Ciências e Tecnologias Espaciais

Sensores e Atuadores Espaciais

Métodos Numéricos e Aplicações em Clusters I – Básico

Lista de Exercícios 5

Professor: Angelo Passaro
Aluno: Lucas Kriesel Sperotto

31 de Maio de 2012

1 – Resolver equação de Laplace para o problema de placa quadrada com condições de contorno C_1 = 100, C_2 = 50, C_3 = 50, C_4 = 75, através do método "Fixed Random Walk", baseando-se nos capítulos 2 e 4 da referência [1].

Modelagem Matemática:

Como exposto em [1], a aplicação do método "Fixed Random Walk" (FRW) envolve três passos:

1) Obter as probabilidades transitórias resultantes da equação de diferenças finitas equivalente a equação diferencial que descreve o problema.
2) Usar números aleatórios juntamente com as probabilidades transitórias para direcionar vários caminhos aleatórios na região de solução e registrar o potencial final de cada percurso aleatório.
3) Encontrar a média estatística dos potenciais registrados no passo 2.

Para o passo 1, a modelagem matemática da equação e Laplace deve-se desenvolver a equação de forma semelhante ao método de diferenças finitas. Entretanto a grande diferença reside nos termos probabilísticos inerentes aos métodos de Monte Carlo. Tomando a equação de Laplace para material homogêneo:

$$\nabla^2 u = 0 \Longrightarrow \frac{\partial^2 \mathrm{u}}{\partial \mathrm{x}^2} + \frac{\partial^2 \mathrm{u}}{\partial \mathrm{y}^2} = 0 \ (1)$$

Substituindo as derivadas de diferenças finitas considerando h_xe h_y iguais:

$$\frac{\mathrm{u}_1 + \mathrm{u}_3 + \mathrm{u}_2 + \mathrm{u}_4}{4} = \mathrm{u}_P \quad (2)$$

Substituindo pelas contribuições probabilísticas de cada contribuição:

$$\mathrm{p}_1\mathrm{u}_1 + \mathrm{p}_2\mathrm{u}_2 + \mathrm{p}_3\mathrm{u}_3 + \mathrm{p}_4\mathrm{u}_4 = \mathrm{u}_P \quad (3)$$

Onde $\mathrm{p}_1 = \mathrm{p}_2 = \mathrm{p}_3 = \mathrm{p}_4 = 1/4$. No método FRW a partícula de avaliação de u_P move-se nas quatro direções com probabilidade ¼. Para definir o caminho da partícula no domínio [1] faz uso de um conjunto de valores aleatórios e define quatro subconjuntos com tamanhos idênticos onde cada subconjunto de valores é responsável por deslocar a partícula em uma direção.

No momento que a partícula alcança uma aresta com condição de contorno, a probabilidade é computada juntamente com o valor da condição de contorno para contribuir no valor de u_P. Para avaliar corretamente o valor de u_P, várias partículas devem ser lançadas do ponto e suas contribuições adicionadas em u_P.

Pode-se reescrever a equação anterior como:

$$\mathrm{u}_P = \frac{1}{N}\sum_{i=1}^{N} C_i \quad (4)$$

onde N é o numero total partículas e C_i é o valor da condição de contorno alcançada pela partícula.

Uma outra forma de desenvolver o raciocínio é pensar na seguinte equação:

$$\mathrm{p}_1\mathrm{C}_1 + \mathrm{p}_2\mathrm{C}_2 + \mathrm{p}_3\mathrm{C}_3 + \mathrm{p}_4\mathrm{C}_4 + \cdots + \mathrm{p}_i\mathrm{C}_i = \mathrm{u}_P \ (5)$$

onde p_i é a probabilidade da partícula atingir a condição de contorno e C_i é o valor da condição de contorno alcançada pela partícula.

Esta equação pode ser reescrita na forma de um somatório:

$$u_P = \frac{1}{N}\sum_{i=1}^{m} N_i C_i \,, (6)$$

onde N é o numero total de partículas, m é o numero de faces com condição de contorno e N_i é o numero de passos para alcançar a face i. Note que N_i/N é a probabilidade de a partícula alcançar a borda com condição de contorno. Esta equação é idêntica a equação (4).

Modelagem computacional:

A modelagem em JAVA requer o uso de classes, entretanto o diagrama da Figura 1 mostra claramente uma implementação puramente procedural já que nenhuma classe apresentada se configura como um objeto. Este tipo de implementação se justifica pela rápida codificação do método e por nenhuma exigência do exercício sobre uma modelagem orientada a objetos.

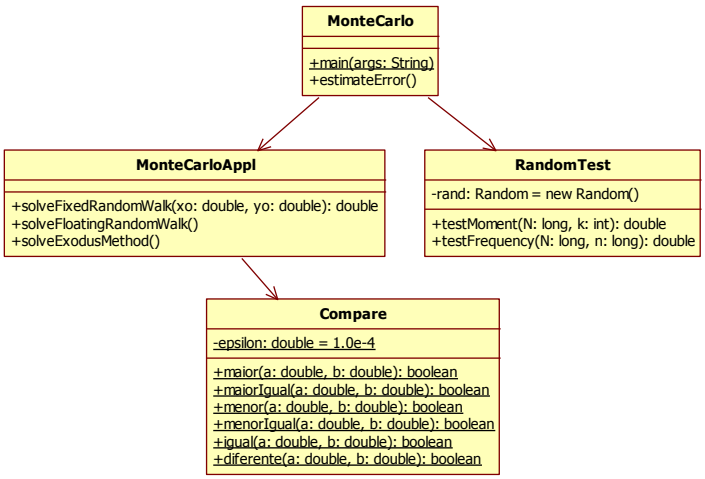

Figura 1 - Diagrama de classes.

A classe “MonteCarlo” é a classe principal, nela é gerada a malha de pontos onde a solução será calculada. Os pontos gerados são uniformemente espaçados, entretanto o método não exige essa distribuição já que o calculo do da aproximação em um ponto independe do calculo dos demais pontos.

Na classe “MonteCarloAppl” está o método que encontra o potencial para cada pondo da malha. Uma classe para escrita dos arquivos de resultados foi desenvolvida, entretanto pela semelhança com as classes já mostradas em exercícios anteriores a mesma foi suprimida do diagrama.

A classe “Compare” é uma classe utilitária usada para comparação de variáveis do tipo *double*. E por fim, a classe “RandomTest” possui métodos para testar os números aleatórios gerados pela biblioteca JAVA.

Resultados:

Para a geração de números aleatórios a biblioteca JAVA fornece a classe “Random” que gera números aleatórios usando como semente a hora da maquina. Primeiramente foi levantado medidas da uniformidade dos números gerados, para mensurar a uniformidade [1] comenta que uma das formas é executar o teste de “momento”. Na tabela 1 encontra-se o erro percentual do momento para $k = 1$ e diferentes valores de N calculado pela expressão (7).

Como esperado pela definição do Momento, o erro diminui à medida que a quantidade de números aleatórios usados aumenta.

$$\frac{1}{N}\sum_{i=1}^{N} U_1^k = \frac{1}{k+1} \quad (7)$$

Tabela 1 - Erro obtido para a medida de Uniformidade (Momento)

Números Gerados	Erro (%)
125	5.114
625	6.293
3125	1.145
15625	0.096

A distribuição das condições de contorno na placa quadrada pode ser visto na figura 2. Executado simulação para diferentes valores de N e cada simulação foi executada 10 vezes para se tomar a média das execuções como resultado final. Este passo foi usado, pois foram percebidas diferenças significativas dos valores de cada ponto entre simulações consecutivas. Na figura 3 tem-se a distribuição da temperatura para $N = 15625$.

Note que ao comparar a solução por FRW (Figura 3) com a solução por diferenças finitas da figura 4 qualitativamente há uma boa representação da distribuição de calor na placa.

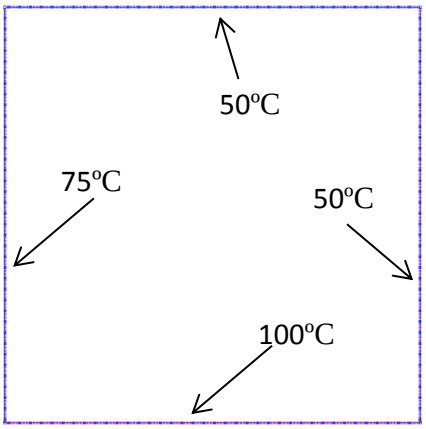

Figura 2 - Detalhes do Teste.

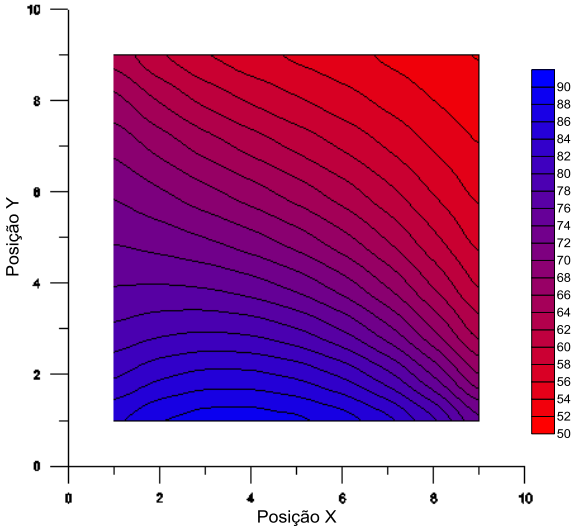

Figura 3 - Resultado Obtido com "FRW".

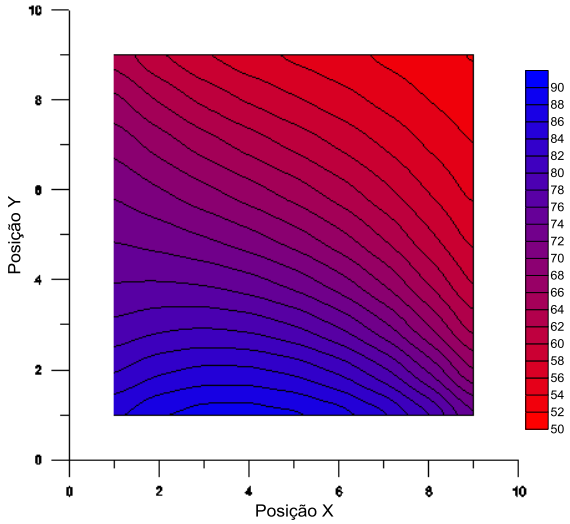

Figura 4 - Resultado Obtido com Diferenças Finitas.

Tabela 2 - Comparação entre o erro em relação à referência e o erro estimado.

Numero de passos	Erro máximo (%)	Erro máximo estimado (%)
125	1.55	2.052
625	1.05	0.951
3125	0.53	0.364
15625	0.48	0.207

De forma a avaliar quantitativamente os resultados dois testes foram executados. Primeiro foi calculada a estimativa de erro pela expressão (8) com intervalo de confiança de 95%, e em segunda instância calculado o erro ponto a ponto em relação à solução por diferenças finitas.

$$\varepsilon = \frac{St_{a/2;N-1}}{\sqrt{N}} \quad (8)$$

onde S é a dispersão e $t_{a/2;N-1}$ é o valor na tabela de distribuição t de Student para a confiança usada.

Na tabela 2 encontra-se o resultado para os diferentes valores de N usados, o erro máximo percentual máximo obtido em relação à solução por Diferenças Finitas e o máximo valor da estimativa de erro.

Note que com poucas partículas, o erro estimado é praticamente o dobro do erro obtido, já para um numero maior de partículas o erro estimado é praticamente a metade do erro obtido. Pode-se atribuir essa diferença ao intervalo de confiança usado, acredito que este valor deva ser diferente para cada numero de partículas usadas, já que para poucas partículas deve-se ter uma confiança menor e para muitas partículas uma confiança maior.

Conclusões:

O método de "Fixed Random Walk" é um método bastante interessante do ponto de vista da curiosidade. Apesar do erro em relação à solução por diferenças finitas ser pequeno, o método FRW é bastante lento se comparado a outras abordagens numéricas já estudadas na disciplina. Outro fator que considero negativo é a instabilidade da solução e ainda ter de executar a simulação mais de uma vez para retirar a média dos valores multiplica o tempo de execução que já não é pequeno dependendo do numero de partículas usadas.

Claro que deve haver aplicações que necessitem deste tipo de método e que não possuam solução numérica simples. Para finalizar o trabalho gostaria de citar uma frase de Albert Einstein "Deus não joga dados..." por qual motivo os físicos o fazem?

Referências:

[1] SADIKU, M. N. O. Monte Carlo Methods for Electromagnetics. United States of America: CRC Press. 2009.

[2] http://pt.wikipedia.org/wiki/Distribui%C3%A7%C3%A3o_t_de_Student, acessado em 29 de Maio de 2012.