

異方性拡散方程式に対する MGCG 法

建部 修見 小柳 義夫
東京大学理学部情報科学科
(O. Tatebe, Y. Oyanagi)

要旨

緩和法に高並列な Red-Black 対称 Gauss-Seidel 法を使う MGCG 法は, 異方性の強い方程式に対しては MG 法と同様に収束が遅い. MG 法では収束を改善するために semicoarsening を行うか, 緩和法に line relaxation を使用している. この研究では MGCG 法でもこれらの方法が有効であることを示すとともに MGCG 法が robust な解法であることを示す.

1 はじめに

MGCG 法 [4] はマルチグリッド (MG) 法を前処理とする共役勾配法で, 拡散係数に強い非連続性のあるポアソン方程式に対して非常に有効な解法である. またこの解法は非常に並列性が高いため, 効率良く分散メモリ型並列計算機に実装を行なうことができ, 高い収束性を保ったままほぼリニアなスピードアップをすることが可能であった [5].

一方, 異方性拡散方程式の解法としては MG 法が良く用いられるが, この MG 法の緩和法としては line または plane relaxation, あるいは不完全 LU 分解が効果的である [1]. また MG 法の収束性をさらに高めるために coarsening を semicoarsening にする. また別の MG 法の収束性を高めるアプローチとして, そのいくつかの可能性のある semicoarse されたグリッドを全て用いる multiple semicoarsed grid (MSG) アルゴリズム [2, 3] がある.

今回はそれらのテクニックを MGCG 法で用い, MGCG 法が異方性拡散方程式に対しても有効であることを示す.

2 異方性方程式

この研究で扱う異方性方程式は,

$$-\varepsilon u_{xx} - u_{yy} = f \quad (1)$$

の軸を β 回転した式で,

$$-(\varepsilon c^2 + s^2)u_{xx} - 2(\varepsilon - 1)csu_{xy} - (\varepsilon s^2 + c^2)u_{yy} = f \quad (2)$$

である. ここで $c = \cos \beta$, $s = \sin \beta$, $\varepsilon > 0$ である.

2.1 離散化方法

式 (2) を離散化する場合, u_{xy} を離散化する必要がある. この離散化の方法にはいくつか考えられるが, 代表的な離散化のステンシルは格子間隔を h とすると次の三つである.

$$-2u_{xy} \simeq \frac{1}{h^2} \begin{bmatrix} 0 & 1 & -1 \\ 1 & -2 & 1 \\ -1 & 1 & 0 \end{bmatrix} \cdots (A), \quad \frac{1}{h^2} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \cdots (B), \quad \frac{1}{2h^2} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \cdots (C) \quad (3)$$

ここで (C) の離散化のステンシルは (A) と (B) の平均をとったものとなっている. これらの離散化誤差はそれぞれ,

$$\begin{aligned} -2u_{xy} &= \frac{1}{h^2} \begin{bmatrix} 0 & 1 & -1 \\ 1 & -2 & 1 \\ -1 & 1 & 0 \end{bmatrix} + \left(\frac{1}{3}u_{xxxy} + \frac{1}{2}u_{xxyy} + \frac{1}{3}u_{xyyy} \right) h^2 + O(h^4) \\ &= \frac{1}{h^2} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} + \left(\frac{1}{3}u_{xxxy} - \frac{1}{2}u_{xxyy} + \frac{1}{3}u_{xyyy} \right) h^2 + O(h^4) \\ &= \frac{1}{2h^2} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} + \left(\frac{1}{3}u_{xxxy} + \frac{1}{3}u_{xyyy} \right) h^2 + O(h^4) \end{aligned}$$

である.

(A) の離散化ステンシルを用いた場合はこの項の係数が負となる場合に, また (B) を用いた場合はこの項の係数が正となる場合に, そして (C) を用いた場合はこの項が存在すれば, 非対角要素が正になるので明らかにいずれの係数行列も M -matrix にならない. また今回は $\epsilon \ll 1$ としているため, (A) の離散化ステンシルを使うのがこの三つのステンシルでもっとも係数行列の対角の要素が大きくなり, (C) の離散化ステンシルを使うのがもっとも対角要素が非対角の要素に比べ大きくなる. ここでももちろん (C) を使ったからといって, u_{xy} の項が存在すれば対角優位にはならない.

3 semicoarsening

異方性のある方程式を扱う場合, point relaxation は非常に貧弱過ぎ, 実際 Red-Black 対称 Gauss-Seidel 法を緩和法とする MGCG 法の収束は遅くなる. しかしながら並列計算機で効率良く実装を行なうためには point relaxation の並列性は魅力的であるため, まず緩和法はそのまま coarsening を semicoarsening とする MGCG 法について考察を行なう.

粗い格子の格子間隔を一度に x, y 方向とも倍にして粗い格子を作る方法を full coarsening というのに対し, semicoarsening は例えば図 1 の様にまず x 方向の格子間隔を倍にしたものを粗い格子とし, 次に y 方向を倍にしたものを次の粗い格子とする coarsening の方法である. この例は水平方向の semicoarsening で, 水平方向を始めに粗くしている. したがって n グリッドレベルの full coarsening で最も粗いグリッドは $2n-1$ グリッドレベルの semicoarsening の最も粗いグリッドと等しくなる.

semicoarsening では一度に一方向しか粗くしないため prolongation はその方向での線形補間とする. restriction はその prolongation に対し adjoint なものを使う. つまり prolongation, restriction をそれぞれステンシ

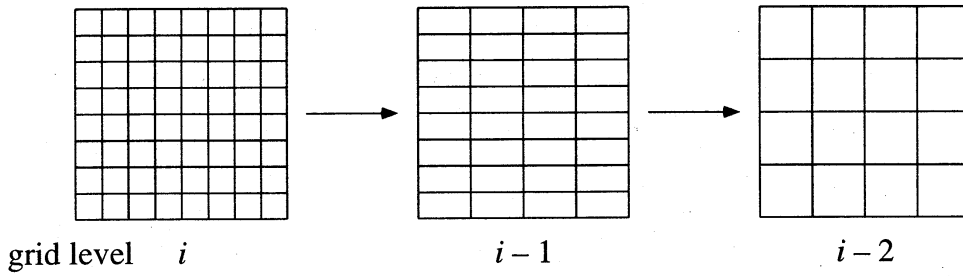


図 1. 水平方向での semicoarsening

表 1. MGCG 法の平均収束率と反復回数

β	# smooth.	1	2	SC
90	conv. rate	.558	.438	.356
	# iter.	41	28	23
70	conv. rate	.371	.266	.193
	# iter.	24	18	15
45	conv. rate	.293	.203	.231
	# iter.	19	16	16
20	conv. rate	.371	.266	.344
	# iter.	24	18	22
0	conv. rate	.558	.438	.652
	# iter.	41	28	54

ルで表すと,

$$p = \frac{1}{2} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}, \quad r = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \quad (4)$$

となる.

CG 法の反復回数は前処理後の係数行列の固有値分布によるため, semicoarsening を行なった MG 前処理後の固有値分布を調べればその効果を知ることができる.

図 2 は semicoarsening による収束率の改善がもっとも顕著な $\beta = 90^\circ$ の時の固有値分布を表している. 図 2 により, 緩和法を 1 反復, 2 反復したもの比べ, semicoarsening を行なったもののほうが固有値がより密集し, 固有値分布がより改善されていることがわかる. semicoarsening を行なうことにより, 使用するグリッド数は約 2 倍になり, 一反復あたりの計算量, 記憶量は約 1.5 倍になるため, 緩和法を 2 反復したもの比べ改善されていないと, semicoarsening の意味がないが, この結果よりそれ以上に改善されていることが分かる.

表 1 は同じ条件の下で, 問題の大きさを 32^2 とした時の平均収束率と反復回数である. この実験では β をいろいろ変えて収束率の変化をみている.

この結果より, semicoarsening が有効なのは $|\beta - 90^\circ| < 45^\circ$ のとき, つまり式 (2) の係数が大きな方向で semicoarsening した場合であることが分かる. そうでない場合は余り有効ではなく, この実験では $\beta = 20^\circ$ ではちょうど緩和法を 1.5 反復した感じとなっている. この時, 固有値分布は緩和法を 1 反復行ったものと 2 反復

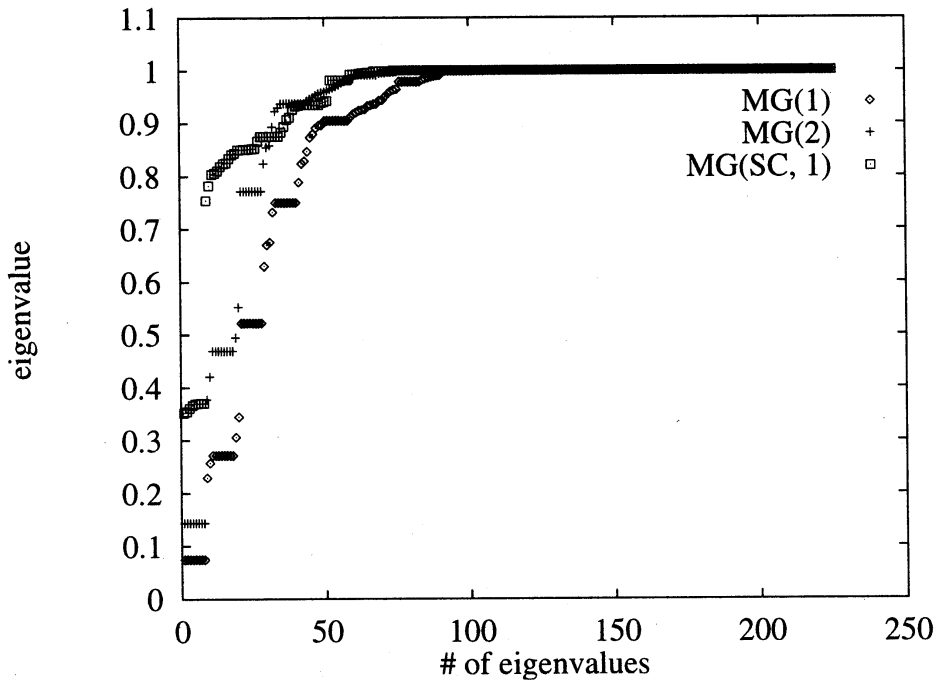


図 2. $\varepsilon = .000001$ の時の MG 前処理後の行列の固有値分布. 問題の大きさは 16^2 である. 括弧内は緩和法の反復回数で, SC は semicoarsening を表している.

行ったもののほぼ中間となっている. さらに $\beta = 0$ では semicoarsening はかえって MGCG 法の収束率を悪くしている. 固有値分布を調べて見ると, 確かに固有値分布は 1 反復と 2 反復行なったもの間にあるが, 固有値がより密集しなくなっていた. 感覚的には, 残差の長波長成分が少ない方向に coarsening をしているためといえるが, これでは緩和法を 1 反復したものより収束率が悪くなることの説明ができない. またこの実験では (A) の離散化ステンシルを用いているが, その他のステンシルを用いても semicoarsening の効果は同様であった.

したがって, 異方性の分かっている場合は, その方向に semicoarsening を行うことにより収束性を改善することが出来ることが分かった.

4 line relaxation

この節では緩和法に line relaxation を用いた MGCG 法についての考察を行なう. 基本的に line relaxation は一つの行の更新を一度に行う緩和法である. 従って並列性はたかだか行の数, あるいはその定数倍しかない. line relaxation にも様々な種類があるが, ここで扱うのは対称 zebra relaxation, alternating direction 対称 zebra relaxation である. 対称 zebra relaxation は格子点のそれぞれの行に対し白, 黒と順番に色をつけていき, 白の行, 黒の行, 白の行の格子点をそれぞれ順番に更新していく緩和法である. 元の行列が 5 重対角行列であればこの行の更新では 3 重対角行列を解く必要がある. alternating 対称 zebra relaxation はこの対称 zebra relaxation を水平方向, 垂直方向に交互に行なうもので, 水平方向の白の行を HW, 黒の行を HB, 垂直方向の白の列を VW, 黒の列を VB とした時 (図 3), HW, HB, VW, VB, VW, HB, HW の順番に更新を行なう. このように更新をすればこの解法は対称になる.

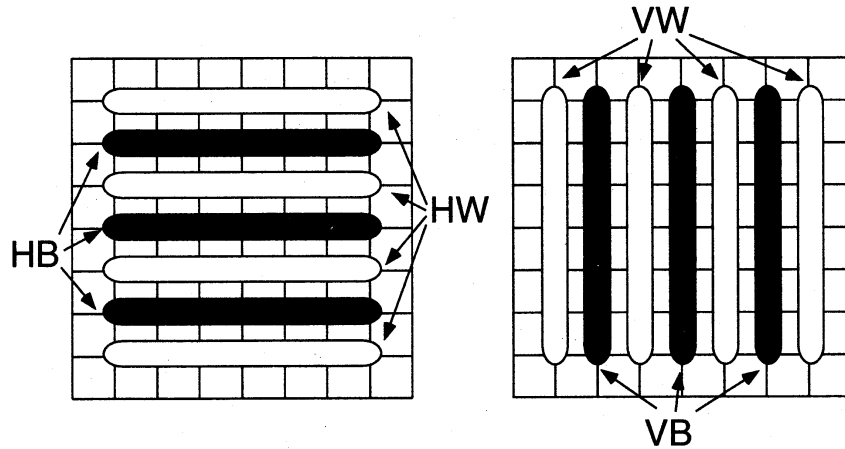


図 3. line relaxation

4.1 収束率

式 (2) に対し $\varepsilon = .000001$ とし, β を変化させた時のそれぞれの解法の平均収束率, 反復回数, 収束までの計算時間を表 2 に示す. 全体の収束率は u_{xy} の離散化方法によって随分収束性が異なるので, (A), (B), (C) の三つの離散化ステンシルのそれぞれを実験の対象としている. いずれも問題のサイズは 128^2 である. この計算時間をはかるにあたり HP9000/720 を用いた.

表 2 で RB は Red-Black 対称 Gauss-Seidel 法, SZ は対称 zebra relaxation, SAZ は alternating 対称 zebra relaxation をそれぞれ緩和法として使った MGCG 法を表し, IC は ICCG 法を表している. RB, IC は比較のために載せてある. まず $\beta = 0^\circ, 90^\circ$ の場合はいずれの離散化ステンシルにもよらないので, 結果はすべて同じとなり, RB を使った MGCG 法は非常に収束が遅いのに対し, line relaxation の line と係数の大きな方向が等しい場合には MGCG 法は非常に速く収束しているのが分かる. つまり SZ であれば β が 90° で非常に速く収束し, また alternating にすることにより $\beta = 0^\circ$ での収束率も改善されている. ICCG 法は一つしかグリッドを使わない分, 一反復にかかる時間が少なく, 収束率も良いため収束までの計算時間が短くなっている.

(A) と (C) の離散化ステンシルでそれぞれの方法を比べると, この場合は RB を緩和法とする MGCG 法の平均収束率が悪くないため, 余り改善されたように見えないが, それでも収束率は line と係数の大きな方向が近い場合に改善され, 大体反復回数が半分になっている. この実験では, 逐次計算機で計算しているため, RB と SZ では一反復にかかる計算時間がほとんど変わらず, その分収束までの計算時間が短くなっている.

問題は (B) の離散化ステンシルを使った場合である. 今は $\varepsilon \ll 1$ としているため, このステンシルでは係数行列の対角要素の絶対値が小さくなってしまふ. $\beta = 20^\circ, 70^\circ$ の場合は ICCG 法を除き他の離散化ステンシルと変わらないが, $\beta = 45^\circ$ では MGCG 法の非常に収束率が悪くなっている. この時の係数行列はステンシルで表すと,

$$\frac{1}{h^2} \begin{bmatrix} -0.5 & -0.0 & 0.0 \\ -0.0 & 1.0 & -0.0 \\ 0.0 & -0.0 & -0.5 \end{bmatrix} \quad (5)$$

となり, ほとんど RB, SZ, SAZ とともに緩和法としては余りにも貧弱なものとなるからである. (これは行列ではなくてステンシルであることに注意.) しかしながら, IC 前処理は係数が -0.5 である要素も使うため, 非常に良い

表 2. MGCG 法の収束率

		β	0	20	45	70	90
(A)	RB	rate	0.878	0.571	0.517	0.571	0.878
		# iter.	183	42	36	42	183
		time	108.0	25.15	21.58	25.15	108.0
	SZ	rate	0.797	0.552	0.410	0.283	2.96e-11
		# iter.	103	39	26	19	1
		time	63.86	24.47	16.39	12.14	0.89
	SAZ	rate	4.79e-11	0.261	0.311	0.241	2.95e-11
		# iter.	1	18	21	17	1
		time	1.68	23.16	27.18	21.9	1.68
	IC	rate	5.65e-07	0.744	0.868	0.909	0.00107
		# iter.	2	78	163	243	4
		time	0.50	12.08	25.1	37.11	0.80
(B)	RB	rate	0.878	0.676	0.950	0.676	0.878
		# iter.	183	59	447	59	183
		time	108.0	34.88	263.07	34.88	108.0
	SZ	rate	0.797	0.650	0.913	0.419	2.96e-11
		# iter.	103	54	254	27	1
		time	63.86	33.53	157	16.95	0.89
	SAZ	rate	4.79e-11	0.414	0.885	0.378	2.96e-11
		# iter.	1	27	189	24	1
		time	1.68	34.56	239.52	30.86	1.68
	IC	rate	5.65e-07	0.488	0.00352	0.930	0.00107
		# iter.	2	34	5	319	4
		time	0.50	5.30	0.98	47.78	0.80
(C)	RB	rate	0.878	0.607	0.572	0.607	0.878
		# iter.	183	47	42	47	183
		time	108.0	28.10	25.09	28.10	108.0
	SZ	rate	0.797	0.585	0.472	0.316	2.96e-11
		# iter.	103	44	31	20	1
		time	63.86	27.98	19.66	12.66	0.89
	SAZ	rate	4.79e-11	0.320	0.384	0.287	2.95e-11
		# iter.	1	21	25	19	1
		time	1.68	27.18	32.21	24.52	1.68
	IC	rate	5.65e-07	0.704	0.872	0.919	0.00107
		# iter.	2	66	169	275	4
		time	0.50	10.31	26.43	42.36	0.80

前処理となっている。

また ICCG 法は全ての離散化ステンシルにおいて $45^\circ < \beta < 90^\circ$ で収束性が極めて悪くなっている。

4.2 固有値分布

line relaxation の MGCG 法に対する効果を見るために、MG 前処理後の係数行列の固有値分布を調べた。まず $\beta = 70^\circ$ の場合に RB を使った MG 前処理と SZ を使ったものを比べたのが図 4 である。問題の大きさは 16^2 である。この図を見ると、RB を使った場合はほとんど固有値分布が改善されていないのに対し、SZ を使うと固有値は密集し、この時条件数は 1.10 である。

$\beta = 20^\circ$ の場合は、前節の数値実験によると、SZ を緩和法とする MGCG 法の収束率は余り改善されていない。図 4 は SZ を使った MG 前処理と SAZ を使ったものを比べたものである。この図を見ると、SZ を使った場合はほとんど固有値分布が改善されていないのに対し、SAZ を使うことにより固有値は密集する。この時条件数は 1.06 である。この固有値解析でも分かるが、異方性の大きな方向に line relaxation をしないと line relaxation の効果はほとんどない。

5 結論

異方性拡散方程式に対し semicoarsening, line relaxation を行なう MGCG 法の検討を行なった。そして両手法とも、異方性の強い方向に用いることにより MG 法と同様に MGCG 法の収束率が改善することが分かった。また異方性がメッシュの方向と異なる場合には、各解法の収束率は u_{xy} の離散化ステンシルに大きく左右されるため注意が必要である。その項の符合により (A) または (B) を選ぶか、あるいはいずれの場合でも (C) を選択するのが良い。

異方性の分からない場合などでは、緩和法に alternating 対称 zebra relaxation を使うことにより、MGCG 法は robust な解法となることが分かった。しかしながらこの緩和法は余り並列性が高くないため、分散メモリ型並列計算機へ実装した場合、どれくらい実行性能が出るかが問題である。

また数値実験の結果、line relaxation を用いた MGCG 法は ICCG 法と比べてほぼ同等の計算時間で収束することが分かった。しかし、これは逐次計算機の結果で、line relaxation の方が IC 前処理よりまだ並列性が高いことを考えると、並列計算機上では line relaxation を用いた MGCG 法の方が有利だと考えられる。また ICCG 法は $45^\circ < \beta < 90^\circ$ ではいずれの離散化ステンシルを用いても収束性が極めて悪くなる。従ってこの部分では対称 zebra relaxation を緩和法とする MGCG 法を使う方が良い。

参考文献

- [1] Dendy Jr., J. E., "Black box multigrid," *J. Comp. Phys.*, vol. 48, pp. 366-386, 1982.
- [2] Mulder, W. A., "A New Multigrid Approach to Convection Problems," *J. Comp. Phys.*, vol. 83, pp. 303-323, 1989.
- [3] Naik, N. H. and J. V. Rosendale, "The Improved Robustness of Multigrid Elliptic Solvers Based on Multiple Semicoarsened Grids," *SIAM J. Numer. Anal.*, vol. 30, no. 1, pp. 215-229, February 1993.
- [4] Tatebe, O., "The Multigrid Preconditioned Conjugate Gradient Method," in *Proceedings of Sixth Copper Mountain Conference on Multigrid Methods*, pp. 621-634, NASA Conference Publication 3224, April 1993.
- [5] Tatebe, O. and Y. Oyanagi, "Efficient Implementation of the Multigrid Preconditioned Conjugate Gradient Method on Distributed Memory Machines," in *Proceedings of Supercomputing '94*, pp. 194-203, IEEE Computer Society, November 1994.

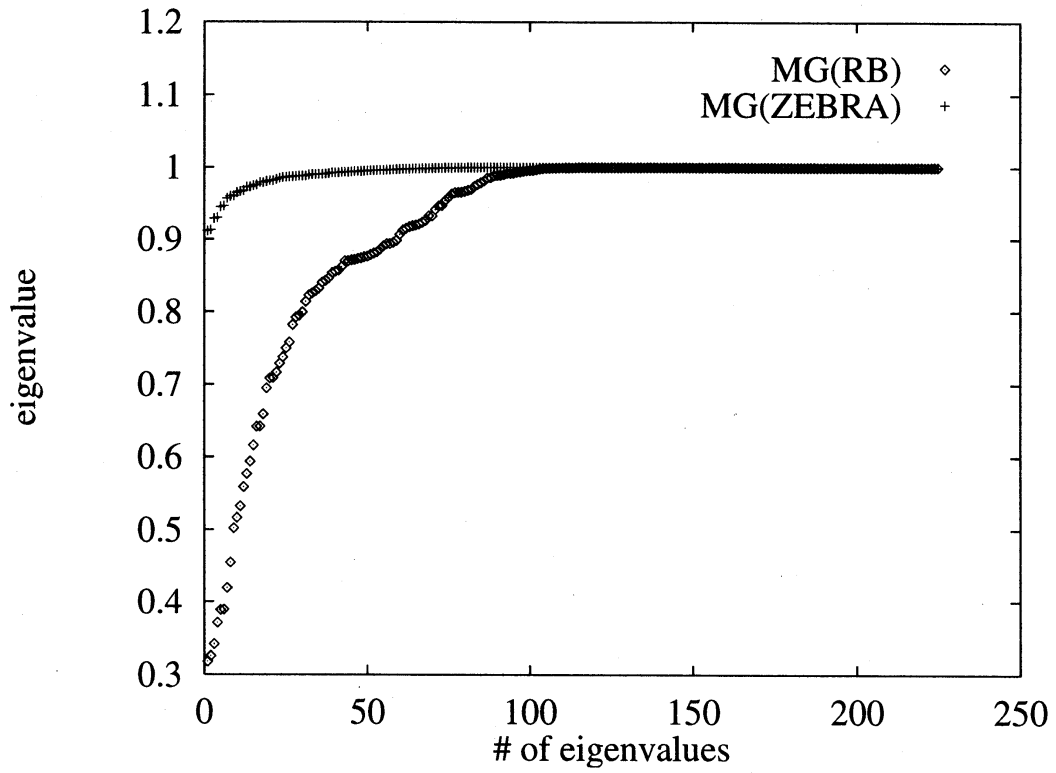


图 4. $\beta = 70^\circ$

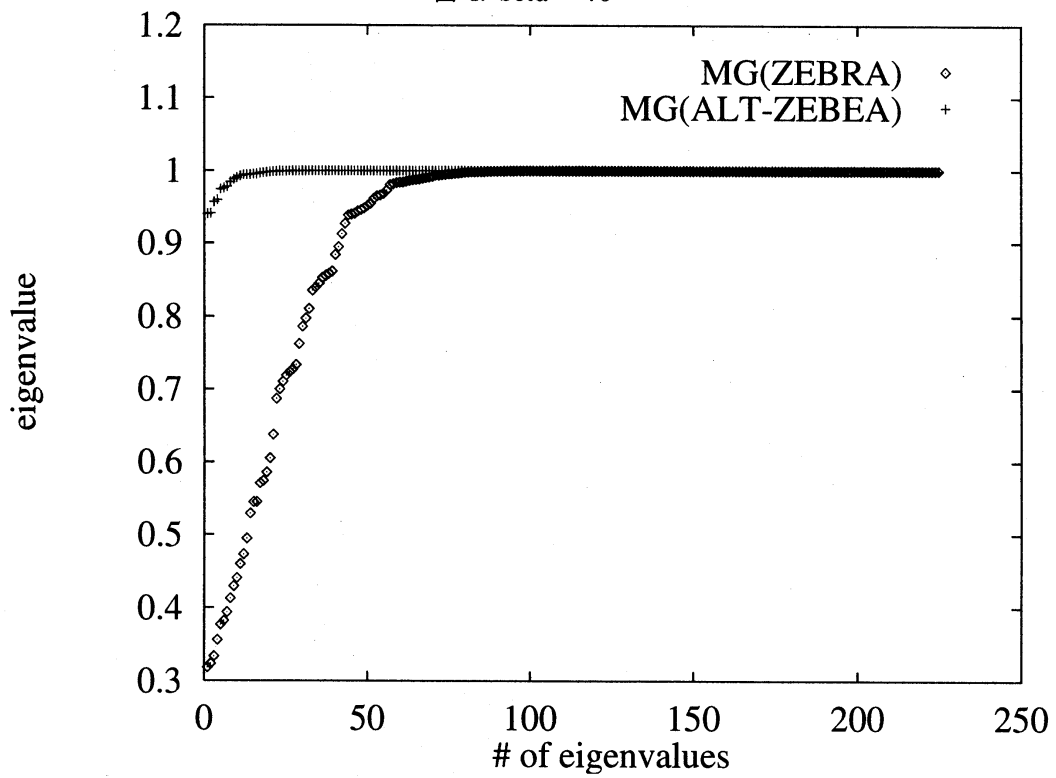


图 5. $\beta = 20^\circ$