Analytical Inversion of Tridiagonal Matrices used in solvers for Diffusion Problems

Institute for Neutron Physics and Reactor Technology (INR)

Till Glage, A. von der Weth, D. Piccioni Koch, F. Arbeiter, M. R. Schulz, D. Klimenko, G. Schlindwein, V. Pasler, K. Zinn

Table of Contents

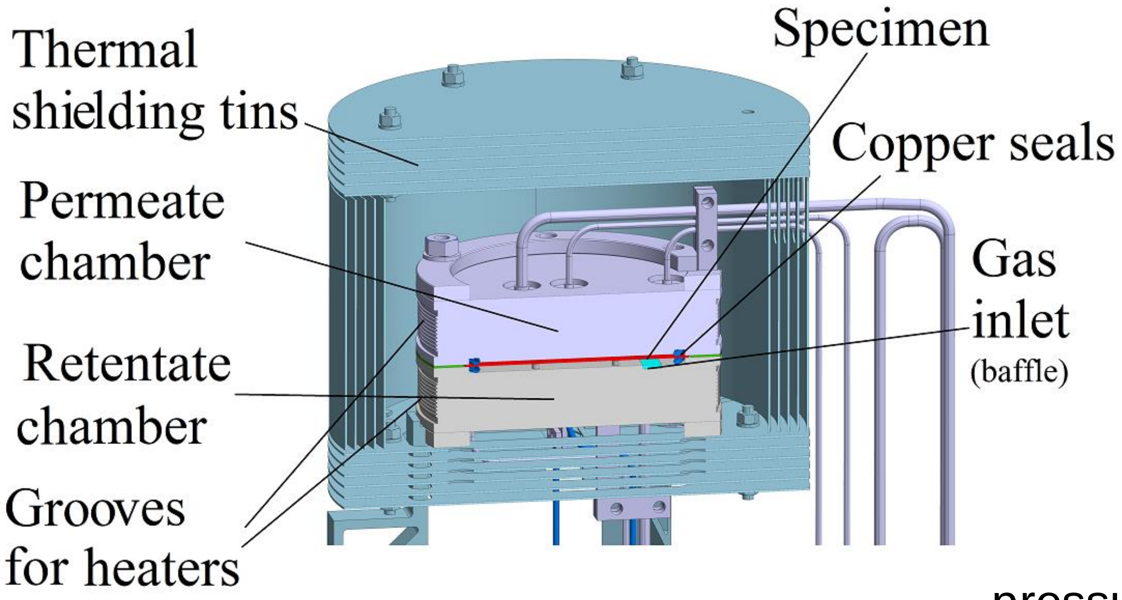
1. Short Introduction

Evaluation of transport parameters

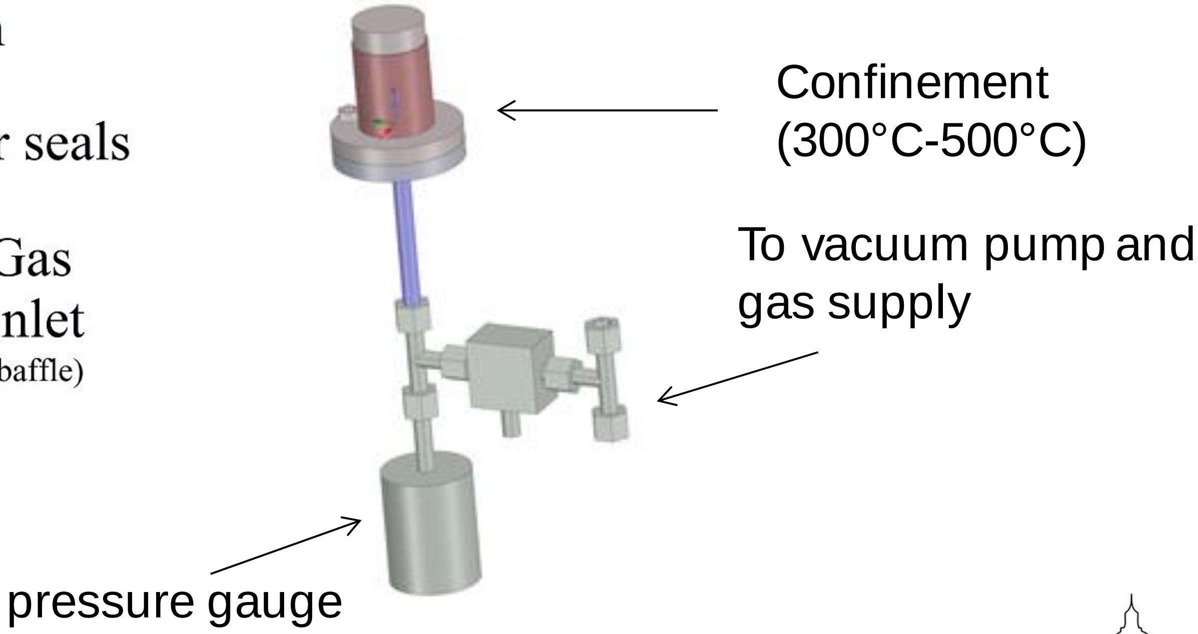
$D = ?; k_s = ?$

D – Diffusion constant; k_s - Sieverts constant

Q-PETE: Gas Permeation Experiment [1]



GRID: Gas Release Experiment [2]



[1] A. von der Weth et al.; Permeation Dataanalysis considering non-zero hydrogen concentration on the low pressure detector side fit a purged permeation experiment, Defect and Diffusion Forum, 2019
[2] Schulz, Marvin R. et al., Analytical Solution of a Gas Release problem considering permeation with time-dependent boundary conditions, Journal of Computational and Theoretical Transport, 2019

1. Short Introduction

- What we want to solve: $$D_{eff} \cdot \left(\frac{\partial^2 c}{\partial r^2} + \frac{1}{r} \frac{\partial c}{\partial r} + \frac{\partial^2 c}{\partial z^2} \right) = \frac{\partial c}{\partial t} \quad (1)$$
- Finite Difference Method of order 1 (see Taylor expansion):

$$c_{i,k+1} = D^* \left(1 - \frac{1}{2i}\right) \cdot c_{i-1,k} + c_{i,k} \cdot (1 - 2D^*) + D^* \left(1 + \frac{1}{2i}\right) c_{i+1,k} \quad (2)$$

- Why do we use approximative methods?
  - Rediffusion: Currently not solvable analytically
- Discretization:

$$D^* = \frac{D_{eff} \cdot \tau}{h^2} \quad (3)$$

 Till Glage, Institute for Neutron Physics und Reactor Technology (INR), till.glage9@kit.edu

2. Kinds of Solvers

- Matrix Solvers: (See Axel von der Weth's publications)
  - Von Neumann boundary condition for symmetry reasons:

$$\frac{\partial c}{\partial r} = 0 \quad \text{on} \quad \Gamma \qquad (4)$$

$$\text{for} \quad t = t_0 + \tau : \quad c_{k+1} = \mathbf{S} \cdot c_k; \quad (5)$$

$$\mathbf{S} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{3D^*}{4} & 1 - 2D^* & \frac{5D^*}{4} & & & \\ & \ddots & \ddots & & & \\ \dots & D^*\left(1 - \frac{1}{2i}\right) & 1 - 2D^* & D^*\left(1 + \frac{1}{2i}\right) & \dots & \\ & & \ddots & \ddots & & \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

2. Kinds of Solvers

- Standard way: forward solver (stability problems further point)
  - Scalar:

$$y_{k+1} = y_k + h \cdot f(x_k, y_k) \qquad y' = f(x, y) \quad (7)$$

  - One dimensional with linear algebra methods:

$$\mathbf{T} := \mathbf{S} - \mathbf{I} \Rightarrow c_{k+1} = \mathbf{T} \cdot c_k + c_k \quad (8)$$

- Backward solver:
  - Scalar:

$$y_{k+1} - h \cdot f(x_{k+1}, y_{k+1}) = y_k \quad (9)$$

  - One dimensional:

$$c_{k+1} - \mathbf{T} \cdot c_{k+1} = (\mathbf{I}_n - \mathbf{T}) \cdot c_{k+1} = c_k \Leftrightarrow c_{k+1} = (\mathbf{I}_n - \mathbf{T})^{-1} \cdot c_k = (2\mathbf{I}_n - \mathbf{S}_n)^{-1} \cdot c_k \quad (10)$$

3. Matrix Inversion

- Some abbreviations and the matrix to be inverted:

$$\mathcal{B}^{-1}_{i,j} = \begin{cases} \widetilde{\delta_i^-} := -D^*(1 - \frac{1}{2*i}) & \text{if} \quad j = i - 1 \\ \widetilde{\delta^*} := 1 + 2D^* & \text{if} \quad j = i \\ \widetilde{\delta_i^+} := -D^*(1 + \frac{1}{2*i}) & \text{if} \quad j = i + 1 \\ -1 & \text{if} \quad i = 1; \quad j = 2 \\ 2 & \text{if} \quad i = 1; \quad j = 1 \\ 0 & \text{else} \end{cases} \qquad (11)$$

$$\begin{bmatrix} 2 & -1 & 0 & 0 & 0 & \\ \widetilde{\delta_1^-} & \widetilde{\delta^*} & \widetilde{\delta_1^+} & 0 & 0 & \\ 0 & \widetilde{\delta_2^-} & \widetilde{\delta^*} & \widetilde{\delta_2^+} & 0 & \\ 0 & 0 & \widetilde{\delta_3^-} & \widetilde{\delta^*} & \widetilde{\delta_3^+} & \\ & & & \ddots & \ddots & \end{bmatrix}$$

DSL2021 Malta
28 JUNE–2 JULY
17th International Conference on Diffusion in Solids and Liquids

Till Glage, Institute for Neutron Physics und Reactor Technology (INR), till.glage9@kit.edu

3. Matrix Inversion

- What we need:

$$\mathcal{B}_n := (2\mathbf{I}_n - \mathbf{S}_n)^{-1} \quad (12)$$

$$\mathcal{B}_{i,j} = \begin{cases} \frac{(-1)^{i+k}}{\det(\mathcal{B}^{-1})} \cdot \Delta_{j-1} \cdot \prod_{k=j+1}^{i} \widetilde{\delta}_k^- \cdot \nabla_{i+1} & n > i > j > 2 \\ \frac{\Delta_{i-1} \cdot \nabla_{i+1}}{\det(\mathcal{B}^{-1})} & n > i = j > 1 \\ \frac{(-1)^{i+j}}{\det(\mathcal{B}^{-1})} \cdot \Delta_{i-1} \cdot \prod_{k=i}^{j-1} \widetilde{\delta}_k^+ \cdot \nabla_{j+1} & n > j > i > 1 \end{cases} \quad (13)$$

3. Matrix Inversion

- How formula (13) was deducted:
  - What we know:

$$\mathcal{B}_{i,j} = \frac{\det(b_1, \ldots, b_{i-1}, e_j, b_{i+1}, \ldots, b_n)}{\det(\mathcal{B}^{-1})} \quad (14)$$

  - So, all we need are two (or three) determinant expansions!

- Laplace's Expansion!

3. Matrix Inversion

- The two determinants used in the equation for the inversion:
  - “Forward” expansion (not to be confused with solvers!)

$$\Delta_1 = 2 \qquad \Delta_2 = 2 \cdot \widetilde{\delta}^* + \widetilde{\delta}_2^- \qquad \Delta_n = \widetilde{\delta}^* \cdot \Delta_{n-1} - \widetilde{\delta}_n^- \cdot \widetilde{\delta}_{n-1}^+ \cdot \Delta_{n-2} \tag{15}$$

  - “Backward” expansion

$$\nabla_n = 1 \qquad \nabla_{n-1} = \widetilde{\delta}^* \qquad \nabla_k = \widetilde{\delta}^* \cdot \nabla_{k+1} - \widetilde{\delta}_{k+1}^- \widetilde{\delta}_k^+ \cdot \nabla_{k+2} \tag{16}$$

$$\begin{vmatrix} 2 & -1 & 0 & 0 & 0 \\ \widetilde{\delta}_1^- & \widetilde{\delta}^* & \widetilde{\delta}_1^+ & 0 & 0 \\ 0 & \widetilde{\delta}_2^- & \widetilde{\delta}^* & \widetilde{\delta}_2^+ & 0 \\ 0 & 0 & \widetilde{\delta}_3^- & \widetilde{\delta}^* & \widetilde{\delta}3^+ \\ & & & \ddots & \ddots \end{vmatrix}$$

3. Matrix Inversion

- An example on how the equation (13) was deducted:

$$
\begin{vmatrix}
\ddots & \ddots & & & & & & & & \\
\widetilde{\delta}^-_{i-2} & \widetilde{\delta}^* & \widetilde{\delta}^+_{i-2} & & & & & & & \\
& \widetilde{\delta}^-_{i-1} & \widetilde{\delta}^* & \not{0} & & & & & & \\
& & \widetilde{\delta}^-_{i} & \not{0} & \widetilde{\delta}^+_{i} & & & & & \\
& & & \vdots & \ddots & \ddots & & & & \\
& & & & \widetilde{\delta}^-_{j-2} & \widetilde{\delta}^* & \widetilde{\delta}^+_{j-2} & & & \\
& & & & & \widetilde{\delta}^-_{j-1} & \widetilde{\delta}^* & \widetilde{\delta}^+_{j-1} & & \\
& & & \not{1} & & & \not{\widetilde{\delta}^-_{j}} & \not{\widetilde{\delta}^*} & \not{\widetilde{\delta}^+_{j}} & \\
& & & & & & & \widetilde{\delta}^-_{j+1} & \widetilde{\delta}^* & \widetilde{\delta}^+_{j+1} \\
& & & & & & & & \ddots & \ddots
\end{vmatrix}
\quad (17)
$$

 Till Glage, Institute for Neutron Physics und Reactor Technology (INR), till.glage9@kit.edu

3. Matrix Inversion

- Sub-determinant:

$$= (-1)^{i+j} \begin{vmatrix} \ddots & \ddots & & & & & & & \\ \widetilde{\delta}^{-}_{i-2} & \widetilde{\delta}^{*} & \widetilde{\delta}^{+}_{i-2} & & & & & & \\ & \widetilde{\delta}^{-}_{i-1} & \widetilde{\delta}^{*} & & & & & & \\ & & \widetilde{\delta}^{-}_{i} & \widetilde{\delta}^{+}_{i} & & & & & \\ & & & \ddots & \ddots & & & & \\ & & & \widetilde{\delta}^{-}_{j-2} & \widetilde{\delta}^{*} & \widetilde{\delta}^{+}_{j-2} & & & \\ & & & & \widetilde{\delta}^{-}_{j-1} & \widetilde{\delta}^{*} & \widetilde{\delta}^{+}_{j-1} & & \\ & & & & & & \widetilde{\delta}^{-}_{j+1} & \widetilde{\delta}^{*} & \widetilde{\delta}^{+}_{j+1} \\ & & & & & & & \ddots & \ddots \end{vmatrix} \qquad (18)$$

3. Matrix Inversion

- An intermediate result: $$\Delta_{i-1} \cdot \prod_{\text{diag. elements}} \widetilde{\delta}^{+} \cdot \nabla_{j+1} \tag{19}$$

$$\mathcal{B}_{i,j} = \begin{cases} \frac{(-1)^{i+k}}{\det(\mathcal{B}^{-1})} \cdot \Delta_{j-1} \cdot \prod_{k=j+1}^{i} \widetilde{\delta}_k^{-} \cdot \nabla_{i+1} & n > i > j > 2 \\ \frac{\Delta_{i-1} \cdot \nabla_{i+1}}{\det(\mathcal{B}^{-1})} & n > i = j > 1 \\ \frac{(-1)^{i+j}}{\det(\mathcal{B}^{-1})} \cdot \Delta_{i-1} \cdot \prod_{k=i}^{j-1} \widetilde{\delta}_k^{+} \cdot \nabla_{j+1} & n > j > i > 1 \end{cases} \tag{13}$$

- However: still problem with the boundary elements of the first/last rows columns. (Not further discussed)

4. Comparison to other Inversion Method

- The idea of a backward solver is not new
- Inversion has always been the disadvantage of backward solvers

- The numerical method by Axel von der Weth (presentation at this conference)
  - Advantages
    - Numerically stable
    - Not limited to tridiagonal matrices
  - Disadvantages
    - Long execution times (4s to hours)
    - Problem with the initial values

5. Eigenvalue Investigation

- Why is that interesting?

→ One iteration can be rewritten as:

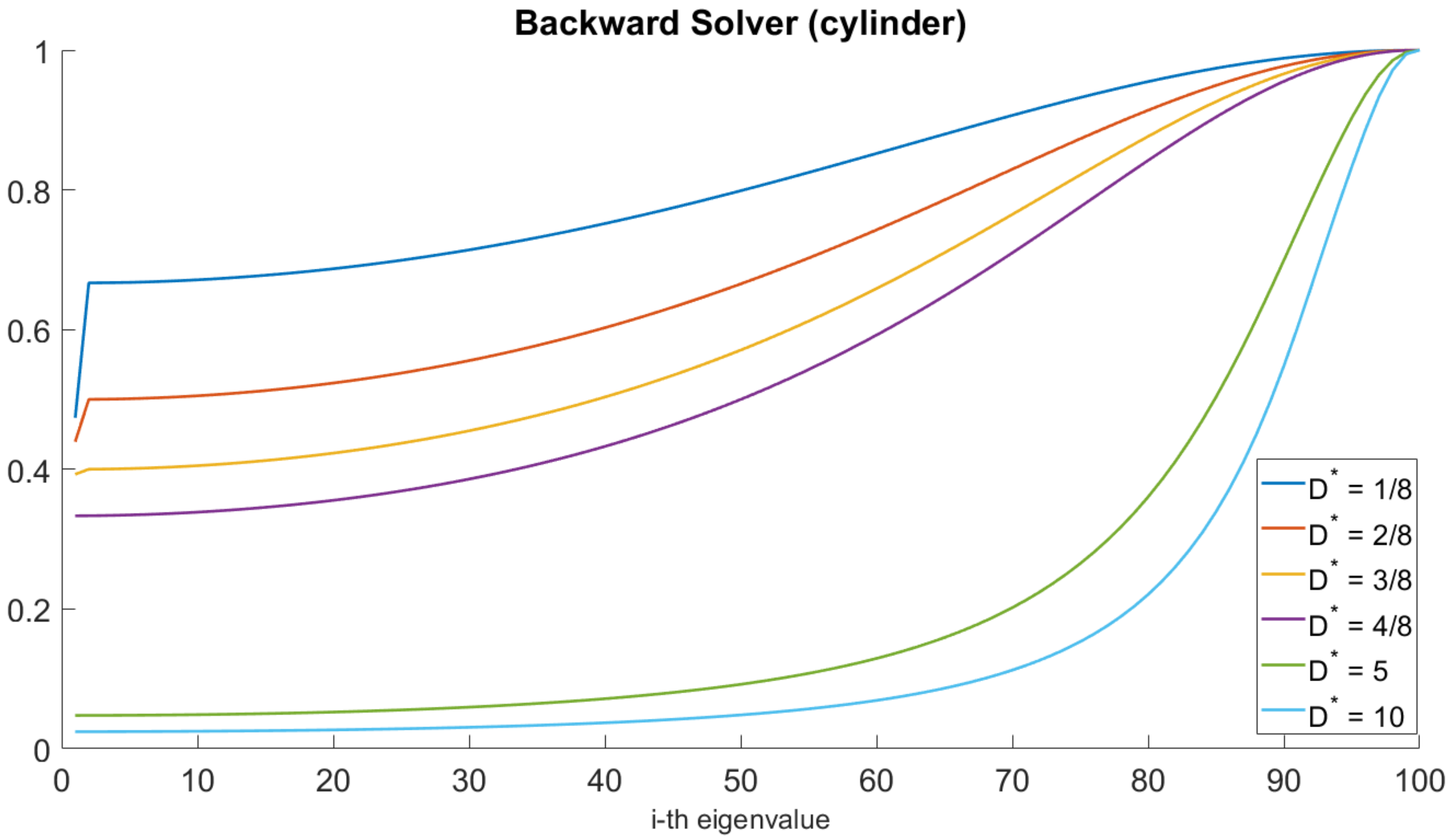
$$\mathbf{S} \cdot c = \mathbf{S} \cdot (\alpha_1 q_1 + \cdots + \alpha_n q_n) = \lambda_1 \alpha_1 q_1 + \cdots + \lambda_n \alpha_n q_n \quad (20)$$

- Parts of the QR algorithm:
  - Using orthogonal transformations to get a similar upper-triangular matrix

- Disclaimer: Eigenvalues give information about **stability, <u>not</u> accuracy**!

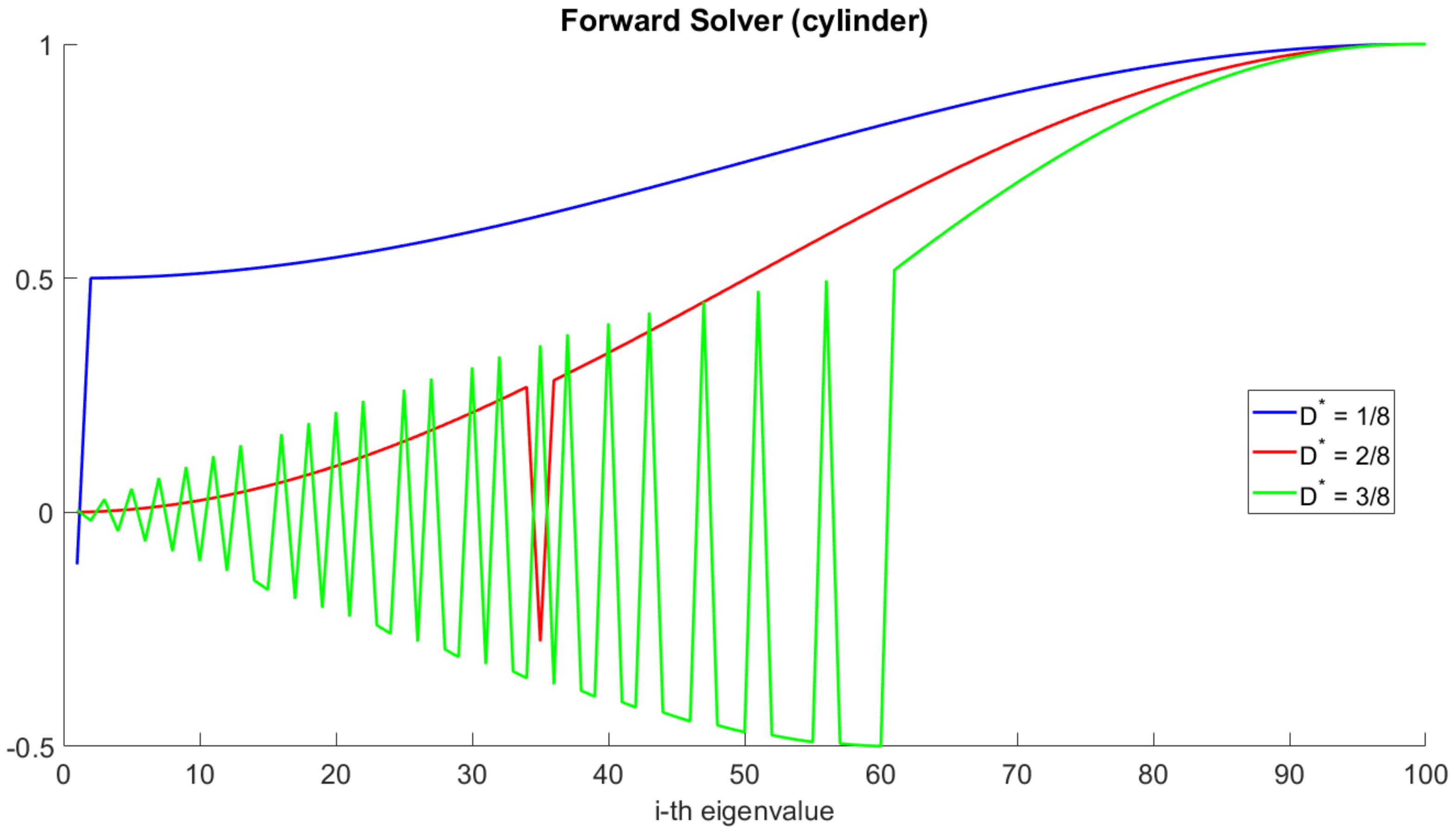
5. Eigenvalue Investigation

- Eigenvalues from the QR algorithm



5. Eigenvalue Investigation

- Forward solver as a reference



6. Properties of Different Solvers and Comparison

- There is a variety of solvers for such problems:
  - Euler forward
  - Euler backward
  - Combination of the previous two ones
  - Crank-Nicolson
- Both geometries: cartesian, cylindrical for the first two solvers

- The question which solver you should use will be addressed by Axel von der Weth in his presentation at this conference.

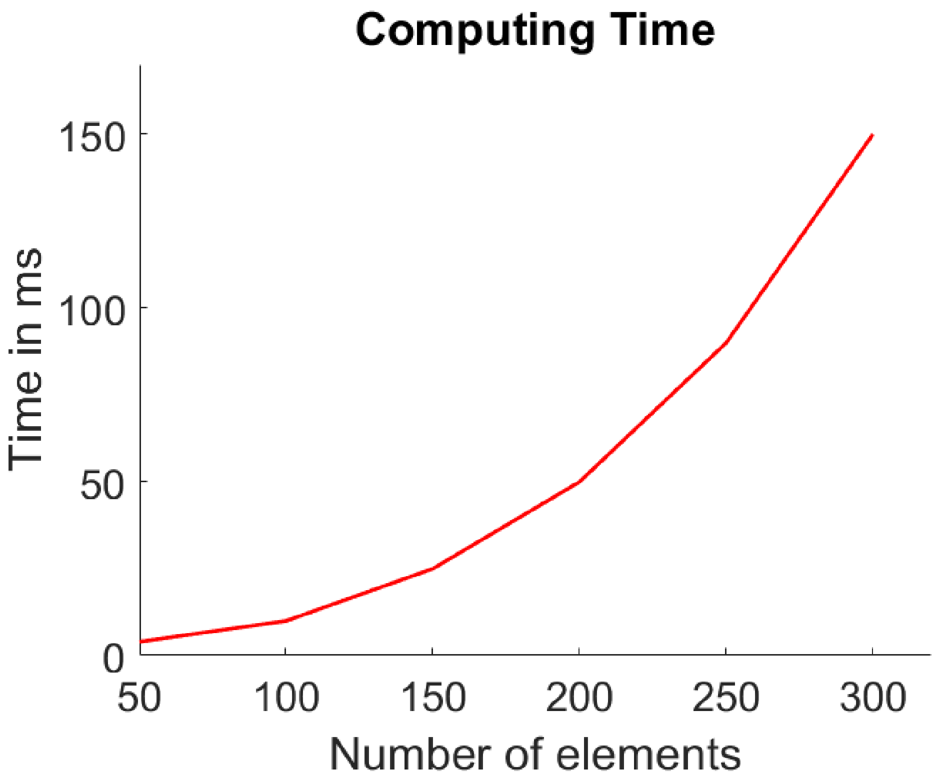
6. Properties of Different Solvers and Comparison

- Forward solver:
  - Limited D* value for both coordinate systems
  - Has a error minimum according to Axel von der Weth’s research (only for Cartesian solver; no minimum for cylindrical coordinates)

- Question: Why is the backward solver sensible?
  - Enables us to use arbitrary D* values
  - The experimental data could be processed with the same sample rate as they are measured
  - However: $D^* \propto$ error $\rightarrow$ suitable configuration needed

6. Properties of Different Solvers and Comparison

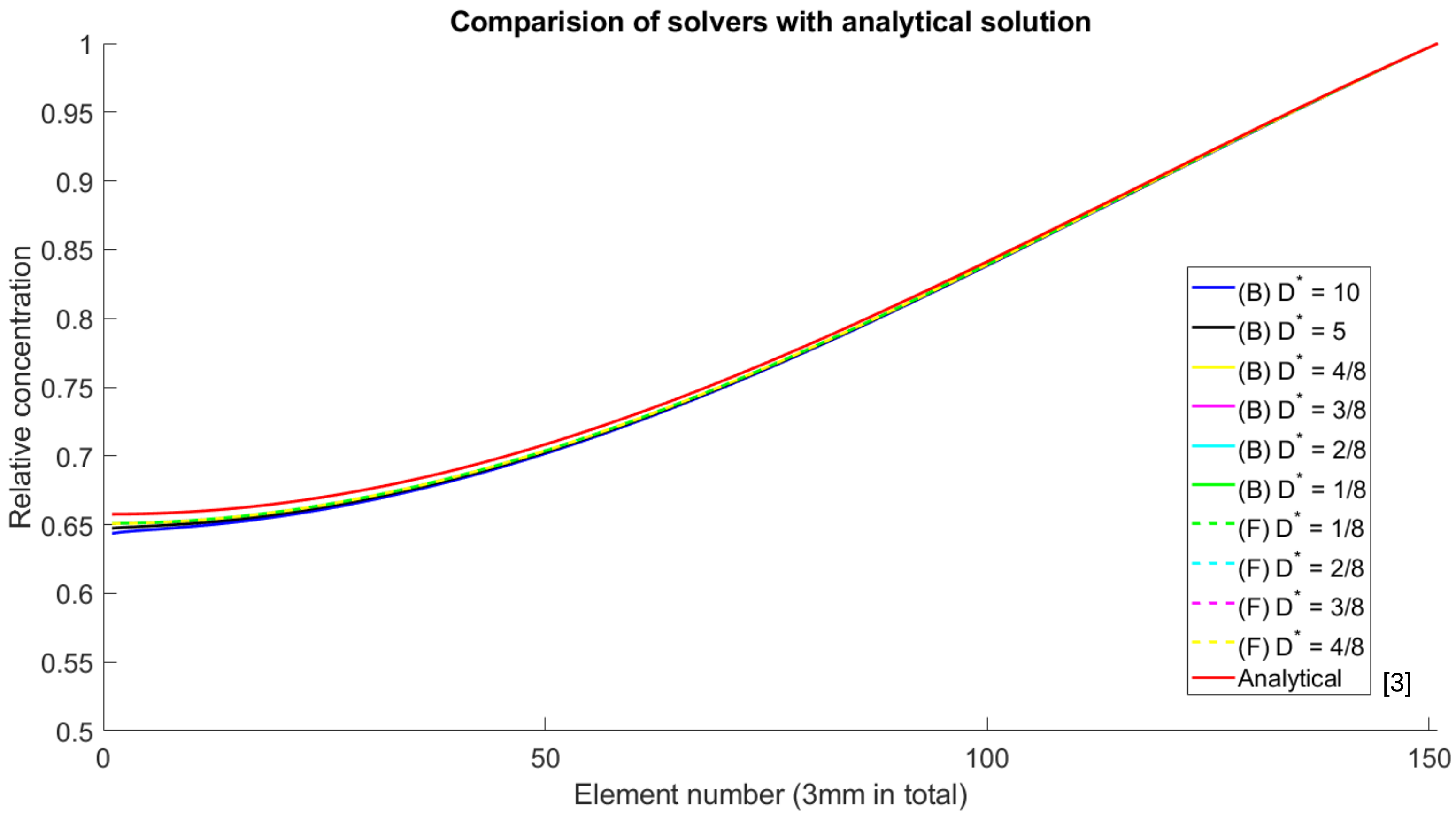
- Former approaches: not usable (10k curves have to be fitted)
  - Now: better chances since lower computational effort:
  - 50x50: 4ms; 100x100: 10ms; 150x150: 24ms; 200x200: 50ms



Main Frame: BWUnicluster 2

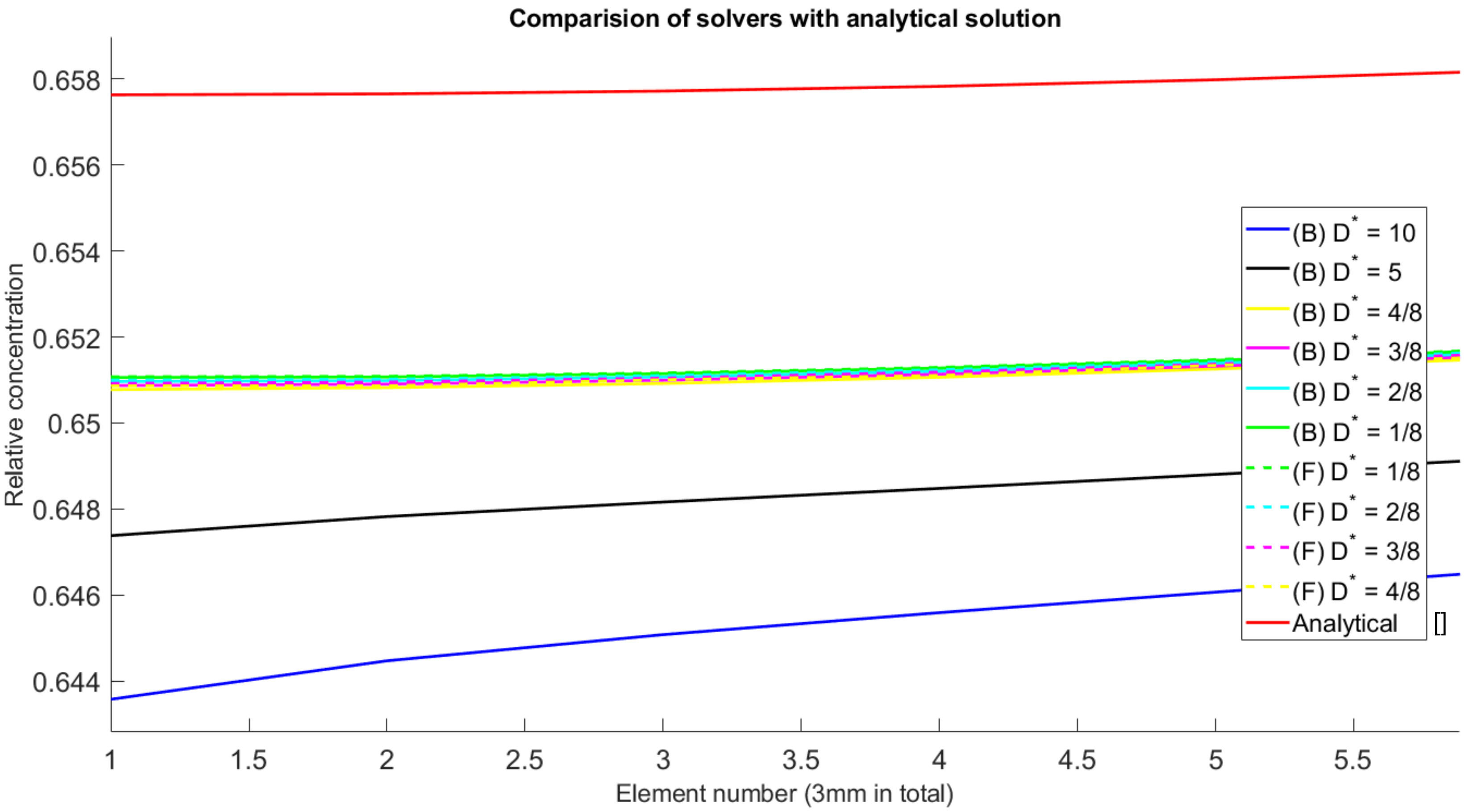
6. Properties of Different Solvers and Comparison

- Simulation of a loading phase (300s) e.g. in gas release experiment
- Rel. dev. 4.6*10^-3; Higher D* → larger error (fewer multiplications ; rougher discretization)



[3] Private communication Marvin R. Schulz

6. Properties of Different Solvers and Comparison



7. Outlook

- Where can this method be applied?
  - All Tridiagonal-solvers
    - Backward
    - Crank-Nicolson (consists of an inverted tridiagonal matrix and a not inverted one multiplied together)
    - Combined Solver
  - In general mathematical problems

Thank you for your attention!

Acknowledgments: The authors acknowledge support by the state of Baden-Württemberg through bwHPC