

IMPLEMENTAÇÃO PARALELA BASEADA EM *OpenMP* DO PROCESSO DE SIMULAÇÃO DA ESCAVAÇÃO DE UM TÚNEL

GUILHERME PIVA¹; ALINE PALIGA²; GERSON CAVALHEIRO³; JANSEN AVILA⁴; EDUARDO COUTO⁵

¹Universidade Federal do Rio Grande do Sul – guilherme.piva.santos@gmail.com

²Universidade Federal de Pelotas – aline.paliga@ufpel.edu.br

³Universidade Federal de Pelotas – gerson.cavalheiro@inf.ufpel.edu.br

⁴Universidade Federal de Pelotas – jrdavila@inf.ufpel.edu.br

⁵Universidade Federal de Pelotas – e.costacouto@gmail.com

1. INTRODUÇÃO

Diversos problemas com importância para a Engenharia podem ser modelados por meio de equações diferenciais parciais. Com exceção de alguns casos particulares, não é possível chegar uma solução analítica exata para estes problemas. O Método dos Elementos Finitos (MEF) é, atualmente, o método numérico mais utilizado para obter soluções aproximadas para este tipo de problema (CHANDRUPATLA, 2014).

O MEF gera um sistema de equações lineares cuja resolução é a solução aproximada do problema em estudo. A decisão de utilizar o MEF em uma simulação numérica implica em duas escolhas importantes: (i) a técnica a ser utilizada para o armazenamento do sistema de equações e (ii) o método de solução deste sistema. Estas duas escolhas são de fundamental importância, pois o sistema de equações poderá ter dimensões tais que uma técnica imprópria de armazenamento poderá ultrapassar a capacidade de memória do computador. Por outro lado, a escolha do método de solução do sistema de equações tem relação direta com o tempo de processamento (VAZ, 2011).

Na solução do sistema de equações gerado pelo MEF, entre os métodos iterativos, destaca-se o Método dos Gradientes Conjugados Pré-condicionados (MGCP).

Com relação as técnicas, associadas aos métodos iterativos, de armazenagem da matriz de rigidez, destaca-se a técnica elemento por elemento (EPE) que dispensa a montagem da matriz de rigidez global.

Em 1997 foi criado o *Open Multi Processing (OpenMP)*, que é uma interface baseada em *threads* (fluxos de processamento) utilizada para prover paralelismo incremental em máquinas com memória compartilhada (multiprocessadores) que tem sido bastante utilizada, sobretudo por sua simplicidade (CHAPMAN et al., 2008).

No MGCP o produto matriz-vetor e o produto interno aparecem de forma recorrente. Essas operações são facilmente paralelizáveis e isso estimulou o uso combinado do MGCP e da interface *OpenMP* que proporciona maior velocidade na obtenção dos resultados acrescida da técnica de armazenagem EPE que proporciona menor uso da memória (SHENG et al, 2016).

O presente trabalho tem como objetivo desenvolver um novo *solver* baseado no MGCP e que utiliza a técnica de armazenagem EPE e avaliar sua eficiência em um código que analisa os deslocamentos em um maciço rochoso quando é realizada a escavação de um túnel. Com esse fim foram realizadas comparações de desempenho entre algoritmos sequencial e paralelo do MGCP-EPE e um algoritmo sequencial direto que utiliza o método de eliminação de Gauss

associado à técnica de armazenagem *skyline*. O código usado nas comparações de desempenho foi implementado em FORTRAN90 e validado em COUTO (2011).

2. METODOLOGIA

Para avaliar a eficiência computacional do *solver* desenvolvido em simulações que visam encontrar os deslocamentos de um maciço rochoso na escavação de um túnel, realizou-se uma simulação onde o maciço rochoso está representado por uma malha de 6808 elementos hexaédricos de 20 nós e 3 graus de liberdade por nó. A malha que representa o maciço ilustra a Figura 1.

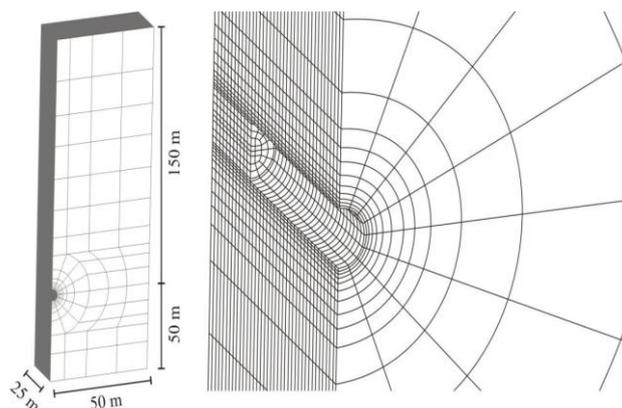


Figura 1: Detalhe da malha de 6808 elementos.

Nessa simulação o maciço foi considerado elástico e seus parâmetros estão na Tabela 2. Os parâmetros do túnel e do processo de escavação estão na Tabela 3.

Tabela 2 – Parâmetros do maciço.

E_m	ν_m	C_m	ϕ_m	k_0	γ_m
500 MPa	0,45	1,2 MPa	4°	0,75	25 kN/m ³

Sendo que, E_m é o módulo de elasticidade do maciço; ν_m é o coeficiente de Poisson; C_m a coesão; ϕ_m o ângulo de atrito; k_0 o coeficiente de empuxo horizontal e γ_m peso específico.

Tabela 3 – Parâmetros do túnel e do processo de escavação.

Seção Transversal	Circular
Raio (R)	1,65 m
Passo do avanço	1/3
Distância do Revestimento à face (d_0)	0
Profundidade	150 m

Em relação a simulação numérica da escavação de um túnel, a distância escavada foi de 0,55 m e a tolerância utilizada nas simulações que foram

realizadas com o *solver* implementado com o MGCP-EPE foi de 0,001. Além disso o equipamento utilizado para geração dos resultados foi uma máquina dedicada, com dois processadores Intel Xeon CPU ES-2630 2.2 GHz, cada um possuindo 20 threads, 32GB de memória, sistema operacional Windows 10 e compilador PGI versão 19.04

3. RESULTADOS E DISCUSSÃO

Após simular uma escavação de 0,55 m em um maciço rochoso, na Figura 2 encontra-se a comparação dos deslocamentos radiais no teto de um túnel, obtidos com o *solver* original, que usa um método direto, e com o *solver* desenvolvido.

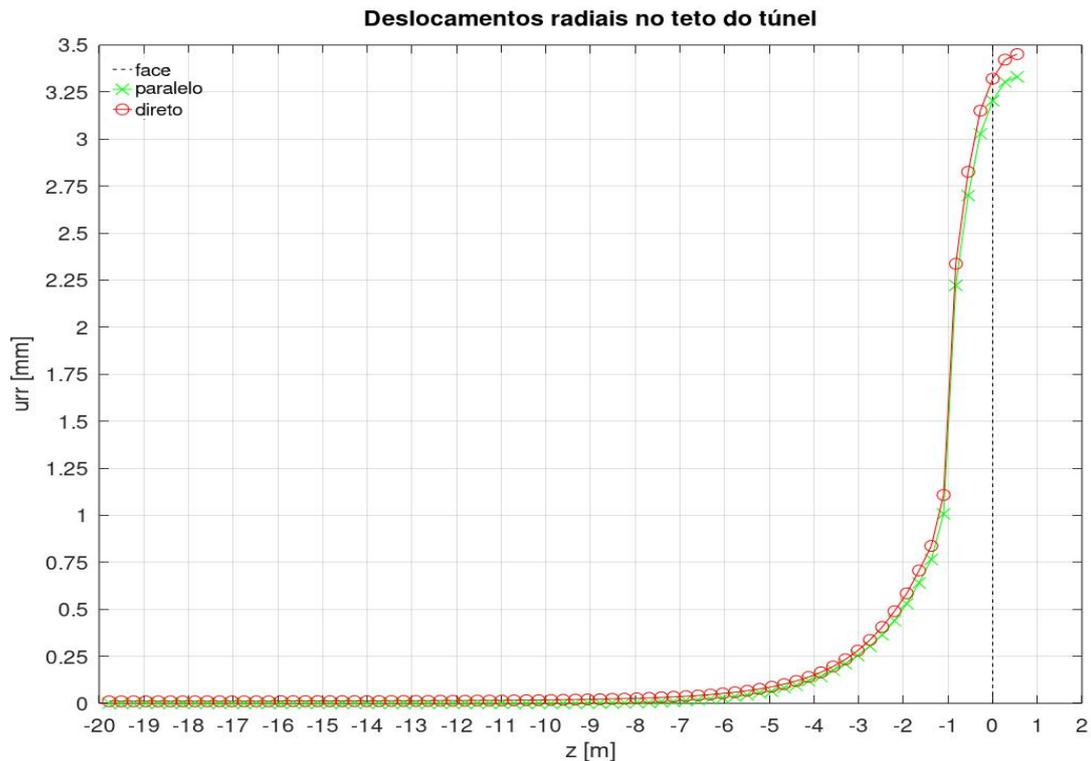


Figura 2 - Deslocamentos radiais no teto do túnel versus z.

A boa aproximação das curvas representadas na Figura 2 mostram a eficiência do *solver* desenvolvido. O maior deslocamento radial corresponde à posição z igual à 0,55 m. À esse deslocamento corresponde a maior diferença entre a solução que utiliza o método direto e a solução paralela que utiliza 32 *threads*. Essa diferença é de 3,90%. É decorrente das tarefas, na solução paralela, serem concorrentes e acontecerem em uma ordem não determinista como ocorre na solução direta. A Tabela 4 compara o tempo consumido e a memória RAM ocupada pelas simulações representadas na Figura 2.

Tabela 4 - Tempo e memória RAM das versões direta e paralela com 32 threads.

Solver	Tempo (s)	Memória RAM (GB)
Direto	1.695	3,5
Paralelo com 32 threads	270	1,9

As informações da Tabela 4 mostram que o solver paralelo proporciona grande redução no custo computacional quando comparado com o que usa o método direto. O tempo consumido pela simulação é 84,07% menor e a memória RAM ocupada é 45,71% menor.

4. CONCLUSÕES

Este trabalho apresentou o algoritmo de um *solver* baseado no Método dos Gradientes Conjugados Pré-condicionado associado a técnica de armazenamento Elemento por Elemento e a interface *OpenMP*. O código deste foi escrito em FORTRAN90. O *solver* daí resultante foi testado em um código pré-existente que analisa os deslocamentos de um maciço rochoso quando é realizada a escavação de um túnel. O solver original deste código é baseado no método de eliminação de Gauss associado à técnica de armazenagem *skyline*.

Posteriormente foram realizadas as simulações de escavação de um túnel. Confirmando os resultados anteriores, o *solver* desenvolvido apresentou uma significativa redução de tempo e de memória RAM ocupada quando comparado ao solver original que usa um método direto.

5. AGRADECIMENTOS

O orientador Eduardo Costa Couto agradece à FUNDAÇÃO DE AMPARO À PESQUISA DO ESTADO DO RIO GRANDE DO SUL – FAPERGS pelo financiamento da pesquisa onde este trabalho se insere.

6. REFERÊNCIAS BIBLIOGRÁFICAS

CHANDRUPATA, T. R., BELEGUNDU, A. D. **ELEMENTOS FINITOS**. São Paulo: Pearson, 2014.

CHAPMAN, B., JOST, G., PAS, R. V. **Using OpenMP: Portable Shared Memory Parallel Programming**. Massachusetts: MIT Press, 2008.

COUTO, Eduardo Costa. **A Numerical Approach for Design of Bolt-Supported Tunnels Regarded as Homogenized Structures**. Elsevier, Tunneling and Underground Space Technology, Vol. 24, pp 533-546, 2011.

FU, X. D., SHENG, Q., ZHANG, Y. H., CHEN, J. Investigation of Highly Efficient Algorithms for Solving Equations in the Discontinuous Deformation Analysis Method. **International Journal for Numerical and Analytical Methods in Geomechanics**, Wiley, v.4, p. 469 - 486, 2016.

VAZ, L. E. **Método dos Elementos Finitos em Análise de Estruturas**. Rio de Janeiro: Elsevier, 2011.