

# MÉTODOS MULTIGRID EM MALHAS NÃO-ANINHADAS APLICADOS A PROBLEMAS ELÁSTICOS

MARCO LÚCIO BITTENCOURT\*

e

RAÚL ANTONIO FEIJÓO\*\*

*\* Departamento de Projeto Mecânico (DPM)  
Faculdade de Engenharia Mecânica (FEM)  
Universidade Estadual de Campinas (UNICAMP)  
Caixa Postal 6051, CEP 13083-971, Campinas/SP, Brasil  
Tel: + 55-19-788 8384 Fax: + 55-19-239 3722  
E-mail: mlb@dpm.fem.unicamp.br*

*\*\* Laboratório Nacional de Computação Científica (LNCC/CNPq)  
Rua Lauro Muller 455, Caixa Postal 56018, CEP 22290-000, Rio de Janeiro/RJ, Brasil  
Tel: + 55-21-541 2132 Fax: + 55-21-295 7944  
E-mail: feij@server01.lncc.br*

## SUMÁRIO

Neste trabalho apresentam-se conceitos de métodos multigrid aplicados a problemas elásticos lineares. Consideram-se exemplos bi e tridimensionais com malhas não-estruturadas e não-aninhadas. Tradicionalmente, os métodos multigrid têm sido utilizados com malhas aninhadas devido a maior simplicidade dos operadores de transferência. No entanto, para modelar problemas de engenharia, a aplicação de malhas não-estruturadas é mais conveniente. Comparações com métodos diretos e iterativos são apresentados, estendendo os resultados dados em<sup>3</sup>.

## MULTIGRID METHODS IN NON-NESTED MESHES APPLIED TO ELASTIC PROBLEMS

### SUMMARY

In this paper, multigrid methods applied to linear elastic problems are presented, taking two and three dimensional examples modelled by non-structured and no-nested meshes. Generally, the multigrid methods have been used with nested meshes which simplify the transfer operators. However, to model engineering problems non-structured meshes are more suitable. Comparisons with direct and iterative methods are presented, extending the results indicated in<sup>3</sup>.

Recibido: Noviembre 1996

## INTRODUÇÃO

Em geral, na aplicação do Método de Elementos Finitos (MEF) para a solução de problemas de engenharia, é necessário resolver um sistema de equações de seguinte forma

$$\mathbf{A}\mathbf{u} = \mathbf{b} \quad (1)$$

onde a matriz  $\mathbf{A}_{N \times N}$  é em geral simétrica e positiva-definida (SPD);  $\mathbf{u}_{N \times 1}$  e  $\mathbf{b}_{N \times 1}$  são os vetores das incógnitas e termos independentes, respectivamente.

Para resolver o sistema (1), pode-se empregar métodos diretos e iterativos. A aplicação destes algoritmos para problemas elásticos lineares está apresentada em<sup>3</sup>. Uma outra técnica empregada na resolução de sistemas de equações são os métodos multigrid<sup>1,2,8</sup>, os quais proporcionam uma maior taxa de convergência dos métodos iterativos, atingindo uma ordem linear  $\mathcal{O}(N)$  para  $N$  grande.

Basicamente, os métodos multigrid consideram uma sequência de malhas para a solução do sistema de equações. O objetivo é resolver o problema na malha fina empregando as demais malhas como esquemas de correção. Tal fato é empregado devido a propriedade espectral dos métodos iterativos estacionários, onde os modos de alta frequência do erro de aproximação são eliminados nas primeiras iterações através do algoritmo de relaxação empregado.

Observa-se que tradicionalmente os métodos multigrid têm sido aplicados em malhas denominadas aninhadas, ou seja, os nós de uma malha grossa estão sempre presentes na malha fina subsequente, como ilustrado na Figura 1a). Esta propriedade permite simplificar a forma dos operadores para a transferência de informações entre malhas. No entanto, os problemas de engenharia possuem domínios com contornos complexos dificultando a obtenção de malhas aninhadas. Para contornar esta limitação, empregam-se malhas não-aninhadas, como também mostrado na Figura 1b).

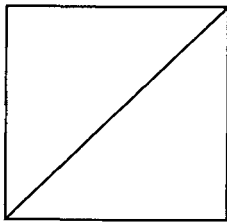


Figura 1 a) Malhas aninhadas

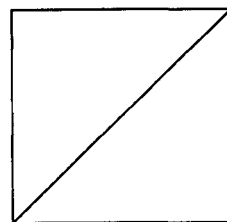
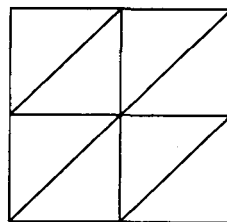
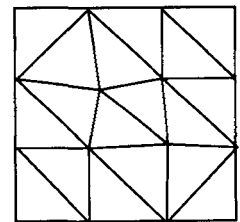


Figura 1 b) Malhas não-aninhadas



O objetivo deste trabalho é estender os resultados apresentados em<sup>3</sup>, considerando a aplicação de métodos diretos, iterativos e multigrid para a solução de sistemas de equações obtidos da aplicação do MEF a problemas elásticos lineares. Estes resultados estão na forma do custo computacional e espaço de memória.

Inicialmente, discute-se o comportamento dos métodos iterativos estacionários, ressaltando a propriedade espectral dos mesmos. Consideram-se então os elementos

principais dos métodos multigrid, tais como iterações aninhadas, correção de malha grossa e operadores de transferência. Posteriormente, abordam-se várias estratégias multigrid e o custo computacional associado. Finalmente, apresentam-se os resultados da aplicação dos métodos multigrid a problemas elásticos bi e tridimensionais, comparando-se a performance em termos do número de operações e espaço de memória, com os procedimentos de Gauss e iterativos baseados em gradiente conjugado.

## COMPORTAMENTO DOS MÉTODOS ITERATIVOS BÁSICOS

Visando estudar o comportamento dos métodos iterativos básicos, toma-se o PVC unidimensional homogêneo  $-u''(x) = 0$  ( $0 < x < 1$ ) com condições de contorno homogêneas<sup>5</sup>  $u(0) = u(1) = 0$ .

Empregando o método de diferenças finitas ao PVC anterior, o domínio do problema é particionado em  $N$  subintervalos pelas coordenadas  $x_j = jh$  definindo uma malha  $\Omega^h$ , com o tamanho do elemento  $h = \frac{1}{N}$ . Em cada um dos  $N - 1$  pontos interiores, introduz-se uma aproximação de segunda ordem  $\frac{1}{h^2}(-u_{j-1} + 2u_j - u_{j+1}) = 0$  ( $1 \leq j \leq N - 1$ ) com  $u_0 = u_N = 0$ , originando um sistema  $\mathbf{A}\mathbf{u} = \mathbf{0}$ . Como o sistema é homogêneo, e tomando-se uma aproximação inicial arbitrária  $\mathbf{v}$ , a solução exata  $\mathbf{u} = \mathbf{0}$  é conhecida e o erro algébrico é simplesmente  $\mathbf{e} = -\mathbf{v}$ .

Adota-se como solução inicial  $\mathbf{v}$  os vetores ou modos de Fourier  $v_j = \sin\left(\frac{jk\pi}{N}\right)$  ( $0 \leq j \leq N$ ) ( $1 \leq k \leq N - 1$ ). O índice  $j$  denota a componente do vetor  $\mathbf{v}_k$ , onde  $k$  é o número de onda representando a quantidade de meio-senos presentes em  $\mathbf{v}_k$ . Observa-se que valores pequenos de  $k$  correspondem a ondas longas e suaves, enquanto valores maiores estão relacionados a modos altamente oscilatórios.

Utiliza-se o método de Jacobi<sup>5</sup> ponderado com coeficiente de relaxação  $\omega = 2/3$  numa malha com  $N = 64$ . Empregam-se as aproximações iniciais  $\mathbf{v}_1$ ,  $\mathbf{v}_3$  e  $\mathbf{v}_6$  executando 100 iterações. Em cada passo, calcula-se a norma infinita do erro  $\|\mathbf{e}\|_\infty$ . A Figura 2a) ilustra o logaritmo do erro em função do número de iterações. Verifica-se um decrescimento linear do erro, sendo a taxa mais pronunciada para números de onda maiores.

Em geral, a aproximação inicial contém componentes de vários vetores de Fourier. Assim na Figura 2b), tem-se a norma do erro  $\|\mathbf{e}\|_\infty$  para uma solução inicial dada pela soma dos modos de baixa ( $k = 1$ ), média ( $k = 6$ ) e alta ( $k = 32$ ) frequências. Neste caso, observa-se uma queda acentuada do erro nas 5 primeiras relaxações e uma estagnação na taxa de decrescimento nas demais iterações.

Os modos de Fourier podem ser classificados como *modos suaves ou de baixa frequência* para  $1 \leq k \leq N/2$  ou como *modos oscilatórios ou de alta frequência* para  $N/2 \leq k \leq N - 1$ . Assim, na Figura 2b), o decrescimento inicial corresponde a rápida eliminação dos modos oscilatórios, enquanto os modos suaves implicam num decrescimento menor da norma do erro.

Esta é uma propriedade típica dos métodos estacionários, ou seja, observa-se uma rápida convergência quando o erro possui componentes de alta frequência. Da mesma maneira, tem-se uma baixa performance na eliminação das componentes suaves,

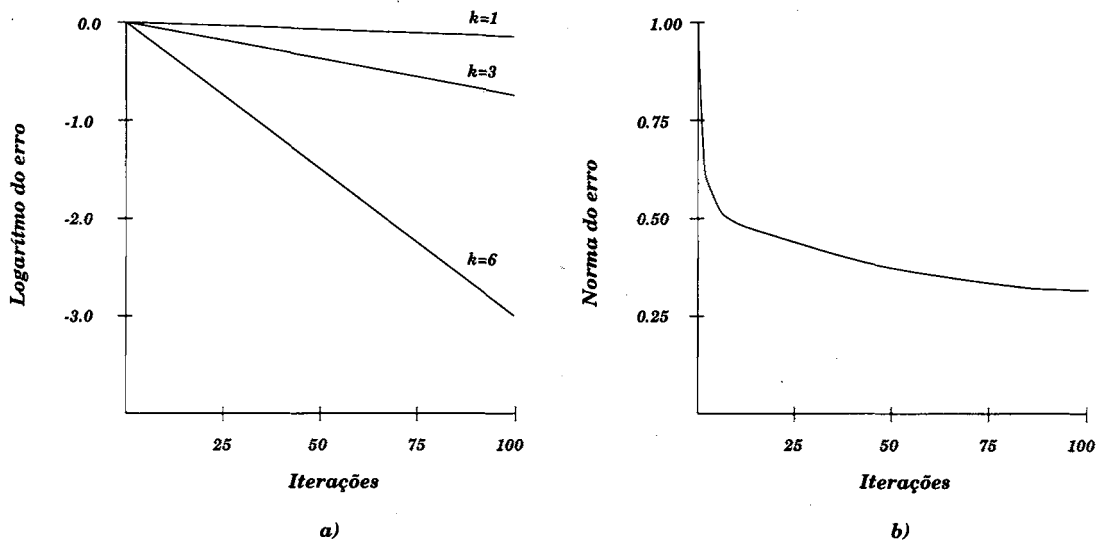


Figura 2. Norma infinita do erro para cada iteração com soluções iniciais<sup>5</sup>:  
 a)  $\mathbf{v} = \mathbf{v}_1$ ,  $\mathbf{v} = \mathbf{v}_3$ ,  $\mathbf{v} = \mathbf{v}_6$ ; b)  $\mathbf{v} + \frac{1}{3}(\mathbf{v}_1 + \mathbf{v}_6 + \mathbf{v}_{32})$

degradando a performance destes métodos. Denomina-se esta característica como *propriedade de suavização*<sup>5</sup>.

As técnicas multigrid procuram evitar esta limitação dos algoritmos estacionários, empregando os elementos discutidos na próxima seção. A função dos métodos iterativos é eliminar em cada malha as componentes oscilatórias do erro através de algumas poucas iterações.

Portanto, ao se verificar uma estagnação no algoritmo de relaxação na malha fina, indicando que as componentes oscilatórias foram eliminadas, restando apenas os modos suaves, deve-se passar para a malha grossa. Desta forma, os modos suaves projetados da malha fina tornam-se oscilatórios, sendo eficientemente eliminados através de relaxações. Aplicam-se esquemas de relaxação simples como Jacobi e Gauss-Seidel, pois não se procura alcançar a convergência em cada malha, mas sim a solução do problema proposto utilizando várias malhas.

## ELEMENTOS DOS MÉTODOS MULTIGRID

### Iterações aninhadas

Uma forma de aumentar a performance de um esquema de relaxação é empregar uma melhor aproximação inicial, obtida por exemplo através de iterações numa malha grossa. Como esta malha possui um número menor de variáveis, tem-se um menor custo computacional em relação a uma iteração realizada na malha mais fina  $\Omega^h$ .

Assim, a seguinte estratégia, denominada *iteraões aninhadas*, permite obter uma melhor aproximação inicial para a malha mais fina<sup>5</sup>.

Iterar em  $\mathbf{Au} = \mathbf{b}$  na malha mais grossa

⋮

Iterar em  $\mathbf{Au} = \mathbf{b}$  na malha  $\Omega^{4h}$  para obter uma aproximação inicial para  $\Omega^{2h}$

Iterar em  $\mathbf{Au} = \mathbf{b}$  na malha  $\Omega^{2h}$  para obter uma aproximação inicial para  $\Omega^h$

Iterar em  $\mathbf{Au} = \mathbf{b}$  na malha mais fina  $\Omega^h$  para obter uma aproximação final para a solução  $\mathbf{u}$

Esta estratégia não garante que ao final, a solução em  $\Omega^h$  não contenha componentes suaves. A possibilidade de empregar a correção do erro proviene das malhas grossas evita esta limitação.

### Correção de malha grossa

Verifica-se que relaxar na equação original  $\mathbf{Au} = \mathbf{b}$ , com uma aproximação  $\mathbf{v}$ , é equivalente a relaxar na equação do resíduo  $\mathbf{Ae} = \mathbf{r}$ , com aproximação inicial  $\mathbf{e} = \mathbf{0}$ . A partir deste fato, propõe-se incorporar a equação de resíduo para relaxar sobre o erro, permitindo definir o procedimento de *correção de malha grossa* da seguinte maneira

Iterar em  $\mathbf{Au} = \mathbf{b}$  na malha  $\Omega^h$  para obter uma aproximação  $\mathbf{v}^h$

Calcular  $\mathbf{r} = \mathbf{b} - \mathbf{Av}^h$

Iterar na equação  $\mathbf{Ae} = \mathbf{r}$  em  $\Omega^{2h}$  determinando uma aproximação  $\mathbf{e}^{2h}$

Corrigir a aproximação obtida em  $\Omega^h$  com o erro estimado em  $\Omega^{2h}$ :  $\mathbf{v}^h \leftarrow \mathbf{v}^h + \mathbf{e}^{2h}$

Portanto, após relaxar na malha fina  $\Omega^h$  até que a convergência se deteriore, passa-se a iterar na equação de resíduo numa malha mais grossa  $\Omega^{2h}$ , obtendo-se uma aproximação para o erro, a qual corrige a solução na malha fina.

### Operadores de transferência entre malhas

Os elementos discutidos nas seções anteriores revelam a necessidade de operadores para transferir informações entre as malhas. Inicialmente, assume-se que a malha grossa está contida na malha fina, ou seja, estão aninhadas (*nested meshes*).

O primeiro operador transfere funções da malha grossa  $\Omega^{2h}$  para a malha fina  $\Omega^h$ , sendo denominado *operador de interpolação ou pronlogamento* e denotado por  $I_{2h}^h$ . É empregado para se passar o erro  $\mathbf{e}^{2h}$  ou aproximação inicial  $\mathbf{v}^{2h}$  da malha grossa para a fina. Logo

$$\begin{aligned} I_{2h}^h : \Omega^{2h} &\rightarrow \Omega^h \\ \mathbf{v}^{2h} &\rightarrow \mathbf{v}^h = I_{2h}^h \mathbf{v}^{2h} \end{aligned} \quad (2)$$

A Figura 3 mostra o esquema de interpolação linear. Para o caso unidimensional tem-se

$$\begin{cases} v_{2j}^h = v_j^{2h} & 0 \leq j \leq \frac{N}{2} - 1 \\ v_{2j+1}^h = \frac{1}{2}(v_j^{2h} + v_{j+1}^{2h}) \end{cases} \quad (3)$$

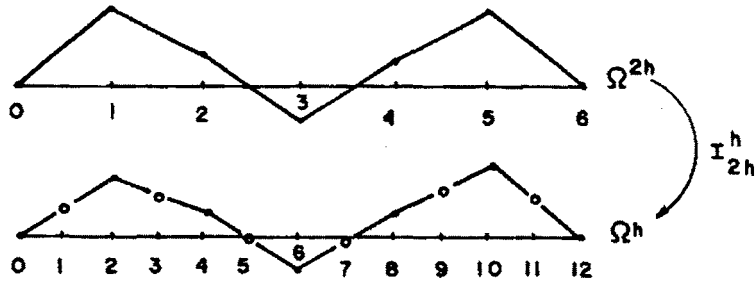


Figura 3. Interpolação linear de uma função da malha grossa  $\Omega^{2h}$  para a malha fina  $\Omega^h$

O segundo operador  $I_h^{2h}$  transforma funções da malha fina  $\Omega^h$  para a grossa  $\Omega^{2h}$ , como ao se projetar o resíduo  $r^h$  em  $\Omega^{2h}$ , sendo conhecido como *operador de restrição* e denotado por

$$\begin{aligned} I_h^{2h} : \Omega^h &\rightarrow \Omega^{2h} \\ \mathbf{v}^h &\rightarrow \mathbf{v}^{2h} = I_h^{2h} \mathbf{v}^h \end{aligned} \quad (4)$$

O tipo de restrição mais simples em malhas aninhadas é o operador injetivo

$$v_j^{2h} = v_{2j}^h \quad (5)$$

ou seja, os pontos da malha grossa assumem os valores correspondentes da malha fina.

Uma outra forma de restrição emprega uma ponderação, como ilustrado na Figura 4, sendo dado por

$$v_j^{2h} = \frac{1}{4}(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h) \quad 1 \leq j \leq \frac{N}{2} - 1 \quad (6)$$

No entanto, neste trabalho, considera-se que uma malha grossa não está necessariamente contida na malha fina seguinte, ou seja, não se tem uma relação fixa de  $2^{-d}$  entre os tamanhos dos elementos, onde  $d$  é a dimensão do problema. Designando por  $N^h$  os nós da malha fina  $\Omega^h$ , observa-se que  $N^{2h}$  não está contido em  $N^h$ . Esta condição define o que é denominado na literatura como malhas não-aninhadas (*non-nested meshes*).

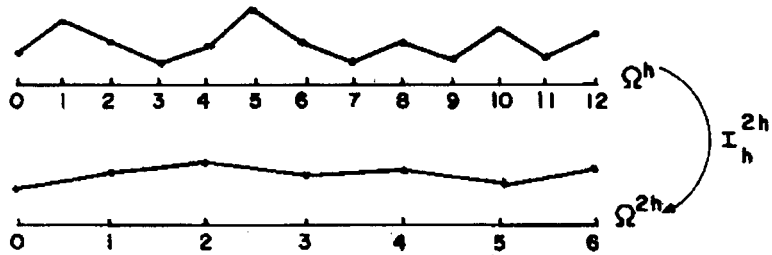


Figura 4. Restrição ponderada de uma função da malha fina  $\Omega^h$  para a malha grossa  $\Omega^{2h}$

### Extensão dos operadores para malhas não-aninhadas

Supondo que um problema bidimensional seja resolvido pelo MEF empregando triângulos lineares, definem-se os operadores de restrição e pronlogamento utilizando as funções de forma dos elementos, dadas em coordenadas de área  $A_i (i = 1, 2, 3)$ . A Figura 5 apresenta duas malhas triangulares permitindo observar a forma dos operadores. De maneira geral, basta identificar para cada nó fino  $I$ , o triângulo grosso  $T_g$  contendo  $I$ . A partir daí, determinam-se as coordenadas de área do nó  $I$ , tomando-se o elemento  $T_g$ .

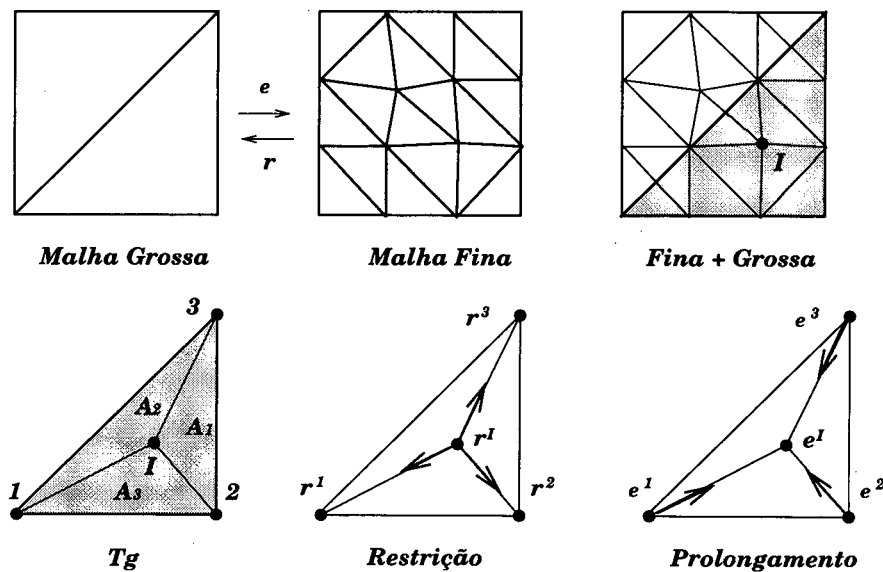


Figura 5. Operadores de restrição e interpolação para malhas 2D não-aninhadas

Inicialmente, deseja-se projetar o resíduo da malha fina para a grossa. Considerando a notação utilizada na Figura 5, o operador de restrição determina o

resíduo em cada nó do triângulo  $T_g$ , a partir da contribuição dos resíduos dos nós finos presentes em  $T_g$ . Logo

$$\mathbf{r}^i = I_h^{2h} \mathbf{r}^h = \sum_{l=1}^{N_f} A_l^I \mathbf{r}_l^I \quad i = 1, 2, 3 \quad (7)$$

onde  $N_f$  é o número de nós finos  $I$ , com respectivo resíduo  $\mathbf{r}^I$  e coordenadas  $A^I$ , contidos no triângulo da malha grossa  $T_g$ .

Analogamente, para mapear o erro da malha grossa para a fina basta utilizar uma interpolação via as funções de forma de  $T_g$ . Logo

$$\mathbf{e}^I = I_{2h}^h \mathbf{e}^{2h} = \sum_{l=1}^3 A_l^I \mathbf{e}^l \quad (8)$$

Observa-se que a extensão destes operadores para problemas tridimensionais empregando tetraedros lineares é análoga, devendo-se usar as coordenadas de volume  $V_i (i = 1, \dots, 4)$ .

A possibilidade de utilizar malhas não-estruturadas permite uma maior flexibilidade na aplicação dos algoritmos multigrid, principalmente para domínios complexos onde a geração da malha torna-se trabalhosa ao se impor uma fator fixo de  $2^{-d}$  no tamanho dos elementos entre malhas consecutivas. Porém, no caso de malhas não-estruturadas este fator deve estar próximo  $2^{-d}$  visando garantir a convergência do algoritmo. Uma dedução variacional dos operadores e aspectos de convergência estão apresentados em<sup>2</sup>.

### ESTRATÉGIAS MULTIGRID

A partir da definição dos operadores de transferência, pode-se reescrever a estratégia de correção de malha grossa (CMG) como<sup>5</sup>

$\mathbf{v}^h \leftarrow CMG(\mathbf{v}^h, \mathbf{b}^h)$ <p>Iterar <math>\nu_1</math> vezes em <math>\mathbf{A}^h \mathbf{u}^h = \mathbf{b}^h</math> na malha <math>\Omega^h</math> com aproximação inicial <math>\mathbf{v}^h</math></p> <p>Calcular <math>\mathbf{r}^{2h} = I_h^{2h} (\mathbf{b}^h - \mathbf{A}^h \mathbf{v}^h)</math></p> <p>Resolver a equação <math>\mathbf{A}^{2h} \mathbf{e}^{2h} = \mathbf{r}^{2h}</math> em <math>\Omega^{2h}</math></p> <p>Corrigir a aproximação da malha fina <math>\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{e}^{2h}</math></p> <p>Iterar <math>\nu_2</math> vezes em <math>\mathbf{A}^h \mathbf{u}^h = \mathbf{b}^h</math> na malha <math>\Omega^h</math> com aproximação inicial <math>\mathbf{v}^h</math></p>
--

Portanto, itera-se  $\nu_1$  vezes na malha fina até que a taxa de decrescimento do erro se estabilize. Calcula-se o resíduo na aproximação  $\mathbf{v}^h$  em  $\Omega^h$  e transfere-se o mesmo, através do operador de restrição  $I_h^{2h}$ , para a malha grossa  $\Omega^{2h}$ . Determina-se a solução  $\mathbf{e}^{2h}$  da equação do resíduo e aplica-se o operador de prolongamento  $I_{2h}^h$  para corrigir a solução corrente  $\mathbf{v}^h$  em  $\Omega^h$ . Executam-se então  $\nu_2$  iterações para eliminar as componentes oscilatórias de  $\mathbf{v}^h$ .



Alguns comentários podem ser feitos sobre a estratégia de correção definida:

- em relação ao número de pré e pós-relaxações  $\nu_1$  e  $\nu_2$ , estabelecem-se valores fixos para os mesmos no início do algoritmo. Outra possibilidade é utilizar um critério adaptável onde se calcula a norma do resíduo após cada iteração. Caso entre duas iterações sucessivas não se verificar uma queda em torno de 60 % na norma de  $\mathbf{r}^{nh}$  em  $\Omega^{nh}$ , deve-se passar para a próxima malha<sup>4</sup>;
- o termo  $\mathbf{A}^{2h}$  indica a representação na malha  $\Omega^{2h}$  da matriz  $\mathbf{A}^h$ . Para malhas aninhadas, tem-se uma propriedade variacional relacionando estas matrizes<sup>5</sup>

$$\mathbf{A}^{2h} = I_h^{2h} \mathbf{A}^h I_{2h}^h \quad I_h^{2h} = c(I_{2h}^h)^T \quad c \in \mathcal{R} \quad (9)$$

Para malhas não-aninhadas, deve-se calcular para cada malha  $\Omega^{nh}$  a matriz  $\mathbf{A}^{nh}$  correspondente através do método usado na solução do problema;

- a equação de resíduo na malha grossa  $\mathbf{A}^{2h} \mathbf{e}^{2h} = \mathbf{r}^{2h}$  é análoga ao sistema  $\mathbf{A}^h \mathbf{u}^h = \mathbf{b}^h$  em  $\Omega^h$ . Assim, torna-se viável aplicar o mesmo esquema de correção anterior utilizando agora uma malha  $\Omega^{4h}$ . Este procedimento pode ser aplicado sucessivamente chagando-se a um algoritmo recursivo onde a malha mais grossa, possuindo um número menor de variáveis, é resolvida de forma direta.

A recursividade na estratégia de correção da malha grossa permite definir uma família de métodos multigrid, denominada Ciclos  $\mu$ , dada por<sup>5</sup>

$$\mathbf{v}^h \leftarrow M_{\mu}^h(\mathbf{v}^h, \mathbf{b}^h)$$

1. Iterar  $\nu_1$  vezes em  $\mathbf{A}^h \mathbf{u}^h = \mathbf{b}^h$  com aproximação inicial  $\mathbf{v}^h$

2. Se  $\Omega^h =$  malha mais grossa  $\rightarrow$  passo 4

$$\text{Senão } \mathbf{b}^{2h} \leftarrow I_h^{2h}(\mathbf{b}^h - \mathbf{A}^h \mathbf{v}^h)$$

$$\mathbf{v}^{2h} \leftarrow \mathbf{0}$$

$$\mathbf{v}^{2h} \leftarrow M_{\mu}^{2h}(\mathbf{v}^{2h}, \mathbf{b}^{2h}) \nu_0 \text{ vezes}$$

3. Corrigir  $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{v}^{2h}$

4. Iterar  $\nu_2$  vezes em  $\mathbf{A}^h \mathbf{u}^h = \mathbf{b}^h$  com aproximação inicial  $\mathbf{v}^h$

Na notação usada,  $\mathbf{v}^{2h}$  indica a solução da equação do resíduo  $\mathbf{e}^{2h}$ , da mesma maneira  $\mathbf{b}^{2h}$  é utilizando ao invés de  $\mathbf{r}^{2h}$ , pois os mesmos são termos independentes dos sistemas de equações envolvidos.

Para  $\nu_0 = 1$ , tem-se o Ciclo  $\mu = V$ , o qual partindo da malha mais fina alcança a malha mais grossa mapeando o resíduo entre as malhas, retornando para a malha mais fina aplicando as correções em cada nível. Para  $\nu_0 = 2$ , define-se o Ciclo  $\mu = W$  que juntamente com o esquema  $V$  estão ilustrados na Figura 6.

No entanto, as estratégias multigrid apresentadas não incorporam o conceito de iterações aninhadas. Para isso, tem-se a estratégia FMV, onde cada ciclo  $V$  é precedido por um ou mais ciclos  $V$  menores, conforme o valor do parâmetro  $\nu_0$ , com o objetivo de prover uma melhor aproximação inicial em cada nível. Uma outra estratégia denominada FMW, utiliza ciclos  $W$  menores para obter uma melhor aproximação inicial

para um próximo nível, como apresentado juntamente com o procedimento FMV na Figura 6 para  $\nu_0 = 1$ . Algumas variantes destes algoritmos podem ser empregados. Por exemplo, ao final de um ciclo FMV, concatenam-se vários ciclos  $V$ , sendo este esquema denominado FMVV.

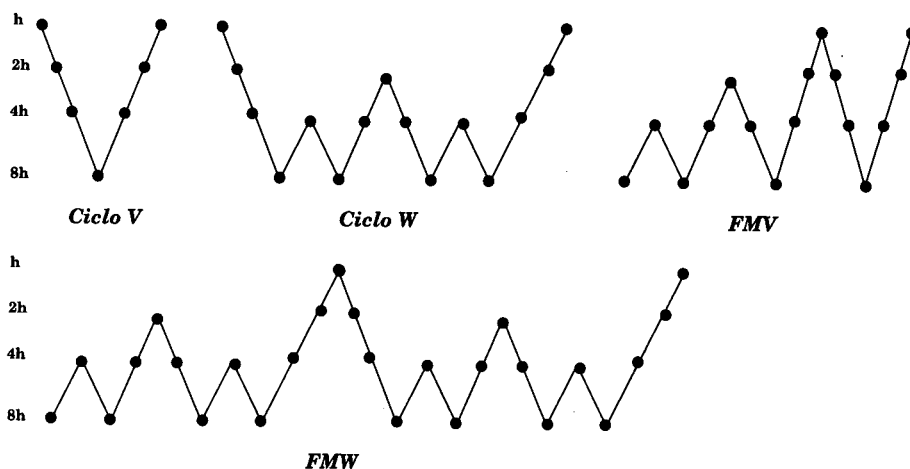


Figura 6. Estratégias multigrid

## CUSTOS DOS ESQUEMAS MULTIGRID

Torna-se fundamental determinar o custo computacional e o espaço de memória dos algoritmos multigrid discutidos na seção anterior. Em geral, o custo é representado como o número equivalente de iterações do algoritmo de relaxação na malha fina.

Supondo que sejam empregadas  $N_m$  malhas para a solução de um problema por multigrid, uma forma mais precisa para estimar o número de operações deve considerar os seguintes itens.

### Iterações

Excetuando a malha mais grossa, na qual aplica-se método direto de Gauss, a demanda computacional em cada nível  $i$  será igual ao número total de iterações realizado ( $NIT_i$ ) em todos os ciclos vezes o custo da iteração ( $CIT_i$ ) do método numérico empregado. Logo

$$\sum_{i=2}^{N_m} (NIT_i)(CIT_i) \quad (10)$$

Utilizando como esquema de relaxação o algoritmo de Gauss-Seidel tem-se que  $CIT_i = N_i(2m_i - 1)$ , onde  $N_i$  e  $m_i$  denotam respectivamente o número de equações e o número médio de elementos por linha na matriz esparsa da malha  $i$ .

### ***Solução direta na malha grossa***

Neste caso, deve-se fatorar a matriz  $\mathbf{A}$  uma única vez e empregar processos de substituição sempre que a malha mais grossa for visitada. Portanto, o custo computacional destes procedimentos é dado por<sup>3</sup>

$$\left[ \sum_{i=1}^{N_1} m_i^U (m_i^U + 3) \right] + 2N_1(m_1 - 1)n_1 \quad (11)$$

onde  $m_i^U = m_i - 1$  e  $n_1$  representa o número total de vezes que a técnica multigrid utilizada passou pelo nível 1. Por exemplo, para um ciclo  $\mathbf{W}$  tem-se  $n_1 = 4$ .

### ***Operadores***

A partir da Figura 5, no caso de problemas bidimensionais e triângulos lineares, observa-se que em cada equação de um nó fino  $I$  são necessárias 3 multiplicações para mapear o resíduo ou a correção nos respectivos operadores. Portanto, tem-se um custo de  $3N_i$  operações em cada nível  $i$  sempre que uma restrição ou um pronlogamento é executado. Para casos tridimensionais, como os tetraedros lineares possuem 4 nós, o custo passa a ser de  $4 N_i$ .

Denotando  $n_i^r$  e  $n_i^p$  como o número total de operações de restrição e pronlogamento executados nos ciclos da estratégia multigrid, o custo dos operadores é dado por

$$N_{\text{nos}} \sum_{i=2}^{N_m} N_i (n_i^r + n_i^p) \quad (12)$$

sendo  $N_{\text{nos}}$  o número de nós do elemento, 3 para triângulos e 4 para tetraedros.

### ***Cálculo do resíduo***

Ao se aplicar o operador de restrição entre as malhas  $i$  a  $i - 1$ , deve-se calcular o resíduo  $\mathbf{r}_i$  demandando  $N_i(2m_i - 1)$  operações. Assim, para todos os níveis, tem-se o seguinte custo

$$\sum_{i=2}^{N_m} N_i (2m_i - 1) n_i^r \quad (13)$$

Observa-se que o cálculo do resíduo é igual a uma iteração do método de Gauss-Seidel no mesmo nível  $i$ . Neste sentido, a estratégia de correção adaptável discutida anteriormente não é interessante, pois o cálculo do resíduo a cada iteração do esquema de relaxação demanda um esforço equivalente ao da própria iteração.

O número de operações total para a solução de um sistema de equações via técnicas multigrid em malhas não-aninhadas é igual a soma das expressões (10) a (13). Para se obter o número equivalente de iterações na malha fina, basta dividir o valor obtido pelo custo de uma iteração do esquema de relaxação utilizado.

No que se refere ao espaço de memória, tem-se uma demanda computacional superior em relação aos métodos iterativos<sup>3</sup>, não apenas pelo fato de se trabalhar

com várias malhas, mas também pela necessidade de se armazenar informações auxiliares utilizadas nos operadores de restrição e pronlogamento, tais como incidência e numeração das variáveis. Desta forma, deve-se contabilizar o espaço de memória para os seguintes parâmetros:

### **Matrizes + vetores**

Neste caso, consideram-se as matrizes e os vetores dos sistemas de equações de cada malha. Para um nível  $i$ , tem-se um total de  $N_i(m_i + 2)$  posições de memória, além do espaço  $n_i^{aux}$  dos vetores auxiliares do esquema de relaxação empregado. Por exemplo, para o método de Gauss-Seidel em matrizes esparsas é necessário mais um vetor fazendo  $n_i^{aux} = N_i$  (ref.<sup>2,3</sup>). Logo, de forma geral o número de posições de ponto flutuante é dado por

$$\sum_{i=1}^{N_m} [N_i(m_i + 2) + n_i^{aux}] \quad (14)$$

### **Tabelas de mapeamento**

Para aplicar os operadores de transfêrencia, deve-se conhecer para cada nó fino  $I$ , o triângulo da malha grossa que o contém e neste elemento o valor das coordenadas de área ( $A_1, A_2, A_3$ ) do nó  $I$ . Por exemplo, supondo que o nó fino 100 está contido no triângulo 32 e possui coordenadas de área (0, 1; 0, 5; 0, 4) armazenam-se os seguintes dados

$$32 \quad 0,1 \quad 0,5$$

onde emprega-se a relação  $A_3 = 1 - A_1 - A_2$ .

Observa-se que o número do elemento, armazenado como um inteiro, requer a metade do número de *bytes* de cada coordenada, considerada como um termo de precisão dupla. Logo, no total são necessárias 2,5 posições de ponto flutuante. No caso tridimensional, tem-se 4 coordenadas de volume  $V_i (i = 1, \dots, 4)$  resultando um total de 3,5 posições.

Portanto, estas tabelas necessitam do seguinte espaço de precisão dupla

$$n_t \sum_{i=2}^{N_m} N_i^{nos} \quad n_t = 2,5 \quad \text{ou} \quad n_t = 3,5 \quad (15)$$

onde  $N_i^{nos}$  é o número total de nós da malha  $i$ .

### **Equações**

Armazena-se para cada nó, a numeração das suas variáveis. Para problemas elásticos bi e tridimensionais, tem-se, respectivamente, 2 e 3 variáveis em cada nó, resultando em

$$n_{eq} \sum_{i=1}^{N_m} N_i^{\text{nos}} \quad (16)$$

posições de precisão dupla com  $n_{eq} = 1$  e  $n_{eq} = 1,5$ , respectivamente para problemas planos e espaciais.

### ***Incidência***

Da mesma forma, armazena-se a incidência de cada elemento necessitando o seguinte número de posições de precisão dupla

$$n_{inc} \sum_{i=1}^{N_m} N_i^{\text{els}} \quad (17)$$

sendo  $N_i^{\text{els}}$  o número de elementos da malha  $i$ ; com  $n_{inc} = 1,5$  e  $n_{inc} = 2$ , respectivamente para problemas bi e tridimensionais.

O espaço de memória total é dado pela soma das expressões (14) a (17), lembrando-se que o valor obtido deve ser multiplicado pelo fator  $\frac{8}{1024}$  para se ter resultados em Kbytes.

## **RESULTADOS**

Visando avaliar o desempenho dos algoritmos multigrid, tomaram-se exemplos elásticos planos e espaciais apresentados a seguir. Primeiramente, consideram-se os casos planos de uma placa com furo e um problema de fratura discutidos em<sup>3</sup>, empregando malhas com triângulos lineares. Posteriormente, analisa-se uma viga em balanço, discretizada com tetraedros lineares.

Procurou-se realizar uma comparação entre os métodos multigrid, direto, e iterativos baseados em gradiente conjugado (GC), incluindo algumas versões com pré-condicionadores apresentados em<sup>3</sup>: diagonal (GCD), com matrizes de partição SSOR (GCGS) e Gauss-Seidel simétrico (GCGS). Todos os algoritmos foram implementados em C<sup>++</sup><sup>7</sup>, considerando armazenamento em matriz esparsa<sup>2</sup>, e incorporados a uma base de programas disponível<sup>6</sup>. Observa-se que para todos os métodos, indica-se o número equivalente de iterações (NIT) de malha fina.

Para o método GCGS e algoritmos multigrid, utilizou-se o critério de convergência  $\|\mathbf{r}^{(n)}\|_2 = \frac{\|\mathbf{A}\mathbf{u}_{ii}^{(n)} - \mathbf{b}\|_2}{\|\mathbf{b}\|_2} < \xi$  com precisão  $\xi = 10^{-4}$ . Nos esquemas multigrid, as relaxações foram efetuadas pelo método de Gauss-Seidel.

Para comparar as soluções obtidas pelos métodos de Gauss ( $\mathbf{u}_{dir}$ ), iterativos ( $\mathbf{u}_{ite}$ ) e multigrid ( $\mathbf{u}_{mg}$ ), consideraram-se os seguintes erros relativos

$$\|e_r^{dir/ite}\|_2 = \frac{\|\mathbf{u}_{dir} - \mathbf{u}_{ite}\|_2}{\|\mathbf{u}_{dir}\|_2} \quad \|e_r^{dir/mg}\|_2 = \frac{\|\mathbf{u}_{dir} - \mathbf{u}_{mg}\|_2}{\|\mathbf{u}_{dir}\|_2} \quad \|e_r^{ite/mg}\|_2 = \frac{\|\mathbf{u}_{ite} - \mathbf{u}_{mg}\|_2}{\|\mathbf{u}_{ite}\|_2} \quad (18)$$

### Placa com furo e problema de fratura

Nesta seção, analisam-se os exemplos de placa com furo e de fratura analisados em<sup>3</sup>. Empregaram-se elementos lineares nas 4 malhas consideradas, estando as mesmas ilustradas nas Figuras 7 e 8, respectivamente, para a placa com furo e fratura. Da mesma maneira, os atributos das malhas estão dados na Tabela I.

Placa com furo							
Malha	Nós	Elementos	Equações	$NElems_{esp}^{dir}$	$m_{esp}^{dir}$	$NElems_{esp}^{ite}$	$m_{esp}^{ite}$
1	38	57	64	593	9,3	381	6,0
2	121	206	220	3504	15,9	1471	6,7
3	445	817	846	21620	25,6	5985	7,1
4	1679	3215	3272	126794	38,8	23833	7,3

Fratura							
Malha	Nós	Elementos	Equações	$NElems_{esp}^{dir}$	$m_{esp}^{dir}$	$NElems_{esp}^{ite}$	$m_{esp}^{ite}$
1	51	80	89	911	10,2	555	6,2
2	170	299	317	5607	17,7	2171	6,8
3	626	1171	1207	32422	26,9	8647	7,2
4	2383	4608	4679	204249	43,7	34312	7,3

Tabela I. Atributos das malhas para os problemas de placa com furo e fratura

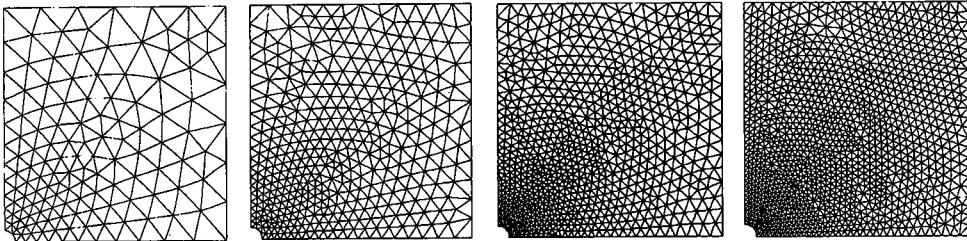


Figura 7. Malhas com elementos lineares a placa com furo

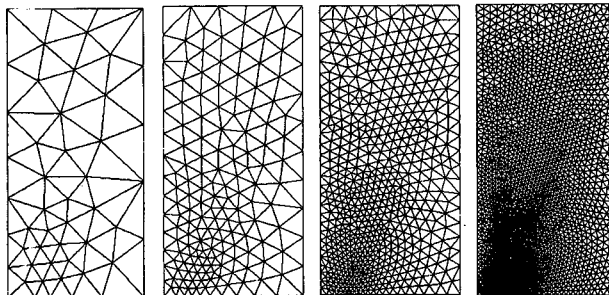


Figura 8. Malhas com elementos lineares para o problema de fratura

As Tabelas II e III apresentam os resultados da aplicação dos métodos direto de Gauss, iterativo GCGS e multigrid a estes dois problemas. Os parâmetros  $NElems_{esp}^i$  e  $m_{esp}^i$  ( $i = ite, dir$ ) representam, respectivamente, o número total de coeficientes e o número médio de elementos por linha na matriz esparsa, para os casos de métodos direto ( $i = dir$ ) e iterativo/multigrid ( $i = ite$ ).

Método	NIT	Ciclos $\nu_0, \nu_1, \nu_2$	$\ e_r^{dir/ite}\ _2$	$\ e_r^{dir/mg}\ _2$	$\ e_r^{ite/mg}\ _2$
Gauss	99	-	-	-	-
GCGS	208	-	$8,72 \times 10^{-6}$	-	-
FMV	34	4,1,1,1	-	$4,71 \times 10^{-6}$	$1,01 \times 10^{-5}$
	32	3,1,2,1	-	$1,12 \times 10^{-5}$	$1,47 \times 10^{-5}$
	42	3,2,1,1	-	$1,06 \times 10^{-6}$	$8,85 \times 10^{-6}$
	35	2,2,2,1	-	$1,09 \times 10^{-5}$	$1,44 \times 10^{-5}$
FMVV	38	8,1,1,1	-	$2,05 \times 10^{-5}$	$2,28 \times 10^{-5}$
	37	6,1,2,1	-	$2,82 \times 10^{-5}$	$3,01 \times 10^{-5}$
	35	7,2,1,1	-	$2,55 \times 10^{-5}$	$2,74 \times 10^{-5}$
	39	6,2,2,1	-	$1,74 \times 10^{-5}$	$2,00 \times 10^{-5}$
FMW	41	4,1,1,1	-	$2,52 \times 10^{-7}$	$8,72 \times 10^{-6}$
	39	3,1,2,1	-	$3,97 \times 10^{-7}$	$8,73 \times 10^{-6}$
	34	2,2,1,1	-	$1,12 \times 10^{-6}$	$8,80 \times 10^{-6}$
	44	2,2,2,1	-	$2,19 \times 10^{-8}$	$8,73 \times 10^{-6}$

Tabela II. Resultados para a placa com furo

Método	NIT	Ciclos $\nu_0, \nu_1, \nu_2$	$\ e_r^{dir/ite}\ _2$	$\ e_r^{dir/mg}\ _2$	$\ e_r^{ite/mg}\ _2$
Gauss	130	-	-	-	-
GCGS	273	-	$1,37 \times 10^{-5}$	-	-
FMV	43	5,1,1,1	-	$3,09 \times 10^{-6}$	$1,51 \times 10^{-5}$
	39	4,1,2,1	-	$8,76 \times 10^{-6}$	$1,86 \times 10^{-5}$
	42	3,2,1,1	-	$4,45 \times 10^{-6}$	$1,58 \times 10^{-5}$
	47	3,2,2,1	-	$2,78 \times 10^{-6}$	$1,49 \times 10^{-5}$
FMVV	43	9,1,1,1	-	$2,56 \times 10^{-5}$	$3,30 \times 10^{-5}$
	44	9,1,2,1	-	$2,08 \times 10^{-5}$	$2,86 \times 10^{-5}$
	44	9,2,1,1	-	$1,40 \times 10^{-5}$	$2,28 \times 10^{-5}$
	41	8,2,2,1	-	$2,42 \times 10^{-5}$	$3,17 \times 10^{-5}$
FMW	41	4,1,1,1	-	$1,77 \times 10^{-6}$	$1,44 \times 10^{-5}$
	38	3,1,2,1	-	$3,43 \times 10^{-6}$	$1,52 \times 10^{-5}$
	54	3,2,1,1	-	$2,22 \times 10^{-7}$	$1,38 \times 10^{-5}$
	43	2,2,2,1	-	$1,69 \times 10^{-6}$	$1,44 \times 10^{-5}$

Tabela III. Resultados para o problema de fratura

Os resultados em termos do número de operações em MFlop e espaço de memória em Kbytes, para escala logarítmica, estão ilustrados na Figuras 9 e 10, indicando-se entre colchetes os coeficientes angulares das retas ajustadas. Não se utilizou nenhum estimador de erro para refinar as malhas, apenas reduzindo-se pela metade o tamanho dos elementos.

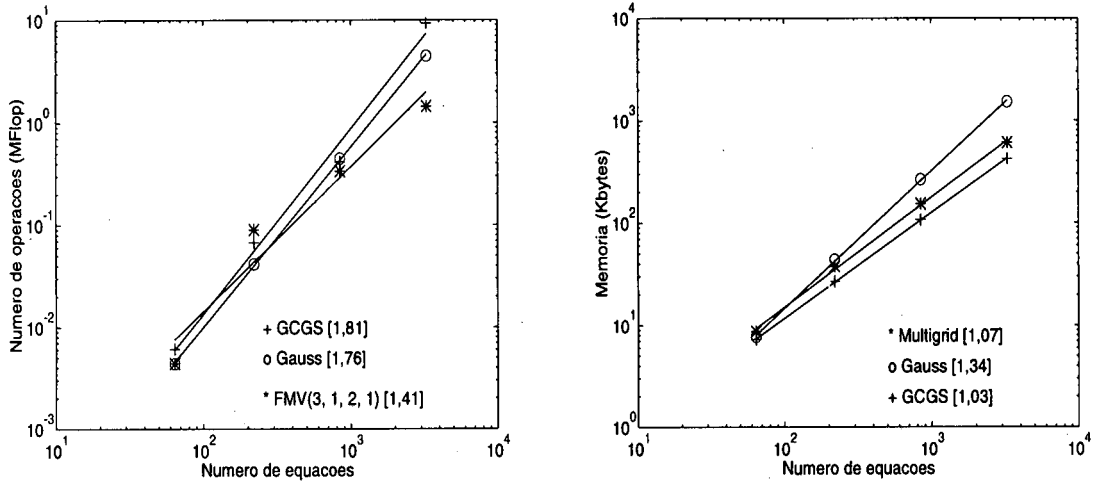


Figura 9. Número de operações e espaço de memória para os métodos de Gauss, GCGS e Multigrid para a placa com furo

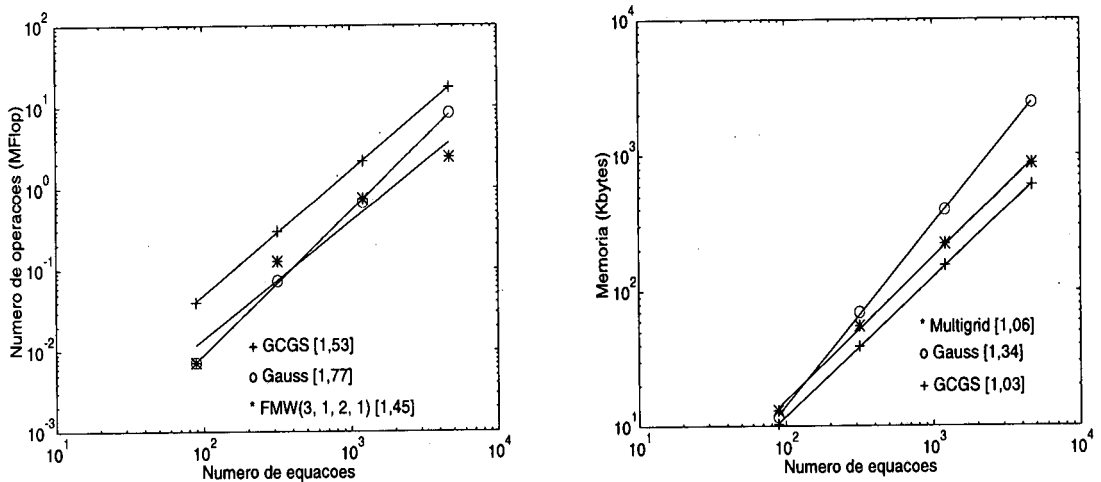


Figura 10. Número de operações e espaço de memória para os métodos de Gauss, GCGS e Multigrid para o problema de fratura



### Viga em balanço

A Figura 11 ilustra uma viga em balanço com suas dimensões e o carregamento aplicado, além das quatro malhas geradas com tetraedros lineares, estando as características das mesmas indicadas na Tabela IV.

Malha	Nós	Elementos	Equações	$m_{esp}^{dir}$	$NElems_{esp}^{dir}$	$m_{esp}^{ite}$	$NElems_{esp}^{ite}$
1	25	921	963	40,0	38511	16,0	15444
2	1130	3922	3279	81,2	266226	17,6	57804
3	3024	11813	8724	164,2	1432572	18,6	162276
4	8633	37976	24606	361,9	8904168	19,6	482364

Tabela IV. Atributos das malhas de viga em balanço

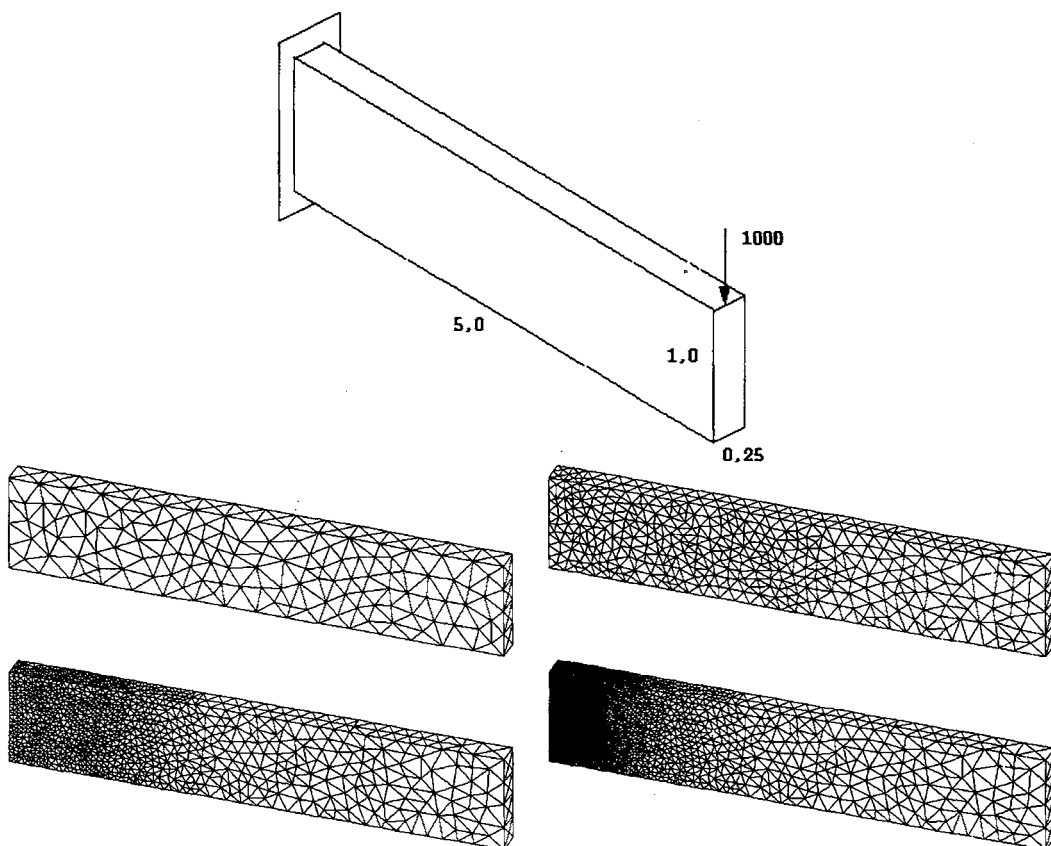


Figura 11. Malhas geradas para a viga em balanço ( $E = 1,0 \times 10^5$ ;  $\nu = 0,3$ )

Para a solução deste problema, aplicaram-se os algoritmos iterativos GC, GCD, GCSS e GCGS, além do método de Gauss e estratégias multigrid. Na Tabela V, tem-se o número equivalente de iterações na malha fina para todos os algoritmos utilizados, sendo ainda indicados os erros relativos da expressão (18). As Figuras 12 e 13 contém

Método	NIT	Ciclos $\nu_0, \nu_1, \nu_2$	$\ e_r^{dir/ite}\ _2$	$\ e_r^{dir/mg}\ _2$	$\ e_r^{ite/mg}\ _2$
Gauss	4345	---	—	—	—
GC	2361	---	—	—	—
GCD	1279	—	—	—	—
GCSS	844	—	—	---	—
GCGS	672	—	$3,19 \times 10^{-8}$	---	—
FMV	89	9,1,1,1	—	$2,80 \times 10^{-6}$	$2,80 \times 10^{-6}$
	105	8,1,2,1	—	$3,77 \times 10^{-6}$	$3,77 \times 10^{-6}$
	98	6,2,1,1	—	$9,83 \times 10^{-7}$	$9,84 \times 10^{-7}$
	108	5,2,2,1	—	$3,40 \times 10^{-6}$	$3,40 \times 10^{-6}$
FMW	101	8,1,1,1	—	$5,74 \times 10^{-8}$	$6,60 \times 10^{-8}$
	100	6,1,2,1	—	$1,76 \times 10^{-7}$	$1,79 \times 10^{-7}$
	111	5,2,1,1	—	$3,59 \times 10^{-8}$	$4,82 \times 10^{-8}$
	116	4,2,2,1	—	$4,06 \times 10^{-8}$	$5,18 \times 10^{-8}$
FMVV	117	25,1,1,1	—	$1,28 \times 10^{-5}$	$1,28 \times 10^{-5}$
	140	23,1,2,1	—	$1,67 \times 10^{-5}$	$1,67 \times 10^{-5}$
	115	23,2,1,1	—	$1,17 \times 10^{-5}$	$1,17 \times 10^{-5}$
	137	21,2,2,1	—	$1,53 \times 10^{-5}$	$1,53 \times 10^{-5}$

Tabela V. Resultados para a viga em balanço

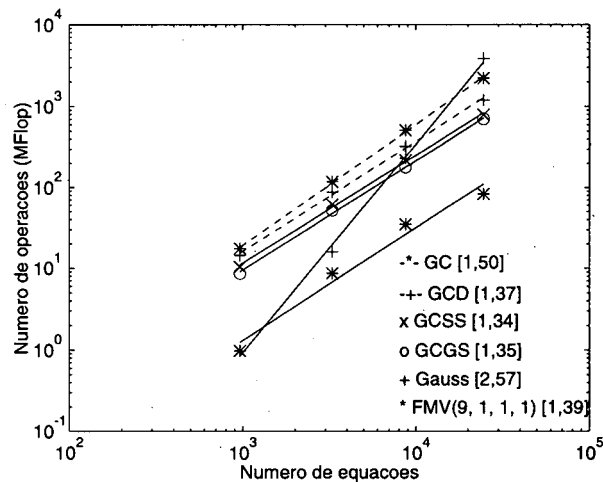


Figura 12. Número de operações (MFlop) para os métodos de Gauss, GC, GCD, GCSS, GCGS e FMV (9,1,1,1) na viga em balanço

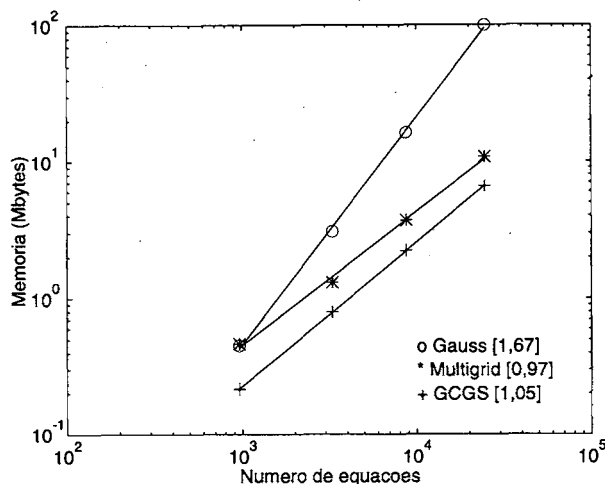


Figura 13. Espaço de memória (Mbytes) para os métodos de Gauss, iterativos e multigrid na viga em balanço

os gráficos em escala logarítmica com o comportamento em termos do número de operações e espaço de memória para os métodos empregados, estando indicados os coeficientes das retas ajustadas.

## DISCUSSÃO DOS RESULTADOS

A partir dos resultados obtidos para os problemas analisados, observa-se que:

- Nos casos da placa com furo e fratura, a partir das Tabelas II e III, verifica-se que os procedimentos FMV, FMVV e FMW possuem um comportamento semelhante entre si, sendo as performances bem superiores aos métodos de Gauss e GCGS, tomando-se o número equivalente de iterações de malha fina, assim como as Figuras 9 e 10. No entanto, no caso de FMVV tem-se um número maior de ciclos para se atingir um mesmo condicionamento numérico. Em relação ao número de pré-relaxações, tanto  $\nu_1 = 1$  como  $\nu_1 = 2$  mostraram-se razoáveis, observando-se que para este último caso foi possível reduzir, na maioria dos casos, o número total de ciclos. Esta mesma observação se aplica ao se utilizar  $\nu_0 = 2$ . No que se refere ao espaço de memória, as estratégias multigrid são, nestes casos, superiores ao método de Gauss, pois demandam menos memória como pode ser visto nas Figuras 9 e 10. Os coeficientes angulares das retas de número de operações para as estratégias multigrid são inferiores em relação àqueles dos procedimentos de Gauss e GCGS. No que se refere ao espaço de memória, os coeficientes de multigrid e GCGS são praticamente iguais, sendo inferiores ao método de Gauss e próximos de 1 nos exemplos de placa com furo e fratura.
- As estratégias multigrid baseadas em FMV e FMW são superiores em relação a FMVV, necessitando um menor número de ciclos para atingir a precisão especificada.

- Para problemas tridimensionais, torna-se interessante aplicar técnicas iterativas devido a menor demanda em termos de memória. A aceleração via procedimentos multigrid mostra-se bastante superior aos métodos de gradiente conjugado em relação ao número de operações. O espaço de memória acrescentado pelos métodos multigrid não é tão sensível, permitindo que todos os dados necessários sejam carregados na memória principal.

Logo, como conclusão verifica-se que sem dúvidas as estratégias multigrid mostram-se bastante superiores aos algoritmos iterativos baseados em gradiente conjugado, e conseqüentemente também em relação ao método direto de Gauss para a classe de problemas analisados.

- De forma geral, tanto para casos bi e tridimensionais, torna-se essencial ter ferramentas de geração de malhas, para o tratamento efetivo de problemas com multigrid não-aninhado.
- O comportamento médio dos métodos de Gauss, GCGS e multigrid, tomando os resultados obtidos nos exemplos apresentados e outros dados em<sup>2</sup>, estão dados na Figura 14a) e b), respectivamente, para casos bi e tridimensionais. O comportamento geral é semelhante ao já indicado. No entanto, para multigrid em problemas tridimensionais, obteve-se o número de operações variando linearmente com o número de equações.

Desta forma, mesmo tomando o comportamento médio, alcançou-se um custo de solução da ordem  $\mathcal{O}(N)$ , para um número de equações inferior a 100.000, em malhas não-aninhadas, comprovando assim os resultados teóricos previstos para os métodos multigrid.

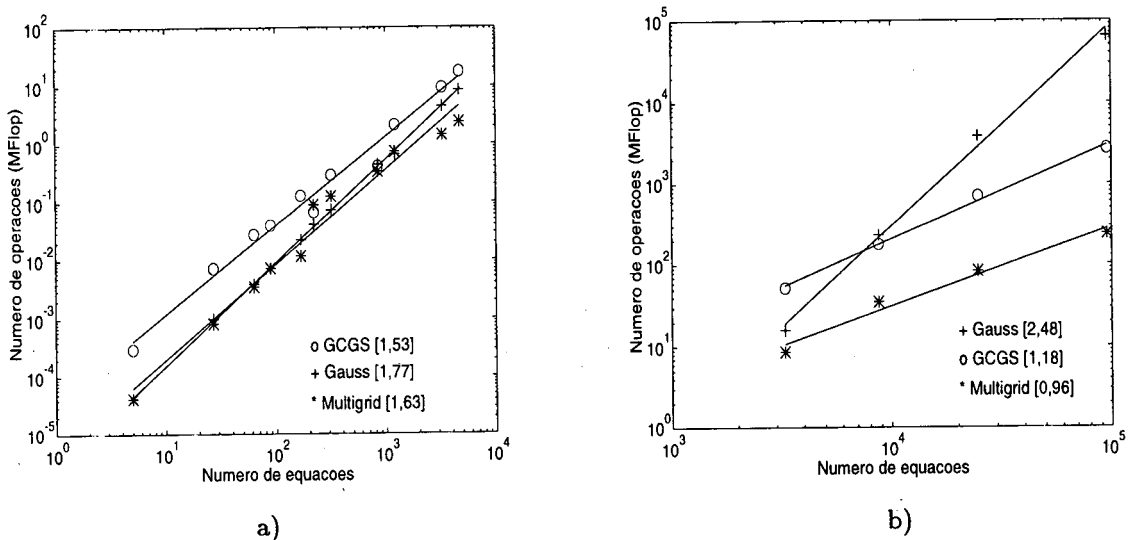


Figura 14. Número de operações (MFlop) para os métodos de Gauss, GCGS e Multigrid: a) exemplos bidimensionais; c) exemplos tridimensionais

- Observa-se que a sequência de malhas para os métodos multigrid pode ser obtida aplicando-se estimadores de erro, definindo-se estratégias multigrid adaptáveis como apresentado em<sup>2</sup>.

## AGRADECIMENTOS

Os autores agradecem aos seguintes órgãos pelo apoio ao desenvolvimento do trabalho: CNPq (Proc. 523.982/94-1), CNPq-RHAE (Proc. 610.035/94-0), FAPESP (91/2545-3), LNCC, UNICAMP, CENAPAD. Os autores também agradecem as facilidades de software gentilmente fornecidas pelo Grupo TACSOM ([www.lncc.br/~tacsom](http://www.lncc.br/~tacsom)).

## REFERÊNCIAS

1. O. Axelsson y V.A. Barker, "*Finite Element Solution of Boundary Value Problems-Theory and Computation*", Academic Press, (1984).
2. M.L. Bittencourt, "Métodos iterativos e multigrid adaptáveis em malhas não-estruturadas", Tese de doutorado, DPM/FEM/UNICAMP, (1996).
3. M.L. Bittencourt y R.A. Feijóo, "Análise comparativa entre métodos diretos e iterativos para a solução de sistemas de equações ", *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, Vol. **13**, 2, pp. 123-148, 1997.
4. A. Brandt, "Multilevel Adaptive Solutions to Boundary-Value Problems ", *Mathematics of Computation*, Vol. **31**, 138, pp. 333-390, (1977).
5. W.L. Briggs, "*A Multigrid Tutorial*", Pennsylvania, SIAM, (1987).
6. A.C.S. Guimarães y R.A. Feijóo, "ASDP: Um ambiente computacional para desenvolvimento de programas", *Relatório de Pesquisa e Desenvolvimento*, LNCC, 027/89, Rio de Janeiro, Brasil.
7. S.B. Lippman, "*C<sup>++</sup> Primer*", Addison-Wesley, (1991).
8. S.F. Mc Cormick (Ed.), "*Multigrid Methods*", SIAM, (1987).