Frontal and multi-frontal solvers: orderings, elimination trees, refinement trees

Maciej Paszynski
Department of Computer Science
AGH University of Science and Technology, Krakow, Poland
maciej.paszynski@agh.edu.pl
http://www.ki.agh.edu.pl/en/research-groups/a2s
EXEMPLARY SIMPLE 1D NUMERICAL PROBLEM

Find temperature distribution \( R \ni x \mapsto u(x) \in R \) such that

\[
\frac{d^2 u}{dx^2} = 0 \text{ for } x \in [0, 1] \quad u(0) = 0 \quad u(1) = 20
\]

Finite difference discretization

\[
\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = 0 \text{ for } i = 1, \ldots, N - 1
\]

\[
u_N = 20
\]

\[
h = \frac{1}{N}
\]

\[
u_i = u(x_i) = u \left( \frac{i}{N} \right)
\]

\[
[0, 1] = \bigcup_{i=1}^{N} [x_{i-1}, x_i]
\]
TRIDIAGONAL MATRIX FOR EXEMPLARY 1D PROBLEM

\[
\begin{align*}
\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} &= 0 \text{ for } i = 1, \ldots, N - 1 \\
u_0 &= 0 \\
u_N &= 20
\end{align*}
\]

\[
\begin{bmatrix}
1 & 0 & \cdots & 0 & \cdots & 0 & 0 \\
\frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & 0 & \cdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & 0 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\
0 & 0 & \cdots & 0 & \cdots & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_i \\
\vdots \\
x_{N-1} \\
x_N
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
\vdots \\
0 \\
20
\end{bmatrix}
\]
FRONTAL SOLVER

Frontal matrix focuses on forward elimination of the first row
FRONTAL SOLVER

\[
\begin{bmatrix}
1 & 0 & \ldots & 0 \\
1/h^2 & -2/h^2 & 1/h^2 & 0 \\
0 & 1/h^2 & -2/h^2 & 1/h^2 \\
0 & \ldots & 1/h^2 & -2/h^2 \\
0 & \ldots & 0 & 1/h^2 \\
0 & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\ldots \\
x_i \\
\ldots \\
x_{N-1} \\
x_N
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
20
\end{bmatrix}
\]

Front matrix

2\textsuperscript{nd} row = 2\textsuperscript{nd} row \ - \ \frac{1}{h^2} \times 1\textsuperscript{st} row
FRONTAL SOLVER

\[
\begin{bmatrix}
1 & 0 & \ldots & 0 & \ldots & 0 & 0 \\
0 & -\frac{2}{h^2} & \frac{1}{h^2} & 0 & \ldots & 0 & 0 \\
0 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & 0 & \ldots & 0 \\
0 & \ldots & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & \ldots & 0 \\
0 & \ldots & 0 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & 0 \\
0 & 0 & \ldots & 0 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\
0 & 0 & \ldots & 0 & \ldots & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\ldots \\
x_i \\
x_{N-1} \\
x_N
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 \\
\ldots \\
0 \\
20
\end{bmatrix}
\]

Front matrix

\[2^{nd}\text{ row} = 2^{nd}\text{ row} - \frac{1}{h^2} \times 1^{st}\text{ row}\]
FRONTAL SOLVER

The first row has been eliminated

Frontral matrix focuses on the elimination of the second row
FRONTAL SOLVER

\[
\begin{bmatrix}
1 & 0 & \ldots & 0 & \ldots & 0 & 0 \\
0 & -\frac{2}{h^2} & \frac{1}{h^2} & 0 & \ldots & 0 & 0 \\
0 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & 0 & \ldots & 0 \\
0 & \ldots & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & \ldots & 0 \\
0 & \ldots & 0 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & 0 \\
0 & 0 & \ldots & 0 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\
0 & 0 & \ldots & 0 & \ldots & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\ldots \\
x_i \\
x_{N-1} \\
x_N
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
\ldots \\
0 \\
0 \\
20
\end{bmatrix}
\]

Front matrix

\[
\begin{bmatrix}
-\frac{2}{h^2} & \frac{1}{h^2} & 0 \\
\frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2}
\end{bmatrix}
\]

3^{rd} row = 3^{nd} row − [1/h^2] / [-2/h^2] * 2^{nd} row
The matrix equation is given by:

$$
\begin{bmatrix}
1 & 0 & \ldots & 0 & \ldots & 0 & 0 \\
0 & -\frac{2}{h^2} & \frac{1}{h^2} & 0 & \ldots & 0 & 0 \\
0 & 0 & -\frac{3}{2h^2} & \frac{1}{h^2} & 0 & \ldots & 0 \\
0 & \ldots & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & \ldots & 0 \\
0 & \ldots & 0 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & 0 \\
0 & 0 & \ldots & 0 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\
0 & 0 & \ldots & 0 & \ldots & 0 & 1 
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_i \\
\vdots \\
x_{N-1} \\
x_N 
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
20 
\end{bmatrix}
$$

The 3rd row equals the 3rd row minus \( \frac{1}{h^2} / [\frac{-2}{h^2}] \) times the 2nd row.
Construct multiple frontal matrices in such a way so they sum up to the full matrix

Variables must be split into parts

$$x_i = x_i^I + x_i^\Pi$$
MULTI-FRONTAL SOLVER ALGORITHM

First all frontal matrices are constructed
First and second frontal matrices are sum up into new 3x3 frontal matrix

Its first and second rows are fully assembled
MULTI-FRONTAL SOLVER ALGORITHM

\[\begin{pmatrix}
1 & 0 & 0 \\
\frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\
0 & \frac{1}{h^2} & -\frac{1}{h^2}
\end{pmatrix} \begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix} = \begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}\]

First column is eliminated by using first row

2\text{nd row} = 2\text{nd row} - [1/h^2] \times 1\text{st row}
MULTI-FRONTAL SOLVER ALGORITHM

\[
\begin{align*}
&\begin{bmatrix}
1 & 0 & 0 \\
0 & -\frac{2}{h^2} & \frac{1}{h^2} \\
0 & \frac{1}{h^2} & -\frac{1}{h^2}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}

\end{align*}
\]

2\textsuperscript{nd} row = 2\textsuperscript{nd} row − \left[\frac{1}{h^2}\right] * 1\textsuperscript{st} row
Third and fourth frontal matrices are sum up into new 3x3 frontal matrix.
Now only the second row is fully assembled.
MULTI-FRONTAL SOLVER ALGORITHM

Change of the ordering
MULTI-FRONTAL SOLVER ALGORITHM

Eliminate entries below the diagonal

2\textsuperscript{nd} row = 2\textsuperscript{nd} row − \left[\frac{1}{h^2}\right] / \left[-\frac{2}{h^2}\right] \times 1\textsuperscript{st} row

3\textsuperscript{rd} row = 3\textsuperscript{rd} row − \left[\frac{1}{h^2}\right] / \left[-\frac{2}{h^2}\right] \times 1\textsuperscript{st} row
Why can I substract fully assembled 1\textsuperscript{st} row from not fully assembled rows 2\textsuperscript{nd} and 3\textsuperscript{rd}?

Eliminate entries below the diagonal

2\textsuperscript{nd} row = 2\textsuperscript{nd} row – [1/h\textsuperscript{2}] / [-2/h\textsuperscript{2}] * 1\textsuperscript{st} row

3\textsuperscript{rd} row = 3\textsuperscript{rd} row – [1/h\textsuperscript{2}] / [-2/h\textsuperscript{2}] * 1\textsuperscript{st} row
MULTI-FRONTAL SOLVER ALGORITHM

This is because substraction and addition are interchangable (now I am substructing from the 2\textsuperscript{nd} not fully assembled row, and later I will add the remaining part)

Moreover the 1\textsuperscript{st} row which is utilized for the substractions in other columns contains only zeros

Eliminate entries below the diagonal

2\textsuperscript{nd} row = 2\textsuperscript{nd} row − [1/h\textsuperscript{2}] / [-2/h\textsuperscript{2}] * 1\textsuperscript{st} row

3\textsuperscript{rd} row = 3\textsuperscript{rd} row − [1/h\textsuperscript{2}] / [-2/h\textsuperscript{2}] * 1\textsuperscript{st} row
MULTI-FRONTAL SOLVER ALGORITHM

Eliminate entries below the diagonal


Fifth and sixth frontal matrices are sum up into new 3x3 frontal matrix
Now the second and third rows are fully assembled
MULTI-FRONTAL SOLVER ALGORITHM

Continue until root node
PARALLEL MULTI-FRONTAL SOLVER ALGORITHM

All frontal matrices are generated at the same time
PARALLEL MULTI-FRONTAL SOLVER ALGORITHM

Summing up and elimination are executed at the same time over different pairs of frontal matrices.
PARALLEL MULTI-FRONTAL SOLVER ALGORITHM

Summing up and elimination are executed at the same time over different pairs of frontal matrices.
The algorithm is recursively repeated until we reach the root of the tree.

The algorithm results in upper triangular matrix stored in distributed manner.

Computational complexity = height of the tree = \( \log N \) (where \( N \) = \#unknowns - 1)
GENERALIZATION TO 1D FINITE ELEMENT METHOD

Strong formulation

Find \( u \in C^2(0,l) \) such that

\[
-(a(x)u'(x))' + b(x)u'(x) + c(x)u(x) = f(x)
\]

\( u(0) = 0 \)

\( a(l) u'(l) + \beta u(l) = \gamma \)

Weak formulation

Find \( u \in V \) such that

\[
\int_0^l \left\{ au'v' + bu'v + cuv \right\} dx + \beta u(l)v(l) = \int_0^l f v dx + \gamma v(l)
\]

\( b(u,v) = l(v) \)

\( \forall v \in V = \left\{ v \in H^1(0,l) : v(0) = 0 \right\} \)
GENERALIZATION TO 1D FINITE ELEMENT METHOD

Finite element method discretization

\[ u = \sum_{i=1}^{N} a_i e_i \quad v = e_j \]

\[ \sum_{i=1}^{N} a_i b(e_i, e_j) = l(e_j) \quad j = 1, \ldots, N \]

\[ b(e_i, e_j) = \int_{0}^{l} \left\{ a e_i' e_j' + b e_i' e_j + c e_i e_j \right\} dx + \beta e_i(l) e_j(l) \]

\[ l(e_j) = \int_{0}^{l} f e_j dx + \gamma e_j(l) \]

Exemplary basis functions for \([0, l] = [0, 1]\), for two finite elements

\[ e_1(x) = \begin{cases} 1 - 2x & \text{for } x \in (0, 0.5) \\ 0 & \text{for } x \in (0.5, 1) \end{cases} \]

\[ e_2(x) = \begin{cases} 2x & \text{for } x \in (0, 0.5) \\ 2 - 2x & \text{for } x \in (0.5, 1) \end{cases} \]

\[ e_3(x) = \begin{cases} 0 & \text{for } x \in (0, 0.5) \\ 2x - 1 & \text{for } x \in (0.5, 1) \end{cases} \]

\[ \frac{de_1(x)}{dx} = \begin{cases} -2 & \text{for } x \in (0, 0.5) \\ 0 & \text{for } x \in (0.5, 1) \end{cases} \]

\[ \frac{de_2(x)}{dx} = \begin{cases} 2 & \text{for } x \in (0, 0.5) \\ -2 & \text{for } x \in (0.5, 1) \end{cases} \]

\[ \frac{de_3(x)}{dx} = \begin{cases} 0 & \text{for } x \in (0, 0.5) \\ 2 & \text{for } x \in (0.5, 1) \end{cases} \]
GENERALIZATION TO 1D FINITE ELEMENT METHOD

\[ u = \sum_{i=1}^{N} a_i e_i \quad v = e_j \]

\[
\begin{bmatrix}
    b(e_1, e_1) & b(e_2, e_1) & b(e_3, e_1) \\
    b(e_1, e_2) & b(e_2, e_2) & b(e_3, e_2) \\
    b(e_1, e_3) & b(e_2, e_3) & b(e_3, e_3)
\end{bmatrix}
\begin{bmatrix}
    a_1 \\
    a_2 \\
    a_3
\end{bmatrix} =
\begin{bmatrix}
    l(e_1) \\
    l(e_2) \\
    l(e_3)
\end{bmatrix}
\]

\[
b(e_i, e_j) = \int_{0}^{l} \left( ae_i' e_j' + be_i' e_j + ce_i e_j \right) dx + \beta e_i(l)e_j(l) \quad l(e_j) = \int_{0}^{l} f e_j dx + \gamma e_j(l)
\]

Exemplary basis functions for \([0, l] = [0, 1]\), for two finite elements

\[ e_1 = \chi^K_1 \]

\[ e_2 = \chi^K_1 + \chi^K_2 \]

\[ e_3 = \chi^K_2 \]

\[
\begin{align*}
    b(e_1, e_3) &= \int_{0}^{1} \left( ae_1' e_3' + be_1' e_3 + ce_1 e_3 \right) dx + \beta e_1(l)e_3(l) = 0 \\
    b(e_1, e_2) &= \int_{0}^{0.5} \left( ae_1' e_2' + be_1' e_2 + ce_1 e_2 \right) dx + \beta e_1(l)e_1(l) = \int_{0}^{0.5} \left( a \frac{d\chi^K_1}{dx} + b \frac{d\chi^K_2}{dx} \right) \chi^K_2 + c \chi^K_1 \chi^K_2 \int_{0}^{0.5} dx \\
    b(e_2, e_3) &= \int_{0}^{1} \left( ae_2' e_3' + be_2' e_3 + ce_2 e_3 \right) dx + \beta e_2(l)e_3(l) = \int_{0}^{0.5} \left( a \frac{d\chi^K_1}{dx} + b \frac{d\chi^K_2}{dx} \right) \chi^K_2 + c \chi^K_1 \chi^K_2 \int_{0}^{0.5} dx
\end{align*}
\]
GENERALIZATION TO 1D FINITE ELEMENT METHOD

\[ u = \sum_{i=1}^{N} a_i e_i \quad v = e_j \]

\[
\begin{bmatrix}
  b(e_1, e_1) & b(e_2, e_1) & b(e_3, e_1) \\
  b(e_1, e_2) & b(e_2, e_2) & b(e_3, e_2) \\
  b(e_1, e_3) & b(e_2, e_3) & b(e_3, e_3)
\end{bmatrix}
\begin{bmatrix}
  a_1 \\
  a_2 \\
  a_3
\end{bmatrix} = 
\begin{bmatrix}
  l(e_1) \\
  l(e_2) \\
  l(e_3)
\end{bmatrix}
\]

\[ b(e_i, e_j) = \int_{0}^{l} (ae_i ' e_j + be_i ' e_j + ce_i e_j) dx + \beta e_i (l) e_j (l) \quad l(e_j) = \int_{0}^{l} f e_j dx + \gamma e_j (l) \]

Exemplary basis functions for \([0,l] = [0,1]\), for two finite elements

1

\[ e_1 = \chi_{1}^{K_1} \]

1

\[ e_2 = \chi_{1}^{K_2} + \chi_{2}^{K_1} \]

1

\[ e_3 = \chi_{2}^{K_2} \]

\[ b(e_1, e_1) = \int_{0}^{1} (ae_1 ' e_1 + be_1 ' e_1 + ce_1 e_1) dx + \beta e_1 (l) e_1 (l) = \int_{0}^{0.5} \left\{ a \frac{d\chi_{1}^{K_1}}{dx} \frac{d\chi_{1}^{K_1}}{dx} + b \frac{d\chi_{1}^{K_1}}{dx} \chi_{1}^{K_1} + c \left(\chi_{1}^{K_1}\right)^2 \right\} dx \]

\[ b(e_2, e_2) = \int_{0}^{1} (ae_2 ' e_2 + be_2 ' e_2 + ce_2 e_2) dx + \beta e_2 (l) e_2 (l) = \int_{0}^{0.5} \left\{ a \frac{d\chi_{2}^{K_1}}{dx} \frac{d\chi_{2}^{K_1}}{dx} + b \frac{d\chi_{2}^{K_1}}{dx} \chi_{2}^{K_1} + c \left(\chi_{2}^{K_1}\right)^2 \right\} dx + \int_{0.5}^{1} \left\{ a \frac{d\chi_{1}^{K_2}}{dx} \frac{d\chi_{1}^{K_2}}{dx} + b \frac{d\chi_{1}^{K_2}}{dx} \chi_{1}^{K_2} + c \left(\chi_{1}^{K_2}\right)^2 \right\} dx \]

\[ b(e_3, e_3) = \int_{0}^{1} (ae_3 ' e_3 + be_3 ' e_3 + ce_3 e_3) dx + \beta e_3 (l) e_3 (l) = \int_{0}^{0.5} \left\{ a \frac{d\chi_{2}^{K_2}}{dx} \frac{d\chi_{2}^{K_2}}{dx} + b \frac{d\chi_{2}^{K_2}}{dx} \chi_{2}^{K_2} + c \left(\chi_{2}^{K_2}\right)^2 \right\} dx + \beta \]
\[
\chi^K_1 = 1 - \frac{x_r - x}{x_r - x_l}, \quad \chi^K_2 = \frac{x_r - x}{x_r - x_l}
\]

\[
\begin{bmatrix}
    b(e_1, e_1) & b(e_2, e_1) & b(e_3, e_1) \\
    b(e_1, e_2) & b(e_2, e_2) & b(e_3, e_2) \\
    b(e_1, e_3) & b(e_2, e_3) & b(e_3, e_3)
\end{bmatrix}
\begin{bmatrix}
    a_1 \\
    a_2 \\
    a_3
\end{bmatrix} = \begin{bmatrix}
    l(e_1) \\
    l(e_2) \\
    l(e_3)
\end{bmatrix}
\]

\[
\begin{bmatrix}
    b(\chi^K_1, \chi^K_1) & b(\chi^K_2, \chi^K_1) & 0 \\
    b(\chi^K_1, \chi^K_2) & b(\chi^K_2, \chi^K_1) + b(\chi^K_1, \chi^K_2) & b(\chi^K_2, \chi^K_2) \\
    0 & b(\chi^K_1, \chi^K_2) & b(\chi^K_2, \chi^K_2)
\end{bmatrix}
\begin{bmatrix}
    a_1 \\
    a_2 \\
    a_3
\end{bmatrix} = \begin{bmatrix}
    l(\chi^K_1) \\
    l(\chi^K_2) + l(\chi^K_1) \\
    l(\chi^K_2)
\end{bmatrix}
\]

Notice that when we switch from finite difference to finite elements, it only changes the local systems of equations at tree nodes.

\[
\begin{bmatrix}
    b(\chi^K_1, \chi^K_1) & b(\chi^K_2, \chi^K_1) \\
    b(\chi^K_1, \chi^K_2) & b(\chi^K_2, \chi^K_2)
\end{bmatrix}
\begin{bmatrix}
    x^K_1 \\
    x^K_2
\end{bmatrix} = \begin{bmatrix}
    l(\chi^K_1) \\
    l(\chi^K_2)
\end{bmatrix}
\]
GENERALIZATION TO 1D FINITE ELEMENT METHOD

\[
\chi_1^K = 1 - \frac{x_r - x}{x_r - x_l} \quad \chi_2^K = \frac{x_r - x}{x_r - x_l}
\]

Global basis functions are composed with local shape functions, e.g.

\[
e_2 = \chi_1^K + \chi_2^K
\]

Local system of equations generated over the element K

\[
\begin{bmatrix}
b\left(\chi_1^K, \chi_1^K\right) & b\left(\chi_2^K, \chi_1^K\right) \\
b\left(\chi_1^K, \chi_2^K\right) & b\left(\chi_2^K, \chi_2^K\right)
\end{bmatrix}
\begin{bmatrix}
x_1^K \\
x_2^K
\end{bmatrix}
= 
\begin{bmatrix}
l\left(\chi_1^K\right) \\
l\left(\chi_2^K\right)
\end{bmatrix}
\]
Notice that when we switch from finite difference to finite elements, it only changes the local systems of equations at tree nodes.
2D $hp$ FINITE ELEMENT METHOD

We seek the solution $u$ of some weak form of PDE as a linear combination of shape functions $e_h^i$ spread over finite element mesh.

$$u = \sum_{i=1}^{15} u_h^i e_h^i$$
The coefficients $u^i_{hp}$ (called „degrees of freedom” d.o.f.) are obtained by solving system of linear equations – finite element discretization of a weak (variational) form of PDE

$$\sum_{i=1}^{15} u^i_{hp} b(e^i_{hp}, e^j_{hp}) = l(e^j_{hp}) \quad j = 1, \ldots, 15$$

where $b(e^i_{hp}, e^j_{hp})$ and $l(e^j_{hp})$ are some integrals of shape functions $e^i_{hp}, e^j_{hp}$.
FRONTAL SOLVER
SOLUTION BASED ON LINEAR ORDER OF ELEMENTS

Generates frontal matrix for the first element, eliminates fully assembled degrees of freedom.
Generates frontal matrix for the second element, merges with the current frontal matrix, eliminates fully assembled degrees of freedom.
MULTI-FRONTAL SOLVER
SOLUTION BASED ON THE ELIMINATION TREE

Partial forward elimination
MULTI-FRONTAL SOLVER
SOLUTION BASED ON THE ELIMINATION TREE

Partial forward elimination
MULTI-FRONTAL SOLVER
SOLUTION BASED ON THE ELIMINATION TREE

Full forward elimination of the interface problem matrix
COMPARISON OF COSTS

Number of operations for partial forward elimination

\[
\sum_{m=1}^{b} m^2 - \sum_{m=1}^{(b-a)} m^2 = \frac{a(6b^2 - 6ab + 6b + 2a^2 - 3a + 1)}{6}
\]

Frontal solver

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>computational cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>9</td>
<td>271</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>9</td>
<td>729</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>1000</td>
</tr>
</tbody>
</table>

Multi-frontal solver

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>computational cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>9</td>
<td>271</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>9</td>
<td>271</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>27</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>569</td>
</tr>
</tbody>
</table>
CONSTRUCTION OF THE ELIMINATION TREE BASED ON THE HISTORY OF MESH REFINEMENTS

For any initial mesh, the elimination tree can be created based on nested dissection algorithm.
CONSTRUCTION OF THE ELIMINATION TREE BASED ON THE HISTORY OF MESH REFINEMENTS

The elimination tree created for the initial mesh is updated when the mesh is refined (elimination tree is constructed dynamically, during mesh refinements)
The elimination tree created for the initial mesh is updated when the mesh is refined (elimination tree is constructed dynamically, during mesh refinements)
CONSTRUCTION OF THE REFINEMENT TREES
CONSTRUCTION OF THE REFINEMENT TREES
CONSTRUCTION OF THE REFINEMENT TREES

1
- 3
  - 11
    - 15
      - 18

2
- 5
  - 6
    - 9
      - 10
  - 16
    - 17
      - 18

3
- 4
  - 7
    - 8

4
- 12
  - 13
    - 14

5
- 15
  - 16

6
CONSTRUCTION OF THE REFINEMENT TREES
ELIMINATIONS BASED ON REFINEMENT TREES

- Local matrices for active elements – leaves of the elimination tree – are created

Level of initial mesh elements

Level of refinement trees
ELIMINATIONS BASED ON REFINEMENT TREES

• Interior degrees of freedom are eliminated at every leaf

Level of initial mesh elements

Level of refinement trees
ELIMINATIONS BASED ON REFINEMENT TREES

- Schur complement contributions are merged at parent level
Fully assembled degrees of freedom are eliminated at parent nodes level (degrees of freedom related to common edges shared by both son elements can be eliminated now)
• Contributions to Schur complement are merged again at the next level
ELIMINATIONS BASED ON REFINEMENT TREES

- Fully assembled degrees of freedom are eliminated

Level of initial mesh elements

Level of refinement trees
ELIMINATIONS BASED ON REFINEMENT TREES

Level of initial mesh elements

Level of refinement trees

- Finally, Schur complement contributions are merged at the tree root node
• The common interface problem is solved at the tree root node
  (The size of the common interface problem corresponds to the size of
crossection of the entire domain)
• The solution obtained at root node is utilized at son nodes.
• The backward substitution is executed
The solution utilized at the second level nodes is utilized at their some nodes.

The backward substitution is executed.
ELIMINATIONS BASED ON REFINEMENT TREES

- The solution obtained at the third level nodes is utilized at leaf nodes.
- The backward substitution is executed on the level of leaf nodes.
PERFORMANCE OF THE 1st VERSION OF THE DISTRIBUTED MEMORY MULTI-FRONTAL PARALLEL SOLVER

1,482,570 degrees of freedom (68,826,475 non-zeros)

- Embedded into hp code
+ Outperforms MUMPS for large number of processors
- Slower than MUMPS for low number of processors

Profiling showed that the time consuming part for low number of processors was actually process of merging of matrices – performed on the level of unknowns, with moving of matrix entries.

Switch to the level of nodes, do not touch matrix entries – work with pointers
PAPERS
http://home.agh.edu.pl/~paszynsk/Publications.html

Maciej Paszyński, David Pardo, Carlos Torres-Verdin, Leszek Demkowicz, Victor Calo
A PARALLEL DIRECT SOLVER FOR SELF-ADAPTIVE HP FINITE ELEMENT METHOD
Journal of Parallel and Distributed Computing, 70, 3 (2013) 270-281

Anna Paszynska, Maciej Paszynski, Konrad Jopek, Maciej Wozniak, Damian Goik, Piotr Gurgul, Hassan AbouEisha, Mikhail Moshkov, Victor Calo, Andrew Lenharth, Donald Nguyen, Keshav Pingali
QUASI-OPTIMAL ELIMINATION TREES FOR 2D GRIDS WITH SINGULARITIES

Maciej Paszynski, David Pardo, Anna Paszynska, Leszek Demkowicz
OUT-OF-CORE MULTI-FRONTAL SOLVER FOR MULTI-PHYSICS HP ADAPTIVE PROBLEMS
Procedia Computer Science, 4 (2011) 1788-1797