Methods for improving SLA

Delft University of Technology

Wouter Swart

February 23, 2018

Outline

Motivation

Motivation: NLFEA incremental-iterative solution

- Assess structural strength using nonlinear finite element analysis.
- Search for equilibrium in-between load increments.

  $\rightarrow$ using Newton-Raphson type methods
- When analysing brittle structures large deformations can occur in-between load increments.

  $\rightarrow$ convergence is difficult
- Proposed solution: sequentially linear analyses (Rots)

Motivation: sequentially linear analysis

- Damage is incremented instead of load.

- Assumption: material degrades stepwise.

  $\rightarrow$ nonlinear behaviour can be approximated with sequence of linear analyses.

- Requirement: solve large number of linear systems.

Outline

Direct solution methods

Want to solve:

$$K\mathbf{u} = \mathbf{f}\ .$$

Direct solution methods typically consist of three stages

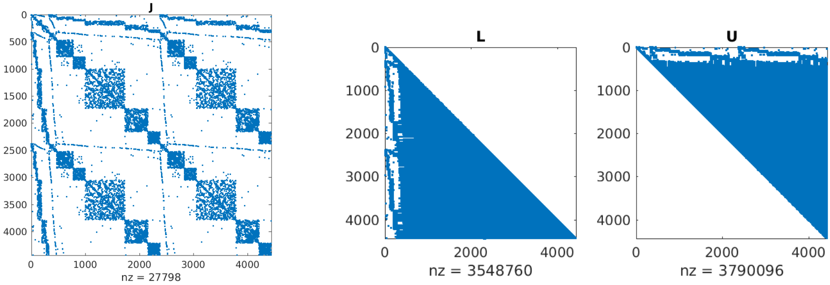
- Matrix reordering
- Factorisation
- Forward/backward substitution

Direct solution methods: factorisation

Want to write the stiffness matrix K as a product:

$$K = LU = \begin{pmatrix} l_{1,1} & & 0 \\ \vdots & \ddots & \\ l_{n,1} & \dots & l_{n,n} \end{pmatrix} \begin{pmatrix} u_{1,1} & \dots & u_{1,n} \\ & \ddots & \vdots \\ 0 & & u_{n,n} \end{pmatrix}.$$

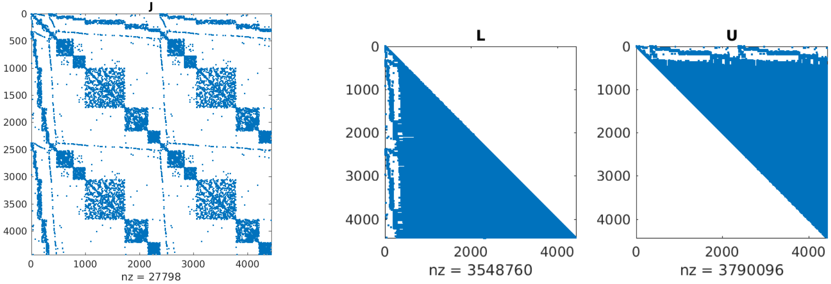
Example:



TU Delft

Direct solution methods: factorisation

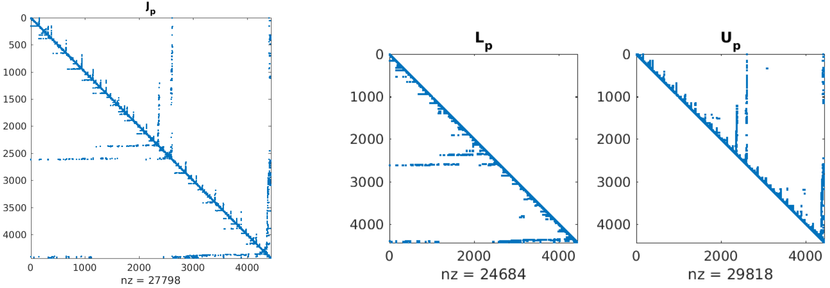
- Occurrence of fill-in.
  → Solution: reordering of the matrix prior to factorisation.



Direct solution methods: factorisation

- Occurrence of fill-in.

  → Solution: reordering of the matrix prior to factorisation.



Direct solution methods: factorisation

- Once factorisation is known solution can be easily computed.

$$K\mathbf{u} = \mathbf{f} \quad \Rightarrow \quad LU\mathbf{u} = \mathbf{f}$$

- Introducing $\mathbf{y} = U\mathbf{u}$ it suffices to solve:

$$\begin{aligned} L\mathbf{y} &= \mathbf{f} && \text{(Forward substitution)} \\ U\mathbf{u} &= \mathbf{y} && \text{(Backward substitution)} \end{aligned}$$

- Costs:

$$\begin{aligned} \text{Factorisation:} \quad & \mathcal{O}(n^3) \\ \text{Forward/backward subtitution:} \quad & \mathcal{O}(n^2) \end{aligned}$$

Outline

Iterative solution methods

Want to solve:

$$K\mathbf{u} = \mathbf{f} \,.$$

Instead of exact solution, create a sequence of approximations

$$\mathbf{u}_0, \mathbf{u}_1, \ldots$$

If K is symmetric and positive definite

$\rightarrow$ Conjugate gradient method is the best choice

Conjugate gradient method

- Constructs solution from a Krylov subspace
- Every iteration Krylov subspace dimension increases by 1
- After n iterations, the Krylov subspace spans the entire space

  $\rightarrow$ CG requires at most n iterations
- If the stiffness matrix only contains r distinct eigenvalues

  $\rightarrow$ only r iterations are required

Outline

Woodbury identity

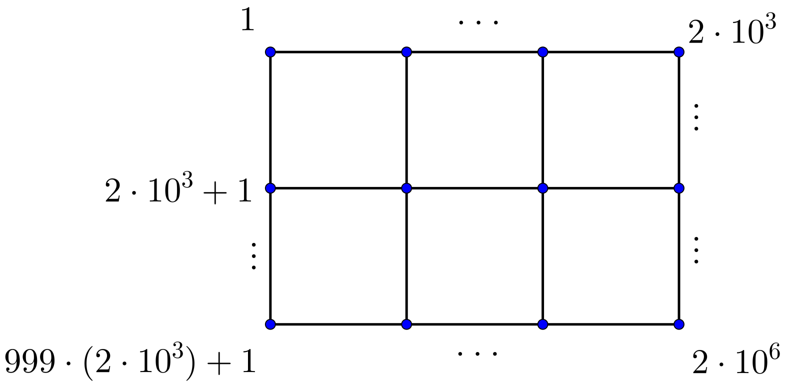
Theorem (Woodbury identity)

The inverse of a rank-k correction of the matrix A is given by:
$(K + UCV)^{-1} = K^{-1} - K^{-1}U\left(C^{-1} + VK^{-1}U\right)^{-1} VK^{-1}$

Solution to $K_{\mathrm{new}}\mathbf{u} = \mathbf{f}_{\mathrm{new}}$ can then be obtained as:

$$\begin{aligned}
\mathbf{u}_{\mathrm{new}} &= K_{\mathrm{new}}^{-1}\mathbf{f}_{\mathrm{new}} \\
&= (K + UCV)^{-1}\,\mathbf{f}_{\mathrm{new}} \\
&= \left(K^{-1} - K^{-1}U\left(C^{-1} + VK^{-1}U\right)^{-1} VK^{-1}\right)\mathbf{f}_{\mathrm{new}} \\
&= K^{-1}\mathbf{f}_{\mathrm{new}} - K^{-1}U\left(C^{-1} + VK^{-1}U\right)^{-1} VK^{-1}\mathbf{f}_{\mathrm{new}}
\end{aligned}$$

Woodbury identity: example



Woodbury identity: example

Direct method	**Flopcount**
Fact. (LU)	$1.6 \cdot 10^{13}$
Back/forward sub.	$1.6 \cdot 10^{10}$
Total	$1.6016 \cdot 10^{13}$

Woodbury identity	**Flopcount**
Fact. (LU)	$\frac{2}{3}k^3$
Back/forward sub.	$2k^2$
Additional	$1.6002 \cdot 10^{10} + k \cdot 1.6008 \cdot 10^{10} + k^2 \cdot 4 \cdot 10^6$

TU Delft

Woodbury identity: example

Determine k for which cost of Woodbury identity is costlier:

$$k = 828\,.$$

Assume: stiffness matrix gets rank-12 update after each analysis

$\rightarrow$ 69 linear analyses can be performed before restart is necessary

Outline

Preconditioning CG

- Possible improvement: precondition with (approximate) factorisation of initial stiffness matrix

  $\rightarrow$ $n - k$ eigenvalues will be equal to 1

  $\rightarrow$ only $k + 1$ CG iterations are required

- Use approximate factorisation when complete factorisation too expensive

TU Delft

Outline

Research proposal

For the remainder of the thesis, the following main research question is formulated:

> *How can SLA be improved such that it requires reduced CPU time?*

To answer this, the following approaches are proposed:

- Implement Woodbury matrix identity
- Investigate effectiveness of preconditioning iterative method with (approximate) factorisation

Outline

TU Delft

Preliminary results

- Per iteration only one element stiffness matrix is recomputed.
- Parallel computing environment was set-up nonetheless
  $\rightarrow$ unnecessary computational overhead.
- Removal resulted in significant improvements in CPU times:

Test problem	**SL Testsuit**	**Shear wall**	**Reinforced slab**
Before	16:58	50:00	5:00:38
After	14:18	46:39	3:54:21
	-15.72%	-6.70%	-22.05%

Methods for improving SLA

Delft University of Technology

Wouter Swart

February 23, 2018