

[moodle.lasdpc.icmc.usp.br](http://moodle.lasdpc.icmc.usp.br)

# Especificação e Entrega do Primeiro Trabalho

## Especificação do Primeiro Trabalho

### Jacobi-Richardson

### Especificação do Problema

Desenvolva uma aplicação seqüencial e depois uma parela utilizando Pthreads que resolva um sistema linear ( $Ax = B$ ), segundo o método iterativo Jacobi-Richardson (também conhecido como Gauss-Jacobi). Para a execução, a aplicação deve receber um arquivo de entrada contendo as configurações e os elementos das [matrizes](#) A e B. Os arquivos de entrada ([matrizes](#)) estão disponibilizados abaixo da especificação. Um arquivo em PDF mostrando o funcionamento do método, também está disponível para auxiliá-los no desenvolvimento do trabalho.

Exemplo do arquivo de entrada:

3 -> ordem da matriz (J\_ORDER)

2 -> fila para ser avaliada (J\_ROW\_TEST)

0.001 -> erro permitido (J\_ERROR)

20000 -> número máximo de iterações (J\_ITE\_MAX)

4 2 1 -|

1 3 1 -|-> matriz A (MA)

2 3 6 -|

7 -|

-8 -|-> matriz B (MB)

6 -|

Em anexo encontra-se um exemplo da utilização do algoritmo.

## Obrigatório

A saída dos programas deve mostrar:

- Número de iterações efetuadas pelos programas
- Devera ser mostrado o valor aproximado resultado do processo. Para isso deve ser utilizando a fila 2 da matriz A e comparar com o valor da matriz B na fila 2

## Exemplo de saída

-----  
Iterations: 1607

RowTest: 2 => [36.924484] =? 37.000000

-----

1607 = número de iterações

2 = índice da fila da matriz A para efetuar a comprovação, esse índice pode ser: [0, 1, 2, ...n-1], onde n = ordem da matriz

$$36.924484 = 2(X_0) + 3(X_1) + 6(X_2)$$

$$MA[2,0] = 2, MA[2,1] = 3, MA[2,2] = 6$$

$X_i$  = elementos resultado da operação

37.000000 = elemento da matriz B,  $MB[2]$

### O critério de parada:

- Atingir o erro (**J\_ERROR**) ou
- Atingir o número de iterações máximo (**J\_ITE\_MAX**)
- **Mostrar na tela somente a média da 10 execuções**, para que o professor possa verificar o resultado.
- Enviar para um arquivo todas as saídas geradas da execução

### Para os Programas

- Fazer uma versão sequencial utilizando C

- Fazer uma versão paralela utilizando Posix Pthreads
- **Tanto a versão sequencial quanto a versão paralela devem ser executadas considerando o mesmo hardware.**
  - **Se o grupo considerar a utilização de um nó do cluster Cosmos, ambas as versões precisam ser executadas no mesmo nó. O mesmo vale caso a escolha seja seu notebook.**
- Fazer um comparativo entre os algoritmos e entre as versões sequenciais e paralelas de cada um deles (Calcular o Speedup).
- Realizar no **mínimo 10 execuções** e calcular o tempo de execução considerando, **média e desvio padrão**.

### **Para o Relatório**

- O relatório deverá ser entregue **SOMENTE** no formato **PDF** com o seguinte nome: **Relatorio1-grupoX-turmaY**, onde X é o número do grupo e Y representa a turma(A, B ou PosGrad).
- Um relatório apresentando uma introdução sobre os algoritmos, os resultados obtidos e as soluções. O relatório deve ter no mínimo 4 páginas e no máximo 8 páginas desconsiderando capas, índices e bibliografia. No máximo 1 página explicando cada algoritmo e seus comentários.

- Discuta as soluções, as dificuldades, os resultados obtidos, o **hardware utilizado**, a metodologia de execução dos experimentos, etc.
- O relatório deve ser enviado via Moodle conforme combinado no primeiro dia de aula
- O relatório deve apresentar a forma de execução dos codigos (README)
- Apresentar resultados somente em gráficos ou tabelas:
  - Tempo de execução seqüencial e tempo de execução paralelo
  - Speedup

## **Avaliação**

### **A avaliação deve considerar:**

- Qualidade do relatório e descrição dos resultados com base na execução dos códigos.
- Saída correta dos códigos bem como a qualidade do mesmo (comentários, indentação, menus de utilização, etc).

## **Redmine**

- **Em breve!!!**

## **Prazos**

## **Relatório e Códigos**

**06/09/15 - 23:55h (Turmas A, B e PosGrad)**