

# Introdução aos métodos multigrid para resolução de sistemas lineares e não lineares

Aluno: Gabriel Haeser - RA: 001744  
Orientadora: Prof. Dra. Márcia A. Gomes-Ruggiero

## 1 Resumo do plano inicial

A proposta deste projeto é o estudo de técnicas de Multigrid para resolução dos sistemas lineares que resultam da discretização de Problemas de Valor de Contorno. O objetivo do trabalho é uma introdução ao tema que permita ao aluno entrar em contato com esta técnica e desenvolver programas em MatLab para resolver problemas de valor de contorno que modelam alguns problemas aplicados. Desta forma, o projeto inclui etapas que envolvem estudo teórico, elaboração de programas computacionais e resolução de problemas aplicados, possibilitando ao aluno o desenvolvimento completo de um trabalho em Matemática Aplicada.

## 2 Trabalho elaborado neste período

### 2.1 Introdução

Inicialmente trabalhamos com sistemas lineares, pois a grande dificuldade nos métodos multigrid é o ajuste dos parâmetros do algoritmo (número de iterações por malha), além de tentarmos implementar outras idéias que surgiram para o cálculo do resíduo na malha grossa. Em seguida trabalhamos com sistemas não lineares e implementamos em Matlab o algoritmo FAS, que será descrito aqui. Todos os códigos implementados podem ser acessados no endereço: [www.ime.unicamp.br/~ghaeser/fapesp](http://www.ime.unicamp.br/~ghaeser/fapesp)

### 2.2 Multigrid para sistemas lineares

No primeiro semestre do projeto implementamos em Matlab versão 6.1 o algoritmo ciclo V completo (full multigrid), para resolvermos um sistema linear  $Ax = b$  resultante da

discretização de um problema de valor de contorno. O algoritmo havia sido implementado de maneira recursiva, o que tornou difícil a sua modificação, embora seja muito útil para testes onde utilizamos muitos níveis. O algoritmo descrito abaixo (em 3 níveis, nas malhas  $h$ ,  $2h$  e  $4h$ ) foi implementado para facilitar a modificação dos parâmetros.

- Iniciar a resolução do sistema linear na malha mais grossa ( $4h$ ) a partir de uma aproximação inicial  $x_0$ , realizar  $v$  iterações do processo iterativo ou se a dimensão for pequena, resolver o sistema por um método direto, obtendo a solução  $x_{4h}^a$ .
- Transferir esta solução para a malha  $2h$ :  $x_{2h}^a = I_{4h}^{2h}(x_{4h}^a)$  onde  $I_{4h}^{2h}$  é o operador linear que leva um vetor na malha  $4h$  para um vetor na malha  $2h$  e realizar  $v$  iterações na malha  $2h$  a partir de  $x_{2h}^a$  obtendo  $x_{2h}^b$  e o resíduo  $r_{2h}^a = b_{2h} - A^{2h}x_{2h}^b$ .
- Transferir o resíduo para a malha  $4h$ :  $r_{4h}^a = I_{2h}^{4h}(r_{2h}^a)$  e resolver o sistema linear  $A^{4h}e = r_{4h}^a$  através de um método direto ou realizar  $v$  iterações na malha  $4h$  a partir de  $e_0 = 0$ , obtendo  $e_{4h}^a$
- Transferir o erro na malha  $4h$  para a malha  $2h$ :  $e_{2h}^a = I_{4h}^{2h}(e_{4h}^a)$  e atualizar a solução atual:  $x_{2h}^c = x_{2h}^b + e_{2h}^a$
- Aplicar  $v$  iterações de um método iterativo no sistema  $A^{2h}x = b_{2h}$ , com aproximação inicial  $x_{2h}^c$  obtendo  $x_{2h}^d$
- Transferir a aproximação da solução na malha  $2h$  para a malha  $h$ :  $x_h^a = I_{2h}^h(x_{2h}^d)$  e iterar no sistema  $A^h x = b_h$ , com aproximação inicial  $x_h^a$ , obtendo  $x_h^b$  e o resíduo  $r_h^a = b_h - A^h x_h^b$
- Transferir o resíduo da malha  $h$  para a malha  $2h$ :  $r_{2h}^b = I_h^{2h}(r_h^a)$  e iterar no sistema linear  $A^{2h}e = r_{2h}^b$ , com aproximação inicial  $e_0 = 0$ , obtendo  $e_{2h}^b$  e o resíduo  $r_{2h}^c = r_{2h}^b - A^{2h}e_{2h}^b$
- Transferir o resíduo da malha  $2h$  para a malha  $4h$ :  $r_{4h}^b = I_{2h}^{4h}(r_{2h}^c)$  e aplicar  $v$  iterações no sistema linear  $A^{4h}e = b_{4h}$  com aproximação inicial  $e_0 = 0$  ou resolver por um método direto obtendo  $e_{4h}^b$
- Transferir o erro na malha  $4h$  para a malha  $2h$ :  $e_{2h}^c = I_{4h}^{2h}(e_{4h}^b)$  e atualizar a solução:  $e_{2h}^d = e_{2h}^b + e_{2h}^c$
- Aplicar  $v$  iterações no sistema  $A^{2h}e = r_{2h}^c$  com aproximação inicial  $e_{2h}^d$  obtendo  $e_{2h}^e$
- Transferir o erro na malha  $2h$  para a malha  $h$ :  $e_h^a = I_{2h}^h(e_{2h}^e)$ , corrigir a solução:  $x_h^c = x_h^b + e_h^a$
- Aplicar  $v$  iterações ou iterar até a precisão desejada no sistema linear  $A^h x = b_h$  com aproximação inicial  $x_h^c$  obtendo  $x_h^d$ , aceita como solução do problema original  $Ax = b$

Com este algoritmo implementado, fizemos testes resolvendo o sistema linear tanto com gradientes conjugados como com GMRES, os testes com GMRES estão mostrados abaixo.

### 2.2.1 Experimento numérico

Testamos o algoritmo descrito acima na resolução de um problema de valor de contorno unidimensional:

$$-u''(x) = f(x), \quad x \in [0, 1], \quad x(0) = x(1) = 0$$

onde  $f(x)$  é tal que a solução exata do sistema seja  $10x(x-1)(0.5-x)\text{sen}(2x)$ .

Os resultados obtidos estão mostrados na tabela 1 abaixo, no qual foi usado um esquema onde a malha mais grossa tem dimensão  $n=15$  pontos. Executamos o algoritmo para  $v = 3$ ,  $v = 10$  e  $v = 20$  (o número de iterações por malha) e em seguida verificamos quantas iterações do GMRES (com restart 30, 50 e sem restart) são necessárias para atingir a mesma precisão atingida pelo ciclo V completo, quando a resolução é na malha fina em todas as iterações.

|      |    |         | Iterações GMRES  |            |            |             |
|------|----|---------|------------------|------------|------------|-------------|
| n    | v  | resíduo | ciclo V completo | restart=30 | restart=50 | sem restart |
| 511  | 3  | 1.9E-2  | 90               | 4673       | 2847       | 489         |
| 511  | 10 | 2.0E-3  | 300              | 8209       | 4983       | 507         |
| 511  | 20 | 5.0E-4  | 600              | 10463      | 6342       | 510         |
| 1023 | 3  | 1.9E-2  | 126              | 18551      | 11125      | 977         |
| 1023 | 10 | 2.1E-3  | 420              | 32604      | 19521      | 1014        |
| 1023 | 20 | 5.3E-4  | 840              | 41440      | 24802      | 1020        |
| 2047 | 3  | 1.9E-2  | 168              | 74165      | 44438      | 1954        |
| 2047 | 10 | 2.1E-3  | 560              | 130298     | 78043      | 2028        |
| 2047 | 20 | 5.5E-4  | 1120             | 165317     | 99008      | 2040        |
| 4095 | 3  | 1.9E-2  | 216              | 296549     | 177841     | –           |

Tabela 1 - Comparação entre o ciclo V completo e iterações direto

Analisando a tabela 1 constatamos a eficiência da técnica multigrid pois o número total de iterações foi sempre menor que o total realizado pelo método GMRES aplicado diretamente na malha fina, ressaltando que com a dimensão 4095 o GMRES sem restart não conseguiu resolver o problema, por falta de memória do computador.

O processo multigrid envolve a resolução de um sistema linear por um método direto (no caso um sistema linear de dimensão 15), mas mesmo assim é vantajoso pois muitas

iterações do GMRES são realizadas em sistemas com malha mais grossa e portanto com dimensão menor. É fácil contar quantas iterações são feitas em cada malha, sendo  $1h$  a malha cuja dimensão é  $2^n - 1$ ,  $2h$  é a malha cuja dimensão é  $2^{n-1} - 1$ , e a malha  $(2^i)h$  é a malha cuja dimensão é  $2^{n-i} - 1$ , o número de iterações realizadas na malha  $(2^i)h$  é  $2(i+1)v$ , a menos na malha mais grossa ( $n = 15 = 2^4 - 1$ ), onde resolvemos o sistema linear exatamente. Podemos ver que o número de iterações é menor nas malhas mais finas (i pequeno) o que é muito interessante pois as malhas mais finas possuem maior dimensão.

O objetivo neste teste foi comparar o desempenho da técnica multigrid ao variar o número de iterações realizadas em cada nível.

Observamos que poucas iterações em cada nível são suficientes para obter uma boa aproximação para a solução na malha mais fina. Para obtermos precisões maiores, podemos iterar no último passo do algoritmo até a precisão desejada, ou então distribuir estas iterações ao longo das malhas intermediárias, o que pode ser mais interessante já que os problemas intermediários possuem dimensão menor.

### 2.2.2 O cálculo do resíduo

Ao utilizarmos o ciclo V completo temos duas opções para calcularmos o resíduo na malha grossa.

A primeira delas, é a maneira usual:

Seja  $\mathbf{A}^h \mathbf{u} = \mathbf{b}^h$  o sistema a ser resolvido na malha  $\Omega^h$  e seja  $\mathbf{v}^h$  uma aproximação para a solução do sistema nesta malha. Calculamos o resíduo  $\mathbf{r}^h$  fazendo  $\mathbf{r}^h = \mathbf{b}^h - \mathbf{A}^h \mathbf{v}^h$  e o resíduo na malha  $\Omega^{2h}$  será  $\mathbf{r}^{2h} = \mathbf{I}_h^{2h} \mathbf{r}^h$ , onde  $\mathbf{I}_h^{2h}$  é o operador que transfere vetores da malha  $\Omega^h$  para a malha  $\Omega^{2h}$ .

A segunda maneira para calcularmos o resíduo na malha grossa é a seguinte:

Seja  $\mathbf{v}^{2h} = \mathbf{I}_h^{2h} \mathbf{v}^h$  a aproximação que tínhamos para a solução na malha  $\Omega^h$  levada para a malha  $\Omega^{2h}$ , o resíduo na malha  $\Omega^h$  é dado por  $\mathbf{r}^{2h} = \mathbf{b}^{2h} - \mathbf{A}^{2h} \mathbf{v}^{2h}$

A tabela 2 abaixo mostra os resultados obtidos ao compararmos as duas maneiras em um ciclo V completo em três níveis, sendo que a resolução é exata no nível mais baixo.  $v$  é o número de iterações feitas pelo GMRES em cada passo, sendo que no passo final iteramos até a precisão de  $10^{-6}$  na malha  $\Omega^h$  utilizando restart 30. Na tabela, o primeiro número é o total de iterações realizadas na malha  $\Omega^h$  no último passo do algoritmo e o segundo número é a soma do número de iterações em cada uma das malhas. Na coluna “direto na malha fina” é contado o número de iterações ao aplicarmos GMRES com restart 30 na malha  $\Omega^h$  até a precisão de  $10^{-6}$ . O problema testado foi

$$-\Delta u(x, y) = x(1-x)\text{sen}(16y)$$

para  $x, y$  no interior do quadrado unitário e  $u(x, y) = 0$  na fronteira do quadrado.

| n          | v  | primeira maneira | segunda maneira | direto na malha fina |
|------------|----|------------------|-----------------|----------------------|
| 31x31=961  | 3  | 37/52            | 68/83           | 100                  |
| 31x31=961  | 5  | 37/62            | 71/96           | 100                  |
| 31x31=961  | 7  | 28/63            | 71/106          | 100                  |
| 31x31=961  | 10 | 27/77            | 68/118          | 100                  |
| 31x31=961  | 15 | 18/92            | 64/139          | 100                  |
| 63x63=3969 | 3  | 49/64            | 118/133         | 234                  |
| 63x63=3969 | 5  | 50/75            | 116/141         | 234                  |
| 63x63=3969 | 7  | 40/75            | 116/141         | 234                  |
| 63x63=3969 | 10 | 37/87            | 117/167         | 234                  |
| 63x63=3969 | 15 | 30/105           | 118/193         | 234                  |

Tabela 2 - Comparação entre iteração direto e duas maneiras de se calcular o resíduo

Pela tabela 2 acima podemos ver que a segunda maneira é sempre inferior à primeira, enquanto que ambas são sempre superiores à iterações do GMRES diretamente na malha mais fina.

## 2.3 Multigrid para sistemas não lineares

### 2.3.1 Introdução

Nossos estudos agora se voltam para os métodos multigrid aplicados à sistemas não lineares. Começamos mostrando a principal diferença entre os sistemas de equações lineares e não lineares. Considere um sistema de equações não lineares,  $\mathbf{A}(\mathbf{u}) = \mathbf{0}$ , onde  $\mathbf{u} \in \mathbf{R}^n$  e  $\mathbf{A} : \mathbf{R}^n \rightarrow \mathbf{R}^n$ . Suponha que  $\mathbf{v}$  seja uma aproximação para a solução exata  $\mathbf{u}$ . É possível definir o erro e o resíduo da mesma maneira que definíamos no caso linear: o erro é simplesmente dado por  $\mathbf{e} = \mathbf{u} - \mathbf{v}$  e o resíduo é  $\mathbf{r} = -\mathbf{A}(\mathbf{v})$ . Como  $\mathbf{A}(\mathbf{u}) = \mathbf{0}$ , então podemos escrever:

$$\mathbf{A}(\mathbf{u}) - \mathbf{A}(\mathbf{v}) = \mathbf{r}.$$

Que será denominada *equação residual*. Como o operador  $\mathbf{A}$  é não linear, não podemos concluir como fizemos no caso linear e dizer que  $\mathbf{A}(\mathbf{u}) - \mathbf{A}(\mathbf{v}) = \mathbf{A}(\mathbf{e})$ . Isto implica que necessitamos de mudanças nos métodos.

### 2.3.2 Newton Multigrid

Uma primeira idéia para resolvermos o sistema não linear  $\mathbf{A}(\mathbf{u}) - \mathbf{A}(\mathbf{v}) = \mathbf{r}$ , é resolver o sistema linear oriundo da aproximação linear de  $\mathbf{A}$  em torno de  $\mathbf{v}$ :

$$\mathbf{A}(\mathbf{u}) = \mathbf{A}(\mathbf{v}) + \mathbf{J}(\mathbf{v})\mathbf{e} \Rightarrow \mathbf{r} = \mathbf{A}(\mathbf{u}) - \mathbf{A}(\mathbf{v}) = \mathbf{J}(\mathbf{v})\mathbf{e}$$

onde  $\mathbf{J}(\mathbf{v})$  é o Jacobiano da função  $\mathbf{A}$  aplicado em  $\mathbf{v}$ . O problema com esta idéia é que não estamos utilizando multigrid para tratar a não linearidade.

### 2.3.3 Full Approximation Scheme (FAS) [1]

Vamos introduzir a idéia do uso de multigrid em duas malhas. Suponha que temos uma aproximação para a solução na malha fina  $\Omega^h$ , denotada por  $\mathbf{v}^h$ , para o problema

$$\mathbf{A}^h(\mathbf{u}^h) = \mathbf{0}.$$

Procedendo de maneira análoga ao caso linear, queremos usar a equação residual na malha grossa  $\Omega^{2h}$  para aproximar  $\mathbf{e}^h$ , o erro associado à aproximação  $\mathbf{v}^h$ . A equação residual na malha grossa é dada por: ( $\mathbf{u}^{2h}$  denota a solução exata na malha grossa)

$$\begin{aligned} \mathbf{A}^{2h}(\mathbf{u}^{2h}) - \mathbf{A}^{2h}(\mathbf{v}^{2h}) &= \mathbf{A}^{2h}(\mathbf{v}^{2h} + \mathbf{e}^{2h}) - \mathbf{A}^{2h}(\mathbf{v}^{2h}) = \mathbf{r}^{2h} \implies \\ \mathbf{A}^{2h}(\mathbf{v}^{2h} + \mathbf{e}^{2h}) &= \mathbf{r}^{2h} + \mathbf{A}^{2h}(\mathbf{v}^{2h}) \end{aligned}$$

Onde o resíduo na malha grossa é dado por  $\mathbf{r}^{2h} = \mathbf{I}_h^{2h} \mathbf{r}^h$ . Se  $\mathbf{w}^{2h} = \mathbf{v}^{2h} + \mathbf{e}^{2h}$ , resolvemos o sistema não linear em  $\mathbf{w}$ :  $\mathbf{A}^{2h}(\mathbf{w}) = \mathbf{r}^{2h} + \mathbf{A}^{2h}(\mathbf{v}^{2h})$ . Realizando algumas iterações na malha  $\Omega^{2h}$  obtemos uma solução aproximada  $\bar{\mathbf{w}}^{2h}$  e a partir daí, o erro  $\mathbf{e}^{2h} = \bar{\mathbf{w}}^{2h} - \mathbf{v}^{2h}$ , e assim podemos atualizar a aproximação fazendo  $\mathbf{v}^h \leftarrow \mathbf{v}^h + \mathbf{I}_{2h}^h \mathbf{e}^{2h}$ .

O método descrito acima é a versão não linear do multigrid mais usada atualmente, que leva o nome de *full approximation scheme* (FAS). Uma versão em duas malhas deste método está descrita abaixo:

- Aplique o operador de restrição à aproximação atual e ao resíduo:  $\mathbf{r}^{2h} = \mathbf{I}_h^{2h}(-\mathbf{A}^h(\mathbf{v}^h))$  e  $\mathbf{v}^{2h} = \mathbf{I}_h^{2h} \mathbf{v}^h$ .
- Realize  $v$  iterações de um método iterativo para sistemas não lineares ao sistema na malha grossa:  $\mathbf{A}^{2h}(\mathbf{w}^{2h}) = \mathbf{A}^{2h}(\mathbf{v}^{2h}) + \mathbf{r}^{2h}$ .
- Calcule a aproximação para o erro na malha grossa:  $\mathbf{e}^{2h} = \mathbf{w}^{2h} - \mathbf{v}^{2h}$ .
- Aplique o operador de interpolação ao erro na malha grossa e corrija a aproximação atual na malha fina:  $\mathbf{v}^h \leftarrow \mathbf{v}^h + \mathbf{I}_{2h}^h \mathbf{e}^{2h}$ .
- Aplique o método iterativo na malha fina até atingir a precisão requerida.

Vale notar que se  $\mathbf{A}$  é um operador linear então FAS se reduz à correção linear em duas malhas. Logo, FAS pode ser visto como uma generalização do esquema de correção em duas malhas.

### 2.3.4 Experimentos numéricos - Newton puro

Neste projeto implementamos o método FAS em duas malhas utilizando o método de Newton (puro e com busca linear, resolvendo exatamente o sistema linear  $\mathbf{J}(\mathbf{x}^k)\mathbf{e} = -\mathbf{A}(\mathbf{x}^k)$ ) para iterar no sistema. A idéia foi iterar poucas vezes na malha fina, em seguida iterar na malha grossa até uma certa precisão, voltar para a malha fina e repetir o processo, até atingir a precisão na malha fina desejada ( $10^{-10}$ ).

Algumas idéias surgiram para melhorarmos a performance do algoritmo. Na primeira vez que iteramos na malha fina, iteramos no sistema com aproximação inicial nula, até que o resíduo atingido seja menor do que um, isso foi feito para garantir que a aproximação inicial seja razoavelmente boa, e também para que o critério adotado para iterarmos na malha grossa funcione corretamente. O critério adotado é iterarmos até que o resíduo atingido seja menor do que o máximo entre  $10^{-10}$  e um centésimo do resíduo na malha fina. Isso foi feito para evitarmos o fenômeno de *oversolving*, que consiste em iterarmos além da precisão necessária (pois há um erro associado ao processo de troca de malha).

Essas idéias foram testadas no teste numérico descrito abaixo.

### Primeiro teste numérico

O primeiro problema testado foi o problema de valor de contorno proposto em [1] descrito por:

$$-\Delta u(x, y) + \gamma u(x, y)e^{u(x, y)} = f(x, y)$$

para  $x, y$  no interior do quadrado unitário e  $u(x, y) = 0$  na fronteira do quadrado.

Discretizamos este problema com espaçamentos  $h = 1/32$  e  $h = 1/64$ , usando diferenças finitas centradas. Assim, o sistema de equações discretizadas é:

$$\frac{4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}}{h^2} + \gamma u_{i,j}e^{u_{i,j}} = f_{i,j}, \quad 1 < i, j < n$$

sendo que  $u_{0,j} = u_{n,j} = u_{i,0} = u_{i,n} = 0$ , para todo  $i, j$ , onde  $n$  é o número de pontos da malha, dado por  $n = 1/h - 1$  e  $u_{i,j}$  é o valor da função  $u$  no ponto  $(x_i, y_j) = (ih, jh)$ . Sendo que  $f(x, y)$  é escolhido tal que a solução exata seja:

$$u(x, y) = (x - x^2)(y - y^2).$$

Na tabela 3 abaixo vemos o desempenho do Ciclo FAS em dois níveis, utilizando o método de Newton para iterarmos no sistema. Este processo foi comparado à resolução do sistema não linear através do método de Newton até atingir a precisão  $10^{-10}$ , com iterações somente na malha fina. A coluna “evalf” representa o número de avaliações de função realizadas. Este número é maior que o número total de iterações pois inclui a

avaliação de função necessária a cada troca de malha. A aproximação inicial utilizada foi nula.

| Newton Puro                  |            |            |       |       |               |       |       |
|------------------------------|------------|------------|-------|-------|---------------|-------|-------|
| n, $\lambda$                 | Ciclo FAS  |            |       |       | Newton Direto |       |       |
|                              | iter em 2h | iter em 1h | evalf | tempo | iter          | evalf | tempo |
| $\lambda = 5$<br>31x31=961   | 1          | 4          | 11    | 2s    | 5             | 6     | 1s    |
| $\lambda = 10$<br>31x31=961  | 1          | 5          | 14    | 2s    | 5             | 6     | 1s    |
| $\lambda = 20$<br>31x31=961  | 2          | 5          | 15    | 2s    | 5             | 6     | 1s    |
| $\lambda = 40$<br>31x31=961  | 1          | 5          | 13    | 2s    | 6             | 7     | 1s    |
| $\lambda = 5$<br>63x63=3969  | 1          | 4          | 11    | 27s   | 5             | 6     | 6s    |
| $\lambda = 10$<br>63x63=3969 | 1          | 4          | 11    | 27s   | 5             | 6     | 6s    |
| $\lambda = 20$<br>63x63=3969 | 1          | 5          | 13    | 29s   | 5             | 6     | 6s    |
| $\lambda = 40$<br>63x63=3969 | 1          | 5          | 13    | 28s   | 6             | 7     | 7s    |

Tabela 3 - Comparação entre o ciclo FAS e iterações diretas do método de Newton

Neste teste numérico o tempo gasto pelo método multigrid foi superior ao tempo das iterações diretas, isso ocorreu pois o tempo para inicialização das matrizes para transição de malhas é bastante superior ao tempo de resolução do problema.

Os programas em Matlab utilizados para este teste numérico podem ser acessados pelo endereço:

<http://www.ime.unicamp.br/~ghaeser/fapesp/briggs>

No próximo teste numérico, implementamos o FAS de maneira semelhante ao teste anterior, a diferença é que realizamos uma iteração na malha fina no início da resolução. Na malha grossa, iteramos como anteriormente, a menos que o resíduo na malha fina seja maior do que um, neste caso, iteramos na malha grossa até a precisão de  $10^{-2}$ . A aproximação inicial utilizada foi nula.

Este mesmo teste foi executado utilizando o critério descrito acima, e os resultados foram semelhantes.



## Segundo teste numérico

O segundo problema testado foi o problema de convecção-difusão descrito por:

$$-\Delta u(x, y) + \lambda u(x, y)(u_x(x, y) + u_y(x, y)) = f(x, y)$$

para  $x, y$  no interior do quadrado unitário e  $u(x, y) = 0$  na fronteira do quadrado.  $u_x$  e  $u_y$  são as derivadas de  $u$  em relação a  $x$  e  $y$  respectivamente, que foram discretizadas por diferenças finitas centradas, assim como o Laplaciano. Logo, o sistema de equações discretizadas é:

$$\frac{4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}}{h^2} + \lambda u_{i,j} \left( \frac{u_{i+1,j} - u_{i-1,j}}{2h} + \frac{u_{i,j+1} - u_{i,j-1}}{2h} \right) = f_{i,j}$$

para  $1 < i, j < n$  e as condições de contorno são impostas fazendo  $u_{0,j} = u_{n,j} = u_{i,0} = u_{i,n} = 0$ , para todo  $i, j$ .

A função  $f(x, y)$  é escolhida tal que a solução exata do sistema seja:

$$u(x, y) = 10xy(1-x)(1-y)e^{x^{4.5}}$$

Executamos os testes para  $h = 1/32$  e  $h = 1/64$

| Newton Puro                  |            |            |       |         |               |       |         |
|------------------------------|------------|------------|-------|---------|---------------|-------|---------|
| n, $\lambda$                 | Ciclo FAS  |            |       |         | Newton Direto |       |         |
|                              | iter em 2h | iter em 1h | evalf | tempo   | iter          | evalf | tempo   |
| $\lambda = 5$<br>31x31=961   | 3          | 3          | 12    | 8s      | 4             | 5     | 8s      |
| $\lambda = 10$<br>31x31=961  | 4          | 3          | 13    | 8s      | 5             | 6     | 10s     |
| $\lambda = 20$<br>31x31=961  | 6          | 4          | 18    | 10s     | 6             | 7     | 12s     |
| $\lambda = 40$<br>31x31=961  | 7          | 4          | 19    | 10s     | 7             | 8     | 14s     |
| $\lambda = 50$<br>31x31=961  | 9          | 5          | 24    | 13s     | 8             | 9     | 17s     |
| $\lambda = 5$<br>63x63=3969  | 3          | 3          | 12    | 2min2s  | 4             | 5     | 2min11s |
| $\lambda = 10$<br>63x63=3969 | 4          | 3          | 13    | 2min6s  | 5             | 6     | 2min54s |
| $\lambda = 20$<br>63x63=3969 | 5          | 3          | 14    | 2min5s  | 6             | 7     | 3min29s |
| $\lambda = 40$<br>63x63=3969 | 7          | 4          | 19    | 2min48s | 7             | 8     | 4min7s  |
| $\lambda = 50$<br>63x63=3969 | 8          | 4          | 20    | 3min2s  | 8             | 9     | 5min15s |

Tabela 4 - Comparação entre o ciclo FAS e iterações diretas do método de Newton

Observamos que a técnica multigrid na resolução deste problema foi mais eficiente que aplicar o método de Newton na malha fina, pois obtém a solução com um tempo menor de execução. No teste com  $\lambda = 40$  e dimensão 961 o número total de iterações foi 11 sendo 7 iterações no sistema com dimensão 625 e 4 no sistema com dimensão 961. O tempo total de execução foi 4 segundos a menos no ciclo FAS. Considerando ainda que o ciclo FAS envolve as operações matriciais de troca de malha e um número maior de avaliações, notamos que este esforço é vantajoso pois resulta num número menor de iterações na malha fina e conseqüentemente em menor tempo total de execução. O teste com  $\lambda = 50$ , dimensão 961 e 3969 indica que um ajuste no processo deve ser ainda realizado pois o número total de iterações na malha grossa foi excessivo. Devemos ressaltar que este problema se torna realmente difícil para valores de  $\lambda$  maiores que 20.

Na figura 1 abaixo apresentamos seis figuras referentes à solução do problema com dimensão 961 e  $\lambda = 5$ . A primeira figura é a aproximação inicial nula, a segunda figura é a solução obtida após uma iteração do método de Newton na malha fina, podemos ver que a solução obtida já é visualmente boa. A terceira figura é a solução obtida na malha grossa após duas iterações do método de Newton, as demais figuras são as soluções obtidas

após uma iteração do método de Newton, alternando entre malha fina e malha grossa, sendo que a última figura representa a solução final na malha fina com resíduo  $3.7E-12$ .

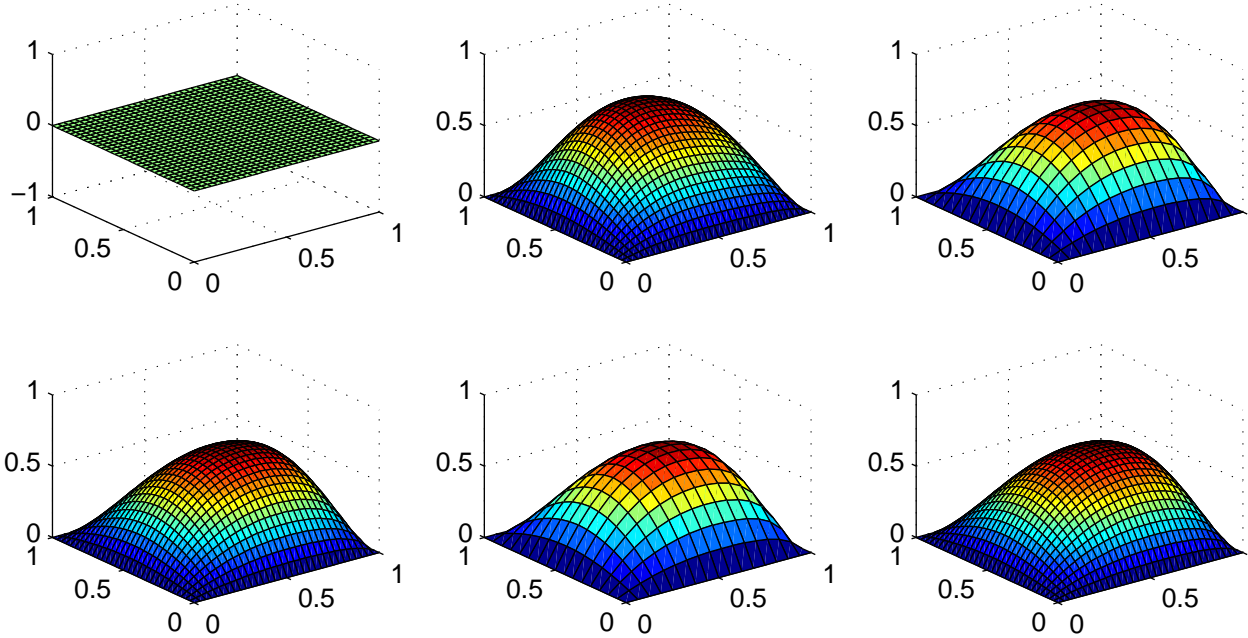


Figura 1: solução obtida após iteração de Newton alternando malhas

Os programas em Matlab utilizados para este teste numérico podem ser acessados pelo endereço:

<http://www.ime.unicamp.br/~ghaeser/fapesp/convdif>

### 2.3.5 Experimentos numéricos - Newton com busca linear

Nos testes da seção anterior, resolvemos a equação residual utilizando o método de Newton, que apresenta convergência quadrática local. Agora implementamos uma busca linear descrita em [2] que garante convergência global onde os problemas testados foram os mesmos.

No método de Newton sem busca, encontramos uma direção ao resolvermos o sistema linear  $J(x^k)d = -A(x^k)$ , encontrando  $d^k$  como solução e a nova aproximação para a solução é simplesmente  $x^{k+1} = x^k + d^k$ . Com a busca implementada, após encontrarmos  $d^k$ , devemos encontrar o tamanho do passo  $\alpha$ , tal que:

$$\|A(x^k + \alpha d^k)\| \leq (1 - \alpha 10^{-4})\|A(x^k)\| + \eta_k.$$

Onde  $\|\cdot\|$  representa a norma euclidiana e  $\eta_k$  é definida por  $\eta_k = \frac{ftip(k)}{(k+1)^{1.1}}$ , e a sequência  $ftip(k)$  é definida por:

- $ftip(0) = \|A(x^0)\|$ .
- Se  $k$  é um múltiplo de 3, então  $ftip(k) = \min\{\|A(x^k)\|, ftip(k-1)\}$ , senão  $ftip(k) = ftip(k-1)$ .

Esta busca tem por objetivo garantir um decréscimo em  $\|A\|$ . Nas iterações iniciais este decréscimo é mais tolerante (o valor em  $\|A\|$  pode até aumentar dependendo de  $\eta_k$ ). Mas, a medida em que prosseguem as iterações o valor  $\eta_k$  decresce, e aumenta portanto a exigência de um decréscimo em  $\|A\|$

Para o primeiro problema testado temos que a busca em nenhuma vez foi acionada.

Para o problema de convecção difusão, que é de resolução mais difícil, temos a seguinte tabela de resultados:

| Newton com busca linear      |            |            |           |         |               |           |         |
|------------------------------|------------|------------|-----------|---------|---------------|-----------|---------|
| n, $\lambda$                 | Ciclo Fas  |            |           |         | Newton Direto |           |         |
|                              | iter em 2h | iter em 1h | evalf     | tempo   | iter          | evalf     | tempo   |
| $\lambda = 5$<br>31x31=961   | 3          | 3          | 12        | 8s      | 4             | 5         | 8s      |
| $\lambda = 10$<br>31x31=961  | 4          | 3          | 13        | 8s      | 5             | 6         | 10s     |
| $\lambda = 20$<br>31x31=961  | 6          | 4          | 18        | 10s     | 6             | 7         | 12s     |
| $\lambda = 40$<br>31x31=961  | 7          | 4          | <b>21</b> | 10s     | 8             | <b>13</b> | 16s     |
| $\lambda = 50$<br>31x31=961  | 8          | 4          | <b>24</b> | 11s     | 8             | <b>13</b> | 17s     |
| $\lambda = 5$<br>63x63=3969  | 3          | 3          | 12        | 2min2s  | 4             | 5         | 2min10s |
| $\lambda = 10$<br>63x63=3969 | 4          | 3          | 13        | 2min2s  | 5             | 6         | 2min49s |
| $\lambda = 20$<br>63x63=3969 | 5          | 3          | 14        | 2min4s  | 6             | 7         | 3min25s |
| $\lambda = 40$<br>63x63=3969 | 7          | 4          | <b>21</b> | 2min48s | 8             | <b>13</b> | 4min44s |
| $\lambda = 50$<br>63x63=3969 | 8          | 4          | <b>24</b> | 2min55s | 8             | <b>13</b> | 4min49s |

Tabela 5 - Comparação entre o ciclo FAS e iterações diretas do método de Newton

A busca linear foi acionada nos casos com  $\lambda = 40$  e  $\lambda = 50$  que são os problemas mais difíceis. Estamos em fase de análise destes resultados para ajustar algum parâmetro (provavelmente  $\eta_k$  nas iterações iniciais) de modo a obter um algoritmo mais eficiente, embora os resultados obtidos com busca tenham sido mais eficientes.

### 3 Detalhamento dos progressos no período

Neste período foram estudadas as técnicas multigrid para sistemas lineares através de uma análise mais detalhada do ciclo “multigrid completo”. Em seguida, foi realizado um estudo introdutório sobre técnicas multigrid para sistemas não lineares. Estas técnicas permitem compor algoritmos diversos bastando alterar a forma como se alternam as iterações entre as diferentes malhas. Por esta razão o estudo computacional é bastante trabalhoso, pois, requer várias rotinas de suporte e principalmente, vários testes para obter um esquema adequado de decisão para troca de malhas. Para o caso linear foram obtidos bons resultados demonstrando que as técnicas multigrid devem ser empregadas, sempre que possível. Vimos também que detalhes no algoritmo, notadamente, o número de iterações realizadas em cada nível podem resultar em um processo mais ou menos eficiente.

No caso não linear, a técnica resulta bastante vantajosa em problemas de resolução mais difícil, pois nestes casos, o método de Newton realiza um número alto de iterações e se a malha fina tem dimensão grande, cada iteração de Newton é bastante cara. Nestes casos, alternar iterações entre a malha fina e malhas mais grossas, acarreta num processo com custo computacional menor, ainda que resulte em um número total de iterações maior, pois iterações em malhas mais grossas realizam um número menor de operações.

Este projeto permitiu um forte aprendizado em elaboração e implementação de algoritmos e resultou em um bom trabalho de pesquisa para o aluno. Certamente, este tipo de atividade contribuirá para a sua formação, uma vez que o aluno já está aceito para o curso de Pós-Graduação em Matemática Aplicada, sendo que obteve nota acima de 7.0 no exame de admissão ao curso, o que possibilita que curse diretamente o doutorado, de acordo com as regras do curso.

O trabalho será apresentado em dois congressos: XXVI CNMAC - Congresso Nacional em Matemática Aplicada e Computacional a ser realizado entre os dias 8 e 11 setembro de 2003 e no Congresso Interno de Iniciação Científica a ser realizado no final de setembro de 2003. Estamos elaborando um relatório de pesquisa que será publicado aqui no Imecc. Agradecemos à Fapesp por apoiar a realização deste trabalho.

## Referências

- [1] Briggs, William L., *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.
- [2] Diniz-Ehrhardt, M.A., Gomes-Ruggiero, M.A., Lopes, V.L., Martínez, J.M., *Discrete Newton's method with local variations for solving large-scale nonlinear systems*, RP63/02, aceito para publicação em OPTIMIZATION.
- [3] Brandt, A., *Multi-Level Adaptive Solutions to Boundary-Value Problems*, Mathematics of Computation, Vol.31, 138, pp.:333–390.
- [4] Gomes-Ruggiero, M.A., Kozakevich D. N. e Martínez, J. M. : *A Numerical Study on Large-Scale Nonlinear Solver*, Computers and Mathematics with Applications, Vol.32, nº 3, págs.: 1–13, 1996.
- [5] Friedlander, A., Gomes-Ruggiero, M. A., Kozakevich, D.N, Martínez, J. M., Santos, S. A.: *Solving Nonlinear Systems of Equations by Means of Quasi-Newton Methods with a Nonmonotone Strategy* – Optimization Methods and Software, Vol.8, págs.:25-51, 1997.
- [6] Hackbusch, W. and Trottenberg, U (editors), *Multigrid Methods*, Lecture Notes in Mathematics, 960, Springer, 1982.
- [7] Knoll, D.A. and Rider, W. J., *A Multigrid Preconditioned Newton-Krylov Method*, SIAM J. Sci. Comput., vol.21, no.2, pp.:691–710.
- [8] Wesseling, P., *An Introduction to Multigrid Methods*, John Wiley & Sons, 1991.
- [9] Adams L., J.L. Nazareth, *Linear and Nonlinear Conjugate Gradient - Related Methods*, SIAM
- [10] Shewchuk, J. R., *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, <http://www-2.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.ps>
- [11] Saad Y., Schultz M., *GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci Statist. Comput., 7(1986), pp.:856–869.

Gabriel Haeser

Márcia A. Gomes Ruggiero