

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

***Citation for published version (APA):***

Schoutrop, C., ten Thije Boonkkamp, J. H. M., Vijaykumar, A., & van Dijk, J. (2019). *An open source implementation of the IDR(S)Stab(L) solver.*

***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](http://www.tue.nl/taverne)

***Take down policy***

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

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

C.E.M. Schoutrop<sup>1</sup>, J.H.M. ten Thije Boonkkamp<sup>2</sup>, A. Vijaykumar<sup>3</sup>, J. van Dijk<sup>1</sup>

<sup>1</sup> Department of Applied Physics, Eindhoven University of Technology

<sup>2</sup> Department of Mathematics and Computer Science, Eindhoven University of Technology

<sup>3</sup> Netherlands eScience center

c.e.m.schoutrop@tue.nl

## 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

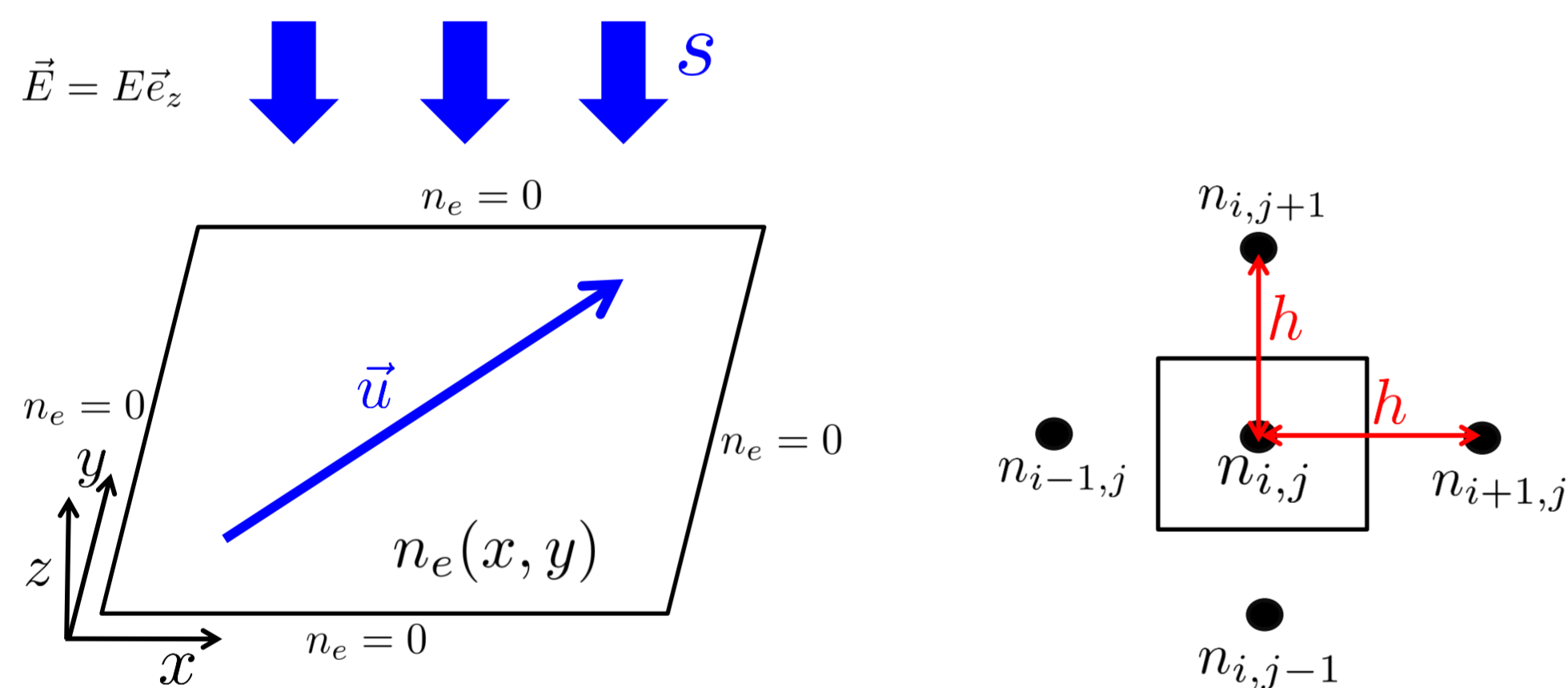
GCC 7.4.1, Eigen 3.3.7, optimization flags: -fopenmp, -O3 -march=native

### 2D model problem:

Electrons accelerated toward a plate with a layer of neutral gas, compute e<sup>-</sup> density n<sub>e</sub>.

Parameters: Advection u diffusion ε, influx s, reaction β.

Length=1 → non-dimensional βL<sup>2</sup>/ε (Damköhler) and uL/ε (Peclet) numbers.



$$-\mathcal{E}\nabla^2 n_e + \vec{u} \cdot \nabla n_e - \beta n_e = s, \quad \vec{u} = u\vec{e}_x + u\vec{e}_y$$

### Discretization:

- Finite Volume Method (FVM)
- Exponential scheme / Homogeneous flux scheme [4]
- 161x161 grid with gridspace h
- Resulting scheme:

$$\begin{aligned} & \left[ \frac{1}{2}u - \frac{1}{2h}\mathcal{E}Q \right] n_{i+1,j} + \left[ \frac{1}{2}u - \frac{1}{2h}\mathcal{E}Q \right] n_{i,j+1} + \\ & \quad \left[ \frac{2}{h}\mathcal{E}Q - \beta h \right] n_{i,j} + \\ & \left[ -\frac{1}{2}u - \frac{1}{2h}\mathcal{E}Q \right] n_{i-1,j} + \left[ -\frac{1}{2}u - \frac{1}{2h}\mathcal{E}Q \right] n_{i,j-1} + \\ & \quad = hs \end{aligned}$$

$$Q := \frac{P}{e^P - 1} - \frac{P}{e^{-P} - 1}, \quad P := \frac{uh}{\mathcal{E}}$$

### Resulting linear system:

- Solve for the points n<sub>i,j</sub>
- 25,921 equations with 25,921 unknowns

## 3. Matrix Market results

Every linear system Ax=b with rhs from the Matrix Market [3] database

Tolerance: 1e-12, max iterations: 2000 IDR(S)Stab(L), 13000 BiCGStab

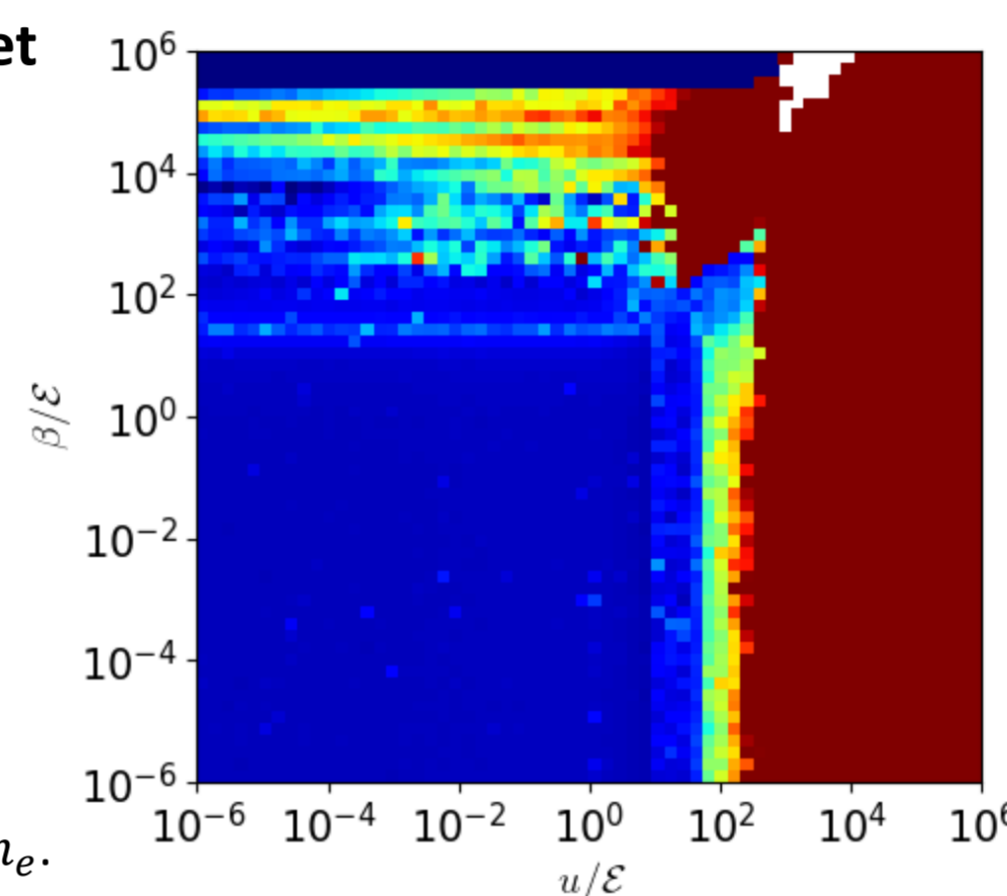
Total: 113 linear systems

**BiCGStab converges for 37**

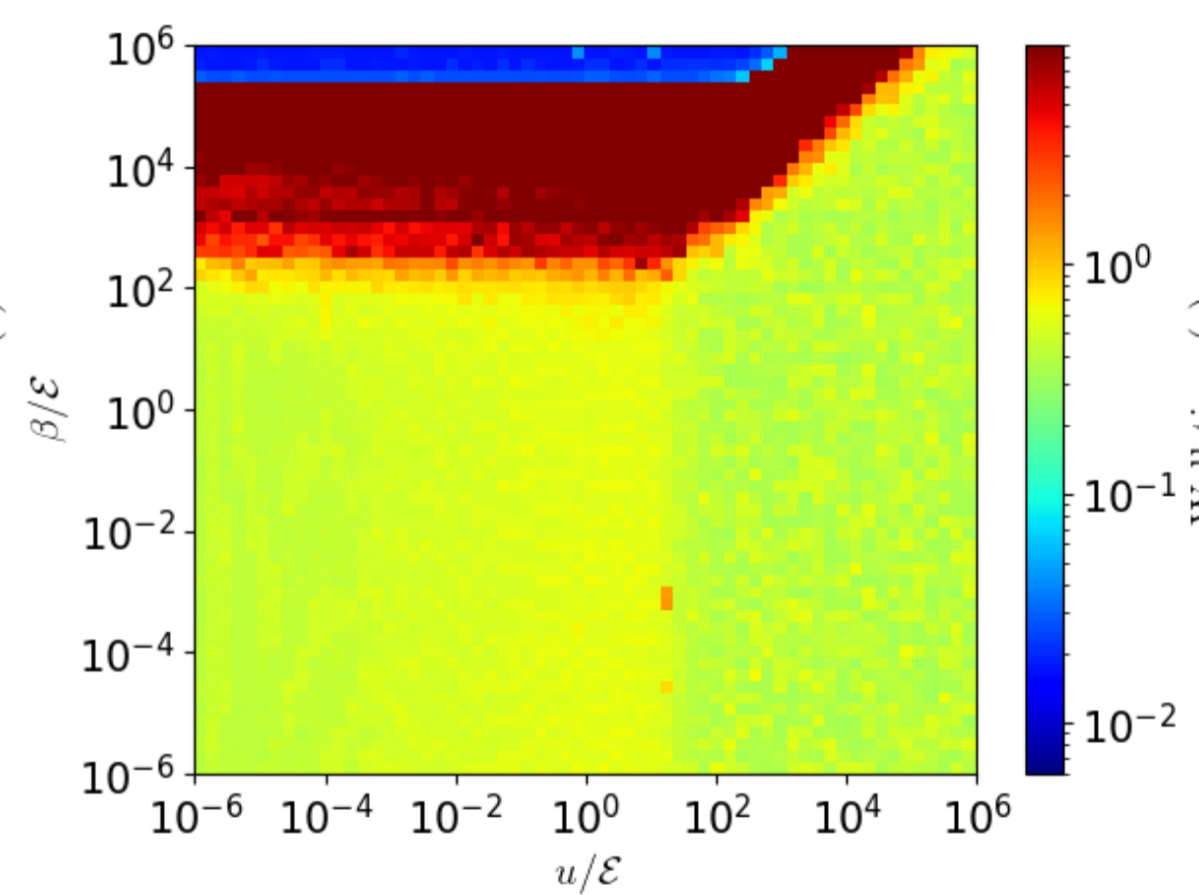
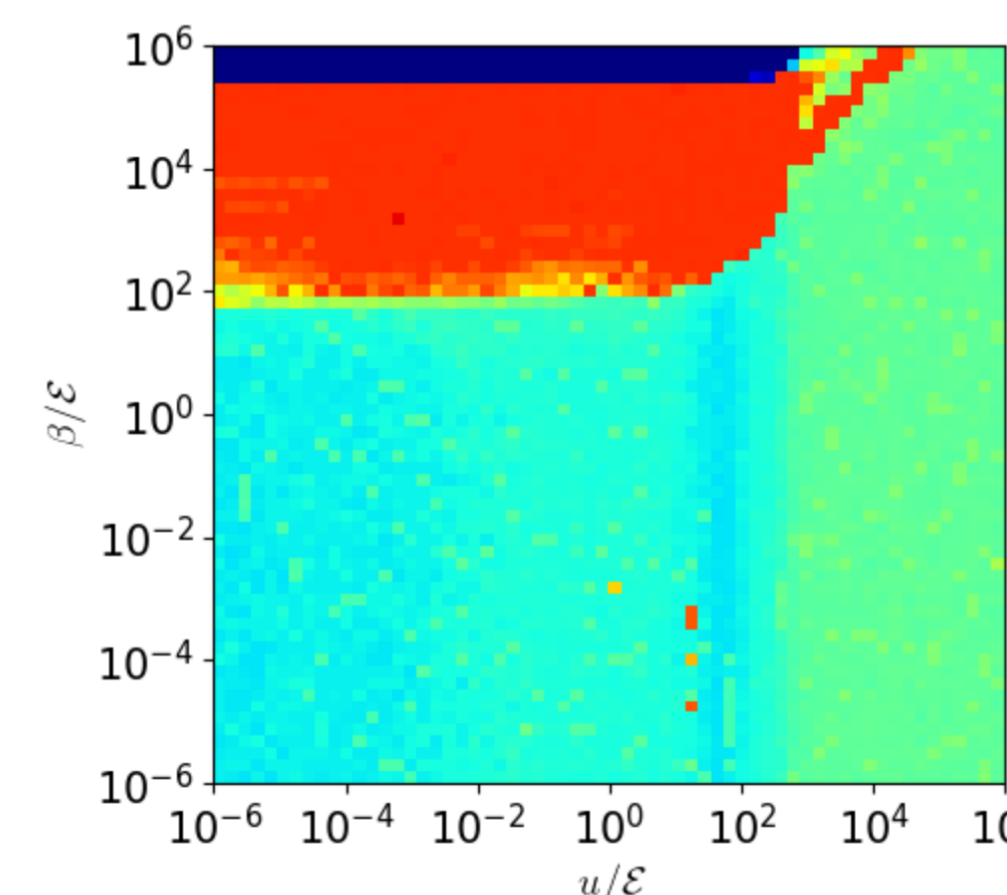
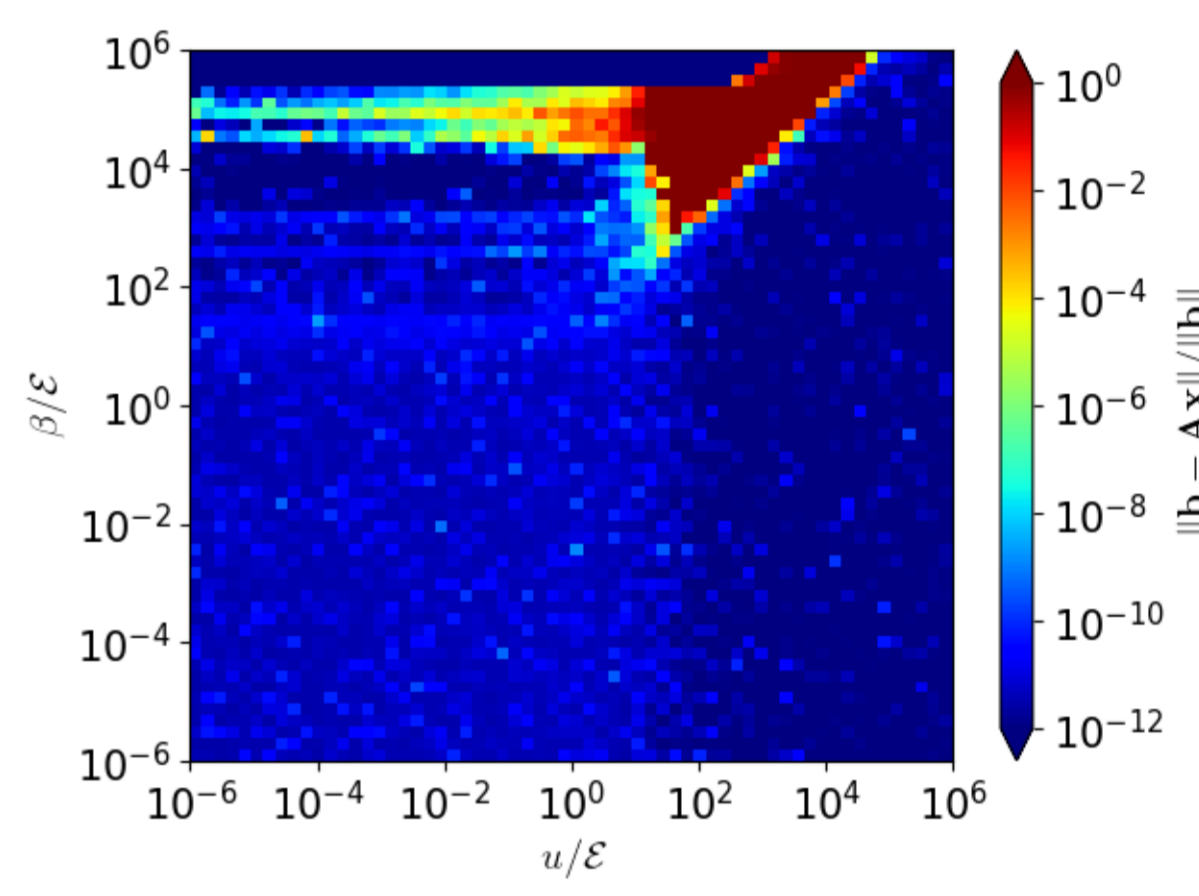
**IDRStab converges for 47**

## 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 in the framework of the JCER program. The Netherlands eScience Center is funded by NWO and SURF. **Special acknowledgement to M. Senders for his work on an initial reference implementation of IDR(S)Stab(L) in MATLAB and L. Kuijpers for his work on analyzing the MATLAB solvers.**

[1] Guennebaud, Gael, and Benoit Jacob. "Eigen: a C++ linear algebra library." URL <http://eigen.tuxfamily.org>

[2] Aihara, Kensuke, Kuniyoshi Abe, and Emiko Ishiwata. "A variant of IDRStab with reliable update strategies for solving sparse linear systems." Journal of Computational and Applied Mathematics

[3] Boisvert, Ronald F., et al. "Matrix market: a web resource for test matrix collections." Quality of Numerical Software. Springer, Boston, MA, 1997. 125-137.

[4] Thije Boonkkamp, ten, J. H. M., & Anthonissen, M. J. H. (2010). The finite volume-complete flux scheme for advection-diffusion-reaction equations.