

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA APLICADA

**Algoritmos Adaptativos para o
método GMRES(m)**

por

Tífani Teixeira Gonçalves

Dissertação submetida como requisito parcial
para a obtenção do grau de
Mestre em Matemática Aplicada

Prof. Dr. Rudnei Dias da Cunha
Orientador

Porto Alegre, Janeiro de 2005.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Teixeira Gonzalez, Tífani

Algoritmos Adaptativos para o método GMRES(m) / Tífani
Teixeira Gonzalez.—Porto Alegre: PPGMAp da UFRGS, 2005.

171 p.: il.

Dissertação (mestrado) —Universidade Federal do Rio Grande
do Sul, Programa de Pós-Graduação em Matemática Aplicada,
Porto Alegre, 2005.

Orientador: da Cunha, Rudnei Dias

Dissertação: Matemática Aplicada

Análise numérica, Métodos iterativos, GMRES

Algoritmos Adaptativos para o método GMRES(m)

por

Tífani Teixeira Gonçalves

Dissertação submetida ao Programa de Pós-Graduação em Matemática Aplicada do Instituto de Matemática da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de

Mestre em Matemática Aplicada

Linha de Pesquisa: Algoritmos Numéricos e Algébricos

Orientador: Prof. Dr. Rudnei Dias da Cunha

Banca examinadora:

Prof. Dr.
Dalcídio Moraes Cláudio

Profa. Dra.
Liliane Basso Barrichello

Prof. Dr.
Paulo Ricardo de Ávila Zingano

Dissertação apresentada e aprovada em
10 de Janeiro de 2005.

Profa. Dra. Maria Cristina Varriale
Coordenador

*À minha mãe Sandra, a quem
amo mais do que tudo.*

AGRADECIMENTO

Gostaria de agradecer aos meus pais, Sandra e Learás, pelo apoio financeiro, afetivo, pelos puxões de orelha, pelas lágrimas compreendidas e pelas palavras de apoio que sempre dispensaram a mim.

Gostaria também de agradecer à família Soares por todo o carinho e atenção, em especial gostaria de deixar registrada a minha maior gratidão aos melhores amigos que alguém pode ter: Rita Soares e Edilon Trisch, os quais acompanharam toda essa etapa da minha vida: vibrando e respeitando cada lágrima.

Por fim, gostaria de agradecer ao meu orientador, o professor Rudnei, por sua excelente orientação, paciência, incentivo, e principalmente por sua amizade.

Conteúdo

LISTA DE FIGURAS	9
LISTA DE TABELAS	14
LISTA DE ABREVIATURAS	16
RESUMO	17
ABSTRACT	18
1 INTRODUÇÃO	19
2 DESCRIÇÃO DOS MÉTODOS ITERATIVOS	22
2.1 O Método do Subespaço de Krylov	22
2.2 O Método de Arnoldi	23
2.3 O Método da Ortogonalização Completa - FOM	25
2.4 Método do Resíduo Mínimo Generalizado - GMRES	27
2.5 Simpler GMRES - SGMRES	34
2.5.1 Loose GMRES - LGMRES(m, k)	38
2.6 Diagrama dos Métodos	40
3 ELABORAÇÃO DOS CRITÉRIOS	42
3.1 GMRES-ADAPTATIVO	42
3.2 BC-GMRES	45
3.3 Critério 1	48
3.4 Critério 2	50

3.5 Critério 3	53
3.6 Critério 4	57
4 RESULTADOS EXPERIMENTAIS	63
4.1 Experimentos Realizados	64
4.1.1 ADD32	64
4.1.2 ARC_130	70
4.1.3 CAVITY05	76
4.1.4 CDDE1	81
4.1.5 E05R0000	86
4.1.6 E05R0300	91
4.1.7 FIDAPM05	96
4.1.8 FS_183_1	101
4.1.9 FS_760_1	106
4.1.10 GRE_115	112
4.1.11 IMPCOL_A	117
4.1.12 JPWH991	122
4.1.13 LNS_131	128
4.1.14 NOS3	133
4.1.15 ORSIRR1	138
4.1.16 PDE900	143
4.1.17 SAYLR3	148
4.1.18 SHERMAN1	153

4.1.19 STEAM1	158
5 CONCLUSÕES	164
BIBLIOGRAFIA	166
APÊNDICE A CONVERGÊNCIA DO MÉTODO GMRES	169

Lista de Figuras

Figura 2.1	a)Autovalores da matriz, b) “Zoom” dos Autovalores da matriz	31
Figura 2.2	Diagrama dos Métodos	41
Figura 3.1	IMPCOL_A: Distribuição de zeros do polinômio residual (◦: zeros ideais; ×: zeros aproximados). Extraído de [Nodera e Moriya, 2003].	46
Figura 3.2	GMRES Adaptativo: a) $m = 2$ e b) $m = 10$	49
Figura 3.3	DIGSE_1: a) $m = 2$ e b) $m = 10$	51
Figura 3.4	<i>DIGSE_2</i> : a) $m = 2$ e b) $m = 10$	52
Figura 3.5	Logaritmo_1: a) $m = 2$ e b) $m = 10$	54
Figura 3.6	Logaritmo_2: a) $m = 2$ e b) $m = 10$	55
Figura 3.7	Logaritmo_3: a) $m = 2$ e b) $m = 10$	56
Figura 3.8	Adaptativo-1 a) $m = 2$ e b) $m = 10$	59
Figura 3.9	Adaptativo a) $m = 2$ e b) $m = 10$	61
Figura 4.1	ADD32: a)Estrutura da matriz, b)Autovalores da matriz	64
Figura 4.2	ADD32: Gráfico do resíduo e m versus iterações	66
Figura 4.3	ADD32: Gráfico do resíduo e m versus iterações	67
Figura 4.4	ADD32: Gráfico do resíduo e m versus iterações	68
Figura 4.5	ARC_130: a)Estrutura da matriz, b)Autovalores da matriz	70
Figura 4.6	ARC_130: Gráfico do resíduo e m versus iterações	72
Figura 4.7	ARC_130: Gráfico do resíduo e m versus iterações	73
Figura 4.8	ARC_130: Gráfico do resíduo e m versus iterações	74
Figura 4.9	CAVITY05: a)Estrutura da matriz, b)Autovalores da matriz	76

Figura 4.10	CAVITY05: Gráfico do resíduo e m versus iterações	78
Figura 4.11	CAVITY05: Gráfico do resíduo e m versus iterações	79
Figura 4.12	CAVITY05: Gráfico do resíduo e m versus iterações	80
Figura 4.13	CDDE1: a)Estrutura da matriz, b)Autovalores da matriz	81
Figura 4.14	CDDE1: Gráfico do resíduo e m versus iterações	83
Figura 4.15	CDDE1: Gráfico do resíduo e m versus iterações	84
Figura 4.16	CDDE1: Gráfico do resíduo e m versus iterações	85
Figura 4.17	E05R0000: a)Estrutura da matriz, b)Autovalores da matriz	86
Figura 4.18	E05R0000: Gráfico do resíduo e m versus iterações	88
Figura 4.19	E05R0000: Gráfico do resíduo e m versus iterações	89
Figura 4.20	E05R0000: Gráfico do resíduo e m versus iterações	90
Figura 4.21	E05R0300: a)Estrutura da matriz, b)Autovalores da matriz	91
Figura 4.22	E05R0300: Gráfico do resíduo e m versus iterações	93
Figura 4.23	E05R0300: Gráfico do resíduo e m versus iterações	94
Figura 4.24	E05R0300: Gráfico do resíduo e m versus iterações	95
Figura 4.25	FIDAPM05: a)Estrutura da matriz, b)Autovalores da matriz	96
Figura 4.26	FIDAPM05: Gráfico do resíduo e m versus iterações	98
Figura 4.27	FIDAPM05: Gráfico do resíduo e m versus iterações	99
Figura 4.28	FIDAPM05: Gráfico do resíduo e m versus iterações	100
Figura 4.29	FS_183.1: a)Estrutura da matriz, b)Autovalores da matriz	101
Figura 4.30	FS_183.1: Gráfico do resíduo e m versus iterações	103
Figura 4.31	FS_183.1: Gráfico do resíduo e m versus iterações	104

Figura 4.32 FS_183_1: Gráfico do resíduo e m versus iterações	105
Figura 4.33 FS_760_1: a)Estrutura da matriz, b)Autovalores da matriz	106
Figura 4.34 FS_760_1: Gráfico do resíduo e m versus iterações	108
Figura 4.35 FS_760_1: Gráfico do resíduo e m versus iterações	109
Figura 4.36 FS_760_1: Gráfico do resíduo e m versus iterações	110
Figura 4.37 GRE_115: a)Estrutura da matriz, b)Autovalores da matriz	112
Figura 4.38 GRE_115: Gráfico do resíduo e m versus iterações	114
Figura 4.39 GRE_115: Gráfico do resíduo e m versus iterações	115
Figura 4.40 GRE_115: Gráfico do resíduo e m versus iterações	116
Figura 4.41 IMPCOL_A: a)Estrutura da matriz, b)Autovalores da matriz	117
Figura 4.42 IMPCOL_A: Gráfico do resíduo e m versus iterações	119
Figura 4.43 IMPCOL_A: Gráfico do resíduo e m versus iterações	120
Figura 4.44 IMPCOL_A: Gráfico do resíduo e m versus iterações	121
Figura 4.45 JPWH991: a)Estrutura da matriz, b)Autovalores da matriz	122
Figura 4.46 JPWH991: Gráfico do resíduo e m versus iterações	124
Figura 4.47 JPWH991: Gráfico do resíduo e m versus iterações	125
Figura 4.48 JPWH991: Gráfico do resíduo e m versus iterações	126
Figura 4.49 LNS_131: a)Estrutura da matriz, b)Autovalores da matriz	128
Figura 4.50 LNS_131: Gráfico do resíduo e m versus iterações	130
Figura 4.51 LNS_131: Gráfico do resíduo e m versus iterações	131
Figura 4.52 LNS_131: Gráfico do resíduo e m versus iterações	132
Figura 4.53 NOS3: a)Estrutura da matriz, b)Autovalores da matriz	133

Figura 4.54 NOS3: Gráfico do resíduo e m versus iterações	135
Figura 4.55 NOS3: Gráfico do resíduo e m versus iterações	136
Figura 4.56 NOS3: Gráfico do resíduo e m versus iterações	137
Figura 4.57 ORSIRR1: a)Estrutura da matriz, b)Autovalores da matriz	138
Figura 4.58 ORSIRR1: Gráfico do resíduo e m versus iterações	140
Figura 4.59 ORSIRR1: Gráfico do resíduo e m versus iterações	141
Figura 4.60 ORSIRR1: Gráfico do resíduo e m versus iterações	142
Figura 4.61 PDE900: a)Estrutura da matriz, b)Autovalores da matriz	143
Figura 4.62 PDE900: Gráfico do resíduo e m versus iterações	145
Figura 4.63 PDE900: Gráfico do resíduo e m versus iterações	146
Figura 4.64 PDE900: Gráfico do resíduo e m versus iterações	147
Figura 4.65 SAYLR3: a)Estrutura da matriz, b)Autovalores da matriz	148
Figura 4.66 SAYLR3: Gráfico do resíduo e m versus iterações	150
Figura 4.67 SAYLR3: Gráfico do resíduo e m versus iterações	151
Figura 4.68 SAYLR3: Gráfico do resíduo e m versus iterações	152
Figura 4.69 SHERMAN1: a)Estrutura da matriz, b)Autovalores da matriz	153
Figura 4.70 SHERMAN1: Gráfico do resíduo e m versus iterações	155
Figura 4.71 SHERMAN1: Gráfico do resíduo e m versus iterações	156
Figura 4.72 SHERMAN1: Gráfico do resíduo e m versus iterações	157
Figura 4.73 STEAM1: a)Estrutura da matriz, b)Autovalores da matriz	158
Figura 4.74 STEAM1: Gráfico do resíduo e m versus iterações	160
Figura 4.75 STEAM1: Gráfico do resíduo e m versus iterações	161

Figura 4.76 STEAM1: Gráfico do resíduo e m versus iterações	162
Figura A.1 Autovalores de A pertencentes a reta	171
Figura A.2 Autovalores de A próximos a reta	171

Lista de Tabelas

Tabela 2.1	Custo/Armazenamento das Ortogonalizações	25
Tabela 2.2	GMRES e LGMRES	40
Tabela 3.1	Critério 1	50
Tabela 3.2	Critério 2a	52
Tabela 3.3	Critério 2b	52
Tabela 3.4	Conclusões finais quanto ao critério 2	53
Tabela 3.5	Critério 3a	55
Tabela 3.6	Critério 3b	56
Tabela 3.7	Critério 3c	57
Tabela 3.8	Conclusões finais quanto ao critério 3	57
Tabela 3.9	Critério 4a	60
Tabela 3.10	Critério 4b	61
Tabela 3.11	Conclusões finais quanto ao critério 4	61
Tabela 4.1	ADD32	65
Tabela 4.2	ARC_130	71
Tabela 4.3	CAVITY_05	77
Tabela 4.4	CDDE1	82
Tabela 4.5	E05R0000	87
Tabela 4.6	E05R0300	92
Tabela 4.7	FIDAPM05	97
Tabela 4.8	FS_183.1	102

Tabela 4.9	FS_760.1	107
Tabela 4.10	GRE_115	113
Tabela 4.11	IMPCOLA	118
Tabela 4.12	JPWH_991	123
Tabela 4.13	LNS_131	129
Tabela 4.14	NOS3	134
Tabela 4.15	ORSIRR_1	139
Tabela 4.16	PDE_900	144
Tabela 4.17	SAYLR_03	149
Tabela 4.18	SHERMAN_1	154
Tabela 4.19	STEAM_1	159

LISTA DE ABREVIATURAS

GMRES	Generalized Minimum Residual
GMRES(m)	Generalized Minimum Residual reinicializado, com base de dimensão m
GMRES-A(m)	Generalized Minimum Residual reinicializado adaptativo, com base de dimensão m
GMRESH(m)	Generalized Minimum Residual com a transformação de Householder
GMRESH-A(m)	Generalized Minimum Residual adaptativo com a transformação de Householder
SGMRES(m)	Simpler GMRES
SGMRESH(m)	Simpler GMRES com a transformação de Householder
SGMRES-A(m)	Simpler GMRES adaptativo
SGMRESH-A(m)	Simpler GMRES adaptativo, com a transformação de Householder
LGMRES(m, k)	Loose GMRES
LGMRES-A(m, k)	Loose GMRES adaptativo
BC-GMRES(m)	Bi-Condition GMRES
BC-GMRES-A(m)	Bi-Condition GMRES adaptativo

RESUMO

Nesse trabalho apresentamos algoritmos adaptativos do Método do Resíduo Mínimo Generalizado (GMRES) [Saad e Schultz, 1986], um método iterativo para resolver sistemas de equações lineares com matrizes não simétricas e esparsas, o qual baseia-se nos métodos de projeção ortogonal sobre um subespaço de Krylov.

O GMRES apresenta uma versão *reinicializada*, denotada por $\text{GMRES}(m)$, também proposta por [Saad e Schultz, 1986], com o intuito de permitir a utilização do método para resolver grandes sistemas de n equações, sendo n a dimensão da matriz dos coeficientes do sistema, já que a versão não-reinicializada (“Full-GMRES”) apresenta um gasto de memória proporcional a n^2 e de número de operações de ponto-flutuante proporcional a n^3 , no pior caso. No entanto, escolher um valor apropriado para m é difícil, sendo m a dimensão da base do subespaço de Krylov, visto que dependendo do valor do m podemos obter a estagnação ou uma rápida convergência.

Dessa forma, nesse trabalho, acrescentamos ao $\text{GMRES}(m)$ e algumas de suas variantes um critério que tem por objetivo escolher, adequadamente, a dimensão, m da base do subespaço de Krylov para o problema o qual deseja-se resolver, visando assim uma mais rápida, e possível, convergência.

Aproximadamente duas centenas de experimentos foram realizados utilizando as matrizes da Coleção Harwell-Boeing [MCSD/ITL/NIST, 2003], que foram utilizados para mostrar o comportamento dos algoritmos adaptativos. Foram obtidos resultados muito bons; isso poderá ser constatado através da análise das tabelas e também da observação dos gráficos expostos ao longo desse trabalho.

ABSTRACT

In this work we present adaptive algorithms for the Generalized Minimum Residual method (GMRES) [Saad and Schultz, 1986], an iterative method for solving sparse, nonsymmetric systems of linear equations, based on orthogonal projection over Krylov subspaces methods.

GMRES has a *restarted* version, GMRES(m), also proposed by [Saad and Schultz, 1986], which allows its use when the system has a large number n of equations, as the non-restarted version (“Full-GMRES”) has a memory cost proportional to n^2 and a floating-point cost proportional to n^3 , in the worst case. However choosing the appropriate value of m is still a challenge.

Therefore, in this work we added to GMRES(m) and some of its variants a criterion which aims to choose the dimension m of the Krylov subspace basis adequately, for a given system, in order to obtain convergence and, if possible, in the most fastest way.

Around two hundred experiments were carried out using matrices of the Harwell-Boeing Collection [MCSD/ITL/NIST, 2003], to show the behaviour of the adaptive algorithms. Good results have been obtained; this may be ascertained from the data presented in the tables and the graphs contained in this work.

1 INTRODUÇÃO

Nessa dissertação são estudados aprimoramentos ao método iterativo “Generalized Minimum Residual” (GMRES), proposto em [Saad e Schultz, 1986], e algumas de suas variantes, utilizados na solução de sistemas de equações lineares na forma

$$Ax = b \tag{1.1}$$

onde $A \in \mathbb{R}^{n \times n}$ é não-simétrica e esparsa e $x, b \in \mathbb{R}^n$.

No GMRES, conforme proposto originalmente por Saad, em 1986, o resíduo inicial $r_0 = b - Ax_0$ é utilizado para gerar uma base ortonormal para um subespaço de Krylov,

$$\mathcal{K}_k(A, r_0) = \text{subespaço}(r_0, Ar_0, \dots, A^{k-1}r_0) \tag{1.2}$$

onde k é o número de iterações do método GMRES, através do processo de Arnoldi, o qual é uma modificação do processo de ortonormalização modificado de Gram-Schmidt (MGS). A solução do sistema (1.1) é tomada como solução de um problema de mínimos-quadrados que envolvem uma matriz e um vetor gerados durante o processo de Arnoldi, cujos detalhes serão apresentados no Capítulo 2.

Dentre algumas das características apresentadas pelo GMRES, salienta-se a convergência em, no máximo, n iterações. Obviamente, se considerarmos matrizes em que n é grande, no pior caso (convergência na n -ésima iteração), essa garantia de convergência não será vantagem, pois o processo de Arnoldi requererá o armazenamento de n vetores de dimensão n , além de uma matriz de dimensão $(n + 1) \times n$ – o que levará a uma complexidade espacial $O(n^2)$ – e exigirá n produtos matriz-vetor, com uma complexidade temporal $O(n^3)$. Isso poderá fazer com que não seja possível obter a solução do sistema num determinado computador, pois a memória disponível não será suficiente; ou a resposta não será obtida em tempo hábil.

Dessa forma, já em [Saad e Schultz, 1986], Saad e Schultz propuseram uma versão chamada de *reinicializada*, GMRES(m), onde m é a dimensão da base do subespaço de Krylov, na qual se produz uma base ortonormal para um subespaço de Krylov de dimensão m , $\mathcal{K}_m(A, r_k)$; isto é, a cada iteração k , calcula-se o resíduo $r_k = b - Ax_k$, calculam-se os m vetores ortonormais através do processo de Arnoldi e, a partir deles,

resolve-se um problema de mínimos-quadrados, obtendo-se uma aproximação x_{k+1} para a solução do sistema (1.1). Deve-se dar certa atenção se o resíduo r_{k+1} não for suficientemente pequeno, faz-se uma nova iteração. O método GMRES(m) será apresentado com maior detalhe no Capítulo 2, mas pode-se salientar desde já que o processo de Arnoldi apresenta, nesse caso, uma complexidade espacial $O(m^2)$ e temporal $O(mn)$, o que torna-o vantajoso.

Infelizmente, ao se optar por usar o método GMRES(m), perde-se a garantia de convergência apresentada pelo método GMRES, a menos que m seja escolhido convenientemente. Se ele for muito pequeno, a convergência pode não ser obtida, nem mesmo em n iterações; se for muito grande, a convergência poderá ser obtida, porém com um gasto excessivo de operações aritméticas de ponto-flutuante e de memória. Em [Saad e Schultz, 1986], há um teorema que apresenta uma fórmula para a escolha de m de acordo com a distribuição de autovalores de A . Infelizmente, tal teorema é aplicável a apenas um pequeno número de matrizes; no Capítulo 2, apresentamos um exemplo, analisando o m apresentado através do uso do teorema e do GMRES propriamente dito.

Dessa forma, nessa dissertação iremos estudar maneiras de determinar dinamicamente, ao longo do processo de solução do sistema (1.1), um valor adequado para m , de forma a garantir a convergência do método GMRES(m) e de algumas de suas variantes, bem como obter a solução com o menor gasto computacional possível, em termos do número de operações de ponto-flutuante.

Esse trabalho é eminentemente experimental, de forma a se evidenciar a validade, ou não, dos diferentes critérios de determinação de m , estudados e propostos. Os experimentos realizados consistiram na solução de sistemas de equações lineares cujas matrizes encontram-se disponíveis no sítio Internet “Matrix Market” [MCSD/ITL/NIST, 2003].

Os experimentos foram realizados utilizando-se o método GMRES (m); uma variante do GMRES(m) utilizando transformações de Householder, ao invés do processo de Arnoldi, proposta por Walker [Walker, 1988], a qual chamaremos de GMRESH(m); o “Simpler GMRES” e sua variante usando transformações de Householder [Walker e Zhou, 1994], os quais chamaremos de SGMRES(m) e SGMRESH(m); o “Loose-

GMRES”, proposto por [Baker et al., 2003], chamados de LGMRES(m) e LGMRESH(m); o GMRES adaptativo proposto em [Driver, 1997], chamado de A-GMRES(m); e o BC-GMRES(m), proposto em [Nodera e Moriya, 2003]. Todos esses métodos (com exceção dos dois últimos), por sua vez, serão utilizados em combinação com o critério de determinação de m por nós proposto – nesse caso, adicionaremos a letra “A” ao final, a fim distinguir os métodos.

No capítulo 2, apresentamos uma descrição dos métodos iterativos acima, os quais serão apresentados em ordem cronológica em que foram propostos, a fim de se ter uma visão histórica do desenvolvimento desses métodos. No capítulo 3, os critérios desenvolvidos para a escolha adequada de m serão apresentados, de forma evolutiva, bem como outros dois critérios propostos em [Driver, 1997] e [Nodera e Moriya, 2003]. No capítulo 4, descrevemos as matrizes utilizadas, caracterizando-as quanto ao seu tamanho, estrutura e distribuição de autovalores, bem como os experimentos realizados com as mesmas, discutindo os resultados obtidos. Finalmente, apresentamos, no capítulo 5, as conclusões sobre o trabalho realizado, bem como sugerimos alguns trabalhos futuros.

2 DESCRIÇÃO DOS MÉTODOS ITERATIVOS

Neste capítulo, inicialmente introduzimos alguns conceitos básicos para um bom entendimento dos métodos iterativos que utilizamos nesse trabalho, como o subespaço de Krylov, o Processo de Arnoldi utilizando o processo de Gram-Schmidt e a fatoração de Householder. Em seguida descrevemos os métodos iterativos utilizados no desenvolvimento dos testes realizados com os critérios para a escolha da dimensão do subespaço de Krylov, m , a saber: GMRES(m), GMRESH(m), SGMRES(m), SGMRESH(m) e LGMRES(m, k).

2.1 O Método do Subespaço de Krylov

O método do subespaço de Krylov é um método de projeção que, para resolver sistemas lineares na forma (1.1) procura uma solução aproximada x_k a partir de um subespaço $x_0 + \kappa$, de dimensão k , impondo a condição de Petrov-Galerkin,

$$b - Ax_k \perp \mathcal{L}_k \quad (2.1)$$

onde \mathcal{L}_k é um subespaço de dimensão k e x_0 é uma solução inicial aproximada. Existem diferentes métodos de projeção, que surgem através da escolha do subespaço \mathcal{L}_k e pela forma em que o sistema é pré-condicionado; veja [Saad, 2003, pp. 144-279]. Um desses métodos é o método do subespaço de Krylov, tal que κ_k é o subespaço de Krylov

$$\kappa_k(A, r_0) = \text{subespaço}(r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0) \quad (2.2)$$

onde $r_0 = b - Ax_0$. As aproximações obtidas a partir do método do subespaço de Krylov são da forma

$$A^{-1}b \approx x_k = x_0 + q_{k-1}(A)r_0 \quad (2.3)$$

onde q_{k-1} é um polinômio de grau $k - 1$. No caso em que $x_0 = 0$, então

$$A^{-1}b \approx q_{k-1}(A)b \quad (2.4)$$

ou seja, $A_{k-1}b$ é aproximado por $q_{k-1}(A)b$. Assim, o método do subespaço de Krylov é um subespaço na forma

$$\kappa_k(A, v) = \text{subespaço}(v, Av, A^2v, \dots, A^{k-1}v) \quad (2.5)$$

que denotaremos apenas por κ_k se não houver ambiguidade. Mais detalhes sobre o subespaço de Krylov podem ser encontrados em [Saad, 2003, pp. 151-153].

2.2 O Método de Arnoldi

O método de Arnoldi, conforme [Arnoldi, 1951], é um método de projeção ortogonal sobre κ_m para matrizes não Hermitianas. Esse processo foi introduzido em 1951 e consiste em reduzir uma matriz densa em uma matriz de Hessenberg superior. Conforme [Saad, 2003], pelo processo de Arnoldi os autovalores obtidos da matriz de Hessenberg, em menos de n iterações, aproximam os autovalores da matriz original, e esse fato apresenta-se ideal para resolver sistema com grande matrizes esparsas.

O método de Arnoldi baseia-se no método de Gram-Schmidt para calcular uma base l_2 -ortonormal $(v_1, v_2, \dots, v_{k-1}, v_k)$ de κ_k ; mais detalhes podem ser encontrados em [Saad e Schultz, 1986].

O método de Arnoldi pode ser escrito como:

Algoritmo 1: Arnoldi

1. *Início:* Escolher um vetor inicial v_1 com $\|v_1\|_2 = 1$
2. *Iterações:* Para $j = 1, 2, \dots$
3. $h_{i,j} = (Av_j, v_i), i = 1, 2, \dots, j$
4. $\hat{v}_{j+1} = Av_j - \sum_{i=1}^j h_{i,j} v_i$
5. $h_{j+1,j} = \|\hat{v}_{j+1}\|_2$
6. $v_{j+1} = \hat{v}_{j+1} / h_{j+1,j}$
7. fim para.

Se V_k é a matriz $n \times k$ cujas colunas formam a base l_2 -ortonormal (v_1, \dots, v_k) , então $H_k \equiv V_k^T A V_k$ é a matriz de Hessenberg superior $k \times k$, cujos elementos são os escalares $h_{i,j}$ gerados pelo algoritmo de Arnoldi. Sendo P_k a projeção l_2 -ortogonal de κ_k , e denotamos por A_k a seção de A em κ_k , isto é, o operador $A_k = P_k A|_{\kappa_k}$, pode-se notar que H_k é a matriz que representa A_k na base $\{v_1, v_2, \dots, v_k\}$. Assim o método original de Arnoldi é um método de Petrov-Galerkin para aproximar os autovalores de A por H_k , [Saad, 2003].

Métodos iterativos do subespaço de Krylov tais como o método de Arnoldi para sistemas lineares também são chamados de métodos da ortogonalização completa, os

quais são bem conhecidos para aproximar a solução de grandes sistemas lineares esparsos não-simétricos.

Na prática é possível melhorar o algoritmo de Arnoldi, que utiliza o método de Gram-Schmidt (GS), através do método de Gram-Schmidt Modificado (MGS) ou da transformação de Householder (HO).

Algoritmo 2: Arnoldi MGS

1. *Início:* Escolha o vetor v_1 com $\|v_1\|_2 = 1$
2. *Iterações:* Para $j = 1, 2, \dots, m$
3. $w_j = Av_j$
4. Para $i = 1, 2, \dots, j$
5. $h_{i,j} = (w_j, v_i)$
6. $w_j = w_j - h_{i,j}v_i$
7. fim para
8. $h_{j+1,j} = \|w_j\|_2$
9. $v_{j+1} = \frac{w_j}{h_{j+1,j}}$
10. fim para

Em aritmética exata pode-se dizer que o algoritmo de Arnoldi é equivalente matematicamente ao algoritmo MGS, contudo o mesmo pode apresentar casos onde no passo da ortogonalização ocorra cancelamento computacionalmente, tornando assim o MGS inadequado. Quando isso surge podemos utilizar uma outra técnica de ortogonalização mais segura que é a transformação de Householder, a qual utiliza matrizes de reflexão na forma $P_k = I - 2w_k w_k^T$, a que transforma uma matriz X em uma outra matriz na forma triangular superior; o vetor w é o chamado vetor de Householder que determina a matriz P_k e $\|w\|_2 = 1$. Observe que os vetores coluna da matriz A no processo de Arnoldi não são avaliados durante todo o decorrer do processo, ou seja, no processo de Arnoldi o próximo vetor é obtido a partir de Av_j , onde v_j é o vetor da base. Já utilizando a transformação de Householder, a coluna ortonormal v_j é obtida como $P_1 P_2 \dots P_j e_j$, onde $P_1 P_2 \dots P_j$ são as matrizes de Householder. Note que na transformação de Householder temos $P_k^{-1} = P_k^T = P_k$. Esse processo foi originalmente proposto por [Walker, 1988].

Algoritmo 3: Arnoldi Householder

1. *Início:* Escolha o vetor não-nulo v . Tome $z_1 = v$
2. *Iterações:* Para $j = 1, 2, \dots, m, m + 1$
3. Calcule o vetor unitário de Householder tal que $w_{j_i} = 0, i = 1, 2, \dots, j - 1$
4. $P_j z_{j_i} = 0, i = j + 1, \dots, n$. onde $P_j = I - 2w_j w_j^T$
5. $h_{j-1} = P_j z_j$
6. $v_j = P_1 P_2 \dots P_j e_j$
7. Se $j \leq m$, calcule $z_{j+1} = P_j P_{j-1} \dots P_1 A v_j$
8. fim para

Observe que a matriz P_j não necessariamente é formada explicitamente.

Em [Saad, 2003, pp. 158] é apresentada uma comparação entre o custo em armazenamento e operações de ponto-flutuante (flops) do processo de Arnoldi com GS, MGS e HO, conforme a tabela (2.1). Observe que o maior custo em flops apresentado pelo uso das transformações de Householder é necessário em função da melhor qualidade numérica obtida com eles.

	GS	MGS	HO
flops	$2m^2n$	$2m^2n$	$4m^2n - \frac{4m^3}{3}$
Armazenamento	$(m + 1)n$	$(m + 1)n$	$(m + 1)n - \frac{m^2}{2}$

Tabela 2.1: Custo/Armazenamento das Ortogonalizações

2.3 O Método da Ortogonalização Completa - FOM

O método da ortogonalização completa, denotado por FOM, surgiu em 1981; veja [Saad e Schultz, 1986], e foi baseado em outros dois métodos: o método do resíduo mínimo [Paige e Saunders, 1975] [Elman, 1982a], aplicável a sistemas simétricos visando minimizar o resíduo na norma-2 e o ORTHODIR [Young e Jea, 1980].

Citamos a seguir o algoritmo do FOM, onde tomamos m como sendo a dimensão do subespaço de Krylov, de acordo com [Saad, 2003]:

Algoritmo 4: FOM

1. *Início:* Calcule $r_0 = b - Ax_0, \beta = \|r_0\|_2$ e $v_1 = \frac{r_0}{\beta}$
2. Defina a matriz $H_m = h_{i,j}, i, j = 1, 2, \dots, m$. Tome $H_m = 0$

3. *Iterações:* Para $j = 1, 2, \dots, m$
4. $w_j = Av_j$
5. Para $i = 1, 2, \dots, j$
6. $h_{ij} = (w_j, v_i)$
7. $w_j = w_j - h_{ij}v_i$
8. fim para
9. $h_{j+1,j} = \|w_j\|_2$. Se $h_{j+1,j} = 0$ tome $m = j$ e vá para (12.)
10. $v_{j+1} = \frac{w_j}{h_{j+1,j}}$
11. fim para
12. $y_m = H_m^{-1}\beta e_1$ e $x_m = x_0 + V_m y_m$

Por [Saad, 2003], podemos concluir que o número m de iterações no passo (2.) do algoritmo FOM é suficiente para aproximar a solução x_m , sendo praticamente a solução exata. É claro que, se o algoritmo pára em m iterações é desnecessário calcular v_{m+1} .

O algoritmo FOM têm importantes propriedades [Saad, 1984]:

1. o vetor residual r_m é apenas o vetor v_{m+1} , assim podemos dizer que os vetores residuais do algoritmo FOM são l_2 -ortogonais a cada outro;
2. o algoritmo FOM não pára se e somente se o grau do polinômio mínimo de v_1 é menor do que m e a matriz H_m é não-singular;
3. o processo termina em, no máximo, n passos.

A dificuldade com o método FOM, segundo [Saad e Schultz, 1986], é que o mesmo apresenta um alto custo computacional à medida que o número m é incrementado; em média, o custo a cada iteração é aproximadamente de $2Nz(A) + 2mn$, onde Nz é o número de elementos não-nulos de A e m é a dimensão do subespaço de Krylov gerado pelo processo de Arnoldi.

2.4 Método do Resíduo Mínimo Generalizado - GMRES

O método do Resíduo Mínimo surgiu em 1986 [Saad e Schultz, 1986], e é matematicamente equivalente aos métodos GCR [Elman, 1982b] [Vinsome, 1976] e o ORTHODIR [Young e Jea, 1980].

De acordo com [Saad e Schultz, 1986] quando A é definida positiva real (veja [Cláudio, 1989]) ambos os métodos aproximam x_m da solução exata; porém, quando A não é positiva real, ambos os métodos podem falhar, ou seja, o GCR pode parar e o ORTHODIR pode não apresentar uma boa estabilidade numérica. Algumas relações entre o GMRES e o FOM podem ser encontradas em [Saad, 2003].

Em [Saad, 1992], Saad e Schultz apresentam o algoritmo iterativo Resíduo Mínimo Generalizado (GMRES), para resolver sistemas de equações lineares não simétricos, não singulares como em (1.1).

O GMRES é uma modificação do FOM descrito acima, que usa o processo de Arnoldi para calcular uma base ortonormal (v_1, v_2, \dots, v_k) do subespaço de Krylov $\kappa_k(A, v_1)$. A solução x_k é dada por $x_0 + Vy_k$, onde V é a matriz cujas colunas são os vetores v_k calculados pelo processo de Arnoldi. O vetor y_k é obtido a partir da solução do sistema $H_k y_k = \beta_k e_1$, onde H_k é uma matriz de Hessenberg superior $k \times k$, cujos elementos são calculados durante a ortogonalização no processo de Arnoldi, $\beta_k = \|r_0\|_2$, $r_0 = b - Ax_0$ e e_1 é o vetor canônico de dimensão k .

O GMRES possui quatro características importantes:

1. Nunca pára, a não ser que tenha convergido; [Saad e Schultz, 1986, pp. 865];
2. Converge em, no máximo, n iterações; [Saad e Schultz, 1986, pp. 865];
3. A convergência é monotônica, isto é, $\|r_{k+1}\|_2 \leq \|r_k\|_2$. Se o campo de valores de A , definido por

$$W(A) = \left(\frac{x^T Ax}{x^T x} \mid x \in \mathbb{C}^N \right) \quad (2.6)$$

[Nachtigal et al., 1990], estiver contido no semi-plano direito aberto, é possível que haja monotocidade estrita;

4. O número de vetores de ordem n que precisam ser armazenados cresce linearmente com k e o número de multiplicações cresce quadraticamente.

Apresentamos agora o algoritmo GMRES, segundo [Saad e Schultz, 1986, pp. 860]; nele tomamos ϵ como sendo o ϵ do computador, ou seja, o número de máquina:

Algoritmo 5: GMRES

1. *Início:* Escolha x_0 e calcule $r_0 = b - Ax_0$, $v_1 = \frac{r_0}{\|r_0\|_2}$
2. *Iterações:* Para $j = 1, 2, \dots, n$
3. $h_{ij} = (Av_j, v_i); i = 1, 2, \dots, j,$
4. $\hat{v}_{j+1} = Av_j - \sum_{i=1}^j h_{ij}v_i$
5. $h_{j+1,j} = \|\hat{v}_{j+1}\|_2$
6. Se $\|h_{j+1,j}\|_2 < \epsilon$
7. vá para (11.)
8. fim se
9. $v_{j+1} = \frac{\hat{v}_{j+1}}{h_{j+1,j}}$
10. fim para
11. $x_k = x_0 + V_k y_k$ onde y_k minimiza $J(y) = \|\beta e_1 - \bar{H}_k y\|_2$

Quando usamos o algoritmo GMRES pode-se facilmente usar a matriz H_k para estimar os autovalores de A , conforme comentado na subsecção 2.1. Se observarmos o algoritmo do GMRES com cuidado percebemos que o mesmo pode parar apenas no laço do processo de Arnoldi, quando $w_j = 0$, isto é, quando $h_{j+1,j} = 0$ no passo 5, assim o algoritmo pára porque o vetor de Arnoldi não pode ser generalizado devido à ocorrência de um subespaço A -invariante, [Saad, 2003]. É claro que o mesmo apresenta algumas dificuldades, por exemplo quando temos n sendo um valor grande, os custos de armazenamento e de flops crescem proporcionalmente a n^2 e n^3 , respectivamente, no pior caso; então para sanar tal dificuldade surgiu o método GMRES reinicializado, denotado por GMRES(m), onde $m \leq n$.

Durante a geração da base ortonormal escolhemos um valor m , e geramos uma aproximação para a solução usando uma base ortonormal de dimensão m . Claramente, com esse esquema a quantidade de armazenamento necessária pode ser calculada

previamente e mantida dentro dos limites aceitáveis. Apresentamos agora o algoritmo GMRES(m):

Algoritmo 6: GMRES(m)

1. *Início:* Escolha x_0 e compute $r_0 = b - Ax_0$, $v_1 = \frac{r_0}{\|r_0\|_2}$
2. *Iterações:* Para $k = 1, 2, \dots, \text{maxit}$
3. Para $j = 1, 2, \dots, m$
4. $h_{ij} = (Av_j, v_i); i = 1, 2, \dots, j$,
5. $\hat{v}_{j+1} = Av_j - \sum_{i=1}^j h_{ij}v_i$
6. $h_{j+1,j} = \|\hat{v}_{j+1}\|_2$
7. $v_{j+1} = \frac{\hat{v}_{j+1}}{h_{j+1,j}}$
8. fim para
9. $x_k = x_0 + V_k y_k$ onde y_k minimiza $J(y) = \|\beta e_1 - \bar{H}_k y\|_2, y \in \mathbb{R}^m$
10. Se $r_k = b - Ax_k$ satisfaz uma certa tolerância pré-especificada
11. para
12. senão $x_0 = x_k$, $v_1 = \frac{r_k}{\|r_k\|_2}$
13. fim se
14. fim para

As duas operações no GMRES(m) que consomem o maior tempo de execução são o processo de Arnoldi e a solução do problema de mínimos quadrados

$$\bar{H}_m y_m = \beta_k e_1 \quad (2.7)$$

A solução de (2.7) é obtida usando a fatoração QR de \bar{H}_m , onde y_m é a solução do sistema triangular

$$Ry_m = g, \quad g = Q\beta_k e_1 \quad (2.8)$$

Para maiores detalhes veja [Saad e Schultz, 1986].

Uma das características tanto do GMRES como do GMRES(m), é que a norma residual da solução aproximada pode ser obtida sem calcular x_k explicitamente. A norma do resíduo é dado por

$$\|\beta_m e_1 - \bar{H}_m y_m\|_2 = \|Q(\beta_m e_1 - \bar{H}_m y_m)\|_2 = \|g - Ry_m\|_2 \quad (2.9)$$

e por (2.9) esta norma é $\|g_{k+1}\|_2$.

Embora o GMRES(m) não pare, [Saad e Schultz, 1986, pp.865], é possível, dependendo do sistema e do valor de m , que produza uma sequência estacionária de resíduos e, assim, não convirja. Um resultado apresentado em [Saad e Schultz, 1986, pp. 866-867] para escolher um m apropriado para a reinicialização é dado como:

Teorema 1. *Suponha que A é diagonalizável tal que $A = V\Lambda V^{-1}$, assumamos que existem ν autovalores $\lambda_1, \lambda_2, \dots, \lambda_\nu$ de A com parte real não positiva e tome os outros autovalores sendo fechados em um círculo centrado em C , com $C > 0$ e tendo raio R com $C > R$. Então, o GMRES(m) converge para algum valor inicial de x_0 se*

$$m > \nu \frac{\log\left[\frac{DC}{dR} \kappa(X)^\frac{1}{\nu}\right]}{\log \frac{C}{R}} \quad (2.10)$$

onde $d = \min_{i=1, \dots, \nu} |\lambda_i|$ e $D = \max_{\substack{i=1, \dots, n \\ j=n+1, \dots, \nu}} |\lambda_i - \lambda_j|$

Prova: Consulte [Saad e Schultz, 1986].

Podemos perceber que a determinação desse valor mínimo m depende da determinação da distribuição dos autovalores de A e assim é usado somente em casos especiais, não podendo ser tomada como padrão na escolha do m . Ele mostraria que a direção da convergência do GMRES(m) aumenta ou permanece constante quando m aumenta. Maiores detalhes quanto à convergência do GMRES podem ser obtidos em [Saad, 2003].

Para comprovar que nem todas as matrizes satisfazem tal teorema, geramos uma matriz utilizando o programa MATLAB 5.3, a qual satisfaz as condições do teorema apresentado acima. Apresentamos os gráficos da estrutura e dos autovalores de tal matriz, na figura 2.1.

Para $n = 100$ obtivemos os seguintes valores para as incógnitas do teorema: $D = 192,5682$; $d = 54,0479$; $C = 3,0004$; $R = 1,4140$; $\nu = 17$, tal que $m > 84,8287$. Utilizando a mesma matriz no método GMRES(m) obtivemos $m = 20$. Observe que o teorema proposto em [Saad e Schultz, 1986] apresentou um m muito superior ao necessário para a resolução do sistema através do GMRES(m). Isso mostra que tal teorema não é adequado e, particularmente para as matrizes utilizadas nos testes que serão apresentados no capítulo 4, a distribuição dos autovalores das mesmas não satisfaz a hipótese do teorema.

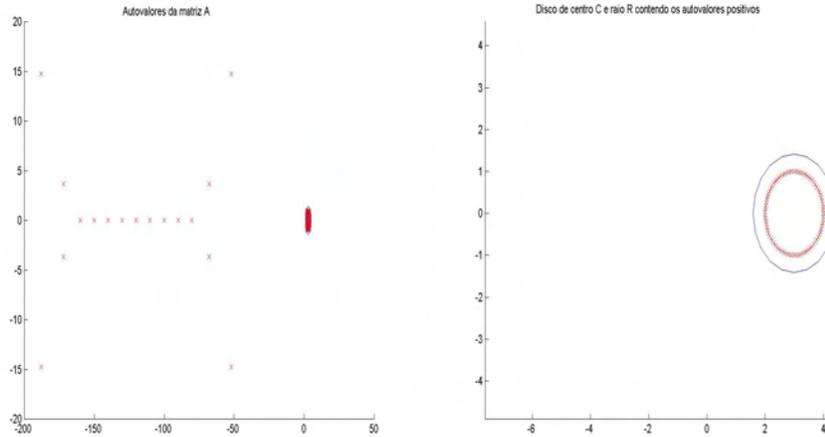


Figura 2.1: a) Autovalores da matriz, b) “Zoom” dos Autovalores da matriz

Em [Embree, 1995], foram propostos três condições para a convergência do GMRES, as quais envolvem os autovalores condicionados aos autovetores, o campo de valores e o pseudoespectro.

Segundo [Saad e Schultz, 1986] o algoritmo GMRES estima a solução geral x_k baseado na solução inicial x_0 , e que $r_k = b - Ax_k$ satisfaz a propriedade

$$\|r_k\|_2 \leq \min_{\substack{p \in \mathbf{P}_k \\ p(0)=1}} \|p(A)\|_2 \|r_0\|_2 \quad (2.11)$$

onde $p(A)$ é o polinômio de grau k ou menor com $p(0) = 1$. As três condições propostas por [Embree, 1995] partiram da equação (2.11) visando escolher um $\|p(A)\|_2$ apropriado.

Inicialmente [Embree, 1995] propôs a condição envolvendo os “Autovalores Condicionados aos Autovetores”, denotada por EV; o qual utiliza o conjunto dos autovalores de A , denotados por $\Lambda(A)$. Supondo que A é diagonalizável, $A = V\Lambda V^{-1}$, temos:

$$\|r_k\|_2 = \min_{\substack{p \in \mathbf{P}_k \\ p(0)=1}} \|p(A)r_0\|_2 \leq \|Vp(\Lambda)V^{-1}\|_2 \|r_0\|_2 \quad (2.12)$$

o qual podemos tomar como:

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \kappa(V) \min_{\substack{p \in \mathbf{P}_k \\ p(0)=1}} \max_{\lambda \in \Lambda(A)} \|p(\lambda)\|_2 \quad (2.13)$$

sendo $\kappa(V) = \|V\|_2 \|V^{-1}\|_2$, que é o número de condição da matriz de autovetores de A .

A segunda condição proposta envolve o “Campo de Valores”, denotados por FOV, que é o quociente de Rayleigh definido em (2.6), onde a maior magnitude de $W(A)$ é chamada de *Raio Numérico* e é dado por $\mu(A) = \max_{z \in W(A)} |z|_2$. O campo de valores forma sempre um conjunto convexo contendo $\Lambda(A)$, e são mais atrativos do que a condição anterior visto que não são sensíveis a pequenas perturbações. Eiermann and Ernst apresentaram em [Eiermann, 1989] e [Eiermann e Ernst, 1998] as condições do GMRES baseada no campo de valores, mostrando que se $0 \notin W(A)$, e que se para alguma matriz A temos $\|A\|_2 \leq 2\mu(A) \implies \|p(A)\|_2 \leq 2\mu(A)p(A)$, então:

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq 2 \min_{\substack{p \in \mathbf{P}_k \\ p(0)=1}} \max_{z \in W(A)} \|p(z)\|_2 \quad (2.14)$$

Essa condição pode não ser exata se existir estagnação inicial.

Por fim, foi sugerido uma condição envolvendo o “Pseudoespectro”, denotado por PSA; o qual é definido por $\Lambda_\epsilon(A) = \{z \in \mathcal{C} \mid \|zI - A^{-1}\|_2 \geq \epsilon^{-1}\}$ onde temos:

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \frac{\mathcal{L}(\Gamma_\epsilon)}{2\pi\epsilon} \min_{\substack{p \in \mathbf{P}_k \\ p(0)=1}} \max_{z \in \Lambda(A)} \|p(z)\|_2 \quad (2.15)$$

onde γ_ϵ é a união das curvas de Jordan que contém $\Lambda(A)$ no seu interior e $\mathcal{L}(\Gamma_\epsilon)$ é o tamanho do contorno das curvas de Jordan.

Note que cada uma das condições é associada a uma constante, ou seja:

$$C_{EV} = \kappa(V) \quad (2.16)$$

$$C_{FOV} = 2 \quad (2.17)$$

$$C_{PSA}(\epsilon) = \frac{\mathcal{L}(\Gamma_\epsilon)}{2\pi\epsilon} \quad (2.18)$$

O comportamento assintótico de cada condição é determinado pela aproximação complexa do problema, ou seja, $\Lambda(A)$, $W(A)$ e $\Lambda_\epsilon(A)$.

Podemos perceber que as condições apresentadas necessitam de elementos extras para que possamos utilizá-las, sendo que tais elementos não são triviais de serem calculados, o que as tornam inviáveis. Dessa forma, concluímos que nem o teorema proposto em [Saad e Schultz, 1986] nem as condições propostas em [Embree, 1995] são adequados para a escolha da dimensão do subespaço de Krylov, m .

É importante destacar que o GMRES(m) apresentado acima teve o processo de Arnoldi acompanhado do MGS; contudo, existe uma ortogonalização alternativa

baseada no uso das transformações de Householder, a qual é mais segura se os vetores ortogonalizados apresentam uma quase dependência linear numérica. Denotamos tal método de GMRESH(m), o qual foi proposto em [Walker, 1988].

O GMRESH(m) apresenta melhores propriedades numéricas que o GMRES(m), especialmente nas iterações finais, quando os resíduos tornam-se cada vez mais próximos, e utiliza menos armazenamento, apesar de requerer duas vezes mais operações aritméticas do que o GMRES(m). Segundo [Walker, 1988], no GMRESH(m) os vetores de Householder são de tamanho menor, e eles, junto com a matriz triangular superior gerada pelo algoritmo, podem ser armazenados em uma matriz de ordem $n \times (\text{maxit} + 1)$, onde maxit é o número máximo de iterações. Isto é um pouco menos armazenamento do que o requerido pelo GMRES(m), o qual armazena toda a base de vetores ortonormal, bem como as matrizes triangulares superiores. Assim, como em geral temos que $\text{maxit} \leq n$ então essa seria a maior vantagem entre esses dois algoritmos.

Apresentamos agora o algoritmo do GMRESH(m):

Algoritmo 7: GMRESH(m)

1. *Início:* Escolha x_0 e compute $r_0 = b - Ax_0$, $z = r_0$
2. *Iterações:* Para $k = 1, 2, \dots, \text{maxit}$
3. Para $j = 1, 2, \dots, m, m + 1$
4. $h_{ij} = (Av_j, v_i); i = 1, 2, \dots, j$,
5. Calcula o vetor unitário de Householder tal que
- $w_{ji} = 0, i = 1, 2, \dots, j - 1$
6. $P_j z_i = 0, i = j + 1, \dots, n$ onde $P_j = I - 2w_j w_j^T$
7. $h_{:,j-1} = P_j z$. Se $j = 1$ então tome $\beta = e_1^T h_0$
8. $v = P_j P_{j-1} \dots P_1 e_j$
9. Se $j \leq m$, calcula $z = P_j P_{j-1} \dots P_1 Av$
10. fim para
11. Defina $\bar{H}_m =$ parte superior de $[h_1, \dots, h_m]$ de dimensão $(m + 1) \times m$
12. Calcula $y_m = \min \|\beta e_1 - \bar{H}_m y\|_2$. Tome $y_m = (\eta_1, \dots, \eta_m)^T$
13. $z = 0$
14. Para $j = m, m - 1, \dots, 1$
15. $z = P_j \eta_j e_j + z$

16. fim para
17. Calcula $x_m = x_0 + z$
18. Se $r_m = b - Ax_m$ satisfaz uma certa tolerância pré-especificada
19. pára
20. caso contrário $x_0 = x_m, v_1 = \frac{r_m}{\|r_m\|_2}$
21. fim se
22. fim para

Sabemos que o custo $GMRESH(m)$ é aproximadamente duas vezes o custo do GMRES. Citamos agora, de acordo com [Da Cunha, 1992] o custo em relação as operações realizadas, bem como o custo em armazenamento do GMRES:

$$O_{GMRES} = \left(\frac{k^2 + 3k}{2}\right) O_{Au} + \left(\frac{k^2 + k}{2}\right) O_{u^T v}(n) + nO_{u^T v}(k) + kO_{|u|_2} + \left(\frac{nk^2 + nk}{2}\right) O_* + nkO_{\dot{v}} + O_{QR} + O_{BS} \quad (2.19)$$

onde O_{QR} é o custo das operações envolvendo a fatoração QR para resolver o sistema triangular superior de ordem k usando a retro-substituição $O_{BS} = kO_{\dot{v}} + \frac{k^2 - k}{2}O_*$. A expressão $O_{u^T v}(n)$ e $O_{u^T v}(k)$ são os produtos internos dos vetores de tamanho n e k . O termo $\frac{nk^2 + nk}{2}O_*$ é o número de multiplicações envolvidas na expressão $\sum_{j=1}^k \bar{H}_{j,k}^k V_{(:,j)}$.

O armazenamento requerido pelo GMRES poderia ser sumarizado como sendo:

$$S_{GMRES} = (2 + k)n + 2k^2 + 5k + 3 \quad (2.20)$$

2.5 Simpler GMRES - SGMRES

O Simpler GMRES surgiu em 1994 com Homer F. Walker e Lu Zhou; veja [Walker e Zhou, 1994], sendo apresentado com uma modificação no processo de Arnoldi original do GMRES. Segundo [Walker e Zhou, 1994], na implementação usual do GMRES, o processo de Arnoldi é aplicado com $v_1 = \frac{r_0}{\|r_0\|_2}$. A ortogonalização em cada passo do algoritmo pode ser feita usando o MGS ou HO. Note que o SGMRES é resultante apenas da modificação feita no v_1 , tornando assim o problema (2.7) em um problema de mínimos quadrados triangular superior, eliminando a matriz de triangularização da matriz de Hessemberg.

Apresentamos os algoritmos reinicializados do Simpler GMRES com a transformação de Householder, o qual denotaremos por SGMRESH, e o SGMRES com o processo MGS.

Algoritmo 8: SGMRES(m)

1. *Início:* Escolha x e calcule $r = b - Ax$, $r = \frac{r}{\|r\|_2}$ e tome $\rho = 1$
2. *Iterações:* Para $k = 1, 2, \dots, \text{maxit}$
3. Para $k = 1, \dots, m$
4. $v_k = Av_{k-1}$, sendo $v_1 = Ar$
5. Se $k > 1$ então para $i = 1, \dots, k - 1$
6. Tome $\rho_{i,k} = v_i^T v_k$
7. $v_k = v_k - \rho_{i,k} v_i$
8. Tome $\rho_{k,k} = \|v_k\|_2$, atualize $v_k = \frac{v_k}{\rho_{k,k}}$
9. Tome

$$\mathbf{R}_k = \begin{pmatrix} \rho_{11} & \cdots & \rho_{1,k} \\ & \ddots & \vdots \\ & & \rho_{k,k} \end{pmatrix}$$
10. Tome $\xi = r^T v_k$. Atualize $\rho = \rho \sin(\cos^{-1} \frac{\xi_k}{\rho})$.
11. Se $\rho \rho_0 \leq \text{tol}$
12. vá para o passo (15.)
13. fim se
13. $r = \xi_k v_k$
14. fim para
15. *Resolva:* Tome k sendo o número final de iterações
16. $R_k y = (\xi_1, \dots, \xi_k)^T$ para $y = (\eta_1, \dots, \eta_k)^T$
17. Forma $z = \begin{cases} \eta_1 r & \text{se } k = 1 \\ \eta_1 r + \sum_{i=1}^{k-1} \eta_{i+1} (\eta_i \xi_i) v_i & \text{se } k > 1 \end{cases}$
18. $x = x + \rho z$
19. Se $\rho \rho_0 < \text{tol}$ pará
20. senão $r = \frac{r - \xi_k v_k}{\rho}$, $\rho_0 = \rho \rho_0$ e $\rho = 1$.
21. fim se
22. fim para

Algoritmo 9: SGMRESH(m)

1. *Início:* Escolha x e calcule $r = b - Ax$, $r = \frac{r}{\|r\|_2}$ e tome $\rho = 1$
2. *Iterações:* Para $k = 1, 2, \dots, \text{maxit}$
3. Para $k = 1, \dots, m$
4. $\tilde{v}_k = P_{k-1} \dots P_1 A P_1 \dots P_{k-1} e_{k-1} (\tilde{v} = Ar)$
5. Determine P_k tal que $P_k \tilde{v}_k = (\rho_{1,k}, \dots, \rho_{k,k}, 0, \dots)^T$
6. Tome

$$\mathbf{R}_k = \begin{pmatrix} \rho_{11} & \dots & \rho_{1,k} \\ & \ddots & \vdots \\ & & \rho_{k,k} \end{pmatrix}$$
7. $r = P_k r$; tome $\xi_k = e_k^T r$ e $\rho = \rho \sin(\cos^{-1} \frac{\xi_k}{\rho})$
8. Se $\rho \rho_0 < \text{tol}$
9. vá para (12.)
10. fim se
11. fim para
12. *Resolva:* Tome k sendo o número final de iterações
13. $R_k y = (\xi_1, \dots, \xi_k)^T$ para $y = (\eta_1, \dots, \eta_k)^T$
14. Forma $z = \begin{cases} P_1 \eta_1 r & \text{se } k = 1 \\ P_1 \dots P_k [\eta_1 r + (\eta_2, \dots, \eta_n, 0, \dots)^T] \end{cases}$ se $k > 1$
15. $x = x + \rho_0 z$
16. Se $\rho \rho_0 < \text{tol}$
17. pára
18. senão
19. $r = P_1 \dots P_k \begin{pmatrix} 0 \\ I_{n-k} \end{pmatrix} \frac{r}{\rho}$, $\rho_0 = \rho \rho_0$ e $\rho = 1$, retorna a (2.)
20. fim para

Observe que no passo (10.) e (7.) do algoritmo do SGMRES e do SGMRESH, respectivamente, temos que, se $\xi = 0$, isso significa que o GMRES estagnou. Segundo [Goossens e Roose, 1997], através dos valores harmônicos de Ritz e dos valores de Ritz pode-se explicar a estagnação do GMRES e do FOM. Entende-se por valores de Ritz a aproximação dos autovalores da matriz A e por valores harmônicos de Ritz os valores de

Ritz da matriz A^{-1} calculados a partir de $A\kappa_m(A, r_0)$. Sabe-se que, através do processo de Arnoldi obtemos a matriz triangular superior de Hessenberg que satisfaz a relação

$$AV_m = V_{m+1}\bar{H}_m \quad (2.21)$$

onde $\bar{H} \in \mathbb{R}^{(m+1) \times m}$ é a matriz de Hessenberg aumentada com uma linha da forma $(0, 0, \dots, h_{m+1,m})$. A partir de (2.21) obtêm-se a seguinte relação

$$H_m = V_m^T AV_m \quad (2.22)$$

Em [Goossens e Roose, 1997] foi apresentada a seguinte relação:

$$\|h_{m+1,m}^2 f_m e_m^T\|_2 \leq \frac{h_{m+1,m}^2}{\sigma_{\min}(H_m)} \quad (2.23)$$

onde $\sigma_{\min}(H_m)$ é o menor valor harmônico de Ritz de A ; pode concluir que os valores de Ritz são iguais aos valores harmônicos de Ritz quando o subespaço invariante for descoberto, ou seja, no caso em que $h_{m+1,m} = 0$. Quando $h_{m+1,m}$ é grande e $\sigma_{\min}(H_m)$ é pequeno, temos o caso da estagnação do GMRES, o qual pode ser traduzido como uma diferença significativa entre os valores harmônicos de Ritz e os valores de Ritz. Destacamos que a curva residual do GMRES apresenta-se estagnada quando os valores harmônicos de Ritz tornam-se muito grandes.

A estagnação do SGMRES corresponde ao fato da projeção $\xi_1 = 0$, o qual apresentou os valores harmônicos de Ritz sendo iguais a infinito.

Os dois algoritmos acima são extremamente fáceis de implementar e requerem menos de $O(m^2)$ operações aritméticas dentro de cada ciclo de iterações nos m passos das implementações usuais do GS e HO. No algoritmo do SGMRES o vetor residual é sempre avaliado ou facilmente obtido. Também podemos destacar que na implementação usual do GMRES, as colunas de B_k são os vetores ortogonais $\{v_1, \dots, v_k\}$ produzidos pelo processo de Arnoldi com $v_1 = \frac{r_0}{\|r_0\|_2}$, conseqüentemente este B_k é idealmente condicionado. Na implementação do SGMRES temos:

$$B_k = \left(\frac{r_0}{\|r_0\|_2}, v_1, v_2, \dots, v_{k-1} \right) \quad (2.24)$$

onde v_1, \dots, v_{k-1} são produzidos pelo processo de Arnoldi com $v_1 = \frac{Ar_0}{\|Ar_0\|_2}$.

2.5.1 Loose GMRES - LGMRES(m, k)

O Loose GMRES surgiu em 2003 com Baker, Jessup e Manteuffel; veja [Baker et al., 2003]; sendo apresentado com uma técnica para acelerar a convergência do GMRES.

O LGMRES(m, k) é muito semelhante ao método do gradiente conjugado completo (CG) com preconditionador polinomial, e sua implementação não exige muitas modificações no GMRES(m).

Segundo [Baker et al., 2003], uma motivação para propor o LGMRES se deu pelo fato de que o GMRES(m) não mantém a ortogonalidade entre espaços aproximados generalizados em sucessivas reinicializações, tornando assim sua convergência vagarosa ou até mesmo levando a divergência. Também é sabido, conforme [Chapman e Saad, 1997], que métodos aumentados formam uma classe de técnicas de aceleração; esses métodos buscam evitar a divergência melhorando assim as informações obtidas com o GMRES em cada reinicialização. Tipicamente, um subespaço A -invariante é agrupado ao subespaço aproximado de Krylov, resultando assim em um subespaço de Krylov aumentado. O subespaço A -invariante associado com os menores autovalores é comumente usado.

A idéia então consiste de adicionar vetores ao subespaço aproximado de Krylov que representaria o espaço aproximado para cada ciclo de reinicialização, ou seja, depois do ciclo $i + 1$ teríamos que $r_{i+1} \perp A\kappa_m(A, r_i)$. Idealmente $r_{i+1} \perp A\kappa_m(A, r_{i-1})$ também serviria, mas infelizmente o GMRES(m) descarta $\kappa_m(A, r_{i-1})$ antes de começar o ciclo $i + 1$.

Suponha que \hat{x} é a solução exata do problema (1.1). O erro depois de i ciclos de reinicialização é denotado por e_i , onde $e_i = \hat{x} - x_i$. Em [Baker et al., 2003] definiu-se $z_i = x_i - x_{i-1}$ como sendo o i -ésimo erro aproximado no i -ésimo ciclo de reinicialização. Como $z_i \in \kappa_m(A, r_{i-1})$, este erro aproximado é a escolha natural do vetor que aumentará o próximo espaço aproximado $\kappa_m(A, r_i)$. Assim, se o espaço aproximado contém a correção exata e_i , tal que $\hat{x} = x_i + e_i$ teríamos o problema resolvido. Daí incluir a aproximação e_i , tal como z_i , no espaço aproximado apresenta-se como uma estratégia aceitável.

Denota-se o novo algoritmo do GMRES(m) aumentado como LGMRES(m, k), o qual aumenta o espaço aproximado de Krylov com k vetores erros aproximados. Então, no final do ciclo de reinicialização $i + 1$, o LGMRES(m, k) acha a solução aproximada de (1.1), como segue:

$$x_{i+1} = x_i + q_{i+1}^{m-1}(A)r_i + \sum_{j=i-k+1}^i \alpha_{ij}z_j \quad (2.25)$$

onde o polinômio q_{i+1}^{m-1} e α_{ij} são escolhidos tais que $\|r_{i+1}\|_2$ é minimizada. Em particular, quando em (2.25), $k = 0$ o LGMRES(m, k) reduz ao GMRES(m).

Apresentamos o algoritmo proposto por [Baker et al., 2003]:

Algoritmo 10: LGMRES(m, k)

1. *Início:* $r_i = b - Ax_i$, $\beta = \|r_i\|_2$, $v_1 = \frac{r_i}{\beta}$, $s = m + k$
2. *Iterações:* Para $k = 1, 2, \dots, \text{maxit}$
3. Para $j = 1, 2, \dots, s$
4. $u = \begin{cases} Av_j & \text{se } j < m \\ Az_{i-(j-m-1)} & \text{nos outros casos} \end{cases}$
5. Para $l = 1, \dots, j$
6. $h_{l,j} = uv_l^T$
7. $u = u - h_{l,j}V_l$
8. fim para
9. $h_{j+1,j} = \|u\|_2$, $v_{j+1} = \frac{u}{h_{j+1,j}}$
10. fim para
11. $W_s = [v_1, \dots, v_m, z_i, \dots, z_{i-k+1}]$, $H_s = h_{l,j_{1 < l < j+1; 1 < j < s}}$
12. Acha y_s tal que (2.7) é minimizado
13. $z_{i+1} = W_s y_s$ (também $Az_{i+1} = V_{s+1} H_s y_s$)
14. $x_{i+1} = x_i + z_{i+1}$
15. fim para

Note que $V_{s+1} = [v_1, v_2, \dots, v_m, \dots, v_{m+k+1}]$ e $W_s = [v_1, \dots, v_m, z_i, \dots, z_{i-k+1}]$ então, a relação $AW_s = V_{s+1}H_s$ também é válida para o LGMRES(m, k), onde H_s denota a matriz $(s + 1) \times s$ de Hessenberg.

Na tabela (2.2) resumimos o custo do GMRES($m + k$) e do LGMRES(s), onde $s = m + k$, sendo que a segunda coluna refere-se ao armazenamento de vetores de tamanho n e a terceira coluna ao número de multiplicações matriz-vetor por ciclo de reinicialização.

	Armazenamento	Multiplicações
GMRES(m, k)	$m + k + 3$	$m + k$
LGMRES(s)	$m + 3k + 3$	m

Tabela 2.2: GMRES e LGMRES

Haveria uma possibilidade de diminuir o custo quanto ao armazenamento referente a segunda coluna da tabela; bastaria calcular Az_i em cada ciclo de reinicialização, tornando-a $m + 2k + 3$. Contudo, isso acarretaria em um acréscimo do custo em relação ao número de multiplicações matriz-vetor, ou seja, teríamos $m + k$ multiplicações por ciclo de reinicialização, o que tornaria o método desvantajoso em relação à uma possível convergência.

Note que o LGMRES(m, k) tipicamente não requer mais iterações do que o GMRES(m), e que o fato do mesmo ser um acelerador de uma possível convergência do GMRES(m) não significa que o mesmo não possa ser aplicado sem o uso de pré-condicionadores.

2.6 Diagrama dos Métodos

Apresentamos a seguir um diagrama que mostra as principais características dos métodos bem como a evolução histórica dos mesmos.

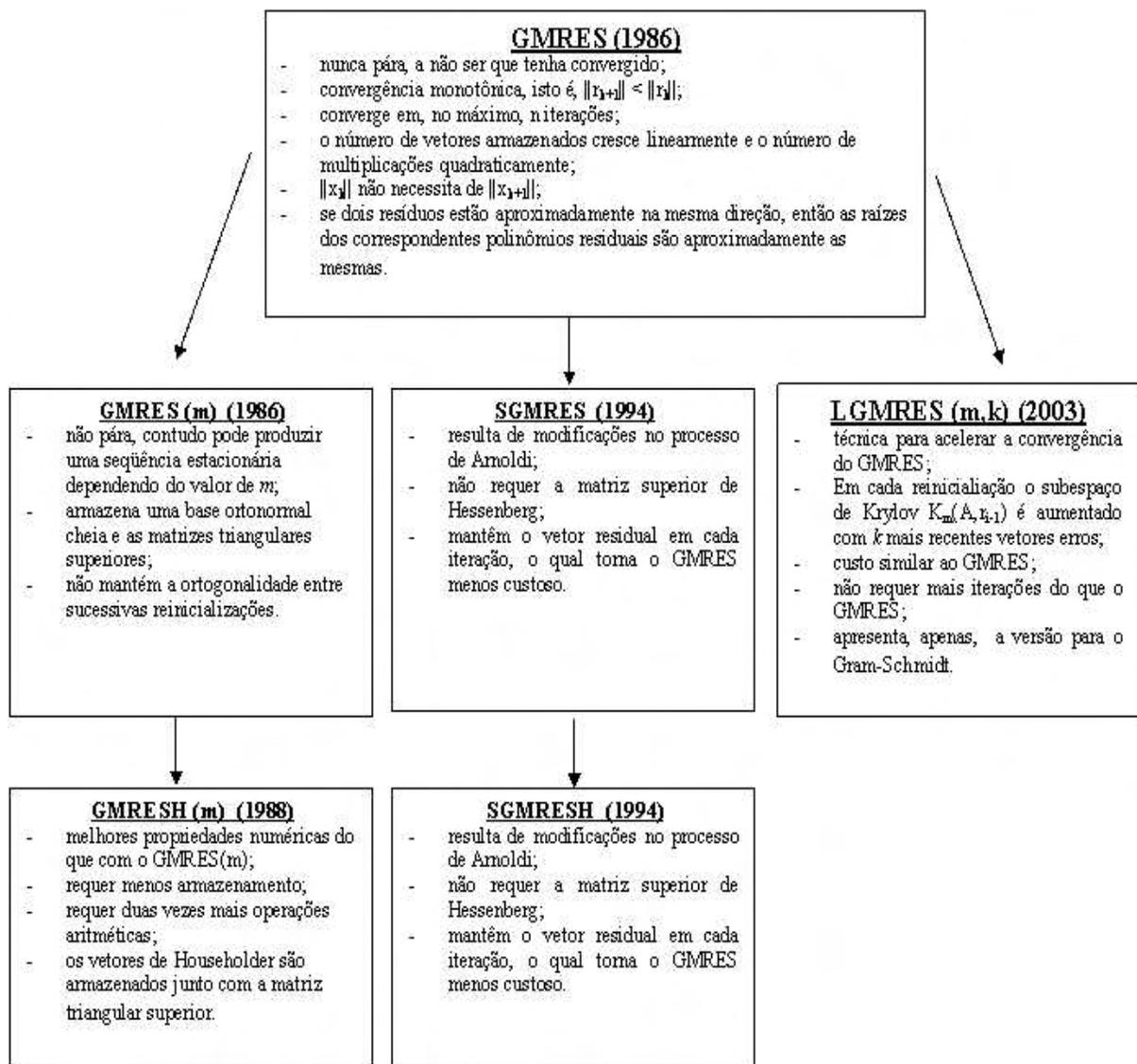


Figura 2.2: Diagrama dos Métodos

3 ELABORAÇÃO DOS CRITÉRIOS

Neste capítulo apresentamos, em ordem cronológica, os diversos critérios elaborados na tentativa de obter a melhor dimensão do subespaço de Krylov, m , bem como as estratégias e técnicas utilizadas em cada escolha realizada.

O primeiro critério apresentado foi inspirado nos testes para prevenir a estagnação presentes no A-GMRES, veja [Driver, 1997]; os demais critérios foram desenvolvidos utilizando conceitos matemáticos, não sendo inspirados em nada existente na literatura atual, ou seja, os mesmos tiveram sua primeira aparição nessa dissertação.

Nos testes que seguem e no capítulo 4 alguns parâmetros foram utilizados, os quais descrevemos agora: utilizou-se uma tolerância de “ 10^{-6} ”, o parâmetro de entrada “maxit” é o número máximo de iterações permitidas, “ m_{orig} ” é o valor da base inicialmente, “ $\|r_k\|_2$ ” é o valor da norma na k -ésima iteração, “mmax” é o máximo valor que o m pode assumir, e é dado através da fórmula:

$$mmax = \min(mmax, mlimit) \quad (3.1)$$

$$mlimit = \min\left(\max(mmax, \sqrt{n}), \frac{n+1}{2}\right) \quad (3.2)$$

Antes de iniciarmos os critérios elaborados propriamente dito, faremos a apresentação de dois métodos adaptativos apresentados por [Driver, 1997] e [Nodera e Moriya, 2003].

O critério proposto em [Driver, 1997] serviu de inspiração para a elaboração do primeiro critério apresentado nesse trabalho e também serviu de fonte de amadurecimento para os demais critérios elaborados.

A partir desse capítulo chamamos de métodos **Adaptativos** aqueles métodos que são acrescidos de um dos critério proposto e de métodos **Não adaptativos** os que não apresentam a presença dos mesmos.

3.1 GMRES-ADAPTATIVO

Dedicamos essa seção ao método GMRES-Adaptativo proposto por Maria Sosonkina Driver em 1997 [Driver, 1997], o qual denotamos por A-GMRES.

A diferença básica em relação ao GMRESH(m) consiste no fato de que o A-GMRES apresenta, se necessário, alguns testes para incrementar o m e então reinicializar o método.

De acordo com [Driver, 1997] a essência do A-GMRES é obter um parâmetro m adequado para a reinicialização do problema. Sabe-se que em [Brown e Walker, 1997] foram discutidos testes de estagnação para sistemas singulares, sendo esses somente aplicados antes do novo ciclo de reinicialização; contudo nada foi proposto para adaptar a estratégia com a dimensão do subespaço de Krylov.

O A-GMRES proposto emprega testes de estagnação para redução da norma residual. O método é declarado como tendo estagnado e as iterações são paradas se a tolerância não pode ser encontrada para o maior múltiplo do máximo número de ciclos permitido, denotado em [Driver, 1997] por “bgv”, essa estimativa do número de ciclos permitidos foi chamada de “*teste*” no algoritmo do A-GMRES.

Conforme visto no capítulo 2, sabe-se que quando a convergência do GMRES(m) é vagarosa indica que o número de vetores do subespaço não é suficiente para resolver tal sistema; podemos então aumentar o valor de m , que deverá beneficiar o método. Quando isso se faz necessário o método segue seu processo no mesmo ciclo onde houve a estagnação, ou seja, repete o ciclo com o novo valor do m . Assim, podemos concluir que a reinicialização para o A-GMRES significaria repetir a iteração não produtiva, onde obteve-se a estagnação parcial ou total.

Veja o algoritmo do A-GMRES abaixo:

A-GMRES:

1. *Início:* Escolha x , \maxit , m e $mmax$;
2. *Iterações:* $r = b - Ax$, $i = 0$, $cnmax = \frac{1}{50\epsilon}$
3. Calcule (3.1), (3.2) e $mincr = 4$
4. Enquanto $\|r\|_2 > tol$
5. $r_{i-1} = r_i$;
6. Determina $P_1 r = \|r\|_2 e_1$
7. $k1 = 1$ e $k2 = m$
8. **L1:** Para $j = k1, \dots, k2$

9. $i = i + 1$ e $v = P_j \dots P_1 A P_1 \dots P_j e_j$
10. Determina P_{j+1}
11. Atualiza $\|r\|_2$
12. Estima $\text{cond}(AV_j)$ para o problema de mínimos quadrados onde
 $V_j = [P_1 e_1, \dots, P_1 \dots P_j e_j]$
13. Se $ICN > cnmax$ então finaliza.
14. Se $\|r\|_2 < tol$ então vá para **L2**
15. $teste = \left(k2 \log \frac{tol}{\|r\|_2} \right) / \left(\log \frac{\|r\|_2}{(1+10u)\|r_{i-1}\|_2} \right)$
16. Se $k2 \leq mmax - m$ e $smv(maxit - i) \leq teste$
17. $k1 = k2 + 1$ e $k2 = k2 + m$
18. fim se
19. Vá para **L1**
20. fim para
21. **L2:** Resolva $\min \left\| \|r\|_2 e_1 - \tilde{H} y \right\|_2$ para y_j .
22. $q = (y_j, 0)^T$ e $x = x + P_1 \dots P_j q$
23. Se $\|r_i\|_2 \leq tol$ então finaliza
24. Se $\|r_{i-1}\|_2 \leq \|r_i\|_2$ então
25. Se $\|r_i\|_2 < tol^{\frac{2}{3}}$ então finaliza
26. caso contrário retorna
27. fim se
28. $teste = \left(k2 \log \frac{tol}{\|r\|_2} \right) / \left(\log \frac{\|r\|_2}{(1+10u)\|r_{i-1}\|_2} \right)$
29. Se $bqv(maxit - i) \leq teste$ então retorna
30. fim enquanto.

A tolerância da convergência é dada por:

$$tol = (\|r_0\|_2, \|b\|) xtol \quad (3.3)$$

onde o erro de arredondamento da multiplicação matriz-vetor depende apenas das entradas não nulas em cada linha da matriz esparsa, e o erro da tolerância denotado por “ $xtol$ ” é proporcional a média do número de elementos não nulos por linhas, ou seja,

$$xtol = \frac{nz}{n} \quad (3.4)$$

Se a norma residual na reinicialização é menor do que a norma anterior, então o A-GMRES termina, e a convergência é considerada sendo obtida quando

$$\|r\|_2 < tol^{\frac{2}{3}}. \quad (3.5)$$

O número de condição do problema de mínimos quadrados do GMRES(m) é monitorado pela estimativa incremental da condição [Bischof e Tang, 1991] e o mesmo é finalizado quando o número de condição estimado é maior do que

$$\frac{1}{50\epsilon} \quad (3.6)$$

onde ϵ é o número de máquina.

Os resultados dos experimentos utilizando tal método, bem como a comparação do mesmo com o GMRESH(m) serão apresentados no próximo capítulo.

3.2 BC-GMRES

O BC-GMRES foi proposto por Takashi Nodera e Kentaro Moriya em 2003 [Nodera e Moriya, 2003], e é um método que escolhe o ciclo de reinicialização do m baseado no teste da norma residual e na distribuição dos zeros do polinômio residual do GMRES(m). Essa estratégia foi baseada em algumas conclusões apresentadas em [Tsuno e Nodera, 1999].

O BC-GMRES executa a reinicialização quando a distribuição dos zeros do polinômio residual do GMRES(m) tornam-se ideais, ou seja, pode-se dizer que quanto mais espalhados estão os zeros do polinômio residual mais provável que a distribuição dos zeros aproximados seja similar aos zeros ideais. A figura 3.1 mostra essa situação.

No BC-GMRES a decisão da reinicialização é tomada através da análise de duas condições, a saber: a distribuição dos zeros aproximados e o teste de convergência da norma residual.

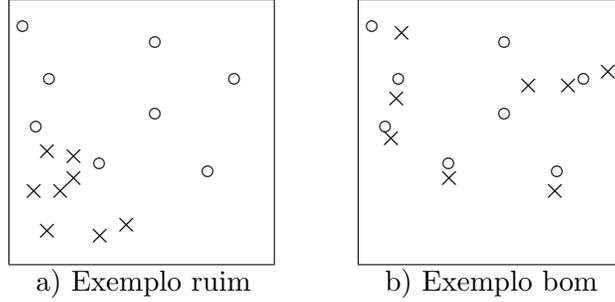


Figura 3.1: IMPCOL_A: Distribuição de zeros do polinômio residual (o: zeros ideais; ×: zeros aproximados). Extraído de [Nodera e Moriya, 2003].

Condição 1: *Condição para a distribuição dos zeros aproximados*

Em [Tsuno e Nodera, 1999] o ciclo de reinicialização é escolhido quando a seguinte condição é satisfeita:

$$\forall \lambda_{s_1}^{i+1}, \exists \lambda_{s_2}^j, \text{ na região retangular,} \\ T(\lambda_{s_1}^{i+1}) = \left(z \in \mathbb{C}, \text{re}\|\lambda_{s_1}^{i+1} - z\| < M_{re} \frac{l-1}{2}, \text{im}\|\lambda_{s_1}^{i+1} - z\| < M_{im} \frac{l-1}{2} \right) \quad (3.7)$$

onde

$$M_{re} = \max_{s_1, s_2, j} \{\text{re}(\lambda_{s_1}^{i+1}), \text{re}(\lambda_{s_2}^j)\} - \min_{s_1, s_2, j} \{\text{re}(\lambda_{s_1}^{i+1}), \text{re}(\lambda_{s_2}^j)\}$$

$$M_{im} = \max_{s_1, s_2, j} \{\text{im}(\lambda_{s_1}^{i+1}), \text{im}(\lambda_{s_2}^j)\} - \min_{s_1, s_2, j} \{\text{im}(\lambda_{s_1}^{i+1}), \text{im}(\lambda_{s_2}^j)\}$$

Para mais detalhes veja [Tsuno e Nodera, 1999].

Condição 2: *Condição da convergência da norma residual*

O vetor residual na m -ésima iteração, com $m = \tilde{m} + k$, é dado por

$$r_m = r_{\tilde{m}} + d_k, \quad d_k = AV_k y_k \quad (3.8)$$

onde \tilde{m} é a iteração que sofreu a última reinicialização, $r_{\tilde{m}}$ é o m -ésimo vetor residual e d_k é o vetor busca e pode ser obtido durante o processo de Arnoldi. Em [Nodera e Moriya, 2003] foi definido o seguinte parâmetro para o teste da convergência da norma residual

$$\begin{aligned} \zeta_m &= \frac{(r_{\tilde{m}}, d_k)}{\|r_{\tilde{m}}\|_2 \|d_k\|_2} = \\ &= \sqrt{1 - \frac{\|r_m\|_2^2}{\|r_{\tilde{m}}\|_2^2}} \end{aligned} \quad (3.9)$$

Quando ζ_m é relativamente grande, a norma residual converge rapidamente. Quando a condição

$$\zeta_m > \varepsilon \quad (3.10)$$

é satisfeita, considera-se a convergência da norma residual como não estagnada. Assim, quando a condição 1 não é satisfeita, executa-se a reinicialização com a condição dada por (3.10) se esta é satisfeita.

Pode-se resumir o processo de reinicialização baseado nessas duas condições como sendo:

1. Quando o número de iterações é par, calcula k zeros e ζ_m ; atribui $\rho = \zeta_m$.
2. A determinação da reinicialização é executada:
 - 2.1. se a condição 1 é satisfeita ou o número total de iterações é 2, a reinicialização é executada com $\rho = \varepsilon$.
 - 2.2. se (3.9) é satisfeita ou a ciclo de reinicialização é m_{max} , a reinicialização é executada com $\rho = \varepsilon$.

Em resumo pode-se dizer analisando as condições acima que a corrente ζ_m é o conjunto de ε quando os zeros aproximados são relativamente espalhados.

Apresentamos abaixo o algoritmo do BC-GMRES conforme utilizamos nesse trabalho. Para obter o algoritmo original veja [Nodera e Moriya, 2003].

BC-GMRES:

1. *Início*: Escolha $x, maxit, m, mmax$;
2. *Iterações*: $r = b - Ax, m = 0, \tilde{m} = 0, k = 1, s2 = \emptyset$
3. Calcula V_k através do processo de Arnoldi.
4. Se $\|r_m\|_2 < tol$ então finaliza
5. Se $mod(k, 2) = 0$
6. Calcula os autovalores de $H_{k \times k}$ armazenando-os em um arranjo s_1 .
7. $\zeta_m = \sqrt{1 - \frac{\|r_m\|_2^2}{\|\tilde{r}_m\|_2^2}}$
8. Se o arranjo s_2 é vazio
9. testa a condição 1

10. fim se
11. Se $m = 2$
12. $s_2 = s_1, k = 0, \tilde{m} = m, r_{\tilde{m}} \text{ e } \epsilon = \rho$
13. caso contrário se $\zeta > \epsilon$ ou $k = mmax$
14. $s_2 = s_1, k = 0, \tilde{m} = m \text{ e } r_{\tilde{m}} = r$
15. fim se
16. $k = k + 1$
17. fim se

Os métodos adaptativos descritos acima serão utilizados nas comparações apresentadas no capítulo 4.

Apresentamos agora, em ordem cronológica, os critérios elaborados ao longo desse trabalho. Analizou-se os seguintes itens no resultado final apresentado com cada critério: número de flops seguido do número de iterações necessárias para obter a convergência.

Todos os critérios desenvolvidos foram alocados, um a um, no método GMRES(m) a fim de avaliar o rendimento dos mesmos, sendo dispostos antes de iniciar o processo de Arnoldi. Utilizamos um sistema cuja matriz dos coeficientes é a matriz CDDE1 [MCSD/ITL/NIST, 2003], com termo independente formado por um vetor aleatório e o GMRES(m) com m fixo ao longo das iterações.

3.3 Critério 1

Começamos o nosso estudo sobre a melhor escolha da dimensão do subespaço de Krylov através de [Driver, 1997]. Inicialmente fizemos algumas modificações no critério utilizado em [Driver, 1997] adequando-o ao nosso objetivo de estudo.

Critério 1:

1. *Início:* teste = $m \left(\log \frac{tol}{\|r\|_2} \right) / \left(\log \frac{\|r_n\|_2}{(1+10u)\|r_{n-1}\|_2} \right)$
2. Se teste $> 40(maxit - it)$ e $k_2 < mmax - m$
3. $m = m + m$

4. caso contrário
5. $m = m$
6. fim se

Nossa preocupação inicial consistiu em elaborar um critério que apresentasse resultados satisfatórios quanto ao número de convergência dos métodos adaptativos e ao número de não convergências.

Nos três primeiros critérios o m apenas sofre acréscimos ou permanece quase que sem alteração; somente no quarto critério ocorre o fato do m sofrer acréscimos e decréscimos de acordo com a necessidade do sistema estudado, visto que no mesmo já tínhamos descoberto qual o melhor conceito a se utilizar a título que tal critério fosse realmente satisfatório. As matrizes teste foram extraídas do sítio “Matrix Market” [MCSD/ITL/NIST, 2003].

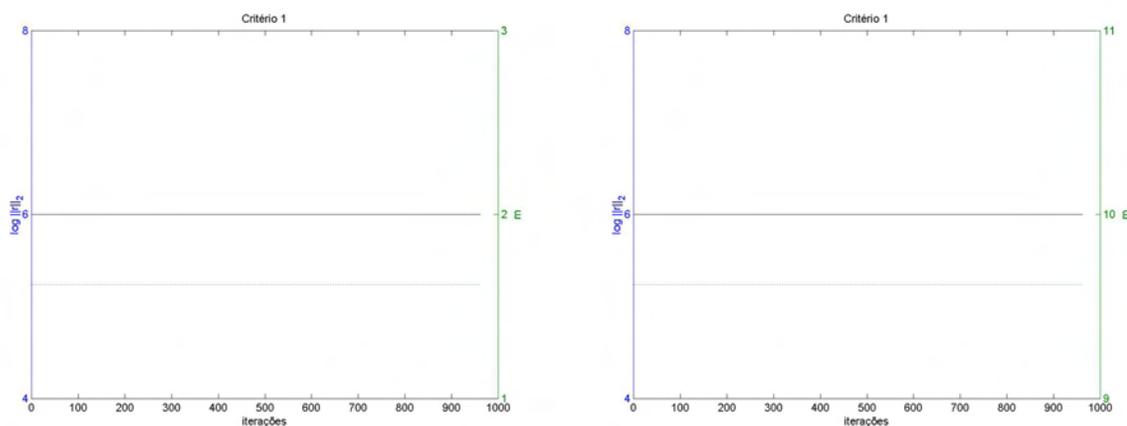


Figura 3.2: GMRES Adaptativo: a) $m = 2$ e b) $m = 10$

Os gráficos da figura (3.2) apresentam o $\log \|r\|_2$ e o m versus o número de iterações, para a matriz CDDE1, com $m = 2$ e $m = 10$, respectivamente. Note que em ambos os gráficos m não sofreu nenhum tipo de alteração durante as iterações; e o método também não convergiu. Concluímos que esse critério necessitava sofrer alterações para que m apresentasse variação e ocorresse uma possível convergência.

A tabela (3.1) mostra os resultados finais quanto a utilização do critério 1.

Total	Adaptativos	Não adaptativos	Não convergiram
150	20	40	90

Tabela 3.1: Critério 1

3.4 Critério 2

Neste critério utilizou-se o número de dígitos significativos exatos (DIGSE) [Cláudio et al., 1987]:

Corolário: A exatidão matemática (DIGSE) de uma aproximação \tilde{x} em relação a um solução teórica $x \neq 0$ é dada por:

$$digse(x) = - \left(0,3 + \log \left(\mu + \frac{|x - \tilde{x}|}{|x|} \right) \right) \quad (3.11)$$

com $\mu = \begin{cases} \frac{b^{1-n}}{2} & \text{arredondamento para o número de máquina mais próximo} \\ b^{1-n} & \text{truncamento} \end{cases}$ onde b é a base do sistema de ponto flutuante e n o número de dígitos na mantissa.

É interessante salientar que quando o DIGSE é um valor muito alto, isso implica estagnação do método. No critério 2 utilizou-se precisão dupla; conforme o MATLAB.

Critério 2:

1. Se $mod(i, 5) = 0$
2. $ndigs = DIGSE(\|r\|_2, \|r_{i-1}\|_2)$
3. Se $ndigs > 30$ então
4. $m = \min(2m, mmax)$
5. caso contrário $ndigs > 0$
6. $m = \min(m + \frac{2m}{3}, mmax)$
7. fim se
8. fim se

Observe que no passo (1.) do critério acima há um teste que permite que a $\|r\|_2$ permaneça sem alteração por, no máximo, cinco iterações. Tal procedimento foi utilizado anteriormente no algoritmo adaptativo para o método SOR; veja

[Hageman e Young, 1981]. Note que estipulamos que ndigs deveria ser maior do que 30 ou maior do que 0, tais valores foram escolhidos experimentalmente por tentativa e erro. Os gráficos na figura (3.3) apresentam a curva do $\log \|r\|_2$ e da variação de m versus o número de iterações.

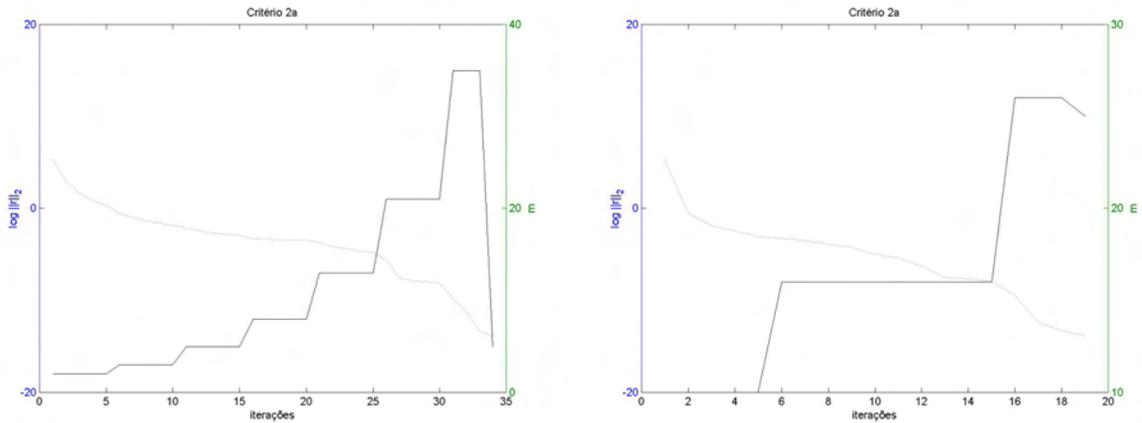


Figura 3.3: DIGSE.1: a) $m = 2$ e b) $m = 10$

Os gráficos foram somente plotados com a matriz CDDE1 tendo $m = 2$ e $m = 10$, respectivamente. Para $m = 2$, $\log \|r\|_2$ se manteve quase que constante até $25 < i < 30$, sendo que quando $i > 30$ o mesmo começou a decrescer lentamente. Quanto à curva de variação do m , pode-se notar que a mesma apresentou-se em forma de “degraus”, crescendo em alguns momentos e mantendo-se constante em outros, alcançando, praticamente, seu valor máximo. O número de flops apresentados foi de $2,7104 \times 10^7$ e $5,3808 \times 10^8$, para $m = 2$ e $m = 10$, respectivamente.

Para $m = 10$, $\log \|r\|_2$ apresentou pouco decréscimo até aproximadamente $i = 10$, depois a mesma decresceu rapidamente. Note que, novamente, a curva de variação do m apresentou momentos de estagnação e cresceu até $m > 40$; o decréscimo apresentado pelo m em tal gráfico não se dá pelo critério utilizado e sim pelo processo de Arnoldi, ou seja, pelo fato de ter sido detectado um subespaço invariante no subespaço de Krylov.

A tabela (3.2) mostra os resultados finais quanto a utilização do critério 2a.

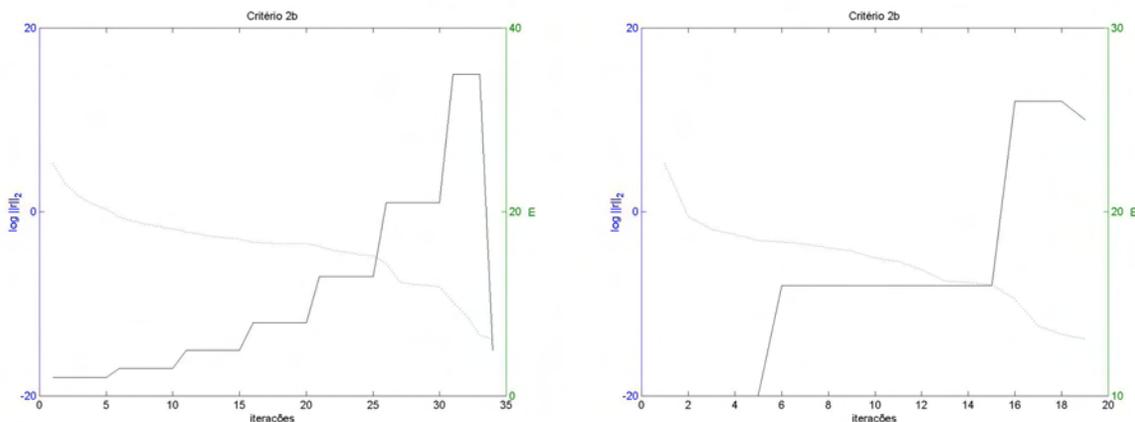
Através da observação dos gráficos, e com o objetivo de minimizar a norma residual mais rapidamente, fizemos então algumas alterações nesse critério, ou seja, no

Total	Adaptativos	Não adaptativos	Não convergiram
98	29	51	18

Tabela 3.2: Critério 2a

passo (4.) do critério 2 tomamos $m = \min(2m, mmax)$, com isso estamos dobrando o valor de m e esperamos que o $\log \|r\|_2$ decrescesse duas vezes mais rápido do que com o m anterior.

Veja na figura (3.4).

Figura 3.4: *DIGSE_2*: a) $m = 2$ e b) $m = 10$

Note que tanto para $m = 2$ quanto para $m = 10$ os gráficos não apresentaram diferença significativa; o que nos faz concluir que a mudança apenas no valor atribuído a m não faz com que a norma residual decresça mais rapidamente. O número de flops apresentados por tal teste são aproximadamente os mesmos apresentados acima, para o critério do *DIGSE_1*.

A tabela (3.3) mostra os resultados finais quanto a utilização do critério 2b.

Total	Adaptativos	Não adaptativos	Não convergiram
98	28	54	16

Tabela 3.3: Critério 2b

Na tabela (3.4) comparamos o critério 2a e 2b.

Total	Adaptativos	Não adaptativos	Não convergiram
98	29	51	18
98	28	54	16

Tabela 3.4: Conclusões finais quanto ao critério 2

Note que todas as alterações propostas nesse critério não resultaram satisfatórias em um decréscimo do logaritmo da norma residual, tendo em vista que, em geral, os métodos não adaptativos apresentaram melhores resultados do que os adaptativos. Sendo assim optou-se por utilizar outro conceito matemático, a saber: o logaritmo, para tentar alcançar o objetivo desse trabalho.

3.5 Critério 3

No terceiro critério descartamos o uso do DIGSE em favor do $\log\|r\|_2$, com base nos gráficos exibidos anteriormente, e sabendo-se que o logaritmo proporciona uma melhor visualização do comportamento do resíduo.

Note que até o momento não apresentamos nenhum critério que venha a reduzir a dimensão da base, ou seja, estamos preocupados em, inicialmente, descobrir um critério que reduza o logaritmo da norma residual adequadamente, para então melhorá-lo, fazendo com que o m possa também sofrer decréscimos.

Critério 3:

1. Se $\text{mod}(i, 5) = 0$
2. Se $\log\|r\|_2 \geq 0$
3. $m = \min(2m, m_{max})$
4. fim se
5. se $\log\|r\|_2 \leq 0$ e $\log\|r\|_2 > -4$
6. $m = \min(m + m, m_{max})$
7. caso contrário
8. $m = \min(m + \frac{m_{orig}}{2}, m_{max})$
9. fim se

10. fim se

Note que novamente no passo (1.) definimos que a norma não poderia sofrer alterações por, no máximo, cinco iterações. A diferença desse critério em relação ao anterior é que, neste, dividimos o intervalo de variação de $\log \|r\|_2$, $[\log(tol), +\infty)$ em três sub-intervalos: $(0, +\infty)$, $[-4, 0]$ e $[\log(tol), -4]$ e aumentam m de forma diferente em cada sub-intervalo, sendo que o maior acréscimo deu-se no primeiro sub-intervalo e o menor no terceiro.

Veja a figura (3.5).

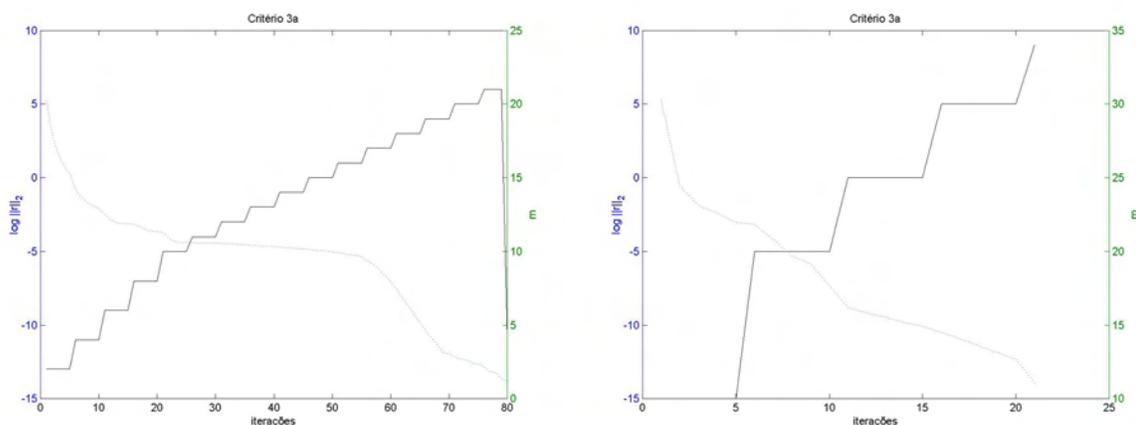


Figura 3.5: Logaritmo.1: a) $m = 2$ e b) $m = 10$

Pode-se observar na figura (3.5) que para ambos os valores de m o método convergiu, sendo que para $m = 10$ a convergência foi mais rápida, necessitando aproximadamente 10 iterações a menos do que para $m = 2$. Note também que, para $m = 10$, a partir de $i = 10$ houve um decréscimo muito rápido do logaritmo da norma residual enquanto que, para $m = 2$, o mesmo só ocorreu quando $i = 20$. Quanto ao m , observe que para $m = 2$, $m_{max} < 40$, enquanto que para $m = 10$ $m_{max} > 40$, ou seja para um m_{orig} maior foi necessário um acréscimo maior para que a convergência fosse obtida. O número de flops obtidos nesse critério foram: $m = 10$, $1,9543 \times 10^7$; $m = 2$, $2,4156 \times 10^7$. Observe aqui que o critério adaptativo já está permitindo uma redução no número de flops, mesmo utilizando uma base maior, o que não aconteceria com m fixo ao longo das iterações.

A tabela (3.5) mostra os resultados finais quanto a utilização do critério 3a.

Total	Adaptativos	Não adaptativos	Não convergiram
115	46	52	17

Tabela 3.5: Critério 3a

No mesmo critério, optamos por fazer uma segunda modificação e ver o comportamento do mesmo; substituímos no passo (8.) $m = \min(m + \frac{2m_{orig}}{3}, mmax)$ e no passo (6.) $m = \min(m + m, mmax)$. Observe os gráficos na figura (3.6).

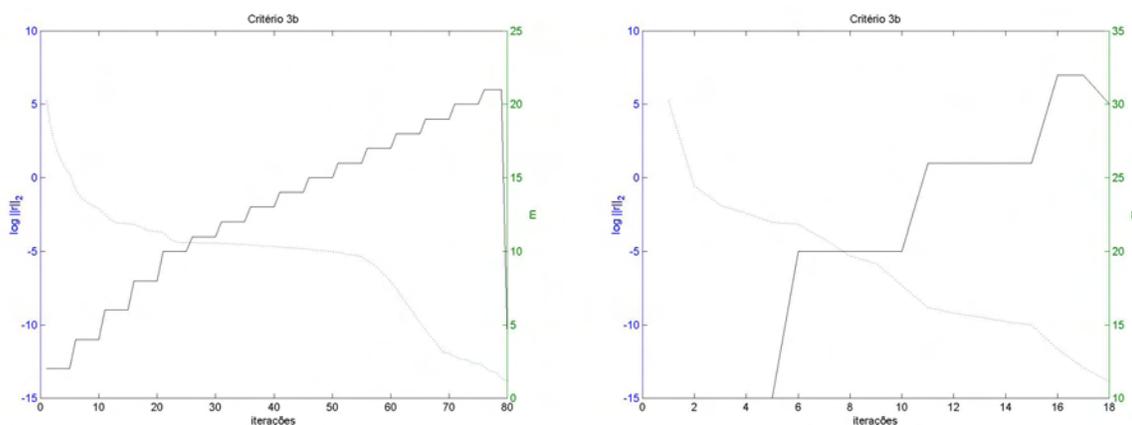


Figura 3.6: Logaritmo.2: a) $m = 2$ e b) $m = 10$

Observe que para $m = 2$ o método convergiu em aproximadamente 35 iterações, sendo que até $i = 20$ o mesmo não apresentou um decréscimo significativo no logaritmo da norma residual, somente para $i > 20$ tal decréscimo começou a ocorrer, sendo necessário mais 20 iterações para obter a convergência. Veja que o m atingiu seu valor máximo como sendo maior do que 40, novamente é importante salientar que o decréscimo apresentado para o m no gráfico não deu-se pelo critério utilizado, e sim pelo processo de Arnoldi, devido à detecção de uma base invariante.

Para $m = 10$ a convergência do método ocorreu em aproximadamente 19 iterações, sendo que para $i > 10$ houve um decréscimo acentuado no logaritmo da norma residual. Podemos concluir através desses dois gráficos que a escolha do m_{orig} influencia em quanto irá decrescer o $\log ||r||_2$. Quanto ao m , neste caso foi necessário que o $mmax > 40$, conforme o exemplo anterior.

A tabela (3.6) mostra os resultados finais quanto a utilização do critério 3b.

Total	Adaptativos	Não adaptativos	Não convergiram
115	52	45	18

Tabela 3.6: Critério 3b

Nossa terceira alteração desse critério foi a: tomamos o passo (8.) como no critério original e o passo (6.) sendo $m = \min(m + m, mmax)$, desejando analisar se essa modificação na escolha do m acarreta em alguma diferença significativa no decréscimo do $\log \|r\|_2$. Observe a figura (3.7).

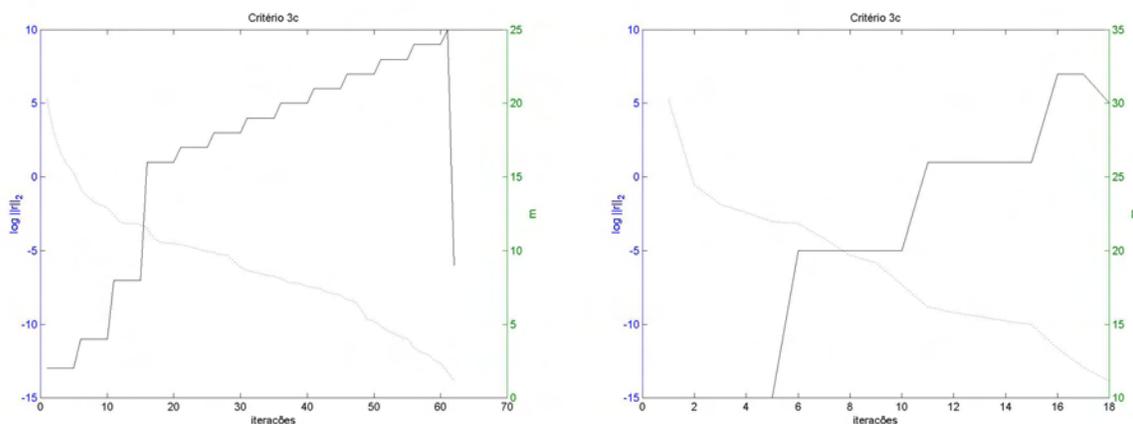


Figura 3.7: Logaritmo_3: a) $m = 2$ e b) $m = 10$

Observando os gráficos nota-se que esse último critério não proporciona nenhuma melhora significativa em relação aos critérios anteriores; uma prova disso seria que para $m = 10$ foram necessárias mais de 20 iterações para a convergência enquanto que no critério anterior o mesmo se deu em aproximadamente 18 iterações. O m também apresentou um comportamento similar.

Concluimos que dentre as modificações realizadas o melhor critério ainda seria o segundo, visto que satisfaz o logaritmo da norma residual sendo extremamente pequeno, necessitando de um número menor de iterações para obter a convergência e apresenta um número menor de flops.

O número de flops para a segunda e terceira tentativa foram praticamente os mesmos, ou seja, para $m = 2$ obtivemos $3,4452 \times 10^7$ e para $m = 10$ obtivemos $5,3808 \times 10^7$.

A tabela (3.7) mostra os resultados finais quanto a utilização do critério 3c.

Total	Adaptativos	Não adaptativos	Não convergiram
115	45	54	16

Tabela 3.7: Critério 3c

Na tabela (3.8), comparamos as três alterações propostas no critério 3.

Total	Adaptativos	Não adaptativos	Não convergiram
115	46	52	17
115	52	45	18
115	45	54	16

Tabela 3.8: Conclusões finais quanto ao critério 3

Observe que dentre as variações propostas no critério 3 a mais satisfatória é a do critério 3b, onde o número de convergência para os métodos adaptativos superou a dos métodos não adaptativos, além do número de não convergências ter sido pequeno.

Podemos concluir com isso que a melhor opção para o critério que estamos buscando com esse trabalho seria envolvendo o conceito de logaritmo. Assim, o próximo critério utiliza tal conceito e tende a apresentar uma forma de fazer com que o m sofra acréscimos e decréscimos, de acordo com a evolução das iterações.

3.6 Critério 4

De posse dos resultados obtidos nos critérios anteriores, onde conseguimos uma redução significativa do logaritmo da norma residual, foi possível então passar ao segundo passo da elaboração do critério desejado, que visa modificar a dimensão do subespaço de Krylov de acordo com a necessidade do sistema, ou seja, desenvolver um critério que possa sofrer acréscimos e decréscimos. Neste critério, utilizamos o $\log \|r\|_2$ e uma nova variação na escolha do m .

Critério 4a:

1. Se $\text{mod}(i, 5) = 0$
2. Se $\log \|r\|_2 > 0$

3. $m = \min(m + m_{orig}, m_{max});$
4. caso contrário se $\log \|r\|_2 \leq 0$ e $\log \|r\|_2 > \frac{2 \log tol}{3}$
5. se $i > 5$ e $\frac{\|r_{i-5}\|_2}{\|r_i\|_2} > 2$
6. $m = \max(m_{orig}, m - \frac{m_{orig}}{3})$
7. caso contrário
8. $m = \min(m + \frac{2m_{orig}}{3}, m_{max});$
9. fim se
10. caso contrário
11. se $i > 5$ e $\frac{\|r_{i-5}\|_2}{\|r_i\|_2} > 2$
12. $m = \max(m_{orig}, m - \frac{m_{orig}}{4});$
13. caso contrário
14. $m = \min(m + \frac{m_{orig}}{3}, m_{max})$
15. fim se
16. fim se
17. $\|r_{i-5}\|_2 = \|r\|_2$
18. fim se

O critério é simples, sendo que apresentamos uma explicação mais detalhada dos seus passos, como segue: o passo (1.) é um teste que não permite que a norma seja alterada antes de fechar um ciclo de cinco iterações, como já foi visto nos demais critérios; o passo (2.) é um teste para detectar se o resíduo é muito alto na iteração em questão, em caso afirmativo então aumenta-se a base; o passo (4.) verifica se $\log \|r\|_2$ está nos $\frac{2}{3}$ anteriores do intervalo de variação do $\log \|r\|_2$ então, pelo passo (5.), se a taxa de redução está boa diminui-se a base; caso contrário aumenta-se, conforme passo (8.). O passo (10.) avalia $\|r_{i-5}\|_2$, ou seja, avalia se o módulo da norma depois de cinco iterações; se a taxa de redução está boa, então diminui-se a base, caso contrário, pelo passo (14.) aumenta-se a mesma. Os valores atribuídos a m são diferentes nos passos (3.),(6.),(8.),(12.) e (14.). Note que esse critério possui a propriedade de aumentar e diminuir a base do subespaço.

Observe a figura (3.8).

No gráfico exibido na figura (3.8), com $m = 2$, a visualização da variação do m não é clara, visto que o m atinge seu valor máximo em menos de 10 iterações, contudo pode-se notar que o logaritmo da norma residual apresentou apenas momentos

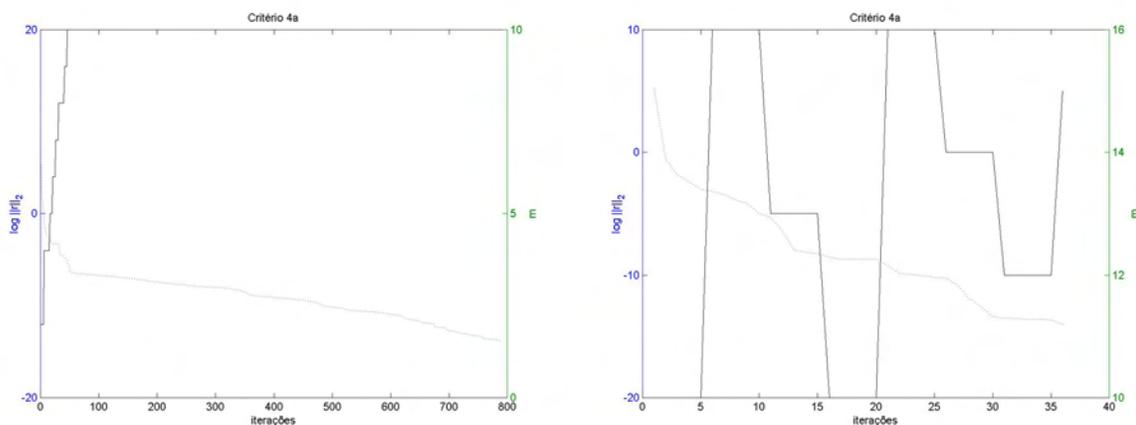


Figura 3.8: Adaptativo-1 a) $m = 2$ e b) $m = 10$

de “quase” estagnação, necessitando esse de aproximadamente 1200 iterações para obter a convergência dentro da tolerância pré-especificada.

Para $m = 10$, a visualização da variação proposta ao m por esse critério é extremamente boa, note que $10 \leq m \leq 20$, sendo que os decréscimos e acréscimos foram muitos ao longo das iterações necessárias para obter a convergência, satisfazendo assim a proposta de escolha de m que se adequasse ao sistema que se deseja resolver. A convergência não estagnou ao longo de todo o processo, sempre esteve decrescendo, comprovando assim que a escolha do $\log \|r\|_2$ dentre as demais testadas é a mais adequada. Observe ainda que $\log \|r\|_2$ aproximou-se significativamente do número de máquina, ou seja, de 10^{-16} mostrando assim que a convergência foi extremamente significativa. O número de flops obtidos nesse critério foi para $m = 2$ obtivemos $2,9801 \times 10^7$ e para $m = 10$ obtivemos $2,0956 \times 10^7$.

Note que os flops do critério 4a apresentaram-se um pouco mais elevados do que os flops apresentados pelo critério 3, contudo é importante salientar que este último avalia a necessidade do sistema, ou seja, analisa se o m precisa ser acrescido ou decrescido, conforme a nossa proposta inicial.

A tabela (3.9) mostra os resultados obtidos quanto a utilização do critério 4a, alocado no método $\text{GMRES}(m)$.

Total	Adaptativos	Não adaptativos	Não convergiram
115	59	50	6

Tabela 3.9: Critério 4a

Podemos notar pela tabela (3.9) que o número de convergência dos métodos adaptativos superou a dos não adaptativos, no entanto desejamos que tal diferença fosse mais significativa, visando isso fizemos algumas modificações nos passos (3.), (8.) e (13.). Apresentamos o critério com tais modificações, o qual chamamos de *Critério 4b*.

Critério 4b:

1. Se $\text{mod}(i, 5) = 0$
2. Se $\log \|r\|_2 > 0$
3. $m = \min(2m, m_{\max})$
4. caso contrário se $\log \|r\|_2 \leq 0$ e $\log \|r\|_2 > \frac{2 \log \text{tol}}{3}$
5. se $i > 5$ e $\frac{\|r_{i-5}\|_2}{\|r_i\|_2} > 2$
6. $m = \max(m_{\text{orig}}, m - \frac{m_{\text{orig}}}{3})$
7. caso contrário
8. $m = \min(m + m_{\text{orig}}, m_{\max})$
9. fim se
10. caso contrário
11. se $i > 5$ e $\frac{\|r_{i-5}\|_2}{\|r_i\|_2} > 2$
12. $m = \max(m_{\text{orig}}, m - \frac{m_{\text{orig}}}{4})$
13. caso contrário
14. $m = \min(m + \frac{m_{\text{orig}}}{2}, m_{\max})$
15. fim se
16. fim se
17. $\|r_{i-5}\|_2 = \|r\|_2$
18. fim se

Novamente plotamos os gráficos com tal critério alocado no método GMRES(m) com a matriz CDDE1. Veja a figura (3.9).

Os gráficos do critério 4a são muito similares aos gráficos do critério 4b, apresentando apenas pequenas diferenças quanto ao número de iterações necessárias a

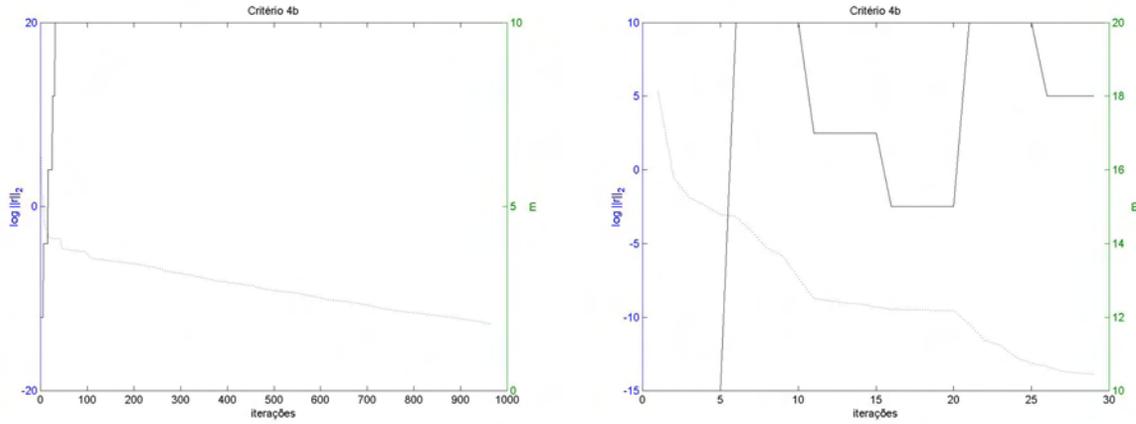


Figura 3.9: Adaptativo a) $m = 2$ e b) $m = 10$

convergência e também ao comportamento da curva do logaritmo da norma residual. O número de flops apresentados foram praticamente os mesmos.

A tabela (3.10) mostra os resultados finais quanto a utilização do critério 4b.

Total	Adaptativos	Não adaptativos	Não convergiram
115	69	41	5

Tabela 3.10: Critério 4b

Observe a diferença significativa entre o número de convergências dos métodos adaptativos e dos métodos não adaptativos. Note que o número de não convergências foi extremamente pequeno.

Observe a tabela (3.11), nela é apresentada uma comparação entre os valores apresentados pelos critério 4a e 4b.

Total	Adaptativos	Não adaptativos	Não convergiram
115	59	50	6
115	69	41	5

Tabela 3.11: Conclusões finais quanto ao critério 4

Note que entre o critério 4a e o critérios 4b houve diferenças significativas, as quais nos levam a concluir que o critério 4b é o mais satisfatório, e é o critério procurado nesse trabalho.

No próximo capítulo apresentamos os testes realizados com o método GMRES(m) e suas variantes, comparando os mesmos com as versões adaptativos e não adaptativas.

4 RESULTADOS EXPERIMENTAIS

Nesse capítulo apresentamos os testes realizados, utilizando o software MATLAB versões 5.3 e 6.2 em um computador PC com processador da classe AMD Athlon XP 2.400 com 2 GHz e 512 de memória RAM; descrevemos as matrizes utilizadas e, por fim, resumizamos os resultados obtidos.

Os testes foram realizados com as matrizes: ADD32, ARC_130, CAVITY05, CDDE1, E05R0000, E05R0300, FIDAPM05, FS_183_1, FS_760_1, GRE115, IMPCOLA, JPWH991, NOS3, ORSIRR1, PDE900, SAYLR3, SHERMAN1 e STEAM1, presentes na coleção “Matrix Market” [MCSD/ITL/NIST, 2003].

As matrizes ADD32, CAVITY05 e SHERMAN1 possuem termo independente próprio, os quais utilizamos nos testes; para as demais matrizes utilizamos um mesmo termo independente, cujos elementos são valores aleatórios gerados segundo uma distribuição normal com média zero e desvio padrão um, obtidos através da função “RANDN” do MATLAB. A fim de garantir o uso dos mesmos valores, o gerador aleatório é reinicializado com a semente inicial pré-definida no MATLAB, antes de cada teste.

Os algoritmos implementados para a realização dos testes foram os seguintes: GMRES(m), GMRESH(m), SGMRES(m), SGMRESH(m), LGMRES(m, k) com $k = 2$ e $k = 3$, BC-GMRES(m) e A-GMRES(m). Além deles, utilizamos as versões adaptativas com o critério 4, conforme especificado no capítulo 3, os quais denotamos por: GMRES-A(m), GMRESH-A(m), SGMRES-A(m), SGMRESH-A(m), LGMRES-A(m, k) com $k = 2$ e $k = 3$ e BC-GMRES-A(m).

Os métodos serão comparados como segue: SGMRES-A(m) com SGMRES(m); SGMRESH-A(m) com SGMRESH(m); BC-GMRES(m) com BC-GMRES-A(m); LGMRES-A(m, k) com LGMRES(m, k) com os respectivos k e, por fim, o A-GMRES(m) com GMRESH(m) visto que estes dois utilizam a formulação do GMRES(m) com base nas transformações de Householder.

Os métodos utilizados para a realização dos testes nesse capítulo foram empregados sem pré-condicionadores e, utilizou-se como critério de parada a seguinte desigualdade: $|r_i|_2 < \epsilon$, sendo $\epsilon = 10^{-6}$.

Por fim, salienta-se que o eixo horizontal, o qual representa as iterações, apresenta escalas diferentes em cada método.

4.1 Experimentos Realizados

4.1.1 ADD32

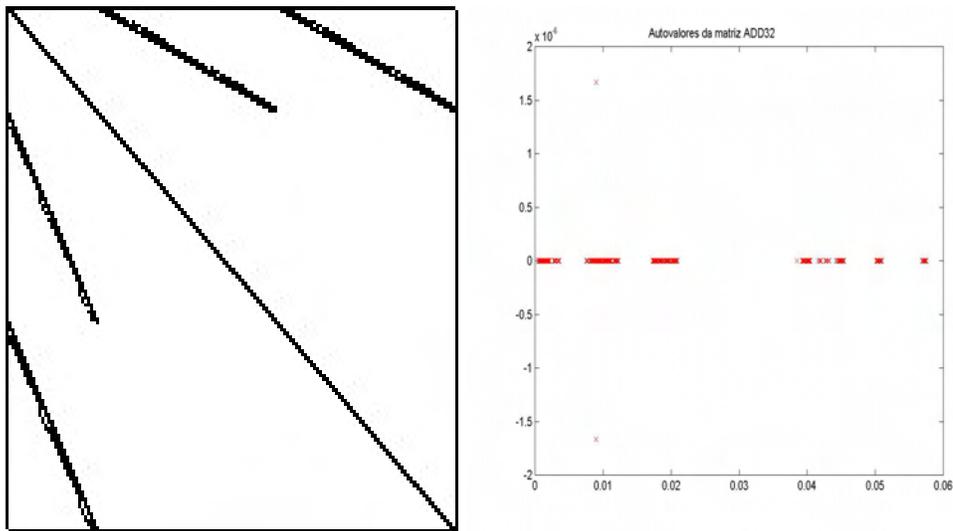


Figura 4.1: ADD32: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 4960×4960
- tipo: real não-simétrica
- entradas não nulas: 19848
- número de condição (estimado): 214
- $\|A\|_2$ estimado: 0,057

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	9	8	10	$5,3464 \times 10^7$
GMRESH-A	sim	6	10	20	$4,9860 \times 10^7$
SGMRES	sim	7	7	10	$2,6169 \times 10^7$
SGMRES-A	sim	8	10	20	$2,3629 \times 10^7$
SGMRESH	sim	7	9	10	$2,7264 \times 10^7$
SGMRESH-A	sim	6	10	13	$2,6102 \times 10^7$
BC-GMRES	sim	9	1	10	$1,3088 \times 10^7$
BC-GMRES-A	sim	12	1	20	$1,3055 \times 10^7$
LGMRES($m,2$)	sim	6	10	10	$1,3832 \times 10^7$
LGMRES-A($m,2$)	sim	6	10	16	$1,7506 \times 10^7$
LGMRES($m,3$)	sim	6	10	10	1.4214×10^7
LGMRES-A($m,3$)	sim	6	10	16	1.8087×10^7

Tabela 4.1: ADD32

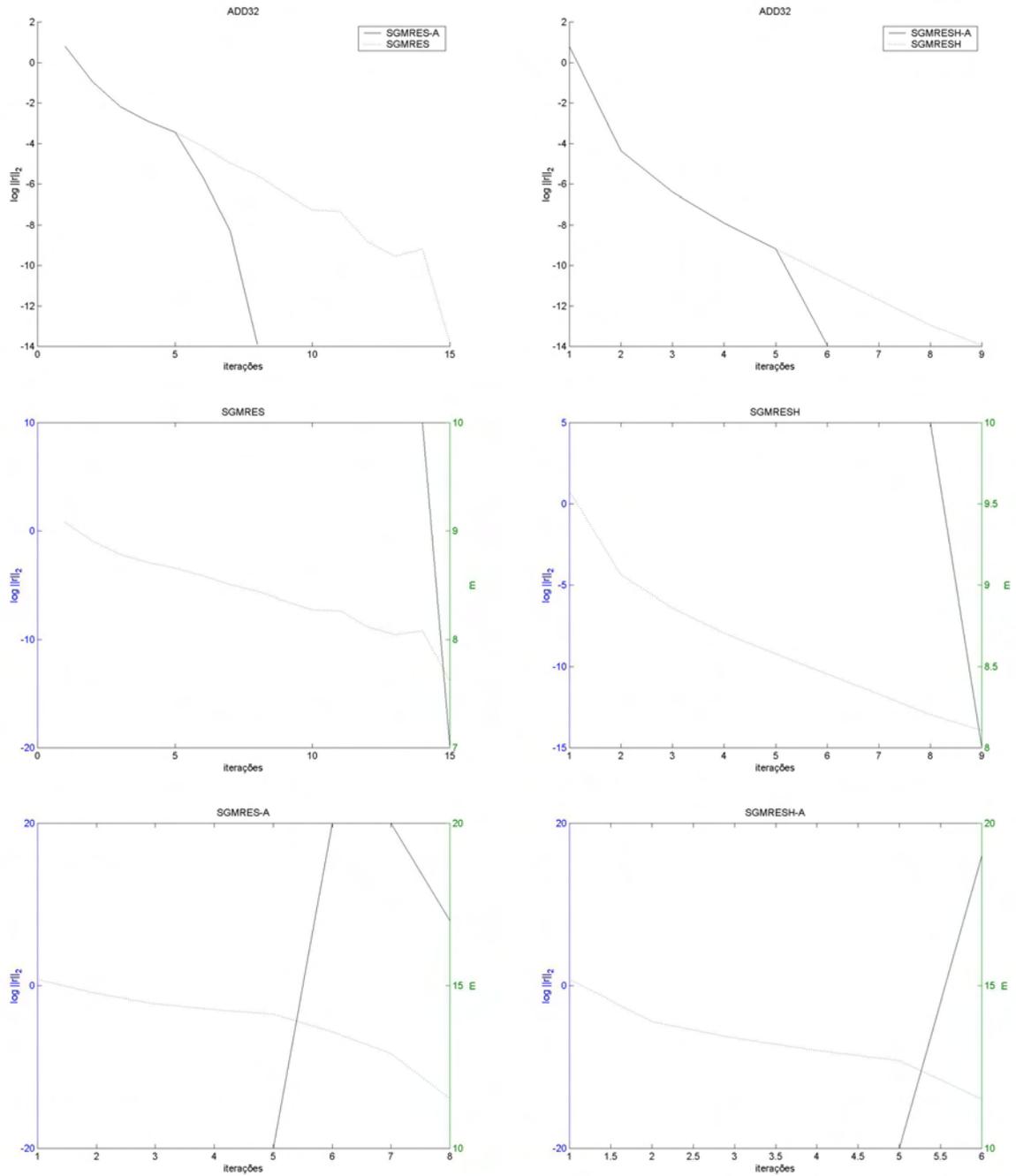


Figura 4.2: ADD32: Gráfico do resíduo e m versus iterações

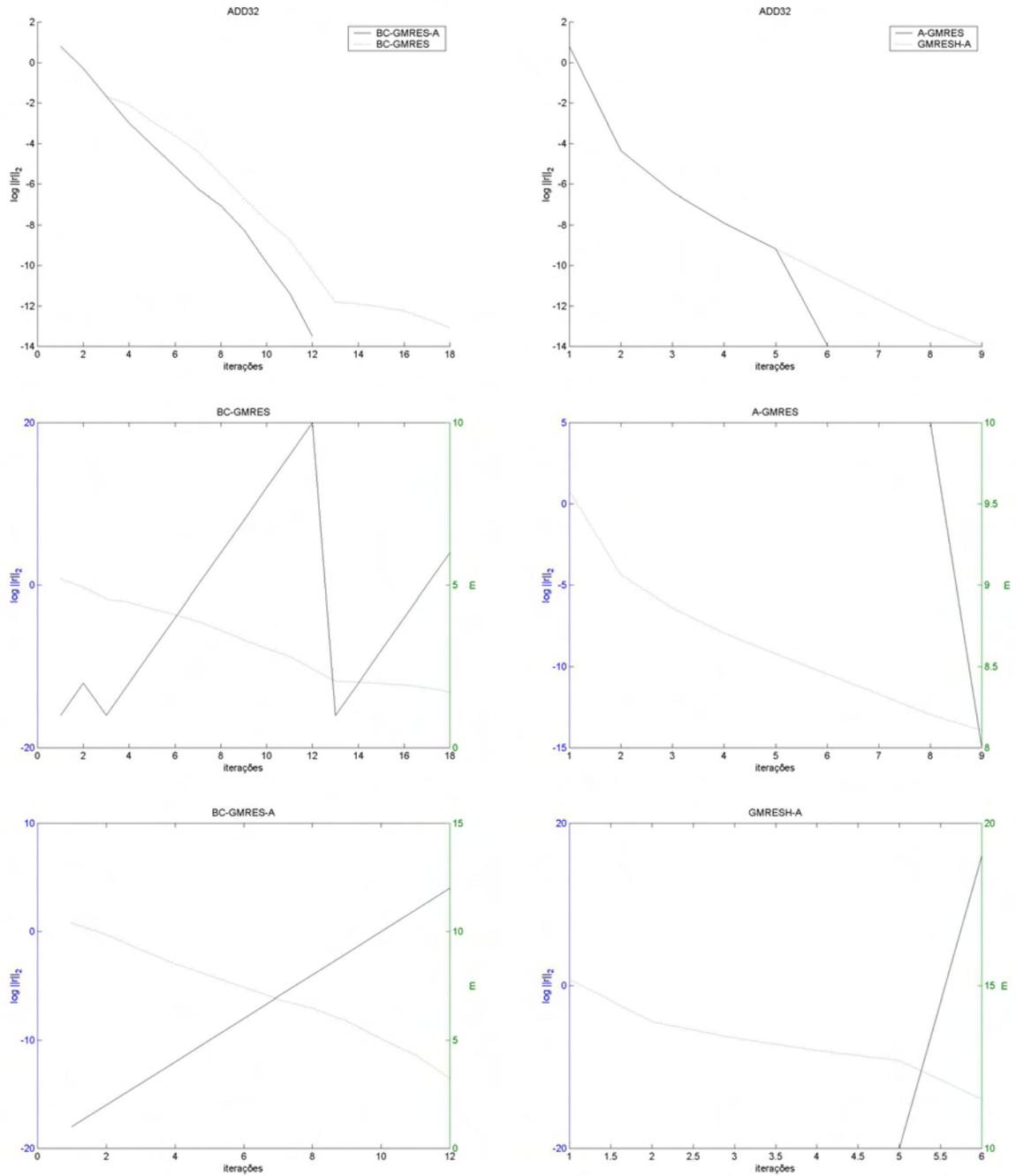


Figura 4.3: ADD32: Gráfico do resíduo e m versus iterações

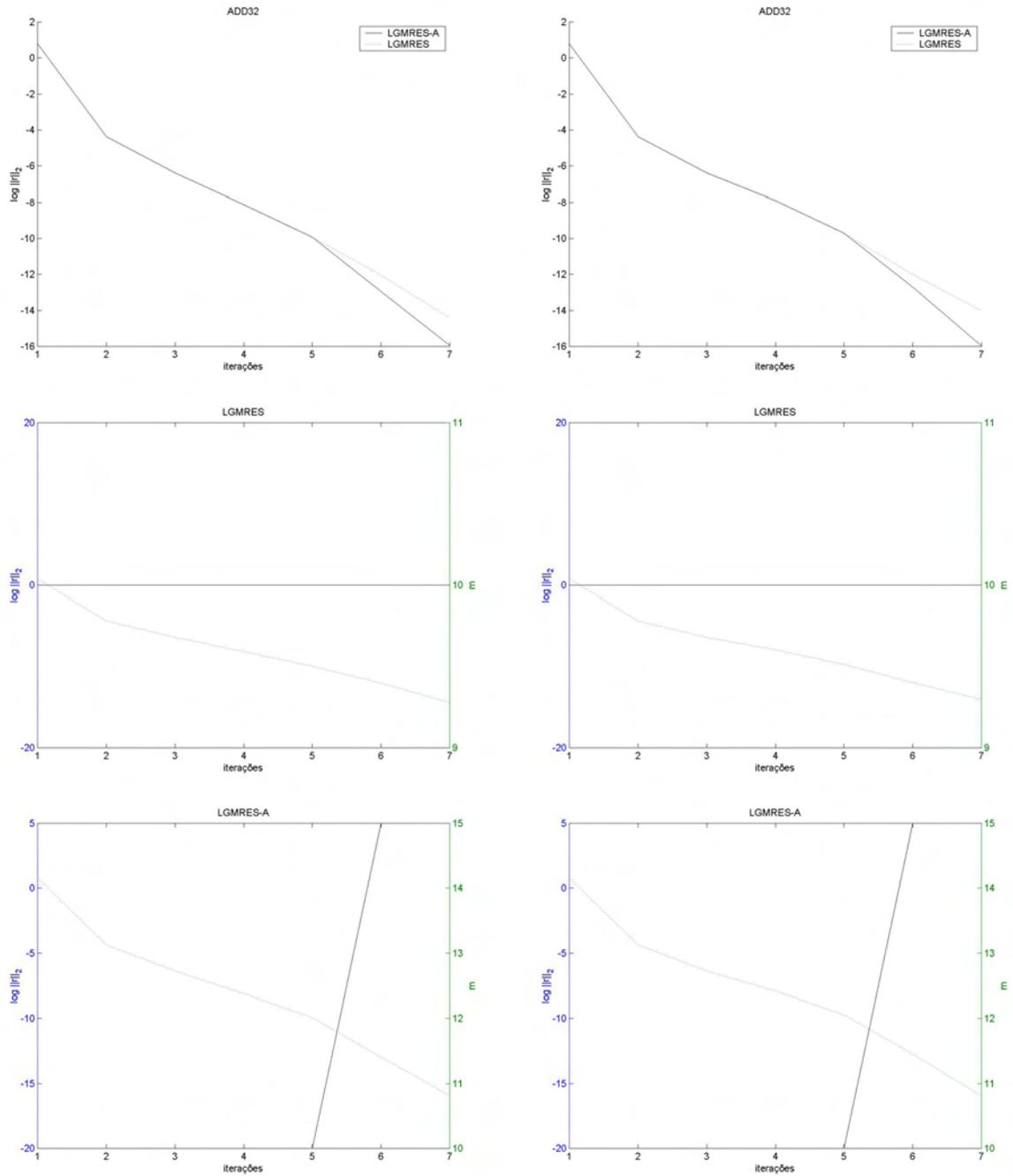


Figura 4.4: ADD32: Gráfico do resíduo e m versus iterações

Note que na figura (4.2) ambos os métodos convergiram, contudo o SGMRES-A e o SGMRESH-A apresentaram menor números de flops e obtiveram a convergência com um número menor de iterações. Note que o decréscimo apresentado na variação do m , tanto no SGMRES como no SGMRESH são devido ao fato de ter sido detectado um subespaço invariante na formação do subespaço de Krylov.

Na tabela (4.1) observe que o BC-GMRES-A apresentou menor número de flops. Note que no gráfico do BC-GMRES o m apresenta “picos”, e esses são devidos a própria estrutura do método, ou seja, quando o logaritmo da norma residual começa a decrescer de forma lenta, o método é reinicializado, sendo que essa reinicialização sempre ocorre com $m = 1$, e devido a isso ocorre tal formação de “picos”. Esse comportamento de “picos” no BC-GMRES é esperado via a sua formulação; veja [Nodera e Moriya, 2003].

Observe a figura (4.3), note que ambos os métodos convergiram, contudo o GMRESH-A apresentou menor número de flops e necessitou de menos iterações para obter a convergência dentro da tolerância pré-especificada do que o A-GMRES; veja [Driver, 1997].

Note na figura (4.4) que ambos os métodos convergiram e apresentaram comportamento similar para $k = 2$ e $k = 3$; observe através da tabela (4.1) que o LGMRES($m, 2$) e o LGMRES($m, 3$) apresentaram menor número de flops do que o LGMRES-A($m, 2$) e o LGMRES-A($m, 3$), respectivamente.

4.1.2 ARC_130

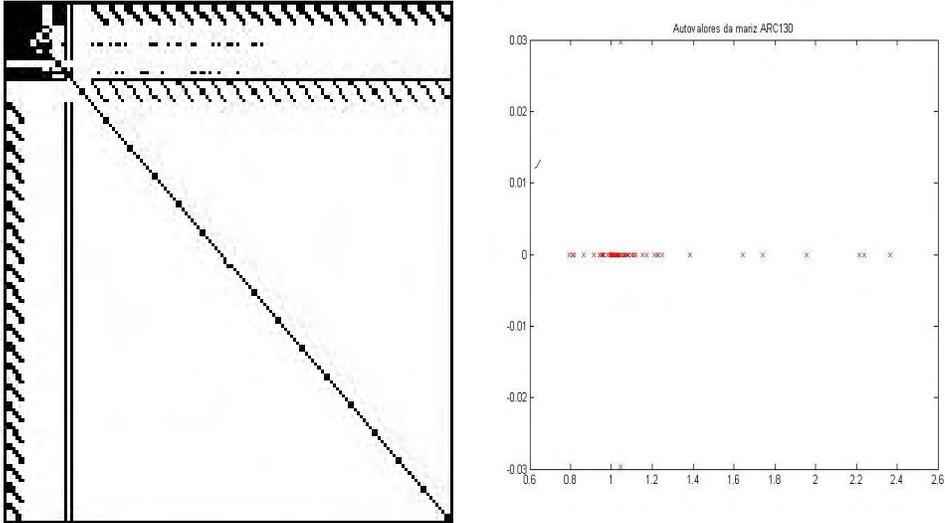


Figura 4.5: ARC_130: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 130×130
- tipo: real não-simétrica
- entradas não nulas: 1037
- número de condição (estimado): 11
- $\|A\|_2$ estimado: $2,4 \times 10^5$

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	3	7	10	3032
GMRESH-A	sim	3	7	10	3031
SGMRES	sim	3	9	10	1121
SGMRES-A	sim	3	9	10	1121
SGMRESH	sim	3	7	10	1906
SGMRESH-A	sim	3	7	10	1906
BC-GMRES	sim	40	1	10	5983
BC-GMRES-A	sim	8	1	10	2028
LGMRES($m,2$)	sim	2	10	10	1255
LGMRES-A($m,2$)	sim	2	10	10	1255
LGMRES($m,3$)	sim	2	10	10	1255
LGMRES-A($m,3$)	sim	2	10	10	1255

Tabela 4.2: ARC_130

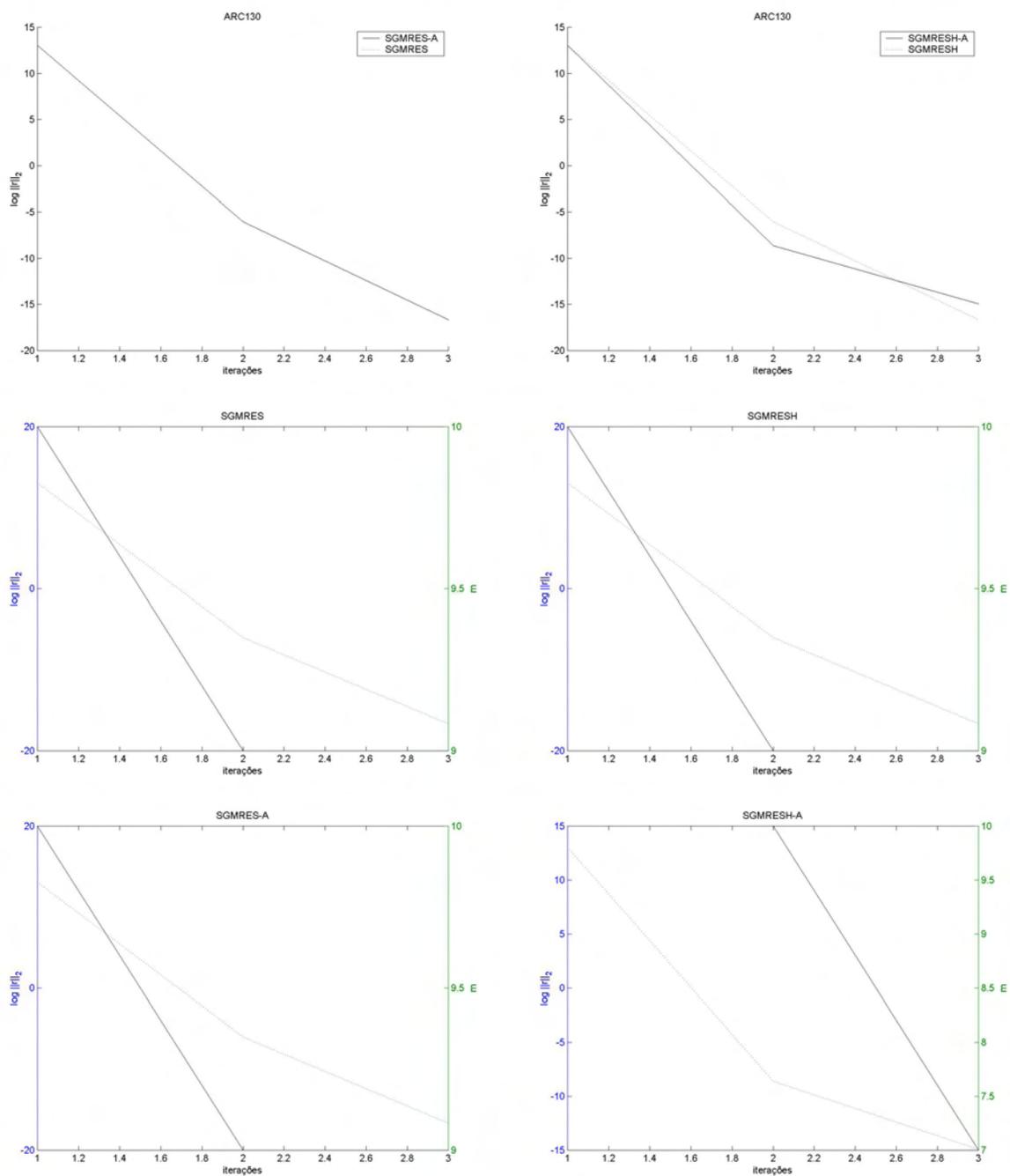


Figura 4.6: ARC_130: Gráfico do resíduo e m versus iterações

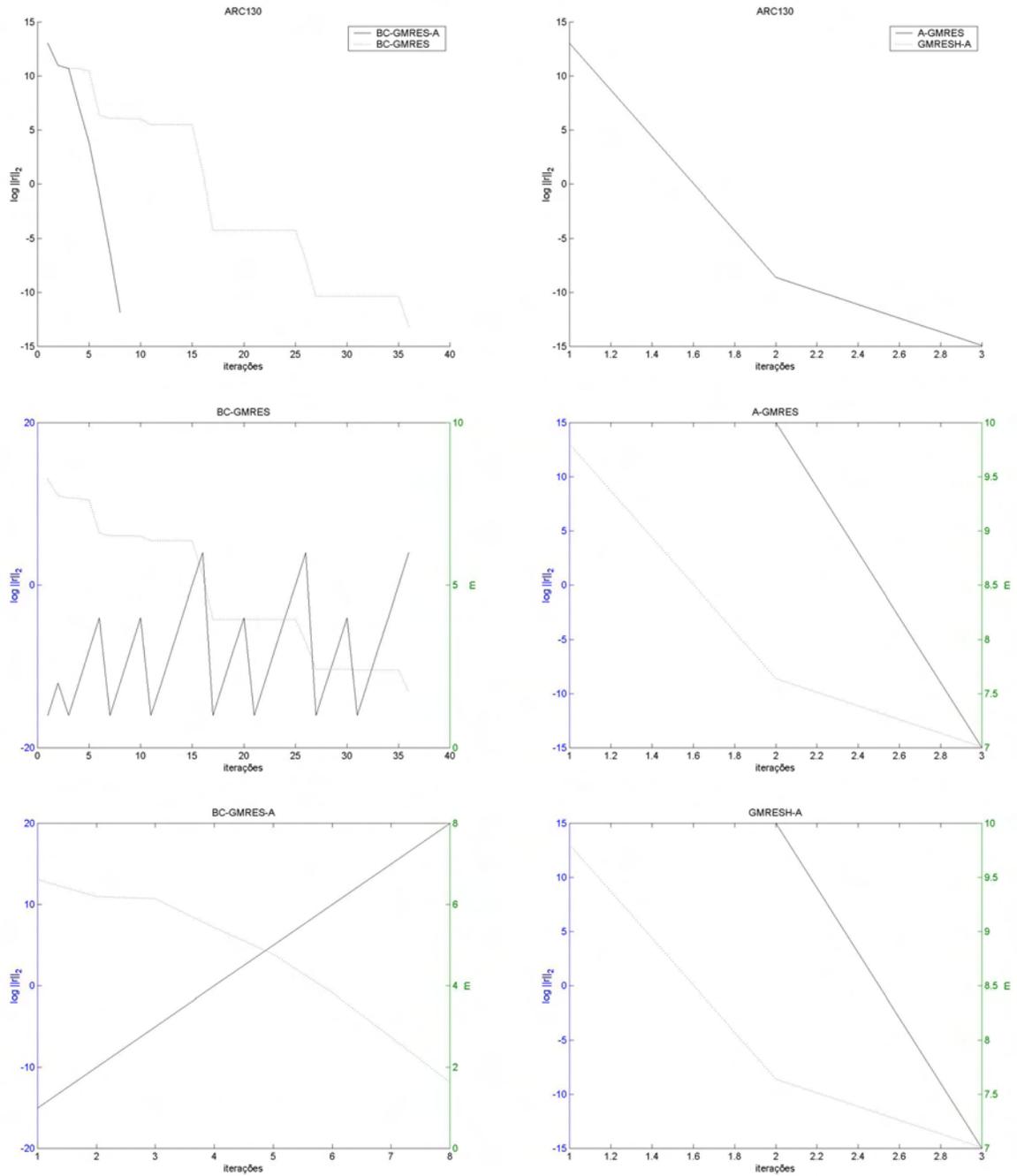


Figura 4.7: ARC_130: Gráfico do resíduo e m versus iterações

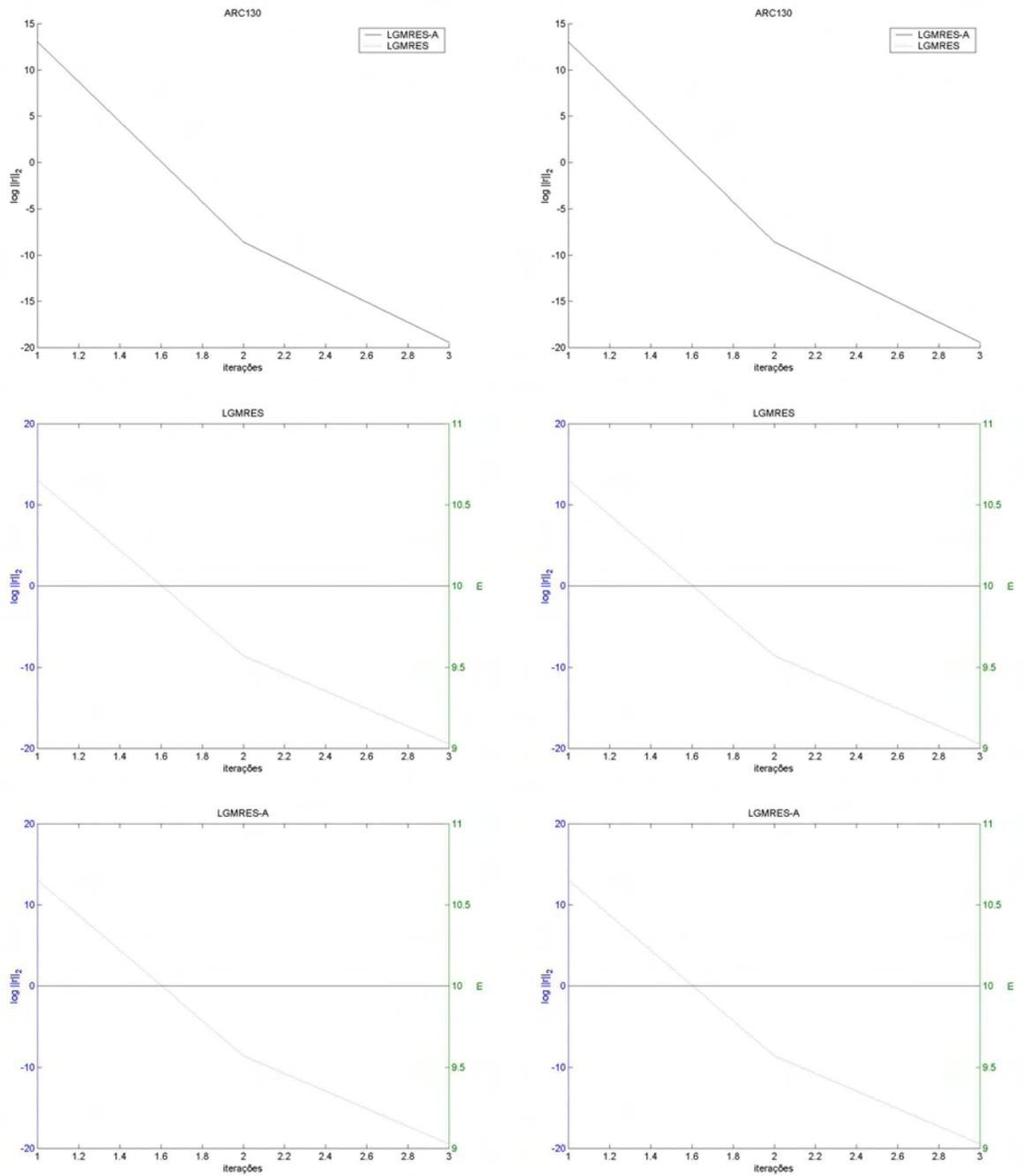


Figura 4.8: ARC_130: Gráfico do resíduo e m versus iterações

Observe na figura (4.6) que ambos os métodos apresentaram comportamento similar e convergiram; sendo que o SGMRES-A apresentou menor número de flops e necessitou de menos iterações para obter a convergência do que o SGMRES. O SGMRESH e o SGMRESH-A apresentaram o mesmo número de flops e o mesmo número de iterações; desta forma nada pode-se concluir em relação a esses métodos.

Na figura (4.7) note que ambos os métodos convergiram, no entanto o BC-GMRES apresentou menor número de flops do que o BC-GMRES-A. Note, novamente, que quando o logaritmo da norma residual começa a decrescer vagarosamente o método é reinicializado, ou seja, cada “quase” degrau formado na curva do logaritmo residual corresponde a um “pico” na curva da variação do m . Assim, note que os “picos” representam as reinicializações sofridas por tal método.

Observe que o GMRESH-A e o A-GMRES apresentaram comportamento similar, sendo que o GMRESH-A apresentou menor número de flops do que o A-GMRES.

Observando a figura (4.8) pode-se notar que, novamente, o LGMRES(m, k) e o LGMRES-A(m, k), para $k = 2$ e $k = 3$, apresentaram o mesmo comportamento e o mesmo número de flops para resolver tal sistema. Podemos concluir que para esse sistema o acréscimo do critério 4 no LGMRES não apresentou diferença significativa.

4.1.3 CAVITY05

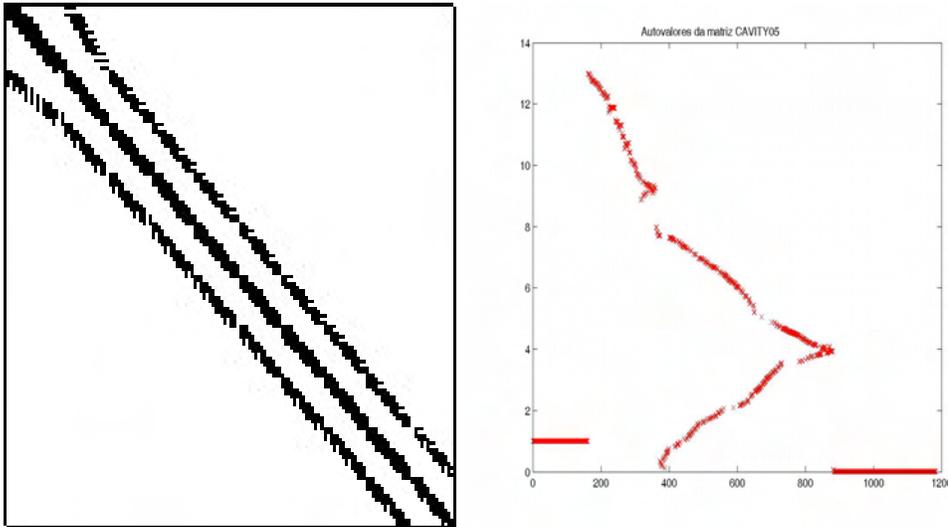


Figura 4.9: CAVITY05: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 1182×1182
- tipo: real não-simétrica
- entradas não nulas: 32632
- número de condição (estimado): Não há
- $\|A\|_2$ estimado: 13

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	1200	9	11	$3,6399 \times 10^9$
GMRESH-A	sim	140	10	60	$2,6787 \times 10^9$
SGMRES	sim	1200	9	11	$1,2652 \times 10^9$
SGMRES-A	sim	120	10	60	$8,3960 \times 10^8$
SGMRESH	sim	300	10	10	$4,7326 \times 10^8$
SGMRESH-A	sim	45	10	40	$3,4098 \times 10^8$
BC-GMRES	sim	1200	1	10	$6,9471 \times 10^8$
BC-GMRES-A	sim	200	1	60	$6,6960 \times 10^8$
LGMRES($m,2$)	sim	70	10	10	$8,0490 \times 10^7$
LGMRES-A($m,2$)	sim	40	10	25	$8,0837 \times 10^7$
LGMRES($m,3$)	sim	80	10	10	$1,2414 \times 10^7$
LGMRES-A($m,3$)	sim	50	10	25	$1,0856 \times 10^7$

Tabela 4.3: CAVITY_05

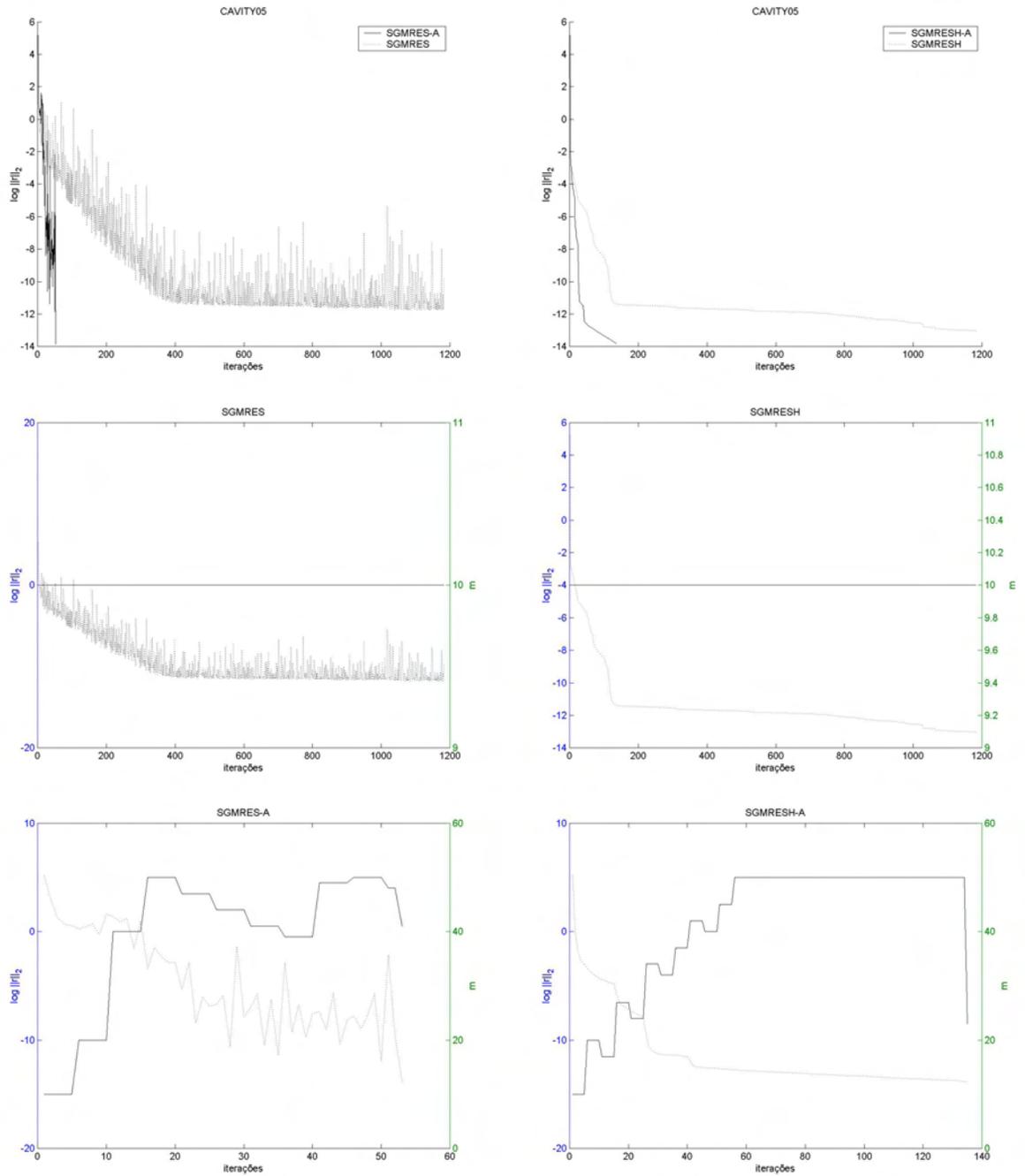


Figura 4.10: CAVITY05: Gráfico do resíduo e m versus iterações

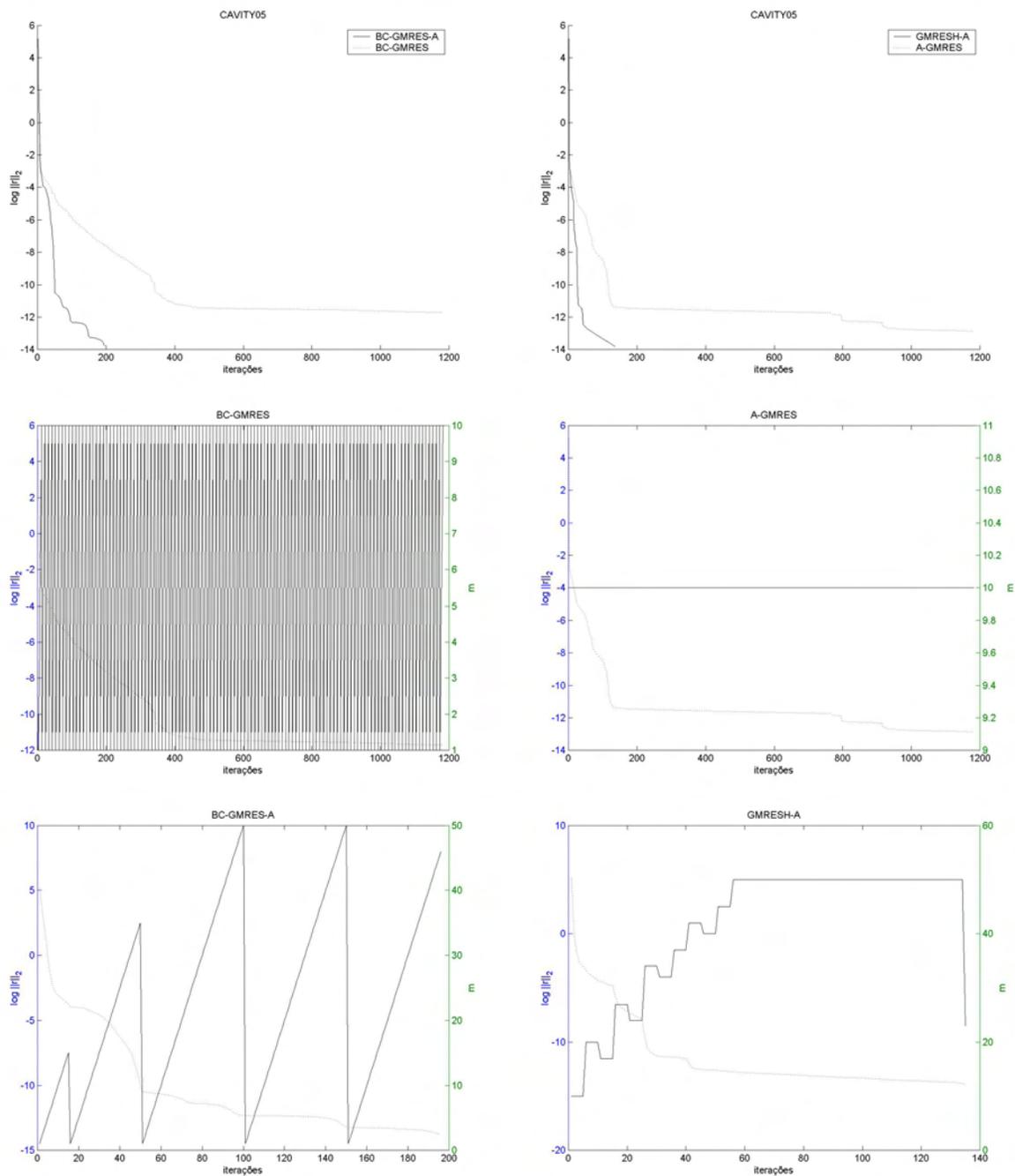


Figura 4.11: CAVITY05: Gráfico do resíduo e m versus iterações

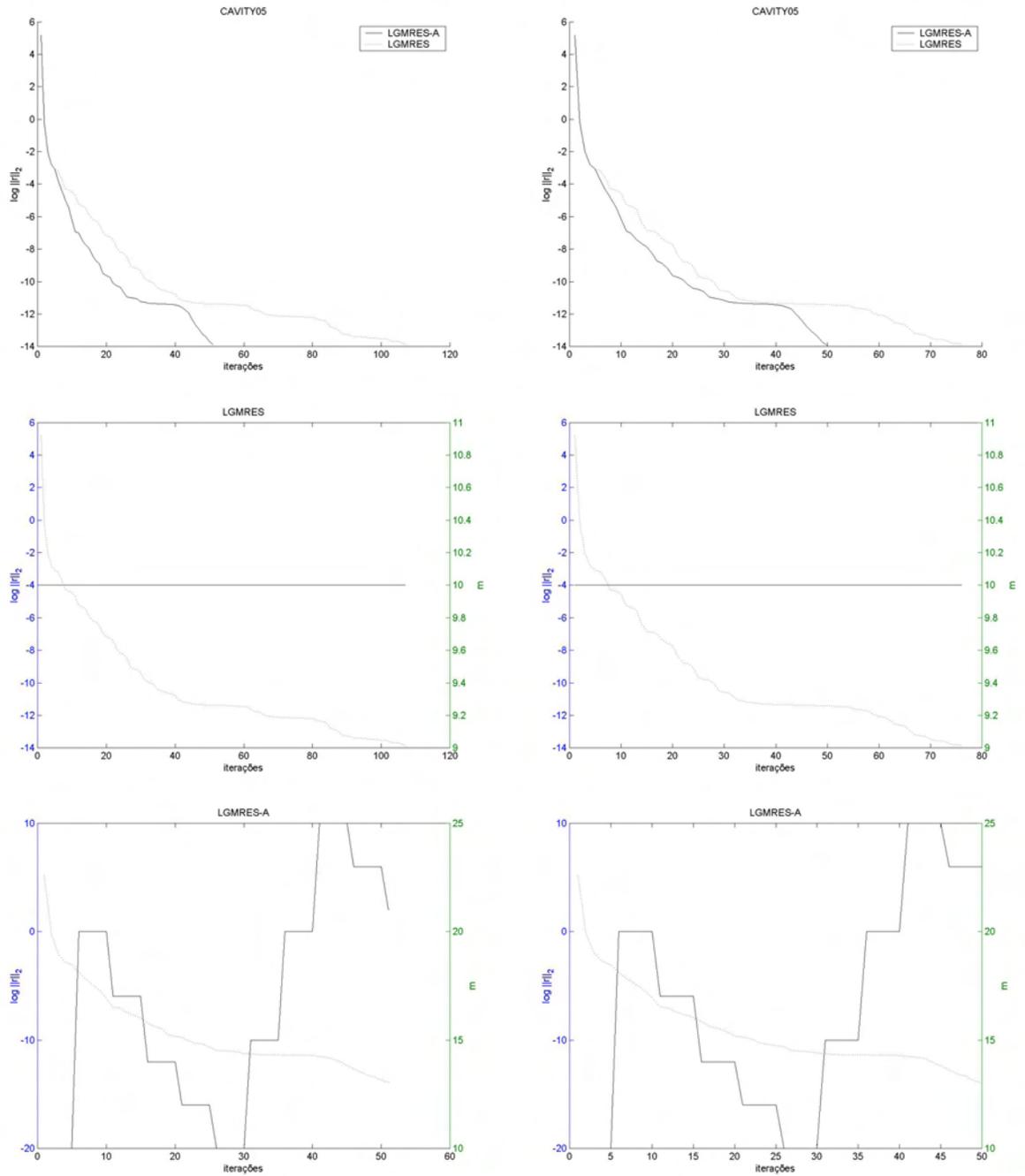


Figura 4.12: CAVITY05: Gráfico do resíduo e m versus iterações

Na figura (4.10) note que o SGMRES-A e o SGMRESH-A apresentaram menor número de flops e necessitaram de menos iterações para obter a convergência dentro da tolerância pré-especificada do que o SGMRES e o SGMRESH, respectivamente. Observe também que no SGMRES e no SGMRESH o m não apresentou variação.

Na figura (4.11) note que ambos os métodos convergiram, no entanto o BC-GMRES-A apresentou menor número de flops e de iterações para obter a convergência do que o BC-GMRES. Nesse sistema pode-se perceber facilmente as diversas reinicializações que o BC-GMRES sofre, enquanto que o BC-GMRES-A sofreu aproximadamente cinco apenas.

Observe que o GMRESH-A apresentou menor número de flops e de iterações do que o A-GMRES. Note também que no A-GMRES o m não apresentou variação.

Na figura (4.12) ambos os métodos convergiram, sendo o LGMRES($m, 2$) e o LGMRES-A($m, 3$) apresentaram menor número de flops do que o LGMRES-A($m, 2$) e o LGMRES($m, 3$), respectivamente. Observe que o m no LGMRES(m, k) não apresentou variação.

4.1.4 CDDE1

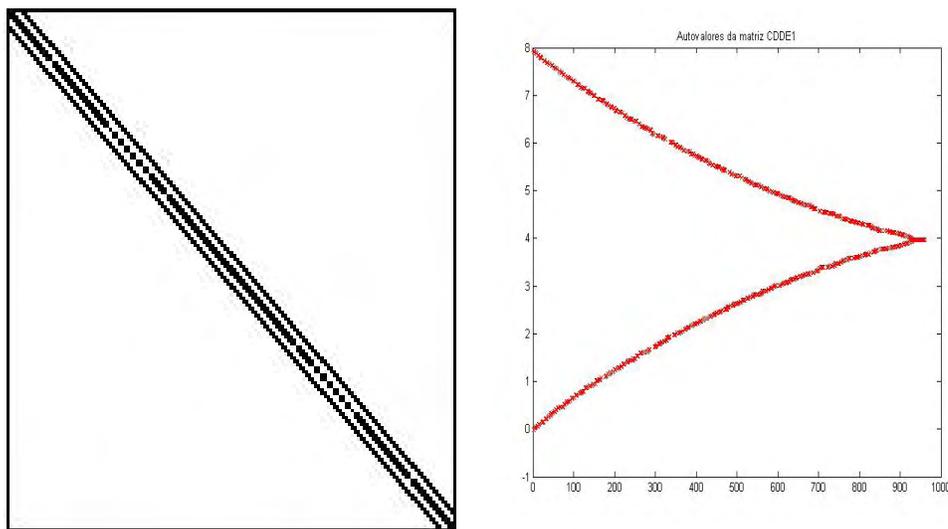


Figura 4.13: CDDE1: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 961×961
- tipo: real não-simétrica
- entradas não nulas: 4681
- número de condição (estimado): $4,1 \times 10^3$
- $\|A\|_2$ estimado: 7,9

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	400	3	10	$5,0780 \times 10^8$
GMRESH-A	sim	30	2	20	$9,5554 \times 10^7$
SGMRES	sim	1000	7	10	$3,5790 \times 10^8$
SGMRES-A	sim	30	10	40	$4,2707 \times 10^7$
SGMRESH	sim	400	2	10	$3,2589 \times 10^8$
SGMRESH-A	sim	25	10	20	$4,4876 \times 10^7$
BC-GMRES	sim	1000	1	10	$1,8685 \times 10^8$
BC-GMRES-A	sim	70	1	60	$3,8236 \times 10^7$
LGMRES($m,2$)	sim	35	10	10	$1,6624 \times 10^7$
LGMRES-A($m,2$)	sim	14	10	20	$1,4284 \times 10^7$
LGMRES($m,3$)	sim	45	10	10	$3,9552 \times 10^7$
LGMRES-A($m,3$)	sim	40	10	20	$2,3118 \times 10^7$

Tabela 4.4: CDDE1

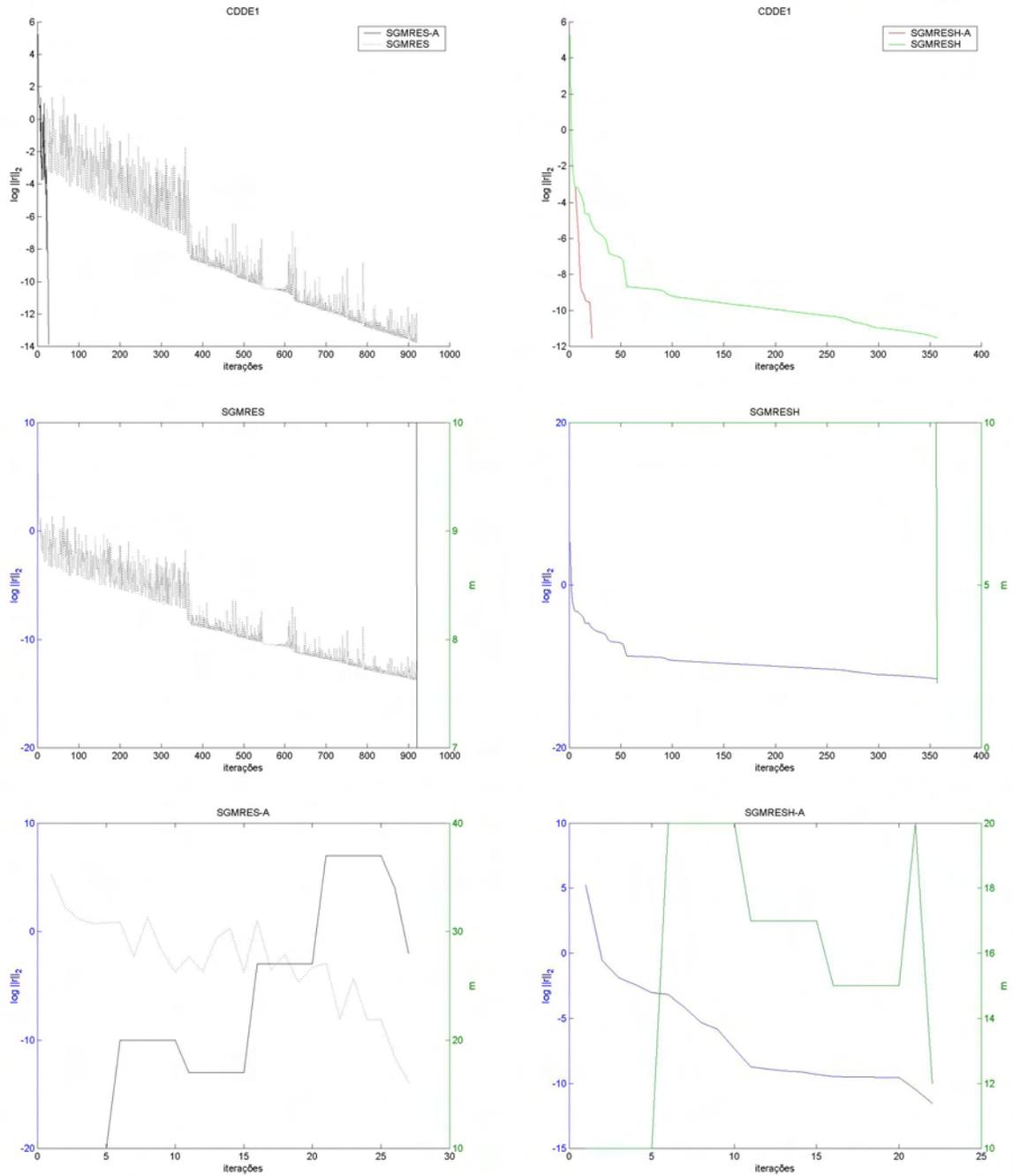


Figura 4.14: CDDE1: Gráfico do resíduo e m versus iterações

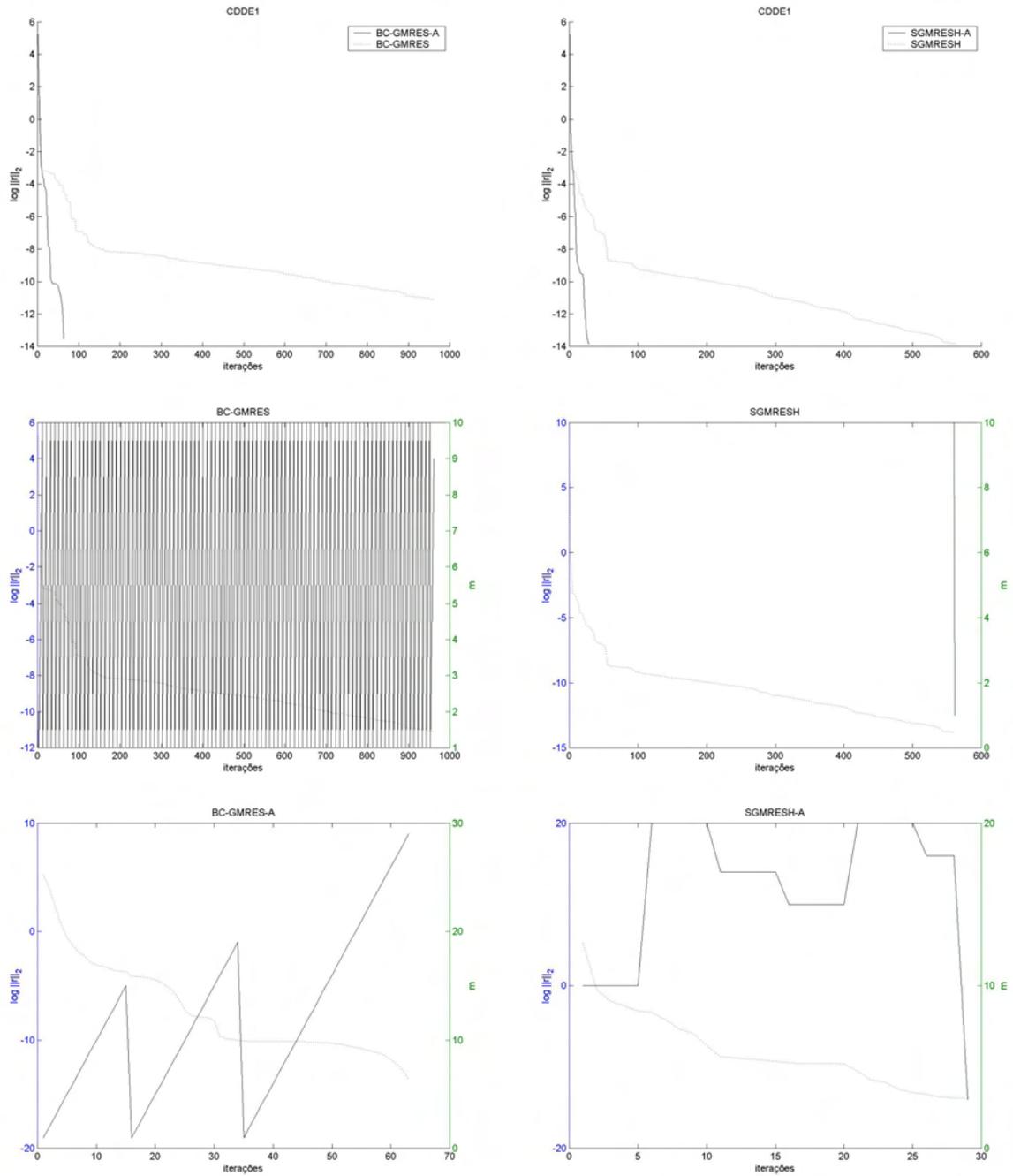


Figura 4.15: CDDE1: Gráfico do resíduo e m versus iterações

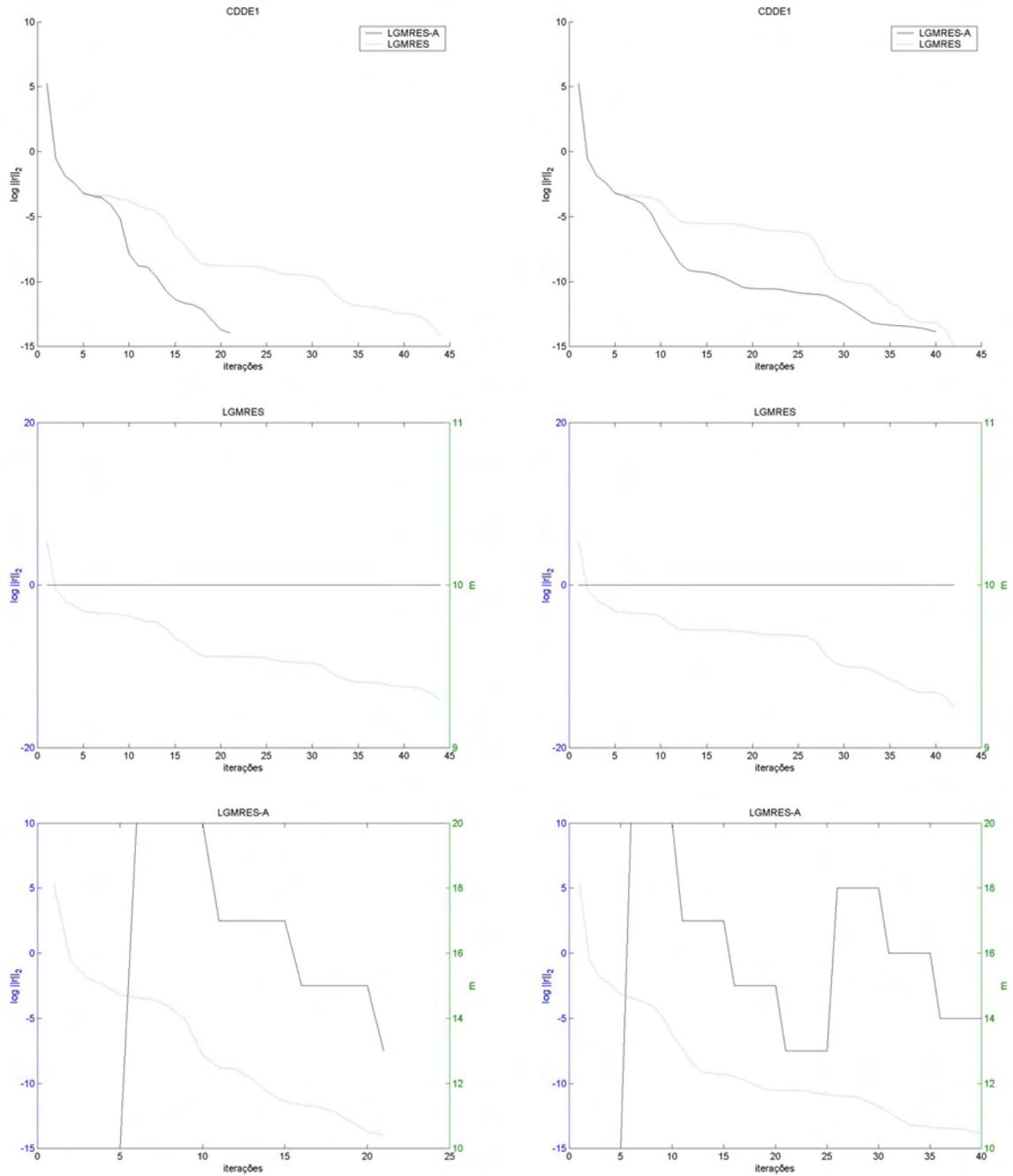


Figura 4.16: CDDE1: Gráfico do resíduo e m versus iterações

Observe a figura (4.14), note que ambos os métodos convergiram, contudo o SGMRES-A e o SGMRESH-A apresentaram menor número de flops e necessitaram de menos iterações para obter a convergência. Destacamos, novamente, que o decréscimo do m apresentado no SGMRES e no SGMRESH deu-se pela detecção de um subespaço invariante na formação do subespaço de Krylov.

Na tabela (4.4) pode-se notar que o BC-GMRES-A apresentou menor número de flops do que o BC-GMRES. Observe novamente o fato mencionado acima referente às reinicializações.

Entre o A-GMRES e o GMRESH-A, note que ambos os métodos convergiram, sendo que o GMRESH-A apresentou menor número de flops e de iterações necessárias para obter a convergência dentro da tolerância pré-especificada.

Na figura (4.16) ambos os métodos convergiram, sendo que o LGMRES-A(m, k), para $k = 2$ e $k = 3$, apresentaram menor número de flops do que o LGMRES(m, k) para os mesmos k . Observe que o m não apresentou variação no LGMRES(m, k).

4.1.5 E05R0000

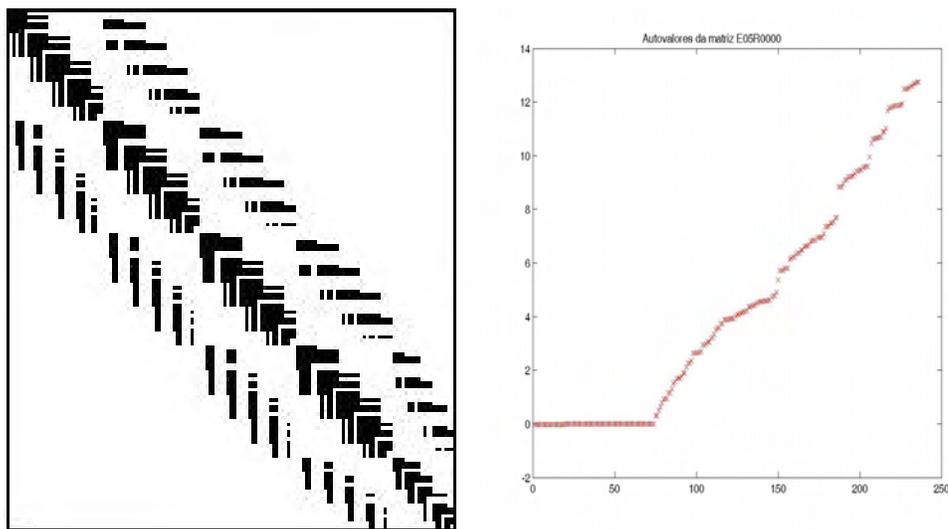


Figura 4.17: E05R0000: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 236×236
- tipo: real não-simétrica
- entradas não nulas: 5846
- número de condição (estimado): 6×10^4
- $\|A\|_2$ estimado: 13

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	250	10	10	$1,0284 \times 10^8$
GMRESH-A	sim	40	10	35	$6,4066 \times 10^7$
SGMRES	sim	250	10	10	$4,7014 \times 10^7$
SGMRES-A	sim	35	10	60	$3,1457 \times 10^7$
SGMRESH	sim	250	10	10	$7,7193 \times 10^7$
SGMRESH-A	sim	70	10	30	$5,5226 \times 10^8$
BC-GMRES	sim	250	1	10	$2,8201 \times 10^7$
BC-GMRES-A	sim	70	1	60	$2,6023 \times 10^8$
LGMRES($m,2$)	sim	90	10	10	$2,1867 \times 10^7$
LGMRES-A($m,2$)	sim	35	10	20	$1,2765 \times 10^7$
LGMRES($m,3$)	sim	90	10	10	$2,3756 \times 10^7$
LGMRES-A($m,3$)	sim	40	10	20	$1,4704 \times 10^7$

Tabela 4.5: E05R0000

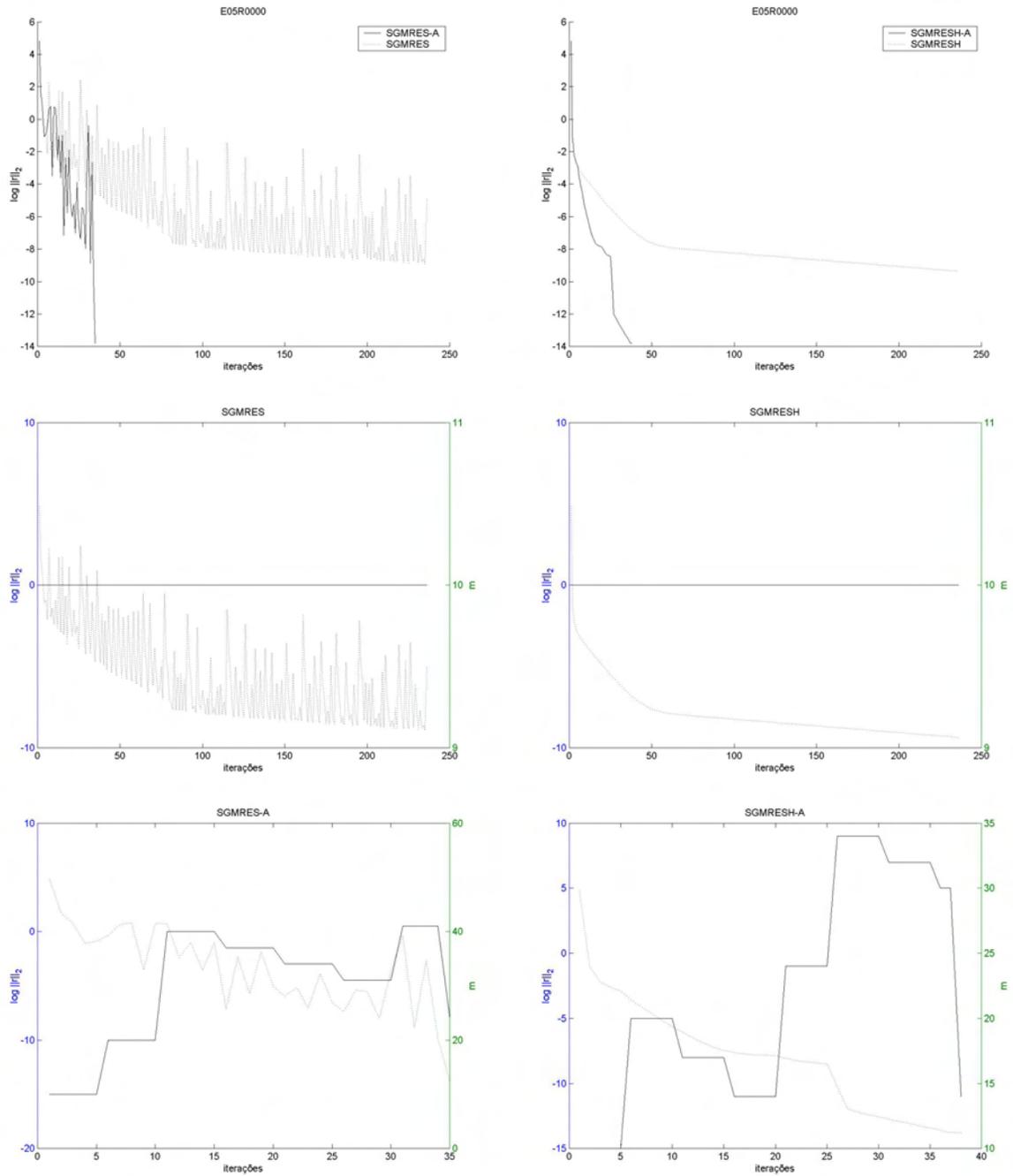


Figura 4.18: E05R0000: Gráfico do resíduo e m versus iterações

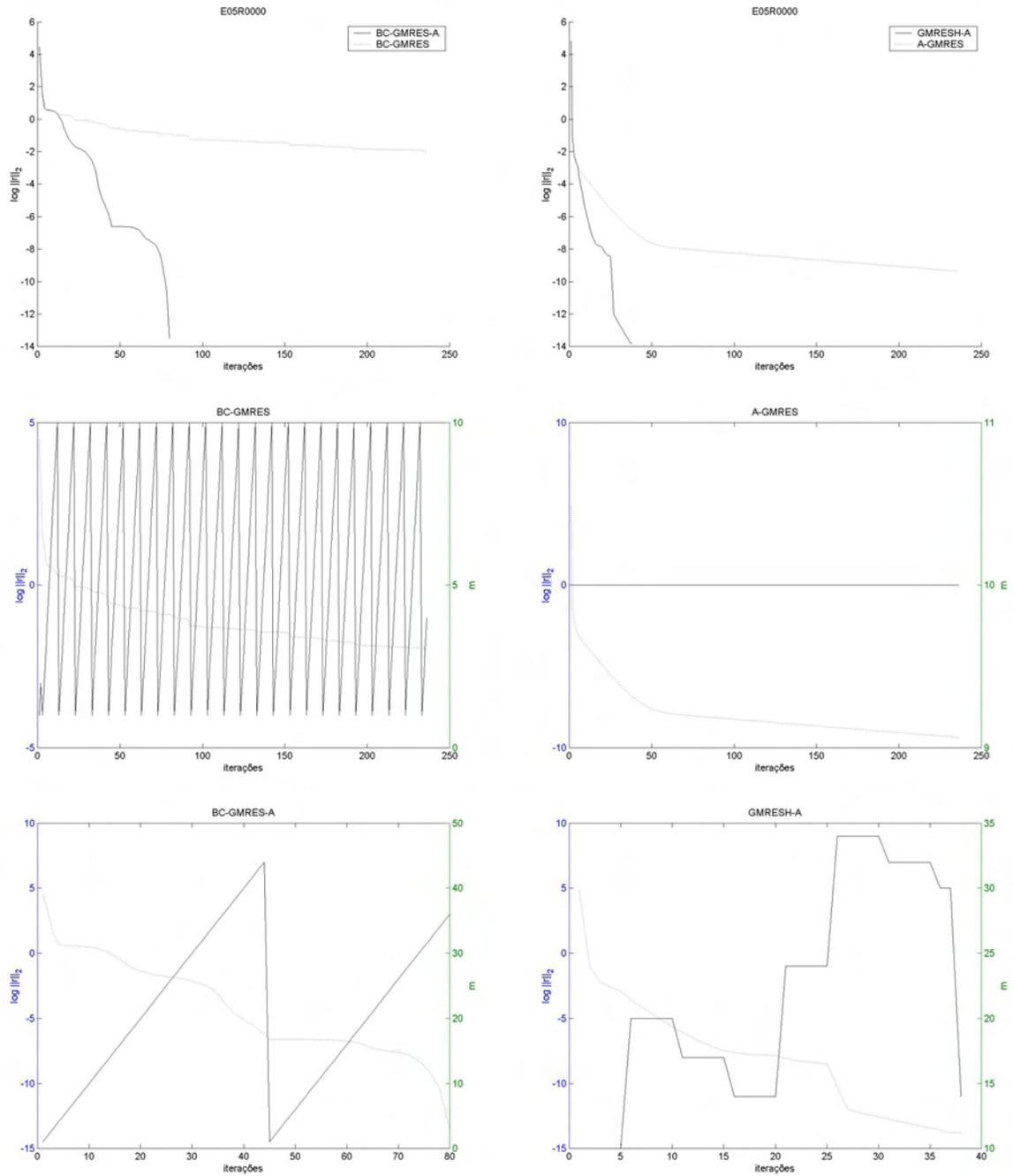


Figura 4.19: E05R0000: Gráfico do resíduo e m versus iterações

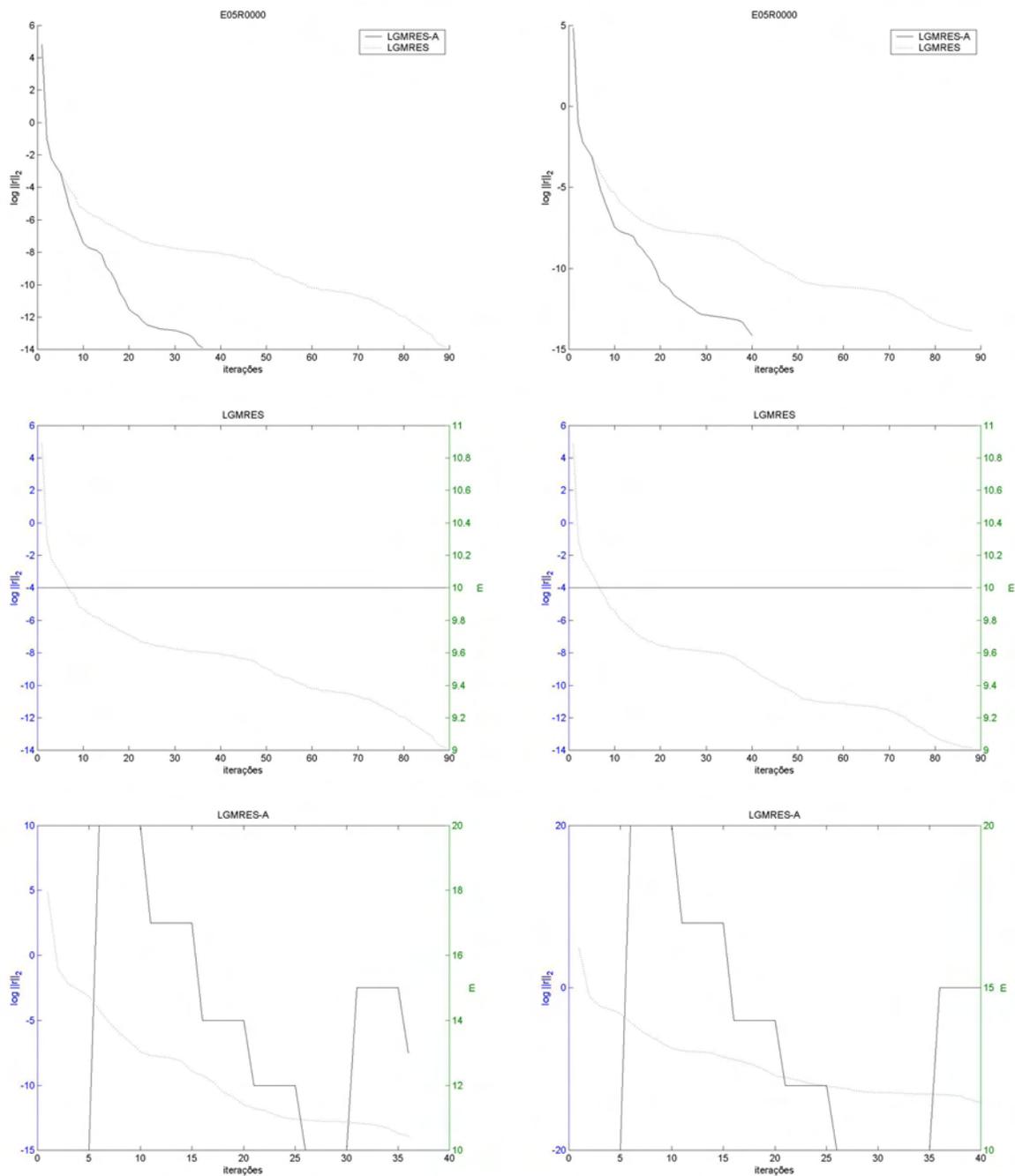


Figura 4.20: E05R0000: Gráfico do resíduo e m versus iterações

Observe a figura (4.18); note que nela ambos os métodos convergiram, sendo que o SGMRES-A e o SGMRESH apresentaram menor número de flops do que o SGMRES e o SGMRESH-A, respectivamente. O SGMRES-A também necessitou de um número menor de iterações para obter a convergência. Observe que o m não apresentou variação no SGMRES e no SGMRESH.

Na figura (4.19) ambos os métodos convergiram, sendo que o BC-GMRES e o GMRESH-A apresentaram menor número de flops e de iterações necessárias para obter a convergência. Note que o BC-GMRES necessitou de várias reinicializações a mais do que o BC-GMRES-A.

Na figura (4.20) note que ambos os métodos apresentaram comportamento similar para $k = 2$ e $k = 3$, sendo que o LGMRES-A($m, 2$) e o LGMRES-A($m, 3$) apresentaram menor número de flops do que o LGMRES($m, 2$) e o LGMRES($m, 3$), respectivamente. Observe que o m em ambos os casos comportou-se da mesma forma.

4.1.6 E05R0300

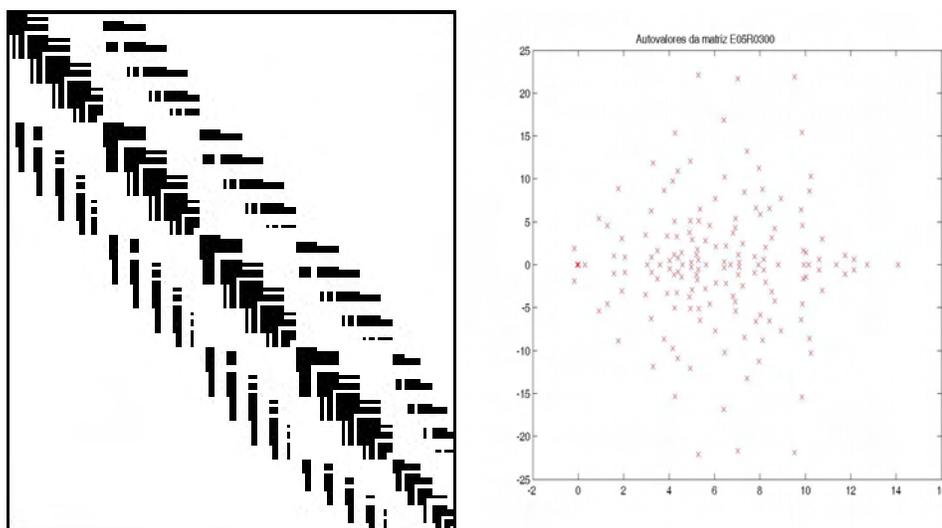


Figura 4.21: E05R0300: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 236×236
- tipo: real não-simétrica

- entradas não nulas: 5846
- número de condição (estimado): $1,3 \times 10^6$
- $\|A\|_2$ estimado: 35

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	não	-	10	50	$1,0277 \times 10^8$
GMRESH-A	não	-	10	50	$1,4516 \times 10^9$
SGMRES	não	-	10	10	$4,7014 \times 10^7$
SGMRES-A	não	-	10	50	$4,2437 \times 10^8$
SGMRESH	não	-	10	10	$7,7192 \times 10^7$
SGMRESH-A	não	-	10	50	$9,8047 \times 10^8$
BC-GMRES	não	-	1	60	$1,7429 \times 10^7$
BC-GMRES-A	não	-	1	50	$1,5275 \times 10^7$
LGMRES($m,2$)	não	-	10	10	$5,8768 \times 10^7$
LGMRES-A($m,2$)	não	-	10	50	$4,4957 \times 10^8$
LGMRES($m,3$)	não	-	10	10	$6,4748 \times 10^7$
LGMRES-A($m,3$)	não	-	10	50	$4,6415 \times 10^8$

Tabela 4.6: E05R0300

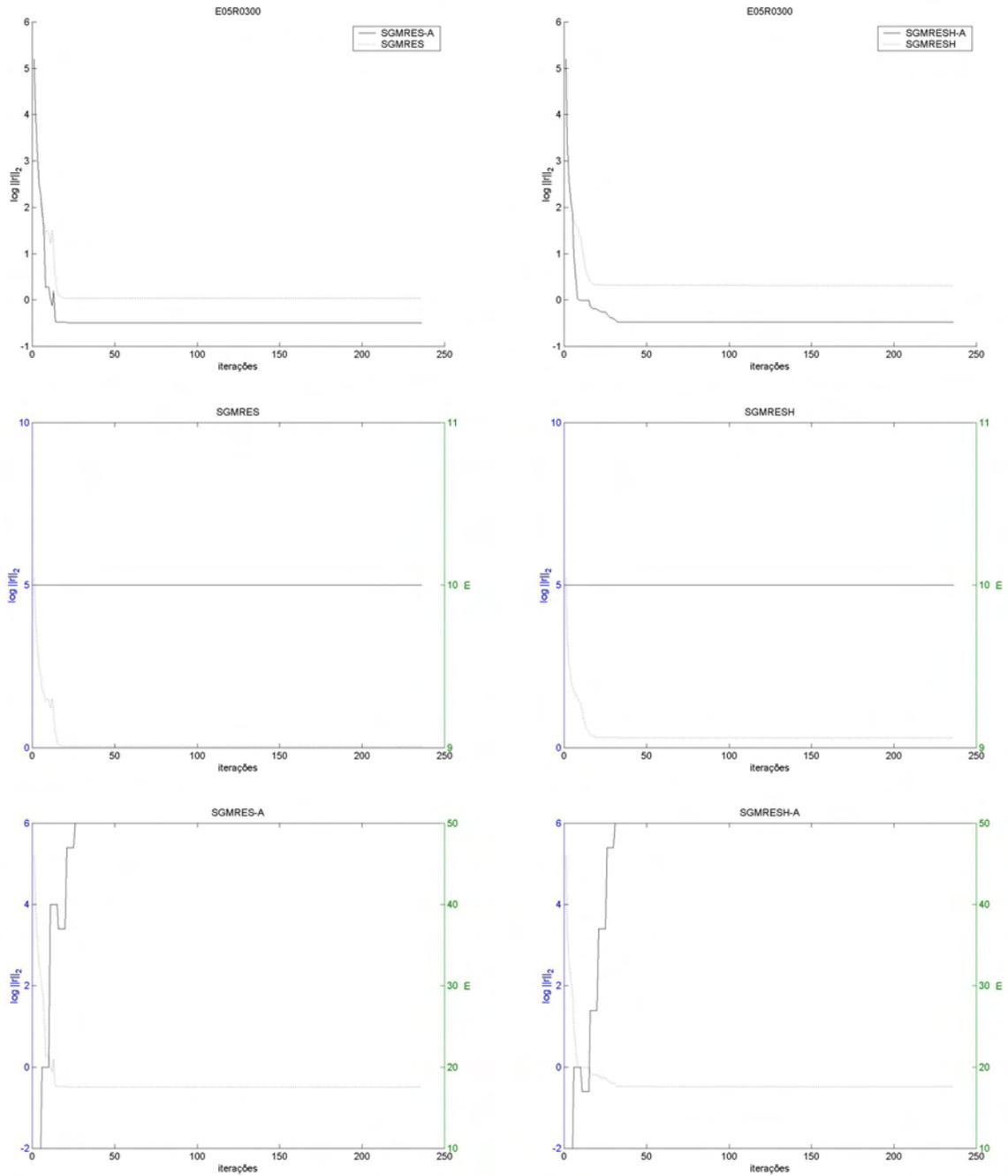


Figura 4.22: E05R0300: Gráfico do resíduo e m versus iterações

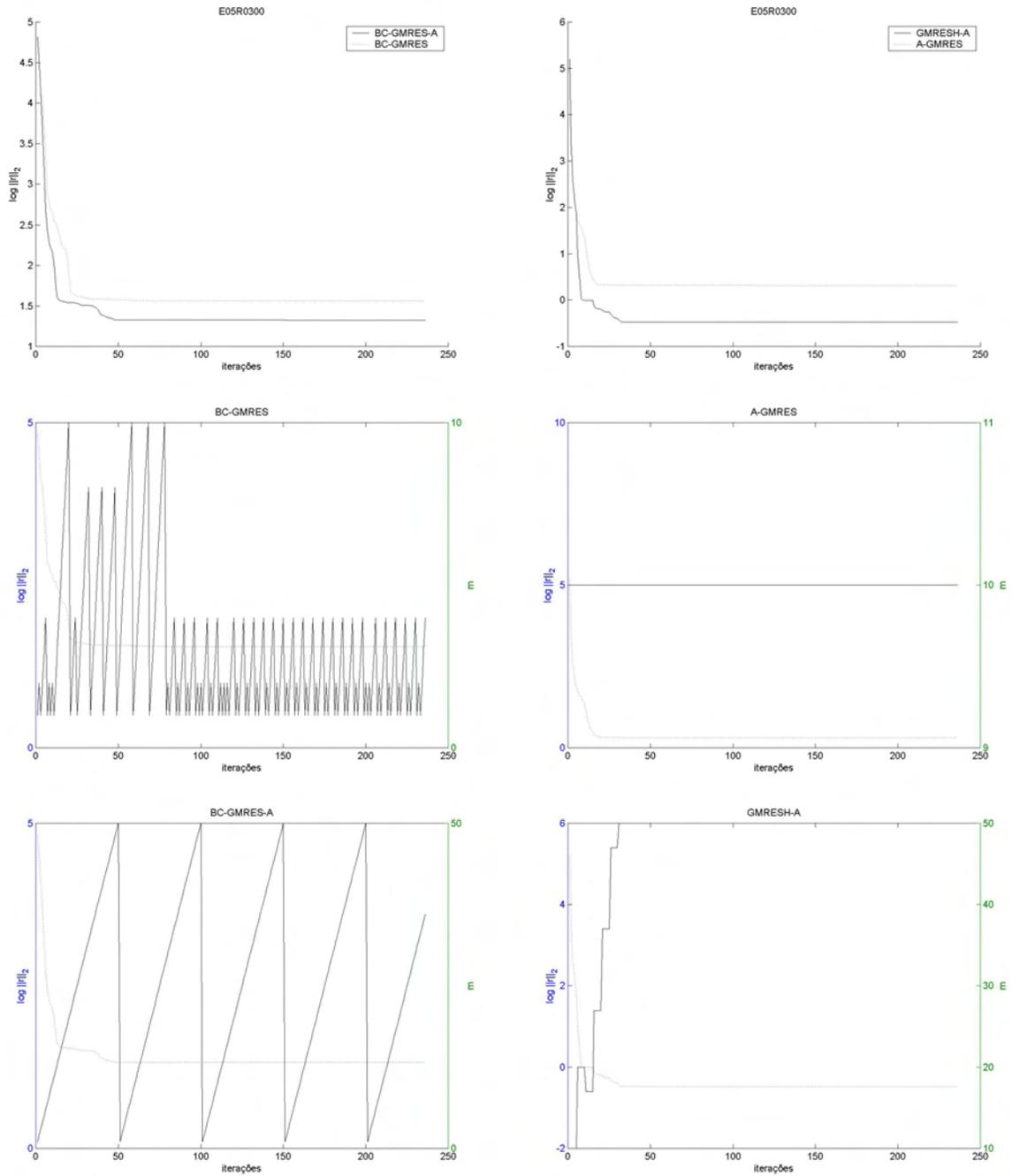


Figura 4.23: E05R0300: Gráfico do resíduo e m versus iterações

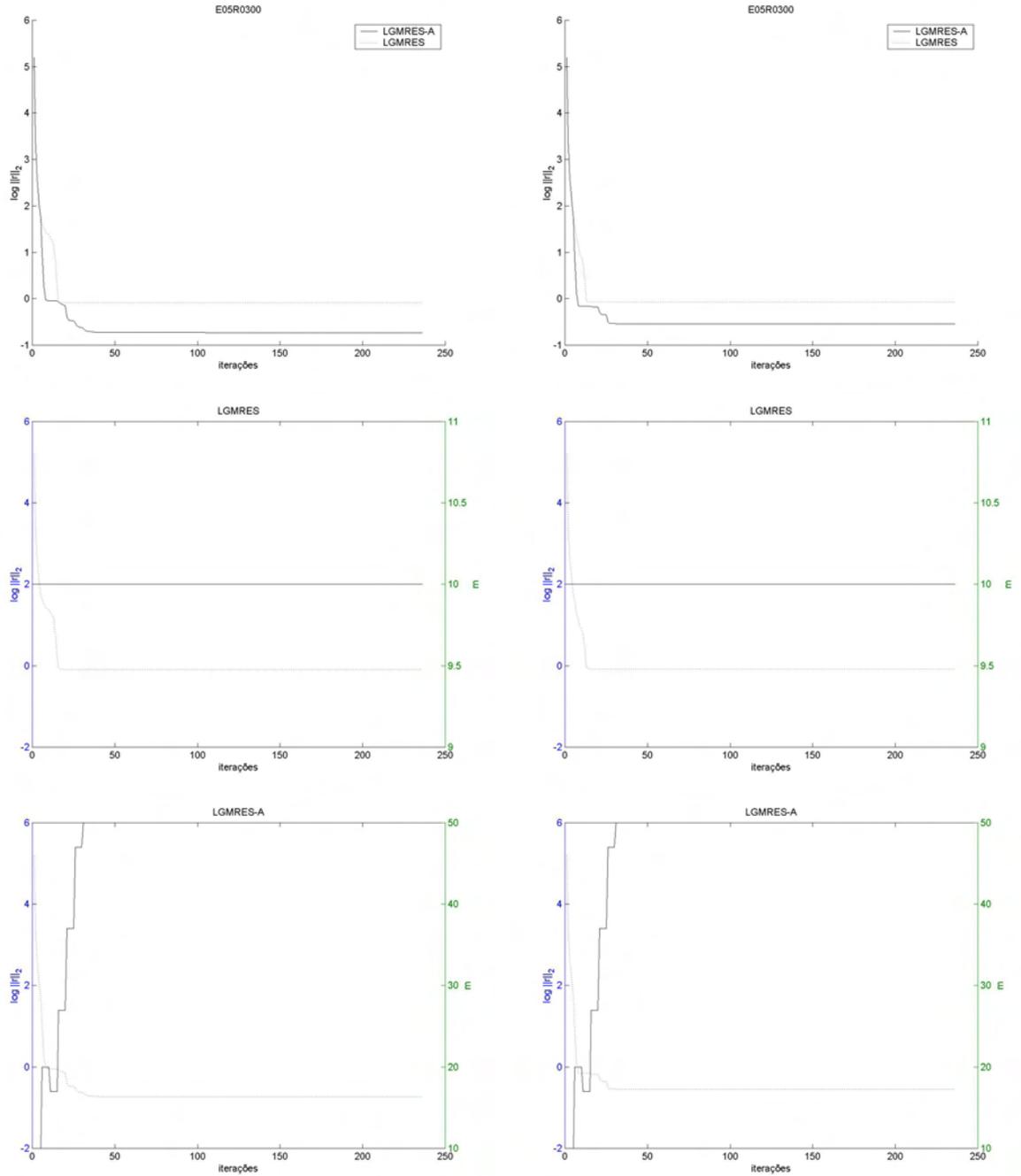


Figura 4.24: E05R0300: Gráfico do resíduo e m versus iterações

Para esse sistema nenhum dos métodos obteve a convergência, sendo assim nada pode-se concluir.

4.1.7 FIDAPM05

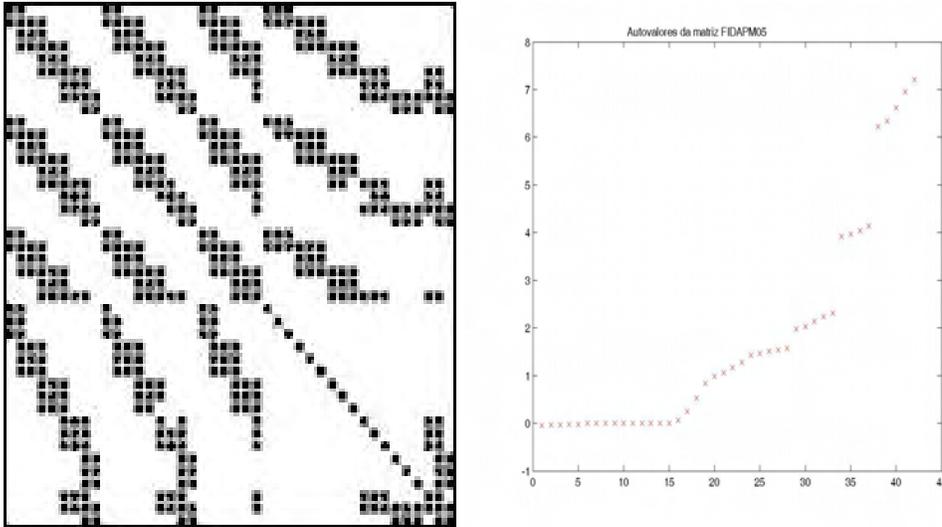


Figura 4.25: FIDAPM05: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 42×42
- tipo: real não-simétrica
- entradas não nulas: 505
- número de condição (estimado): $5,9 \times 10^{18}$
- $\|A\|_2$ estimado: 7,2

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	45	10	10	$2,6736 \times 10^6$
GMRESH-A	sim	43	10	40	$1,0547 \times 10^7$
SGMRES	sim	45	10	10	$1,8467 \times 10^6$
SGMRES-A	sim	25	10	40	$1,0027 \times 10^6$
SGMRESH	sim	45	10	10	$1,8757 \times 10^6$
SGMRESH-A	sim	45	10	25	$5,0867 \times 10^6$
BC-GMRES	sim	45	1	10	9205
BC-GMRES-A	sim	45	1	40	8917
LGMRES($m,2$)	sim	45	10	10	$3,3511 \times 10^6$
LGMRES-A($m,2$)	sim	45	10	40	$1,7852 \times 10^6$
LGMRES($m,3$)	sim	45	10	10	$3,4946 \times 10^6$
LGMRES-A($m,3$)	sim	45	10	40	$1,7267 \times 10^6$

Tabela 4.7: FIDAPM05

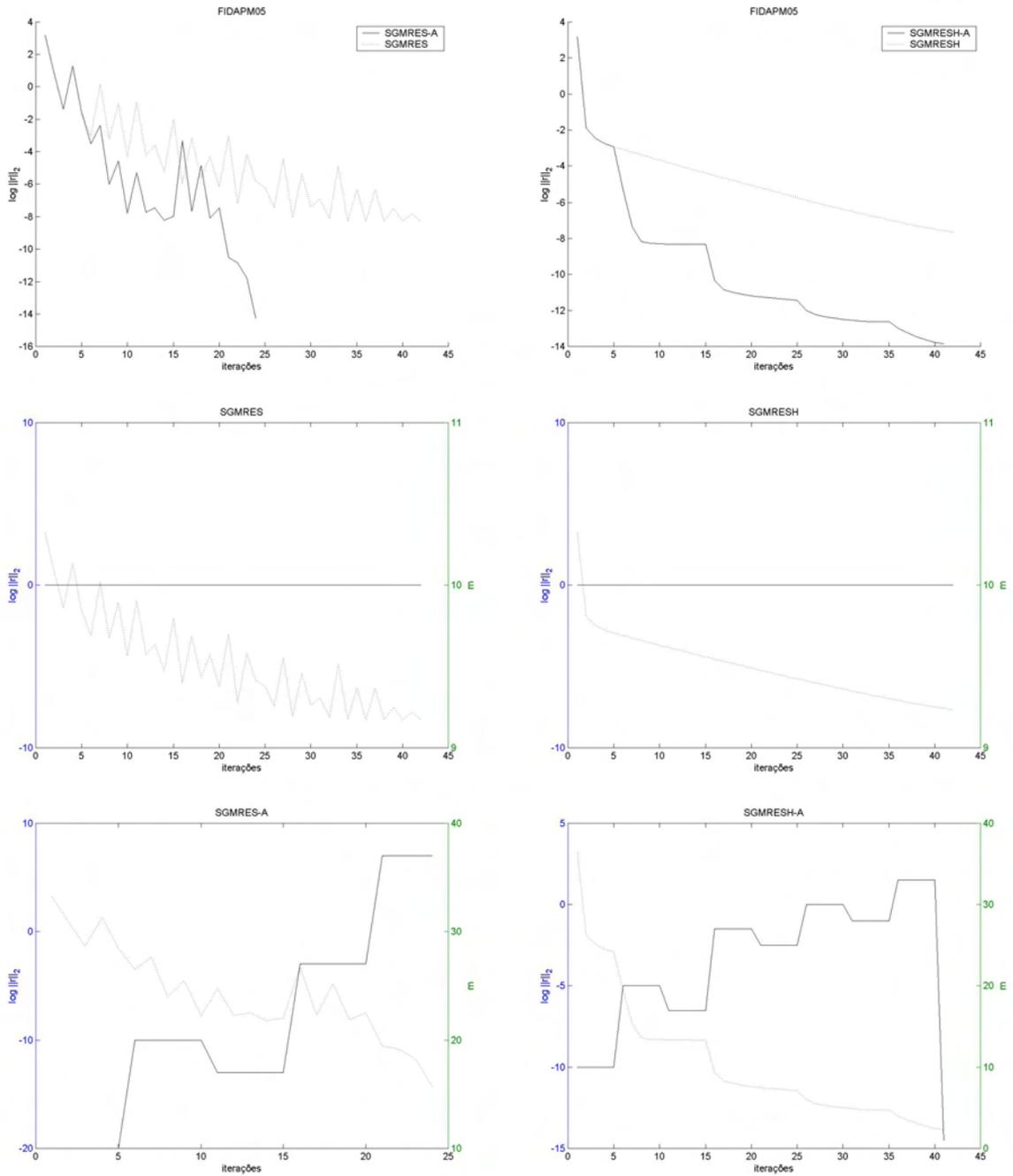


Figura 4.26: FIDAPM05: Gráfico do resíduo e m versus iterações

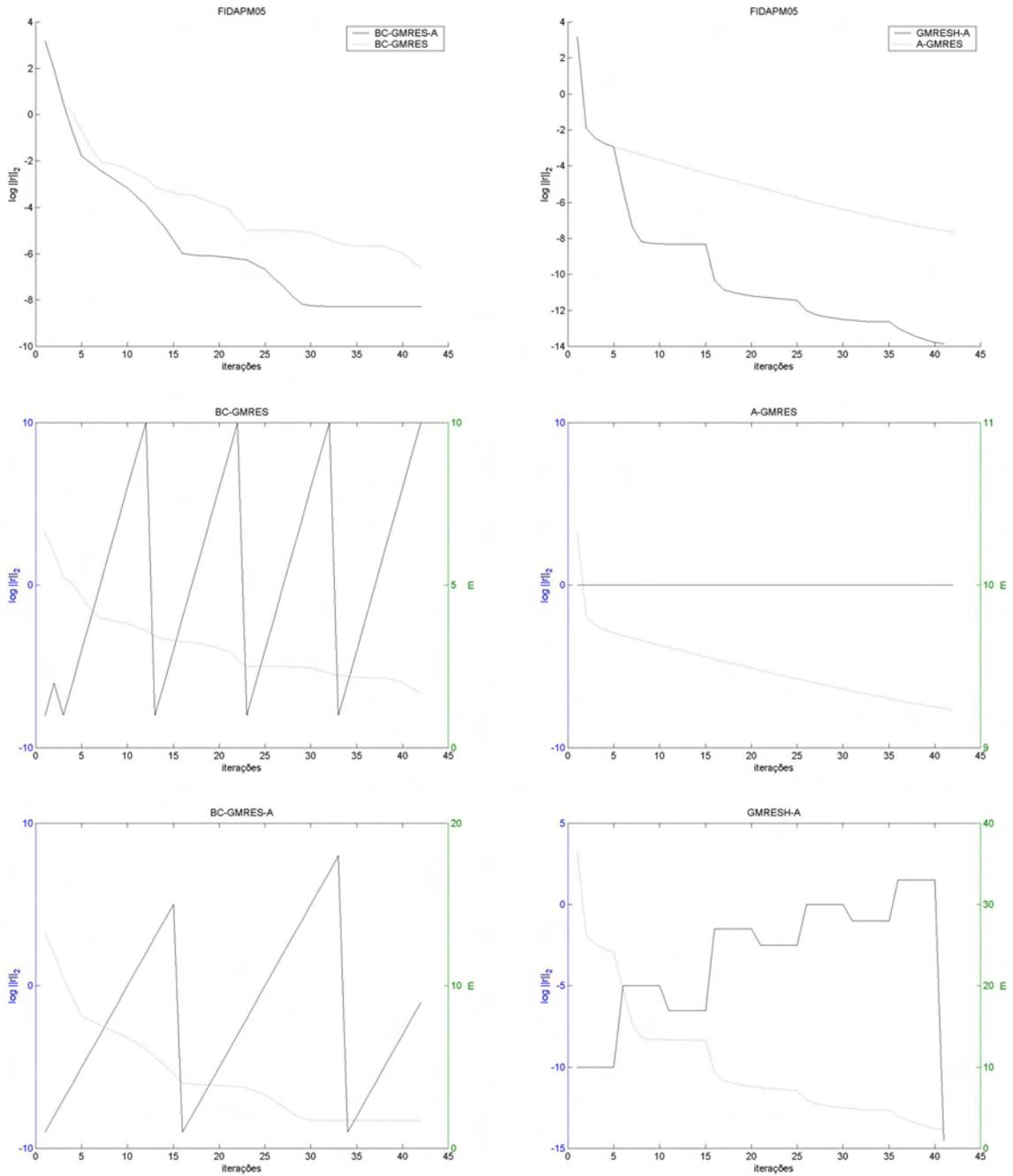


Figura 4.27: FIDAPM05: Gráfico do resíduo e m versus iterações

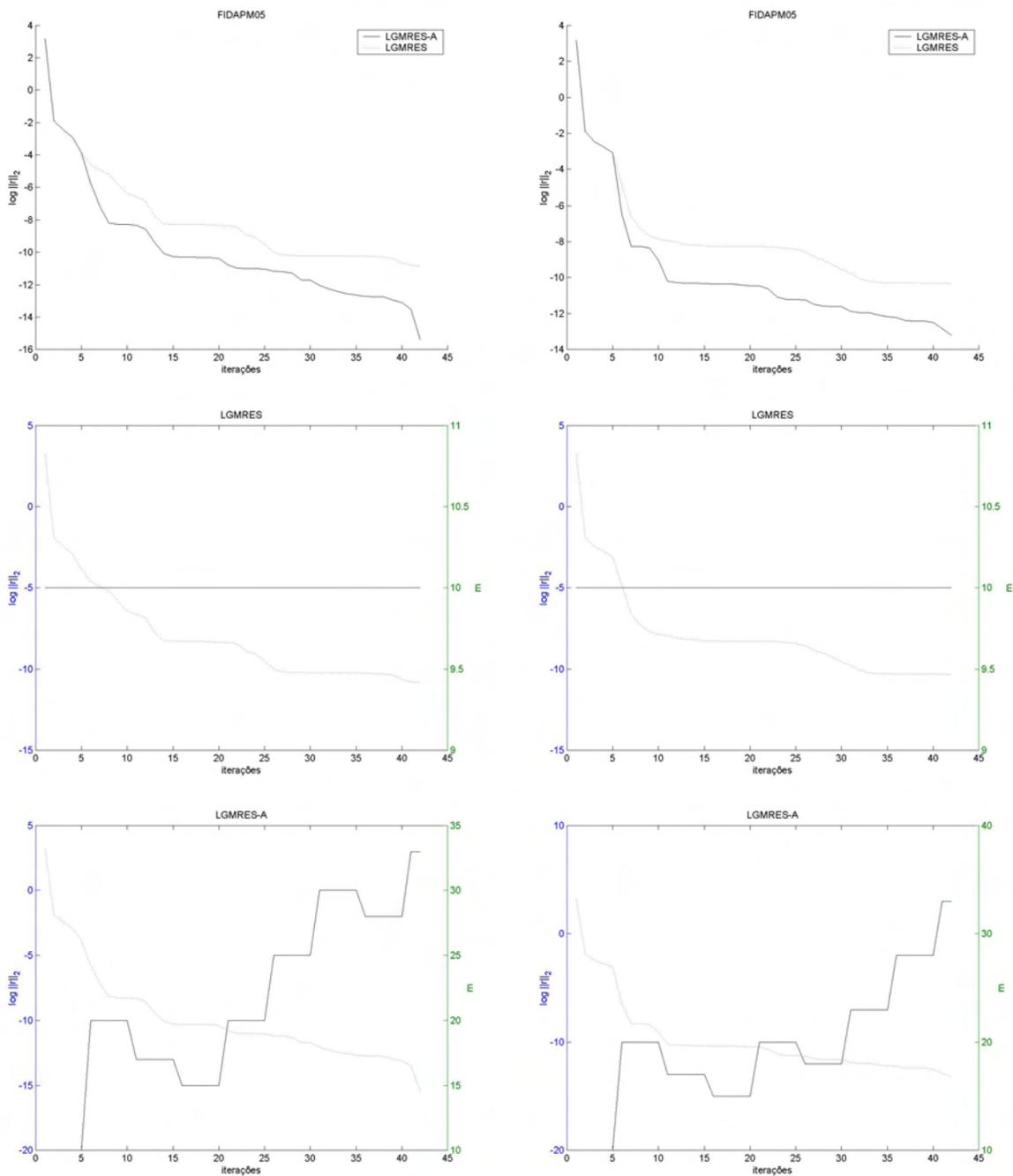


Figura 4.28: FIDAPM05: Gráfico do resíduo e m versus iterações

Na figura (4.26) note que ambos os métodos convergiram, no entanto o SGMRES-A e o SGMRESH obtiveram menor número de flops do que o SGMRES e o SGMRESH-A. Note que no SGMRES e no SGMRESH o m não sofreu variação ao longo das iterações.

Na figura (4.27) note que ambos os métodos convergiram, sendo que o BC-GMRES-A apresentou menor número de flops do que o BC-GMRES. Novamente observe o número de reinicializações necessárias para obter a convergência do BC-GMRES e do BC-GMRES-A.

Em relação ao A-GMRES e o GMRESH-A, note que ambos convergiram no mesmo número de iterações, no entanto o A-GMRES apresentou menor número de flops do que o GMRESH-A. Repare a queda brusca que sofre o logaritmo da norma residual do GMRESH-A em relação ao A-GMRES.

Na figura (4.28) podemos notar que ambos os métodos convergiram, no entanto o LGMRES-A(m, k), para $k = 2$ e $k = 3$, apresentou menor número de flops do que o LGMRES(m, k) para os mesmos k . Observe que os gráficos foram muito similares em relação a variação dos k .

4.1.8 FS_183_1

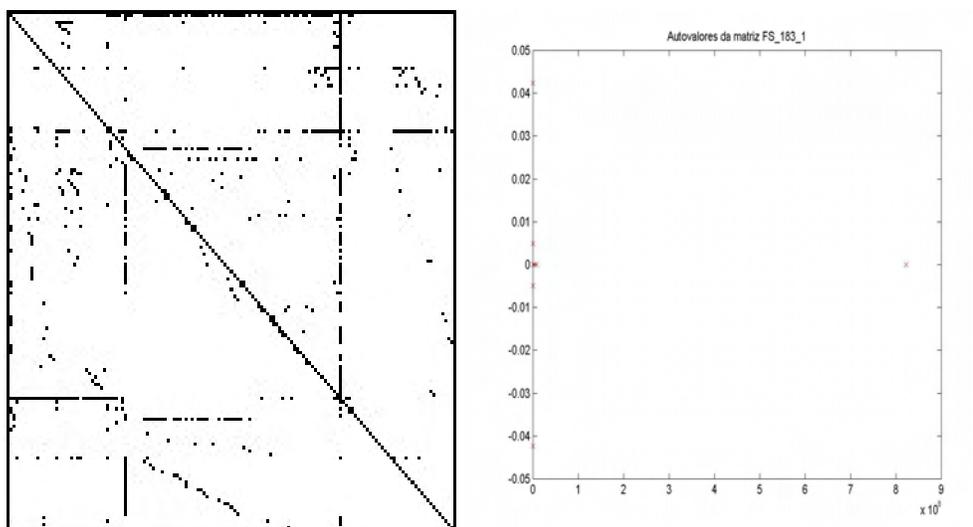


Figura 4.29: FS_183_1: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 183×183
- tipo: real não-simétrica
- entradas não nulas: 998
- número de condição (estimado): $1,5 \times 10^{13}$
- $\|A\|_2$ estimado: $1,1 \times 10^9$

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	não	-	10	10	$4,7438 \times 10^7$
GMRESH-A	sim	25	10	40	$3,7068 \times 10^7$
SGMRES	não	-	10	10	$1,4040 \times 10^7$
SGMRES-A	sim	25	10	50	$1,0266 \times 10^7$
SGMRESH	não	-	10	10	$3,2088 \times 10^7$
SGMRESH-A	sim	35	10	40	$3,3519 \times 10^7$
BC-GMRES	não	-	1	10	$7,2208 \times 10^6$
BC-GMRES-A	não	-	1	50	$6,7654 \times 10^7$
LGMRES($m,2$)	não	-	10	10	$1,8578 \times 10^7$
LGMRES-A($m,2$)	não	-	10	50	$2,0916 \times 10^8$
LGMRES($m,3$)	não	-	10	10	$2,0937 \times 10^7$
LGMRES-A($m,3$)	não	-	10	50	$2,1725 \times 10^8$

Tabela 4.8: FS_183.1

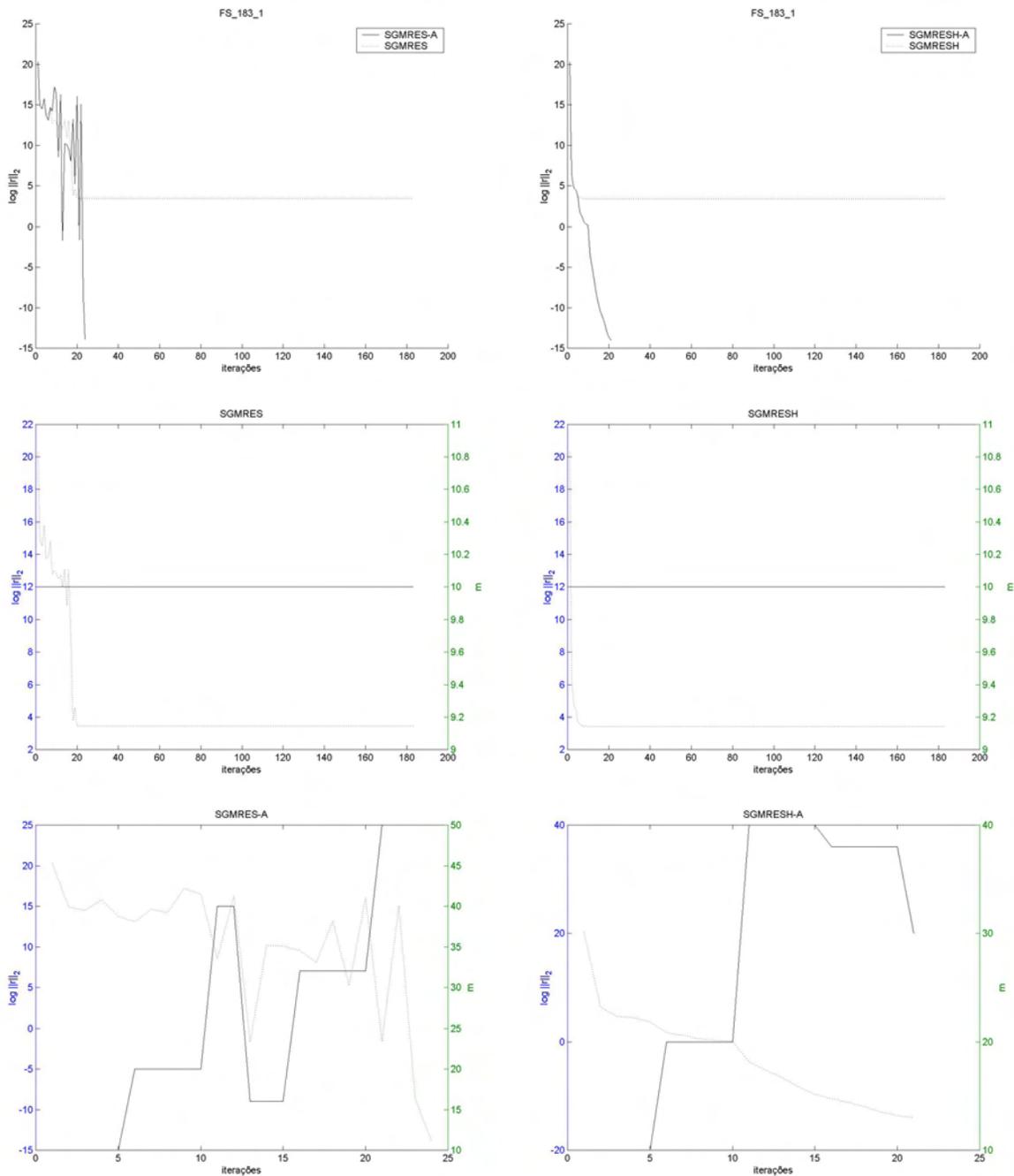


Figura 4.30: FS_183.1: Gráfico do resíduo e m versus iterações

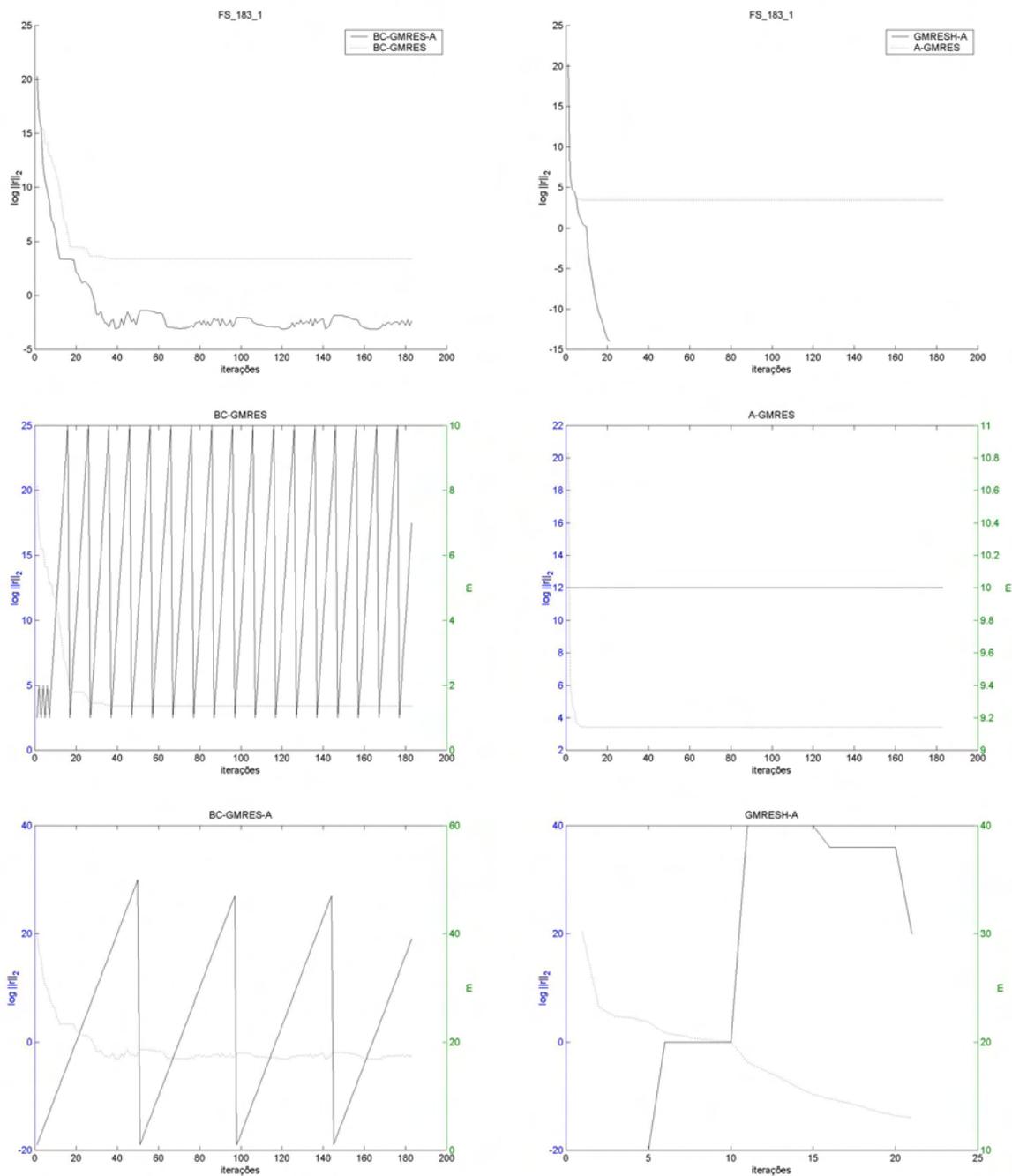


Figura 4.31: FS_183.1: Gráfico do resíduo e m versus iterações

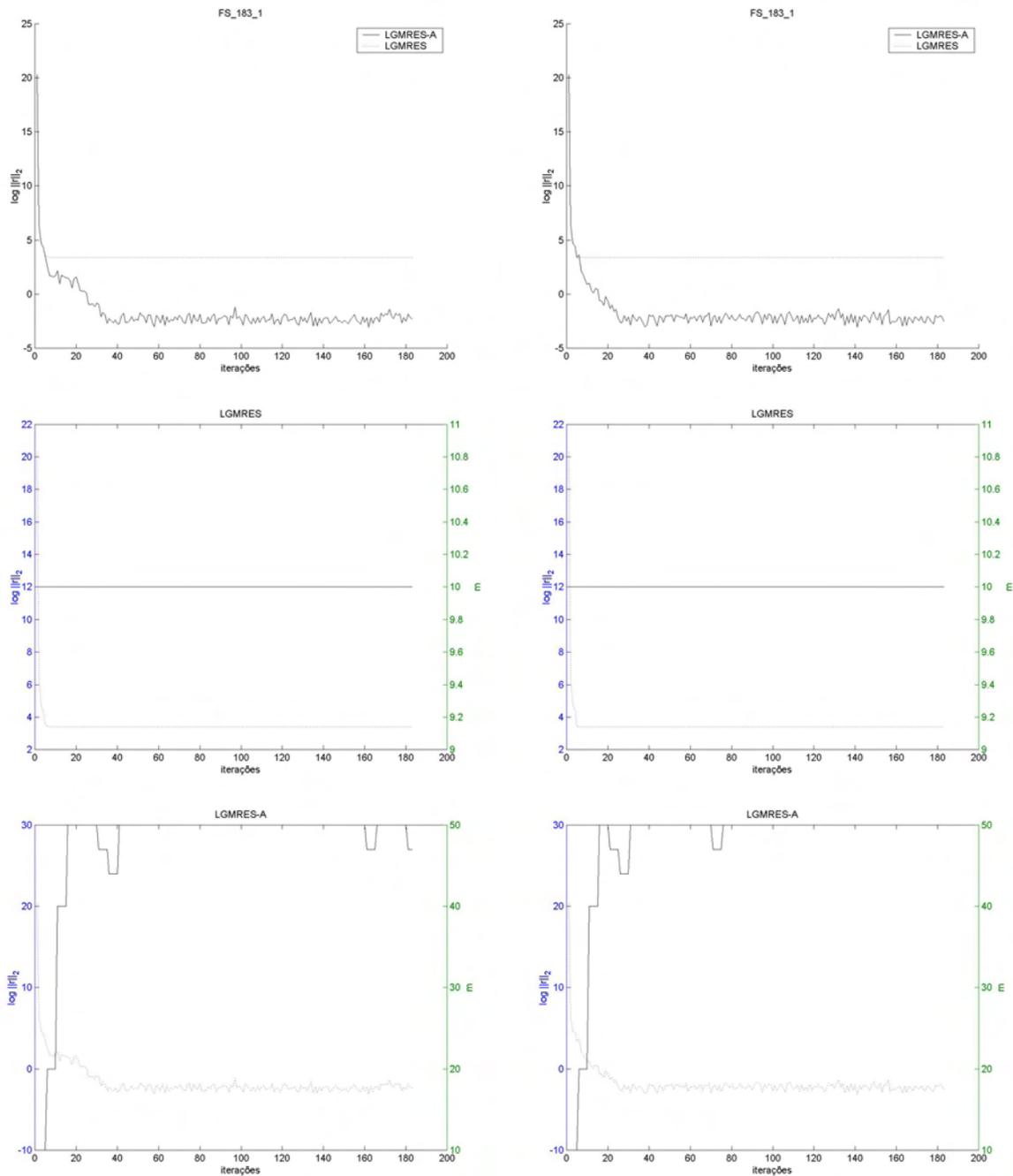


Figura 4.32: FS_183.1: Gráfico do resíduo e m versus iterações

Na figura (4.30) note que apenas o SGMRES-A e o SGMRESH-A obtiveram a convergência dentro da tolerância pré-especificada. Note também que o m não apresentou variação no SGMRES e no SGMRESH.

O BC-GMRES, BC-GMRES-A, A-GMRES, LGMRES(m, k) e o LGMRES-A(m, k), para $k = 2$ e $k = 3$ não convergiram dentro da tolerância pré-especificada, o único que obteve a convergência foi o GMRESH-A; veja as figuras (4.31) e (4.32).

4.1.9 FS_760_1

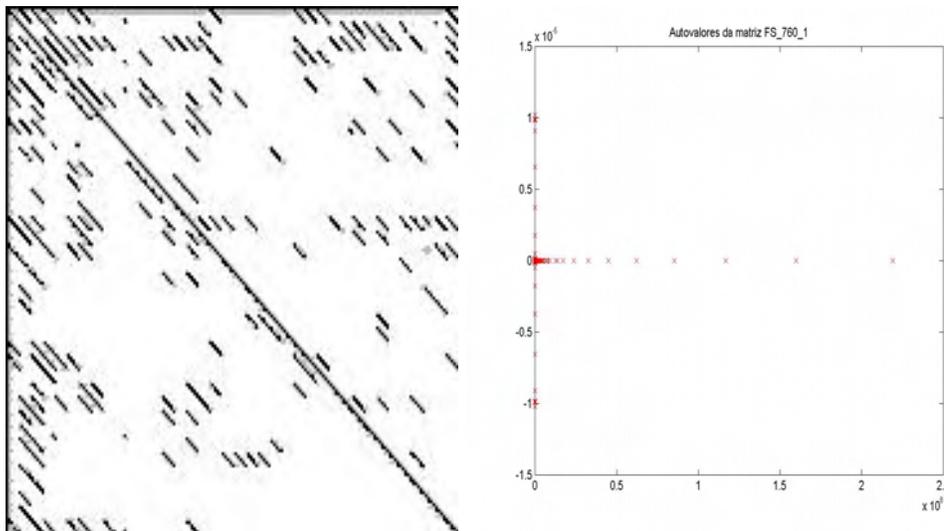


Figura 4.33: FS_760_1: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 760×760
- tipo: real não-simétrica
- entradas não nulas: 5739
- número de condição (estimado): $8,4 \times 10^3$
- $\|A\|_2$ estimado: $3,1 \times 10^8$

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	70	7	10	$7,0129 \times 10^7$
GMRESH-A	sim	12	10	40	$4,4302 \times 10^7$
SGMRES	sim	250	10	10	$8,2974 \times 10^7$
SGMRES-A	sim	15	10	30	$1,4820 \times 10^7$
SGMRESH	sim	70	3	10	$4,6368 \times 10^7$
SGMRESH-A	sim	14	10	30	$2,9837 \times 10^7$
BC-GMRES	sim	150	1	10	$2,6623 \times 10^7$
BC-GMRES-A	sim	25	1	25	$1,4886 \times 10^7$
LGMRES($m,2$)	sim	60	10	10	$2,3134 \times 10^7$
LGMRES-A($m,2$)	sim	11	10	40	$1,4359 \times 10^7$
LGMRES($m,3$)	sim	50	10	10	$2,4208 \times 10^7$
LGMRES-A($m,3$)	sim	11	10	40	$1,5043 \times 10^7$

Tabela 4.9: FS_760_1

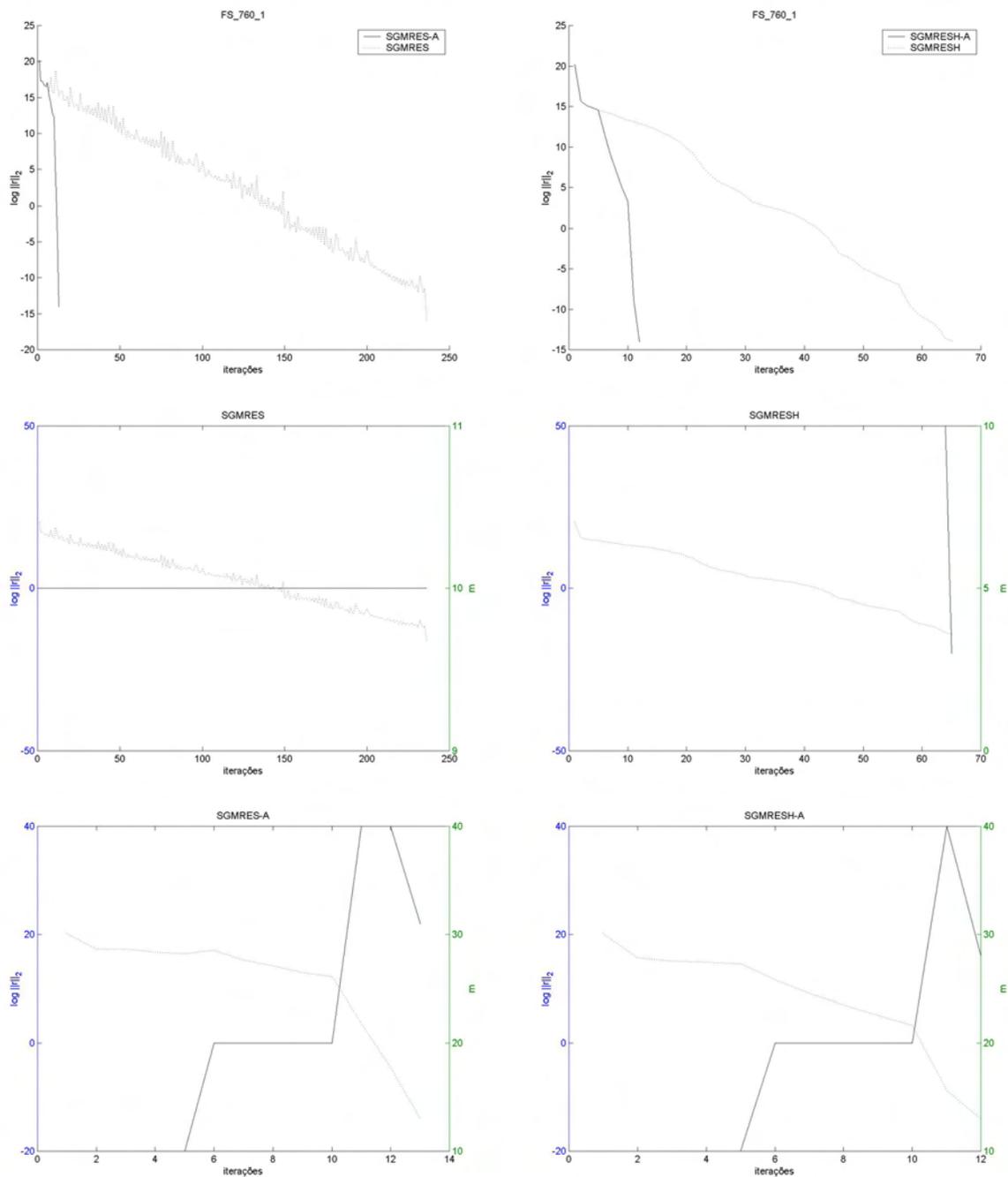


Figura 4.34: FS_760_1: Gráfico do resíduo e m versus iterações

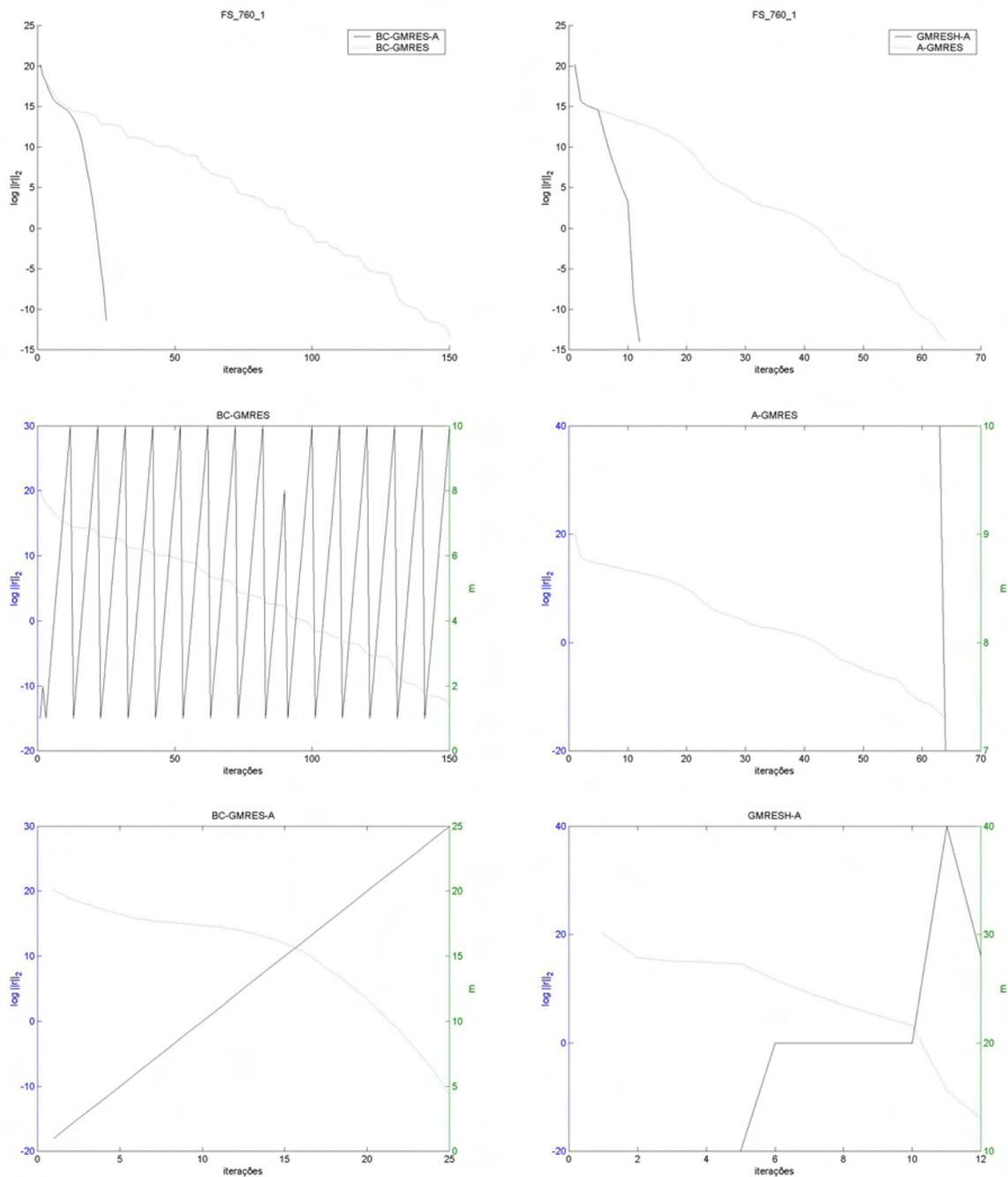


Figura 4.35: FS_760_1: Gráfico do resíduo e m versus iterações

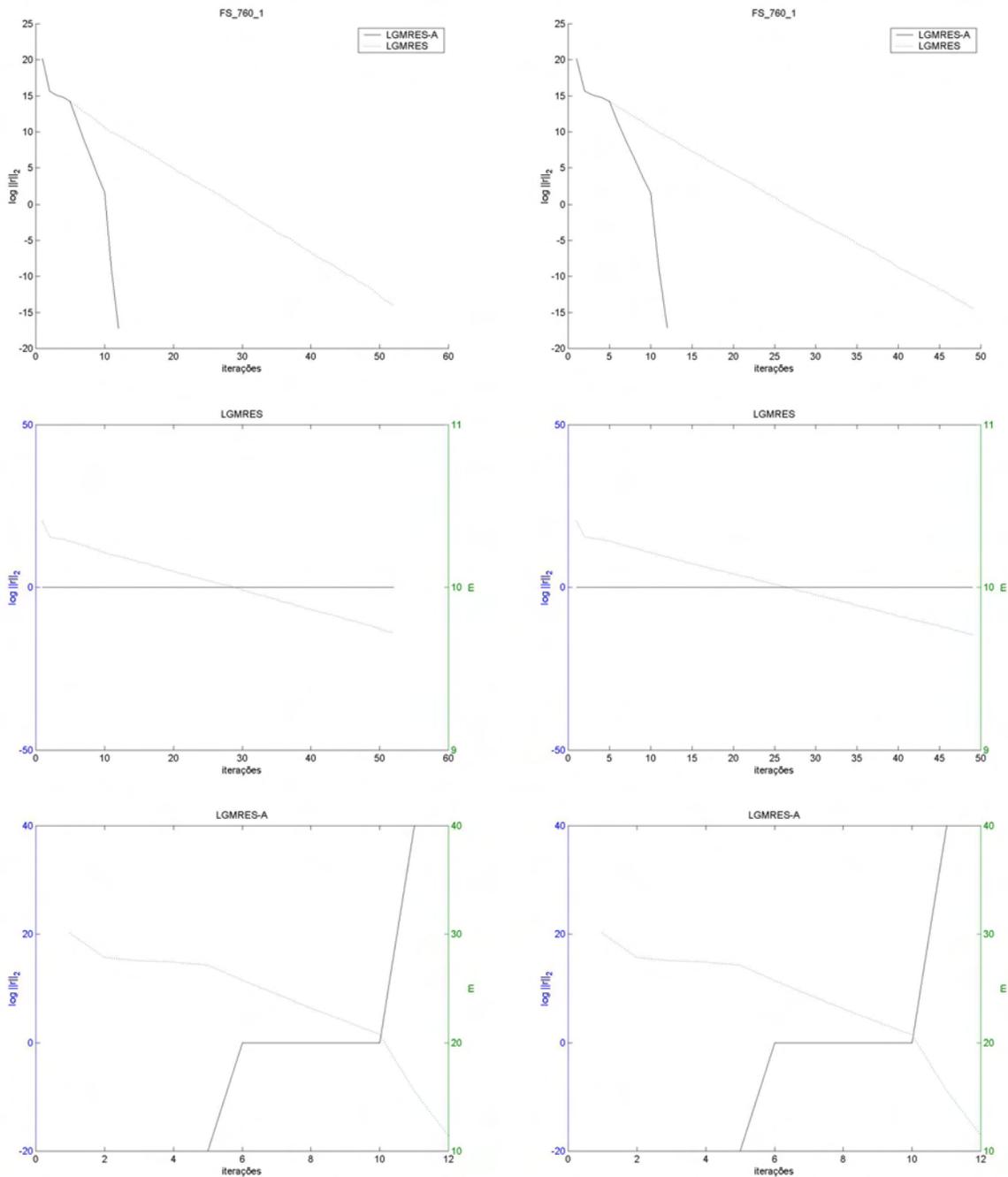


Figura 4.36: FS_760_1: Gráfico do resíduo e m versus iterações

Na figura (4.34) note que ambos os métodos convergiram, sendo que o SGMRES-A e o SGMRESH-A apresentaram menor número de flops do que o SGMRES e o SGMRESH. Observe que o comportamento do SGMRES-A e do SGMRESH-A apresentam algumas similaridades, um exemplo disso seria quanto ao m , visto que o mesmo apresenta os mesmos momentos de estagnação e acréscimo. É importante destacar que o decréscimo do m apresentado no gráfico do SGMRESH deu-se devido a detecção de um subespaço invariante na formação do subespaço de Krylov.

Na figura (4.35) note que ambos os métodos convergiram, no entanto o BC-GMRES-A e o GMRESH-A apresentaram menor número de flops do que o BC-GMRES e o A-GMRES. Observe o comportamento do m no gráfico do BC-GMRES-A, o mesmo apresentou-se em forma de “degraus”, esse comportamento sempre ocorre no BC-GMRES-A visto que o mesmo é estruturado de forma que, quando o resíduo não sofre um decréscimo suficiente, o método tende a ser reinicializado.

Na figura (4.36) ambos os métodos convergiram, no entanto o LGMRES-A(m, k), para $k = 2$ e $k = 3$, apresentou menor número de flops do que o LGMRES(m, k) para os mesmos k . Observe que o comportamento entre o LGMRES($m, 2$) e o LGMRES($m, 3$) deu-se de forma similar, sendo que o mesmo ocorreu para o LGMRES-A($m, 2$) e LGMRES($m, 3$).

4.1.10 GRE_115

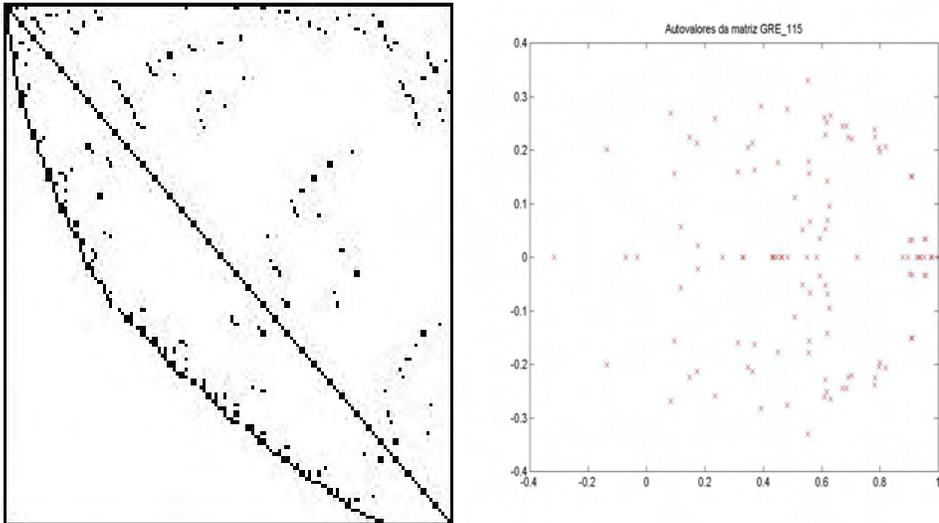


Figura 4.37: GRE_115: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 115×115
- tipo: real não-simétrica
- entradas não nulas: 421
- número de condição (estimado): 150
- $\|A\|_2$ estimado: 1

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	não	-	10	10	$1,7776 \times 10^7$
GMRESH-A	sim	30	10	25	$1,2150 \times 10^7$
SGMRES	não	-	10	10	$5,0327 \times 10^7$
SGMRES-A	sim	20	10	30	$2,0861 \times 10^7$
SGMRESH	não	-	10	10	$1,2057 \times 10^7$
SGMRESH-A	sim	45	10	25	$1,1363 \times 10^6$
BC-GMRES	não	-	1	10	9494
BC-GMRES-A	sim	30	1	40	$2,7070 \times 10^6$
LGMRES($m,2$)	não	-	10	10	$6,8308 \times 10^6$
LGMRES-A($m,2$)	sim	30	10	25	$3,5714 \times 10^6$
LGMRES($m,3$)	não	-	10	10	$7,7084 \times 10^6$
LGMRES-A($m,3$)	sim	20	10	30	$3,1444 \times 10^6$

Tabela 4.10: GRE_115

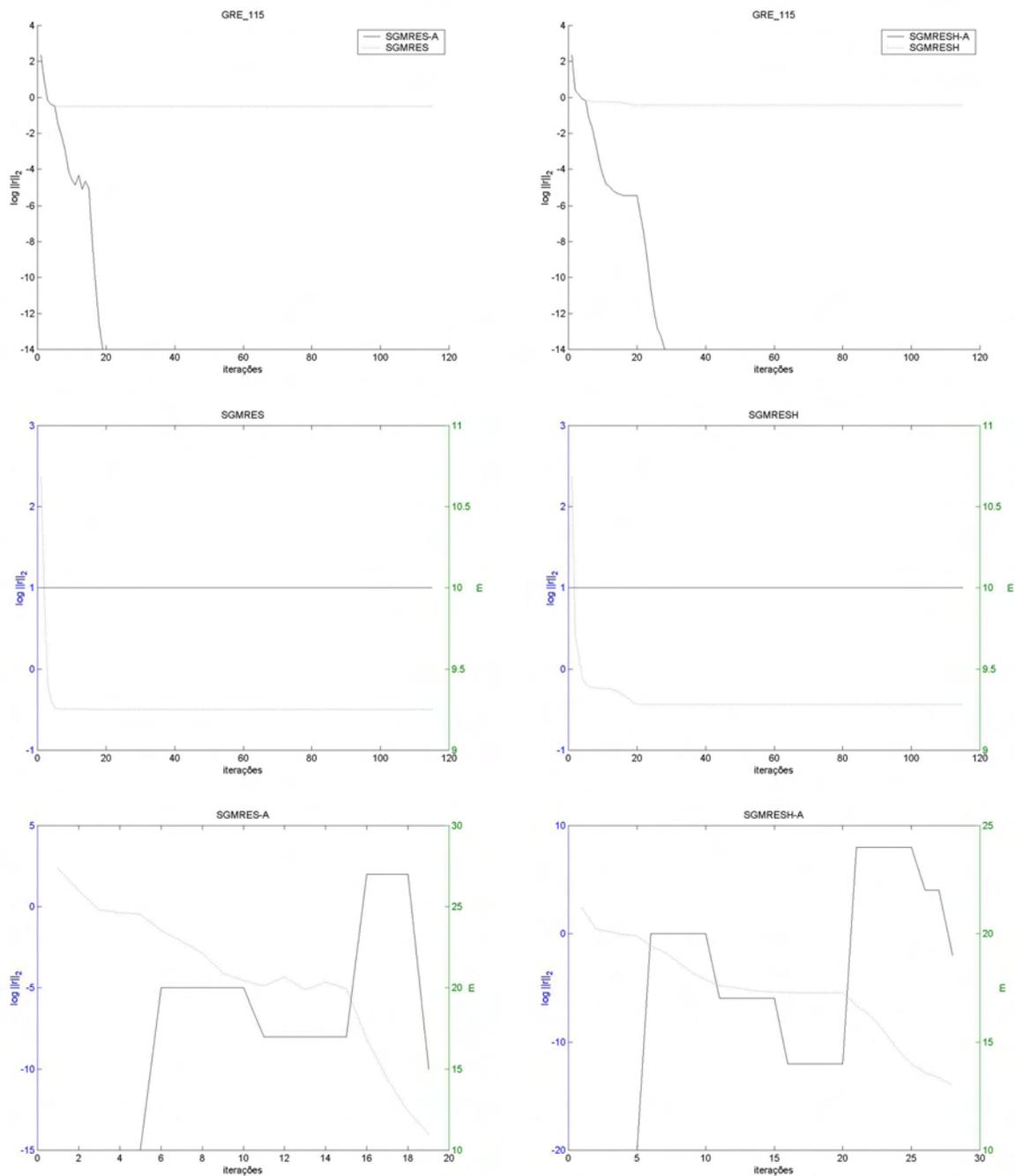


Figura 4.38: GRE_115: Gráfico do resíduo e m versus iterações

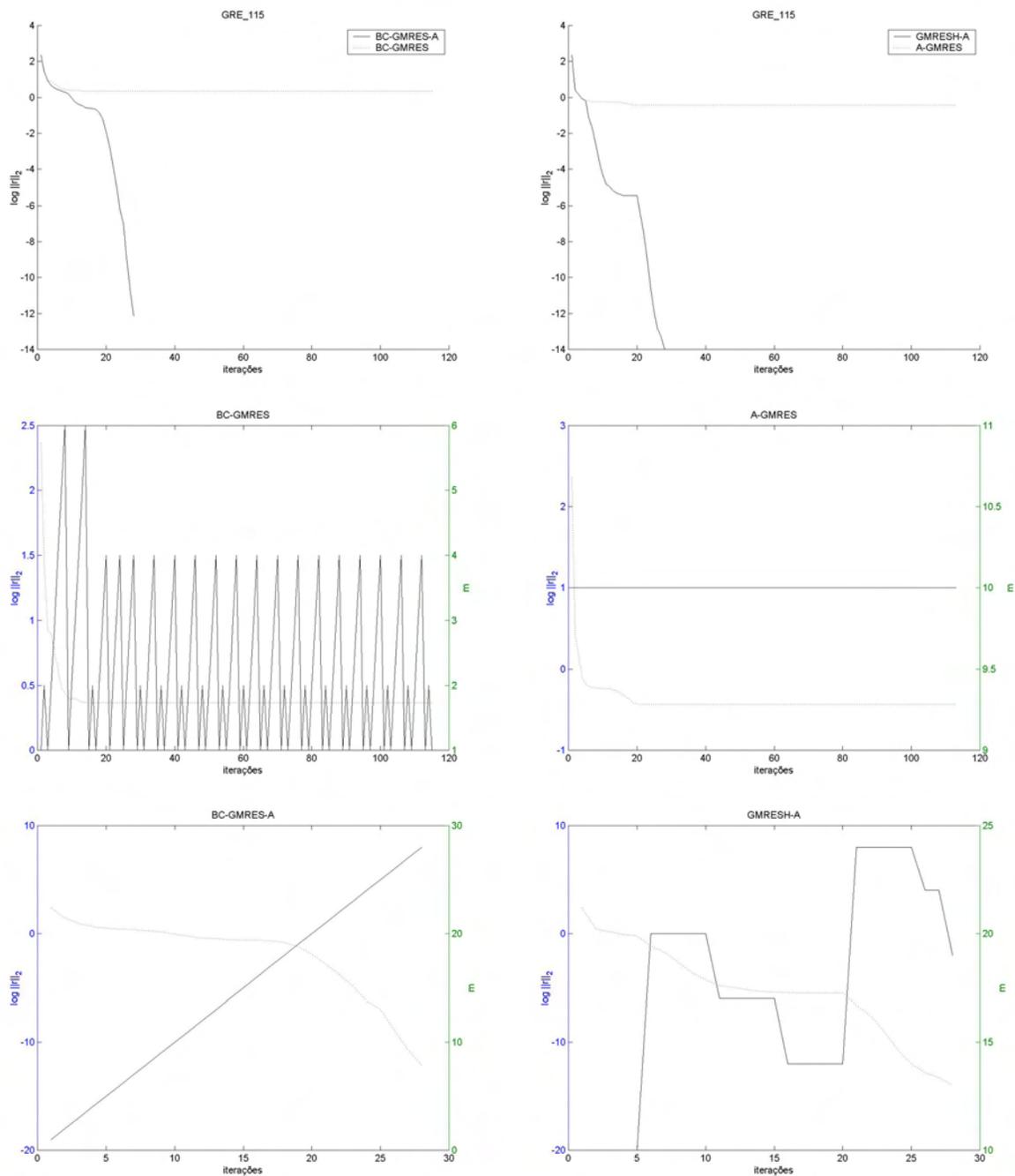


Figura 4.39: GRE_115: Gráfico do resíduo e m versus iterações

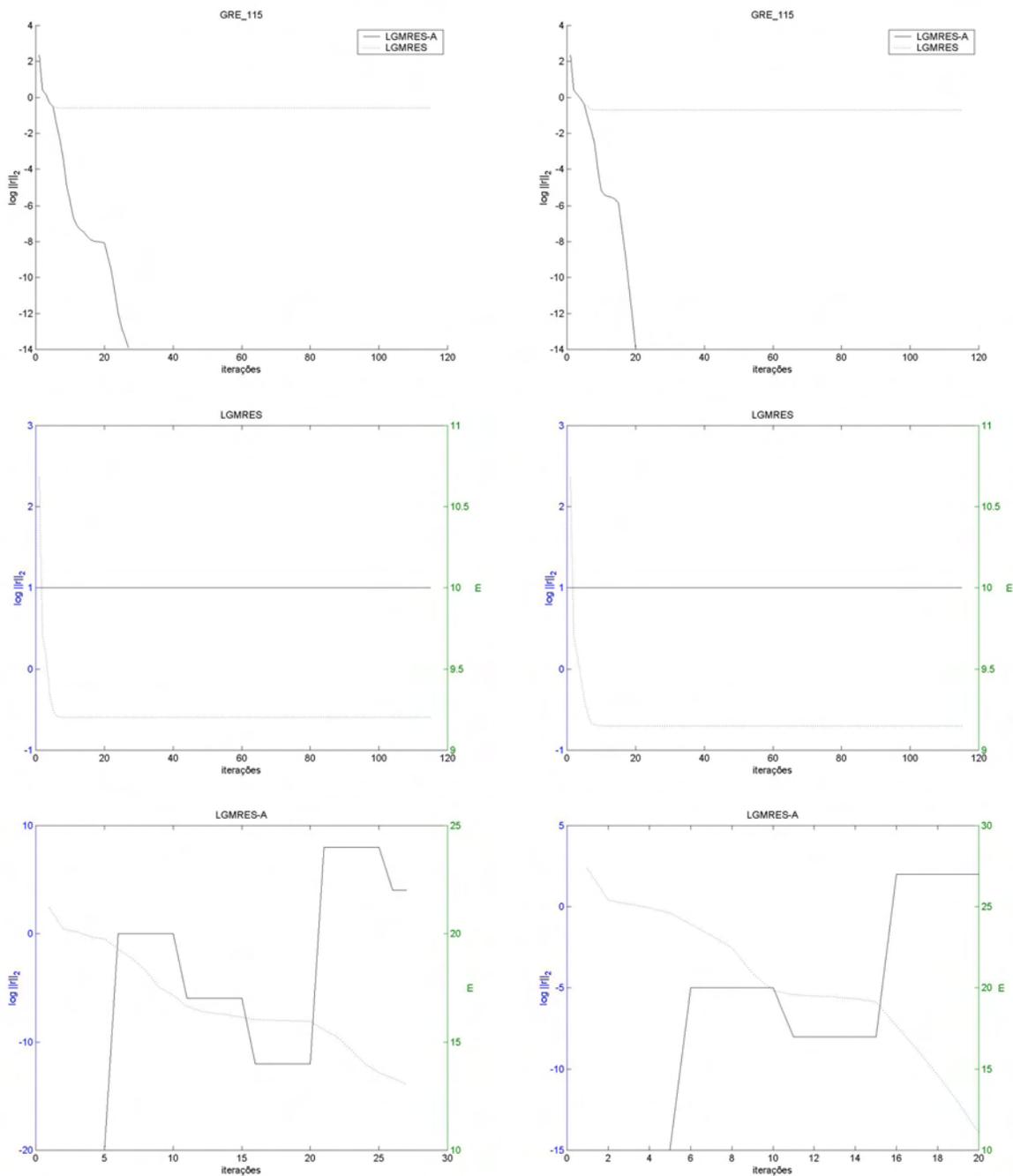


Figura 4.40: GRE_115: Gráfico do resíduo e m versus iterações

Observe através dos gráficos que apenas os métodos acrescidos do critério 4 obtiveram a convergência dentro da tolerância pré-especificada inicialmente.

Observe a figura (4.38): nela percebemos que o SGMRES e o SGMRESH apresentaram estagnação à partir da iteração de número dez e número vinte, respectivamente. Note que o acréscido do critério 4 fez com que o SGMRES e o SGMRESH obtivessem a convergência e fez com que o m se adaptasse às necessidades do sistema em questão.

O mesmo comportamento descrito acima deu-se para o BC-GMRES e para o A-GMRES, novamente apenas o BC-GMRES-A e o GMRESH-A obtiveram a convergência.

Observe a figura (4.40) note que os gráficos referente ao LGMRES($m, 2$) e LGMRES($m, 3$) apresentam um comportamento muito similar, contudo no LGMRES-A($m, 2$) e no LGMRES-A($m, 3$) isso não ocorreu, note que tanto o resíduo como o m variou de forma diferente em ambos os métodos, o que não é um comportamento comum nos testes apresentados até o momento.

4.1.11 IMPCOL_A

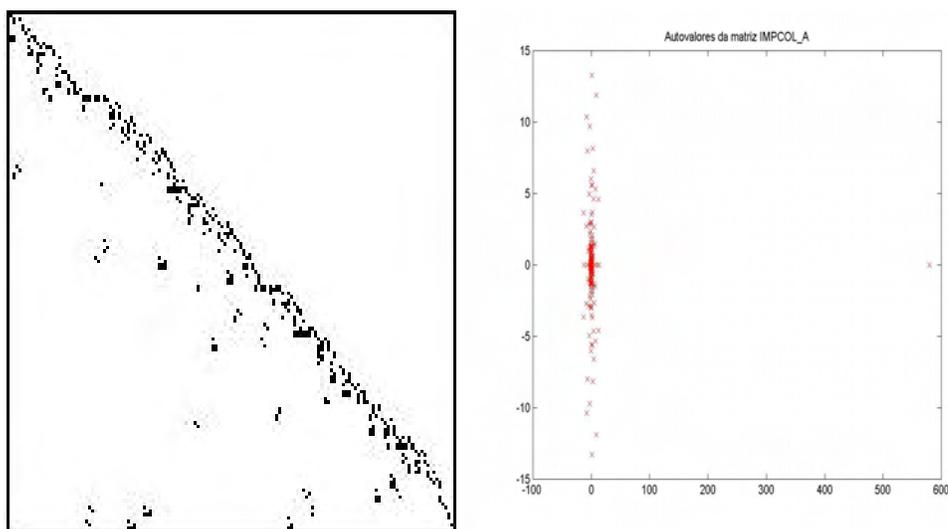


Figura 4.41: IMPCOL_A: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 207×207
- tipo: real não-simétrica
- entradas não nulas: 572
- número de condição (estimado): $4,3 \times 10^7$
- $\|A\|_2$ estimado: 860

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	não	-	9	11	$5,8237 \times 10^7$
GMRESH-A	não	-	10	50	$1,0505 \times 10^9$
SGMRES	não	-	10	10	$1,5421 \times 10^7$
SGMRES-A	não	-	10	50	$2,3529 \times 10^8$
SGMRESH	não	-	10	10	$3,8578 \times 10^7$
SGMRESH-A	não	-	10	50	$6,8723 \times 10^8$
BC-GMRES	não	-	1	6	$3,3971 \times 10^6$
BC-GMRES-A	não	-	1	50	$8,9630 \times 10^7$
LGMRES($m,2$)	não	-	10	10	$2,0770 \times 10^7$
LGMRES-A($m,2$)	não	-	10	50	$2,6046 \times 10^8$
LGMRES($m,3$)	não	-	10	10	$2,3540 \times 10^7$
LGMRES-A($m,3$)	não	-	10	50	$2,7005 \times 10^8$

Tabela 4.11: IMPCOLA

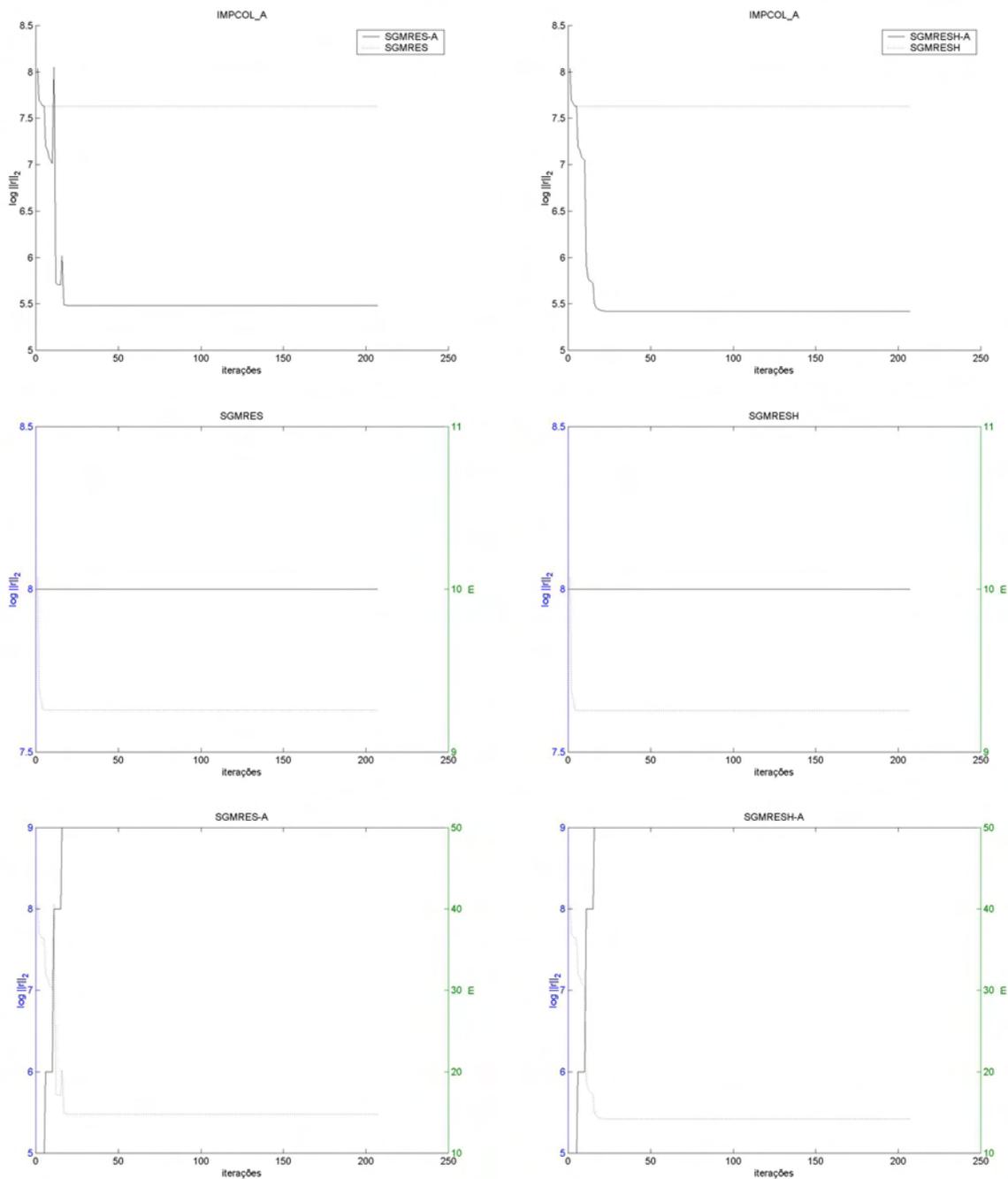


Figura 4.42: IMPCOL_A: Gráfico do resíduo e m versus iterações

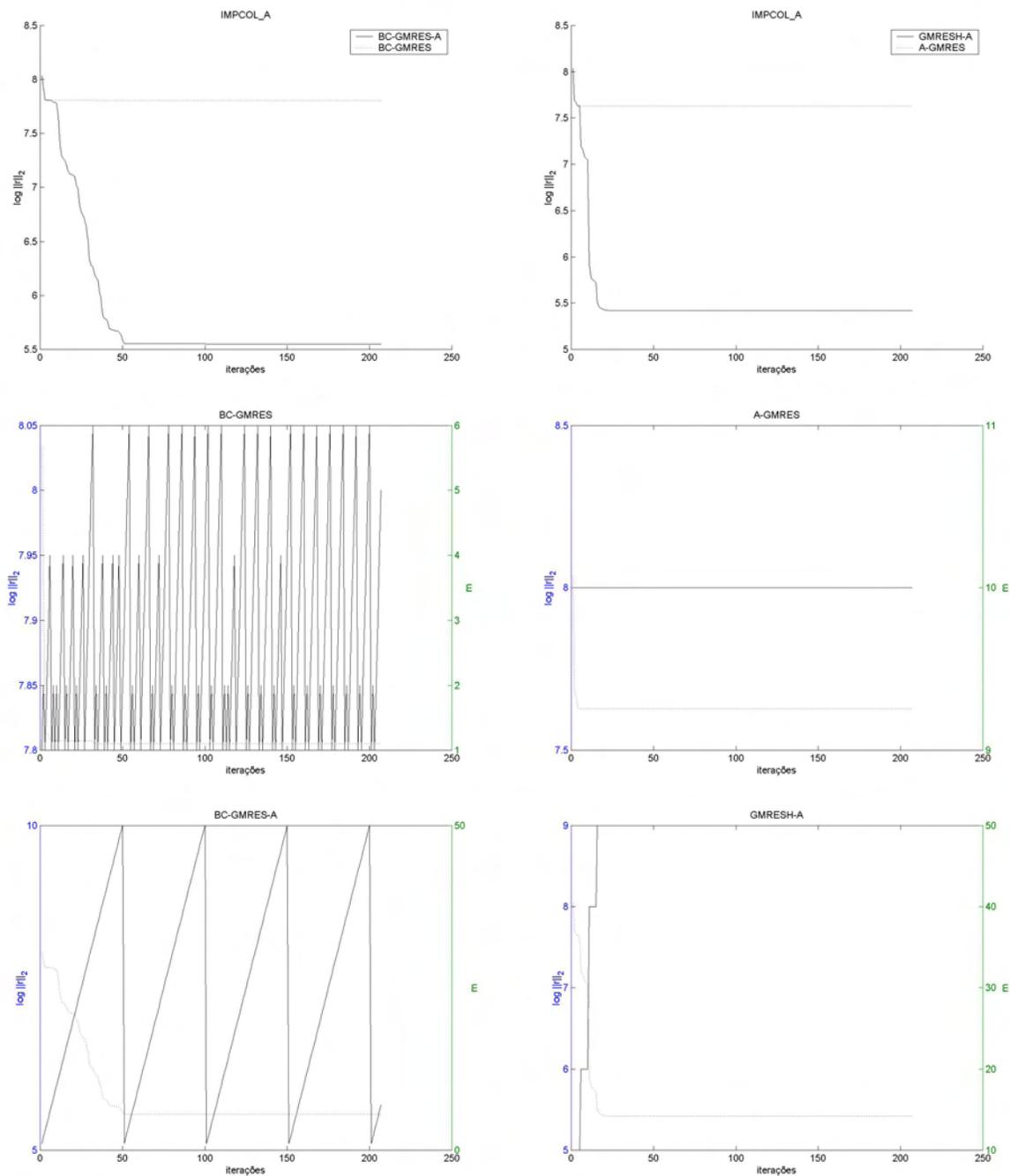


Figura 4.43: IMPCOL_A: Gráfico do resíduo e m versus iterações

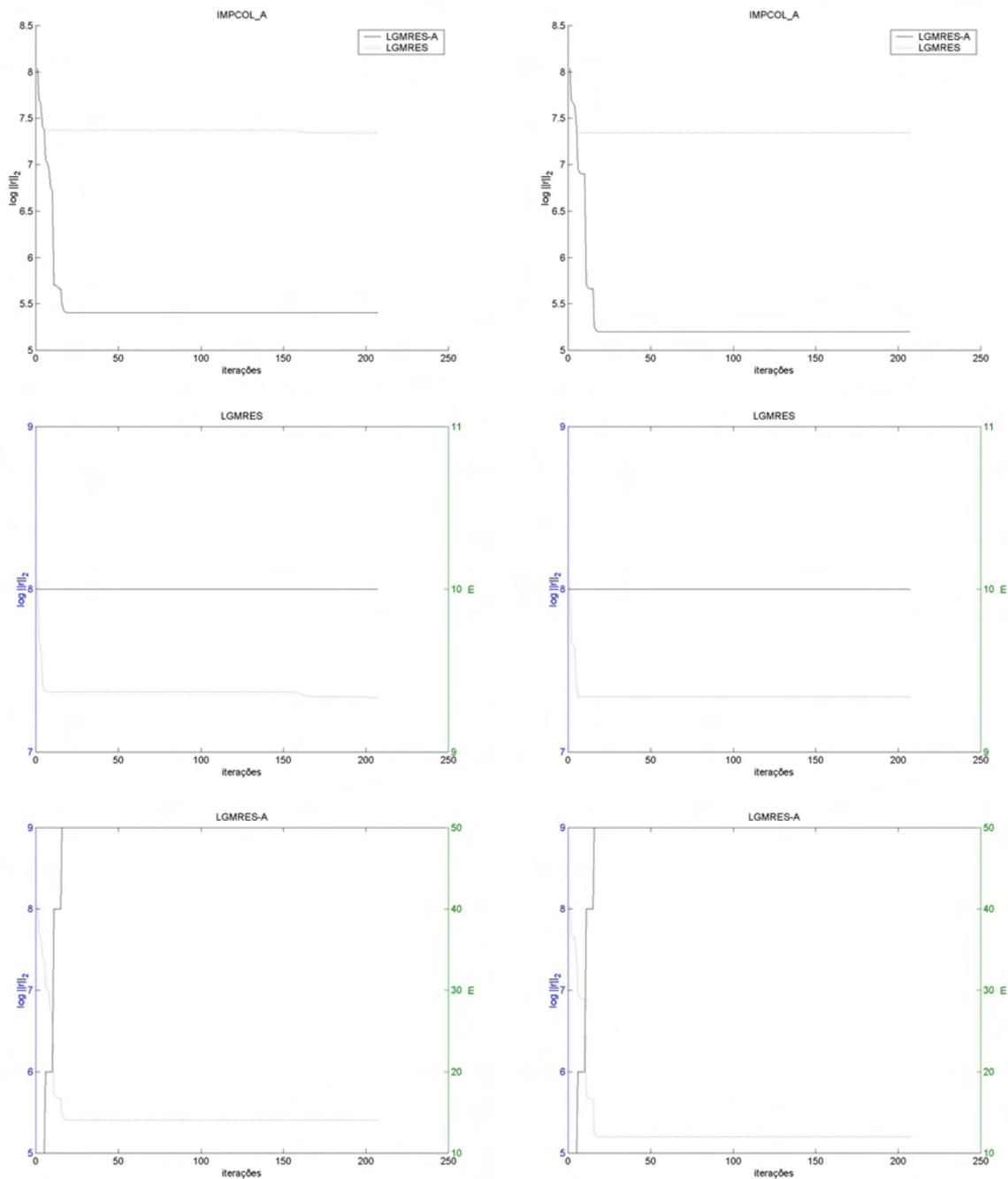


Figura 4.44: IMPCOL_A: Gráfico do resíduo e m versus iterações

Para esse sistema todos os métodos não conseguiram obter a convergência dentro da tolerância pré-especificada inicialmente, então nada pode-se concluir.

4.1.12 JPWH991

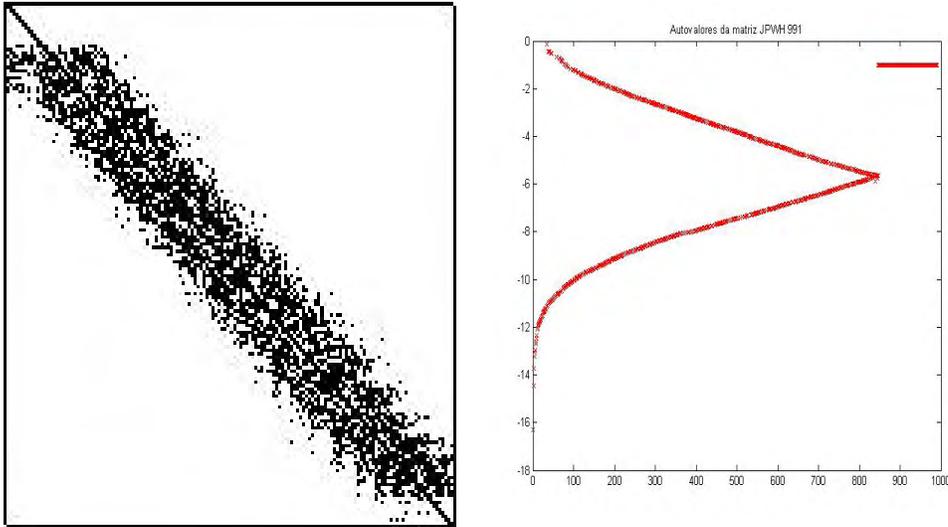


Figura 4.45: JPWH991: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 991×991
- tipo: real não-simétrica
- entradas não nulas: 6,027
- número de condição (estimado): 730
- $\|A\|_2$ estimado: 16

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	12	4	10	$1,4674 \times 10^7$
GMRESH-A	sim	7	10	20	$1,2075 \times 10^7$
SGMRES	sim	25	9	10	$9,8204 \times 10^6$
SGMRES-A	sim	10	10	20	$7,3785 \times 10^6$
SGMRESH	sim	10	8	10	$8,4884 \times 10^6$
SGMRESH-A	sim	7	10	16	$7,1012 \times 10^6$
BC-GMRES	sim	25	1	10	$5,6895 \times 10^6$
BC-GMRES-A	sim	15	1	20	$4,3139 \times 10^6$
LGMRES($m,2$)	sim	9	10	10	$4,1994 \times 10^6$
LGMRES-A($m,2$)	sim	6	10	20	$4,8373 \times 10^6$
LGMRES($m,3$)	sim	7	10	10	$4,9977 \times 10^6$
LGMRES-A($m,3$)	sim	6	10	20	$3,7925 \times 10^6$

Tabela 4.12: JPWH.991

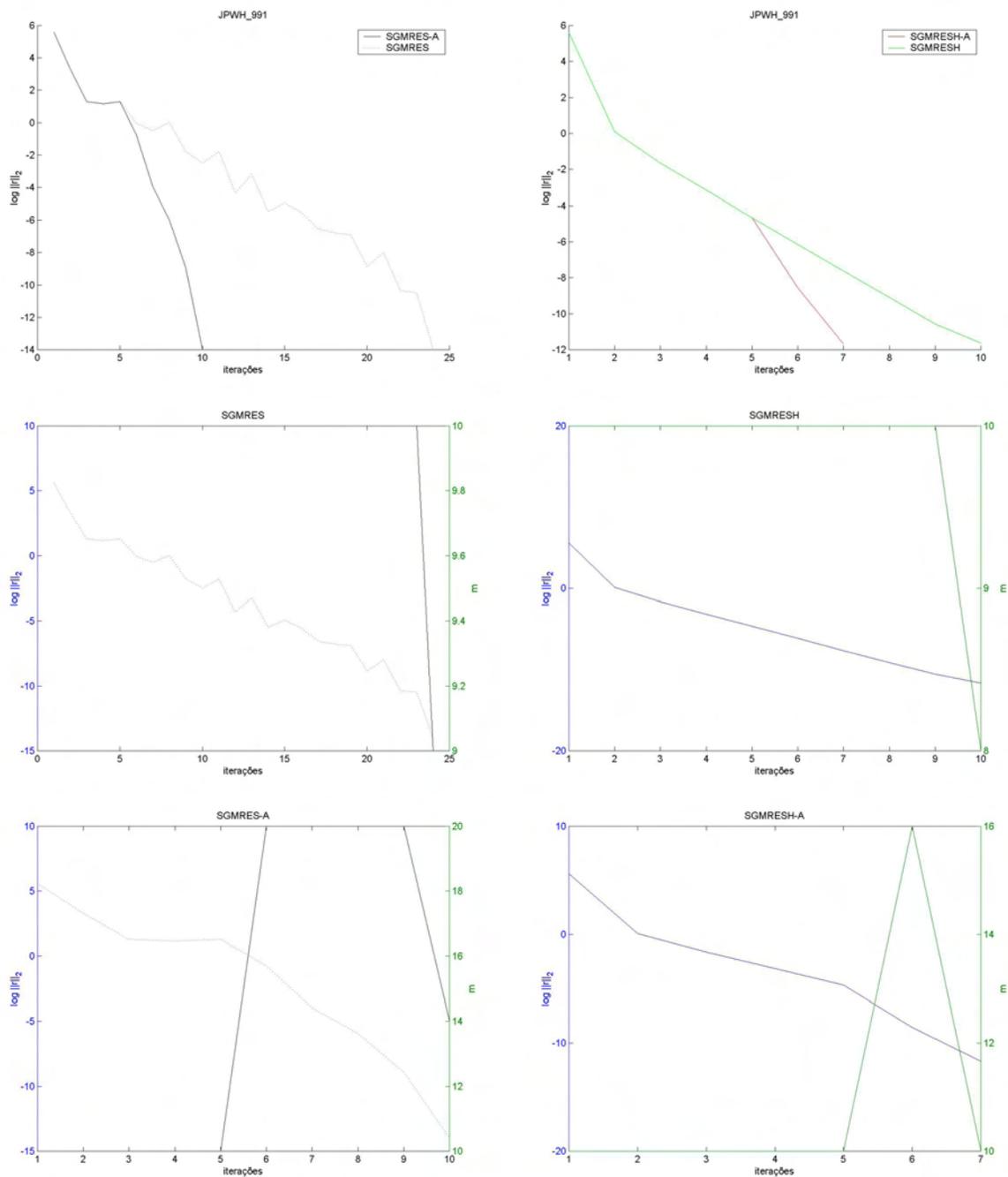


Figura 4.46: JPWH991: Gráfico do resíduo e m versus iterações

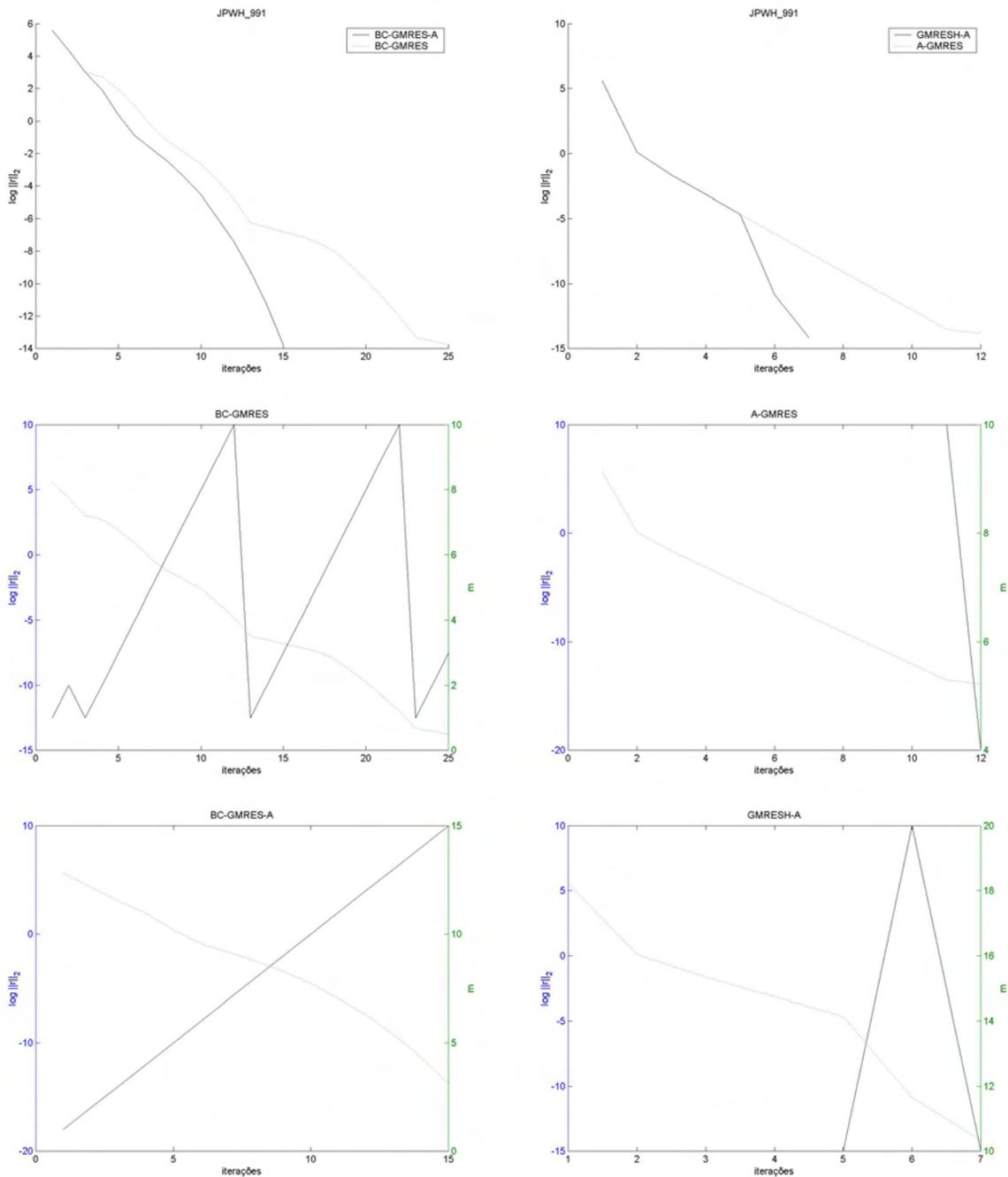


Figura 4.47: JPWH991: Gráfico do resíduo e m versus iterações

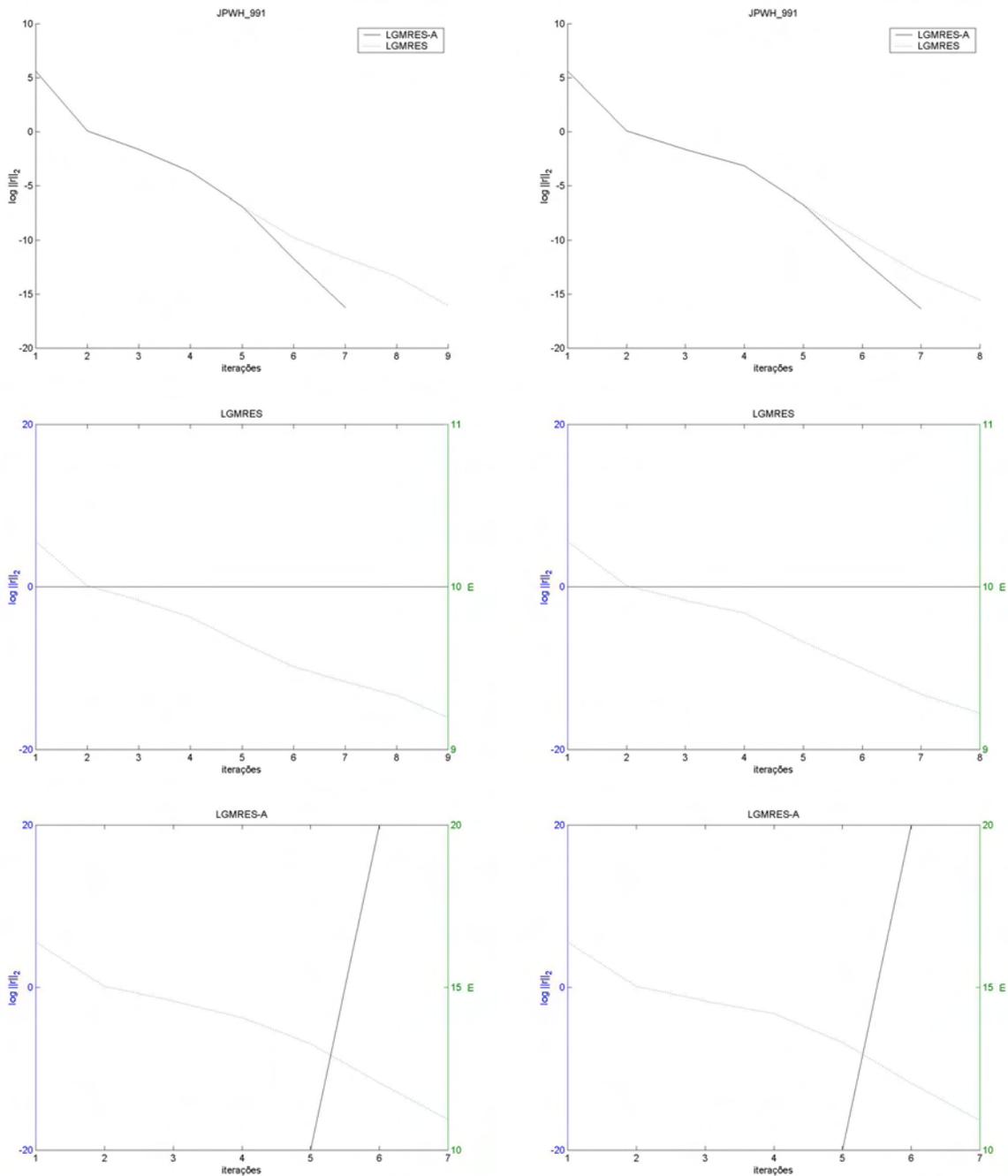


Figura 4.48: JPWH991: Gráfico do resíduo e m versus iterações

Na figura (4.46) note que ambos os métodos convergiram, no entanto o SGMRES-A e o SGMRESH-A apresentaram menor número de flops do que o SGMRES e o SGMRESH. Observe novamente que o decréscimo apresentado no m não deu-se pela presença de alguma critério, e sim pela detecção de um subespaço invariante na formação do subespaço de Krylov. O comportamento do m no SGMRES-A e no SGMRESH-A apresenta uma certa similaridade, ou seja, ambos iniciaram sendo iguais a dez, cresceram até atingir seu valor máximo e decrescendo em seguida até seu valor inicial; a diferença deu-se apenas que o SGMRES-A apresentou um momento de estagnação quando o m atinge seu valor máximo, o que não ocorreu utilizando-se o SGMRESH-A.

Na figura (4.47) note que ambos os métodos convergiram, sendo que o BC-GMRES-A apresentou menor número de flops e de iterações necessárias para obter a convergência do que o BC-GMRES. Os “picos” no gráfico do BC-GMRES em relação ao m segue explicação anterior. O BC-GMRES-A não necessitou de nenhuma reinicialização, visto que o gráfico do m não apresentou nenhum “pico”.

Na figura (4.48) note que ambos os métodos convergiram e apresentaram comportamento similar ao longo das iterações, sendo que o LGMRES($m, 2$) e o LGMRES-A($m, 3$) apresentaram menor número de flops do que o LGMRES-A($m, 2$) e o LGMRES($m, 3$), respectivamente. Observe que o comportamento do m e do logaritmo da norma residual foi praticamente o mesmo, apenas diferindo no número de iterações apresentadas entre o LGMRES($m, 2$) e o LGMRES($m, 3$).

4.1.13 LNS_131

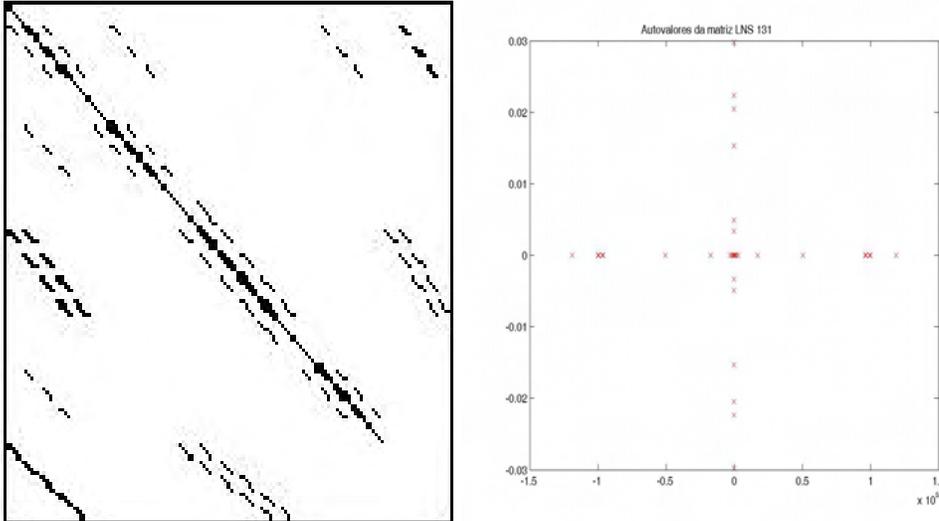


Figura 4.49: LNS_131: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 131×131
- tipo: real não-simétrica
- entradas não nulas: 536
- número de condição (estimado): $1,5 \times 10^{15}$
- $\|A\|_2$ estimado: $9,8 \times 10^9$

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	não	-	10	10	$2,3689 \times 10^7$
GMRESH-A	sim	140	10	50	$3,8550 \times 10^8$
SGMRES	não	-	10	10	$6,6890 \times 10^7$
SGMRES-A	sim	140	10	50	$6,7305 \times 10^8$
SGMRESH	não	-	10	10	$1,5848 \times 10^7$
SGMRESH-A	sim	140	10	50	$2,3803 \times 10^8$
BC-GMRES	não	-	1	10	$1,3590 \times 10^7$
BC-GMRES-A	não	-	1	50	$3,3743 \times 10^6$
LGMRES($m,2$)	não	-	10	10	$9,0154 \times 10^6$
LGMRES-A($m,2$)	não	-	10	50	$1,0539 \times 10^8$
LGMRES($m,3$)	não	-	10	10	$1,0174 \times 10^7$
LGMRES-A($m,3$)	não	-	10	50	$1,0923 \times 10^8$

Tabela 4.13: LNS_131

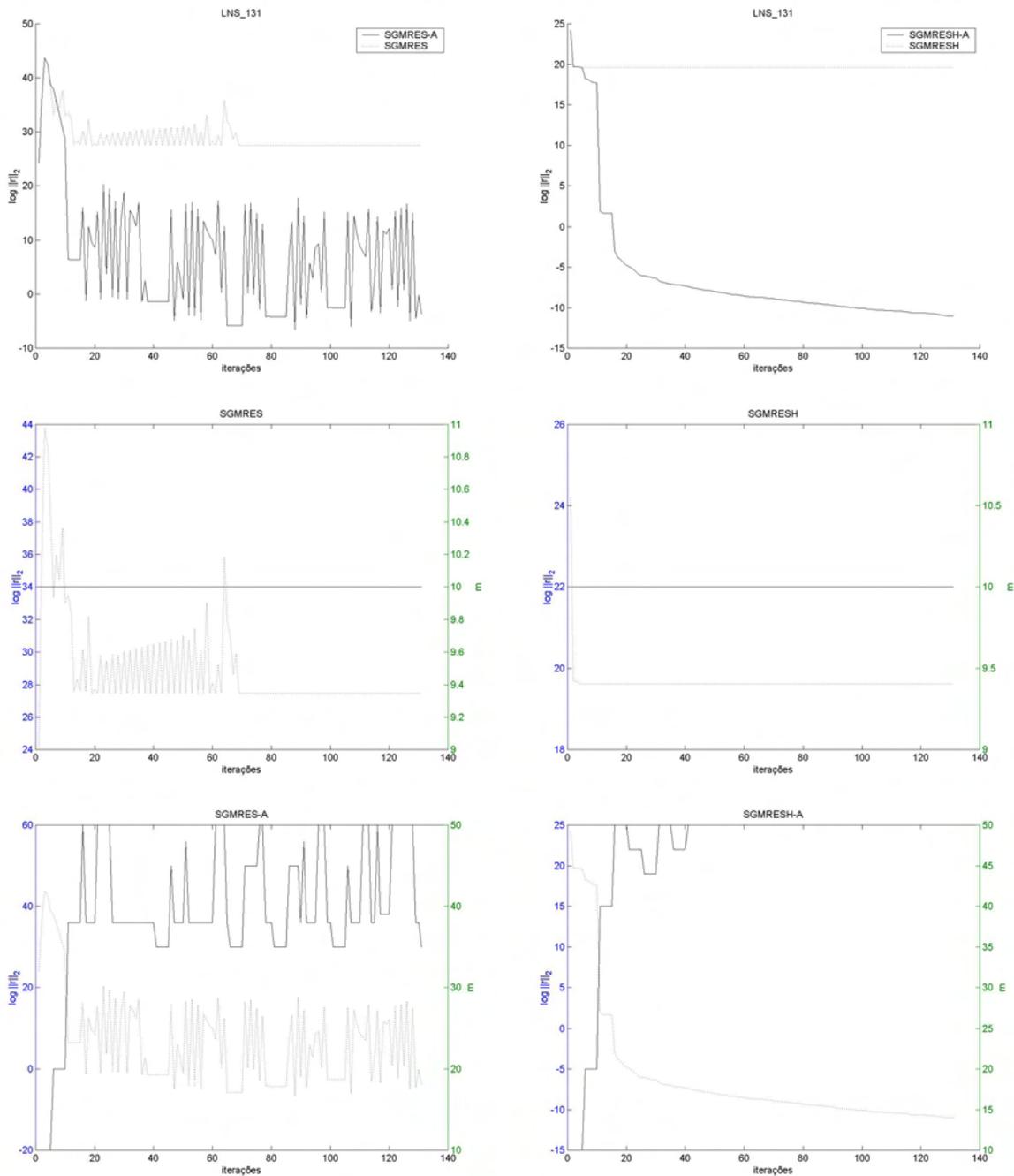


Figura 4.50: LNS_131: Gráfico do resíduo e m versus iterações

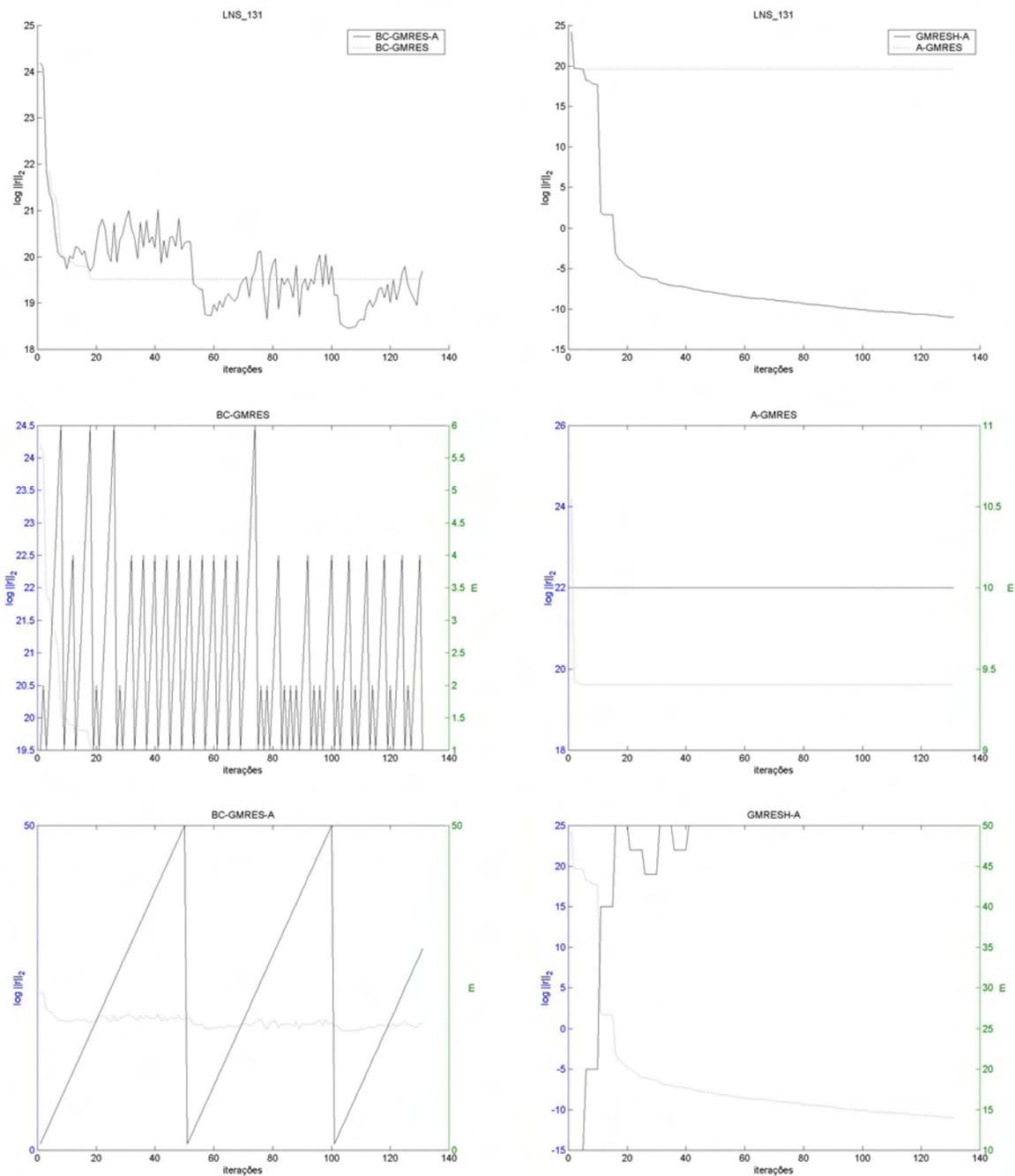


Figura 4.51: LNS_131: Gráfico do resíduo e m versus iterações

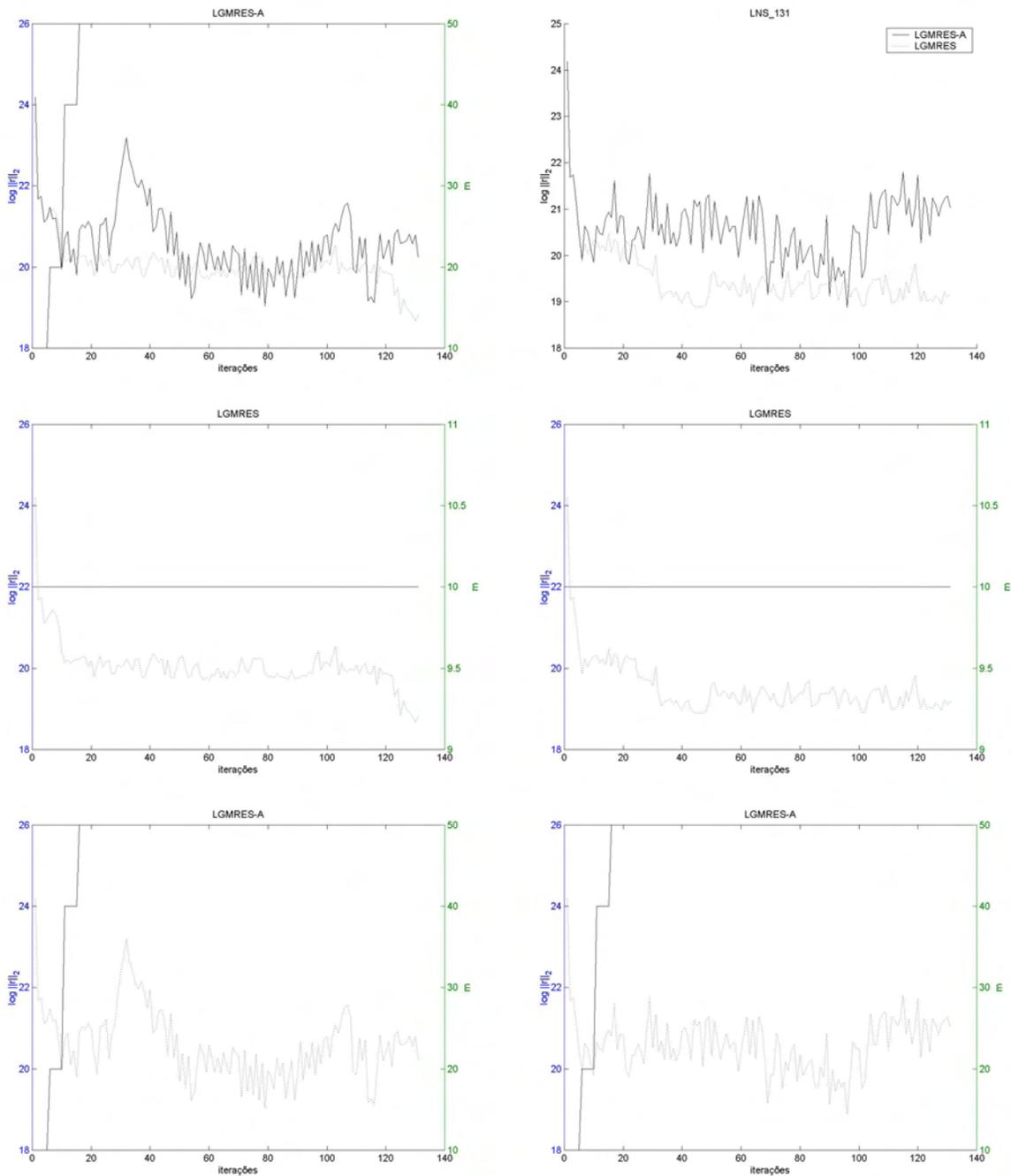


Figura 4.52: LNS_131: Gráfico do resíduo e m versus iterações

Na figura (4.50) note que apenas o SGMRES-A e o SGMRESH-A obtiveram a convergência dentro da tolerância pré-especificada. Observe que o SGMRES e o SGMRESH apresentaram estagnação após algumas iterações e o m não apresentou variação ao longo das iterações. Note também o comportamento oscilante do m no SGMRES-A.

Na figura (4.51) apenas o GMRESH-A obteve a convergência dentro da tolerância pré-especificada inicialmente, e o A-GMRES apresentou-se estagnado ao longo das iterações.

Na figura (4.52) note que ambos os métodos não conseguiram obter a convergência dentro da tolerância pré-especificada.

4.1.14 NOS3

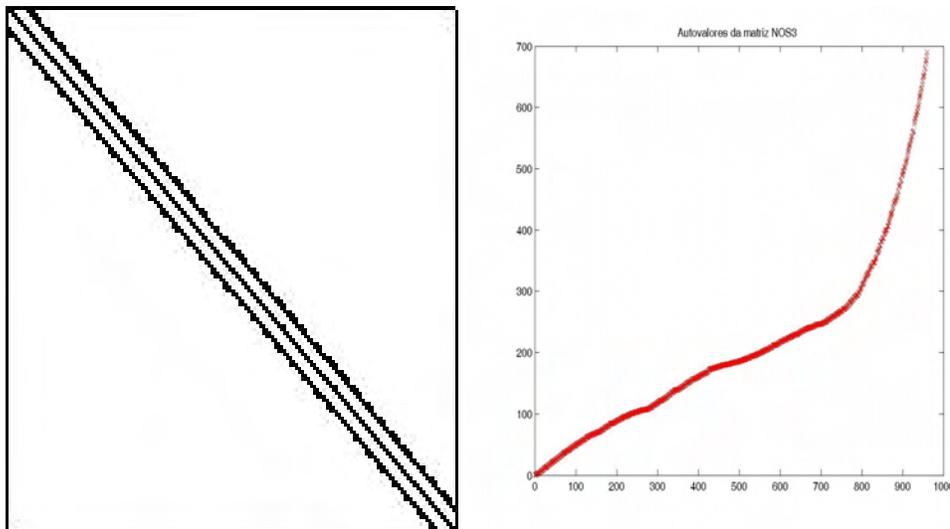


Figura 4.53: NOS3: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 960×960
- tipo: real simétrica definida positiva
- entradas não nulas: 15844
- número de condição (estimado): $7,3 \times 10^4$

- $\|A\|_2$ estimado: 680

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	1000	10	10	$1,5413 \times 10^9$
GMRESH-A	sim	45	4	50	$3,5546 \times 10^8$
SGMRES	sim	1000	10	10	$6,0951 \times 10^8$
SGMRES-A	sim	60	10	60	$2,9282 \times 10^8$
SGMRESH	sim	1000	10	10	$1,1159 \times 10^9$
SGMRESH-A	sim	80	10	40	$3,4514 \times 10^8$
BC-GMRES	sim	1000	1	10	$3,2530 \times 10^8$
BC-GMRES-A	sim	150	1	60	$2,8910 \times 10^8$
LGMRES($m,2$)	sim	60	10	10	$5,3196 \times 10^7$
LGMRES-A($m,2$)	sim	40	10	25	$4,3745 \times 10^7$
LGMRES($m,3$)	sim	45	10	10	$4,6488 \times 10^7$
LGMRES-A($m,3$)	sim	35	10	25	$3,8401 \times 10^7$

Tabela 4.14: NOS3

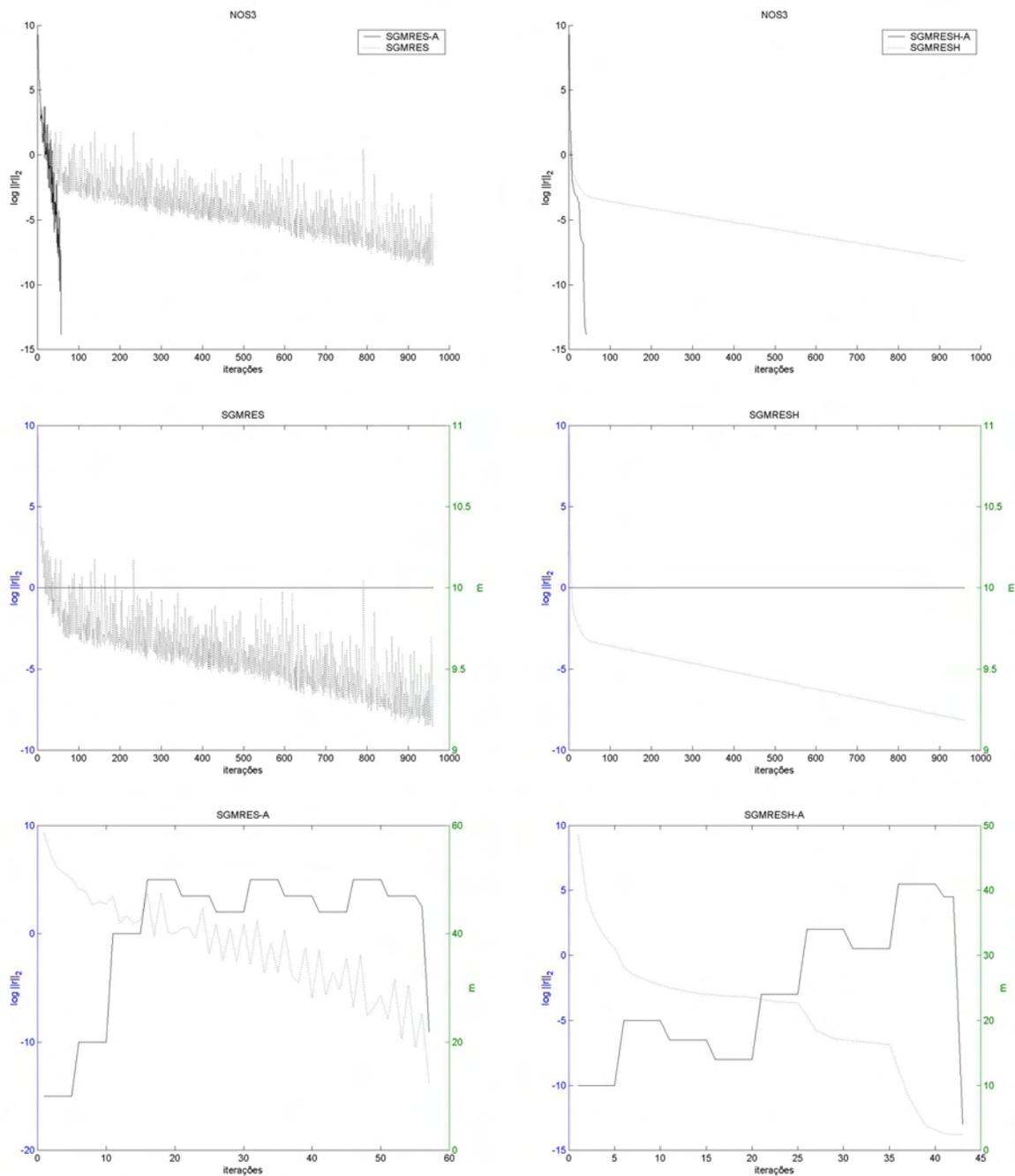


Figura 4.54: NOS3: Gráfico do resíduo e m versus iterações

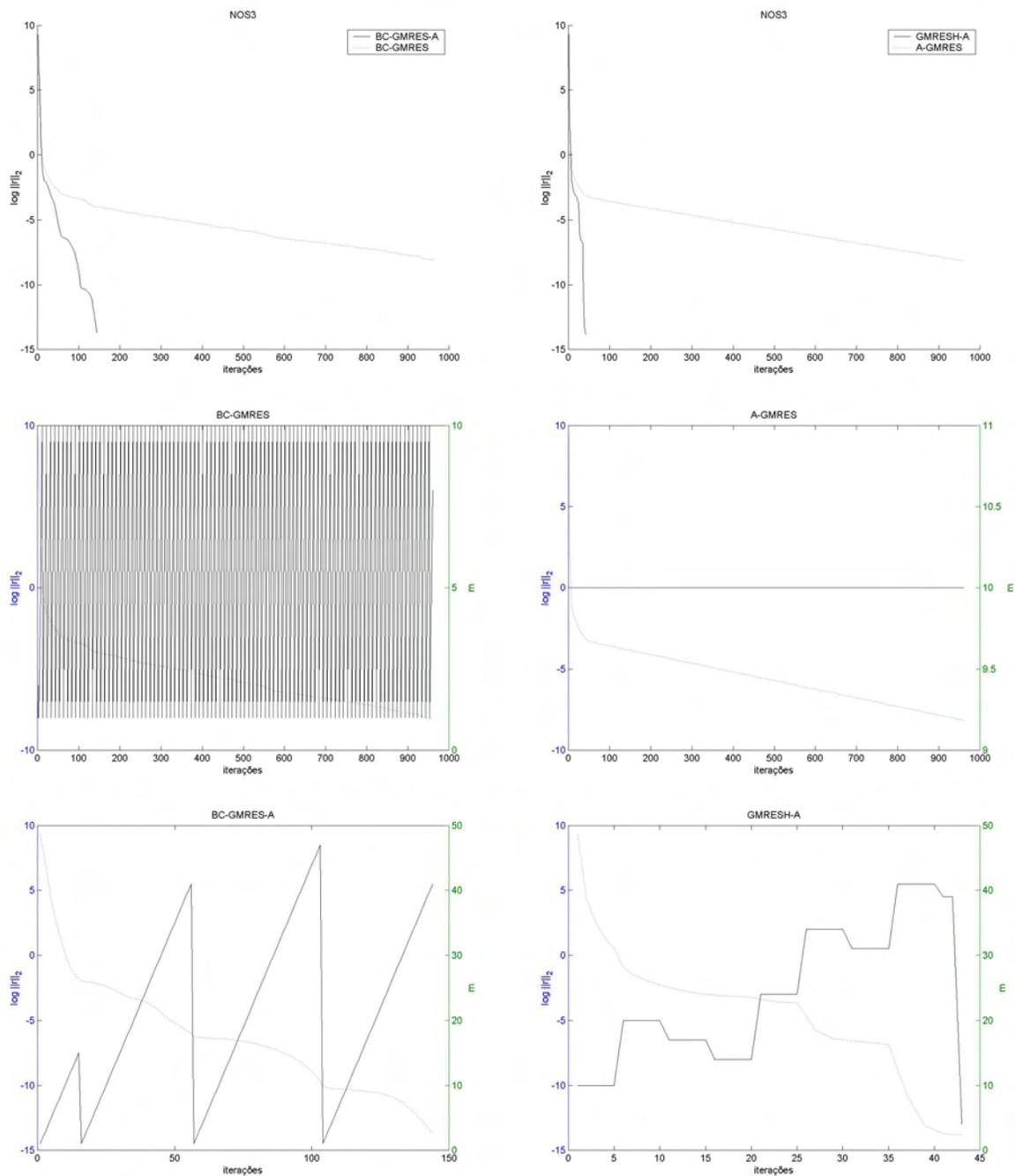


Figura 4.55: NOS3: Gráfico do resíduo e m versus iterações

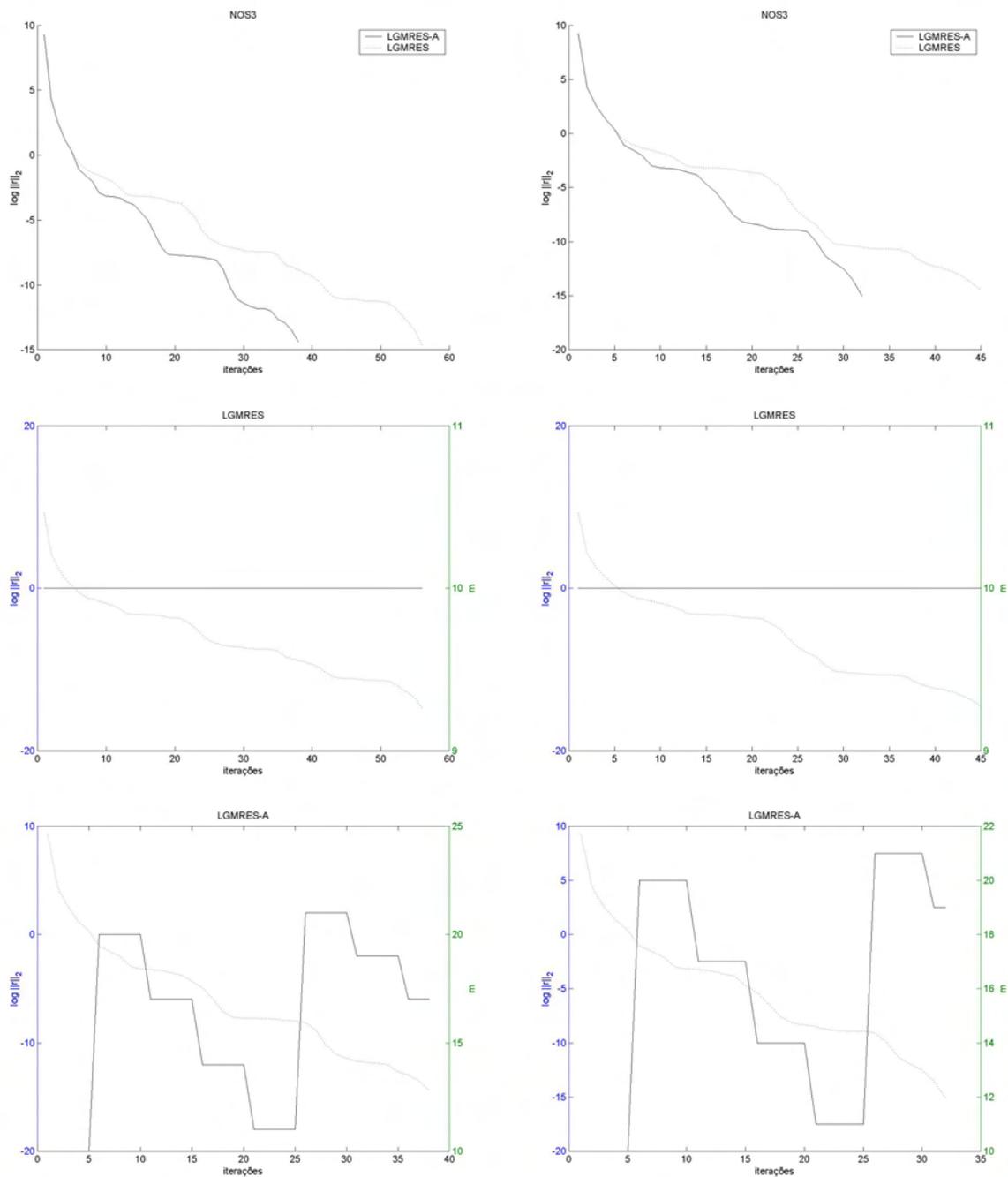


Figura 4.56: NOS3: Gráfico do resíduo e m versus iterações

Na figura (4.54) note que ambos os métodos convergiram, sendo que o SGMRES-A e o SGMRESH-A apresentaram menor número de flops e de iterações necessárias para obter a convergência. Observe o grande número de oscilações no logaritmo da norma residual do SGMRES. Novamente o m não apresentou variação ao longo das iterações para o SGMRES e o SGMRESH.

Na figura (4.55) note que ambos os métodos convergiram, sendo que o BC-GMRES-A apresentou menor número de flops e de iterações necessárias para obter a convergência dentro da tolerância pré-especificada inicialmente do que o BC-GMRES. Observe o grande número de reinicializações necessárias para obter a convergência no BC-GMRES, contra as poucas reinicializações necessárias no BC-GMRES-A.

Na figura (4.56), note que os gráficos apresentaram uma certa similaridade entre o LGMRES(m, k) e o LGMRES-A(m, k); até mesmo o m apresentou-se de forma similar entre os métodos. O LGMRES-A($m, 2$) e o LGMRES-A($m, 3$) apresentaram menor número de flops do que o LGMRES($m, 2$) e o LGMRES($m, 3$).

4.1.15 ORSIRR1

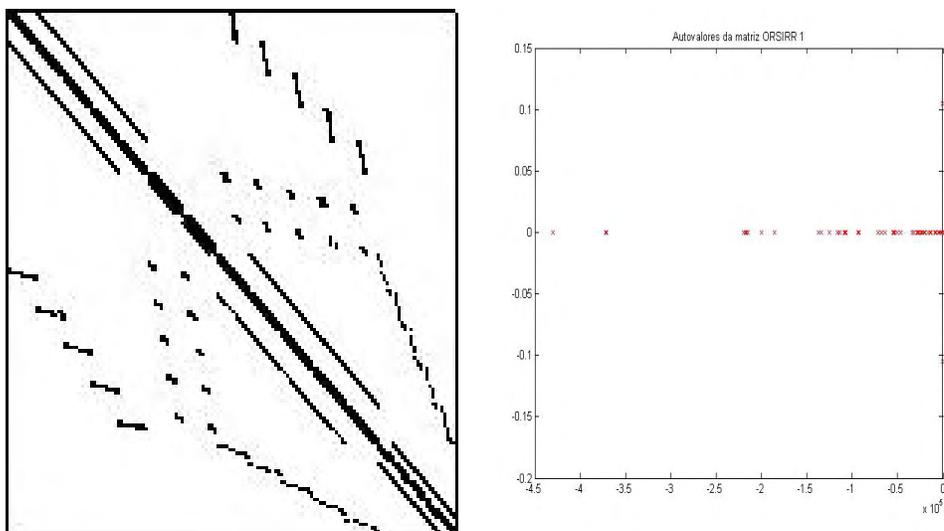


Figura 4.57: ORSIRR1: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 1030×1030

- tipo: real não-simétrica
- entradas não nulas: 6858
- número de condição (estimado): 100
- $\|A\|_2$ estimado: $4,6 \times 10^5$

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	1200	10	10	$1,5447 \times 10^9$
GMRESH-A	sim	80	10	50	$1,5362 \times 10^9$
SGMRES	não	-	10	10	$4,7179 \times 10^8$
SGMRES-A	sim	80	10	50	$3,7563 \times 10^8$
SGMRESH	sim	1100	10	10	$1,0549 \times 10^9$
SGMRESH-A	sim	90	10	50	$9,3249 \times 10^8$
BC-GMRES	não	-	1	10	$2,9748 \times 10^8$
BC-GMRES-A	sim	180	1	50	$1,5805 \times 10^8$
LGMRES($m,2$)	sim	250	10	10	$2,0414 \times 10^8$
LGMRES-A($m,2$)	sim	50	10	50	$1,3435 \times 10^8$
LGMRES($m,3$)	sim	250	10	10	$2,0895 \times 10^8$
LGMRES-A($m,3$)	sim	50	10	50	$1,5225 \times 10^8$

Tabela 4.15: ORSIRR_1

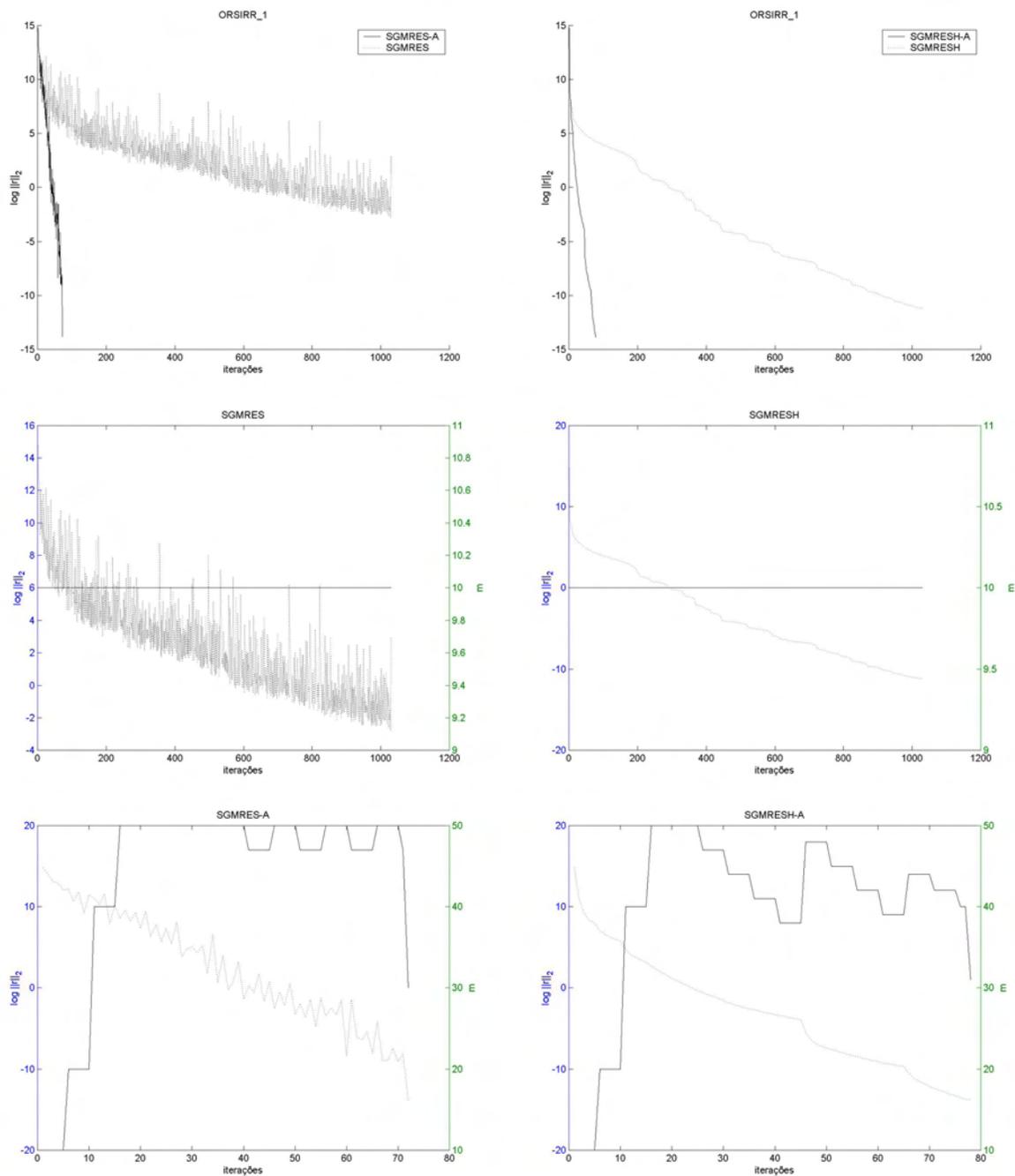


Figura 4.58: ORSIRR1: Gráfico do resíduo e m versus iterações

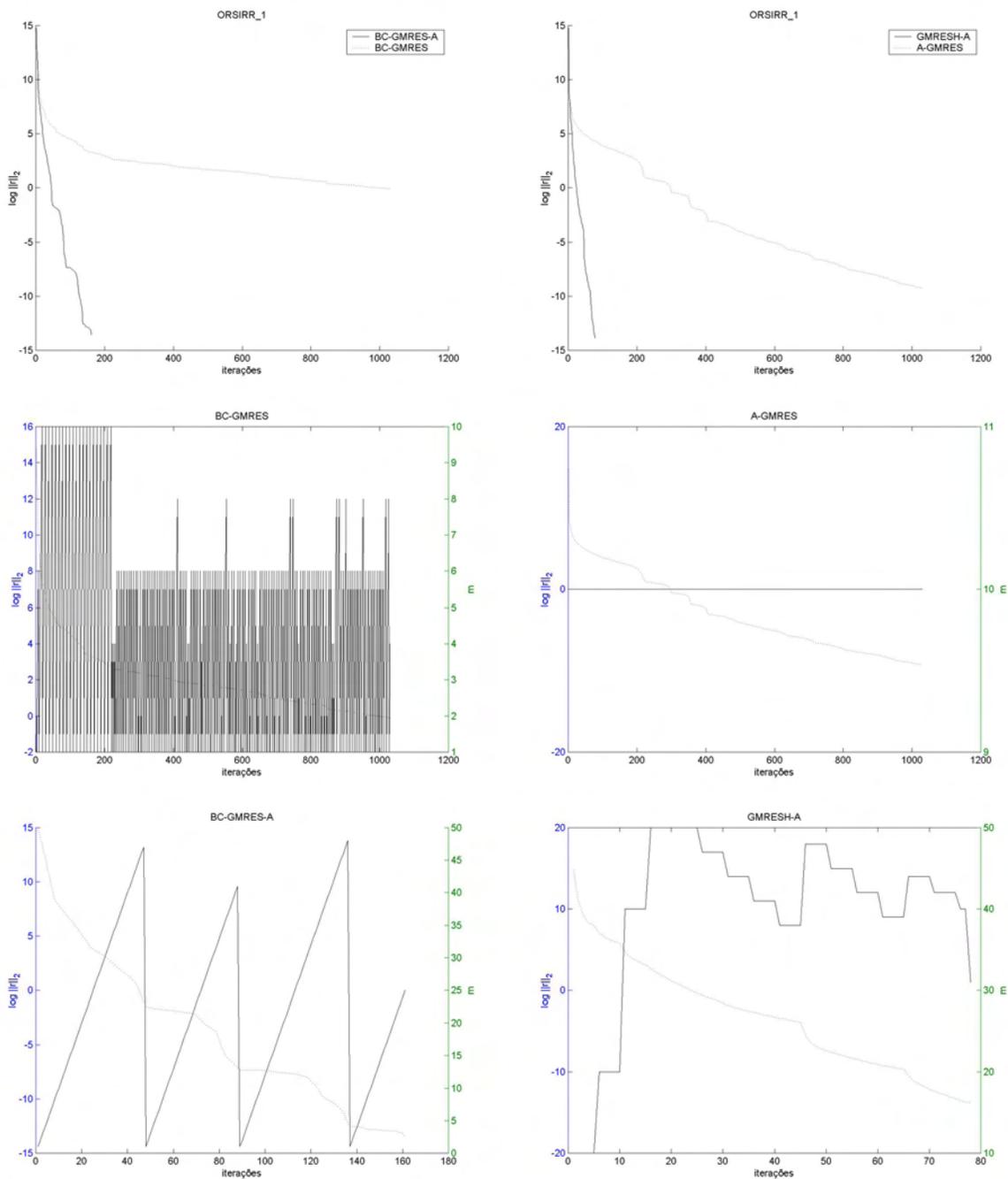


Figura 4.59: ORSIRR1: Gráfico do resíduo e m versus iterações

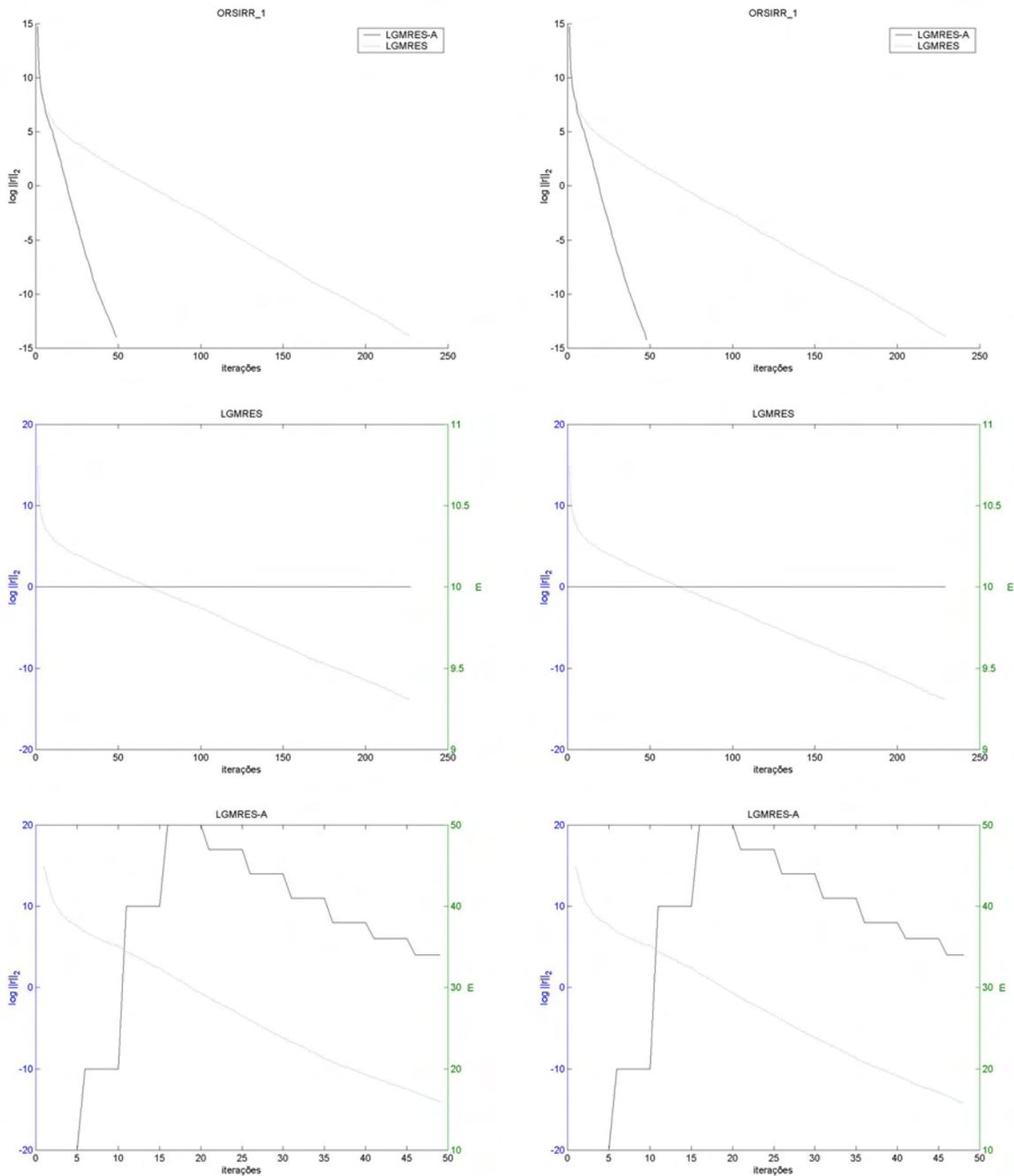


Figura 4.60: ORSIRR1: Gráfico do resíduo e m versus iterações

Na figura (4.58) note que apenas o SGMRES não conseguiu obter a convergência dentro da tolerância pré-especificada. Entre o SGMRESH e o SGMRESH-A, o SGMRESH-A apresentou menor número de flops e de iterações necessárias para obter a convergência do que o SGMRESH.

Na figura (4.59) apenas o BC-GMRES não obteve a convergência dentro da tolerância pré-especificada. Em relação ao A-GMRES e o GMRESH-A, o GMRESH-A apresentou menor número de flops e de iterações para obter a convergência do que o A-GMRES. Observe também que o m não apresentou variação ao longo das iterações para o A-GMRES.

Observe a figura (4.60): note que os gráficos apresentam o mesmo comportamento ao longo das iterações, até mesmo em relação ao m . O LGMRES-A($m, 2$) e o LGMRES-A($m, 3$) apresentaram menor número de flops do que o LGMRES($m, 2$) e o LGMRES($m, 3$).

4.1.16 PDE900

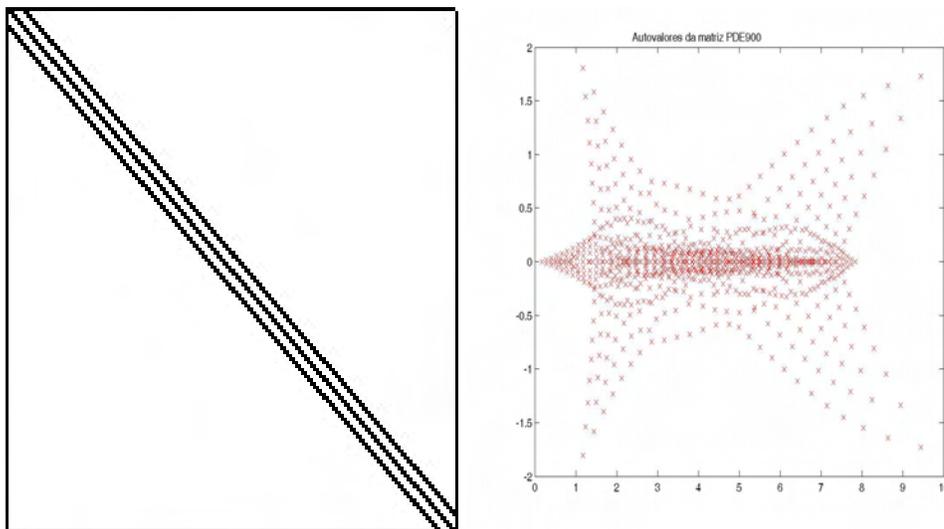


Figura 4.61: PDE900: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 900×900
- tipo: real não-simétrica

- entradas não nulas: 4380
- número de condição (estimado): 290
- $\|A\|_2$ estimado: 10

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	18	7	10	$2,1118 \times 10^7$
GMRESH-A	sim	12	10	20	$3,1990 \times 10^7$
SGMRES	sim	30	6	10	$1,0056 \times 10^6$
SGMRES-A	sim	15	5	20	$1,0046 \times 10^6$
SGMRESH	sim	16	4	10	$1,2259 \times 10^7$
SGMRESH-A	sim	12	5	20	$1,2105 \times 10^7$
BC-GMRES	sim	70	1	10	$6,2225 \times 10^6$
BC-GMRES-A	sim	25	1	20	$5,9397 \times 10^6$
LGMRES($m,2$)	sim	18	10	10	$1,1222 \times 10^7$
LGMRES-A($m,2$)	sim	12	10	20	$8,3847 \times 10^6$
LGMRES($m,3$)	sim	18	10	10	$8,6616 \times 10^6$
LGMRES-A($m,3$)	sim	11	10	20	$1,0765 \times 10^7$

Tabela 4.16: PDE_900

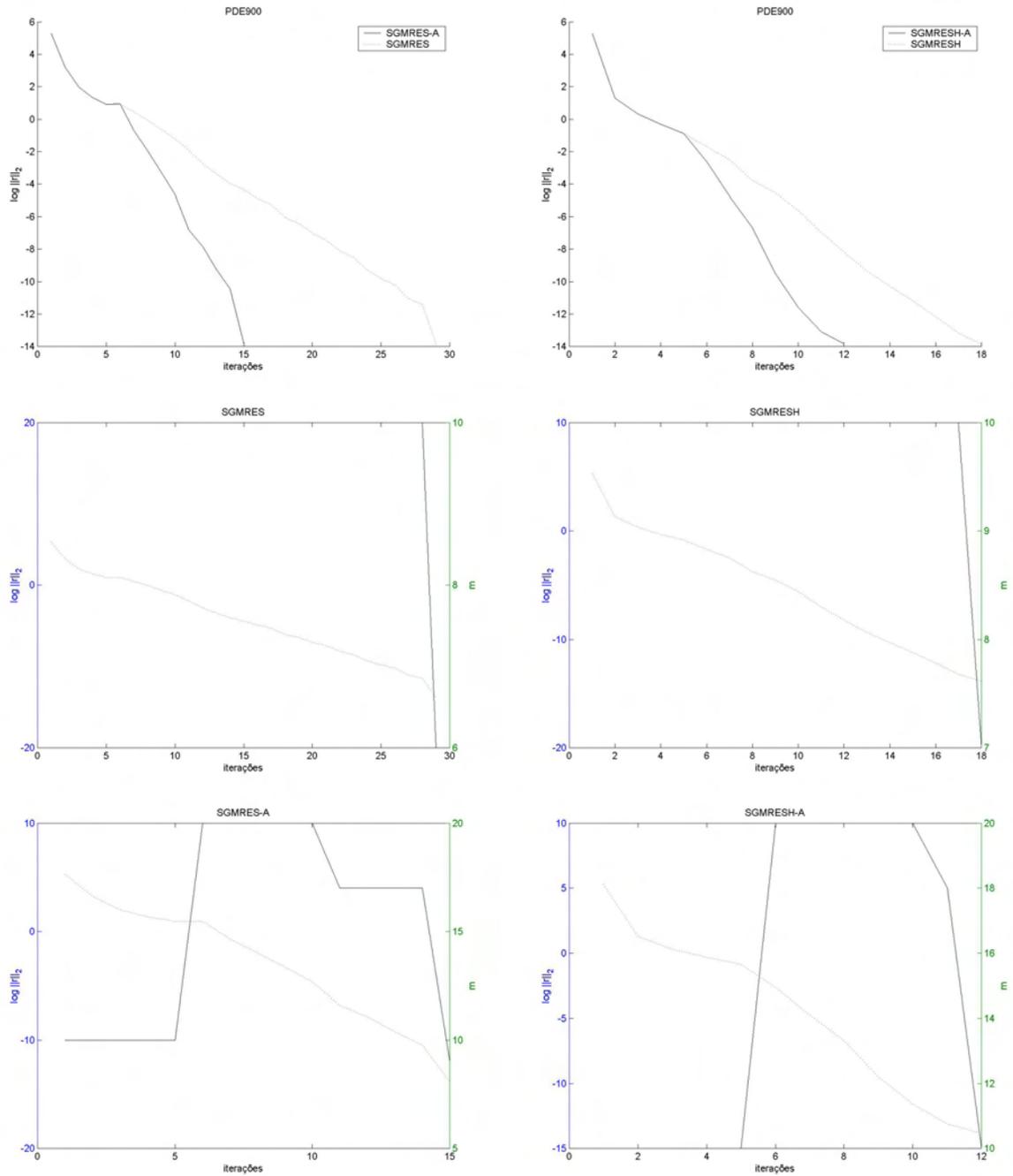


Figura 4.62: PDE900: Gráfico do resíduo e m versus iterações

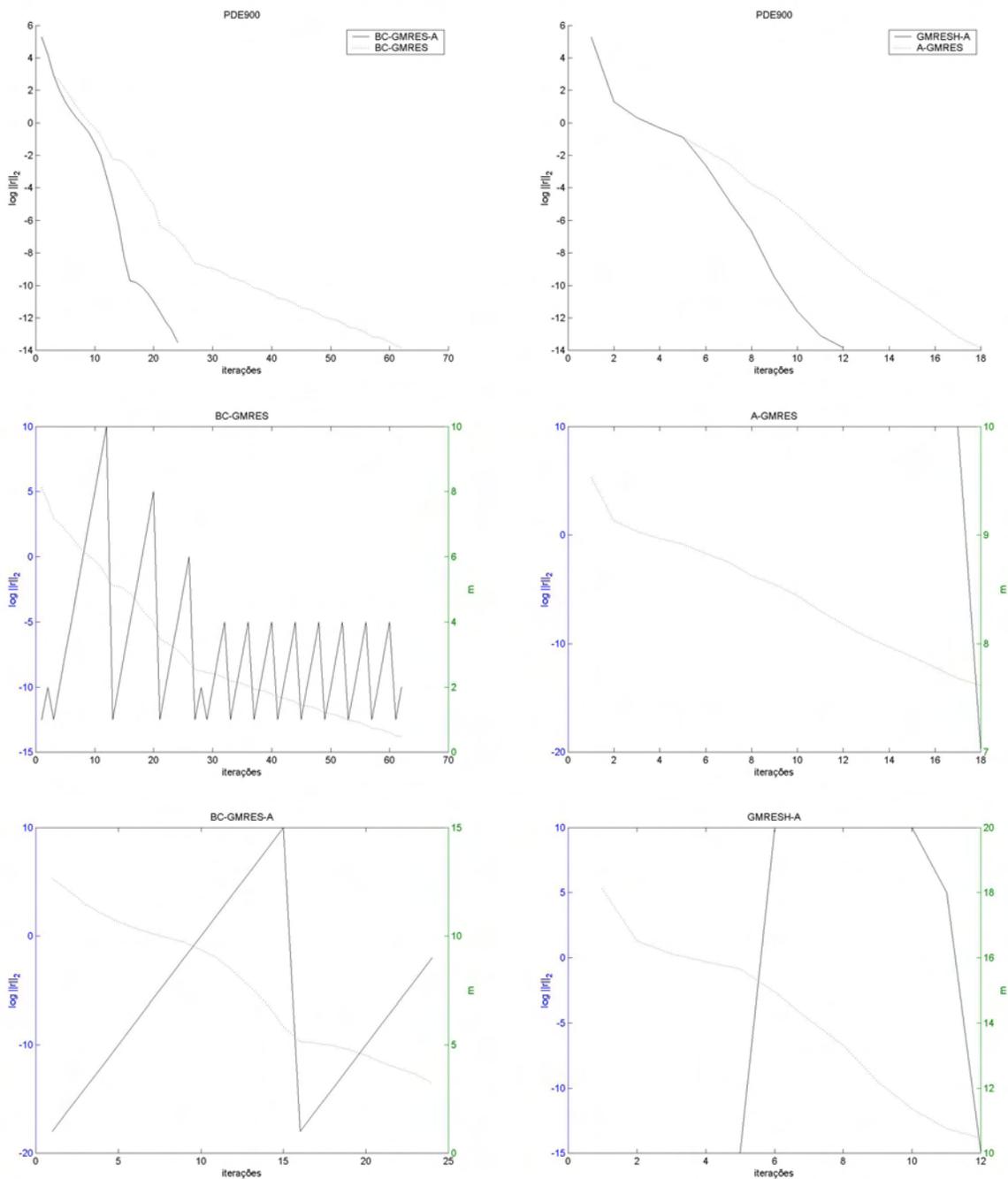


Figura 4.63: PDE900: Gráfico do resíduo e m versus iterações

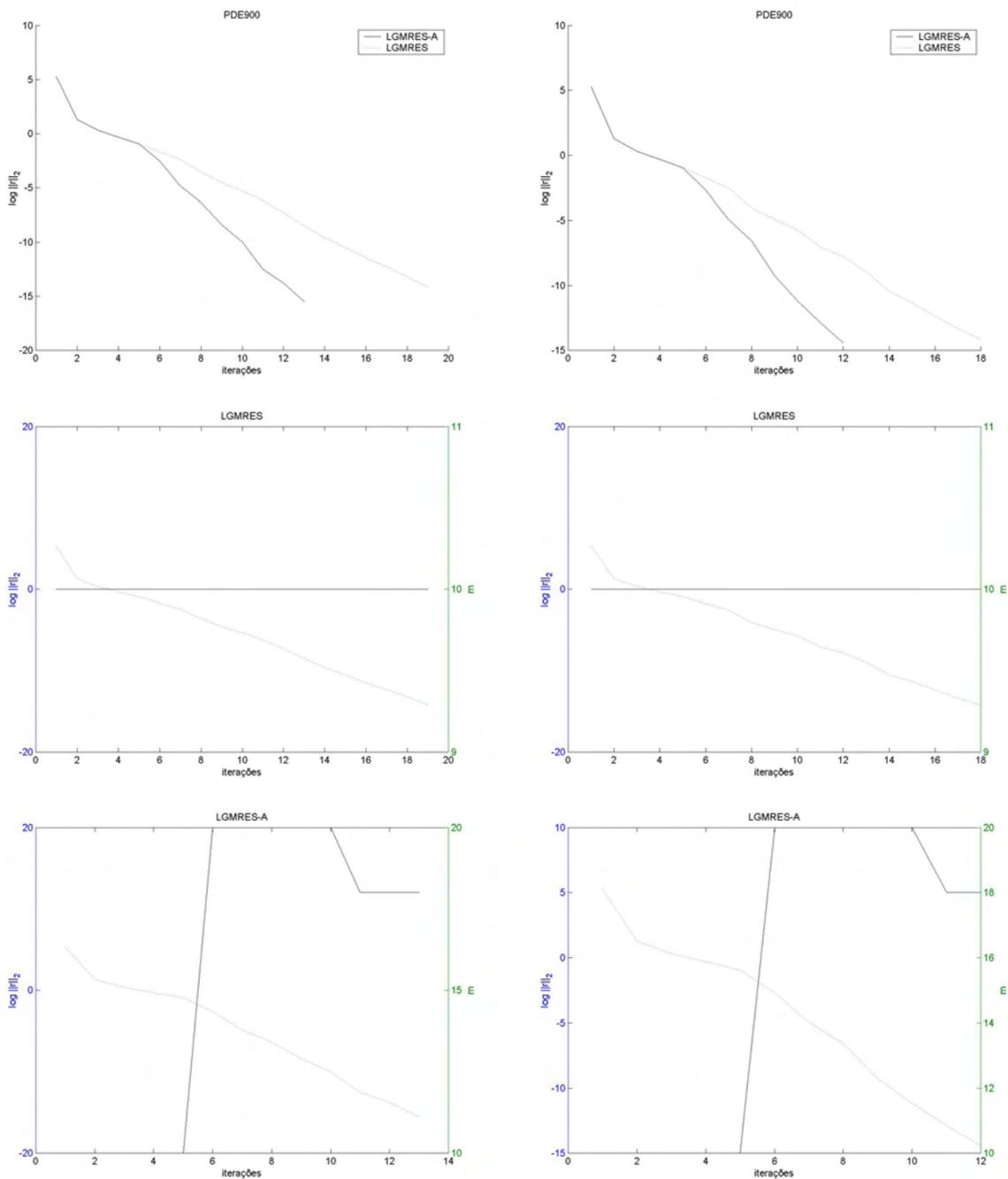


Figura 4.64: PDE900: Gráfico do resíduo e m versus iterações

Na figura (4.62) note que ambos os métodos convergiram, sendo que o SGMRES-A e o SGMRESH-A apresentaram menor número de flops do que o SGMRES e o SGMRESH. Observe que o comportamento do SGMRES e do SGMRESH apresenta uma certa similaridade, novamente ocorreu a detecção de um subespaço invariante na formação do subespaço de Krylov.

Na figura (4.63) observe que ambos os métodos convergiram, no entanto o BC-GMRES-A apresentou menor número de flops e de iterações necessárias para obter a convergência do que o BC-GMRES. Note o número de reinicializações necessárias no BC-GMRES e no BC-GMRES-A.

Na figura (4.64) note que ambos os métodos convergiram, sendo que o LGMRES-A($m, 2$) e o LGMRES-A($m, 3$) apresentaram menor número de flops do que o LGMRES($m, 2$) e o LGMRES($m, 3$). Observe que os gráficos apresentados na figura (4.64) são muito similares.

4.1.17 SAYLR3

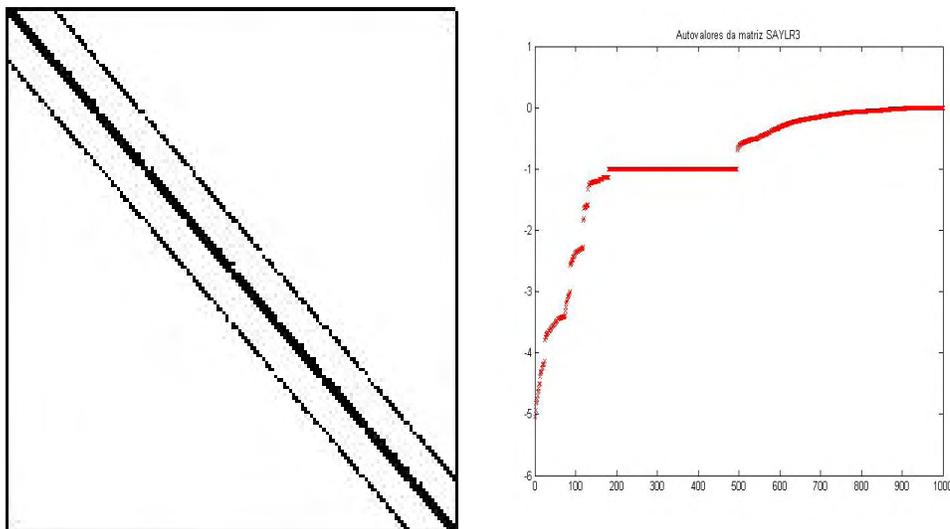


Figura 4.65: SAYLR3: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 1000×1000
- tipo: real não-simétrica

- entradas não nulas: 3750
- número de condição (estimado): 100
- $\|A\|_2$ estimado: 5

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	300	2	10	$3,9817 \times 10^8$
GMRESH-A	sim	47	10	30	$1,8778 \times 10^8$
SGMRES	sim	300	9	10	$1,1190 \times 10^8$
SGMRES-A	sim	45	10	35	$6,2846 \times 10^7$
SGMRESH	sim	140	7	10	$1,2613 \times 10^8$
SGMRESH-A	sim	45	5	20	$6,4821 \times 10^7$
BC-GMRES	sim	300	1	10	$5,2762 \times 10^7$
BC-GMRES-A	sim	80	1	60	$5,0573 \times 10^7$
LGMRES($m,2$)	sim	50	10	10	$2,2879 \times 10^7$
LGMRES-A($m,2$)	sim	30	10	20	$2,2473 \times 10^7$
LGMRES($m,3$)	sim	50	10	10	$3,0844 \times 10^7$
LGMRES-A($m,3$)	sim	30	10	20	$2,5633 \times 10^7$

Tabela 4.17: SAYLR.03

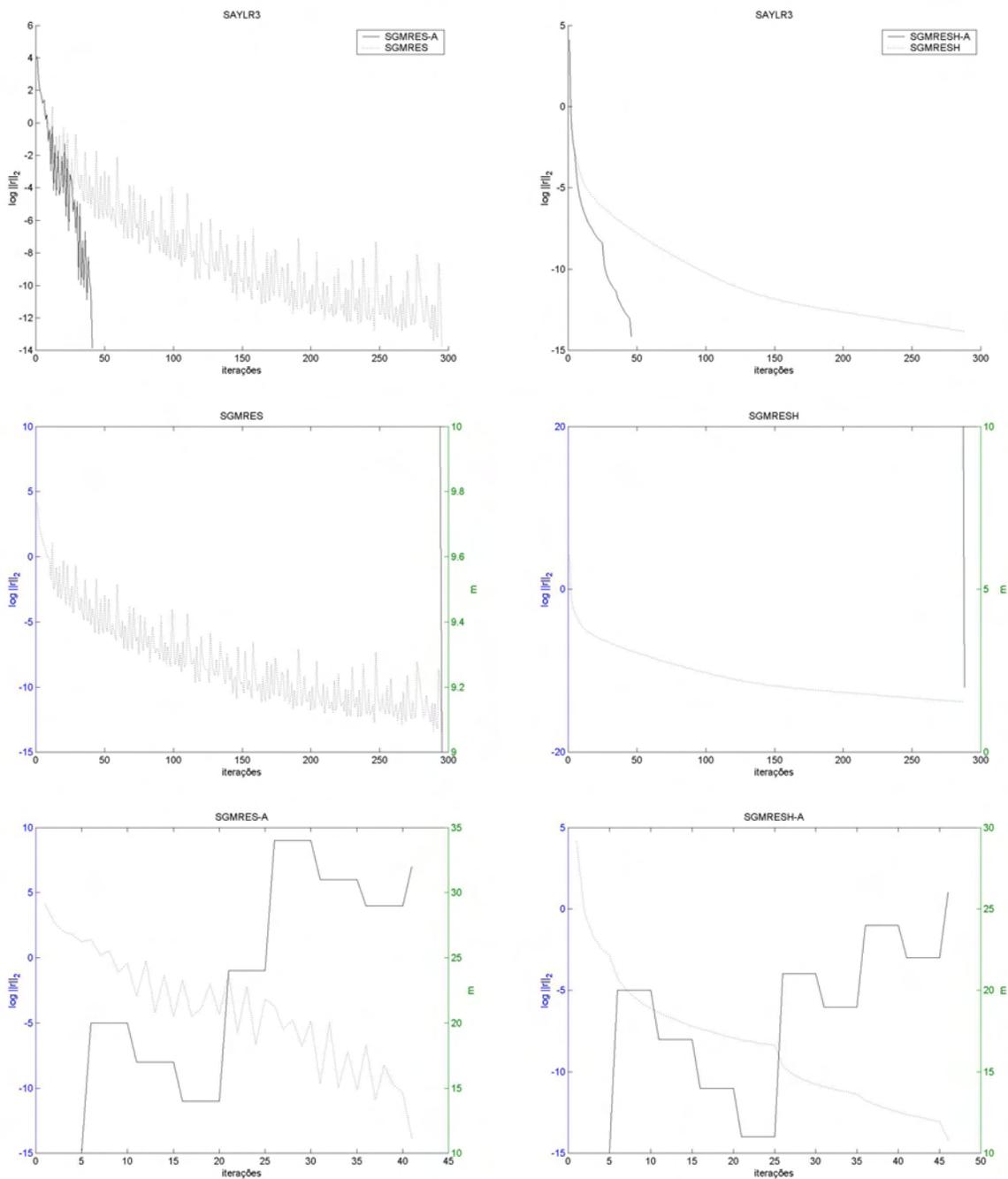


Figura 4.66: SAYLR3: Gráfico do resíduo e m versus iterações

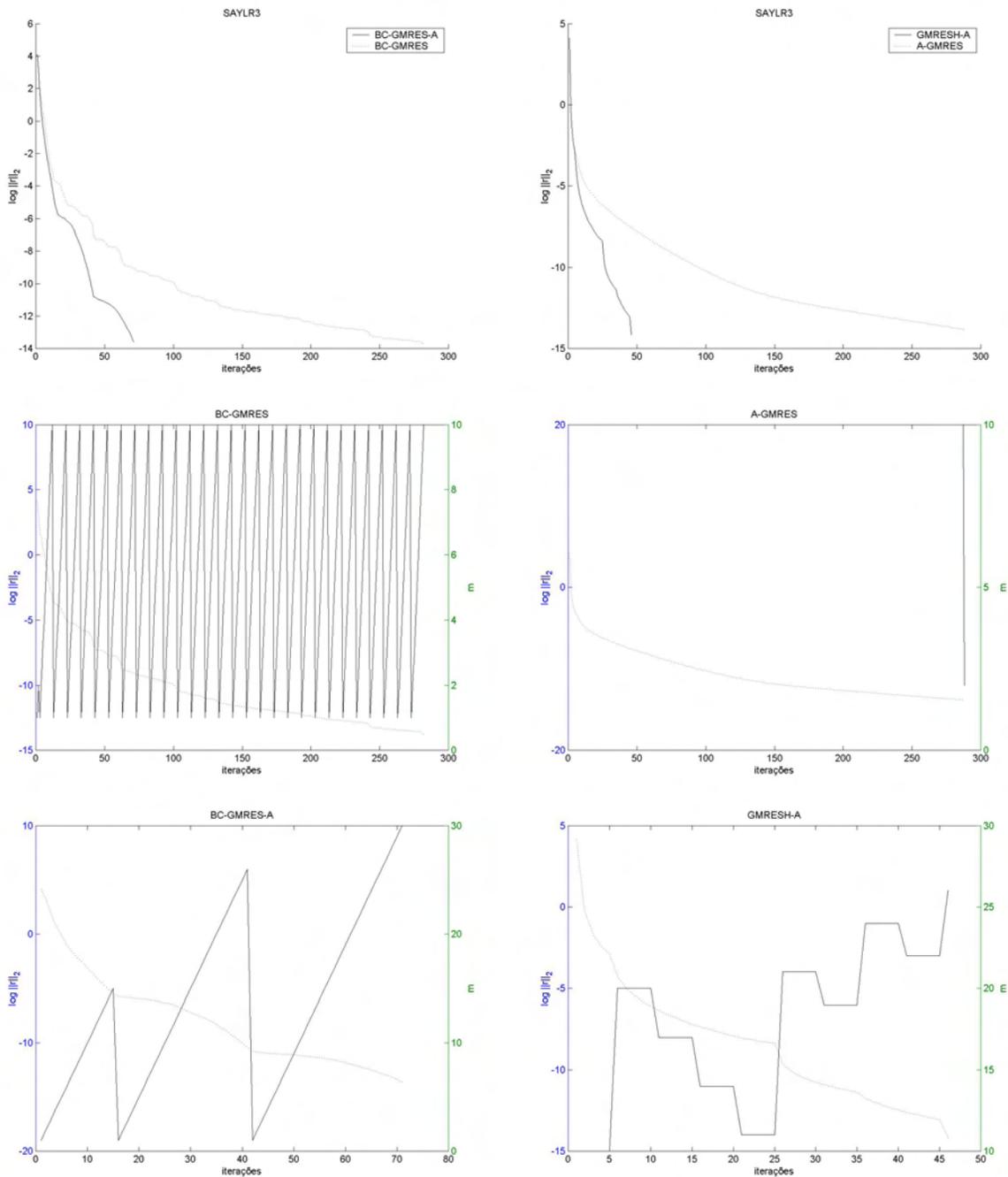


Figura 4.67: SAYLR3: Gráfico do resíduo e m versus iterações

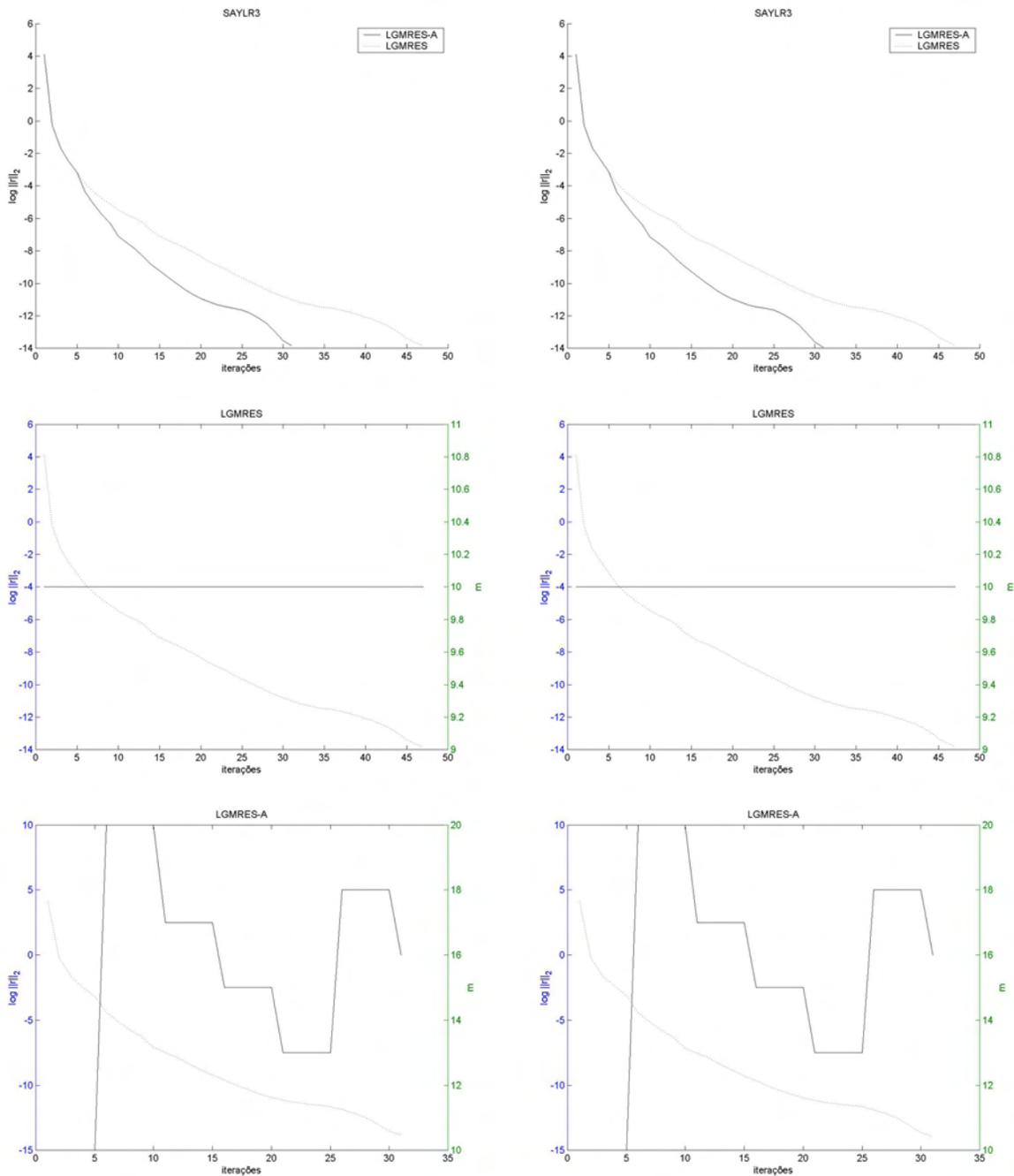


Figura 4.68: SAYLR3: Gráfico do resíduo e m versus iterações

Na figura (4.66) note que ambos os métodos convergiram, sendo que o SGMRES-A e o SGMRESH-A apresentaram menor número de flops e de iterações necessárias para obter a convergência do que o SGMRES e o SGMRESH, respectivamente. Novamente ocorreu o decréscimo do m no SGMRES e no SGMRESH, isso, conforme explicado anteriormente deu-se pela detecção de um subespaço invariante na formação do subespaço de Krylov.

Na figura (4.67) note que ambos os métodos obtiveram a convergência sendo que o BC-GMRES-A apresentou menor número de flops e de iterações necessárias para obter a convergência do que o BC-GMRES. Claramente, pode-se notar as iterações em que o BC-GMRES sofreu reinicialização, observe que a curva do logaritmo da norma residual decresce lentamente e em certos momentos apresenta algumas “lombas”; essas iterações correspondem aos “picos” na curva do m , ou seja, conforme dito anteriormente, o resíduo não está decrescendo de forma satisfatória aos dados de entrada do método, então o mesmo é reinicializado.

Na figura (4.68) note que ambos os métodos convergiram, sendo que o LGMRES($m, 2$) e o LGMRES-A($m, 3$) apresentaram menor número de flops do que o LGMRES-A($m, 2$) e o LGMRES($m, 3$).

4.1.18 SHERMAN1

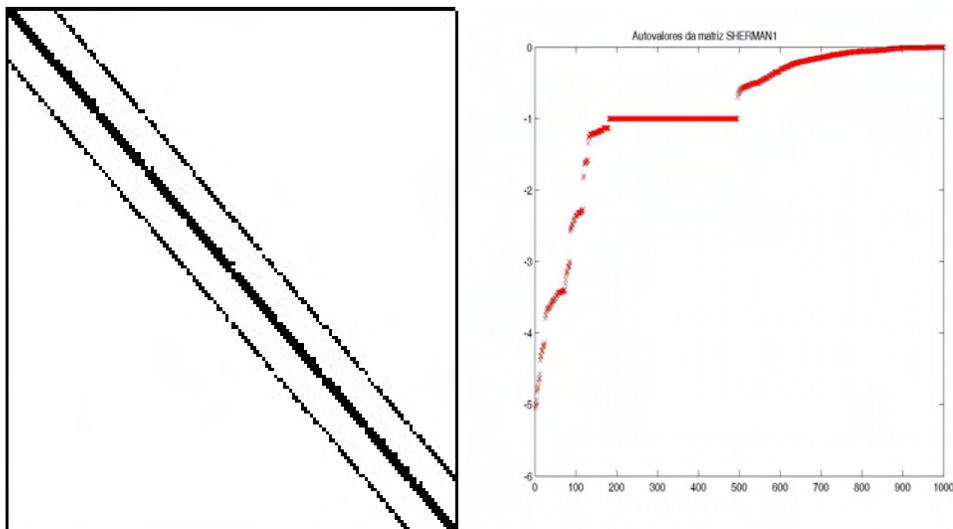


Figura 4.69: SHERMAN1: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 1000×1000
- tipo: real não-simétrica
- entradas não nulas: 3750
- número de condição (estimado): $2,3 \times 10^4$
- $\|A\|_2$ estimado: 5

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	sim	700	4	10	$9,5363 \times 10^8$
GMRESH-A	sim	45	5	40	$2,6427 \times 10^8$
SGMRES	sim	270	10	10	$9,9771 \times 10^7$
SGMRES-A	sim	50	10	35	$7,0267 \times 10^7$
SGMRESH	sim	140	5	10	$1,2592 \times 10^8$
SGMRESH-A	sim	45	2	20	$6,7034 \times 10^7$
BC-GMRES	sim	700	1	10	$1,1292 \times 10^8$
BC-GMRES-A	sim	80	1	50	$7,6439 \times 10^7$
LGMRES($m,2$)	sim	60	10	10	$3,4758 \times 10^7$
LGMRES-A($m,2$)	sim	50	10	20	$2,5897 \times 10^7$
LGMRES($m,3$)	sim	60	10	10	$3,6628 \times 10^7$
LGMRES-A($m,3$)	sim	50	10	20	$3,3041 \times 10^7$

Tabela 4.18: SHERMAN_1

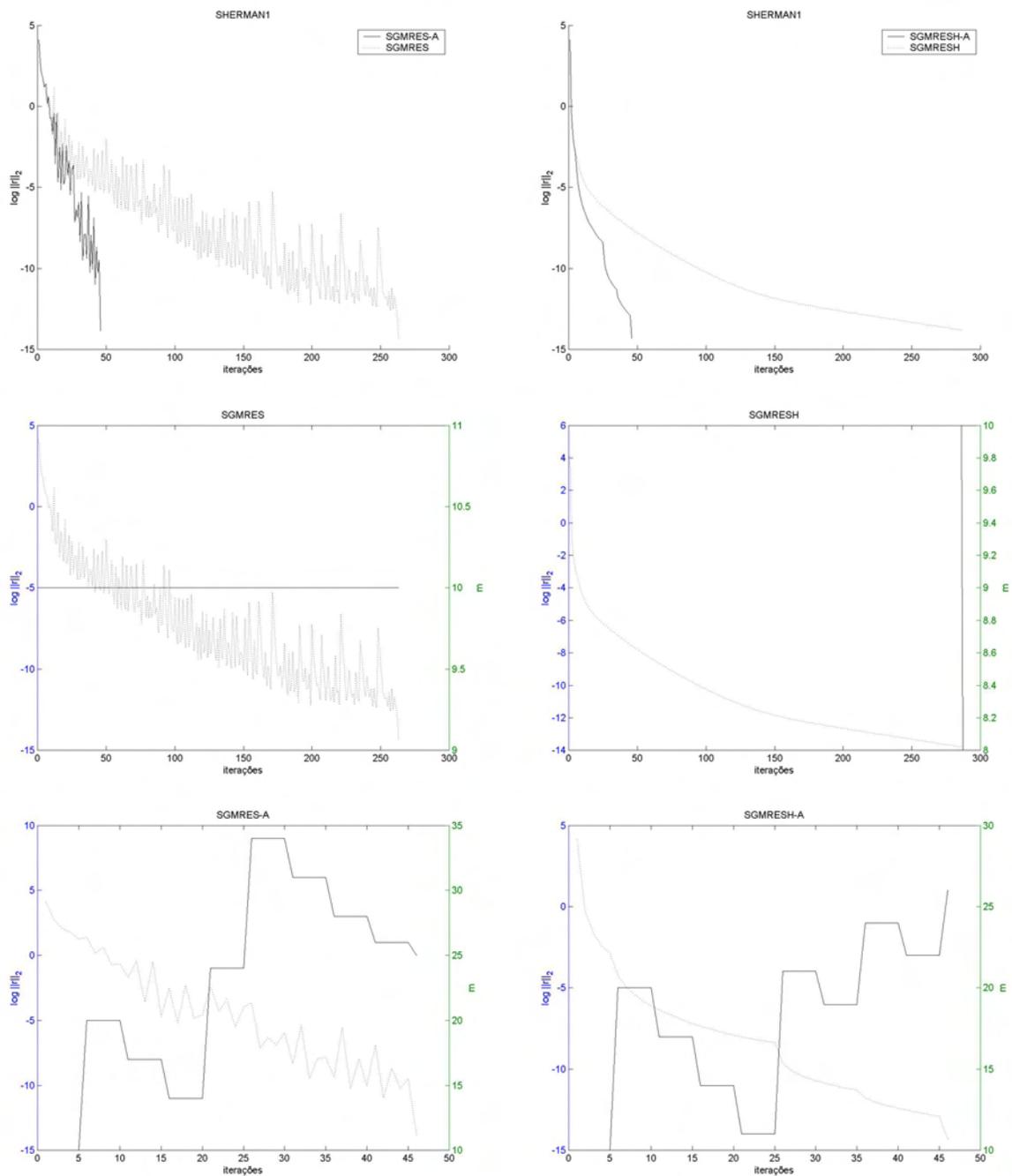


Figura 4.70: SHERMAN1: Gráfico do resíduo e m versus iterações

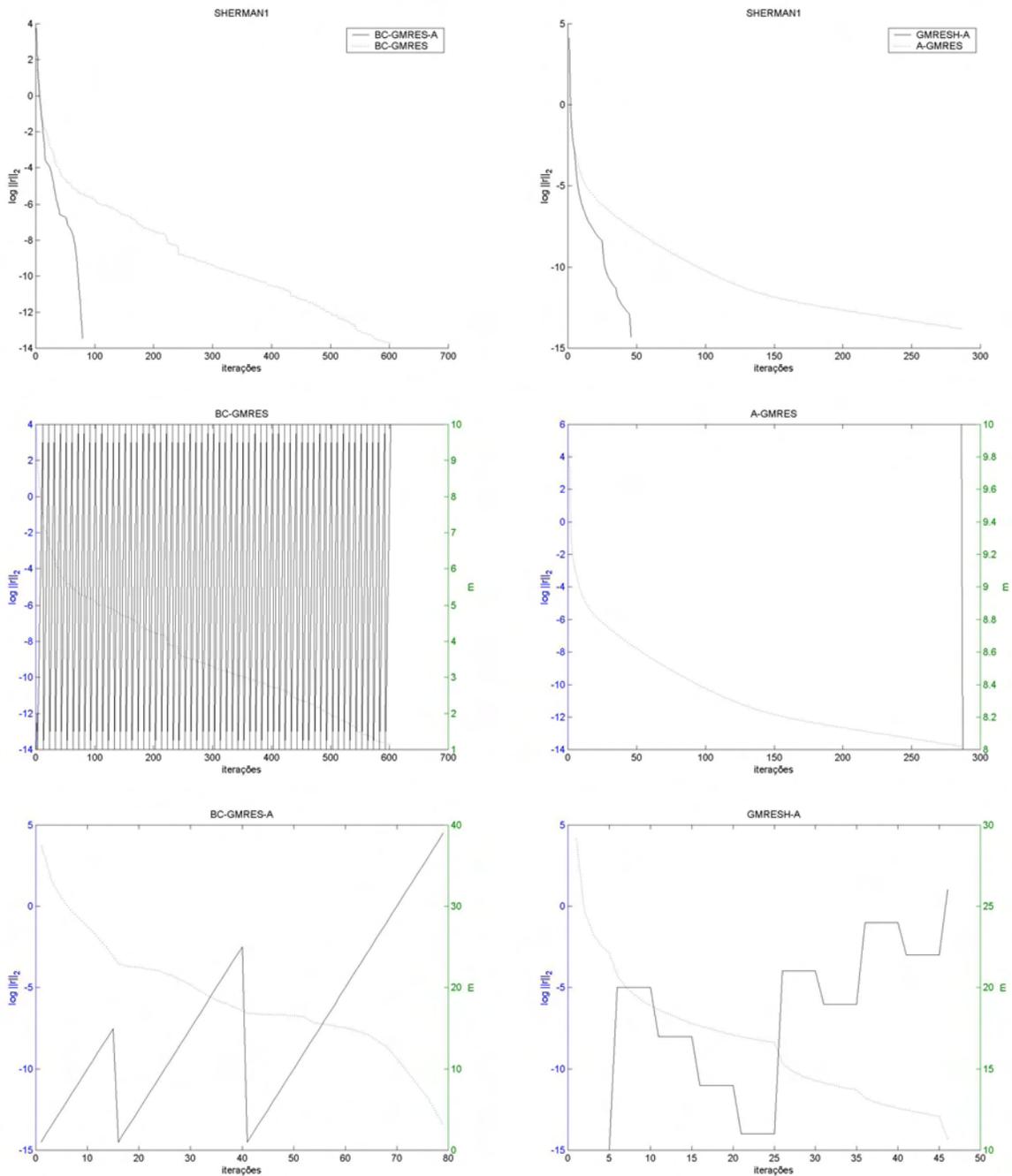


Figura 4.71: SHERMAN1: Gráfico do resíduo e m versus iterações

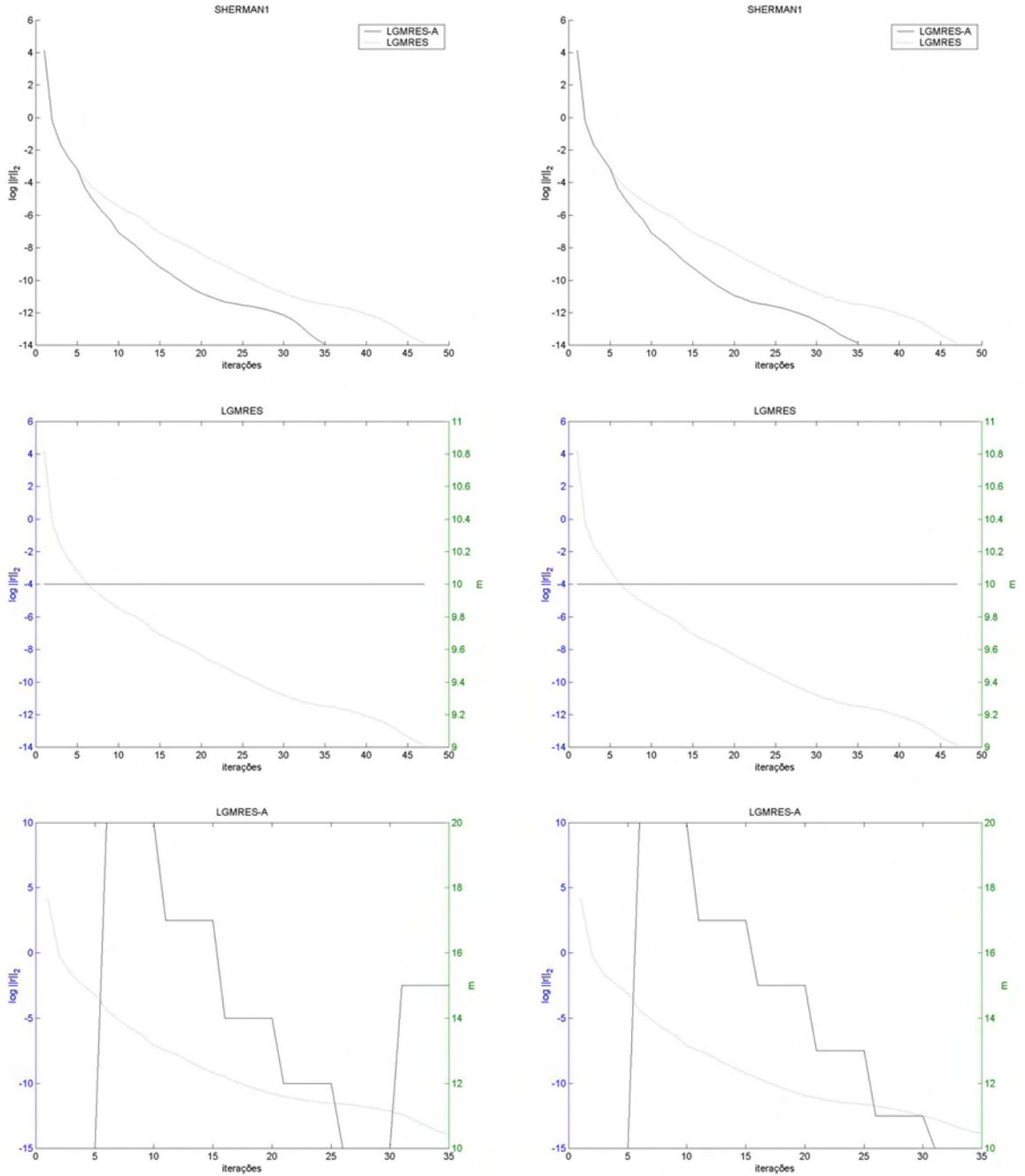


Figura 4.72: SHERMAN1: Gráfico do resíduo e m versus iterações

Na figura (4.70) note que ambos os métodos convergiram, sendo que o SGMRES-A e o SGMRESH-A apresentaram menor número de flops do que o SGMRES e o SGMRESH. Note que no gráfico do SGMRES, o m não apresentou variação ao longo das iterações e o logaritmo da norma residual variou bruscamente. O contrário ocorreu com o SGMRESH, onde o logaritmo da norma residual decaiu de forma “suave”, sem apresentar grandes oscilações; e o m apresentou um decréscimo oriundo da detecção de um subespaço invariante na formação do subespaço de Krylov.

Na figura (4.71) note que os gráficos comportam-se de forma similar ao sistema anterior, sendo que novamente o BC-GMRES-A apresentou menor número de flops e necessitou de um número menor de iterações para obter a convergência dentro da tolerância pré-especificada. Observe que o BC-GMRES-A necessitou de um número menor de reinicializações do que o BC-GMRES.

Na figura (4.72) note que ambos os métodos convergiram, no entanto o LGMRES-A($m, 2$) e o LGMRES($m, 3$) apresentaram menor número de flops do que o LGMRES($m, 2$) e o LGMRES($m, 3$). Observe o quão similar foi o comportamento desses métodos nessa figura.

4.1.19 STEAM1

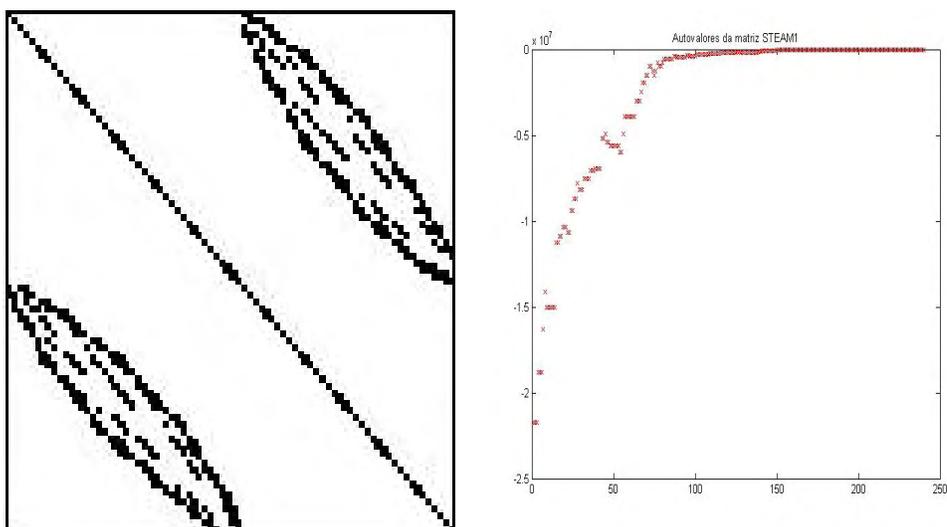


Figura 4.73: STEAM1: a)Estrutura da matriz, b)Autovalores da matriz

Características:

- tamanho: 240×240
- tipo: real não-simétrica
- entradas não nulas: 2248
- número de condição (estimado): 3×10^7
- $\|A\|_2$ estimado: $2,2 \times 10^7$

Testes realizados:

Método	Convergiu?	i	m mínimo	m máximo	Flops
A-GMRES	não	-	10	10	$8,6764 \times 10^7$
GMRESH-A	não	-	10	50	$1,4748 \times 10^9$
SGMRES	não	-	10	10	$2,9092 \times 10^7$
SGMRES-A	não	-	10	50	$3,5593 \times 10^8$
SGMRESH	não	-	10	10	$6,0312 \times 10^7$
SGMRESH-A	não	-	10	50	$9,7988 \times 10^8$
BC-GMRES	não	-	1	10	$1,5510 \times 10^7$
BC-GMRES-A	não	-	1	50	$1,3674 \times 10^7$
LGMRES($m,2$)	não	-	10	10	$3,7738 \times 10^7$
LGMRES-A($m,2$)	sim	200	10	50	$3,0835 \times 10^8$
LGMRES($m,3$)	não	-	10	10	$4,2212 \times 10^7$
LGMRES-A($m,3$)	sim	180	10	50	$2,8215 \times 10^8$

Tabela 4.19: STEAM_1

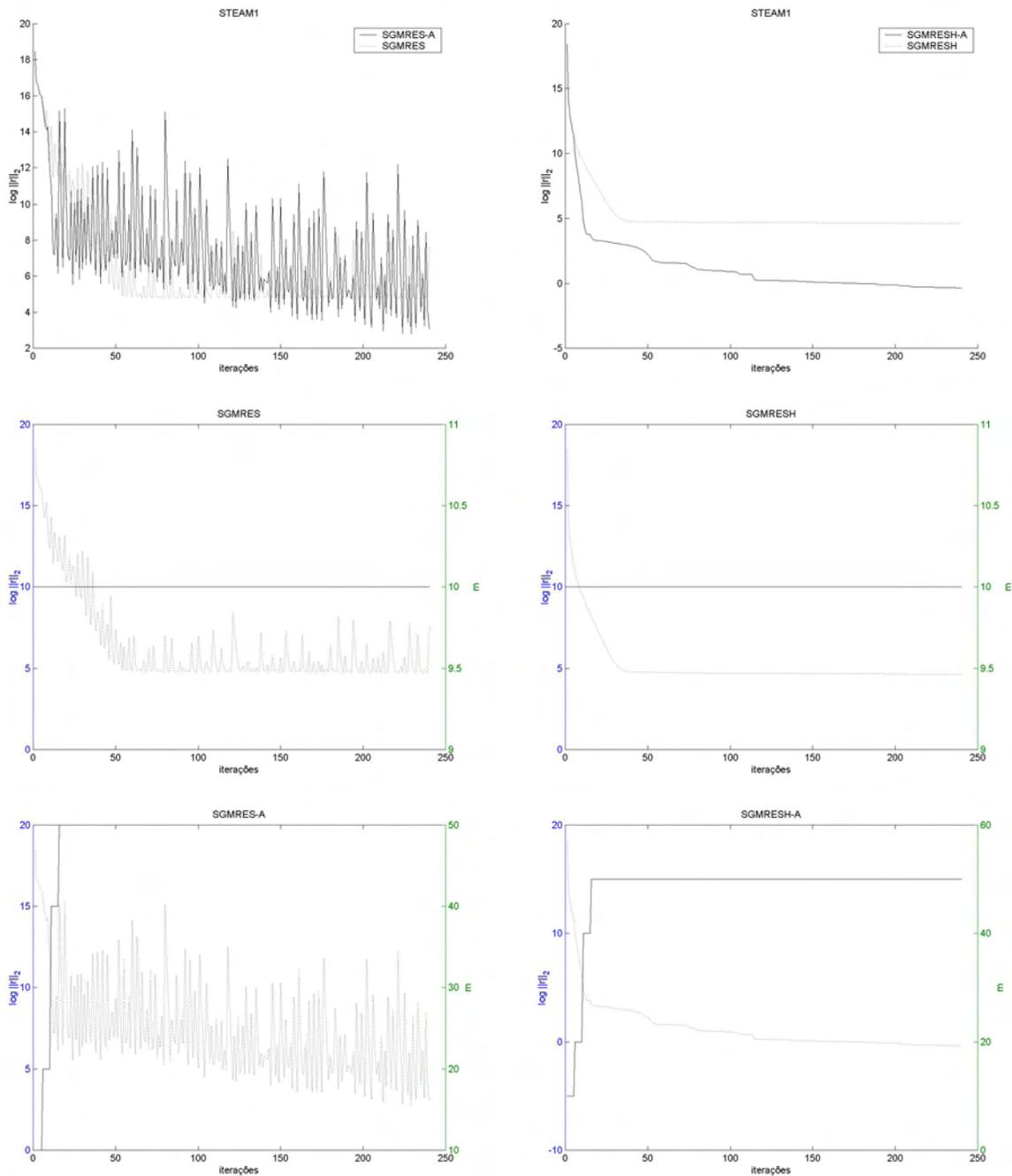


Figura 4.74: STEAM1: Gráfico do resíduo e m versus iterações

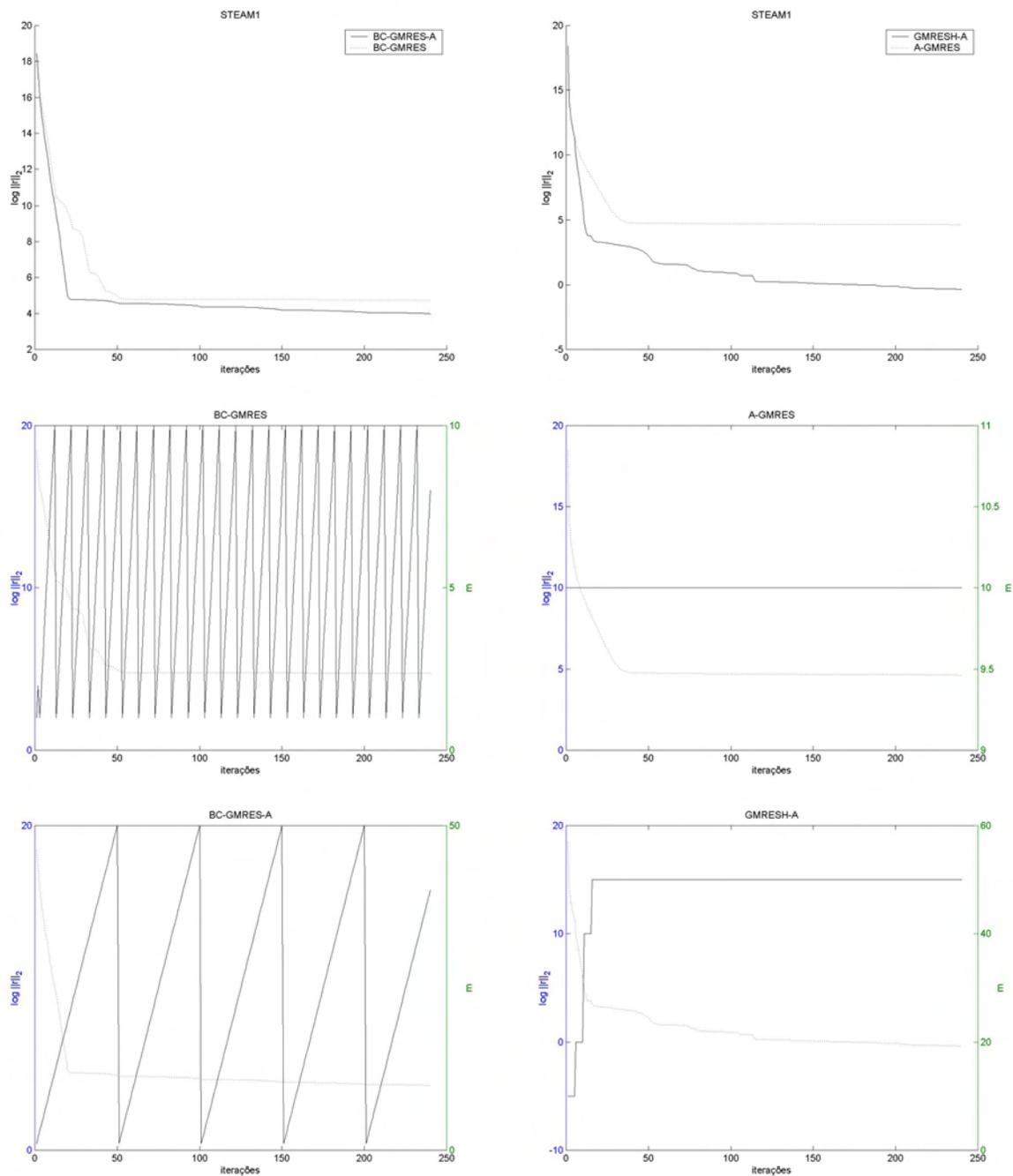


Figura 4.75: STEAM1: Gráfico do resíduo e m versus iterações

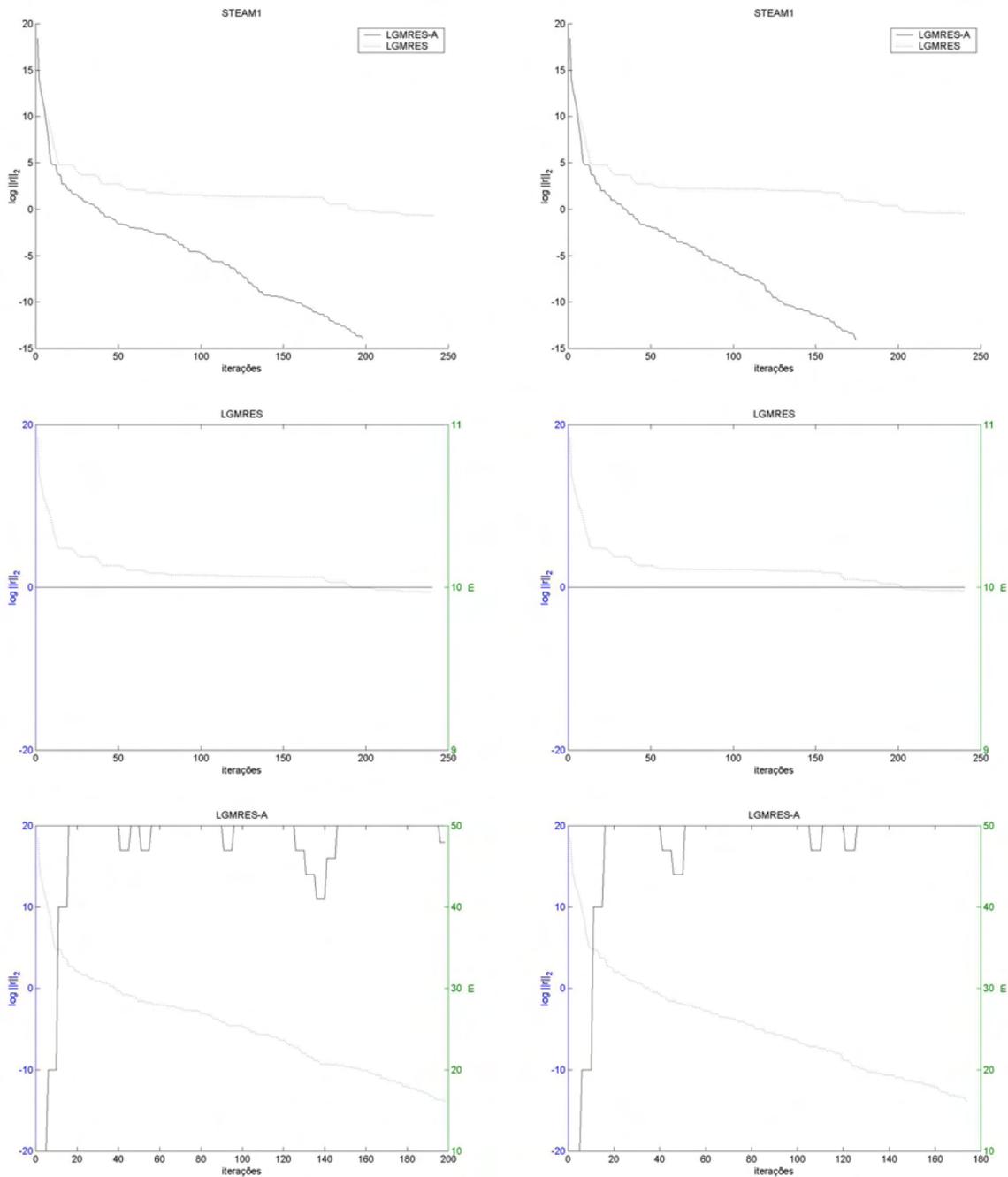


Figura 4.76: STEAM1: Gráfico do resíduo e m versus iterações

Nas figuras (4.74) e (4.75) note que ambos os métodos não obtiveram a convergência dentro da tolerância pré-especificada.

Na figura (4.76), apenas o LGMRES-A($m, 2$) e o LGMRES-A($m, 3$) obtiveram a convergência dentro da tolerância pré-especificada.

5 CONCLUSÕES

Nesse trabalho, estudamos diferentes maneiras de modificar a dimensão da base do subespaço de Krylov, de acordo com o sistema estudado. Apresentamos, agora, após o desenvolvimento e a aplicação nos experimentos realizados, a exposição de algumas conclusões sobre a realização dos mesmos.

As comparações foram realizadas entre os seguintes métodos: A-GMRES(m) e GMRESH-A(m); BC-GMRES(m) e BC-GMRES-A(m); SGMRES(m) e SGMRES-A(m); SGMRESH(m) e SGMRESH-A(m) e por fim LGMRES(m, k) e LGMRES-A(m, k) com $k = 2$ e $k = 3$.

Foram realizados um total de 114 testes comparativos entre os métodos citados acima, ou seja, utilizamos 19 matrizes, selecionadas do sítio “Matrix Market” [MCSD/ITL/NIST, 2003].

De forma sucinta podemos apresentar os resultados da seguinte forma: 76 testes convergiram para os métodos adaptativos, ou seja, para os métodos acrescidos do critério desenvolvido nesse trabalho para a escolha do m adequado ao sistema estudado, apresentando menor número de flops e de iterações do que os demais; 24 métodos não adaptativos, ou seja, métodos que não apresentam nenhum critério para a escolha do m , convergiram apresentando menor número de flops do que as versões adaptativas e 12 testes não apresentaram convergência dentro da tolerância pré-especificada, em nenhum dos casos.

Dentre os valores apresentados devemos destacar que, para algumas matrizes, as versões adaptativas mantiveram-se absolutas, ou seja, todos os métodos adaptativos convergiram e apresentaram menor número de flops do que os demais, basta observar os resultados obtidos com as matrizes: CDDE1, GRE115, NOS3, ORSIRR1, SAYLR3 e SHERMAN1. Em outros, pode-se notar que apenas as versões adaptativas obtiveram a convergência dentro da tolerância pré-especificada, veja por exemplo: FS_183_1; GRE115; LNS_131 e STEAM1, o que indica que o uso das versões adaptativas é indicado. Note que entre 114 testes realizados, 5 apresentaram convergência apenas para as versões adaptati-

vas, nenhuma apenas para as versões não-adaptativas, e dentre todos os testes, somente 2 sistemas não obtiveram a convergência em nenhum dos casos.

Salientamos que, para os sistemas utilizando as matrizes E05R0300 e a IMPCOLA, não se obteve convergência em nenhuma das versões; e, nos testes realizados com a matriz ARC_130, alguns métodos apresentaram o mesmo número de flops para as versões adaptativas e não adaptativas, fazendo com que não pudéssemos concluir nada a respeito das diferenças apresentadas por essas versões.

Um fato interessante foi que o número de iterações necessárias para obter a convergência das versões adaptativas sempre foram menores do que as demais; isso, independentemente, do número de flops apresentados pelos métodos.

Em virtude dos dados apresentados podemos concluir que as versões adaptativas formam uma excelente alternativa para a resolução de sistemas não-simétricos esparsos, visto que tendem a diminuir o custo computacional e o tempo de execução dos mesmos. Note que os testes foram realizados sem o uso de pré-condicionadores, visto que o nosso objetivo nesse trabalho era de elaborar um critério adequado para a escolha da dimensão do subespaço de Krylov; devido a isso não nos detivemos nos mesmos.

Como sugestão para o prosseguimento desse trabalho de pesquisa, podemos citar o desenvolvimento de um critério para a resolução de sistemas com múltiplos termos independentes utilizando o GMRES em blocos [Saad, 2003], a análise da aplicação de pré-condicionadores no critério apresentado nesse trabalho, e a escolha do parâmetro L no método BICGSTAB(L) [Sleijpen e Fokkema, 1993], onde nesse último a variável L apresenta a mesma função do m .

Bibliografia

- [Arnoldi, 1951] Arnoldi, W. (1951). The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29.
- [Baker et al., 2003] Baker, A., Jessup, E., e Manteuffel, T. (2003). A Technique for Accelerating the Convergence of Restarted GMRES. Technical Report CU-CS-945-03, Dept. of Computer Science, College of Engineering and Applied Science, University of Colorado at Boulder.
- [Bischof e Tang, 1991] Bischof, C. e Tang, P. (1991). Robust incremental condition estimation. Technical Report 33, University of Tennessee, Computer Science Department.
- [Brown e Walker, 1997] Brown, P. e Walker, H. (1997). GMRES on (nearly) singular systems. *SIAM J. Matrix Anal Appl.*, pages 37–51.
- [Chapman e Saad, 1997] Chapman, A. e Saad, Y. (1997). Deflated and augmented krylov subspace techniques. *Numerical Linear Algebra with Applications*, 4:15–41.
- [Cláudio, 1989] Cláudio, D. (1989). *Cálculo numérico computacional : teoria e prática*. Atlas, São Paulo.
- [Cláudio et al., 1987] Cláudio, D., Diverio, T., e Toscani, L. (1987). *Fundamentos de matemática computacional*. D.C. Luzzatto.
- [Da Cunha, 1992] Da Cunha, R. (1992). *A study on iterative methods for the solution of systems of linear equations on transputer networks*. PhD thesis, University of Kent, Canterbury.
- [Driver, 1997] Driver, M. (1997). *Parallel sparse linear algebra for homotopy methods*. PhD thesis, Faculty of the Virginia Polytechnic Institute and state university.
- [Eiermann, 1989] Eiermann, M. (1989). Field of values and iterative methods. *Linear Algebra Applics.*, 56:139–156.
- [Eiermann e Ernst, 1998] Eiermann, M. e Ernst, O. (1998). Geometrics aspects in the theory of krylov methods. *submitted*.

- [Elman, 1982a] Elman, H. (1982a). Iterative methods for large sparse nonsymmetric systems of linear equations. *Ph.D. thesis; Computer science Dept., Yale Univ, New Haven, CT.*
- [Elman, 1982b] Elman, H. (1982b). *Iterative methods for large sparse nonsymmetric systems of linear equations.* PhD thesis, Computer Science Dept., Yale University.
- [Embree, 1995] Embree, M. (1995). How descriptive are GMRES convergence bounds? *Oxford University Computing Laboratory.*
- [Goossens e Roose, 1997] Goossens, S. e Roose, D. (1997). Ritz and harmonic ritz values and the convergence of FOM and GMRES. *submitted, -:-.*
- [Hageman e Young, 1981] Hageman, L. e Young, D. (1981). *Applied Iterative Methods.* Academic Press, New York.
- [MCSD/ITL/NIST, 2003] MCSD/ITL/NIST (2003). Matrix Market: A visual repository of test data for use in comparative studies of algorithms for numerical linear algebra. <http://math.nist.gov/MatrixMarket/>.
- [Nachtigal et al., 1990] Nachtigal, N., Reddy, S., e Trefethen, L. (1990). How fast are nonsymmetric matrix iterations? Numerical analysis report, 90-2, Department of Mathematics, Massachusetts Institute of Technology.
- [Nodera e Moriya, 2003] Nodera, T. e Moriya, K. (2003). New adaptive GMRES(m) method with choosing suitable restart cycle m . *PPAM*, 18(1):1105–1113.
- [Paige e Saunders, 1975] Paige, C. C. e Saunders, M. (1975). Solution of sparse indefinite systems of linear equations. *SIAM*, 12:617–629.
- [Saad, 1984] Saad, Y. (1984). Practical use of some krylov subspace methods for solving indefinite and unsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 5:203–228.
- [Saad, 1992] Saad, Y. (1992). Analysis of some krylov subspace approximations to the matrix exponential operator. *SIAM*, 1:209–228.
- [Saad, 2003] Saad, Y. (2003). *Iterative methods for sparse linear systems.* SIAM, Philadelphia, 2nd edition.

- [Saad e Schultz, 1986] Saad, Y. e Schultz, M. (1986). GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7:856–869.
- [Sleijpen e Fokkema, 1993] Sleijpen, G. e Fokkema, D. (1993). BICGSTAB(L) for linear equations involving unsymmetric matrices with complex spectrum. *ETNA*, 1:11–32.
- [Tsuno e Nodera, 1999] Tsuno, N. e Nodera, T. (1999). The speedup of the GMRES(m) method using the early restarting procedure. *IPSSJ*, 40(4):1760–1773.
- [Vinsome, 1976] Vinsome, P. (1976). ORTHOMIN, an iterative methods for solving sparse sets of simultaneous linear equations. *Society of Petroleum Engineers of AIME*, pages 149–159.
- [Walker, 1988] Walker, H. (1988). Implementation of the GMRES method using householder transformations. *SIAM*, 9(1):152–163.
- [Walker e Zhou, 1994] Walker, H. e Zhou, L. (1994). A simpler GMRES. *Numerical Linear Algebra with Applications*, 1:571–581.
- [Young e Jea, 1980] Young, D. e Jea, K. (1980). Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods. *Linear Algebra Appl.*, 34:159–194.

Apêndice A CONVERGÊNCIA DO MÉTODO GMRES

Esse apêndice foi sugerido pelo professor Paulo Ricardo de Ávila Zingano, o qual propôs tais explicações a fim de um melhor entendimento do capítulo 2, o qual apresenta os métodos propriamente ditos.

1. O GMRES resolve sistemas $Ax = b$, com $A \in \mathbb{R}^n$ e $b \in \mathbb{R}^n$. Supondo A inversível, temos que dada uma aproximação x_0 para a solução x , com resíduo $r_0 = b - Ax_0$, a k -ésima aproximação x_k para a solução x é dada por

$$x_k = x_0 + \kappa \quad (\text{A.1})$$

onde $\kappa \in \mathbb{R}^n$ é a solução no subespaço de Krylov, (secção 2.1), para $A\kappa = r_0$ no sentido de mínimos quadrados, isto é:

$$\kappa = \sum_{j=0}^{k-1} \alpha_j A^j r_0 \quad (\text{A.2})$$

com $\alpha_0, \alpha_2, \dots, \alpha_{k-1}$ escolhidos de modo a minimizar

$$\|A\kappa - r_0\|_2 = \left\| \sum_{j=1}^{k-1} \alpha_{j-1} A^j r_0 - r_0 \right\|_2 \quad (\text{A.3})$$

O resíduo da solução x_k é:

$$r_k = b - Ax_k = r_0 - A\kappa = r_0 - \sum_{j=1}^k \alpha_{j-1} A^j r_0 = \hat{p}(A)r_0 \quad (\text{A.4})$$

onde $\hat{p}(z) = 1 - \sum_{j=1}^k \alpha_{j-1} z^j$, ou seja $\|r_k\|_2 = \|\hat{p}(A)r_0\|_2$ minimiza (A.4).

Em particular, segue que: $\|r_k\|_2 = \|\hat{p}(A)r_0\|_2 \leq \|p(A)r_0\|_2 \leq \|p(A)\|_2 \|r_0\|_2$ $\forall p \in \mathbf{P}_k$, onde para $B \in \mathbb{R}^n$ temos que $\|B\|_2 = \sup_{\|x\|_2=1} \|Bx\|_2$, a saber, norma espectral, conforme aparece na secção (2.4).

2. Conforme a secção (2.4), a propriedade minimal

$$\|r_k\|_2 = \min_{p \in \mathbf{P}_k} \|p(A)r_0\|_2 \quad (\text{A.5})$$

é um resultado fundamental quanto á convergência do GMRES. Por exemplo:

(a) Segue de (A.5) que, no GMRES, tem-se

$$\|r_0\|_2 \geq \|r_1\|_2 \geq \dots \geq \|r_{k+1}\|_2 \geq \dots \quad (\text{A.6})$$

Este comportamento monotônico dos resíduos também é válido em geral para o GMRES(m), pela mesma razão.

(b) Um resultado de convergência para o GMRES(m), para qualquer m escolhido, e qualquer $x_0 \in \mathbb{R}^n$ tomado é que se $|\lambda - z_0| < \rho$ para todo autovalor λ de A , com $z_0 \in \mathbb{C}$, $\rho > 0$ tais que $\rho < |z_0|$. Então GMRES e o GMRES(m) satisfazem

$$\|r_k\|_2 \leq C \left(\frac{\rho}{|z_0|} \right)^k \|r_0\|_2, \quad \forall k = 1, 2, \dots \quad (\text{A.7})$$

onde C depende somente de A . Para outros resultados quanto a convergência do GMRES veja 3.

PROVA:

Sendo $p(z) = \left(\frac{z_0 - z}{z_0}\right)^k$, temos $p \in \mathbf{P}_k$, e então:

$$\|r_k\|_2 \leq \|p(A)\|_2 \|r_0\|_2 \leq C_A \left(\frac{\rho}{|z_0|} \right) \|r_0\|_2 \quad (\text{A.8})$$

TEOREMA: Em aritmética exata, o método GMRES produz uma solução exata em $\leq n$ iterações.

PROVA: Tomando $k = n$ e $p(z) = \frac{\det(zI - A)}{(-1)^n \det A}$, observe que $p \in \mathbf{P}_n$, para qualquer $x_0 \in \mathbb{R}^n$, temos:

$$\|r_n\|_2 \leq \|p(A)\|_2 \|r_0\|_2 = 0 \quad (\text{A.9})$$

pois $p(A) = 0$.

TEOREMA: Se a dimensão do

$$\kappa_k(A, r_0) = \text{subespaço}(r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0) \quad (\text{A.10})$$

para p , então o GMRES produz a solução exata em $\ll p$ iterações.

PROVA: Nesse caso, tem-se

$$\|r_q\|_2 = \|r_n\|_2 \quad (\text{A.11})$$

$\forall p \leq q \leq n$, de modo que por (2):

$$\|r_q\|_2 = 0 \forall p \leq q \leq n \quad (\text{A.12})$$

Em particular: $\|r_q\|_2 = 0$.

3. Se os autovalores de A estiverem sobre uma semi-reta partindo da origem, conforme a figura A.1,

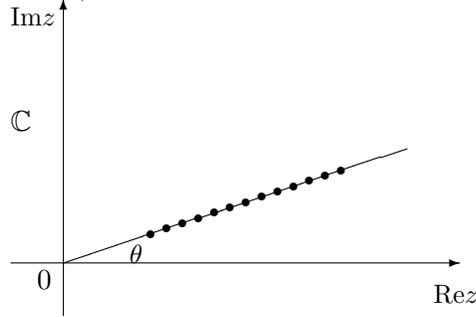


Figura A.1: Autovalores de A pertencentes a reta

ou então, se os autovalores estiverem suficientemente próximos de uma semi-reta, conforme a figura A.2,

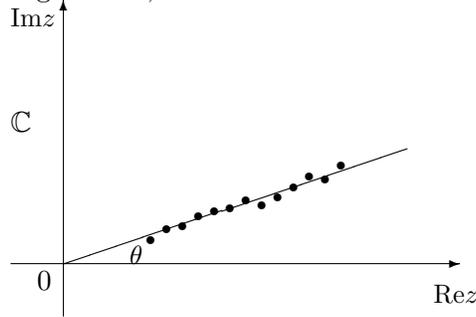


Figura A.2: Autovalores de A próximos a reta

então o GMRES(m) converge, com $\|r_k\|_2 \leq C_A \beta^k \|r_0\|_2$, $\forall k = 1, 2, \dots$, para algum $\beta \in]0, 1[$, independente de x_0 .

PROVA: Tome $\gamma > 0$ suficientemente pequeno tal que se tenha $q(z) = 1 - \gamma e^{-i\theta} z$, onde $i = \sqrt{-1}$ satisfazendo $0 < q(\lambda) < 1 \forall \lambda \in \text{spec}(A)$. Tomando então $p(z) = q(z)^k$, temos $p \in \mathbf{P}_k$ e então pela propriedade fundamental de convergência do GMRES temos que:

$$\|r_k\|_2 \leq \|p(A)\|_2 \|r_0\|_2 \leq C_A \left(\max_{\lambda \in \text{spec}(A)} q(\lambda) \right)^k \|r_0\|_2. \quad (\text{A.13})$$