ADAPTAÇÃO TIPO H APLICADA À SIMULAÇÃO DE RESERVATÓRIOS DE PETRÓLEO USANDO PROGRAMAÇÃO ORIENTADA A OBJETOS EM C++

Érika Silva, Rogério Soares, Bruno Luna,
Darlan K. E. De Carvalho, Paulo R. M. Lyra
Depto de Engenharia Mecânica, CTG, UFPE,
50670-901, Recife, PE
kikasilva28@hotmail.com, rogsoares@yahoo.com.br, bruno_bl@yahoo.com.br,
dkarlo@uol.com, prmlyra@ufpe.com.br

Resumo: *No presente trabalho foi desenvolvido um programa computacional para refinamento adaptativo de malhas "tipo h", utilizando os conceitos de programação orientada a objetos em C++. Este programa foi implementado com o uso de um gerenciador de malhas gratuito chamado FMDB (Flexible Distributed Mesh Data Base). A adaptação de malhas tipo h é usada para se obter a nova discretização do domínio, e ela trabalha não apenas no enriquecimento da malha, mas também no sentido contrário pelo reagrupamento de elementos. A determinação do reagrupamento ou enriquecimento se dá através de uma análise de erros "a posteriori". O programa computacional de adaptação, análise de erros e interpolação linear de dados foi incorporado a um programa de análise via método dos volumes finitos com formulação tipo IMPES (Implicit Pressure Explicit Saturation) para a simulação de escoamentos bifásicos em meios porosos. Um exemplo modelo de adaptação no regime estacionário na solução de um escoamento monofásico é apresentado para demonstrar o funcionamento das ferramentas desenvolvidas.*

Palavras-chave: adaptação de malhas; programação orientada a objeto; simulação de reservatórios

Introdução

Muitos estudos estão sendo realizados para o aperfeiçoamento de ferramentas de simulação numérica de reservatórios, principalmente para a diminuição dos erros nos resultados desta simulação. Para controle da qualidade desses resultados, estratégias de adaptação de malha são as técnicas mais comuns, sendo também recomendáveis para simulação de reservatórios que apresentam problemas com singularidades e frentes se movendo ao longo do domínio. Uma grande vantagem do uso de malhas não estruturadas, e assim os métodos numéricos adequados a este tipo de discretização, é a existência de uma grande flexibilidade em relação aos métodos para malhas estruturadas, permitindo a discretização de geometrias complexas e a incorporação da adaptação de malhas de forma natural e eficiente.

O programa de simulação usado para aplicar os procedimentos de adaptação tipo h é um programa desenvolvido em C++ que simula o escoamentos bifásico óleo-água 2D/3D em meios porosos heterogêneos e anisotrópicos. As equações de fluxo são discretizadas pelo método de volumes finitos centrado nos vértices, com volume de controle dito "median dual", usando uma estrutura de dados baseada em arestas [3,5]. A formulação empregada no simulador é a formulação IMPES (Implicit Pressure Explicit Saturation). Este simulador também faz uso do gerenciador de malhas FMDB. Logo, no presente trabalho o enfoque principal foi a adaptação automática tipo h de malhas aplicada à simulação de reservatórios de petróleo.

Implementação Computacional

A escolha da linguagem de programação C++ foi feita visando as vantagens e os recursos que o paradigma da programação orientada a objetos pode oferecer a quem desenvolve programas grandes e complexos. Por programação orientada a objetos entende-se qualquer linguagem ou técnica que suporte diretamente: abstração, herança e polimorfismo [6].

Desenvolver programas de simulação científica usando C++ traz os benefícios de ter um código fácil de corrigir, mais compreensivo e de fácil manutenção. Com base nessas premissas e nos paradigmas inerentes à linguagem escolhida, foi desenvolvido um adaptador de malhas 2D para em seguida ser incorporado ao um simulador de reservatórios de petróleo, embora possa ser utilizado em outros tipos de simulação.

Adaptação em malhas estruturadas ou não-estruturadas é uma tarefa complexa para domínios em duas dimensões, 2D, e muito mais para 3D, o que torna imperativo o uso de um gerenciador de malhas robusto e flexível. O FMDB (Flexible Distributed Mesh Database) é uma alternativa que tem se mostrado bastante atraente em muitos aspectos. Por ser desenvolvido inteiramente em C++ e fazer uso

constante da biblioteca padrão STL (Standart Template Library), o que torna mais fácil de compreender suas rotinas, este pacote foi empregado no programa de adaptação. As complicadas rotinas de refinamento e desrefinamento necessárias durante a adaptação são todas gerenciadas pelo FMDB através de um recurso chamado *callback.* Por meio de uma interface abstrata (`mSplitCallbacks`) fornecida pelo pacote, o usuário desenvolve um tipo que herda as funcionalidades da classe base (`Adaptation_SplitCallbacks`) como mostra o trecho de código abaixo.

```
class Adaptation_SplitCallbacks : public mSplitCallbacks
{
public:

Adaptation_SplitCallbacks(mMesh *m);
~Adaptation_SplitCallbacks();
int operator()(mEntity *e);

private:

pMesh theMesh;

};
```

Para cada elemento da malha é chamado o operador `operator()`, permitindo que o usuário decida se o elemento deve ser refinado ou não. Todos os procedimentos de alteração da estrutura de dados da malha ficam a cargo do FMDB. Para que o adaptador se torne uma ferramenta de fácil incorporação a um simulador, independente da física envolvida, criou-se um tipo de forma a representar o propósito do adaptador através da classe Adaptation.

```
class Adaptation
{
public:

Adaptation();
~Adaptation();
void meshAdapt(pMesh theMesh);

};
```

Isso de fato representou uma grande economia de tempo quando foi implementado ao simulador de reservatórios de petróleo que também faz uso do FMDB como gerenciador de malhas.

```
while(simPar-finishSimulation(mData->tts))
{
(...)
if ( simPar->withAdaptation() )
{
      pAdapt->meshAdapt(theMesh);
}     (...) }
```

Outras duas características importantes e que serviu de estímulo ao uso deste pacote é a possibilidade de lidar com malhas 3D e com malhas distribuídas em simulação usando computadores paralelos. Estas características serão tratadas em trabalhos futuros.

Adaptação de Malhas Tipo h

A escolha da linguagem orientada a objetos C++ dá maior flexibilidade para o uso do algoritmo criado. A classe Adaptation foi incorporada ao simulador, eximindo o usuário do simulador conhecer a fundo o funcionamento deste algoritmo. A classe para adaptação de malhas realiza todos os passos referentes à adaptação tipo-h e se acopla com o simulador da forma apresentada na Figura 1.

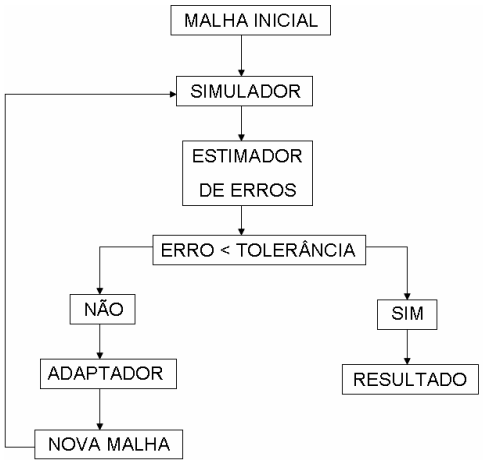

Figura 1 – Fluxograma do procedimento adaptativo para regime permanente

Através da simulação numérica se obtém uma solução aproximada para o fenômeno físico estudado. Essa aproximação é avaliada quanto à sua qualidade através de um estimador de erros [1]. Ao se constatar uma boa qualidade da solução aproximada chega-se ao fim do processo. Caso, ao final da simulação numérica se obtiver uma qualidade de solução abaixo da requerida, a estratégia de adaptação tipo-h é adotada.

No refinamento h temos a flexibilidade de proceder a subdivisões sucessivas de um elemento (ou agrupamentos sucessivos) para atender ao espaçamento da malha traduzido aqui pelo nível de refinamento/desrefinamento, ditado pela estimativa de erro, ou podemos limitar o número de refinamentos/desrefinamentos que executaremos em um único passo da adaptação.

A escolha dentre as opções possíveis tem implicações em termos de eficiência computacional, além do aspecto final da malha. A melhor escolha depende das características da aplicação em estudo, e uma maior investigação deve ser efetuada para tirarmos conclusões sobre este tópico.

O algoritmo foi baseado no trabalho de Araújo[2] e realiza os seguintes passos que serão detalhados a seguir:

i. Determinar nível de refinamento/ desrefinamento para cada elemento (Análise de erros);
ii. Limitar o desnível interelementos em uma unidade (Desnível Unitário);
iii. Assegurar boa regularidade da malha (Regularização da malha);
iv. Corrigir nível entre elementos "irmãos";
v. Efetuar Refinamento/Desrefinamento;
vi. Correção de nós irregulares;
vii. Interpolar linearmente a solução para a nova malha.

De posse da nova malha e solução interpolada, que será utilizada como nova solução inicial para acelerar a convergência do solver iterativo de resolução do sistema de equações lineares, efetua-se nova análise via MVF (Método dos Volumes Finitos). Este procedimento é repetido até que a solução obtida se encontre abaixo de uma tolerância pré-estabelecida.

Estimador de Erros

A ferramenta de análise de erros é a que controla todo o processo adaptativo, ela é quem indica os parâmetros para a formação da nova malha adaptada, assim como também os critérios de parada do macro algoritmo. As funções da ferramenta de análise de erros são duas. Primeiramente, esta ferramenta avalia a qualidade da solução. Se após a verificação da acurácia da aproximação, constatar-se uma baixa resolução, i.e., abaixo daquela requerida pelo usuário, a própria ferramenta de estimativa de erros, através de procedimentos numéricos baseados no estudo da convergência da solução, fornece os parâmetros para a formação da nova malha.

Inicialmente, considerando um problema unidimensional no qual a solução "u" é aproximada por "$\hat{u}$" através de interpolação linear [1]. O erro global ao longo de um elemento "e" pode ser definido como

$$E_e = u - \hat{u} \qquad (1)$$

Expandindo a solução "u" numa extremidade do elemento "e", assumindo ainda que o erro nodal seja zero, o erro E_e pode ser escrito como

$$E_e = \frac{1}{2} x\left(h - x\right) \left.\frac{d^2u}{dx^2}\right|_e \qquad (2)$$

onde "h" representa o comprimento do elemento. O erro de interpolação RMS (Root Mean Square), E_e^{RMS}, sobre um elemento ao longo do intervalo $\left[0, h\right]$ pode ser definido como

$$E_e^{RMS} = \frac{1}{\sqrt{120}} h^2 \left|\frac{d^2u}{dx^2}\right|_e \qquad (3)$$

Portanto, o erro de interpolação para o caso 1-D é proporcional ao produto da derivada segunda e o quadrado do comprimento característico do elemento "h".

Estendendo estas idéias para 2-D, as derivadas segundas são substituídas pela matriz Hessiana. Neste caso, o erro ao longo de uma aresta "ij" de um elemento "e" pode ser estimado como

$$E_{ij}^{RMS} \approx \left|\vec{L}_{ij}\right|^2 \left|\frac{\partial^2 u}{\partial x^2}\right|_{ij} \qquad (4)$$

O termo, $\vec{L}_{ij} = \vec{x}_j - \vec{x}_i$ é o vetor que define a aresta, $\left|\vec{L}_{ij}\right|^2$ é o quadrado do comprimento desta aresta, e, $\left|\partial^2 u / \partial x^2\right|_{ij}$ é a derivada segunda calculada ao longo da aresta. Desta forma, podemos estimar o erro ao longo da aresta como

$$E_{ij}^{RMS} \approx \left|\frac{\nabla\hat{u}_j \cdot \left(\vec{L}_{ij} / \left|\vec{L}_{ij}\right|\right) - \nabla\hat{u}_i \cdot \left(\vec{L}_{ij} / \left|\vec{L}_{ij}\right|\right)}{\left|\vec{L}_{ij}\right|}\right| \left|\vec{L}_{ij}\right|^2 = \left|\left(\nabla\hat{u}_j - \nabla\hat{u}_i\right) \cdot \vec{L}_{ij}\right| \qquad (5)$$

onde $\nabla\hat{u}_i$ é o gradiente aproximado da solução no nó "*i*". É importante mencionar que estes gradientes já estão disponíveis para o módulo de análise de erros, dado que o método de volumes finitos por arestas utilizado no presente trabalho produz esta informação naturalmente durante a discretização. Portanto, o custo de cálculo deste estimador de erros na aresta resume-se a uma subtração de vetores e um produto escalar. Tendo em vista que nosso processo adaptativo envolve a subdivisão ou o agrupamento de elementos, utilizamos os valores por arestas para definir uma estimativa de erros para o elemento, simplesmente como a média aritmética dos valores nas arestas, de modo que, podemos escrever

$$E_e^{RMS} = \frac{1}{3} \sum_{ij} E_{ij}^{RMS} = \frac{1}{3} \left(\sum_{ij} \left|\left(\nabla\hat{u}_j - \nabla\hat{u}_i\right) \cdot \vec{L}_{ij}\right| \right) \qquad (6)$$

onde, $\sum_{ij}$ indica que o somatório se dá sobre as três arestas do elemento triangular.

No presente trabalho, adotamos duas percentagens diferentes para os erros relativos. A primeira computa os erros de todos os elementos da malha (η_1). Visando tratar problemas com singularidades na solução, como no caso dos poços injetores e produtores, adotamos uma segunda percentagem (η_2), retirando-se dos cálculos todos os elementos que atinjem dimensões abaixo de valores pré-estabelecidos (ver [2] para maiores detalhes). Se uma das duas percentagens de erro estiverem acima das tolerâncias, o programa de análise de erros deve definir os parâmetros que conduzirão à construção da malha adaptada. Como estamos utilizando malhas anisotrópicas, estes parâmetros são os espaçamentos desejados para os elementos da nova malha. Desta forma, o indicador produzido para o processo de adaptação é uma distribuição de espaçamentos, para os elementos da malha adaptada, que conduzam a um erro relativo menor ou igual às tolerâncias pré-estabelecidas.

Adotando o princípio da equidistribuição de erros, e aplicando a teoria das estimativas de erros "*a-priori*" podemos obter os espaçamentos requeridos para a malha a cada adaptação, pela seguinte expressão

$$d_e^{m_1} = \frac{d_e^{m_0} \cdot e_m}{\left\| e_e^* \right\|^{m_0}} \qquad (7)$$

sendo e_m o erro médio, calculado como

$$e_m = Tol_i \cdot \frac{\left\| \nabla \hat{u}^* \right\|}{\sqrt{N_e}} \qquad (8)$$

onde

m_0 – malha corrente;
m_1 – malha adaptada;
$d_e^{m_0}$ – dimensão característica do elemento "e" na malha corrente;
$d_e^{m_1}$ – dimensão característica desejada para o elemento *e* na malha adaptada;
N_e – número de elementos da malha.
Tol_i – Tol_1 ou Tol_2 associadas ao erro em todo o domínio e do domínio expurgando os elementos com dimensões menores que a mínima.

Pelo fato de adotarmos duas tolerâncias distintas, a Eq. (7) que calcula as dimensões desejadas para os elementos da nova malha conduzirá a dois espaçamentos distintos para os elementos que deverão ser adaptados [2]. Nesse caso adota-se o parâmetro mais conservador, isto é, o que conduza a uma malha a favor da segurança.

Interpolação entre as Malhas

Para a adaptação tipo h, existe a vantagem de que qualquer novo elemento da malha adaptada sempre será proveniente da subdivisão de outro elemento da malha anterior diretamente conhecido. Dessa forma tem-se grande eficiência no processo, pois não precisamos de buscas na malha. A base para a interpolação de resultados utilizada considerou as funções de interpolação para elementos finitos isoparamétricos lineares por se tratar de uma malha formada apenas por triângulos [2]. Desta forma, a interpolação é dada através de

$$\varphi_n(r,s) = h_1(r,s)\cdot\varphi_1 + h_2(r,s)\cdot\varphi_2 + h_3(r,s)\cdot\varphi_3 \qquad (9)$$

onde r e s formam o sistema de coordenadas locais (ver Figura 2), φ_1 é a aproximação da solução proveniente da análise para o nó i e $h_i(r, s)$ são as funções de forma lineares indicadas pelas Eq. (10).

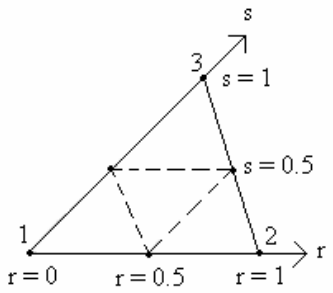

Figura 2 – Sistema de coordenadas locais para elementos triangulares.

$$\begin{aligned} h_1(r,s) &= 1 - r - s \\ h_2(r,s) &= r \\ h_3(r,s) &= s \end{aligned} \qquad (10)$$

Formulação Numérica

O programa de análise numérica utilizado é baseado no Método dos Volumes Finitos por Arestas (MVFA) com volumes de controle centrados nos vértices (vertex centered) com volumes de controle construídos pelo método das medianas (median dual control volume). Nesta formulação, os termos elípticos, existentes na equação de difusão, são calculados em dois "loops" nas arestas da malha. Num primeiro passo são calculados os gradientes nodais os quais são utilizados em conjunto com uma aproximação de diferenças finitas ao longo das arestas da malha para aproximar os termos de difusão cruzada. No caso de meios heterogêneos, estas aproximações são realizadas domínio por domínio a fim de respeitar

a descontinuidade nos gradientes entre os diferentes materiais. Esta formulação tem se mostrado bastante acurada para a discretização de escoamentos em meios heterogêneos e anisotrópicos. Para maiores detalhes, veja [2,3,5].

Resultados

Os resultados apresentados neste trabalho abordam o problema clássico da produção no "five-spot", sendo o escoamento monofásico, e visa demonstrar o desempenho no procedimento adaptativo desenvolvido na solução da equação de pressão no regime permanente. Os parâmetros utilizados para a simulação do problema estão listados na "Tabela 1". O simulador resolve o sistema de equações algébricas resultante implicitamente ($A.x = B$), usando o PETSC (Portable Extensible Toolkit for Scientific Computation) com o solver GMRES (Generalized Minimal Residual method), com tolerância 10^{-6}. Na adaptação da malha limitamos a um refinamento apenas a cada passo do processo, o que garante uma malha final "ótima", porém não garante que o processo como um todo seja o mais eficiente, pois exige sucessivas análises do problema.

Tabela 1 - Parâmetros utilizados no problema Five-Spot.

Condições de Contorno (pressão)	$p_i = 1.0$ (poço injetor) $p_p = 0.0$ (poço produtor) Fluxo zero pela fronteira
Condição Inicial (saturação de água)	$S_{w_i} = 1.0$ (poço injetor) $S_w(x, t = 0) = 0$
Geometria	Quadrado de lado 1.0
Tensor de permeabilidade (meio homogêneo)	$K = \begin{pmatrix} 1.0 & 0 \\ 0 & 1.0 \end{pmatrix}$
Porosidade	$\phi = 0.2$

Nas Figura 3 e Figura **4**, apresentamos a cada adaptação a malha e campo de pressão na região próxima dos poços injetor e produtor. Estas regiões são as que apresentam o gradiente de pressão mais elevado e por isso é necessário um maior refinamento nesta área, aproximando melhor a pressão em todo reservatório.

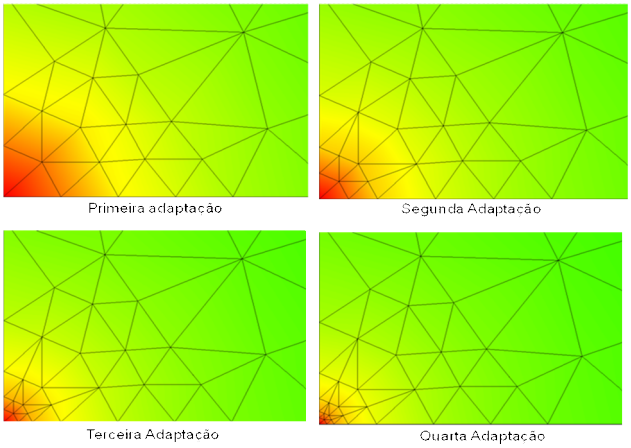

Figura 3 – Campo de pressão no poço injetor após adaptação.

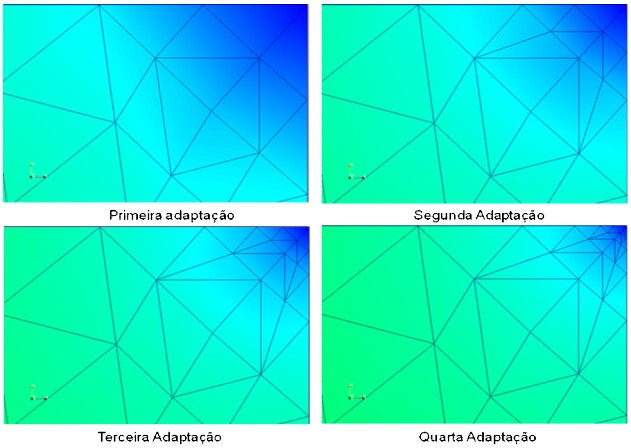

Figura 4 – Campo de pressão no poço produtor após adaptação.

A Figura 5 mostra uma distribuição do campo de pressões (poço a poço). A cada passo de adaptação a distribuição se torna mais suave pela presença do maior refinamento próximo aos poços. Os resultados obtidos comparam o campo de pressão obtido em malhas que passaram pelo processo de adaptação uma, duas, três e quatro vezes, respectivamente.

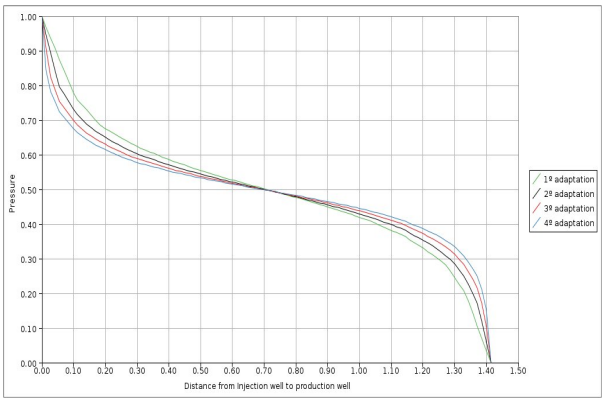

Figura 5 – Distribuição na diagonal do campo de pressão a cada malha adaptada.

Conclusão

O procedimento adaptativo se mostra capaz de capturar gradientes elevados, melhorando assim a captura dos pontos críticos do campo de pressão e com isto de todo o campo de pressão. Os resultados apresentados são preliminares e serviram para verificar todos os componentes do procedimento adaptativo na análise de um problema modelo monofásico, envolvendo apenas a equação de pressão em regime estacionário.

A extensão do algoritmo de adaptação para lidar com aplicações transientes não representa grandes desafios e será o próximo passo desta pesquisa e que permitirá então a incorporação da adaptação na simulação de escoamentos bifásicos em reservatórios. Além disso, todos os algoritmos desenvolvidos foram concebidos visando uma extensão direta para três dimensões, fazendo uso dos recursos existentes no FMDB para lidar com malhas 3D. O uso da linguagem orientada a objetos, C++, torna o algoritmo de fácil manutenção e também facilita a sua incorporação como objeto em um programa de análise.

Finalmente, a longo prazo, esperamos investigar a extensão para lidar com computação paralela distribuída ("cluster" de PCs), onde a adaptação de malhas tem que ser acompanhada de um balanço dinâmico de carga nos diversos processadores.

Agradecimentos

Os autores deste trabalho agradecem ao PRH-26/ANP e ao CNPq pelo suporte financeiro e aos integrantes do grupo PADMEC que contribuíram para a realização desse trabalho.

Referências

[l] D. Ait-Ali-Yahia, W.G. Habashi, A. Tam, A Directionally Adaptive Methodology Using an Edge-Based Error Estimate on Quadrilateral Grids, *Internacional Journal for Numerical Methods in Fluids*, Vol. 23, 1996, pp. 637-690.

[2] F.S. Araújo, Procedimentos Adaptativos na Simulação de Reservatórios de Petróleo em Volumes Finitos, Dissertação de Mestrado, Engenharia Civil - UFPE, Recife, 2004.

[3] D.K.E. Carvalho, B. G. B. Luna, P.R.M. Lyra, R.B. Willmersdorf, J.P.S. Queiroz, Numerical Simulation of Oil-Water Displacements in Porous Media Using a Mesh Adaptive Finite Volume Formulation. COBEM-19th International Congress of Mechanical Engineering, Brasília-DF, 2007.

[4] FMDB, http://www.scorec.rpi.edu/FMDB/

[5] P.R.M. Lyra, R.C.F. Lima, C.S.C. Carvalho, D.K.E. Carvalho, An Edge-Based Unstructured Finite Volume Procedure for the Numerical Analysis of Heat Conduction Applications. *Journal of the Brazilian Society of Mechanical Enginnering*. Brazil, v.26, p. 160-169, 2004.

[6] B. Stroustrup, "Why C++ is not just an Object-Oriented Programming Language", OOPSLA, Austin, Texas, 1995.