \frac{\mu_1}{\mu_2};$
  $\boldsymbol{e} = \boldsymbol{r} + \beta\boldsymbol{h};$
  $\boldsymbol{p} = \boldsymbol{e} + \beta(\boldsymbol{h} + \beta\boldsymbol{p});$
}

表 2. マルチグリッド前処理

Vector MG(L_l, $\boldsymbol{f}$, $\boldsymbol{x}$, γ, μ)
{
  if (l == coarsest_level) Solve $L_l\boldsymbol{x} = \boldsymbol{f}$;
  else {
    $\boldsymbol{x}$ = pre_smoothing(L_l, $\boldsymbol{f}$, $\boldsymbol{x}$, μ);
    $\boldsymbol{d}$ = restrict($\boldsymbol{f} - L_l\boldsymbol{x}$);
    $\boldsymbol{\nu}$ = initial_$\boldsymbol{x}$;
    repeat (γ) $\boldsymbol{\nu}$ = MG(L_{l-1}, $\boldsymbol{d}$, $\boldsymbol{\nu}$, γ, μ);
    $\boldsymbol{x} = \boldsymbol{x}$ + prolongate($\boldsymbol{\nu}$);
    $\boldsymbol{x}$ = post_smoothing(L_l, $\boldsymbol{f}$, $\boldsymbol{x}$, μ);
  }
  return $\boldsymbol{x}$;
}

ここで C^{-1} はマルチグリッド前処理を表している。また、マルチグリッド前処理は表2のように再帰的に表される。特に γ が1のときにはVサイクルマルチグリッド法と呼ばれ2のときにはWサイクルマルチグリッド法と呼ばれる。

2 問題

以下の議論では次のような二次元の移流拡散方程式を離散化した問題を考える。

$$\mathrm{div}\Big(-k\nabla u + \boldsymbol{b}u\Big) = f(x,y) \text{ in } \Omega \subset \Re^2 \tag{1}$$

$$\text{with } u = 0 \text{ on } \Gamma \tag{2}$$

ここで Γ は領域 Ω の境界である。数値実験では対象を二次元正方領域とし、中心差分法を使って方程式を離散化し直すこととする。簡単のため境界条件は固定境界条件とする。この方程式を格子の間隔の長さを h のメッシュで $m \times m$ に区切り、中心差分の公式を使うことによって

$$\left(-1 - \frac{b_x h}{2k}\right) u_{i-m} + \left(-1 - \frac{b_y h}{2k}\right) u_{i-1} + 4u_i + \left(-1 + \frac{b_y h}{2k}\right) u_{i+1} + \left(-1 + \frac{b_x h}{2k}\right) u_{i+m} = \frac{f_i h^2}{k} \tag{3}$$

$$\boldsymbol{b} = \begin{pmatrix} b_x \\ b_y \end{pmatrix} \tag{4}$$

$$f_i = f(x_i) \tag{5}$$

の離散化した式が得られる。実際に計算に使うために離散化する場合には解の安定性から最も細かい格子点上では

$$\frac{|b|h}{2k} < 1 \tag{6}$$

の条件を満たすようにする。

3 セルペクレ数

移流拡散方程式において、解の安定性もしくは解の収束性の指標となるものは離散化した方程式の係数行列のセルペクレ数である。この章ではマルチグリッド前処理を行なう際のセルペクレ数の変化を考える。式(3)のセルペクレ数は

$$\frac{|b|h}{k} \tag{7}$$

として定義することができる。マルチグリッド法は必要とされる格子間隔よりも粗い格子を使って収束を加速させる方法であり、その粗い格子上の方程式の立て方には大きく二つの方法がある。一つは粗い格子上で元の移流拡散方程式を離散化し直す方法であり、もう一つはマルチグリッド法の restriction、 prolongation のオペレータを用いて粗くする方法である。

前者の場合の粗い格子点上における方程式の係数行列の性質を考える。一回粗い格子点上に残差を移すたびに常に格子の幅は二倍になるので、h を一番細かい格子の幅として固定すれば n 番目に粗い格子点に移すとセルペクレ数は

$$\frac{2^n|b|h}{k} \tag{8}$$

と表すことができる。つまり粗い格子になればなるほど、セルペクレ数が上昇するためその格子上での方程式の解の安定性や収束性は悪くなることがわかる。

係数行列が M 行列であれば解は振動しない。これを利用すると粗い格子点上で方程式を離散化し直す場合には、格子を n 段使うマルチグリッド前処理が効率的に収束するためには、

$$\frac{|b|h}{k} \le 2^{1-n} \tag{9}$$

が成り立つ必要があり、 restriction、 prolongation のオペレータを用いて粗くする場合には

$$\frac{|b|h}{k} \le \frac{1+2^{1-2n}}{2^n+2^{-n-1}} \tag{10}$$

が成り立つ必要があることがわかる。

4 数値実験と考察

この章では数値実験を行ないマルチグリッド前処理付き自乗共役勾配法(MGCGS)の有効性を示す。表3、表4はそれぞれ不完全 LU 分解付き前処理自乗共役勾配法(MILUCGS)と MGCGS の収束までの計算時間と反復回数である。計算に用いた計算機は HP 9000/720 である。計算に使用した問題は式(1)の二次元の移流拡散方程式で境界は固定境界で拡散係数は一定で、ソース項は領域内一定であるとする。移流項の係数は 32 に保ってある。前処理として使ったマルチグリッド法はVサイクル1反復を用いて 1×1 の格子点まで粗くしている。マルチグリッド法のスムージング法としてはガウスザイデル法を用いている。粗い格子点上でも係数行列

メッシュの大きさ	計算時間 (秒)	反復回数
63×63	3.2	43
127×127	45.6	108
255×255	494.3	216

表 3. MILUCGS

メッシュの大きさ	計算時間 (秒)	反復回数
63×63	1.7	6
127×127	10.6	6
255×255	56.5	6

表 4. MGCGS

がM行列の必要条件を崩さぬよう、係数行列の非対角要素で正の数となるところは強制的に 0 となるようにしてある。収束条件はどちらの方法でも 10^{-8} としてある。

MILUCGS に比べ非常に少ない計算時間で収束していることがわかる。また問題の大きさが大きくなればなるほど MGCGS の方が有利になっていることもわかる。このことよりマルチグリッド法の特徴であった解が収束するまでの反復回数がメッシュの大きさには関係しないという性質を今回提案した MGCGS も持っていることがわかった。係数行列が正定値対称行列の場合には [4] すでに反復回数がメッシュサイズに依存しないことは示されているが、係数行列が非対称の場合にも成り立っていることがわかる。

次に移流項の係数の大きさを変化させたときの MGCGS の収束性を調べる。その移流項の大きさは移流拡散方程式 (1) の $|b|$ である。ここで使った問題は二次元の移流拡散方程式で境界は固定境界条件で、拡散係数は一定にしソース項は領域内ですべて 1 であるとする。二次元の領域が x 軸、y 軸から張られる空間からなっているとすると、移流項は y 軸方向においては 0 に固定し x 軸方向で値を変化させている。マルチグリッド法の様々なスキームを調べるためにどの段階まで粗くするかは変化させている。また収束条件は以下の場合でも相対残差が 10^{-8} としてある。図 1はマルチグリッド法で粗い格子点で方程式を完全に解いたものであり図 2はマルチグリッド法で粗い格子点で方程式を完全には解かなかった場合である。図 3は MGCGS で粗い格子点で方程式を完全に解いたものであり図 4は MGCGS で粗い格子点で方程式を完全には解かなかった場合である。図では横軸はその移流項の係数を縦軸は収束するまでの反復回数を示している。一つの図の中には三本の線があるがそれらはマルチグリッド法でそれぞれ 7 段、6 段、5 段の格子点を用いることを示している。7 段の場合は 1×1 の格子点まで使い、6 段の場合には 3×3 まで、5 段の場合には 7×7 まで使うことになる。図 1と図 3を見比べることによりマルチグリッド法では移流項の係数が大きな値を持つときには発散してしまうのに対し、MGCGS ではそのような大きさの係数に対しても反復回数は少し増えるが安定に収束している。マルチグリッド法で解に発散があるのは移流項の係数が大きな値を持つときには、粗い格子点においてセルペクレ数が上昇し係数行列の非対称性が増すために、解が不安定になるためである。つまり MGCGS はマルチグリッド法と自乗共役勾配法を組み合わせることで収束性が良くなる。

図 1から 127×127 のメッシュで離散化したとき、粗い格子点上で完全に方程式を解いた場合には最も粗い格子点が細かければ細かいほど反復回数は少なくなる。それに対し粗い格子点上で完全に方程式を解かなかった場合には係数が 32 以下のときには 1×1 まで粗くしたほうが、それよりも大きな値では 7×7 まで粗くしたところで止めたほうが反復回数が少なくなっていることがわかる。図 3から同様のことが MGCGS でも成り立っていることがわかる。最も粗い格子点上で完全に解く場合には粗い格子点を細かいところで止めれば、それだけ粗い格子点上の解が細かい格子点上の解に近付くために反復回数が少なくなり、したがって完全に解く場合

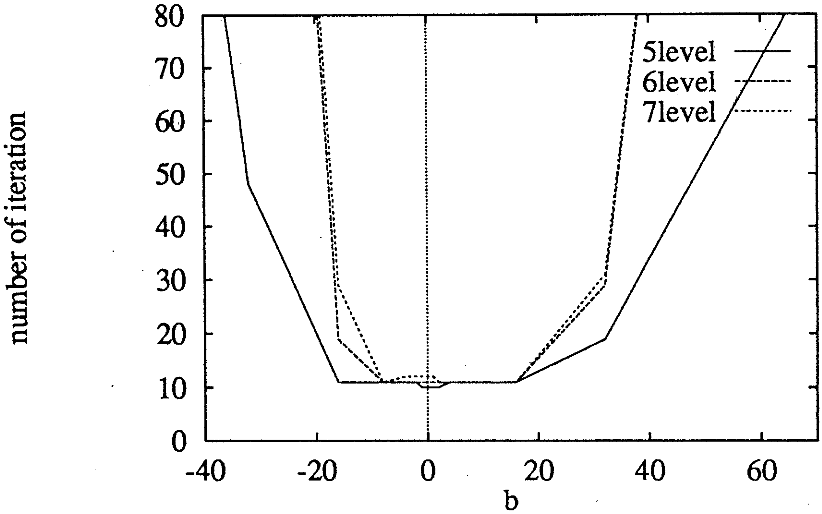

図 1. MG 法で 127×127 のメッシュを使い粗い格子点上で方程式を完全に解いたとき

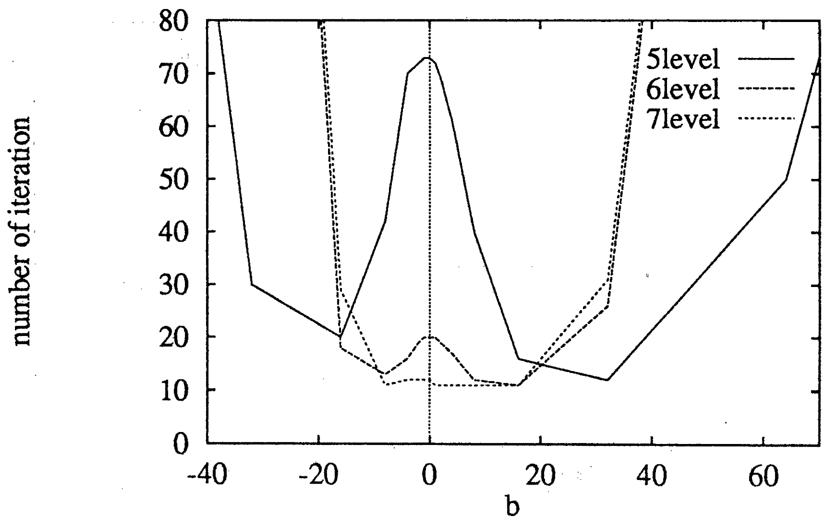

図 2. MG 法で 127×127 のメッシュを使い粗い格子点上で方程式を完全に解かなかったとき

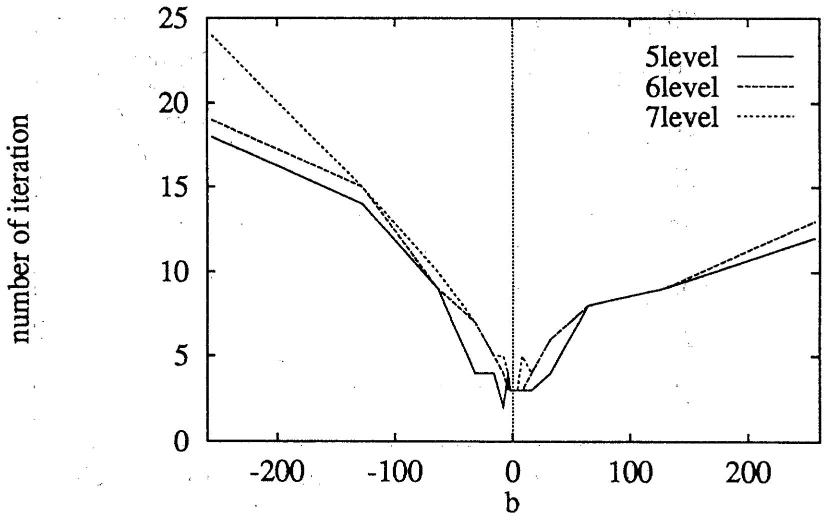

図 3. MGCGS で 127×127 のメッシュを使い粗い格子点上で方程式を完全に解いたとき

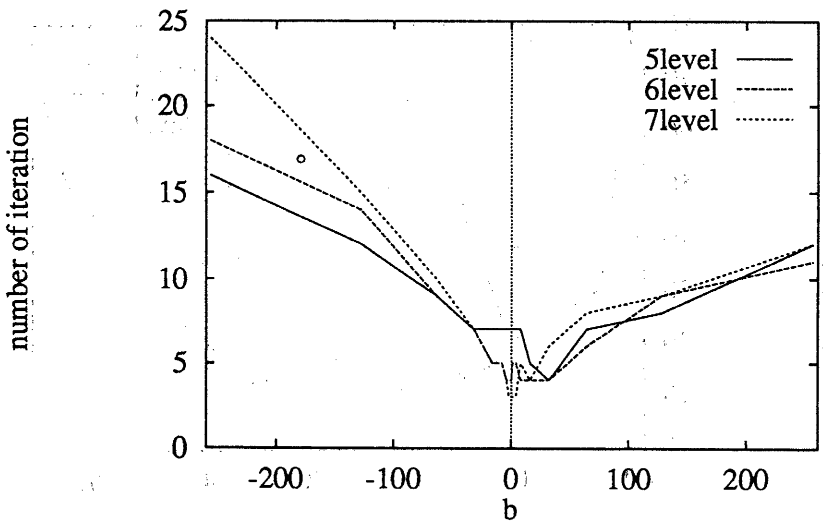

図 4. MGCGS で 127×127 のメッシュを使い粗い格子点上で方程式を完全に解かなかったとき

には最も粗い格子点が細かければ細かいほど反復回数は少なくなることは理解できる。一方、完全に解かない場合には 1×1 に近付ければそれだけ粗い格子点上で方程式は完全に解かれるようになるのでどの段階でも 1×1 まで粗くしたほうが反復回数が少なくなりそうであるが、実はそのようにはならない。それは移流項の大きさが大きいときには、粗い格子点上で係数行列が非対称性を増しているために、粗くすればするほど解の安定性がなくなるためである。現実に最も粗い格子点上でセルペクレ数を計算すると 7×7 を最も粗い格子点とした場合には $|b| = 32$ のときには $\frac{32}{2\times8} = 2$ となり離散化の条件を破っていることになる。したがって $|b| = 32$ のときにはそれよりも格子点を粗くした場合にはさらに安定性がなくなりマルチグリッド法の反復回数は増えることになる。

以上の考察から実際にマルチグリッド法を前処理として用いるためには、127×127 で離散化したときには $|b| < 32$ なら 1×1 まで粗くし、それ以上のときにはそれよりも細かい格子点で止めるべきである。

5 まとめ

この論文では、自乗共役勾配法の前処理としてマルチグリッド法を使うことを検討してきた。この方法は並列性が高いという点で不完全 LU 分解前処理付き自乗共役勾配法に比べ並列計算機上への効率的実装が容易である。非対称な係数行列を持つ場合にはマルチグリッド法のみを使った場合粗い格子点上で係数行列の非対称性が増し、移流項の係数が大きな値のときにはそれによって収束性が悪化することがあることを簡単な計算と数値実験によって確かめた。自乗共役勾配法の前処理としてマルチグリッド法を用いることで解が安定して収束するようになることを数値実験により確かめた。従来使われていきた不完全 LU 分解の前処理よりも速く収束し、解が収束するまでの反復回数が一定であることから大規模な問題なればなるほどマルチグリッド前処理が優勢になることを数値実験で確かめた。それらの実験によって正定値対称な係数行列を持つときと同様にマルチグリッド前処理が非常に有効であることが示された。

6 今後の課題

数値実験の結果から一様な二次元の移流拡散方程式を解く場合にはマルチグリッド前処理付き自乗共役勾配法は従来良く使われてきた不完全 LU 分解前処理をした自乗共役勾配法に比べ非常に収束性が良く優れた前処理であることが示された。対称な係数行列を持つ場合には共役勾配法の前処理としてマルチグリッド前処理が有効でるということは固有値解析を行なうことで確かめることができたが、非対称な係数行列の場合には同じ方法で確かめることはできない。しかしながら類似する方法により有効性を確かめたいと思う。今回は境界が固定境界かつ一様な場だけを扱ったが、それ以外のものについても、今回提案したマルチグリッド前処理付き自乗共役勾配法を適用しその収束性などを研究していきたいと思う。今回は逐次計算機上で計算を行なったが、本来の目的であった高並列で効率の良い前処理を目指すという立場から、この方法を並列計算機上で実装し並列計算機上においても高並列で効率の良い前処理であることを示していきたいと思う。

参考文献

[1] Wolfgang Hackbusch. *Multi-grid Methods and Application.* Springer Verlag, 1985.

[2] Osamu Tatebe. "The Multigrid Preconditioned Conjugate Gradient Method". In *the proceedings of Sixth Copper Montain Conference, NASA CP*, 1993 (to appear).

[3] Pieter Wesseling. *AN INTRODUCTION TO MULTIGRID METHODS.* JOHN WILEY AND SONS, 1991.

[4] 建部 修見 小柳義夫. "マルチグリッド前処理付き共役勾配法の並列化". JSPP '93 論文集 pp.387–394, May 18 1993.

[5] 村田 健郎 名取 亮 唐木幸比古. 大型数値シミュレーション. 岩波書店, 1990.