# Постановка задачи

1) Реализовать последовательный алгоритм метода Якоби
2) Реализовать параллельный алгоритм численного решения СЛАУ с помощью MPI
3) Провести сравнение скорости работы двух алгоритмов

При большом числе неизвестных метод Гаусса становится весьма сложным в плане вычислительных и временных затрат. Поэтому иногда удобнее использовать приближенные (итерационные) численные методы, метод Якоби относится к таким.

В работе требуется решить систему линейных алгебраических уравнений вида:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

$$Ax = b$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1(n-1)} & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2(n-1)} & a_{2n} \\ \cdot & \cdot & \ldots & \cdot & \cdot \\ a_{(n-1)1} & a_{(n-1)2} & \ldots & a_{(n-1)(n-1)} & a_{(n-1)n} \\ a_{n1} & a_{n2} & \ldots & a_{n(n-1)} & a_{nn} \end{pmatrix} b = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ b_{n-1} \\ b_n \end{pmatrix} x = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ x_{n-1} \\ x_n \end{pmatrix}$$

При предположении, что диагональные коэффициенты ненулевые.

$$a_{ii} \neq 0 \qquad i = (1,2,..,n)$$

# Метод решения

Решив 1-ое уравнение системы относительно x1 получим:

$$x_1 = \frac{b_1 - (a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n)}{a_{11}}$$

2-ое - относительно x2 , n-ое - относительно xn

В итоге эквивалентная система, в которой диагональные элементы строки выражены через оставшиеся.

$$\begin{cases} x_1 = \dfrac{b_1-\left(a_{12}x_2+a_{13}x_3+\ldots+a_{1n}x_n\right)}{a_{11}} \\ x_2 = \dfrac{b_2-\left(a_{21}x_1+a_{23}x_3+\ldots+a_{2n}x_n\right)}{a_{22}} \\ \cdots \\ x_n = \dfrac{b_n-\left(a_{n1}x_1+a_{n2}x_2+\ldots+a_{n(n-1)}x_{n-1}\right)}{a_{nn}} \end{cases}$$

или

$$x = \alpha x + \beta$$

$$\alpha = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \cdots & -\frac{a_{2n}}{a_{22}} \\ & & \cdots & & \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \cdots & 0 \end{pmatrix} \qquad \beta = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ . \\ \frac{b_n}{a_{nn}} \end{pmatrix}$$

Далее вводится некоторое начальное приближение - вектор x(0)=[b1/a11, ... ,bi/aii, ... ,bn/ann], затем используя x(1) находится x(2).
Данный процесс называется итерационным, условием окончания является достижение заданной точности (система сходится и есть решение) или прерывание процесса. Процесс прерывается когда число итераций превышает заданное допустимое количество, при этом система не сходится либо заданное количество итераций не хватило для достижения требуемой точности.

Итерационный процесс. Верхний индекс в скобках - номер итерации.

$$\begin{cases} x_1^{(k+1)} = \dfrac{b_1-\left(a_{12}x_2^{(k)}+a_{13}x_3^{(k)}+\ldots+a_{1n}x_n^{(k)}\right)}{a_{11}} \\ x_2^{(k+1)} = \dfrac{b_2-\left(a_{21}x_1^{(k)}+a_{23}x_3^{(k)}+\ldots+a_{2n}x_n^{(k)}\right)}{a_{22}} \\ \cdots \\ x_n^{(k+1)} = \dfrac{b_n-\left(a_{n1}x_1^{(k)}+a_{n2}x_2^{(k)}+\ldots+a_{n(n-1)}x_{n-1}^{(k)}\right)}{a_{nn}} \end{cases}$$

$$x^{(k+1)} = \alpha x^{(k)} + \beta$$

$x^{(1)} = \alpha x^{(0)} + \beta$ - первое приближение

$x^{(2)} = \alpha x^{(1)} + \beta$ - второе приближение

$x^{(k+1)} = \alpha x^{(k)} + \beta$ - (k+1) - ое приближение

Если последовательность приближений (x(0),x(1),...,x(k+1),...) имеет предел

$$x = \lim_{k \to \infty} x^{(k)}$$

то этот предел является решением. k=1,2,3,...N-1. , N-1 - заданное количество итераций

*Достаточный признак сходимости метода Якоби:*
Если в системе выполняется диагональное преобладание, то метод Якоби сходится.

$$|a_{ii}| > \sum_{j=1,(j \neq i)}^{n} |a_{ij}| \qquad i = 1,2,..,n$$

Критерий окончания итераций при достижении требуемой точности имеет вид:

$$||x^{(k+1)} - x^{k}|| < \varepsilon$$

где ε - заданная точность вычисления

# Параллельная схема

При распараллеливании алгоритма предполагается, что размерность системы больше числа процессоров. Каждый процессор считает "свои" элементы вектора X=(x1,x2,x3,...,xn).

Перед началом выполнения метода на каждый процессор рассылаются необходимые данные:

1) размерность системы N

2) начальное приближение X0

3) строки матрицы A

4) элементы вектора свободных членов b.

После получения необходимой информации каждый процессор будет вычислять соответствующие компоненты вектора X и передавать их главному процессору. Главный процессор при получении очередного приближения решения Xk должен сравнить его с предыдущим приближением Xk-1. Если норма разности этих векторов:

||Xk – Xk-1|| = max {|Xk,i – Xk,i-1|, i = 1,1,…n}

окажется меньше или равной заданной точности (ε), то вычисления закончатся. В противном случае вектор Xk поэлементно будет разослан всем процессам и будет вычисляться очередное приближение решения.