Parallel Processing

Parallel Block Algorithms in Linear Algebra

Felicja Okulicka-Dłużewska

`F.Okulicka@mini.pw.edu.pl`

Warsaw University of Technology

Faculty of Mathematics and Information Science

Go to full-screen mode now by hitting CTRL-L

Overview

- **Introduction**
- Block Gauss elimination
- Block backward substitution
- Block LU
- Block cyclic reduction
- Block iterative methods

Introduction

Parallel programming opens new possibility for calculating large problems. There are two main advantages:

- big problems can be calculated in the memory due to the sharing of the data among different machines memory
- the speed-up for long calculation due to the sharing of the computing process allows to apply more sophisticated and adequate models.

Introduction

The large matrixes can be split into blocks and put into separate memories of the distributed computers. The parallel calculations are reached due to the matrix operations on separate blocks. The standard numerical algorithms should be rebuilt for the block version.

Overview

- Introduction
- **Block Gaussian elimination**
- Block backward substitution
- Block LU
- Block cyclic reduction
- Block iterative methods

Block Gaussian elimination

Consider the system of linear equations

$$AX = B$$

We divide our matrix A into $m_i \times m_j$ blocks, where each A_{ij} is the matrix on $n_i \times n_j$ dimensions.

Block Gaussian elimination

$$\begin{pmatrix} A_{11} & ... & A_{1,k} & ... & A_{1,m} \\ ... & ... & ... & ... & ... \\ A_{k,1} & ... & A_{k,k} & ... & A_{k,m} \\ ... & ... & ... & ... & ... \\ A_{m,1} & ... & A_{m,k} & ... & A_{m,m} \end{pmatrix} \bullet \begin{pmatrix} X_1 \\ ... \\ X_k \\ ... \\ X_m \end{pmatrix} = \begin{pmatrix} B_1 \\ ... \\ B_k \\ ... \\ B_m \end{pmatrix}$$

The algorithm works with the assumption that $m_i = m_j = m$. X and B are divided on subvectors

$$X = \begin{pmatrix} X_1 & X_2 & ... & X_k & ... & X_{m-1} & X_m \end{pmatrix}^T$$
$$B = \begin{pmatrix} B_1 & B_2 & ... & B_k & ... & B_{m-1} & B_m \end{pmatrix}^T$$

B_i and X_i are vectors, for i=1,...,m.

Block Gaussian elimination

The method described by Demmel is used. The well-known Gaussian elimination is rewritten using array operations. The block Gaussian elimination on block array is done. In general, after the $(k-1)-th$ stage of elimination the system of linear equations can be written as follows

$$A^{(k-1)}X = B^{(k-1)}$$

Block Gauss alg.

At the $k-th$ stage X_k is eliminated from the block equations $= k+1, k+2, ..., m$, by the subtraction the block equation k multiplied by $A_{ik}^{(k-1)} {A_{kk}^{(k-1)}}^{-1}$ from each of the equations. The changed block elements are given by the following matrix calculations

$$A_{ij}^{(k)} = A_{ij}^{(k-1)} - A_{ik}^{(k-1)} (A_{kk}^{(k-1)})^{-1} A_{ik}^{(k-1)}$$

$$B_{i}^{(k)} = B_{i}^{(k-1)} - B_{ik}^{(k-1)} (B_{kk}^{(k-1)})^{-1} B_{k}^{(k-1)}$$

for $i, j = k+1, k+2, ..., m$.

Block Gaussian elimination

After $m-1$ elimination stages

$$A^{(m-1)}X = B^{(m-1)}$$

where $A^{(m-1)}$ is the block upper triangular matrix.

$A^{(n-1)}$

$$\begin{pmatrix} A_{11} & ... & A_{1,k-1} & A_{1,k} & ... & A_{1,n-1} & A_{1,n} \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & ... & A^{(k-2)}_{k-1,k-1} & A^{(k-2)}_{k-1,k} & ... & A^{(k-2)}_{k-1,n-1} & A^{(k-2)}_{k-1,n} \\ 0 & ... & 0 & A^{(k-1)}_{k,k} & ... & A^{(k-1)}_{k,n-1} & A^{(k-1)}_{k,n} \\ 0 & ... & 0 & 0 & ... & A^{(k)}_{k+1,n-1} & A^{(k)}_{k+1,n} \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & ... & 0 & ... & 0 & A^{(n-2)}_{n-1,n-1} & A^{(n-2)}_{n-1,n} \\ 0 & ... & 0 & 0 & ... & 0 & A^{(n-1)}_{n,n} \end{pmatrix}$$

Overview

- Introduction
- Block Gauss elimination
- **Block backward substitution**
- Block LU
- Block cyclic reduction
- Block iterative methods

Backward subs.

To solve the block upper triangular system of equation

$$A^{(m-1)}X = B^{(m-1)}$$

backward substitution is calculated by matrix operations:

$$X_n = (A_{nn}^{(n-1)})^{-1} \cdot B_n^{(n-1)}$$

$$X_i = (A_{ii}^{(i-1)})^{-1}(B_i^{(i-1)} - \sum_{j=i+1}^{m} A_{ij}^{(i-1)} X_j)$$

Example

$$\begin{pmatrix} A_{11} & ... & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & ... & A_{kk} & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & ... & 0 & ... & A_{n,n} \end{pmatrix} \bullet \begin{pmatrix} X_1 \\ ... \\ X_k \\ ... \\ X_n \end{pmatrix} = \begin{pmatrix} A_{11}X_1 \\ ... \\ A_{kk}X_k \\ ... \\ A_{nn}X_n \end{pmatrix}$$

Example

$$\begin{pmatrix} A_{11}X_1 \\ A_{22}X_2 \\ \dots \\ A_{kk}X_k \\ \dots \\ A_{n-1,n-1}X_{n-1} \\ A_{nn}X_n \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \\ \dots \\ B_k \\ \dots \\ B_{n-1} \\ B_n \end{pmatrix} - X_n \begin{pmatrix} A_{1,n} \\ A_{2,n} \\ \dots \\ A_{k,n} \\ \dots \\ A_{n-1,n} \end{pmatrix} - \dots - X_2 \, (\, A_{12} \,)$$

Step 1 Calculate X_n by solving the set of equations:

$$A_{nn} \cdot X_n = B_n$$

Example

Step 2

We calculate the new right side block vector:

$$\begin{pmatrix} B_1^{(1)} \\ B_2^{(1)} \\ \ldots \\ B_k^{(1)} \\ \ldots \\ B_{n-2}^{(1)} \\ B_{n-1}^{(1)} \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \\ \ldots \\ B_k \\ \ldots \\ B_{n-2} \\ B_{n-1} \end{pmatrix} - X_n \cdot \begin{pmatrix} A_{1,n} \\ A_{2,n} \\ \ldots \\ A_{k,n} \\ \ldots \\ A_{n-2,n} \\ A_{n-1,n} \end{pmatrix}$$

Example

$$\begin{pmatrix} A_{11}X_1 \\ A_{22}X_2 \\ \cdots \\ A_{kk}X_k \\ \cdots \\ A_{n-2,n-2}X_{n-2} \\ A_{n-1,n-1}X_{n-1} \end{pmatrix} = \begin{pmatrix} B_1^{(1)} \\ B_2^{(1)} \\ \cdots \\ B_k^{(1)} \\ \cdots \\ B_{n-1}^{(1)} \\ B_n^{(1)} \end{pmatrix} - X_{n-1} \begin{pmatrix} A_{1,n-1} \\ A_{2,n-1} \\ \cdots \\ A_{k,n-1} \\ \cdots \\ A_{n-2,n-2} \end{pmatrix} - \ldots - X_2 \left(A_{12} \right)$$

Calculate X_{n-1} by solving the set of equations:

$$A_{n-1,n-1} \cdot X_{n-1} = B_{n-1}^{(1)}$$

Example

Step 3

$$\begin{pmatrix} B_1^{(2)} \\ B_2^{(2)} \\ \dots \\ B_k^{(2)} \\ \dots \\ B_{n-2}^{(2)} \\ B_{n-2}^{(2)} \end{pmatrix} = \begin{pmatrix} B_1^{(1)} \\ B_2^{(1)} \\ \dots \\ B_k^{(1)} \\ \dots \\ B_{n-3}^{(1)} \\ B_{n-2}^{(1)} \end{pmatrix} - X_{n-1} \cdot \begin{pmatrix} A_{1n-1} \\ A_{2n-1} \\ \dots \\ A_{kn-1} \\ \dots \\ A_{n-2,n-1} \end{pmatrix}$$

Example

$$\begin{pmatrix} A_{11} & ... & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & ... & A_{kk} & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & ... & 0 & ... & A_{n-2,n-2} \end{pmatrix} \begin{pmatrix} X_1 \\ ... \\ X_k \\ ... \\ X_{n-2} \end{pmatrix} == \begin{pmatrix} A_{11}X_1 \\ ... \\ A_{kk}X_k \\ ... \\ A_{n-2,n-2}X_{n-2} \end{pmatrix}$$

Example

$$\begin{pmatrix} A_{11}X_1 \\ A_{22}X_2 \\ \dots \\ A_{kk}X_k \\ \dots \\ A_{n-2,n-2}X_{n-2} \end{pmatrix} = \begin{pmatrix} B_1^{(2)} \\ B_2^{(2)} \\ \dots \\ B_k^{(2)} \\ \dots \\ B_{n-2}^{(2)} \\ B_{n-2}^{(2)} \end{pmatrix} - X_{n-2} \begin{pmatrix} A_{1n-2} \\ A_{2n-2} \\ \dots \\ \dots \\ A_{n-2,n-2} \end{pmatrix} - \dots\dots - X_2 A_{12}$$

Calculate X_{n-2} by solving the set of equations:

$$A_{n-2,n-2} \cdot X_{n-2} = B_{n-2}^{(2)}$$

Example

Step $(n-k)$

$$\begin{pmatrix} A_{11} & 0 & \dots & 0 \\ 0 & A_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & A_{kk} \end{pmatrix} \cdot \begin{pmatrix} X_1 \\ X_2 \\ \dots \\ X_k \end{pmatrix} = \begin{pmatrix} A_{11}X_1 \\ A_{22}X_2 \\ \dots \\ A_{kk}X_k \end{pmatrix}$$

Example

$$\begin{pmatrix} A_{11}X_1 \\ A_{22}X_2 \\ \dots \\ A_{k-1,k-1}X_{k-1} \\ A_{kk}X_k \end{pmatrix} = \begin{pmatrix} B_1^{(k-1)} \\ B_2^{(k-1)} \\ \dots \\ B_k^{(k-1)} \end{pmatrix} - B_k \cdot \begin{pmatrix} A_{1k-1} \\ A_{2k-1} \\ \dots \\ A_{kk-1} \end{pmatrix} - \dots - X_2 \cdot A_{12}$$

Calculate X_k by solving the set of equations:

$$A_{k,k} \cdot X_k = B_k^{(k-1)}$$

Example

Step (n)

Calculate X_1 by solving the set of equations:

$$A_{11} \cdot X_1 = B_1^{(n-1)}$$

Overview

- Introduction
- Block Gauss elimination
- Block backward substitution
- **Block LU**
- Block cyclic reduction
- Block iterative methods

Block LU

We divide our matrix A into m blocks, where each A_{ij} is the matrix on $n_i \times n_j$ dimensions.

$$\begin{pmatrix} A_{11} & \dots & A_{1,k} & \dots & A_{1,m} & \\ \dots & \dots & \dots & \dots & \dots & \\ A_{k,1} & \dots & A_{k,k} & \dots & A_{k,m} & \\ \dots & \dots & \dots & \dots & \dots & \dots \\ A_{m,1} & \dots & A_{m,k} & \dots & A_{m,m} & \end{pmatrix} \begin{pmatrix} X_1 \\ \dots \\ X_k \\ \dots \\ X_m \end{pmatrix} = \begin{pmatrix} B_1 \\ \dots \\ B_k \\ \dots \\ B_m \end{pmatrix}$$

Block LU

X and B are divided on the subvectors

$$X = \begin{pmatrix} X_1 & X_2 & ... & X_k & ... & X_{m-1} & X_m \end{pmatrix}^T$$
$$B = \begin{pmatrix} B_1 & B_2 & ... & B_k & ... & B_{m-1} & B_m \end{pmatrix}^T$$

B_i and X_i , for i=1,...,m, are vectors.

Block LU

A is decomposed into a product of two matrices L and U where

$$L = \begin{pmatrix} I & 0 & \dots & 0 & \dots & 0 & 0 \\ L_{21} & I & \dots & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ L_{k,1} & L_{k,2} & \dots & I & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ L_{m-1,1} & L_{m-1,2} & \dots & L_{m-1,k} & \dots & I & 0 \\ L_{m,1} & L_{m,2} & \dots & L_{m,k} & \dots & L_{m,m-1} & I \end{pmatrix}$$

I is the unit matrix.

Block LU

$$U = \begin{pmatrix} U_{11} & U_{12} & \ldots & U_{1,k} & \ldots & U_{1,m-1} & U_{1,m} \\ 0 & U_{22} & \ldots & U_{2,k} & \ldots & U_{2,m-1} & U_{2,m} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & U_{k,k} & \ldots & U_{k,n-1} & U_{k,n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & \ldots & U_{m-1,m-1} & U_{m-1,m} \\ 0 & 0 & \ldots & 0 & \ldots & 0 & U_{m,m} \end{pmatrix}$$

Block LU

If A_{11} is non-singular we can define the $L^{(1)}$ and $U^{(1)}$ matrices:

$$L^{(1)} = \begin{pmatrix} I \\ A_{21}A_{11}^{-1} \\ A_{31}A_{11}^{-1} \\ \dots \\ A_{m-1,1}A_{11}^{-1} \\ A_{m,1}A_{11}^{-1} \end{pmatrix} \qquad U^{(1)} = \begin{pmatrix} A_{11} \\ A_{12} \\ A_{13} \\ \dots \\ A_{1,m-1} \\ A_{1,m} \end{pmatrix}$$

Block LU

$A^{(1)} = A - L^{(1)} {U^{(1)}}^T$ =

$$\begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} & \dots & A_{2,m} - A_{21}A_{11}^{-1}A_{1m} \\ \dots & \dots & \dots & \dots \\ 0 & A_{m-1,2} - A_{m-1,1}A_{11}^{-1}A_{12} & \dots & A_{m-1,n} - A_{mn-1,1}A_{11}^{-1}A_{1m} \\ 0 & A_{m,2} - A_{21}A_{11}^{-1}A_{12} & \dots & A_{m,m} - A_{m1}A_{11}^{-1}A_{1m} \end{pmatrix}$$

Block LU

After $k-1$ stages of such elimination we have the partially eliminated matrix:

$$A^{(k)} = A^{(k-1)} - L^{(k)} {U^{(k)}}^T =$$

$$\begin{pmatrix} 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & A_{kk}^{(k)} & \dots & A_{k,m}^{(k)} \\ \dots & \dots & \dots & \dots & & \\ 0 & \dots & 0 & A_{m-1,k}^{(k)} & \dots & A_{n-1,m}^{(k)} \\ 0 & \dots & 0 & A_{m,k}^{(k)} & \dots & A_{m,m}^{(k)} \end{pmatrix}$$

Block LU

Where $L_i^{(k)} = U_i^{(k)} =$ **0** for $i < k-1$,
$L_k^{(k)} = I$
and
$A_{ij}^{(k)} =$ **0** for $i < k$ or $j < k$.

0 denotes the matrix with all elements equal 0.

Block LU

Provided that $A_{kk}^{(k)}$ is non-singular we have:

$$L^{(k)} = \begin{pmatrix} 0 \\ \ldots \\ 0 \\ I \\ A_{k+1,k}^{(k-1)} {A_{kk}^{(k-1)}}^{-1} \\ A_{31}^{(k-1)} {A_{kk}^{(k-1)}}^{-1} \\ \ldots \\ A_{m-1,1}^{(k-1)} {A_{kk}^{(k-1)}}^{-1} \\ A_{m,1}^{(k-1)} {A_{kk}^{(k-1)}}^{-1} \end{pmatrix} \qquad U^{(k)} = \begin{pmatrix} 0 \\ \ldots \\ 0 \\ A_{kk}^{(k-1)} \\ A_{k,k+1}^{(k-1)} \\ A_{k,k+2}^{(k-1)} \\ \ldots \\ A_{k,m-1}^{(k-1)} \\ A_{k,m}^{(k-1)} \end{pmatrix}$$

Block LU

$$A_{ij}^{(k)} = A_{ij}^{(k-1)} - A_{ik}^{(k-1)} {A_{kk}^{(k-1)}}^{-1} A_{kj}^{(k-1)}$$

$i, j = k+1, k+2, ..., m$.

After m stages of this elimination procedure we find that:

$A = L^{(1)} {U^{(1)}}^T + A^{(1)}$

$= L^{(1)} {U^{(1)}}^T + L^{(12)} {U^{(2)}}^T + A^{(2)}$

... $= L^{(1)} {U^{(1)}}^T + L^{(12)} {U^{(2)}}^T + ... + L^{(m)} {U^{(m)}}^T + A^{(m)}$

where $A^{(m)}$ is the zero matrix.

Block LU

Hence

$$L = \begin{pmatrix} L^{(1)} & L^{(2)} & \dots & L^{(m)} \end{pmatrix}$$

$$U = \begin{pmatrix} {U^{(1)}}^T \\ {U^{(2)}}^T \\ \\ \dots \\ {U^{(m)}}^T \end{pmatrix}$$

L is a unit diagonal, lower triangular, block matrix, and U is upper triangular block matrix.

Block LU

Each process P_{st} waits for it's pre-processes.

for P_{st} where $s < t$ (matrix U) there are: $P_{s1}, P_{s2}, ...P_{s,s-1}$ and $P_{1t}, P_{2t}, ...P_{s-1,t}$.

$$
\begin{array}{ccccccccc}
U_{11} & * & ... & * & * & ... & \color{red}{U_{1t}} & ... & * \quad * \\
* & U_{22} & ... & * & * & ... & \color{red}{U_{2t}} & ... & * \quad * \\
* & * & ... & * & * & ... & & ... & * \quad * \\
* & * & ... & U_{s-1,s-1} & * & ... & \color{red}{U_{s-1,t}} & ... & * \quad * \\
\color{red}{L_{s1}} & \color{red}{L_{s2}} & ... & \color{red}{L_{s,s-1}} & U_{ss} & ... & \color{blue}{U_{st}} & ... & * \quad * \\
* & * & ... & * & * & ... & & ... & * \quad * \\
* & * & ... & * & * & ... & & ... & * \quad *
\end{array}
$$

Block LU

For P_{st} where $s > t$ (matrix L) there are: $P_{s1}, P_{s2}, ...P_{s,t-1}$ and $P_{1t}, P_{2t}, ...P_{t,t}$.

$$
\begin{array}{ccccccccccc}
U_{11} & * & ... & * & U_{1t} & ... & ... & * & ... & * & \\
* & U_{22} & ... & * & U_{2t} & ... & ... & * & ... & * & \\
* & * & ... & * & * & ... & ... & * & ... & * & \\
* & * & ... & * & U_{t,t} & ... & ... & * & ... & * & * \\
* & * & ... & * & * & ... & ... & * & ... & * & \\
* & * & ... & * & L_{s-1,t} & ... & U_{s-1,s-1} & * & ... & * & \\
L_{s1} & L_{s2} & ... & L_{s,t-1} & L_{st} & ... & L_{s,s-1} & U_{ss} & ... & * & \\
* & * & ... & * & * & ... & & * & ... & * &
\end{array}
$$

Overview

- Introduction
- Block Gauss elimination
- Block backward substitution
- Block LU
- **Block cyclic reduction**
- Block iterative methods

Cyclic reduction

BLOCK CYCLIC REDUCTION is the Divide and Conquer Algorithm:

- the problem size is reduced due to row operations
- matrix of the set of equations is divided on blocks which can be distributed in the set of memories.

Cyclic reduction

From the discretization of Poison's equation

$$\frac{\delta^2 u}{\delta x^2} + \frac{\delta^2 u}{\delta y^2} = f(x, y)$$

we receive the block tridiagonal matrix A.

Cyclic reduction

A is block tridiagonal:

$$A = \begin{pmatrix} D & F & & & & & \\ F & D & F & & & & \\ & & \cdots & \cdots & & & \\ & & F & D & F & & \\ & & & \cdots & \cdots & & \\ & & & & F & D & F \\ & & & & & F & D \end{pmatrix} \in R^{nq \times nq}$$

where $D, F \in R^{q \times q}$, $DF = FD$ and $n = 2^k - 1$.

$$D = \begin{pmatrix} 4 & -1 & \\ -1 & 4 & -1 \\ & -1 & 4 \end{pmatrix} \qquad F = \begin{pmatrix} -1 & & \\ & -1 & \\ & & -1 \end{pmatrix}$$

Cyclic reduction

Example

Let $n = 7 = 8 - 1$.

$$
\begin{array}{llllllll}
b_1 = & Dx_1 + & Fx_2 & & & & & \\
b_2 = & Fx_1 + & Dx_2 + & Fx_3 & & & & \\
b_3 = & & Fx_2 + & Dx_3 + & Fx_4 & & & \\
b_4 = & & & Fx_3 + & Dx_4 + Fx_5 & & & \\
b_5 = & & & & Fx_4 + Dx_5 + Fx_6 & & & \\
b_6 = & & & & & Fx_5 + Dx_6 + Fx_7 & & \\
b_7 = & & & & & & Fx_6 + Dx_7 &
\end{array}
$$

Cyclic reduction

For $i = 2, 4, 6$ the equations $i-1, i, i+1$ are multiplied by $F, -D, F$ respectively, and the equations $i-1, i+1$ are added. We eliminate the variables: x_1, x_3, x_5, x_7

$$\begin{aligned} (2F^2 - D^2)x_2 + F^2 x_4 \\ = F(b_1 + b_3) - Db_2 \\ F^2 x_2 + (2F^2 - D^2)x_4 + F^2 x_6 \\ = F(b_3 + b_5) - Db_4 \\ F^2 x_4 + (2F^2 - D^2)x_6 \\ = F(b_5 + b_7) - Db_6 \end{aligned}$$

Cyclic reduction

After reduction, assuming that:

$D^{(1)} = 2F^2 - D^2$

$F^{(1)} = F^2$

we have the tridiagonal system:

$$\begin{array}{ccccccl} D^{(1)}x_2 & + & F^{(1)}x_4 & & & = & b_2^{(1)} \\ F^{(1)}x_2 & + & D^{(1)}x_4 & + & F^{(1)}x_6 & = & b_4^{(1)} \\ & & F^{(1)}x_4 & + & D^{(1)}x_6 & = & b_6^{(1)} \end{array}$$

Cyclic reduction

By applying the same method of reduction we have: for $i = 2$ the equations $i-1, i, i+1$ are multiplied by $F^{(1)}, -D^{(1)}, F^{(1)}$, and the equations $i-1, i+1$ are added. We eliminate x_2 and x_6

$$D^{(2)} x_4 = b_4^{(2)}$$

Now we can calculate x_4, and then the others subvectors of the solution.

Cyclic reduction

Algorithm - part 1

`Solve`

$$D^{(k-1)} x_{2^{k-1}} = b^{(k-1)}_{2^{k-1}}$$

$D^{(0)} = D; F^{(0)} = F; b^{(0)} = b$;

Cyclic reduction

```
for (p = 1; p < k; p++)
{
F^(p) = [F^(p-1)]^2;
D^(p) = 2F^(p) - [D^(p-1)]^2;
r = 2^p;
for(j = 1; j < 2^(k-p); j++)
{
b_jr^(p) = F^(p-1)(b_(jr-r/2)^(p-1) + b_(jr+r/2)^(p-1) - D^(p-1) b_jr^(p-1))
}
}
```

for $(p = 1; p < k; p++)$

{

$F^{(p)} = [F^{(p-1)}]^2$;

$D^{(p)} = 2F^{(p)} - [D^{(p-1)}]^2$;

$r = 2^p$;

for$(j = 1; j < 2^{k-p}; j++)$

{

$b_{jr}^{(p)} = F^{(p-1)}(b_{jr-r/2}^{(p-1)} + b_{jr+r/2}^{(p-1)} - D^{(p-1)} b_{jr}^{(p-1)})$

}

}

Algorithm - part 2

for $(p=k-2; p\geq 0; p--)$
{
  $r = 2^p$;
  for$(j=1; j\leq 2^{k-p-1}; j++)$
  {
  if $(j=1)$ $c = b^{(p)}_{(2j-1)r} - F^{(p)}x_{2jr}$;
  else
    if $(j=2^{k-p+1})$ $c = b^{(p)}_{(2j-1)r} - F^{(p)}x_{(2j-2)r}$;
    else $c = b^{(p)}_{(2j-1)r} - F^{(p)}\big(x_{2jr} + x_{(2j-2)r}\big)$;
  }
  Solve $D^{(p)}x_{(2j-1)r} = c$
}

Overview

- Introduction
- Block Gauss elimination
- Block backward substitution
- Block LU
- Block cyclic reduction
- **Block iterative methods**

Iterative methods

The bottleneck of the direct methods is the access to the memory and the cash reuse. This disadvantage disappears in the case of iterative methods. The iterative methods base on vector-matrix operations, which can be easy parallelized.
Consider the system of linear equations

$$AX = B$$

We divide our matrix A into $m \times m$ blocks, where each A_{ij} is the matrix on $n_i \times n_j$ dimensions.

Iterative methods

Block Jacobi iteration:

The $(k+1)-th$ step of iteration in the block Jacobi method for equation can be written as

$$X_i^{(k+1)} = A_{ii}^{-1} * (B_i - \sum_{j=1, i \neq j}^{m} A_{ij} * X_j^{(k)})$$

Block Gauss-Seidel iteration:

$$\sum_{j=1}^{i} A_{ij} * X_i^{(k+1)} = (B_i - \sum_{j=i+1}^{m} A_{ij} * X_j^{(k)})$$

The block backward substitution should be used.

Iterative methods

Advantages of block iterative methods:

- Big matrix divided on blocks are put in the several memories
- The cost of the communication is lower than in the case of direct methods: only vectors are sent between processes
- Standard, efficient for currently used computers, libraries for matrix-vector and matrix-matrix computation can be applied

Disadvantages of block iterative methods:

- The condition coefficient of the matrix should be decreased to improve the convergence
- The preconditioners have to be used - the additional computation for preconditioning is needed

Thank you for your attention!

Any questions are welcome.