

Laboratório de Programação Paralela

Trabalho prático - apresentação e proposta

Arthur De Oliveira Paiva – (116031056)

Lucio Henrik Amorim Reis – (116031051)

Rafael Duarte Campbell – (117031036)

Problema escolhido

Para este trabalho prático, pretende-se implementar – de forma linear e paralela – uma aplicação para **resolução de sistemas de equações lineares por eliminação gaussiana com pivoteamento parcial**. Além da grande aplicabilidade, a resolução de sistemas lineares é uma tarefa bastante paralelizável, ainda mais pelo método de eliminação gaussiana.

Sistemas de equações lineares

Equações lineares são, sobretudo, equações polinomiais que podem ser escritas na forma $a_1X_1 + a_2X_2 + \dots + a_nX_n = b$, onde a_1, a_2, \dots, a_n e b são constantes e X_1, X_2, \dots, X_n são incógnitas. Um *sistema de equações lineares* é, essencialmente, um conjunto de equações lineares que diz respeito ao mesmo conjunto de incógnitas, sendo normalmente escrito como:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + \dots + a_{3n}x_n = b_3 \end{cases} \quad (1)$$

É comum, ainda, que o sistema seja expresso na forma matricial $AX = B$, onde:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad (2)$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (3)$$

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (4)$$

A representação matricial é ainda mais valiosa para o método de eliminação gaussiana, dado que as operações são aplicadas exclusivamente à matriz A . Isto posto, usaremos a representação matricial tanto para exemplificar o método quanto para implementar o método.

Resolução do sistema

O método de *eliminação gaussiana* tem três etapas, sendo elas:

1. Obter uma matriz aumentada $[A|b]$, onde o sistema de equação seja $AX = b$.
2. Obter uma matriz equivalente $[A'|b']$, onde A' seja triangular superior.
3. Resolver o sistema de equações $A'X = b'$ por substituição regressiva.

A etapa mais expressiva – e paralelizável – do processo é a segunda, que consiste em aplicar sucessivas operações elementares até que se obtenha uma matriz triangular superior equivalente. Assumindo uma constante não-nula a e um sistema de equações lineares $\{L_1, L_2, \dots, L_m\}$, as premissas elementares que norteiam o processo são:

- $L'_1 \equiv L_1 \times a$
- $L'_1 \equiv L_1 - L_2$

Como forma de zerar as constantes abaixo da diagonal principal, executa-se:

Para cada coluna i , faça:

Para cada linha L_k , se $k > i$, faça:

$$L'_k \leftarrow L_k - (L_i \times (a_{ki}/a_{ii}))$$

A série de operações que se faz para cada coluna i tem como objetivo zerar todas as constante abaixo de a_{ii} , chamado **pivô**. Entretanto, este método depende que a_{ii} seja não-nulo – e expressivo – para todas as k linhas da iteração, como forma de evitar falhas no cálculo. Uma forma de otimizar o processo – e contornar as falhas citadas – é escolher o melhor pivô possível, ou seja, escolher o valor mais expressivo da coluna i ; a esse processo, damos o nome de *pivoteamento parcial*.

A escolha do pivô ocorre a cada i -ésima execução, onde deve-se executar:

$p = i$ //define-se pivô como i

Para cada linha L_j , se $j > i$, faça:

Se $a_{ji} > a_{pi}$, faça:

$p = j$ //define-se um novo pivô

Se $p \neq i$, faça:

$L_i \leftrightarrow L_p$ //permutar linhas

Caso um sistema consistente e uma matriz densa, o algoritmo apresentado obterá uma matriz triangular superior A' equivalente. Em seguida, deve-se executar o procedimento de substituição regressiva para obter os valores das incógnitas.

Paralelização

Das três etapas da resolução do sistema, apenas a segunda é paralelizável. Isto decorre do fato de que a primeira é inerente à representação do dado – neste caso, representação por uma estrutura matricial – e a terceira, de natureza sequencial, tem substituições que dependem da operação anterior. Dito isso, o trabalho de paralelização foca na segunda etapa, que é também a mais longa e expressiva do método. A segunda etapa, ainda sim, tem duas subetapas de natureza distinta: busca pelo pivô e o cálculo da linha L'_k .

Busca pelo pivô

Neste caso, basta agrupar todos os valores de uma coluna i e particioná-los entre os processos. Cada processo realizará uma comparação sequencial, buscando o maior valor; em seguida, todos os maiores valores são reagrupados e busca-se, entre estes, o maior – e não nulo.

Cálculo da linha L'_k

Uma vez escolhido o pivô, cada processo receberá a linha pivô L_i e um conjunto de linhas, devendo aplicar a fórmula $L'_k \leftarrow L_k - (L_i \times (a_{ki}/a_{ii}))$ a cada uma. A forma como é feita a distribuição das linhas não é um problema central da paralelização, mas é passível de otimização. Neste projeto, foi escolhido o modelo *cyclic striped*, que visa distribuir as linhas de cima para baixo alternando entre os processos. [1]

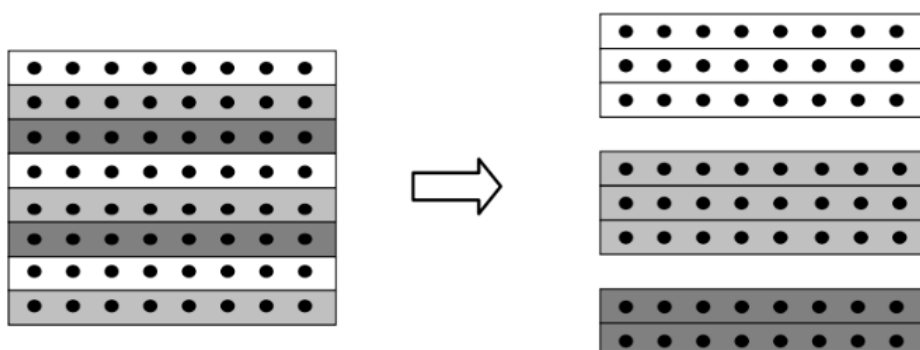


Figura 1: Representação da divisão entre 3 processos. [1]

Tecnologias e implementações

Para este projeto, serão implementadas quatro versões deste algoritmo. A primeira versão é sequencial e regular, será implementada em C e seguirá o algoritmo descrito na seção *Resolução do sistema*. As duas seguintes são paralelas e seguirão o modelo de paralelização descrito na seção anterior, devendo ser implementadas nos padrões MPI (*Message Passing Interface*) e OpenMP (*Open Multi-Processing*). A última versão também é paralela e seguirá as mesmas técnicas descritas, mas será implementada em C usando a biblioteca *pthread*. Ao fim, espera-se obter um comparativo de desempenho entre as quatro implementações, com testes para matrizes de diferentes tamanhos.

[1] *Parallel Methods for Solving Linear Equation Systems*. Disponível em <https://drive.google.com/file/d/1hm-itDI6WYPGkTsF3ZHpxTOhf5KElzBG/view>. Acessado em 17/10/2020.