SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Implementation of FEM on HPC II – Solver types

Miroslav Halilovič, **Bojan Starman**, Janez Urevc, Nikolaj Mole

Faculty of Mechanical Engineering, University of Ljubljana

June/2021

Univerza v Ljubljani

TECHNISCHE UNIVERSITÄT WIEN

CINECA consorzio interuniversitario

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Co-funded by the Erasmus+ Programme of the European Union

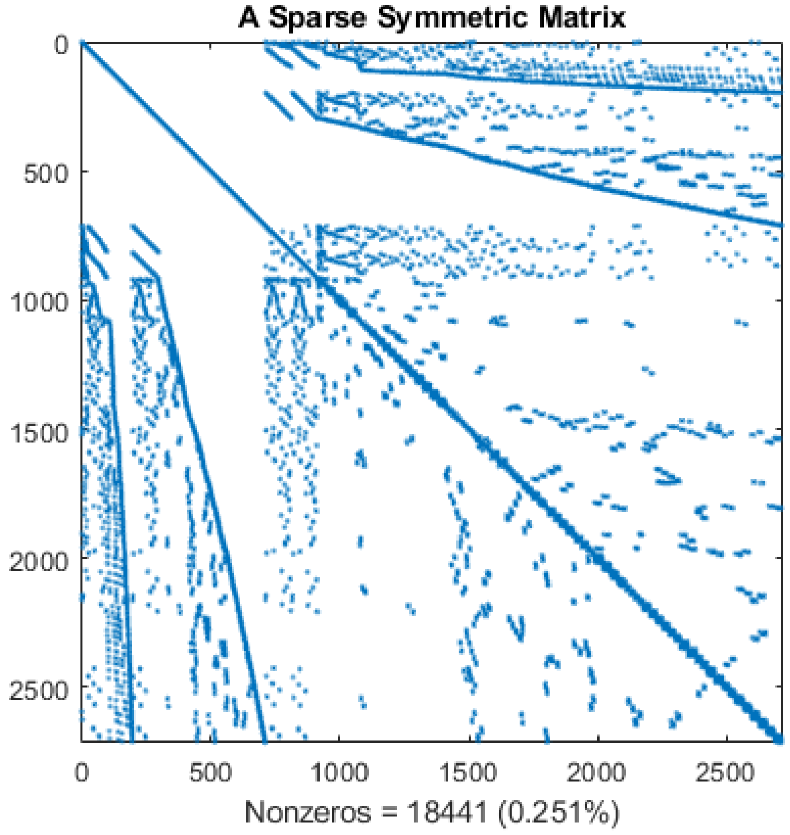
This project has been funded with support from the European Commission.
This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

System of linear equations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Different methods yield different systems of equations:

- Finite Difference Method (FDM) (large, sparse, unsymmetrical matrices)
- Finite Element Method (FEM) (large, sparse, generally unsymmetrical matrices)
- Finite Volume Method (FVM) (large, sparse, generally symmetric matrices)
- Boundary Element Method (BEM) (small systems, dense, unsymmetrical)



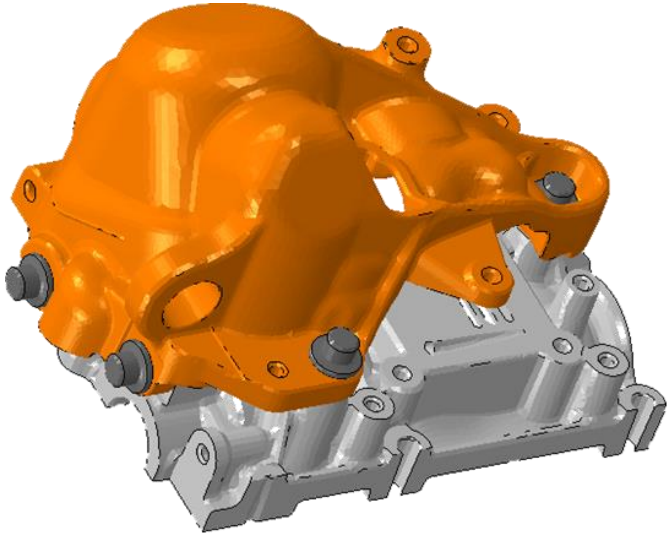
[1] https://au.mathworks.com/help/matlab/math/sparse-matrix-reordering.html

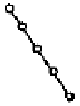
System of linear equations

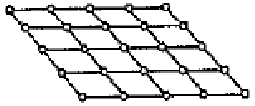
SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

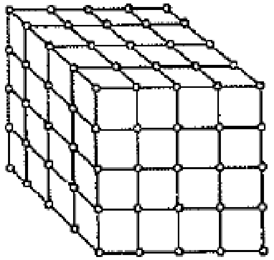
Different applications (meshes) result in different systems of equations:

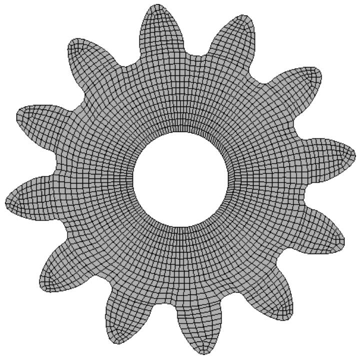
- Physical model is made from several parts or branches that are connected together (e.g. gears, spoked wheel)
- Space frames and other structures modelled with beams, trusses, and shells
- Blocky physical structures (solids, coupled structures in contact)
- 1D, 2D, 3D, degrees of freedom?
- Linear vs. nonlinear analysis?





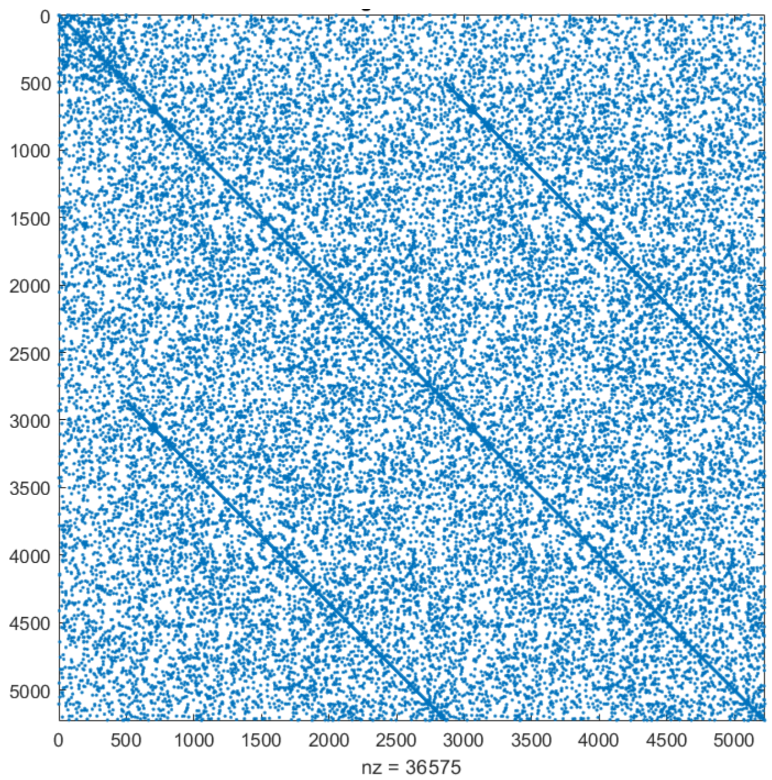


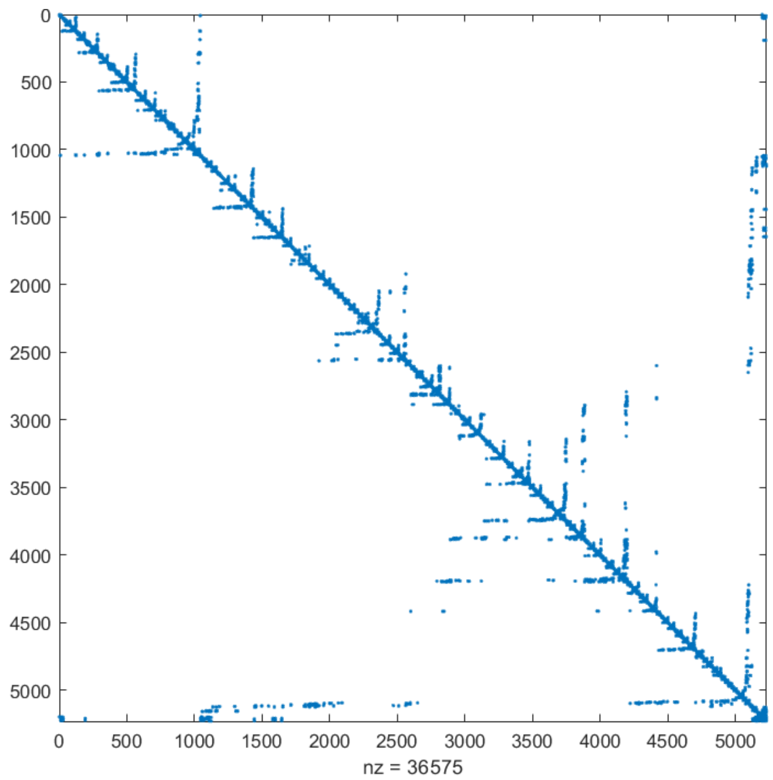




SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

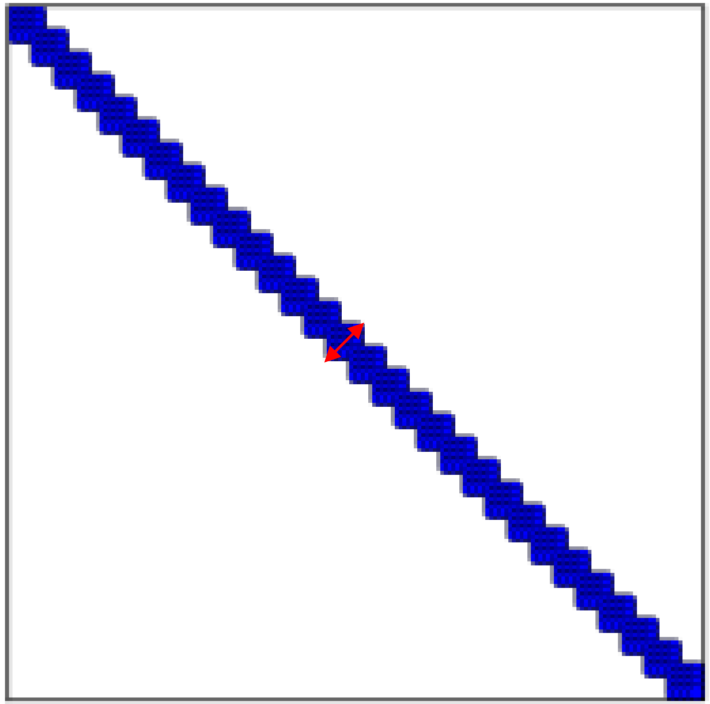
Sparse vs. dense matrices, bandwidth



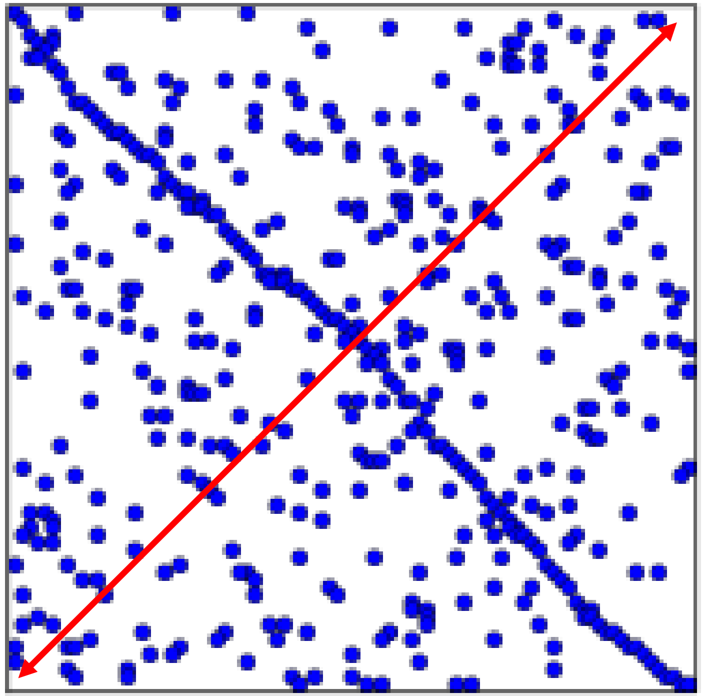


[1] https://au.mathworks.com/help/matlab/math/sparse-matrix-reordering.html

Influence of mesh numbering on bandwidth



Left-to-right numbering



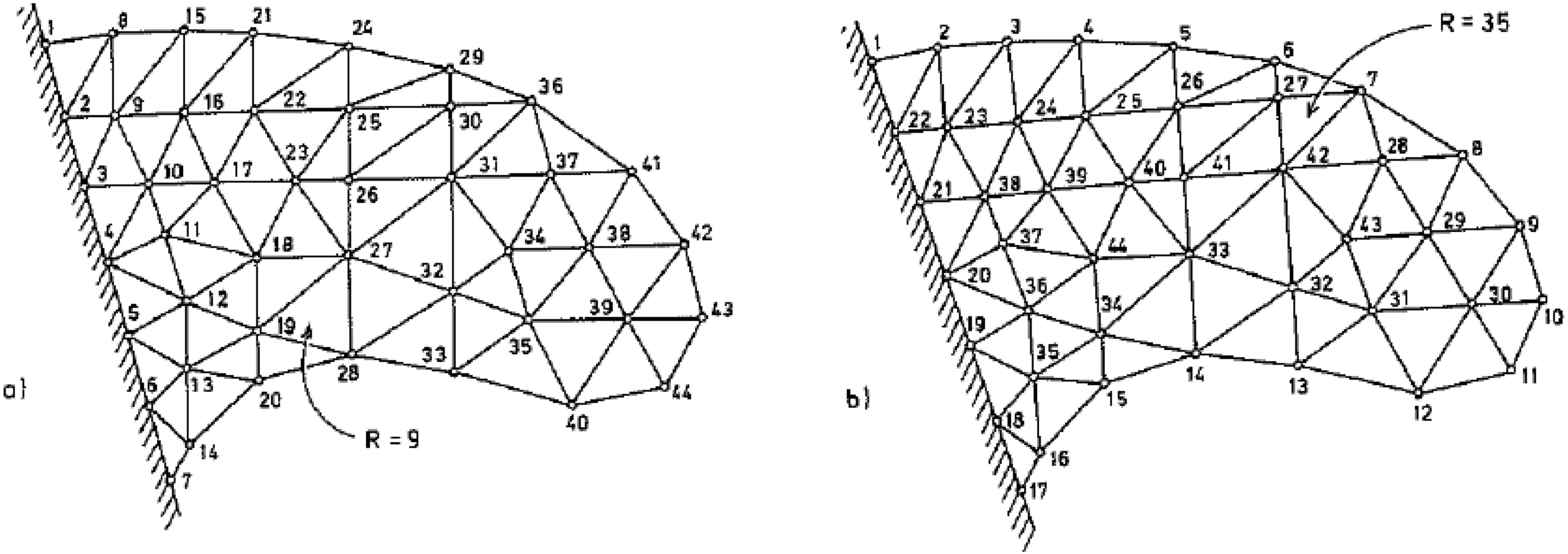
Random numbering

System of linear equations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Influence of mesh numbering on bandwidth

$$B = (R+1)N_{DOF}$$

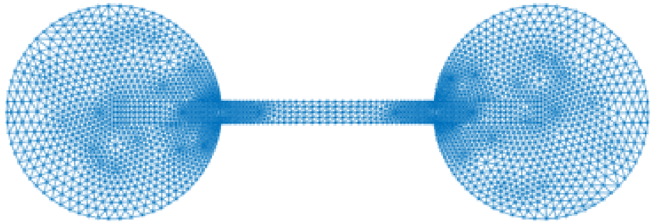


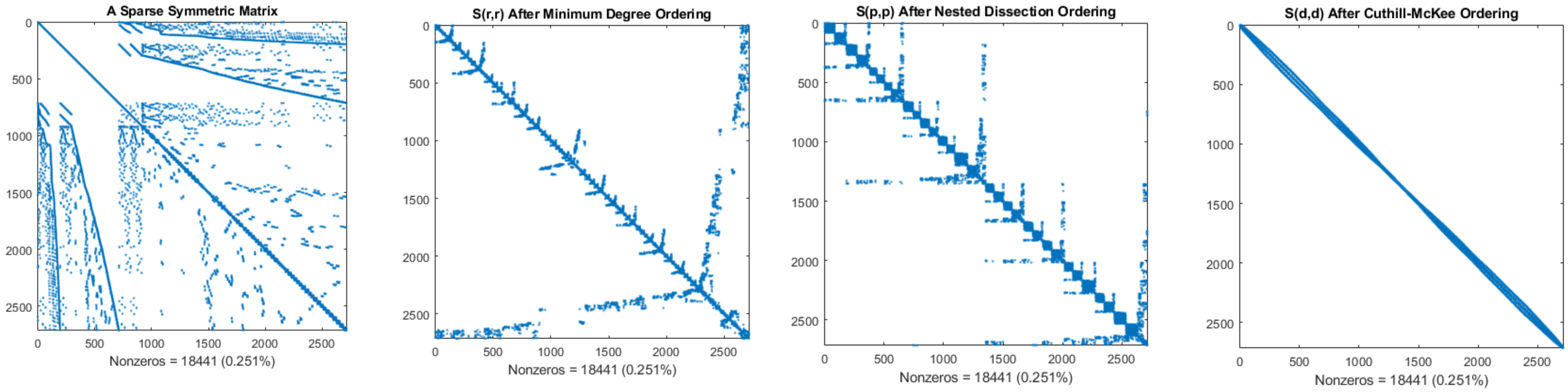
[2] N. S. Ottosen, H. Petersson : Introduction to the Finite Element Method

System of linear equations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Reordering the rows and columns of a sparse matrix can influence the speed and storage requirements of a matrix operation



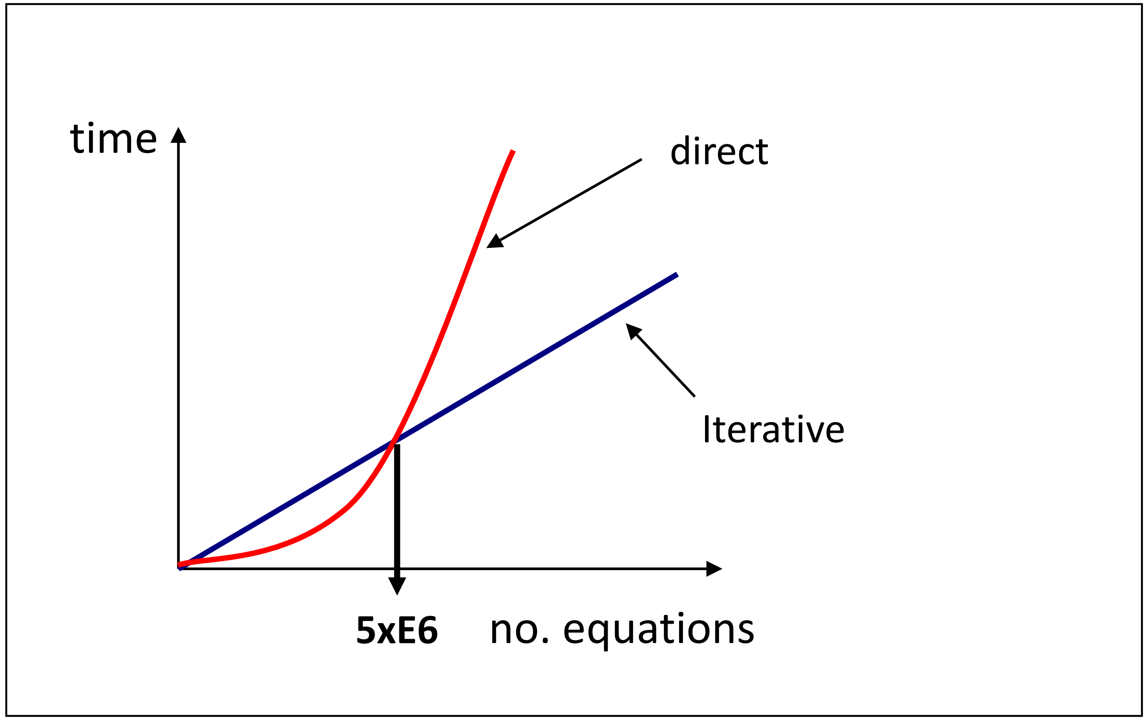


[1] https://au.mathworks.com/help/matlab/math/sparse-matrix-reordering.html

System of linear equations

Solution of a linear system of equations:

- Direct Solvers
  - Gauss elimination
  - Direct sparse solver (**MultiFront solver**)
  - LU decomposition
  - Cholesky method
  - **Domain Decomposition Method**
- Iterative Solvers
  - Gauss-Jacobi method
  - Gauss-Seidel method
  - **Krylov method**



System of linear equations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Direct solvers:

- The direct linear equation solver finds the **exact solution to this system of linear equations** (up to machine precision).
- often represents the most time consuming part of the analysis (especially for large models) **— the storage of the equations occupies the largest part of the disk space during the calculations.**
- **Sparsity and bandwidth** have major impact on the **computational time**
- **physical model that is made from several parts or branches that are connected together;** a spoked wheel is a good example of a structure that has a sparse stiffness matrix

System of linear equations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Iterative solvers:

- linear or nonlinear static, quasi-static, geostatic, pore fluid diffusion, heat transfer analysis ….
- **iterative -> a converged solution to a given system of linear equations cannot be guaranteed**
- when converges, **the accuracy of this solution depends on the relative tolerance** that is used
- **highly sensitive to the model geometry, favouring blocky type structures** (i.e., models that look more like a cube than a plate) **with a high degree of mesh connectivity and a relatively low degree of sparsity**
- The rate at which the approximate solution converges is directly related to the conditioning of the original system of equations. **A linear system that is well conditioned will converge faster than an ill-conditioned system.**

System of linear equations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Deciding to use an iterative solver:

- Element type, contact and constraint equations, material and geometric nonlinearities and material properties
- Ill-conditioned models -> the iterative solver may converge very slowly or fail to converge. This may occur, for example, **if many elements have poor aspect ratios.**
- outperform the direct sparse solver only for **blocky models when number of DOF > 5 millions**
- **for some element types (i.e. cohesive) will likely lead to nonconvergence**
- **constraint equations** (multi-point constraints, surface-based tie constraints, kinematic couplings) solution cost grows linearly -> **recommended to keep such constraints to a minimum if possible, CONTACT -> special care must be taken due to large discontinuities**
- **material properties: large discontinuities in material behaviour (many orders of magnitude) will most likely converge slowly and possibly stagnate.**

System of linear equations

Gauss elimination:

$$computational\ cost = \alpha\ n\ B^2$$

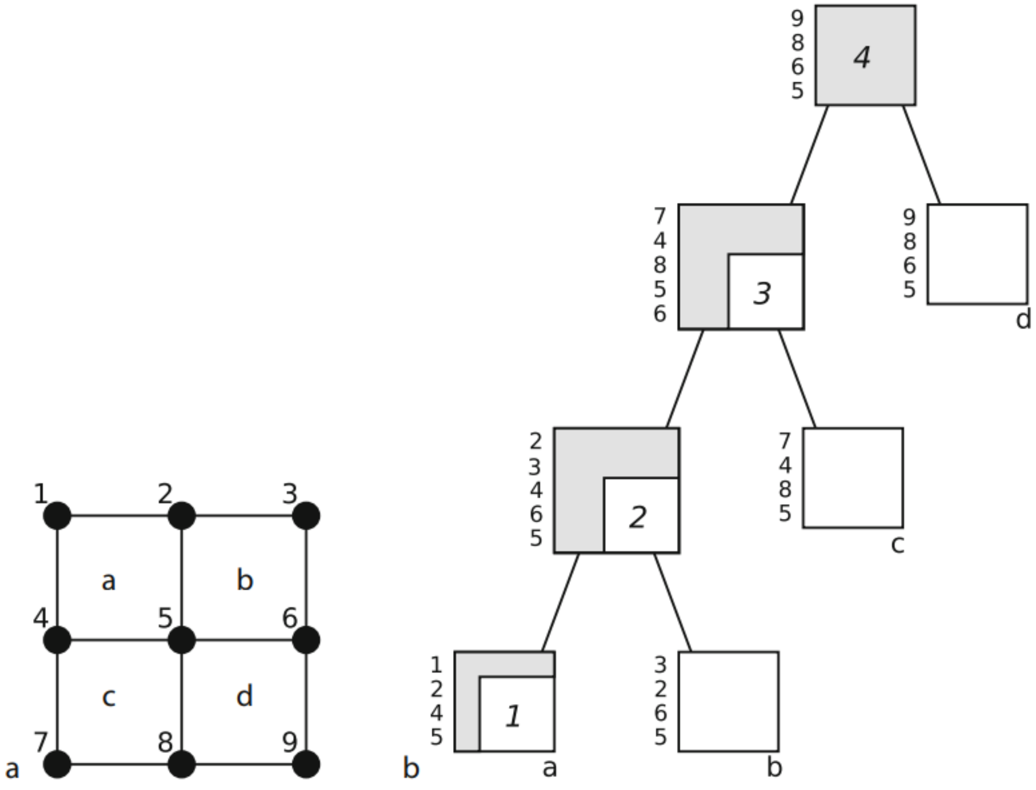
$$\begin{bmatrix} 2 & -2 & -2 & 0 \\ -2 & 4 & -2 & -2 \\ -2 & -2 & 12 & -2 \\ 0 & -2 & -2 & 22 \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -5 \\ 7 \end{pmatrix} \qquad \begin{bmatrix} 2 & -2 & -2 & 0 \\ 0 & 2 & -4 & -2 \\ 0 & -4 & 10 & -2 \\ 0 & -2 & -2 & 22 \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -4 \\ 7 \end{pmatrix}$$

$$\begin{bmatrix} 2 & -2 & -2 & 0 \\ 0 & 2 & -4 & -2 \\ 0 & 0 & 2 & -6 \\ 0 & 0 & -6 & 20 \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -2 \\ 7 \end{pmatrix} \qquad \begin{bmatrix} 2 & -2 & -2 & 0 \\ 0 & 2 & -4 & -2 \\ 0 & 0 & 2 & -6 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -2 \\ 2 \end{pmatrix}$$

System of linear equations

direct sparse solver (MultiFront solver)

SPARSE symmetric matrices -> **assembly and static condensation process can be performed at the same time! -> Frontal Solver can reduce the computational time to solve the equations dramatically if the equation system has a sparse structure**

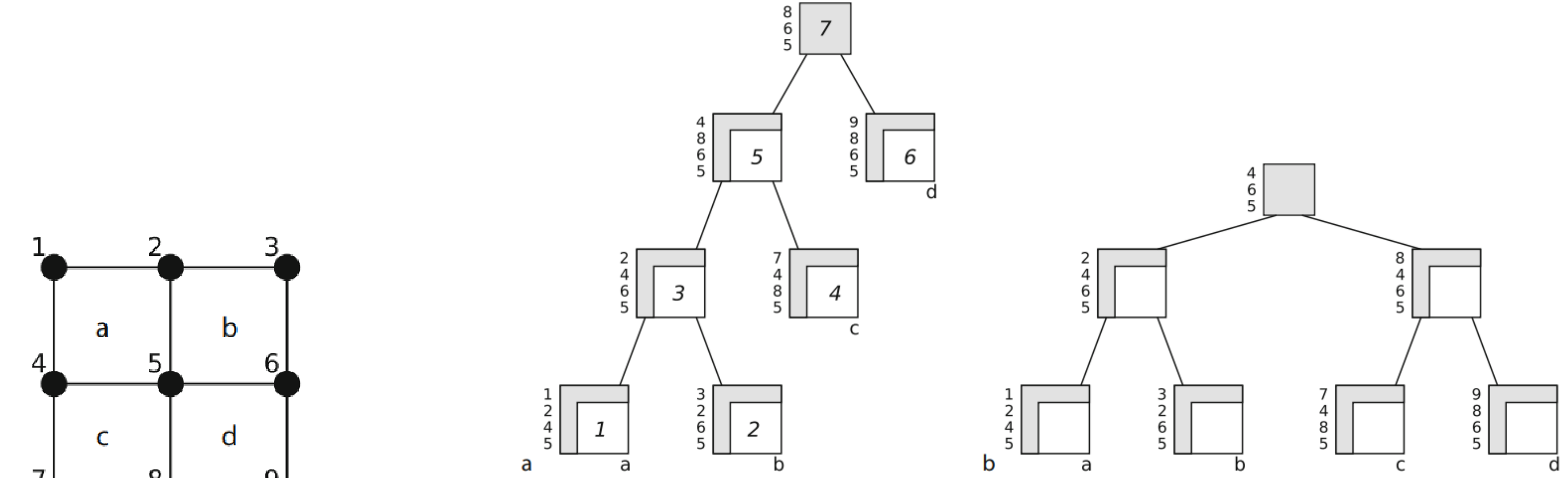


[3] https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-09766-4_86

System of linear equations

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

direct sparse solver (MultiFront solver):

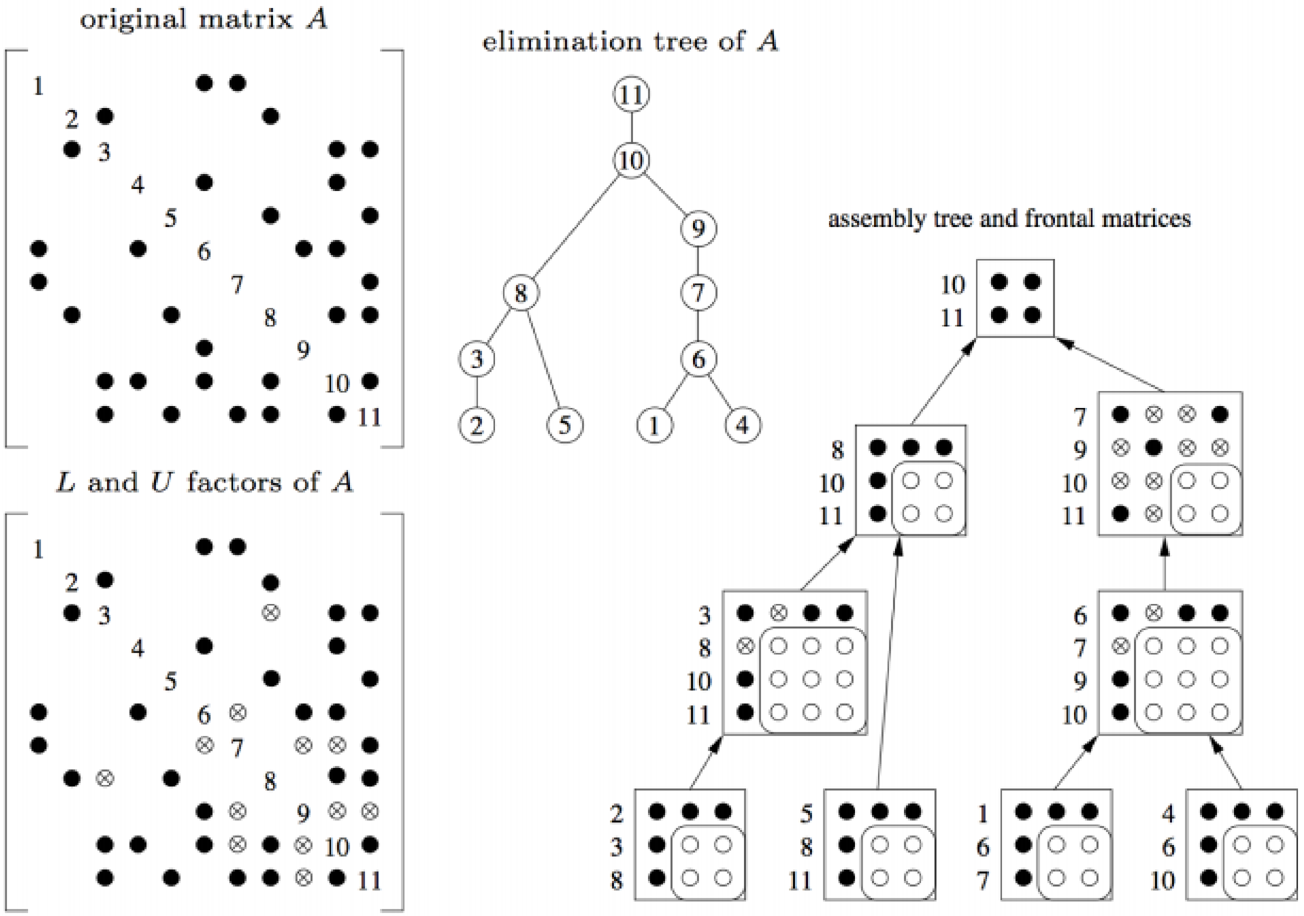


Multifrontal Method. Fig. 3 Finite-element problem and examples of associated assembly trees. Fully assembled variables are shown with a dark-shaded area within each frontal matrix

[3] https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-09766-4_86

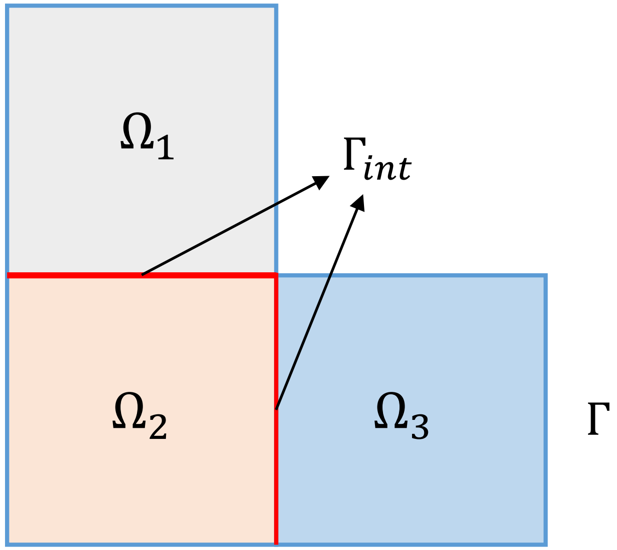
System of linear equations

direct sparse solver (MultiFront solver):



[3] https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-09766-4_86

Domain Decomposition Method:



$x_1 \in \Omega_1$
$x_2 \in \Omega_2$
$x_3 \in \Omega_3$

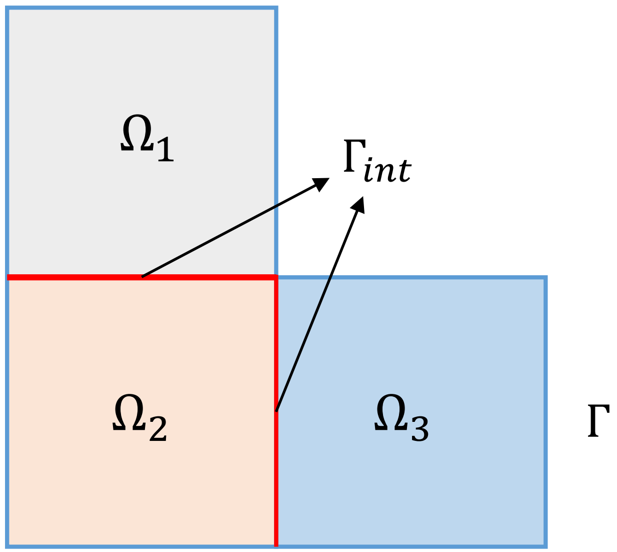
x_i - the solution of the domain internal points
y – the solution in the inter domain boundaries Γ_{int}.

The system of equations can be rewritten as:

$$\begin{bmatrix} \boldsymbol{B_1} & 0 & 0 & \boldsymbol{E_1} \\ 0 & \boldsymbol{B_2} & 0 & \boldsymbol{E_3} \\ 0 & 0 & \boldsymbol{B_3} & \boldsymbol{E_3} \\ \boldsymbol{F_1} & \boldsymbol{F_2} & \boldsymbol{F_3} & \boldsymbol{C} \end{bmatrix} \begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \boldsymbol{x_3} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_2} \\ \boldsymbol{f_3} \\ \boldsymbol{g} \end{pmatrix}$$

System of linear equations

Domain Decomposition Method:



$$\begin{bmatrix} B_1 & 0 & 0 & E_1 \\ 0 & B_2 & 0 & E_3 \\ 0 & 0 & B_3 & E_3 \\ F_1 & F_2 & F_3 & C \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ g \end{pmatrix} \Rightarrow$$

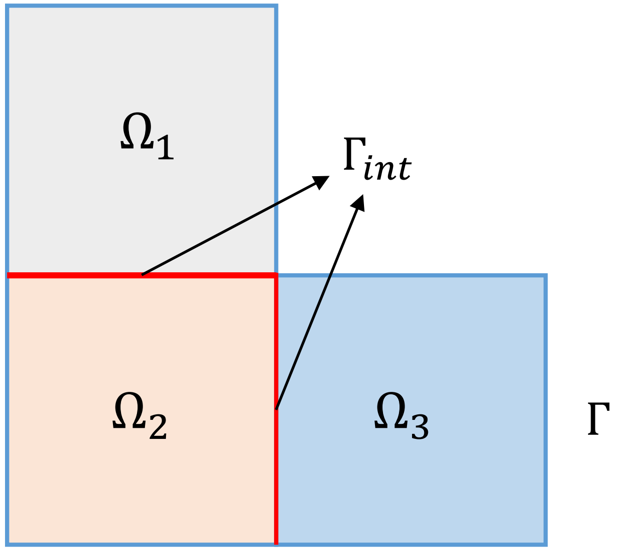
$$\begin{bmatrix} B & E \\ F & C \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \qquad B\,x + E\,y = f \Rightarrow x = B^{-1}(f - E\,y)$$

$$FB^{-1}(f - E\,y) + C\,y = g \Rightarrow \left(C - FB^{-1}E\right)y = g - FB^{-1}f$$

$$\Downarrow$$

$$S\ (Schur\ Component) \qquad \boxed{y = S^{-1}(g - FB^{-1}f)}$$

Domain Decomposition Method:



$$\begin{bmatrix} B_1 & 0 & 0 & E_1 \\ 0 & B_2 & 0 & E_3 \\ 0 & 0 & B_3 & E_3 \\ F_1 & F_2 & F_3 & C \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ g \end{pmatrix}$$

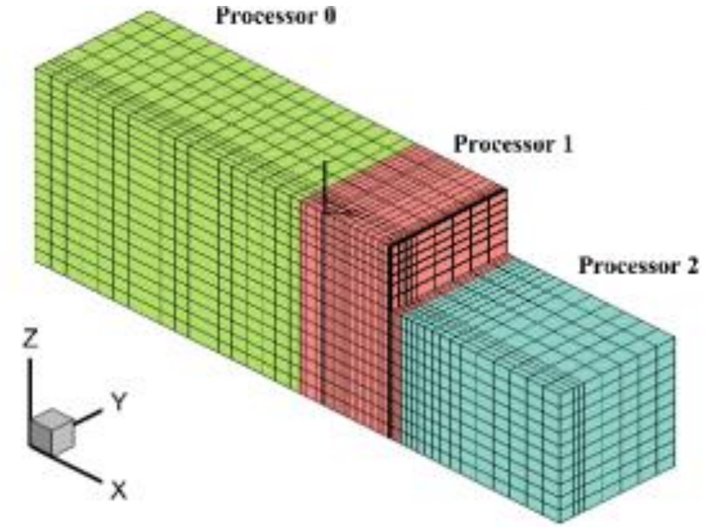
$$y = S^{-1}(g - FB^{-1}f)$$

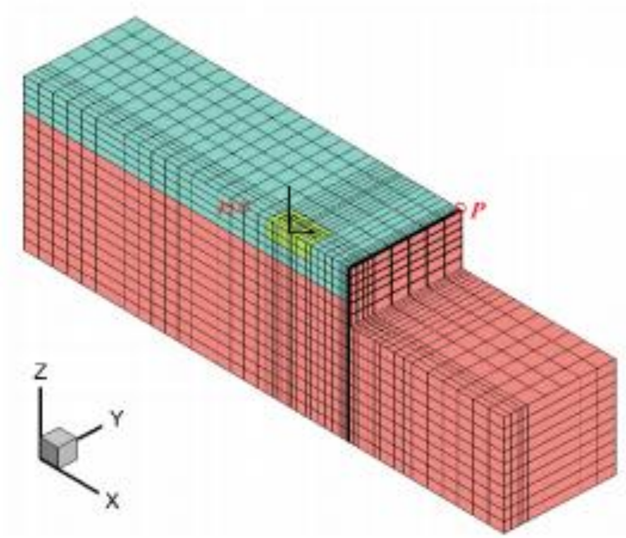
$$x_1 = B_1^{-1}(f_1 - E_1\, y)$$

$$x_2 = B_2^{-1}(f_2 - E_2\, y)$$

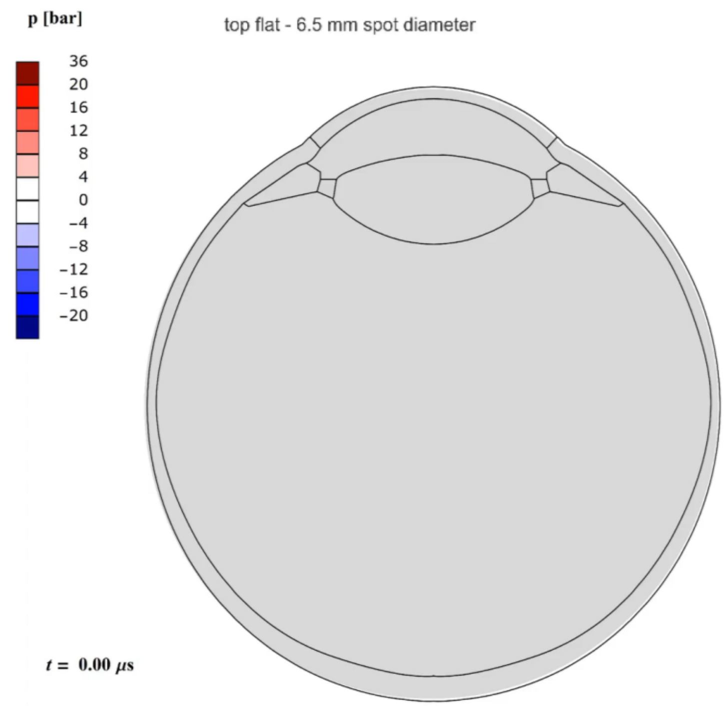
$$x_3 = B_3^{-1}(f_3 - E_3\, y)$$

Domain Decomposition Method:





Best practices – simulation of an laser-induced mechanical waves inside the human eye following laser medical procedures



Thank you for your attention!

http://sctrain.eu/

Univerza v Ljubljani
TECHNISCHE UNIVERSITÄT WIEN
CINECA consorzio interuniversitario
VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission.
This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.