

## CONSTRUÇÃO DE UM PRÉ-CONDICIONADOR USANDO BASES HIERÁRQUICAS

Misael L. M. Santana e Philippe R. B. Devloo .  
Faculdade de Engenharia Civil  
UNICAMP - Universidade Estadual de Campinas  
CP 6021 CEP 13081-970 Campinas - SP - Brasil

### RESUMO

A resolução de problemas de grande escala necessita de técnicas especializadas. A ferramenta mais atual é o processamento em paralelo. Essa ferramenta é usada em conjunto com a técnica da subestruturação, que consiste em dividir o domínio do problema, gerando uma malha que é chamada de malha grossa. Usando a malha grossa monta-se um sistema de equações que é chamado de sistema reduzido. Este sistema é resolvido usando o método do gradiente condicionado pré-condicionado. Este trabalho propõe a construção de um pré-condicionador resultante da mudança de bases dos contornos das subestruturas para bases hierárquicas.

### ABSTRACT

The solution of very large scale problems requires specialized methods. The most recent tool is the parallel processing. This tool is used together with the substructuring technique, which consists in splitting the domain of the problem, generating a coarse mesh. The coarse mesh is assembled into a system of equations named the reduced system. This system is solved by using the pre-conditioned conjugate gradient. This work is devoted to the construction of a pre-conditioner resulting from the modification of the shape functions on the contours of the substructures into hierarchical basis functions.

### INTRODUÇÃO

A evolução acelerada do *hardware* e *software* nos anos noventa tem permitido explorar novos tipos de cálculos que antes eram impossíveis ou muito difíceis de se realizar. No âmbito do desenvolvimento de *software*, um dos maiores progressos foi a introdução da filosofia da programação orientada a objetos. Neste trabalho, esta nova filosofia de programação foi utilizada na implementação da subestruturação dentro da análise estrutural.

A subestruturação vem se mostrando o melhor caminho para a solução em paralelo de grandes sistemas de equações lineares e não lineares, que surgem quando problemas elípticos de elasticidade, dinâmica dos fluidos, e outros problemas de engenharia são discretizados por elementos finitos. Após a divisão do domínio em subdomínios (também chamados de super-elementos), estes são distribuídos entre os processadores. As variáveis internas de cada subdomínio são reduzidas, obtendo-se um novo sistema referente às variáveis do contorno de cada subestrutura. Este novo sistema é cheio, pois é resultante de operações de multiplicação e adição de matrizes. Para este novo sistema, o método de resolução mais conveniente é o método do gradiente conjugado pré-condicionado, pois este é altamente paralelizável e a sua convergência pode ser acelerada com a utilização de pré-condicionadores.

Este trabalho inicialmente apresenta uma revisão geral da técnica de subestruturação para depois propor um pré-condicionador construído usando funções de forma das fronteiras entre os subdomínios. Como primeiro passo muda-se as funções de forma das fronteiras tendo como resultado uma base hierárquica. Com esta nova base constrói-se o pré-condicionador que tem dois blocos. Um destes está referido as funções da interface, e outro às funções correspondentes aos nós internos de cada subdomínio. Os testes numéricos mostraram que este pré-condicionador diminui drasticamente o número de condicionamento da matriz tanto na aproximação tipo  $h$  e tipo  $p$ .

### SUBESTRUTURAÇÃO

A modelagem de um problema de grande escala pode ser organizada pela discretização da equação diferencial em elementos finitos dentro de subdomínios. A idéia central é dividir o problema para resolvê-lo, i. é., "dividir para conquistar". Em cada subdomínio tem-se nós internos e nós de interface (ou de fronteira). Entende-se por nós internos aqueles que só pertencem a um subdomínio, e nós de fronteira aqueles que pertencem a mais de um subdomínio.

Num problema elástico, cuja formulação variacional pode ser expressa por:

$$a_{\Omega}(u_h, v_h) = f(v_h) \quad \forall v \in V^h(\Omega)$$

onde:

$$a_{\Omega}(u, v) = \int_{\Omega} E^{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial v_k}{\partial x_l} dx$$

$$f(v) = \int_{\Omega} f_i v_i dx + \int_{\partial\Omega_1} S_i v_i d$$

$E^{ijkl}$  é o tensor das constantes elásticas de Hooke. Este tensor de quarta ordem satisfaz a condição de simetria  $E^{ijkl} = E^{jikl} = E^{ijlk} = E^{klij}$ . Os vetores dos deslocamentos no sistema cartesiano são  $u = (u_1, u_2)$ ,  $v = (v_1, v_2)$ .  $f_i$  é a força de volume, no caso, o peso próprio, e  $S_i$  é a força aplicada ao corpo na porção  $\delta\Omega_1$  do contorno  $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2 \cup \partial\Omega_3$  conforme mostra a Figura 1.

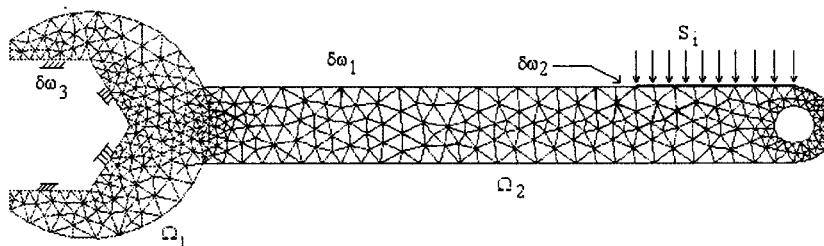


Figura 1 : Problema elástico com dois subdomínios.

Com a introdução das funções de base  $\{\varphi_i\}$  do espaço  $V^h$  é possível usar as funções de forma

$u_h = \sum x_i \varphi_i$  e  $v_h = \sum x_j \varphi_j$  para formar o seguinte sistema de equações:

$$a_{\Omega}(\varphi_i, \varphi_j) x_i = f(\varphi_j)$$

Uma vez que a integral sobre todo o domínio  $\Omega$  pode ser escrita como a soma de integrais sobre subdomínios, a matriz de rigidez pode ser calculada como a soma das integrais sobre os subdomínios. Sobre cada subdomínio  $s$  pode-se definir o seguinte problema:

$$a_{\Omega}^s(\varphi_i, \varphi_j) x_i^s = f^s(\varphi_j)$$

As funções de forma  $\{\varphi_i\}$  de cada subdomínio são divididas em dois conjuntos. Um conjunto é das funções de base  $\{\varphi_i\}$ ,  $i \in I$ , apoiadas no interior do subdomínio. Outro conjunto é das funções de base

$\{\varphi_i\}$ ,  $i \in \tilde{I}$ , que tem valores diferentes de zero em pelo menos numa das interfaces  $\partial\Omega_H$  do subdomínio. Na Figura 2 é mostrado um exemplo com dois subdomínios usando funções lineares de interpolação.

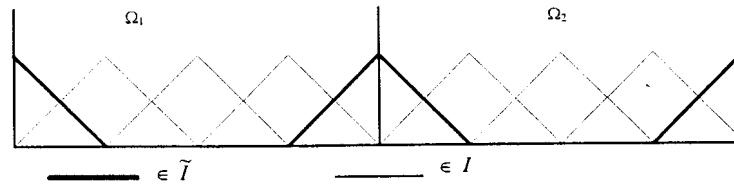


Figura 2 : Divisão das funções de forma em dois conjuntos.

Escrevendo-se o sistema de equações do subdomínio em forma de blocos, tem-se a seguinte expressão:

$$\begin{bmatrix} a_{II} & a_{I\tilde{I}} \\ a_{\tilde{I}I} & a_{\tilde{I}\tilde{I}} \end{bmatrix} \begin{bmatrix} x_I \\ x_{\tilde{I}} \end{bmatrix} = \begin{bmatrix} f_I \\ f_{\tilde{I}} \end{bmatrix}$$

Com a aplicação da condensação estática em cada sub-domínio define-se o seguinte sistema:

$$\begin{bmatrix} a_{II} & a_{I\tilde{I}} \\ 0 & s \end{bmatrix} \begin{bmatrix} x_I \\ x_{\tilde{I}} \end{bmatrix} = \begin{bmatrix} f_I \\ f_{\tilde{I}} - a_{\tilde{I}\tilde{I}}^{-1} a_{\tilde{I}I} f_I \end{bmatrix}$$

Onde a matriz  $s = a_{\tilde{I}\tilde{I}} - a_{\tilde{I}\tilde{I}}^{-1} a_{\tilde{I}I} a_{I\tilde{I}}$  é chamada de complemento de Shur.

Este problema pode ser resolvido em dois passos:

1. Cálculo dos graus de liberdade dos nós externos ao subdomínio:

$$[s] [x_{\tilde{I}}] = [f_{\tilde{I}} - a_{\tilde{I}\tilde{I}}^{-1} a_{\tilde{I}I} f_I]$$

1. Cálculo dos graus de liberdade dos nós internos ao subdomínio:

$$[a_{II}] [x_I] = [f_I - a_{I\tilde{I}} x_{\tilde{I}}]$$

Com cada sistema do passo 1, constrói-se um sistema de equações global que permitirá calcular os graus de liberdade de todas as interfaces  $\partial\Omega$ , do domínio inteiro, expresso por:

$$\tilde{K} x_I = \tilde{F}$$

onde:

$$\tilde{K} = \sum_i^{\Omega_i} s_i$$

$$\tilde{F} = \sum_i^{\Omega_i} [f_{\tilde{I}} - a_{\tilde{I}\tilde{I}}^{-1} a_{\tilde{I}I} f_I]_i$$

Este novo sistema é chamado de sistema reduzido ou sistema de interface, e pode ser resolvido por métodos iterativos pré-condicionados ou diretos. Justifica-se o uso de métodos iterativos, tais como o gradiente conjugado pré-condicionado (quando o sistema é simétrico) pelas seguintes razões:

- O sistema reduzido ainda pode ser grande,

- O sistema reduzido não é um sistema esparso e usar métodos diretos seria muito dispendioso em tempo e memória.
- No âmbito do processamento em paralelo, a eficiência da resolução por métodos diretos não necessariamente aumentará com o número de processadores.
- Ainda este sistema freqüentemente é mal condicionado,

Resumindo pode se dizer que a subestruturação usa inicialmente métodos diretos em cada subdomínio e métodos iterativos para resolver o sistema reduzido. Nota-se que a redução estática pode ser feita em paralelo em cada subdomínio e que o sistema reduzido pode ser resolvido com o gradiente conjugado pré-condicionado, que é altamente paralelizável.

### CONSTRUÇÃO DO PRÉ-CONDICIONADOR BLOCK-DIAGONAL

Assume-se que após o domínio  $\Omega$  ter sido dividido em elementos, as subestruturas  $\Omega_i$  são formadas por conjuntos de elementos tal que não exista acoplamento entre subestruturas ( $\Omega_i \cap \Omega_j = \emptyset$ ). Na malha bidimensional de subestruturas (também conhecida como malha grossa) reconhece-se que os nós podem pertencer aos seguintes três conjuntos:

1. Os vértices: São os nós que pertencem a dois ou mais subdomínios e pertencem ao contorno do domínio.

2. As interfaces entre subestruturas: Nós que pertencem a dois subdomínios e não pertencem ao contorno do domínio.

3. Nós Internos: Nós que pertencem a uma subestrutura apenas.

Conforme já foi mencionado, as variáveis correspondentes aos nós internos a cada subdomínio são primeiro eliminadas. O sistema resultante desta redução envolve apenas variáveis das interfaces. Este sistema é resolvido pelo método do gradiente conjugado.

Este trabalho usa esta sistemática, que também é comum a outros artigos já publicados [1], [2], [3], [4], [5] e [6]. Todas estas publicações se diferenciam entre si pelo tipo de pré-condicionador implementado.

Antes de montar a matriz  $\tilde{K}$ , os vértices  $V$  são renumerados e em seguida os nós de interface  $B$ . Seja a matriz de rigidez composta por quatro blocos, conforme mostrado abaixo:

$$\tilde{K} = \begin{bmatrix} \tilde{K}_{VV} & \tilde{K}_{VB} \\ \tilde{K}_{BV} & \tilde{K}_{BB} \end{bmatrix}$$

Uma das condições para que um pré-condicionador seja eficiente é que ele tem que ser parecido com a matriz pré-condicionada, tal que  $\tilde{K}M^{-1} = I$ , onde  $M$  é a matriz pré-condicionadora. Atendendo a essa condição, para a matriz  $M$  inicialmente foi construído o pré-condicionador sem levar em conta o acoplamento existente entre os subdomínios, ou seja, foram usados somente os blocos  $\tilde{K}_{VV}$  e  $\tilde{K}_{BB}$ , conforme mostrado abaixo:

$$M = \begin{bmatrix} \tilde{K}_{VV} & 0 \\ 0 & \tilde{K}_{BB} \end{bmatrix}$$

Porém, esta técnica não traz bons resultados, pois as funções de forma de cada vértice tem apenas uma influência "local". Esta influência é uma aproximação isolada dos vértices que não tem acoplamento com outros nós do domínio. A Figura 3 mostra as funções de forma dos vértices da malha grossa.

Do apresentado, conclui-se que caso as subestruturas têm poucos elementos, as funções de forma dos vértices aproximarão melhor o problema. No caso oposto, i. é., se dentro dos subdomínios existem muitos elementos, as funções de forma dos vértices não aproximarão bem o problema. Assim conclui-se que a eficiência deste pré-condicionador é inversamente proporcional ao número de elementos dentro de cada sub-domínio.

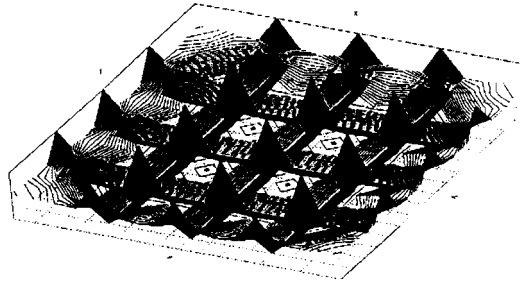


Figura 3: Funções de forma na malha dos subdomínios.

### MUDANÇA DE BASE NAS INTERFACES

Para contornar o problema da influência local dos vértices implementou-se a mudança de bases das funções de forma relacionadas aos vértices. Com essa idéia procurou-se conseguir que as funções de forma dos vértices tivessem uma influência ao longo de cada interface. Como resultado dessa ampliação da influência dos vértices sobre a interface, o espaço formado pelas novas funções dos vértices e as funções das interfaces resulta numa base hierárquica.

A mudança para bases hierárquicas tem suas origens na técnica "multigrid". Esta técnica tem como base a criação de subníveis de refinamento construindo-se sempre duas novas funções de forma lineares a partir de uma outra que esta num nível superior imediato [7] (ver Figura 4).

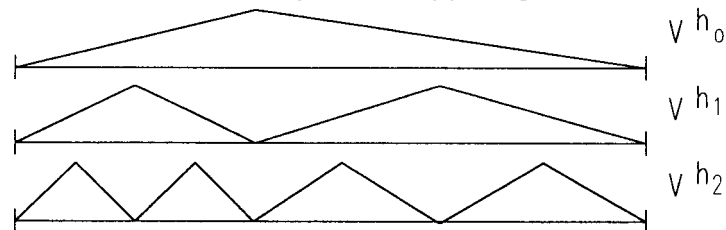


Figura 4 : Mudança de bases implementada por YSERENTANT [7].

Existem artigos onde se usa a mesma técnica mencionada acima, mas a mudança de base acontece apenas nas interfaces dos subdomínios e não no domínio inteiro (ver ref. [8]). Na referêná [9], MANDEL prova teoricamente que o pré-condicionador construído usando bases hierárquicas do tipo  $p$  também converge com taxas quase ótimas.

Seja  $V$  o espaço das funções de todo o domínio, constituído por três subespaços:

- O espaço  $V_{\text{harmonico}}$  das funções referentes aos nós internos de cada subdomínio;
- O espaço  $\tilde{V}_B$  das funções referentes às interfaces;
- O espaço  $\tilde{V}_V$  das funções referentes aos vértices.

O espaço de funções de forma pode ser expresso como:

$$V = V_{\text{harmonico}} \oplus \tilde{V}_B \oplus \tilde{V}_V$$

As funções usadas foram construídas usando-se polinômios de Chebichev e polinômios Lagrangeanos. Na Figura 5 são mostradas as funções de forma de ordem cúbica ao longo das interfaces.

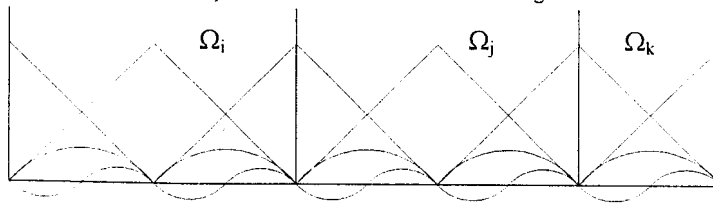


Figura 5 : Funções hierárquicas de ordem cúbica numa interface.

Para obter o valor da função  $\bar{\phi}_k$  sobre o nó  $k$  de tal maneira que tenha uma distribuição linear a longo da interface, usa-se a seguinte fórmula:

$$\bar{\phi}_k = \underbrace{\tilde{\phi}_k}_{f_k} \left( 1 - \frac{r_k}{L} \right)$$

Onde  $r_k$  é a distância entre o vértice atual  $v^j$  e o nó  $k$ . A variável  $L$  representa a distância entre os vértices  $v^j$ - $v^j$  da interface. A Figura 6 mostra esquematicamente como são definidas as distâncias  $r_k$  e  $L$ .

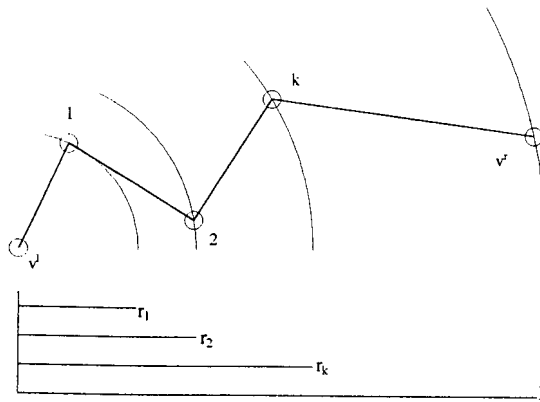


Figura 6 : Relação entre as distância entre os nós da interface e os vértices.

A função  $\tilde{\phi}_k$  é a função de forma linear sobre o nó  $k$ . A Figura 7 mostra como o valor de  $\bar{\phi}_k$  é distribuído linearmente ao longo da interface.

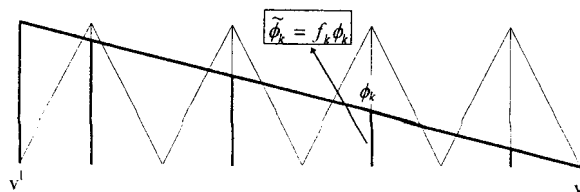


Figura 7 : O valor da função  $\bar{\phi}_k$  é proporcional a  $\tilde{\phi}_k$ .

Depois da mudança de bases o espaço das funções de forma resulta em:

$$V = V_{\text{harmonico}} \oplus \tilde{V}_B \oplus \tilde{V}_V$$

Na Figura 8 mostram-se as funções de forma da Figura 7 depois de distribuir linearmente as funções dos vértices.

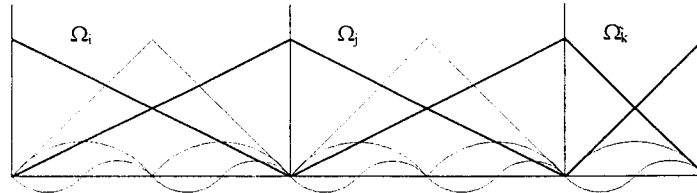


Figura 8 : Funções de forma de ordem cúbicas depois da mudança de bases.

### O PRÉ-CONDICIONADOR BLOCK-DIAGONAL COM BASE MODIFICADA

Após ter sido mudada a base dos vértices com a intenção de se estender a influência de suas funções, de maneira a permitir um acoplamento entre os vértices de subdomínios próximos, constrói-se o pré-condicionador. Este é construído sem levar em conta o acoplamento das funções  $\tilde{\phi}$  dos vértices com as funções  $\tilde{\phi}$  das interfaces. Conforme já foi mencionado anteriormente, os vértices são numerados inicialmente, e em seguida as interfaces. Sendo assim o pré-condicionador terá dois blocos na diagonal. O primeiro bloco refere-se aos vértices com a mudança de bases ( $\tilde{K}_{VV}$ ) e o segundo bloco as interfaces ( $\tilde{K}_{BB}$ ).

$$M = \begin{bmatrix} \tilde{K}_{VV} & 0 \\ 0 & \tilde{K}_{BB} \end{bmatrix}$$

Cabe ressaltar que o segundo bloco  $\tilde{K}_{BB}$  na verdade é um conjunto de blocos  $D_i$ , onde cada bloco representa a matriz de rigidez local de uma interface na malha dos subdomínios, ou melhor, o bloco  $\tilde{K}_{BB}$  reúne as matrizes de rigidez das interfaces como mostra a Figura 9.

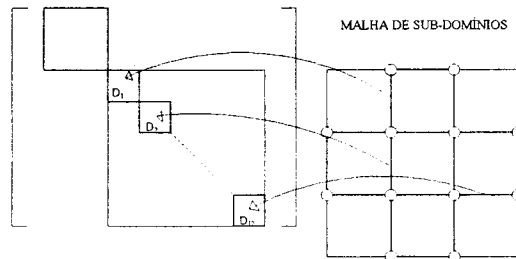


Figura 9 : Localização das influências das interfaces na matriz M.

ENIEF - X Congresso sobre Métodos Numéricos e suas Aplicações

na literatura,  $\kappa_2(A) = \sqrt{\lambda_1 / \lambda_n}$  é definido como sendo o número de condicionamento espectral da matriz A” ou simplesmente “número de condicionamento da matriz A”. Onde  $\lambda_1$  é o autovalor máximo e  $\lambda_n$  é autovalor mínimo da matriz de rigidez. Pode-se provar que as matrizes ditas “mal condicionadas” têm valores grandes de  $\kappa_2(A)$ , enquanto que as matrizes com valores pequenos de  $\kappa_2(A)$  são “bem condicionadas”. Existem ainda as matrizes ortogonais, que são “perfeitamente condicionadas”, para as quais  $\kappa_2(Q)=1$ . Sabe-se também que dizer uma matriz A é mal condicionada uma taxa de convergência lenta. Os problemas de convergência lenta e mal condicionamento têm que ser evitados. A maneira mais comum é construir um sistema equivalente onde o número de condicionamento seja mais favorável.

Para observar o comportamento do pré-condicionamento frente ao refinamento tipo  $h$  foram criadas várias malhas de superelementos. Cada superelemento, que inicialmente tinha 4 elementos, foi refinado para 9 elementos, em seguida para 16 elementos, e por último para 25 elementos. Todas estas aproximações usaram uma ordem de interpolação quadrática.

Para o estudo da adaptatividade tipo  $p$  o domínio foi discretizado em 9 subdomínios, sendo que cada subdomínio foi dividido em 4 elementos. O grau de interpolação inicial foi cúbico, incrementando-se um grau a cada refinamento até o sétimo grau.

A Tabela I mostra os autovalores máximos e mínimos das matrizes  $\tilde{K}$  e  $M^{-1}\tilde{K}$  obtidos para uma problema de Poisson com condições de Dirichlet em todo o contorno do problema.

O melhoramento do condicionamento da matriz também teve bons resultados para o problema elástico descrito a seguir. Foi utilizado o domínio retangular dos primeiros testes, tendo como condições de contorno dois lados paralelos engastados e os outros livres, com uma carga uniformemente distribuída sobre um dos lados livres. Na Tabela 2 podem ser observados os resultados dos cálculos dos autovalores máximos e mínimos para a matriz  $\tilde{K}$  e  $M^{-1}\tilde{K}$ . Igualmente ao teste anterior, estas matrizes correspondem às novas bases.

|   | p=2     |         |         |         | 4 ELEMENTOS POR SUBDOMÍNIO |          |          |          |
|---|---------|---------|---------|---------|----------------------------|----------|----------|----------|
|   | 4       | 9       | 16      | 25      | p=4                        | p=5      | p=6      | p=7      |
| $\lambda(K)_{\min}$                               | 0.0820  | 0.0849  | 0.0858  | 0.0860  | 0.0177                     | 0.0083   | 0.0054   | 0.0032   |
| $\lambda(K)_{\max}$                               | 4.0570  | 3.8308  | 3.6476  | 3.5213  | 2.8707                     | 2.8734   | 2.8737   | 2.8735   |
| $\lambda(K)_{\max}/\lambda(K)_{\min}$             | 49.5014 | 45.1440 | 42.5270 | 40.9538 | 162.4348                   | 346.0888 | 531.0021 | 899.3582 |
| $\lambda(M^{-1}K)_{\min}$                         | 0.2263  | 0.1854  | 0.1623  | 0.1471  | 0.1584                     | 0.1342   | 0.1181   | 0.1064   |
| $\lambda(M^{-1}K)_{\max}$                         | 1.6780  | 1.7172  | 1.7409  | 1.7572  | 1.7194                     | 1.7507   | 1.7726   | 1.7892   |
| $\lambda(M^{-1}K)_{\max}/\lambda(M^{-1}K)_{\min}$ | 7.4140  | 9.2603  | 10.7293 | 11.9454 | 10.8527                    | 13.0422  | 15.0148  | 16.8155  |

Tabela I Melhoramento do número de condicionamento para o problema de Poisson.

|   | p=2     |         |         |         | 4 ELEMENTOS POR SUBDOMÍNIO |          |          |           |
|---|---------|---------|---------|---------|----------------------------|----------|----------|-----------|
|   | 4       | 9       | 16      | 25      | p=4                        | p=5      | p=6      | p=7       |
| $\lambda(K)_{\min}$                               | 0.0550  | 0.0551  | 0.0546  | 0.0535  | 0.0110                     | 0.0052   | 0.0035   | 0.0020    |
| $\lambda(K)_{\max}$                               | 4.2059  | 3.9870  | 3.7871  | 3.6433  | 3.0492                     | 3.0465   | 3.0467   | 3.0465    |
| $\lambda(K)_{\max}/\lambda(K)_{\min}$             | 76.4845 | 72.3790 | 69.3420 | 68.0445 | 278.3374                   | 583.6257 | 861.4444 | 1501.7991 |
| $\lambda(M^{-1}K)_{\min}$                         | 0.1782  | 0.1471  | 0.1294  | 0.1178  | 0.1428                     | 0.1261   | 0.1061   | 0.0984    |
| $\lambda(M^{-1}K)_{\max}$                         | 2.0771  | 2.0887  | 2.0953  | 2.0995  | 2.0908                     | 2.0971   | 2.1037   | 2.1064    |
| $\lambda(M^{-1}K)_{\max}/\lambda(M^{-1}K)_{\min}$ | 11.6545 | 14.2027 | 16.1899 | 17.8248 | 14.6399                    | 16.6319  | 19.8197  | 21.3973   |

Tabela II Melhoramento do número de condicionamento para o problema de Poisson.

Os testes realizados indicam que o número de pré-condicionamento é bastante reduzido melhorando o



condicionamento da matriz de rigidez. Embora esta redução seja mais eficiente na adaptatividade tipo  $p$ , o efeito é o mesmo, pois a redução leva aos mesmos níveis de número de pré-condicionamento que os obtidos na adaptatividade tipo  $h$ . Observou-se que o pré-condicionador teve um bom desempenho para o problema elástico, à medida em que o número de condicionamento melhorou bastante. comportamento deste pré-condicionador tem uma tendência a ter o mesmo comportamento para as adaptatividades tipo  $p$  e tipo  $h$  para ordens de interpolação altas e malhas com alto grau de refinamento.

### AGRADECIMENTOS

À FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), pela concessão da bolsa de estudos e apoio financeiro à pesquisa;

À Faculdade de Engenharia Civil, pela estrutura oferecida;

### REFERÊNCIAS

- [1] AINSWORTH, M. A *hierarchical domain decomposition preconditioner for h-p finite element approximation on locally refined meshes*. SIAM J. Sci. Comp., v. 17, n. 6, p.1395-1413, Nov. 1996.
- [2] BJORSTAD, P. E., WIDLUND, O. B. *Iterative methods for the solution of elliptic problems on regions partitioned in substructures*. SIAM J. Numer Anal., v. 23, n. 6, p. 1097-1120, 1986.
- [3] BRAMBLE, J. H., PASIAK, J. E., SCHATZ, A. H. *An Iterative Method for Elliptic Problems on Regions Partitioned into Substructures*. Math. Comput., v. 46, n. 174, p. 361-369, Apr. 1986.
- [4] LE TALLEC, P., SALTEL, E., VIDRASCU, E. M. *Solving large scale structural problems on parallel computers using domain decomposition techniques*, em Advances in parallel and vector processing for structural mechanics. In: ADVANCES IN PARALLEL AND VECTOR PROCESSING FOR STRUTURAL MECHANICS, 1994. Civil-Comp Ltd, Edinburg, Scotland. p.127-132.
- [5] SMITH, B. F. *An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems*. SIAM J. Sci. Stat. Comput., v. 13, n. 1, p. 364-378, Jan. 1992.
- [6] SMITH, B. F. *A parallel implementation of an iterative substructuring algorithm for problems in three dimensions*. SIAM J. Sci. Comput., v. 14, n. 2, p. 406-423, Mar. 1992. Math., v. 49, p. 379-412, 1986.
- [7] YSERENTANT, H. *On the Multi-Level Splitting of Finite Element Spaces*. Numer.
- [8] SMITH, B. F., WIDLUND, O. B. *A Domain Decomposition Algorithm Using a Hierarchical Basis*. SIAM J. Sci. Stat. Comput., v. 11, p. 1212-1220, 1990.
- [9] MANDEL, J. *Two-level domain decomposition preconditioning for the p-version finite element method in tree dimensions*, International Journal for Numerical Methods in Engineering., v.29, p.1095-1108, 1990.

