Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXIX, págs. 3107-3126 (artículo completo)
Eduardo Dvorkin, Marcela Goldschmit, Mario Storti (Eds.)
Buenos Aires, Argentina, 15-18 Noviembre 2010

RESOLUÇÃO EM MÁQUINA PARALELA DE MODELO VOLUMÉTRICO DE EDIFÍCIOS

Francisco A. Menezes[a], Phillipe R. B. Devloo[b], Nathan Shauer[c], Thiago D. dos Santos[d]

Laboratório de Mecânica Computacional, Universidade Estadual de Campinas, Brasil, http://www.labmec.org.br

[a]fam@fec.unicamp.br, [b]phil@fec.unicamp.br, [c]nathan.sh@gmail.com, [d]santos@labmec.fec.unicamp.br

Keywords: Computação Paralela, Subestruturação, Elementos Finitos, Edifícios

Abstract. A análise de edifícios pelo método dos elementos finitos tridimensionais permite modelar todos os elementos estruturais (lajes, vigas, pilares) de forma única, porém esta abordagem conduz a um sistema de equações com muitos graus de liberdade. A proposta do trabalho é resolver este sistema de equações utilizando um método iterativo cujo precondicionador é baseado em subestruturação originalmente desenvolvido por Dohrmann. O edifício é particionado a partir de seus andares em subestruturas. A técnica possui uma eficiência paralela excelente pois cada andar pode ser calculado de forma independente. A convergência do método iterativo é beneficiada pelo reduzido número de graus de liberdade compartilhados entre andares.

1 INTRODUÇÃO

A abordagem normalmente utilizada para a análise estrutural de edifícios geralmente considera os elementos retilíneos, tais como vigas e pilares, como elementos unifilares, cuja rigidez é definida pelas propriedades do material e geometria das seções. As lajes, nestes casos, são tratadas como elementos de placas. O método dos elementos finitos geralmente é usado para a resolução do problema.

O presente trabalho busca tratar os elementos estruturais de uma forma diferente da usual, considerando-os como elementos finitos tridimensionais. A estrutura é dividida em elementos tridimensionais tetraédricos e a malha é gerada utilizando o programa GID. São usadas as formulações da elasticidade clássica. São considerados seis graus de liberdade por vértice de cada elemento finito.

O sistema linear resultante da abordagem de todo o edifício com os elementos finitos propostos possui dimensão considerável, o que gera um alto custo computacional quando da sua resolução.

Neste trabalho a técnica usada para a solução do problema baseia-se em dividir a malha em várias subestruturas, de forma a se obter sistemas de equações a serem resolvidos de forma iterativa e independente num equipamento com vários processadores atuando em paralelo.

Esta seqüência de cálculos proporciona redução significativa do tempo computacional na resolução do problema. Procura-se subdividir a malha a partir dos andares, de forma que cada andar componha uma subestrutura. Esta proposta alinha-se com o precondionador desenvolvido por Dohrmann (2003), o qual foi acoplado num método iterativo de resolução de sistemas.

O código computacional é desenvolvido sobre uma plataforma de programação orientada a objetos denominada PZ, desenvolvida por Devloo (2005) e colaboradores. A implementação permite a utilização de recursos de paralelismo, tais com *multithreading*, em diversas partes de cálculo. A plataforma PZ já possui uma bem sofisticada biblioteca de operações com matrizes, bem como várias de técnicas distintas de montagens da matriz de rigidez e algoritmos distintos de solução do sistema de equações.

No desenvolvimento do programa diversos testes foram realizados, de forma a analisar a eficiência dos cálculos quanto à quantidade de threads utilizados e número de subestruturas utilizadas.

Para ilustrar o trabalho, adota-se um edifício padrão, de oito andares, com quatro pilares nos cantos. Em cada andar há uma única laje, com quatro vigas de borda. As análises mostradas são para este edifício modelo.

Procura-se comparar a eficiência do cálculo em dois casos distintos:

a) Dividindo-se o prédio de oito andares em oito subestruturas, onde cada andar foi considerado uma subestrutura.
b) Dividindo-se o edifício de oito andares num número maior de subestruturas, para estudar a eficiência do método iterativo. No trabalho foram feitos estudos para até duzentas subestruturas.

2 O MODELO TRIDIMENSIONAL

A análise do edifício é feita considerando-se todos os elementos estruturais trabalhando de forma integrada. A análise baseia-se no método dos elementos finitos. São utilizados elementos finitos tridimensionais do tipo tetraédricos para descrever todos os elementos estruturais, incluindo lajes, pilares e vigas.

As formulações constitutivas do problema são as descritas pela elasticidade clássica, considerando seis graus de liberdade para cada nó do elemento. Utiliza-se um espaço de aproximação de grau 2, sem a utilização de refinamento direcional.

O edifício usado para as análises mostradas no trabalho é um edifício de oito andares, com quatro pilares de canto, uma laje por andar, ladeada por vigas de borda.

A Fig. 1 ilustra o edifício usado para a análise.

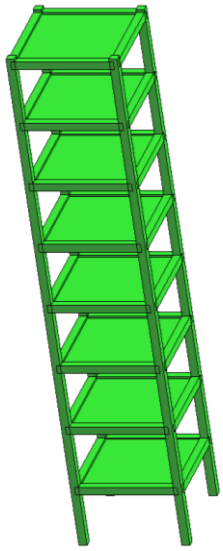

Figura 1 - Edifício Modelo com oito andares

Trata-se de um edifício de 8 andares, com pilares de seção transversal de 0,4x0,4m e altura 3,2m. As lajes têm dimensão 5x5m e espessura 0,15m. As vigas de borda possuem seção quadrada de 0,4x0,4m e comprimento 5m. O módulo de elasticidade considerado foi 10^6 kN/m^2. O coeficiente de Poisson adotado é de 0,3. Os pilares do térreo são engastados na base.

Considera-se como carregamento uma força distribuída por unidade de área, de direção horizontal, em uma das faces do edifício, de forma a simular o efeito do vento sobre a estrutura, com valor em módulo de 20 kN/m^2.

A malha geométrica utilizada para representar o edifício modelo de 8 andares é gerada com elementos tetraédricos com o auxílio do programa GID (2010). Ressalta-se que na formulação da malha busca-se uma uniformização quanto às dimensões dos elementos, procurando evitar a geração de elementos finitos muito diminutos ou com dimensão grande, próxima a dos elementos estruturais.

A Fig. 2 apresenta a malha geométrica para o edifício modelo.

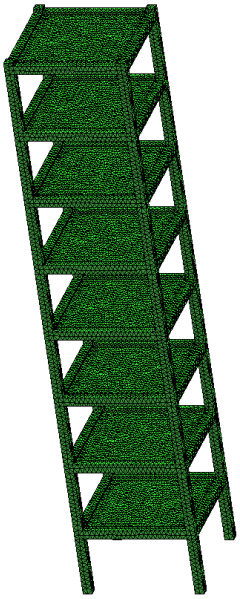

Figura 2 - Malha geométrica do edifício

A malha computacional associada à malha geométrica formulada, apresenta 78390 elementos, levando a análise do edifício todo a possuir um sistema de 437157 equações a 437157 incógnitas.

2.1 Subestruturação

O edifício modelo, num primeiro estudo, é dividido em oito subestruturas, conectadas a partir de seus andares, de modo a se obter subestruturas regulares e de dimensão próxima. Cada subestrutura de um andar contém 4 pilares, 4 vigas de borda e uma laje. A Fig. 3 apresenta o edifício modelo com as oito subestruturas.

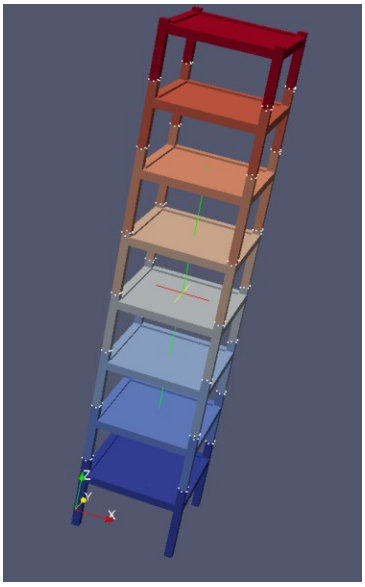

Figura 3 - Edificio modelo dividio em 8 subestruturas

Ressalte-se que a subestruturação é motivada pelo precondicionador utilizado para a resolução do sistema linear. A uniformização das subestruturas não é exigida pela técnica de precondicionamento, porém para a análise de desempenho quando do uso de paralelismo em algumas partes do código, a uniformização é essencial.

3 PRECONDICIONADOR

O método apresentado por Dohmann (2003) possui algumas similaridades com o trabalho de Farhat et al (1991), possuindo algumas diferenças, dentre as quais: 1) As variáveis principais para a solução iterativa são os deslocamentos ao invés dos multiplicadores de Lagrange; 2) a matriz de rigidez para o problema "*coarse*", isto é, de malha mais grosseira, nunca será indefinida; 3) extensões multi-níveis para problemas de alta escala parecem ser mais simples.

Dohrmann (2003) foca em particionar a malha de elementos finitos em subestruturas não sobrepostas. Num método direto de resolução, todos os graus de liberdade exclusivos de cada subestrutura são removidos por meio de condensação estática. O sistema resultante apresentará uma dimensão reduzida em relação ao problema integrado, sendo então solucionado convencionalmente por um método direto. Porém, para problemas com muitos graus de liberdade, a solução do sistema após condensação estática por método direto pode ser impraticável. Nesta contexto o método de Dohrmann (2003) utiliza a divisão em subestruturas como um precondicionamento para o método iterativo utilizado para a solução do sistema final.

Descreve-se nos próximos itens um rápido resumo do precondicionador de Dohrmann (2003).

3.1 Energia interna

A energia interna de uma subestrutura Ω_i é definida como:

$$E_i = \frac{\boldsymbol{u}_i^T \boldsymbol{K}_i \boldsymbol{u}_i}{\boldsymbol{2}} - \boldsymbol{u}_i^T \boldsymbol{F}_i \qquad (1)$$

sendo:

$\boldsymbol{K}_i$: matriz de rigidez da subestrutura *i*

$\boldsymbol{u}_i$: vetor solução da subestrutura i

O vetor $\boldsymbol{u}_i$ (graus de liberdade da subestrutura Ω_i) está relacionado ao vetor $\boldsymbol{u}$ global pela Eq. 2, em que, cada linha de $\boldsymbol{R}_i$ contém exatamente um único componente diferente de zero (tal componente é unitário).

$$\boldsymbol{u}_i = \boldsymbol{R}_i \boldsymbol{u} \qquad (2)$$

3.2 Sistema de Equações Global

As equações de elementos de equilíbrio do conjunto de elementos finitos da

estrutura integrada definem um sistema matricial dada pela Eq. (3), em que: $\boldsymbol{F}$ é o vetor de força global e $\boldsymbol{K}$ é a matriz de rigidez global da estrutura integrada, dada por:

$$\boldsymbol{K}\boldsymbol{u}=\boldsymbol{F} \tag{3}$$

sendo:

$$\boldsymbol{K}=\sum_{i=1}^{N}\boldsymbol{R}_i^T\boldsymbol{K}_i\boldsymbol{R}_i \quad \text{e} \quad \boldsymbol{F}=\sum_{i=1}^{N}\boldsymbol{R}_i^T\boldsymbol{F}_i \tag{4}$$

onde: N é igual ao número de subestruturas.

3.3 Matriz de restrição do problema de minimização

O precondicionamento em nível de subestrutura e em nível global está intimamente ligado à solução do problema de minimização de energia com restrições. Denota-se por ϕ_i^j a solução para o problema de minimização da energia interna E_i da subestrutura $\boldsymbol{i}$, sujeita à restrição:

$$\boldsymbol{C}_i\boldsymbol{u}_i = \boldsymbol{e}_j \tag{5}$$

onde:

$\boldsymbol{C}_i$: matriz de restrição

$\boldsymbol{e}_j$: j-ésima coluna da matriz identidade.

Cada linha da matriz $\boldsymbol{C}_i$ é associada com os graus de liberdade *coarse* comuns a duas ou mais subestruturas.

3.4 Sistema de Minimização de Energia

Define-se:

$$\Phi_i = \left[\phi_i^1 \quad \mathsf{L} \quad \phi_i^{n_{\mathsf{l}i}}\right] \tag{6}$$

onde:

n_{ℓ^i} : número de linhas em $\boldsymbol{C}_i$

Assim, do método de Lagrange para minimização restrita tem-se o seguinte sistema:

$$\begin{bmatrix}\boldsymbol{K}i & \boldsymbol{C}_i\\ \boldsymbol{C}_i & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{\varphi}\\ \boldsymbol{\Lambda}_i\end{bmatrix}=\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}\end{bmatrix} \tag{7}$$

onde:

$\boldsymbol{\Lambda}_i$: matriz dos multiplicadores de Lagrange

$\boldsymbol{I}$: matriz identidade

Este modelo de sistema será utilizado logo a seguir, no procedimento do

precondicionador.

3.5 Definição da matriz C_i

A matriz C_i representa os graus de liberdade "globais" do método iterativo. No trabalho de Dohrmann (2003) os nós da malha que definem os graus de liberdade globais são escolhidos baseados em critério geométrico. O autor sugere escolher três nós de interface distantes do ponto de vista geométrico.

Com o intuito de procurar uma definição de graus de liberdade globais independente da geometria, desenvolveu-se um procedimento para selecionar os graus de liberdade globais baseado na conectividade da malha.

Seja o conjunto de elementos ao qual o nó pertence El_{Ni}. Para nós internos da sub-malha este conjunto contém apenas o próprio elemento. Para nós de face o conjunto El_{Ni} contém dois elementos. Para nós de aresta (em malhas bi dimensionais) o conjunto de elementos contém um número não especificado de elementos.

Para uma dada subestrutura os conjuntos El_{Ni} permitem classificar os nós da mesma. Caso para dois nós *i* e *j* $El_N = El_N$ segue que eles pertencem a uma mesma face, aresta, etc.

Caso $El_N \subset El_N$ segue que o nó *i* pertence a uma entidade topológica (volume, face, aresta, etc.) cujo fechamento inclui a entidade topológica a qual o nó *j* pertence.

Os nós que pertencem a conjuntos que não são inclusos em nenhum outro conjunto formam os graus de liberdade "globais".

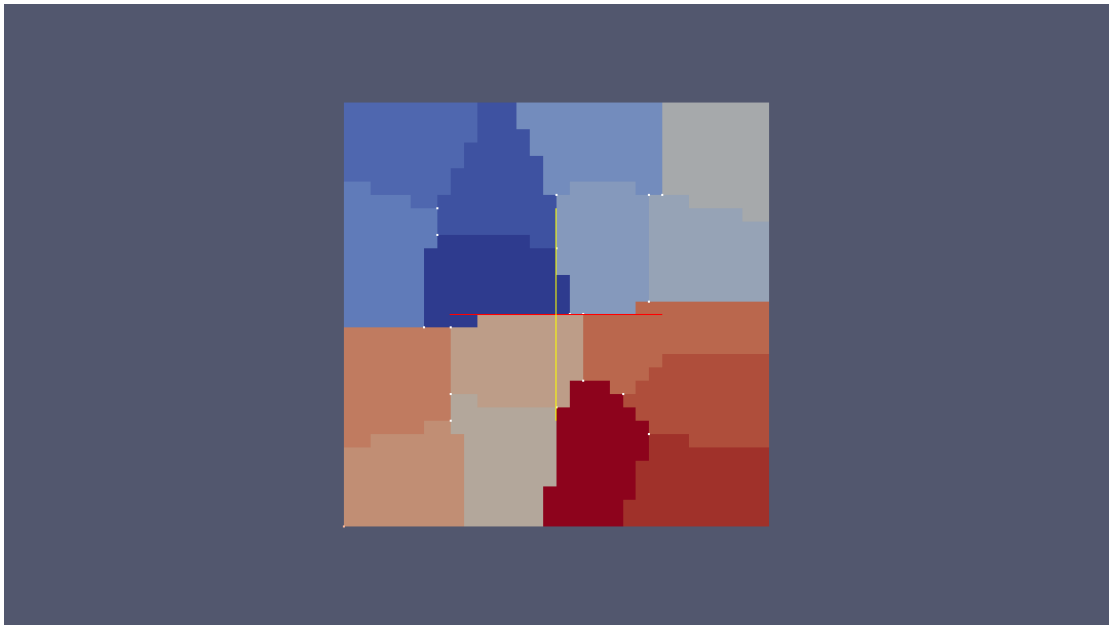

Figura 4 - Subestruturação de um quadrado com o algortimo de Dorhman

A Fig. 4 mostra a subestruturação de um quadrado onde os "pontos" brancos indicam os nós "globais" detectados pelo algoritmo. Note-se que a técnica descrita acima apenas identificou nós internos, pois os nós de borda não receberem

tratamento diferenciado.

Para incluir os nós de borda, foram inseridos elementos no contorno, associados a cada sub-elemento. Estes elementos adicionados ao contorno permitiram diferenciar nós internos, de nós de fronteira. A malha com os nós globais está mostrada na Fig. 5.

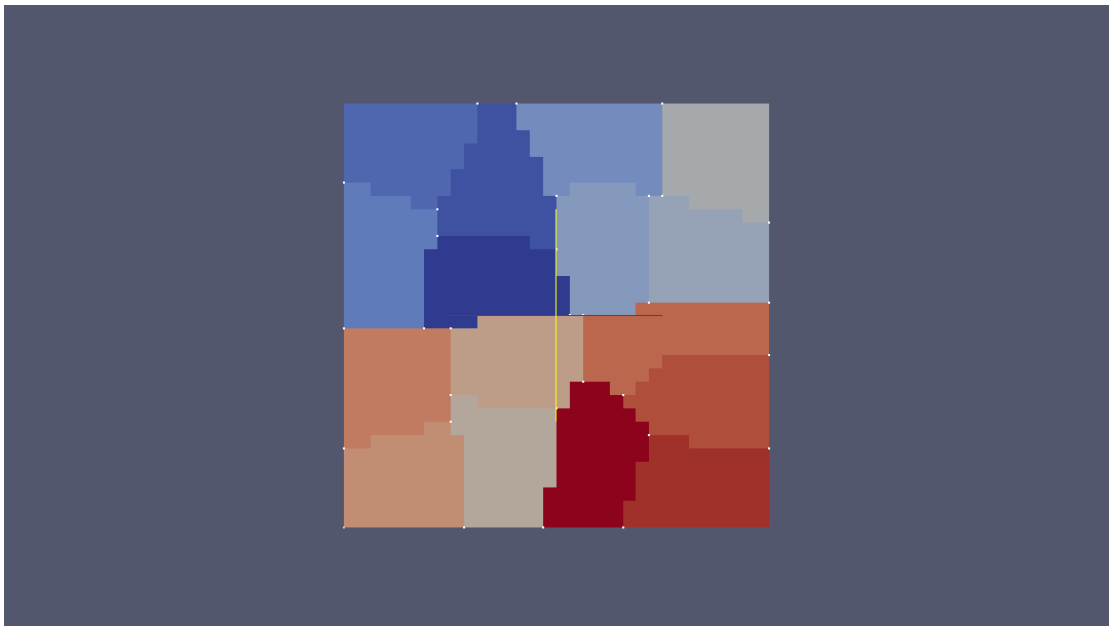

Figura 5 – Quadrado com nós globais da malha

Na subestruturação de modelos elásticos tridimensionais há ainda a restrição de que o número de nós globais precisa ser superior ou igual a três. Caso uma subestrutura tenha menos do que três nós globais, a matriz referenciada na Eq. 7 torna-se singular. Nestes casos acrescentam-se conjuntos El_{Ni} que estão inclusos em outros conjuntos até atingir o número mínimo de nós globais.

A Fig. 6 ilustra a divisão da estrutura tridimensional do edifício modelo em subestruturas junto com os nós globais. A configuração mostrada na Fig. 6a mostra a divisão em 8 subestruturas, onde a divisão das subestruturas é feita ao nível dos pilares, em cada andar. e a configuração direita ilustra uma divisão em 200 subestruturas.

Deve se notado que na subestruturação mostrada na Fig. 6a, a divisão das oito subestruturas se dá nos pilares, ao nível da divisão dos andares. Na divisão ilustrada na Fig. 6b a subestruturação é para 16 divisões. Na Fig. 6c, é ilustrada a divisão em 200 subestruturas não uniformes. Para as divisões mostradas nas figuras 6b e 6c foi utilizando o conjunto de rotinas METIS (1998).

O conjunto de cantos e bordas (*corners* e *edges*) para todas as subestruturas são agrupados em conjunto $\boldsymbol{M} = \{\boldsymbol{M_1}, \boldsymbol{K}, \boldsymbol{M_{Nn}}\}$. Denota-se u_{ik} o grau de liberdade na linha k de $\boldsymbol{u_i}$. Para a matriz $\boldsymbol{C_i}$, o componente na coluna k da linha para componente p de $\mathbf{M}_j$ é dado por

$$c_{ijkp} = \begin{cases} s_{ik} \text{ se } n(u_{ik}) \in M_j \text{ e } c(u_{ik}) = p \\ 0 \text{ casocontrário} \end{cases} \quad (8)$$

Sendo:

$n(u_{ik})$: número de nós de u_{ik}

$c(u_{ik})$: número de componentes de número de u_{ik}

p : inteiro entre 1 e 6.

s_{ik} : escalar que vale a soma dos componentes da diagonal da matriz de rigidez de $\boldsymbol{K}$ associados com $n(u_{ik})$

OBS: As linhas da $\boldsymbol{C_i}$ são então escalonados para que a soma dos componentes em cada linha seja unitária.

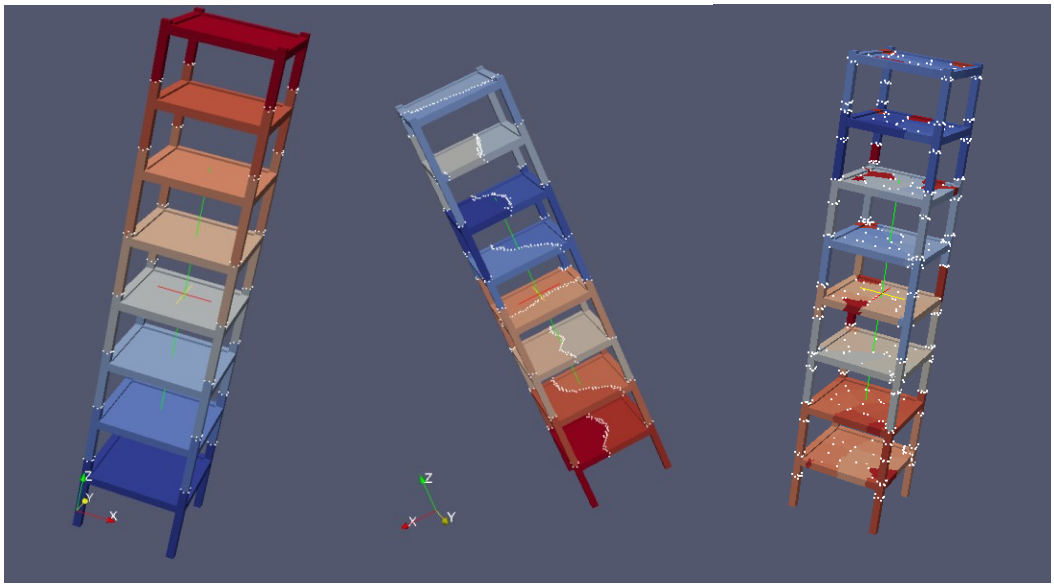

Figura 6a - Divisão em 8 subestruturas **Figura 6b - Divisão em 16 subestruturas** **Figura 6c - Divisão em 200 subestruturas**

3.6 Solução dos nós externos de $\boldsymbol{\Omega_i}$

Seja $\boldsymbol{u_{ci}}$ o vetor solução de dimensão n_{ci}, relacionado aos graus de liberdade *coarse* da subestrutura $\boldsymbol{\Omega_i}$. O vetor $\boldsymbol{u_{ci}}$ é relacionado ao vetor solução $\boldsymbol{u_c}$ relativo aos graus de liberdade *coarse* globais, por

$$\boldsymbol{u_{ci} = R_{ci} u_c} \quad (9)$$

onde:

cada linha de $\boldsymbol{R_{ci}}$ contém um único componente unitário diferente de zero

3.7 Solução dos nós internos de $\boldsymbol{\Omega_i}$

Denota-se por $\boldsymbol{u_{Ii}}$ o vetor solução referente a todos os graus de liberdade internos de $\boldsymbol{\Omega_i}$ (tais graus de liberdade não são compartilhados com nenhuma outra subestrutura), que pode ser calculado a partir do vetor $\boldsymbol{u_i}$ pela relação matricial:

$$\boldsymbol{u_{Ii} = R_{Ii} u_i} \quad (10)$$

em que, cada coluna de $\boldsymbol{R_{li}}$, contém exatamente um único componente unitário diferente de zero.

3.8 Função peso

Para distribuir o resíduo para as subestruturas $\boldsymbol{\Omega_i}$, é necessário definir a função peso para cada um dos graus de liberdade. Existem dois casos a serem considerados:

a) Se $\boldsymbol{u_{ik}}$ não está envolvido em quaisquer restrições, ou seja, a coluna k de $\boldsymbol{C_i}$ é nula, então o peso para $\boldsymbol{u_{ik}}$ é dado por

$$w_{ik} = \frac{S^i_{ik}}{S_{ik}} \tag{11}$$

onde:

S^i_{ik} : soma dos componentes da diagonal da matriz de rigidez $\boldsymbol{K_i}$ da subestrutura associada com $\boldsymbol{n(u_{i\,k})}$

b) Se houver um componente diferente de zero na coluna k da $\boldsymbol{C_i}$, então, define-se S^i_{ik} como a soma de todos os componentes da diagonal de $\boldsymbol{K_{ci}}$ associada com $\boldsymbol{n(u_{i\,k})}$

Define-se a matriz de rigidez *coarse* $\boldsymbol{K_{ci}}$ relacionada com a subestrutura $\boldsymbol{\Omega_i}$ como

$$\boldsymbol{K_{ci} = \Phi_i^T K_i \Phi_i} \tag{12}$$

Similarmente, define-se S^i_{ik} como a soma de todos os componentes da diagonal da matriz de rigidez global *coarse* $\boldsymbol{K_c}$. Neste caso, o peso para u_{ik} é dado por:

$$w_{ik} = \frac{S^{ci}_{ik}}{S^{c}_{ik}} \tag{13}$$

Os pesos w_{ik} são usados para formar a matriz peso $\boldsymbol{W_i}$ da subestrutura, a qual é definida por:

$$\boldsymbol{W_i} = diag(w_{i1}, \cdots, w_{in\,i}) \tag{14}$$

onde:

ni: número de linhas em $\boldsymbol{u_i}$

As matrizes peso da subestrutura são tais que:

$$\sum_{i=1}^{N} \boldsymbol{R_i^T W_i R_i = I} \tag{15}$$

3.9 Cálculo do precondicionador

Dado o vetor de resíduos global $\boldsymbol{r}$, associado com a solução iterativa da Eq. 3, o precondicionador residual $\boldsymbol{M^{-1}r}$ é obtido usando o seguinte processo iterativo:

a) Calcular a correção υ_1 da malha *coarse*:

$$\upsilon_1 = \sum_{i=1}^{N} \boldsymbol{R_i^T W_i \Phi_i R_{ci} K_c^{-1} r_c} \tag{16}$$

A matriz de rigidez da malha coarse é calculada da seguinte maneira:

$$\boldsymbol{K_c} = \sum_{i=1}^{N} \boldsymbol{R_\alpha^T K_\alpha R_\alpha} \tag{17}$$

O resíduo da malha *coarse* é calculado como:

$$\boldsymbol{r_c} = \sum_{i=1}^{N} \boldsymbol{R_\alpha^T \Phi_i^T W_i R_i r} \tag{18}$$

b) Calcular a correção υ_2 para a subestrutura:

$$\upsilon_2 = \sum_{i=1}^{N} \mathbf{R_i^T W_i z_i} \tag{19}$$

Na Eq.19 o vetor $\mathbf{z_i}$ é obtido pela solução do sistema de minimização, baseado no método de Lagrange, dado pela Eq. 20.

$$\begin{bmatrix} \boldsymbol{K_i} & \boldsymbol{C_i^T} \\ \boldsymbol{C_i} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{z_i} \\ \boldsymbol{\lambda_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W_i R_i r} \\ \boldsymbol{0} \end{bmatrix} \tag{20}$$

c) Calcular a correção υ_3 da condensação estática:

$$\upsilon_3 = \sum_{i=1}^{N} R_i^T R_{Ii}^T (R_{Ii} K_i R_{Ii}^T)^{-1} R_{Ii} R_i r_1 \tag{21}$$

Na Eq.21 o vetor $\boldsymbol{r_1}$ é dado por:

$$\boldsymbol{r_1 = r - K(u_1 + u_2)} \tag{22}$$

d) Calcular o resíduo precondicionado

$$\mathbf{M^{-1} r} = \upsilon_1 + \upsilon_2 + \upsilon_3 \tag{23}$$

Os resíduos associados com os graus de liberdade no interior da subestrutura são removidos previamente para a primeira iteração por meio de uma correção por condensação estática. Estes resíduos então permanecem nulos para todas as iterações subsequentes.

Ressalte-se que υ_1 é composta por contribuições sobre subdomínios que são descontínuos entre si. A continuidade é obtida por médias ponderadas entre valores de subdomínios vizinhos. É interessante notar, a partir da construção de $\boldsymbol{K_c}$, que dois graus de liberdade coarse são acoplados nesta matriz somente se ambos aparecem juntos em uma mesma subestrutura. Outro detalhe a ser observado é que a matriz $\boldsymbol{K_c}$ é sempre positiva definida.

4 OTIMIZAÇÃO DO MÉTODO ITERATIVO

O método iterativo publicado por Dohrmann (2003) requer a soma de três soluções: a solução "coarse" baseada nos nós globais, uma contribuição da inversão de cada sub estrutura e finalmente uma terceira correção para zerar o resíduo interno da cada sub estrutura. Na sua descrição original o método requer

a soma de graus de liberdade da estrutura global. Denota-se como $\boldsymbol{R}_{Ei}$ a matriz de seleção de graus de liberdade de interface.

Deste modo a solução u_{EI_i} representa uma permutação da solução original.

$$\boldsymbol{u}_{EI_i} = \begin{bmatrix} \boldsymbol{R}_{Ii} \\ \boldsymbol{R}_{Ei} \end{bmatrix} \boldsymbol{u}_i \tag{24}$$

A matriz $\boldsymbol{K}_{EI_i}$ é uma partição da matriz original ordenando primeiro as equações internas. Ela é dada por:

$$\boldsymbol{K}_{EI_i} = \begin{bmatrix} \boldsymbol{K}_{II} & \boldsymbol{K}_{IE} \\ \boldsymbol{K}_{EI} & \boldsymbol{K}_{EE} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{Ii} \\ \boldsymbol{R}_{Ei} \end{bmatrix} \boldsymbol{K}_i \begin{bmatrix} \boldsymbol{R}_{Ii} \\ \boldsymbol{R}_{Ei} \end{bmatrix}^T \tag{25}$$

A matriz $\overline{\boldsymbol{K}}_{EE_i}$ é a matriz condensada da subestrutura i. Ela é calculada com a Eq. 26.

$$\overline{\mathbf{K}}_{\mathbf{EE}_i} = \left(\mathbf{K}_{\mathbf{EE}} - \mathbf{K}_{\mathbf{EI}} \mathbf{K}^{-1} \mathbf{K}_{\mathbf{IE}}\right) \tag{26}$$

O vetor de carga condensado $\overline{\boldsymbol{F}}_{E_i}$ pode ser escrito utilizando a Eq. 27.

$$\overline{\mathbf{F}}_{\mathbf{E}_i} = \left(\mathbf{F}_{\mathbf{E}} - \mathbf{K}_{\mathbf{EI}} \mathbf{K}_{\mathbf{II}}^{-1} \mathbf{F}_{\mathbf{I}}\right) \tag{27}$$

O vetor resíduo $\bar{r}_{E_i}$ é dado por:

$$\bar{\mathbf{r}}_{\mathbf{E}_i} = \left(\overline{\mathbf{F}}_{\mathbf{E}} - \overline{\mathbf{K}}_{\mathbf{EE}} \mathbf{u}_{\mathbf{E}}\right) \tag{28}$$

Define-se a técnica de subestruturação como sendo um método iterativo aplicado à matriz global formada pela conexão das matrizes condensadas.

Seja $\overline{\boldsymbol{R}}_i$ a matriz que representa a seleção dos graus de liberdade externos da estrutura *i*, a partir da solução global. De fato, na implementação computacional a solução global é composta apenas dos graus de liberdade externos de cada subestrutura.

Define-se ainda $\overline{\boldsymbol{\Phi}}_i$ como a seleção de graus de interface de $\boldsymbol{\Phi}_i$. Ela pode ser calculada da seguinte maneira:

$$\overline{\boldsymbol{\Phi}}_i = \boldsymbol{R}_{Ei} \boldsymbol{\Phi}_{ii} \tag{29}$$

O vetor de correçao $\bar{\upsilon}_1$condensado pode ser calculado de maneira similar ao vetor υ_1 da técnica original, conforme Eq. (30).

$$\bar{\boldsymbol{u}}_1 = \sum_{i=1}^{N} \overline{\boldsymbol{R}}_i^T \overline{\boldsymbol{W}}_i \overline{\boldsymbol{\Phi}}_i \boldsymbol{R}_{ci} \boldsymbol{K}_c^{-1} \boldsymbol{r}_c \tag{30}$$

Similarmente o vetor $\bar{\upsilon}_2$ é a seleção dos graus de liberdade de interface do vetor $\boldsymbol{u}_2$ orginal.

$$\bar{\upsilon}_2 = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{N}} \overline{\mathbf{R}}_{\mathbf{i}}^{\mathbf{T}} \overline{\mathbf{W}}_{\mathbf{i}} \bar{\mathbf{z}}_{\mathbf{i}} \tag{31}$$

A técnica aplicada à matriz condensada é idêntica à técnica de subestruturação

original. As soluções as cada iteração são as mesmas. As vantagens da reformulação são :

a) O número de graus de liberdade utilizados no processo iterativo são apenas os graus de liberdade de interface. Isto representa uma economia muito grande de comunicação de dados.

b) O processo iterativo pode ser escrito apenas em termos dos vetores $\bar{\upsilon}_1$ e $\bar{\upsilon}_2$. Isto representa a economia de um ciclo de comunicação para cada iteração.

5 IMPLEMENTAÇÃO

A técnica de subestruturação de Dohrmann(2003) foi incorporada no ambiente orientado para objetos PZ para programação de elementos finitos. Seguindo a filosofia do ambiente, a funcionalidade algébrica do ambiente foi separada do cálculo de elementos finitos.

A classe que define e implementa as operação algébricas sobre as matrizes de uma subestrutura foi chamada de TPZDohrSubstructCondense. Esta classe é uma evolução de uma primeira implementação TPZDohrSubstruct que implementava a técnica de Dohrmann com três vetores $\boldsymbol{v_1}$, $\boldsymbol{v_2}$ e $\boldsymbol{v_3}$.

A classe que representa o conjunto de subestruturas como um objeto matricial é chamada de TPZDohrMatrix. Esta classe implementa o produto "matriz x vetor" que corresponde à superposição dos produtos "matriz condensada x vetor" de cada subestrutura. É importante notar que a matriz condensada nunca é calculada. Está implementado apenas o produto "matriz condensada x vetor" que é realizado por uma seqüência de operações matriciais, conforme documentado acima.

A classe que representa o precondicionamento é denominada TPZDohrPrecond. Esta classe é derivada da classe abstrata TPZMatrix e implementa no produto "matriz x vetor" a técnica de subestruturação. No caso o produto "matriz x vetor" consiste nos seguintes passos:

a) Cálculo do resíduo *coarse* e *assemblagem* do mesmo
b) Inversão do sistema *coarse* e cálculo de $\bar{\upsilon}_1$
c) Cálculo da contribuição de $\bar{\upsilon}_2$.
d) Cálculo da correção global

Uma classe que inicializa as estruturas de dados de TPZDohrSubstructCondense baseado numa discretização de elementos finitos e que faz a ponte entre a malhas de elementos finitos e as operações algébricas da técnica de subestruturação é chamada TPZDohrStructMatrix. Esta classe recebe como dado de entrada uma malha de elementos finitos e o número de subestruturas desejadas. Os passos para gerar a estrutura de dados são:

a) Decomposição da malha em subestruturas utilizando o programa Metis (1998).
b) Pós-processamento do particionamento feito com o programa Metis para eliminar subestruturas disjuntas e agrupamento de subestruturas com poucos elementos.
c) Determinação de nós globais (ou coarse) utilizando uma técnica baseada

em análise de conjuntos

d) Inicialização dos objetos TPZDohrSubstructCondense
e) Otimização de banda da matriz $\boldsymbol{K_c}$ e das matrizes $\boldsymbol{K_{lli}}$ e $\boldsymbol{K_i}$
f) Cálculo das matrizes $\boldsymbol{K_{lli}}$ e $\boldsymbol{K_i}$. Este processo foi otimizado, tal que cada matriz elementar foi calculada apenas uma vez e "*assemblada*" em duas matrizes diferentes.
g) Cálculo dos vetores $\boldsymbol{\Phi_i}$ e matrizes $\boldsymbol{K_{ci}}$ correspondentes

6 PARALELISMO

O índice de paralelismo da técnica de subestruturação é muito alto. A cada iteração as contribuições das subestruturas podem ser calculadas em paralelo. O cálculo das matrizes das subestruturas também pode ser feito em paralelo, enquanto para cada cálculo de matriz de subestrutura diversos "threads" podem calcular as matrizes elementares em paralelo.

O paradigma de execução em paralelo adotada no programa PZ é baseado nos conceitos da biblioteca JINI (http://jini.org) disponível para a linguagem de programação Java. Nela uma lista de objetos são criados, onde cada qual representa uma tarefa a ser executada. Por outro lado um vetor de "threads" é criado, cuja montagem corresponde a tirar um objeto do tipo "tarefa" da lista e executar a tarefa correspondente. Este paradigma tem como vantagem que o programa em paralelo funciona sem modificações em máquina com número de processadores variados.
Neste trabalho os processos paralelizados são:

1. Cálculo das matrizes das subestruturas e decomposição das mesmas
2. Produto "matriz x vetor" da estrutura global
3. Precondicionamento por subestrutura

Na seção sobre os resultados elencamos a eficiência paralela das diversas partes. De modo geral nota-se que o cálculo e a decomposição das matrizes globais não alcançou eficiência teórica por problemas de acesso a memória. As tarefas de produto e precondicionamento alcançarem eficiência próxima da ideal. Isso também foi constado por outros autores, como Elias (2010).

7 RESULTADOS

O precondicionamento de Dohrmann e sua paralelização foi aplicado à um modelo de sólido tridimensional de um prédio de 8 andares. Os elementos são tetraedros lineares e quadráticos. A malha de elementos finitos é gerada pelo software GiD (2010) e para a decomposição em subdomínios foi utilizada a biblioteca Metis 4.0 (1998).

O tamanho do sistema de equações para a discretização linear é de 71.373 equações enquanto para elementos quadráticos é 437.157.

O equipamento utilizado para os testes é um microcomputador "Macintosh Pro" com dois processadores quad-core Xeon Nehalem e hyperthreading, com 8 Mbytes de Cache tipo L3 e 12 GB de memória.

Atenção particular foi dada à decomposição do edifício em 8 subdomínios pois

neste caso as subestruturas são particionadas pelos pilares. Este particionamento diminui os nós entre as subestruturas e a matriz condensada fica menor.

Em outras avaliações da eficiência do método iterativo, aumentou-se o número de subestruturas até 200. Nestes casos o número de equações de interface é (muito) maior. O número de iterações para convergir o resíduo também é (muito) maior.

Nos testes, primeiro foi avaliada a performance com um só thread. Em seguida, o programa foi executado com 2, 4, 8 e 16 threads a fim de comparar os resultados. Em seguida, realizamos os mesmo testes para 16, 32, 60, 100 e 200 subestruturas. O gráfico abaixo representa o tempo total de execução em relação ao número de threads.

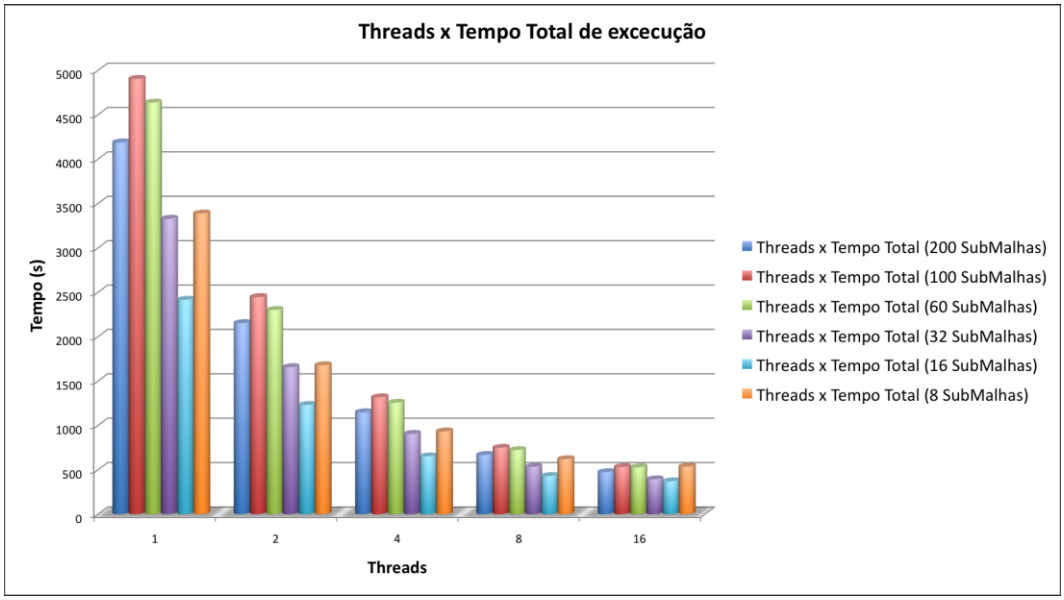

Figura 7 – Tempo Total de Execução x Threads

Para uma melhor visualização da relação do tempo total com o número de subestruturas, o gráfico para 16 threads é mostrado em detalhe na Fig. 8.

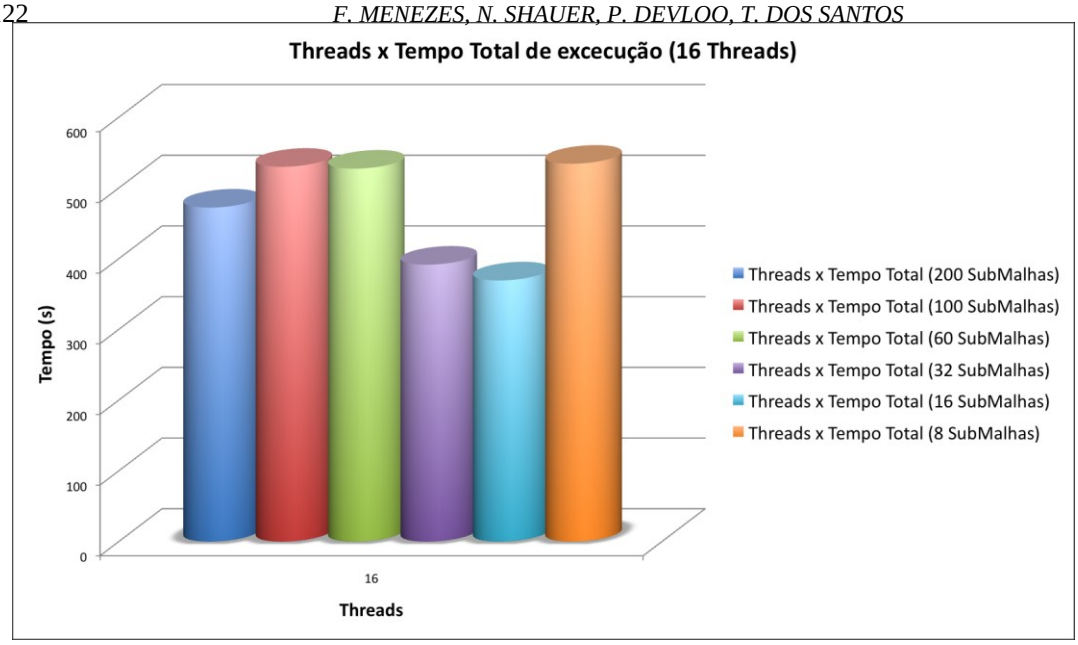

Figura 8 – Tempo Total de Execução x Threads (16 threads)

O gráfico da Fig. 9 ilustra o ganho de velocidade em relação ao número de threads para cada divisão de subestruturas.

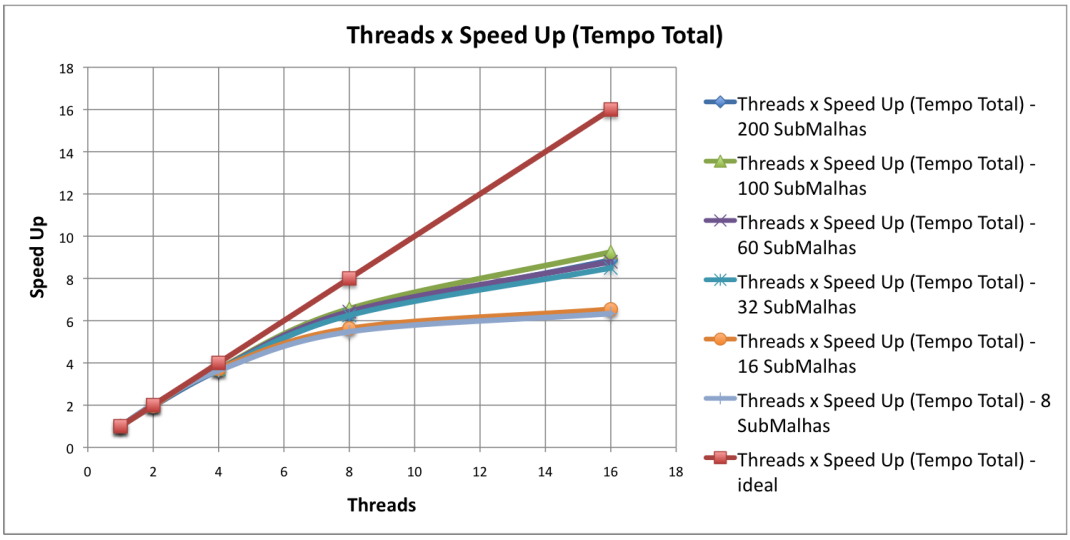

Figura 9 – Speed Up (Tempo Total) x Threads

O gráfico abaixo representa o tempo de decomposição em relação ao número de threads para cada divisão de subestruturas.

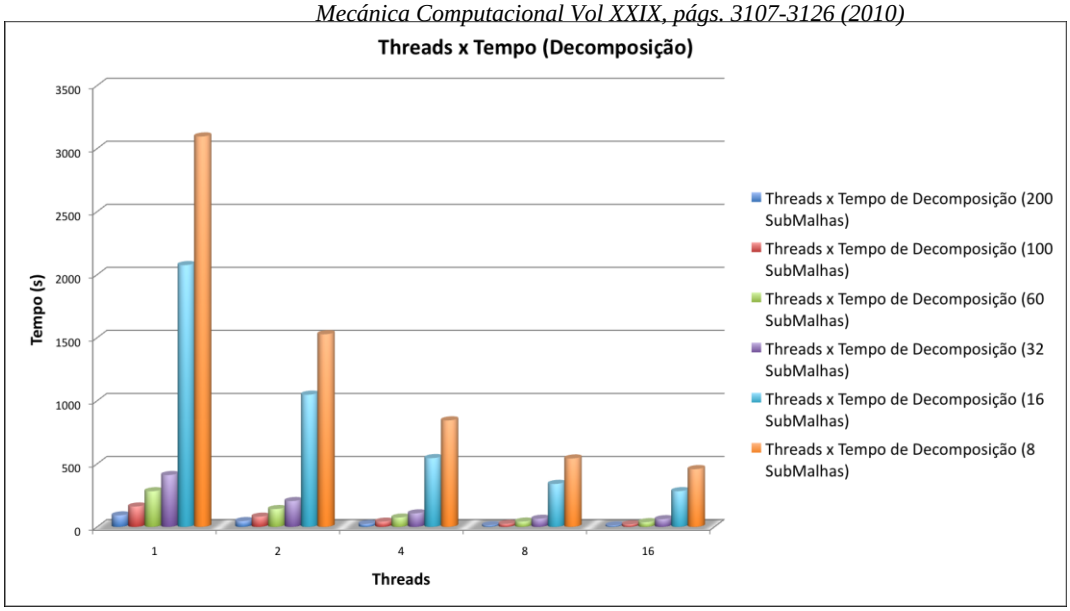

Figura 10 – Tempo de Decomposição x Threads

Em seguida foi gerado um gráfico para o ganho de velocidade (Speed Up) da decomposição dos sistemas de equações das subestruturas em relação ao número de threads para cada divisão de subestruturas.

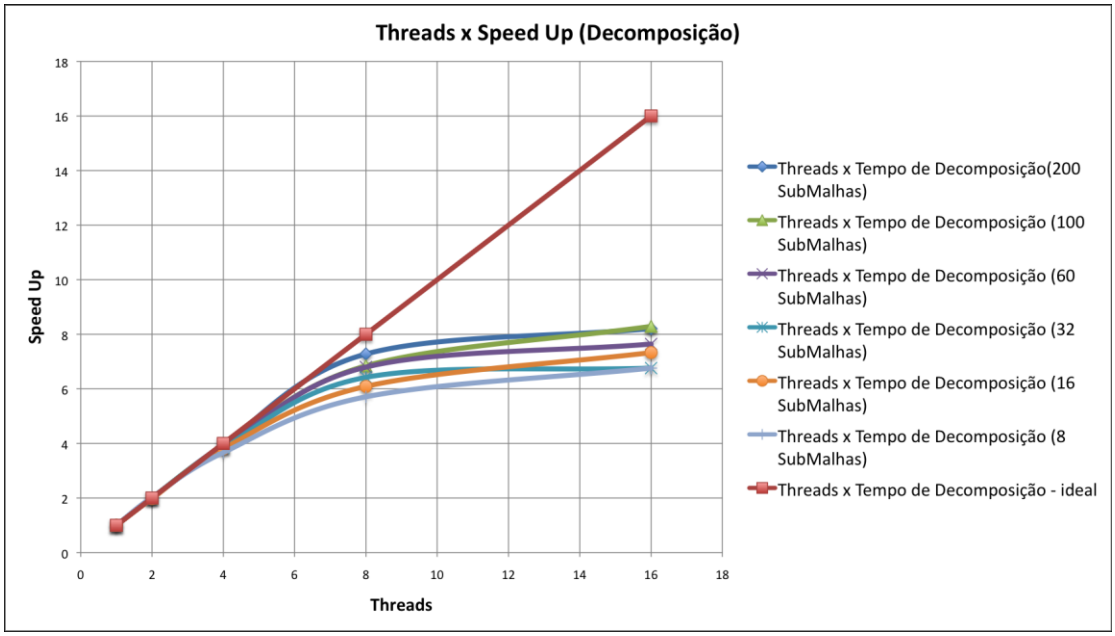

Figura 11 – Speed Up (Decomposição) x Threads

Verifica-se que a decomposição é mais rápida para um número maior de subestruturas, mesmo assim o tempo total de resolução é maior nesses casos. Isso se deve ao fato de o número de iterações para convergência ser maior. O gráfico da Fig. 12 ilustra o número de iterações do método gradiente conjugado para convergir em relação ao número de subestruturas.

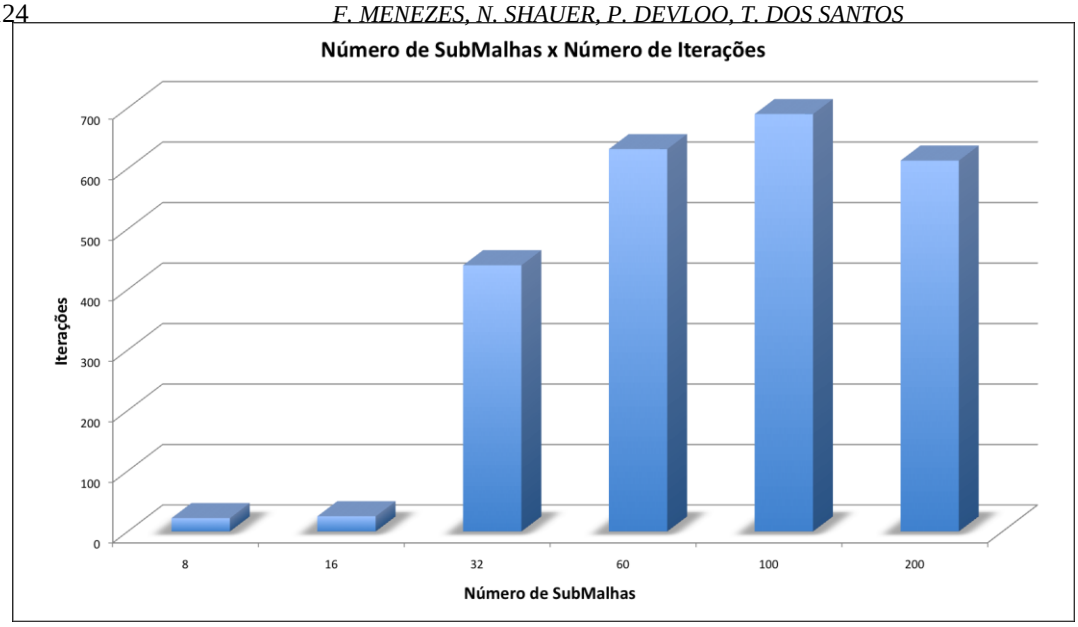

Figura 12 – Número de Iterações x Número de SubMalhas

Também é interessante verificar o tempo necessário para essas iterações em relação ao número de threads e SubMalhas.

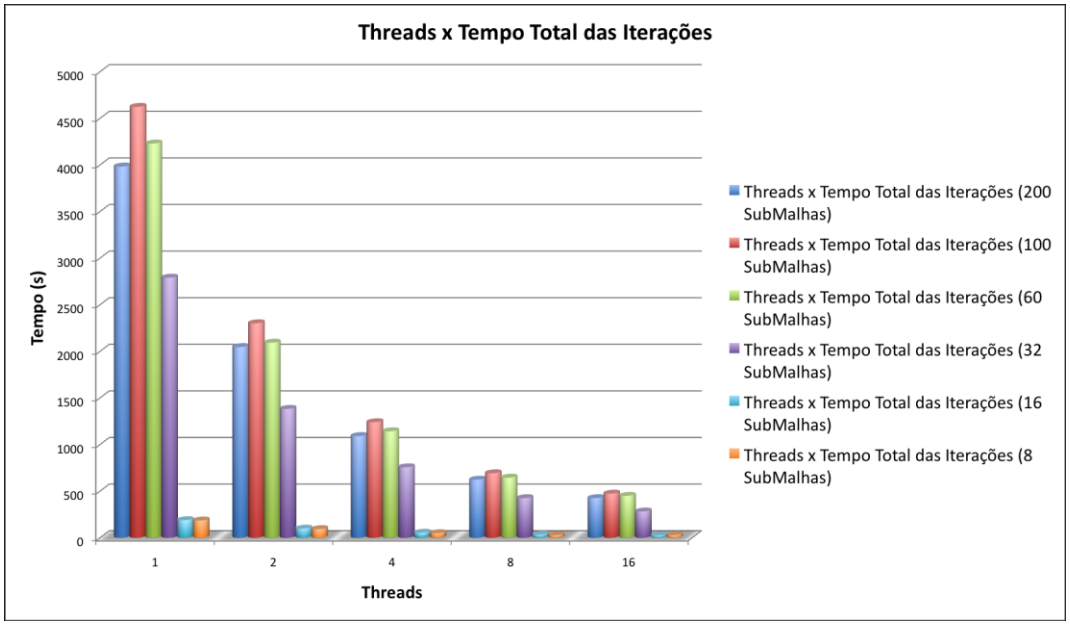

Figura 13 – Tempo Total x Threads

Para a resolução do mesmo sistema usando o método cholesky com armazenamento frontal em paralelo, sem subestruturação, obteve-se um tempo aproximado de duas horas no mesmo equipamento. O tempo para resolução do sistema, dividindo-se o edifício em 16 subestruturas, utilizando-se método de Dohrman (2003) com 8 "threads" é cerca de 6 minutos, o que representa um ganho estimado de mais de 1900%.

8 CONCLUSÃO

Os resultados demonstram a eficiência paralela da técnica de subestruturação, qualquer que seja o número de subdomínios. Repetindo-se o cálculo para um número crescente de processadores, o "*speedup*" ou o ganho de velocidade com relação ao número de processadores é quase linear com até 8 processadores.

Para a decomposição em número maior de subestruturas o número de iterações foi superior a 200. A eficiência da parelelização segue o padrão do particionamento em 8 subestruturas: a CPU consegue decompor até oito sistemas de equações simultaneamente.

Edifícios modelados com elementos sólidos podem ser simulados eficientemente utilizando a técnica iterativa idealizada por Dohrmann (2003). Nas configurações onde cada andar é alocado a um processador a técnica de subestruturação tem teor de paralelização próximo a 100%.

O número de iterações para convergência do método iterativo separando um andar por subestrutura foi de 22, enquanto o número de iterações para a decomposição em 200 subestruturas foi de 620. Isto demonstra que o particionamento de um prédio por andar traz vantagens em termos de convergência.

Apesar do particionamento em 200 subestruturas exigir 620 iterações, este resultou num tempo menor de execução total. O motivo da eficiência deste caso é que o tempo para decomposição das matrizes das subestruturas foi menor do que em relação ao caso com 8 subestruturas. O tempo dominante no caso de 8 subestruturas foi o relacionado à decomposição das matrizes de cada subestrutura. Já no caso de 200 partições, o tempo dominante passou a ser aquele relacionado às iterações do método do gradiente conjugado.

A implementação orientada para objetos da técnica de subestruturação de Dohrmann (2003) resultou em uma estrutura de classes de fácil utilização pelo usuário. A paralelização das operações fez-se em vários níveis, tais como: cálculo das matrizes elementares em paralelo, "assemblagem" e decomposição das matrizes das subestruturas em paralelo, multiplicação "matriz x vetor" em paralelo e precondicionamento paralelo. O número de "threads" utilizado para cada operação pode ser configurado pelo usuário.

REFERÊNCIAS

Devloo, P. R. B., *PZ – Object Oriented Finite Element Software*. Technical report. 2005.

Dohrmann, Clark R., *A Preconditioner for Substructuring Based on Constrained Energy Minimization,* SIAM J. Sci Comput., 25(1):246-258, 2003.

Elias, R. N. ; Camata, J. J. ; Aveleda, A.A. ; Coutinho, A. L. G. A. . *Evaluation of Message Passing Communication Patterns in Finite Element Solution of Coupled Problems*. In: VECPAR - International Meeting High Performance Computing for Computational Science, 2010, Berkeley, California, EUA. Proceedings of the 9th VECPAR (2010), 2010.

Farhat, C. ; Roux, R. X., *A method of finite element tearing and interconnecting and its*

parallel solution algorithm, Internat. J. Numer. Methods Engrg., 32 (1991), pp. 1205–1227.

Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, University of Minnesota – Karypis Lab, 1998.

Karypis, G., *METIS - A Software Package for Partitioning Unstructured*

Ribó, R., Riera, M. P. et al., *GiD 10 – The Personal Pre and post processor (version 10.0).* International Center For Numerical Methods In Engineering (CIMNE), 2010.