Methods for improving the computational performance of sequentially linear analysis

Wouter Swart

Delft University of Technology

August 30, 2018

- Increased attention to numerical predictions

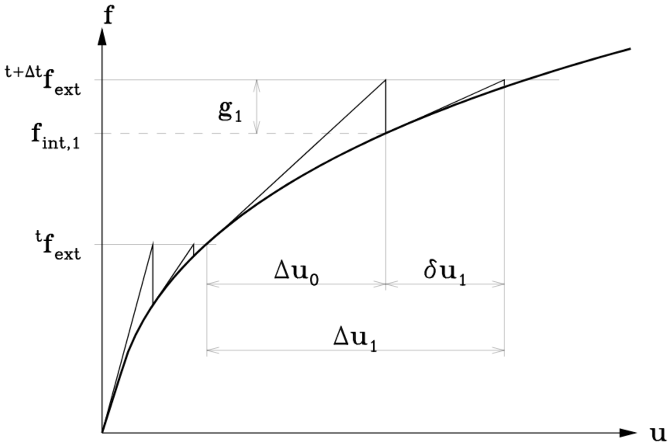
Outline

1. Nonlinear finite element analysis
2. Sequentially linear analysis
3. Solution methods
   - Direct
   - Iterative
4. Results

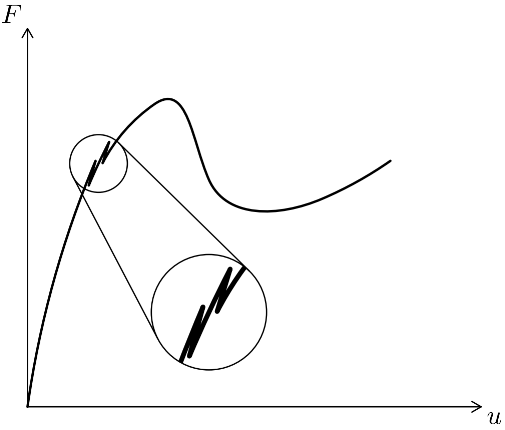
Nonlinear finite element analysis

- Nonlinear relation between external forces and resulting displacements
- Find displacement which balances internal and external forces
- Discretise space and force (increments)
- Within each increment, use iterative method to find force balance

Nonlinear finite element analysis



Nonlinear finite element analysis



- Proposed solution: sequentially linear analysis

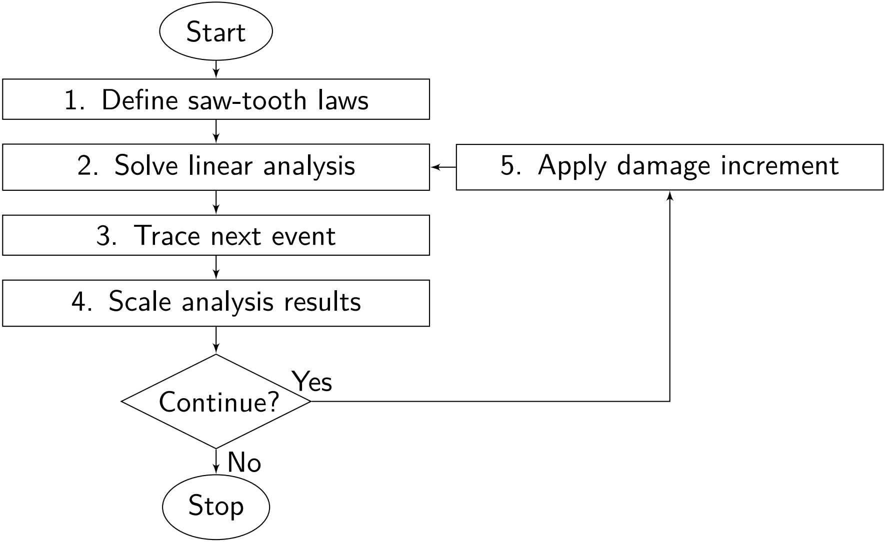
Sequentially linear analysis

- Main assumption: assume a stepwise material degradation
  $\rightarrow$ nonlinear response captured with a series of linear analyses

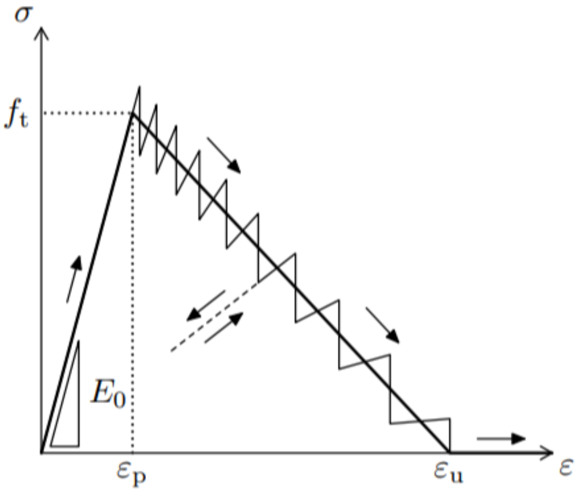
- Only 1 element is damaged per linear analysis

- Use automated selection procedure to find critical element

Sequentially linear analysis



Sequentially linear analysis



Sequentially linear analysis

- Only a single element is damaged in between analysis steps
- Resulting system of equations changes only locally
  $\rightarrow$ stiffness matrix is given low-rank correction
- Current solution methods do not exploit this property

Research question

How can the computational performance of SLA be improved such that it requires reduced computing time?

Next Subsection

Direct solution method

Want to solve:

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

Direct solution methods typically consist of three stages

- Matrix reordering
- Factorisation
- Forward & backward substitution

Direct solution method

- Problem: scalability with problem size:

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

- Prevent factorisation every analysis step

Direct solution method - improvement

Theorem (Woodbury's matrix identity)

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

Solution to $(K + UCV)\,\mathbf{u} = \mathbf{f}$ can then be obtained as:

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

Direct solution method - improvement

- How to determine U, C?

- Update to system stiffness matrix can be written as

$$K^{(n+1)} = K^{(n)} + NDN^T$$

- Calculating eigenvalue decomposition of D

$$D = Q\Lambda Q^T$$

- It follows that

$$\begin{aligned} K^{(n+1)} &= K^{(n)} + (NQ)\Lambda(NQ)^T \\ &:= K^{(n)} + UCU^T \end{aligned}$$

Direct solution method - improvement

Woodbury's identity strategy:

- First analysis step: calculate factorisation
  $\rightarrow$ expensive
- Subsequent analysis steps: use known factorisation
  $\rightarrow$ cheap
- Costs of analysis steps rise as rank increases
- Recalculate factorisation once the rank becomes too large

Next Subsection

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

- Finds approximation along search directions $\mathbf{p}_k$
  $\rightarrow$ conjugate w.r.t. K: $\mathbf{p}_i K \mathbf{p}_j = 0, i \neq j$

- $\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha \mathbf{p}_k$

- 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 $r + 1$ iterations are required

Conjugate gradient method - improvement

- Convergence of CG highly dependent on eigenvalues of K
- Use preconditioner P to obtain more favourable eigenvalues of $P^{-1}K$
- Use factorisation of K as preconditioner
- Result: $n - k$ eigenvalues will be equal to 1
  $\rightarrow$ only $k + 1$ CG iterations required

Results: restarting

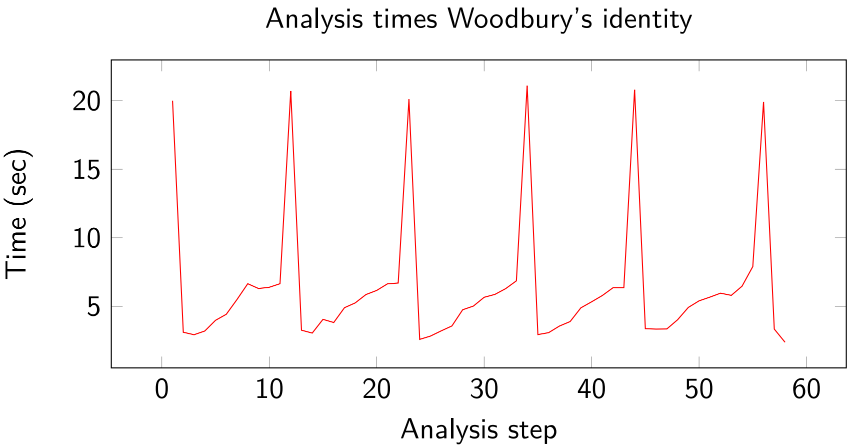
- Woodbury's identity and *PCG* are similar
  $\rightarrow$ One expensive analysis step (factorisation)
  $\rightarrow$ Subsequent analysis steps relatively cheap

- Costs increase as rank increases

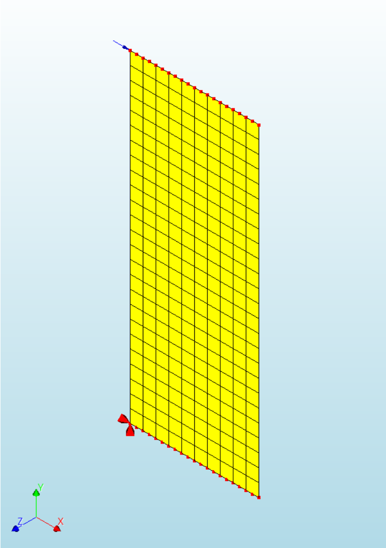
- Need to determine at what point to restart

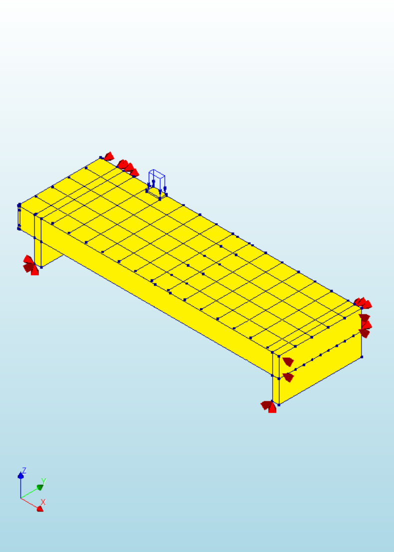
- Measure analysis times and extrapolate cost

Results: restarting

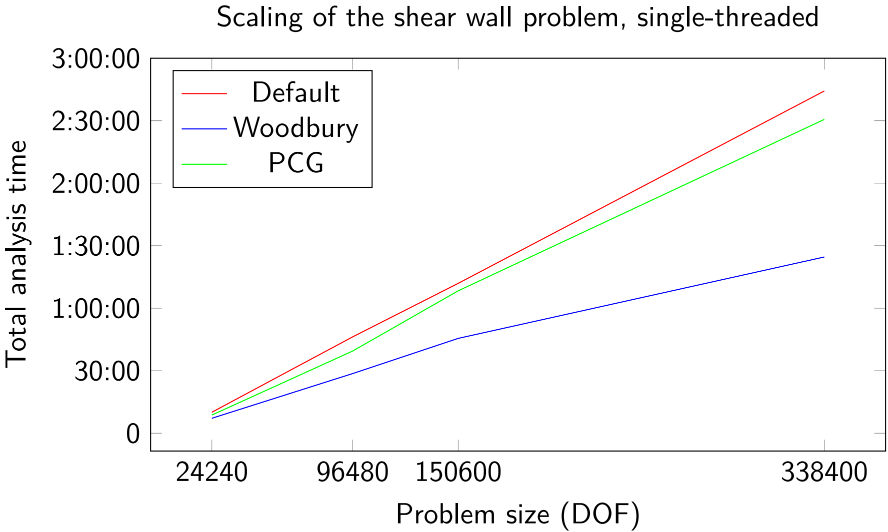


Results: benchmark problems

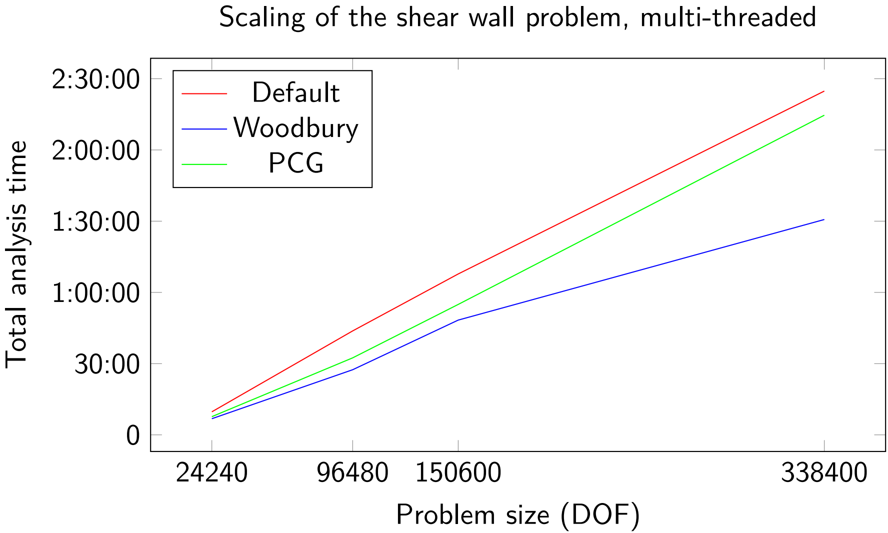




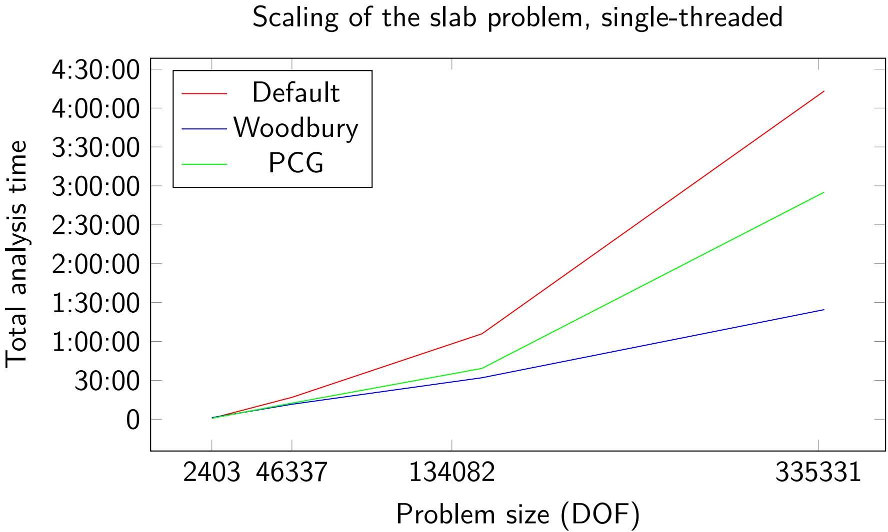
Results: benchmark problems



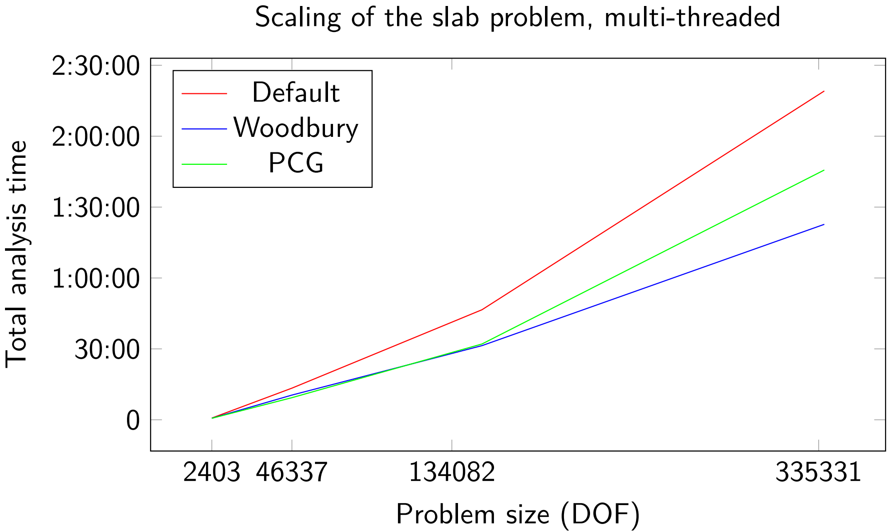
Results: benchmark problems



Results: benchmark problems



Results: benchmark problems



Results

- Contributions of solver to total analysis time

	Shear wall		Slab	
Threads	Single	Multi (4)	Single	Multi (4)
Default	47.9%	41.4%	70.6%	48.8%
Woodbury	16.7%	13.1%	20.0%	16.0%
PCG	39.1%	26.0%	29.2%	20.7%

Results

- Now able to solve larger problems

	Default	Woodbury
Problem size	335.331	386.448 (+15.2%)
Analysis steps	2500	2881 (+15.2%)
Analysis time	23:11:10	23:31:55

- Significantly reduced computing times for large problems
- Increased applicability of *SLA*

Methods for improving the computational performance of sequentially linear analysis

Wouter Swart

Delft University of Technology

August 30, 2018

Direct solution method - improvement

- How to choose eigenvalues in eigenvalue decomposition?

$$D = Q \Lambda Q^T$$

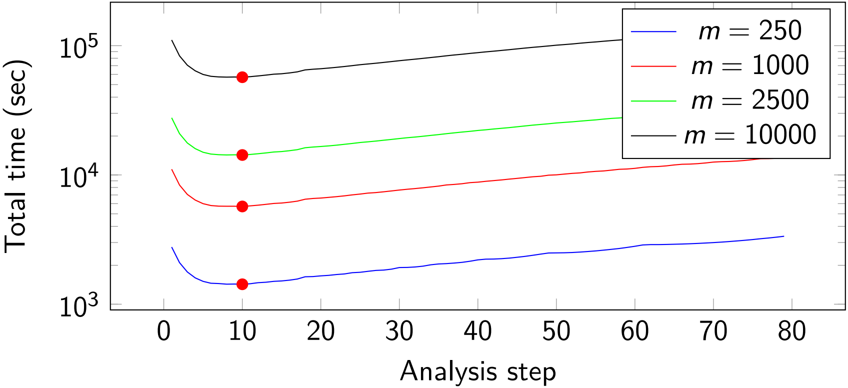
- Only want to choose 'large' eigenvalues corresponding to dominant features of D

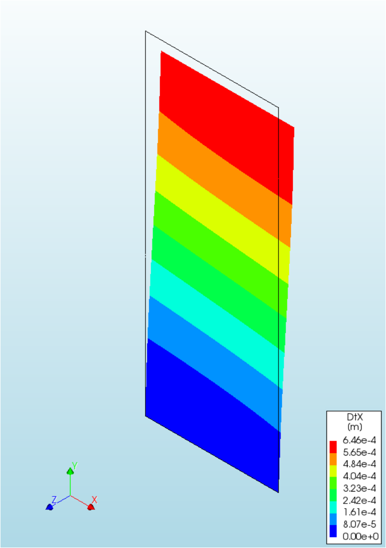
- Define eigenvalue ratio

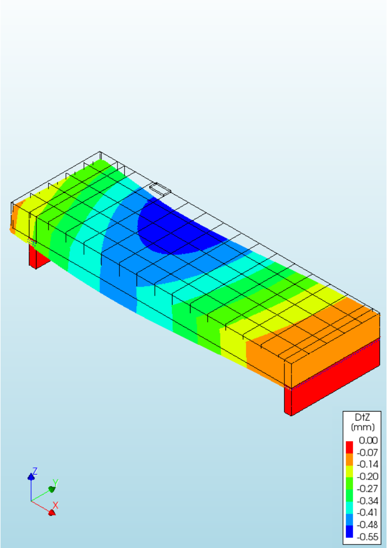
$$\epsilon = \frac{|\lambda_i|}{\lambda_{\text{max}}} \quad , \; \lambda_{\text{max}} = \max \{|\lambda_1|, \ldots, |\lambda_n|\}$$

- Numerical experiments: $\epsilon = 10^{-10}$ good choice

Total expected computing time as a function of restarting point




DtX
(m)
6.46e-4
5.65e-4
4.84e-4
4.04e-4
3.23e-4
2.42e-4
1.61e-4
8.07e-5
0.00e+0
Y
Z
X


DtZ
(mm)
0.00
-0.07
-0.14
-0.20
-0.27
-0.34
-0.41
-0.48
-0.55
Z
Y
X