

Frangopoulos, C. A. (1992b), 'Thermoeconomic versus thermodynamic design optimization of a single gas turbine plant. Part B: Thermoeconomic optimization, *The International Journal of Mechanical Engineering Education*, 20, N°3, 157-168.

Gallo, W. L. R. (1991), 'Análise exergética de motores a gasolina e a álcool.', Tese de Doutorado, Faculdade de Engenharia Mecânica, UNICAMP, Campinas, Brasil.

Larson, R. E., Casti, J. L. (1978), *Principles of Dynamic Programming. Part I & II*, Marcel Dekker, Inc., New York and Basel.

Moran, J. M. (1982), *A Guide to Efficient Energy Use*, Prentice-Hall, Inc. Englewood Cliffs, New Jersey.

Peretti, A., Toth, P. (1982) 'Optimization of a pipe-line for the natural gas transportation.', *European Journal of Operational Research*, 11, 247-254, North-Holland Publishing Company.

Nomenclatura :

c_p	calor específico a pressão constante ;
c	coeficientes de custo ;
C	função local de custo de x em k ;
d_k	perturbação externa ;
H_f	fluxo de energia do combustível : $H_f = m_f \cdot H_u$;
H_u	poder calorífico inferior do combustível ;
I	fluxo de irreversibilidade total ;
J	função de recorrência ;
k_i	$(\gamma_i - 1) / \gamma_i$;
m	fluxo mássico ;
n_w	volume de ar estequiométrico $(n + m/4 + f/2)$, para $C_n H_m O_f$;
R	constante do gás ;
r_b	relação de pressão na câmara de combustão, $r_b = P_3/P_2$;
r_c	relação de pressão no compressor, $r_c = P_2/P_1$;
r_t	relação de pressão na turbina, $r_t = P_3/P_4$;
S_k	entropia gerada ;
u_k	decisão no estado k ;
x_k	estado em k ;
y	fluxo económico ;
Z	custo anual de capital ;

Letras Gregas

α	percentual máximo para a compressão ;
γ	razão entre os calores específicos, $\gamma = c_p/c_v$;
λ	coeficiente de ar teórico ;
η	eficiência isoentrópica ;
η_w	eficiência politrópica ;
Ω_i	conjunto de decisões factíveis ;

Índices

a	ar ;
f	combustível ;
g	gases de combustão ;
B	Câmara de combustão ;
C	Compressor ;
T	Turbina ;

Sobrescrito

.	por unidade de tempo ;
---	------------------------

XIV Congresso Ibero Latino-Americano de Métodos Computacionais em Engenharia

Instituto de Pesquisas Tecnológicas
São Paulo - SP - Brasil
1 a 3 de dezembro de 1993

UM ROBUSTO GERADOR DE DIAGRAMAS DE VORONOI PARA DISCRETIZAÇÃO DE DOMÍNIOS IRREGULARES

Clovis R. Maliska Junior

Laboratório de Simulação Numérica em Mecânica dos Fluidos
e Transferência de Calor - SINMEC
Departamento de Eng. Mecânica - UFSC
Caixa Postal, 476
88040-900 - Florianópolis - SC

SUMÁRIO

Neste trabalho é apresentado um novo e robusto gerador de diagramas de Voronoi. Estes diagramas ganham cada vez mais espaço em simulações numéricas de fenômenos físicos dadas as suas características de construção, que permitem que fluxos sejam facilmente aproximados nas fronteiras dos volumes de controle. O gerador desenvolvido, além de apresentar eficiência computacional, permite gerar diagramas oriundos de distribuições aleatórias de pontos, podendo-se incluir regiões com malhas hexagonais, polares, cartesianas e de outros sistemas coordenados na discretização geral.

1. INTRODUÇÃO

Na simulação numérica de fenômenos físicos um dos passos iniciais, e às vezes mais complexo do processo, principalmente quando domínios bastante irregulares são considerados, é a discretização deste domínio. Na área de Mecânica dos Fluidos Computacional (CFD), com grande tradição no uso do método dos volumes finitos e malhas estruturadas, é grande o número de trabalhos sendo desenvolvidos atualmente na criação de metodologias numéricas para malhas não estruturadas. O uso de malhas estruturadas apresenta duas vantagens em relação às malhas não estruturadas. A primeira é a facilidade de discretizar domínios irregulares que apresentam protuberâncias, e a segunda é a adequação natural destas malhas para adaptatividade.

Usando discretização não estruturada existem duas correntes de pesquisa no desenvolvimento de métodos numéricos para problemas de escoamentos de fluidos. Uma delas é a aplicação do método dos elementos finitos tradicional com alteração das funções interpolantes para tratar problemas de alta convecção. A outra é baseada na experiência do uso do método dos volumes finitos em sistemas coordenados ortogonais procurando transferir estes conhecimentos para formulações com discretização não estruturada.

A geração das malhas não estruturadas, por sua vez, pode ser realizada

diversas formas. Uma delas dá origem a volumes cuja reta que une o seu ponto gerador ao ponto gerador vizinho é ortogonal a aresta comum com o volume vizinho, conforme pode ser visto na Fig.1.1. Os diagramas de Voronoi possuem esta característica e se apresentam como uma excelente alternativa para uma discretização não estruturada em problemas de escoamento de fluidos. Esta discretização não é, entretanto, difundida na área de CFD com volumes finitos, a não ser em recentes trabalhos na área de simulação de reservatórios de petróleo (Palagi, 1992). Os diagramas de Voronoi podem substituir com eficiência os volumes de controle gerados a partir de triangulações quaisquer empregados em formulações de elementos finitos baseados no volume de controle (Baiiga e Patankar, 1988), (Schneider, 1988). Tradicionalmente os diagramas de Voronoi têm sido empregados em aplicações como otimização de construções geométricas (Shamos e Hoey, 1975), reconhecimento de padrões, meteorologia (Rhynsburger, 1973), crescimento de cristais (Gilbert, 1962), etc., conforme relatado em (Avis e Bhattacharya, 1983).

Neste trabalho é apresentado um novo método para gerar diagramas de Voronoi através de um procedimento de incrementação de volumes. O algoritmo foi desenvolvido baseado unicamente na definição geométrica do diagrama de Voronoi, sem ter sido buscado na literatura algoritmos para servirem de base no desenvolvimento. O algoritmo, além de realizar a discretização do domínio, fornece também a numeração e as conexões de cada volume com seus vizinhos, dando a estrutura da matriz do sistema linear resultante do procedimento numérico. Em outras palavras, procurou-se desenvolver um gerador que, além de particionar um domínio, atenda às necessidades do analista numérico que fará uso da discretização. Os diagramas gerados pelo método aqui descrito podem ser usados por analistas

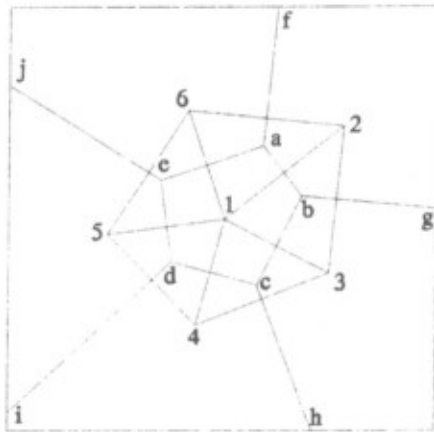


Fig.1.1 - Diagrama de Voronoi para 6 geradores

numéricos interessados em simular fenômenos físicos dos mais diferentes tipos.

2. DEFINIÇÃO DO DIAGRAMA DE VORONOI

O diagrama de Voronoi para um conjunto de pontos foi primeiramente definido por (Dirichlet, 1850) e (Voronoi, 1908). É conhecido também como particionamento de Dirichlet, poliedros de Voronoi, polígonos de Thiessen (Thiessen, 1911) e células ou domínios de Wigner-Seitz (Wigner e Seitz, 1933).

Primeiramente é importante fazer algumas considerações com relação ao diagrama de

Voronoi. Um diagrama de Voronoi é a discretização de uma geometria onde cada volume discreto gerado, também chamado de grid-block, é representado por um ponto gerador, também denominado grid-point. A definição de um volume ou grid-block de Voronoi para o espaço R^2 é:

Dado um conjunto de n pontos p e uma região do espaço R^2 , o volume de Voronoi para o gerador i é o conjunto dos pontos $(x, y) \in R^2$ que obedecem a:

$$(x, y) / (d((x, y) - p_i) < d((x, y) - p_j) \quad \forall j / 1 < j < n \text{ e } j \neq i$$

onde $d((x, y) - p)$ é a distância do ponto (x, y) ao ponto p .

Referindo-se a Fig.1.1, pode-se observar que o diagrama satisfaz a:

1 - O segmento de reta 1-2 é ortogonal ao segmento de reta a-b.

2 - A reta que passa por a-b passa no ponto médio de 1-2.

3 - O círculo que passa por 1,2,3 tem como centro o ponto b.

4 - Qualquer ponto escolhido dentro do polígono abcde está mais próximo do ponto 1 do que dos outros pontos dados.

3. METODOLOGIA BÁSICA DE GERAÇÃO

A metodologia utilizada para a geração do diagrama de Voronoi consiste na técnica de incrementação de volumes, isto é, dado um diagrama para $c - 1$ geradores, deseja-se incluir mais um gerador, ou seja, o gerador c . A criação desta rotina é a parte principal do gerador, pois, dado um espaço do R^2 e um conjunto c de geradores, basta definir-se o diagrama inicial como sendo todo o espaço onde deseja-se fazer a geração, tendo como seu ponto gerador um elemento qualquer da lista de geradores dados. Após este passo inicial para a geração do diagrama, vai-se incluindo todos os outros geradores da lista neste diagrama através da rotina de inclusão de um gerador. Feita a inclusão de todos os geradores, tem-se o diagrama de Voronoi gerado para aquele espaço e c geradores dados. Logicamente a dificuldade na elaboração do gerador está na implementação computacional desta rotina de inclusão. Nas próximas seções são descritos os passos principais de construção desta rotina.

4. PROCEDIMENTO DE INCLUSÃO DE UM VOLUME

4.1 Estrutura Computacional do Diagrama e de um Volume

O diagrama de Voronoi que aqui será gerado é uma lista de volumes, um vetor, onde cada posição deste vetor é uma estrutura do tipo **tipo_volume**. Assim, cada volume que desejamos achar possui um número, que é a sua posição na lista e a estrutura **tipo_volume**, que possui as seguintes partes:

(.cx, .cy) - coordenadas do grid-point do grid-block **vol_c**;

(.x[j], .y[j]) - coordenadas do ponto j do grid-block **vol_c**;

.viz[j] - vizinho direto do grid_block **vol_c** pela aresta j ;

Esta estrutura define uma região do espaço dado, como pode ser visto na Fig.4.1, que são os volumes de Voronoi para estes geradores. Além dessas definições para os elementos do grid-block, tem-se as seguintes definições para elementos da lista de grid-blocks e outros componentes auxiliares para a geração:

(px, py) - coordenadas do grid-point (c) (grid-point que está se incrementando);
vol_c - grid-block em que o ponto (px, py) está contido.
volume[i] - grid-block i do diagrama. Todas as notações sobre informações a cerca do grid-block citadas acima para **vol_c** servem também para o **volume[i]**;
volume[c] - grid-block do grid-point (c). Este é o grid-block que deseja-se achar.

Essas definições irão simplificar a explicação do algoritmo e auxiliarão na montagem

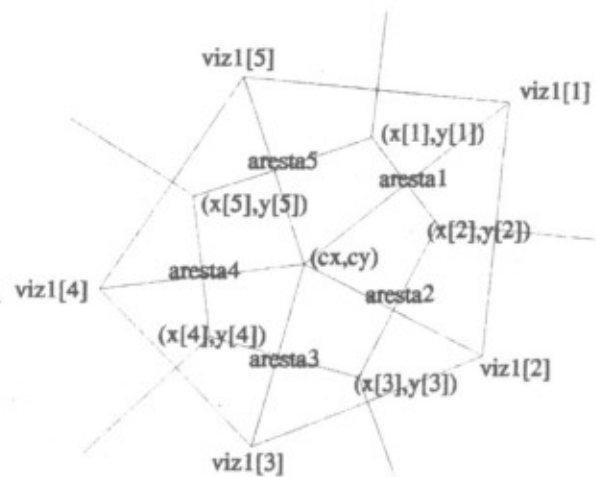


Fig.4.1 - Estrutura computacional (tipo_volume) para um volume de Voronoi.

da estrutura computacional do programa.

4.2 Rotinas básicas para a inclusão de um volume

O algoritmo para a inclusão de um volume baseia-se na definição geométrica do diagrama de Voronoi. Assim, é necessário que se possua uma pequena biblioteca de rotinas que façam as seguintes funções:

- Copiar um volume de uma estrutura para outra (duplicação para uma estrutura auxiliar);
- Dado um volume e uma reta secante a este volume, cortar o mesmo pela reta e devolver os pedaços cortados;
- Dados dois volumes e seus respectivos vizinhos (estes vizinhos já estão implícitos na estrutura do volume) fazer a fusão dos dois.

Essas são as rotinas mais difíceis de serem construídas, e são a essência do gerador. É interessante que estas rotinas sejam muito bem feitas, pois elas serão chamadas inúmeras vezes no gerador, e o desempenho deste será ditado pela velocidade dessas rotinas. Nas Fig.4.2(a) e Fig.4.2(b) podemos ver, respectivamente, o esquema de um volume sendo cortado por uma reta e a fusão de dois volumes com arestas coincidentes.

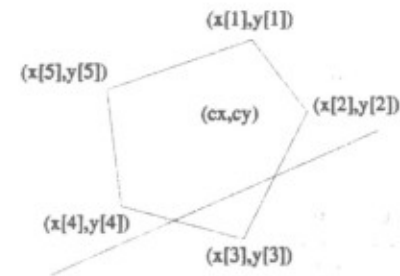


Fig.4.2(a) - Volume de Voronoi sendo cortado por uma reta secante.

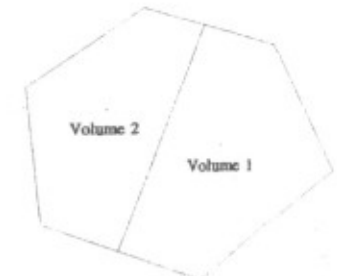


Fig.4.2(b) - Dois volumes de Voronoi sendo fundidos

4.3 Algoritmo para a inclusão de um volume

Abaixo é mostrado os passos para a inclusão de um volume. Esses passos usarão termos definidos nas seções anteriores e é importante que primeiro tenha-se a ideia de como funciona o algoritmo geometricamente antes de qualquer tentativa de implementação.

1 - Descobre-se da lista de grid-blocks do tempo $c-1$ quem é **vol_c**, ou seja, em que grid-block está contido o ponto (px,py).

2 - Traça-se um segmento de reta (px,py) - **vol_c(cx,cy)** e acha-se a reta mediatriz a esse segmento (reta perpendicular que passa pelo ponto médio do segmento, que chamaremos de reta **rc**). Essa reta é a que será a secante do grid-block **vol_c**.

3 - Processa-se o corte de **vol_c** pela reta secante **rc**. Deste corte resultam **vol_1** e **vol_2**. **vol_1** será a nova e final configuração de **vol_c** para o tempo c , e **vol_2** será um pedaço de **volume[c]**. *OBS:* É a partir deste pedaço que irá utilizar-se de um processo recursivo com o qual disporo das informações sobre os vizinhos no instante de geração $c-1$, saberemos dizer quais volumes será afetado pela incrementação do grid-point (c) e consequentemente quais irão entrar na formação do novo grid-block (**volume[c]**).

4 - Neste ponto as definições para poder se iniciar o processo recursivo para a definição de quais volumes serão afetados pela incrementação são dadas. Primeiro monta-se uma lista chamada **volumes**. Esta lista é um array que irá conter os elementos que já entraram na formação de **volume[c]**. Esta lista inicialmente possui um elemento que é **vol_c**, pois foi o único que contribuiu para a formação de **volume[c]**.

5 - Em seguida define-se uma função chamada **fora_lista**. Nesta função passa-se o numero de um volume em questão, e ela retorna 1 se o elemento não está na lista **volumes** e 0 se ele está.

6 - Inicia-se o processo recursivo fazendo um loop i de 1 ao numero de lados que possui **volume[c]**. Dentro deste loop passa-se o valor de **volume[c].viz1[i]** para **fora_lista**. Se **fora_lista** retornar 1, atribui-se o valor de **volume[c].viz1[i]** para **vol_corte**, aborta-se o

loop e passa-se para o passo 7. A variável `vol_corte` possui o número do elemento que irá ser cortado para formar `volume[c]`. Se o loop inteiro de `i` retornar 0, ou seja, não achou-se um valor para `vol_corte`, encerra-se o processo e o diagrama para o nível `c` de geração está pronto.

7 - Esta etapa será realizada somente se foi achado algum valor para `vol_corte`. Acrescenta-se o elemento `vol_corte` à lista `volumes` e processa-se o corte do elemento `volume[vol_corte]` pela reta `rp` (reta mediatriz de $(px,py) - volume[vol_corte].(cx,cy)$, ou seja, reta perpendicular ao segmento que liga esses dois pontos e que passa pelo ponto médio do mesmo). Deste processo resultarão `vol_1` e `vol_2`. `vol_1` será a nova configuração de `volume[vol_corte]` para o instante `c` de geração, e `vol_2` será utilizado no processo 8 para a formação de `volume[c]`.

8 - Após isso, funde-se o elemento `volume[c]` com `vol_2`. Feito isso, volta-se para o item 6.

4.4 Características do Gerador

Cada algoritmo de geração de diagramas de Voronoi possui características diversas. Além do tempo absoluto de processamento, uma característica importante é o aumento do tempo de processamento em função do aumento do número de pontos geradores. Isto é, o aumento do tamanho da malha.

O algoritmo apresentado neste trabalho consome para n geradores um tempo de CPU da $O(n)$, diferente de outros métodos que gastam para um número n de geradores um tempo da $O(\log_2 n)$. Isto deve-se ao fato de no processo de inclusão o algoritmo realizar cálculos apenas com os volumes vizinhos, não interferindo no tempo para estes cálculos o número total de volumes. Isto é vantajoso pois para 10 geradores um algoritmo de ordem n gasta 10 unidades de tempo, onde um algoritmo de ordem $\log_2 n$, por exemplo, gastaria 100 unidades de tempo.

Normalmente para os algoritmos de ordem n a unidade de tempo é bem maior do que para aqueles de ordem $\log_2 n$. Assim, a geração do diagrama para 10 geradores poderia ser mais rápida no segundo algoritmo do que no primeiro, mas para, por exemplo, 10^3 geradores, é quase certo que, apesar do unidade de tempo do primeiro ser maior que a do segundo, o primeiro será mais rápido, pois ele gastará somente $O(10^3)$, enquanto o outro gastará $O(10^6)$, o que pode ficar inviável dependendo da máquina que se está utilizando.

5. INTERAÇÃO USUÁRIO-GERADOR

A potencialidade que possui o diagrama de Voronoi em função da sua ortogonalidade é muito grande, mas é importante lembrar que o diagrama surge de um conjunto de pontos geradores definidos pelo usuário, e são esses pontos que irão determinar a concentração da malha, a distribuição e o formato dos volumes e diversos outros fatores importantes na simulação, que é o objetivo que se busca com a discretização do domínio. É necessário, então, que esses pontos sejam escolhidos de maneira adequada afim que de o diagrama gerado tenha volumes com a relação altura/largura o mais próximo de 1 (um). Para isso, foi escolhido três formas principais de se distribuir pontos dentro de uma área irregular: cartesiana, hexagonal e polar. Essas três configuração distribuem os pontos de maneira que os volumes gerados sejam retângulos, hexágonos ou setores circulares, respectivamente, como pode ser visto na Fig.5.1. Nesta figura pode-se ver uma geometria sendo discretizada com os 3 tipos de configurações de pontos geradores, evidenciando a versatilidade do gerador. Para que o gerador de diagramas de Voronoi seja de fácil utilização pelo usuário a entrega de configurações diferentes de pontos gerados dentro do domínio a ser discretizado deve ser

automatizada.

Para isso, é importante que se possua um programa que através do mouse definam-se as fronteiras, selecionem-se as regiões e o tipo de distribuição de pontos para a geração do diagrama. Com isso o trabalho de definir-se a distribuição dos pontos geradores fica extremamente simplificada e rápida. Tal facilidade está implementada no programa desenvolvido e está disponível no SINMEC, o qual é totalmente orientado por eventos e controlado com mouse. Ele foi desenvolvido em C++ e utiliza a biblioteca gráfica X-lib rodando em qualquer máquina do sistema UNIX e janelamento X-Windows.

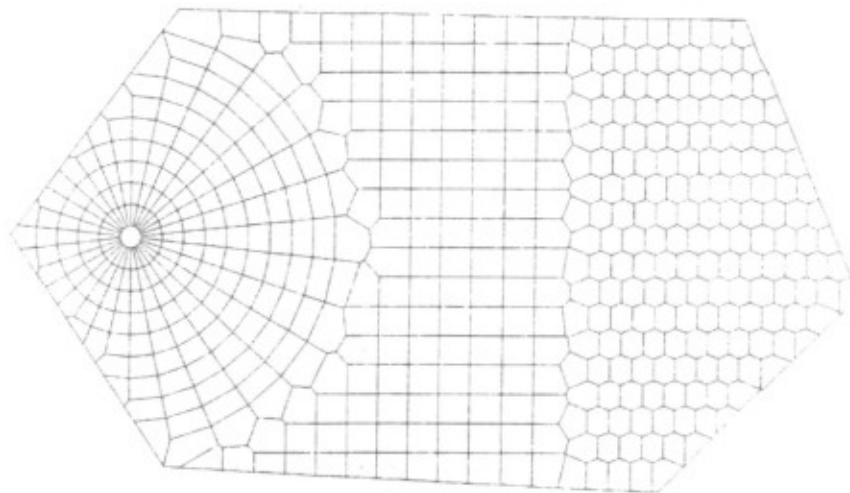


Fig.5.1 - Distribuição de pontos Cartesiana, Hexagonal e Polar para um Diagrama de Voronoi

5.1 O Dual de Voronoi

O dual de Voronoi consiste em, para cada um dos volumes do diagrama, ligar o seu centro (cx,cy) com o centro de cada um dos seus vizinhos. Esta ligação irá gerar a chamada triangulação de Delaunay. Podemos ver na Fig.5.2(a) o exemplo de um diagrama de Voronoi e na Fig.5.2(b) a triangulação para o mesmo. O dual do diagrama de Voronoi é largamente empregado na formulação em elementos finitos. O programa correspondente ao algoritmo aqui apresentado entrega ao usuário também o dual de Voronoi.

6. EXEMPLOS DE DIAGRAMAS GERADOS

Para demonstrar a versatilidade do gerador desenvolvido, nesta secção são apresentados diagramas obtidos para diferentes geometrias e distribuições de pontos geradores. Nas Figs.6.1 a 6.5 podemos observar diagramas de Voronoi, gerados para diferentes domínios onde as distribuições polar, cartesiana e hexagonal de pontos são empregadas. Os tempos de CPU para obter-se os diagramas são consideravelmente menores, para alguns casos testados do que aqueles necessários pelo algoritmo de Sugihara (Sugihara, 1992). Mais testes são entretanto, necessários para confirmar estes dados comparativos preliminares.

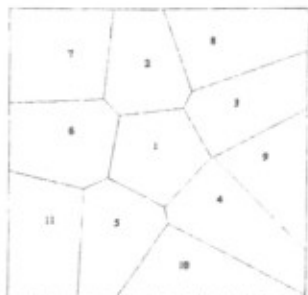


Fig.5.2(a) - Diagrama de Voronoi para 11 pontos geradores

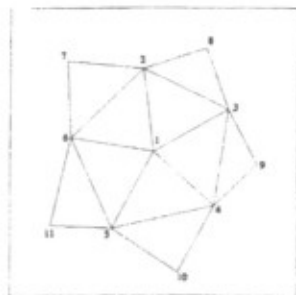


Fig.5.2(b) - Triangulação de Delaunay para 11 pontos geradores

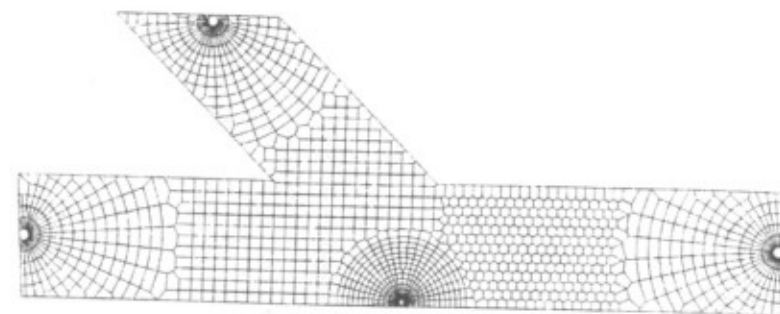


Fig.6.3 - Diagrama de Voronoi para 1390 pontos geradores

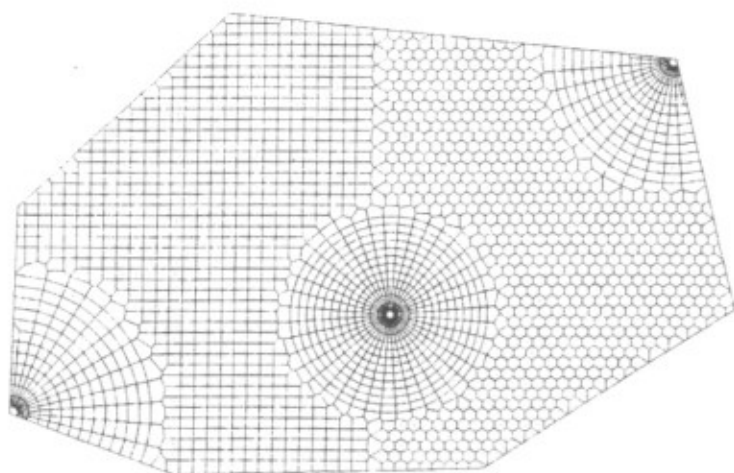


Fig.6.1 - Diagrama de Voronoi para 2201 pontos geradores

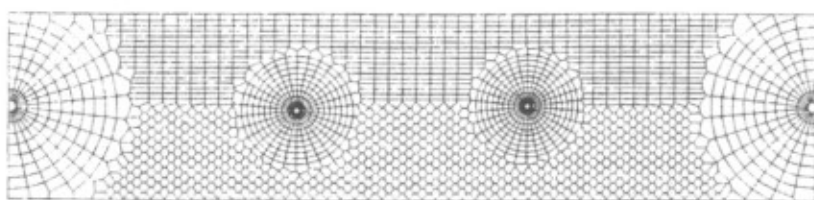


Fig.6.2 - Diagrama de Voronoi para 2121 pontos geradores

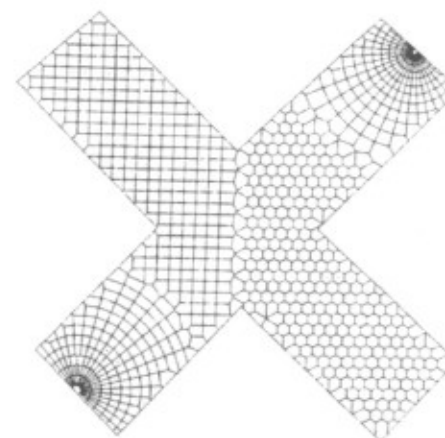


Fig.6.4 - Diagrama de Voronoi para 926 pontos geradores

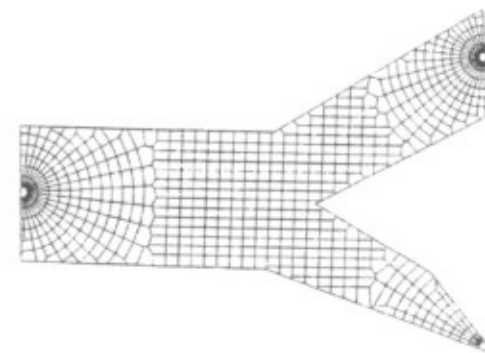


Fig.6.5 - Diagrama de Voronoi para 702 pontos geradores

7. CONCLUSÕES

O gerador de diagramas de Voronoi descrito neste trabalho, utilizando a técnica de incrementação de volumes, mostrou-se eficiente e versátil. Discretizações para diferentes geometrias envolvendo em um mesmo domínio diferentes distribuições de pontos, foram geradas. O tempo de computação é da ordem n , onde n é o número de pontos geradores. Com o intuito de utilizar estas discretizações em simulações numéricas o gerador entrega, no final do processo, todas as conexões entre os volumes vizinhos, facilitando a construção da matriz do sistema linear. Computacionalmente o gerador está implementado de tal forma a permitir ao usuário a entrada fácil, via mouse, da geometria do domínio, do número de pontos e de sua distribuição.

AGRADECIMENTOS

O autor agradece aos professores Waldir Leite Roque e Sergio Peters, do Departamento de Informática e Estatística da UFSC (INE), por terem fornecido uma cópia do gerador de (Sugihara, 1992) para comparações e por terem, também, incentivado o autor para este trabalho. Agradece também ao professor Clovis R. Maliska pelas sugestões e revisão do trabalho.

REFERÊNCIAS

- Avis, D., Bhattacharya, B.K. (1983) - "Algorithms for computing d -dimensional Voronoi Diagrams and their duals", Advances in Computing Research, vol.1, p.159-180.
- Baliga, B. R., Patankar, S. V. (1988) - "Handbook of Numerical Heat Transfer", p.421-461, ed. John Wiley & Sons, Inc.
- Dirichlet, G.L. (1850) - Z Reine Angew Math, vol.40, p.216.
- Gilbert, E.N. (1962) - "Random subdivisions of space into crystals", Ann Math Stat, vol.33, p.958-972.
- Palagi, C.L. (1992) - "Generation and application of Voronoi grid to model flow heterogeneous reservoirs".
- Schneider, G. E. (1988) - "Handbook of Numerical Heat Transfer", p.379-460, ed. John Wiley & Sons, Inc.
- Shamos, M.I., Hoey, D. (1975) - "Closest-point problems", Proc. 16th Annu IEEE Symp on the Foundation of Computer Science, p.151-162.
- Sugihara, K, Iri, M. (1992) - "Construction of the Voronoi Diagram for One Million Generators in Single-Precision Arithmetic", Proc of the IEEE, vol.80, p.1471-1484.
- Rhynsburger, D. (1973) - "Analytic delineation of Thiessen polygons", Geograph Anal 5, p.133-144.
- Thiessen, A.H. (1911) - "Precipitation averages for large areas", Monthly Weather Review, vol.39, p.1082-1084.
- Voronoi, G. (1908) - "Nouvelle applications des paramètres continus à la théorie des formes quadratiques. Deuxième Mémoire, Recherches sur les polyèdres primitifs", Z reine Angew Math, vol.134, p.198-287.
- Wigner, E., Seitz, F. (1933) - "On the constitution of metallic sodium", Phys Rev, vol.34, p.804-810.

XIV Congresso Ibero Latino-Americano de Métodos Computacionais em Engenharia

Instituto de Pesquisas Tecnológicas
São Paulo - SP - Brasil
1 a 3 de dezembro de 1993

FORMULAÇÕES DE ELEMENTOS FINITOS PARA SIMULAÇÃO DE INJEÇÃO DE TRAÇADORES EM RESERVATÓRIOS DE PETRÓLEO

Abimael Fernando Dourado Loula
Eduardo Lúcio Mendes Garcia
Laboratório Nacional de Computação Científica / CNPq
Rua Lauro Muller 455, CEP 22290-160, Rio de Janeiro, RJ
Luiz Landau
Fernando L. B. Ribeiro
Programa de Engenharia Civil - COPPE / UFRJ
Centro de Tecnologia - Sala B100, CEP 21945-970, Rio de Janeiro, RJ

SUMÁRIO

Formulações de elementos finitos são empregadas na solução numérica do sistema de equações diferenciais parciais dependente do tempo que representa o escoamento de fluidos miscíveis visando a simulação de processos de injeção de traçadores em reservatórios de petróleo. Devido a presença de camada limite e fortes gradientes de concentração do traçador discute-se a utilização de métodos adaptativos tipo h na solução deste problema.

1. INTRODUÇÃO

Durante as etapas de exploração e produção de petróleo diversos métodos de caracterização do reservatório podem ser utilizados. Eles vão se complementar com o objetivo de determinar a ligação entre poços, caminhos preferenciais, barreiras de fluxo, comportamento de fluidos injetados e determinação da saturação do óleo residual permitindo que se tenha um mapa realístico do reservatório a fim de tornar eficiente as diversas técnicas de recuperação do petróleo.

Traçadores são materiais químicos ou radioativos adicionados a um fluido em pequenas quantidades a fim de acompanhar seu movimento sem alterar suas propriedades. Neste contexto o tempo de chegada e a forma da curva de concentração do traçador no efluente (poço de produção) são dados importantes no processo de caracterização de reservatórios.

O problema matemático consiste na solução de um sistema de equações diferenciais composto pela equação da pressão,

$$\operatorname{div} \mathbf{u} = f \quad \text{em } \Omega \times [0, T] \quad (1.1)$$