An open source implementation of the IDR(S)Stab(L) solver

Citation for published version (APA):

Document status and date:
Published: 30/10/2019

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 13. May. 2024
1. Introduction

- Linear systems are the backbone of simulations.
- For small systems the robust LU-decomposition is often used.
- Linear systems in 2D and 3D simulations are too large to tackle with this method.
- An alternative to LU is to use iterative methods, for example BiCGStab in the Eigen C++ library [1].
- IDR(S)Stab(L) is a generalization of BiCGStab [2] and is applicable to a wider range of systems.
- Here the Eigen-implementation of BiCGStab is compared with our implementation of IDR(S)Stab(L). For matrices from the Matrix Market [3] and a 2D model problem.

2. Setup

Computational setup:
- Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz 6C/12T
- GCC 7.4.1, Eigen 3.3.7, optimization flags: -fopenmp, -O3 -march=native

2D model problem:
Electrons accelerated toward a plate with a layer of neutral gas, compute e-density $n_e$.
Parameters: Advection $u$, diffusion $\varepsilon$, influx $s$, reaction $\beta$.
Length=1 $\rightarrow$ non-dimensional $\beta L^2/\varepsilon$ (Darmkohler) and $uL/\varepsilon$ (Peclet) numbers.

\[ \mathbf{E} = \mathbf{E} \mathbf{c}, \quad n_e = 0 \]

\[ \mathbf{u} = u \mathbf{e}_x + u \mathbf{e}_y \]

Discretization:
- Finite Volume Method (FVM)
- Exponential scheme / Homogeneous flux scheme [4]
- 161x161 grid with gridspacing $h$
- Resulting scheme:
  \[ \left[ \frac{1}{2} u - \frac{1}{2h} \mathbf{E} \mathbf{Q} \right] n_{i,j+1} + \left[ \frac{1}{2} u - \frac{1}{2h} \mathbf{E} \mathbf{Q} \right] n_{i,j+1} + \]
  \[ \left[ \beta \mathbf{E} \mathbf{Q} - \beta h \right] n_{i,j} + \left[ \frac{1}{2} u - \frac{1}{2h} \mathbf{E} \mathbf{Q} \right] n_{i-1,j} + \left[ \frac{1}{2} u - \frac{1}{2h} \mathbf{E} \mathbf{Q} \right] n_{i,j-1} + \]
  \[ = h s \]

\[ Q := \frac{P}{e^P - 1} - \frac{P}{e^{-P} - 1}, \quad P := \frac{u h}{\varepsilon} \]

Resulting linear system:
- Solve for the points $n_{i,j}$
- 25,921 equations with 25,921 unknowns

3. Matrix Market results

<table>
<thead>
<tr>
<th>IDR(S)Stab(L)</th>
<th>BiCGStab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tolerance</td>
<td>$1e^{-12}$</td>
</tr>
<tr>
<td>Total linear systems</td>
<td>113</td>
</tr>
<tr>
<td>IDRStab converges for</td>
<td>47</td>
</tr>
<tr>
<td>BiCGStab converges for</td>
<td>37</td>
</tr>
</tbody>
</table>

4. 2D model problem results

**BiCGStab**

**IDR(S)Stab(L)**

5. Conclusion

1. IDR(S)Stab(L) can solve problems Eigen-BiCGStab implementation cannot
2. IDR(S)Stab(L) uses less matrix-vector products to reach the tolerance, however
3. current IDR(S)Stab(L) version generally not faster.

6. Outlook

1. Improve IDR(S)Stab(L) speed
2. Propose this IDR(S)Stab(L) implementation for the Eigen-Library

Acknowledgements

This research is supported by the Netherlands Organisation for Scientific Research (NWO), which is partly funded by the Ministry of Economic Affairs. The Netherlands eScience Center is funded by NWO and SURF.
Special acknowledgement to M. Senders for his work on analyzing the MATLAB implementation of IDR(S)Stab(L) in MATLAB and L. Kuijpers for his work on analyzing the MATLAB solvers.