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

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

C.E.M. Schoutrop[1] , J.H.M. ten Thije Boonkkamp[2] , A. Vijaykumar[3], J. van Dijk[1]

[1] Departement of Applied Physics, Eindhoven University of Technology

[2] Departement of Mathematics and Computer Science, Eindhoven University of Technology

[3] 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

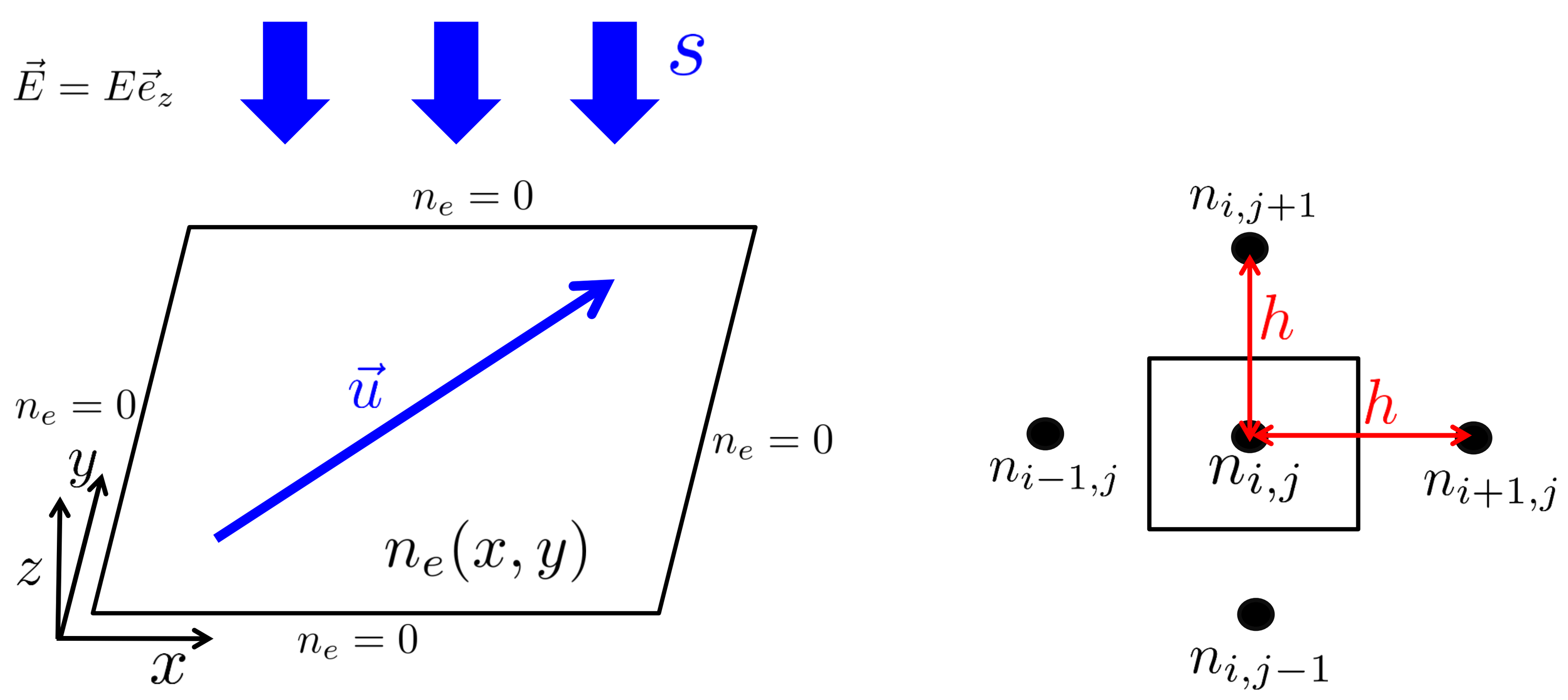
GCC 7.4.1, Eigen 3.3.7, optimization flags: -fopenmp, -O3 –march=native

2D model problem:

Electrons accelerated toward a plate with a layer of neutral gas, compute e- density n_e.

Parameters: Advection u diffusion $\mathcal{E}$, influx s, reaction β.

Length=1 → non-dimensional $\beta L^2/\mathcal{E}$ (Damköhler) and $uL/\mathcal{E}$ (Peclet) numbers.



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

Discretization:

-Finite Volume Method (FVM)

-Exponential scheme / Homogeneous flux scheme [4]

-161x161 grid with gridspacing h

-Resulting scheme:

$$\left[\tfrac{1}{2}u - \tfrac{1}{2h}\mathcal{E}Q\right] n_{i+1,j} + \left[\tfrac{1}{2}u - \tfrac{1}{2h}\mathcal{E}Q\right] n_{i,j+1} +$$
$$\left[\tfrac{2}{h}\mathcal{E}Q - \beta h\right] n_{i,j} +$$
$$\left[-\tfrac{1}{2}u - \tfrac{1}{2h}\mathcal{E}Q\right] n_{i-1,j} + \left[-\tfrac{1}{2}u - \tfrac{1}{2h}\mathcal{E}Q\right] n_{i,j-1} +$$
$$= hs$$

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

Resulting linear system:

-Solve for the points $n_{i,j}$

-25,921 equations with 25,921 unknowns

3. Matrix Market results

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

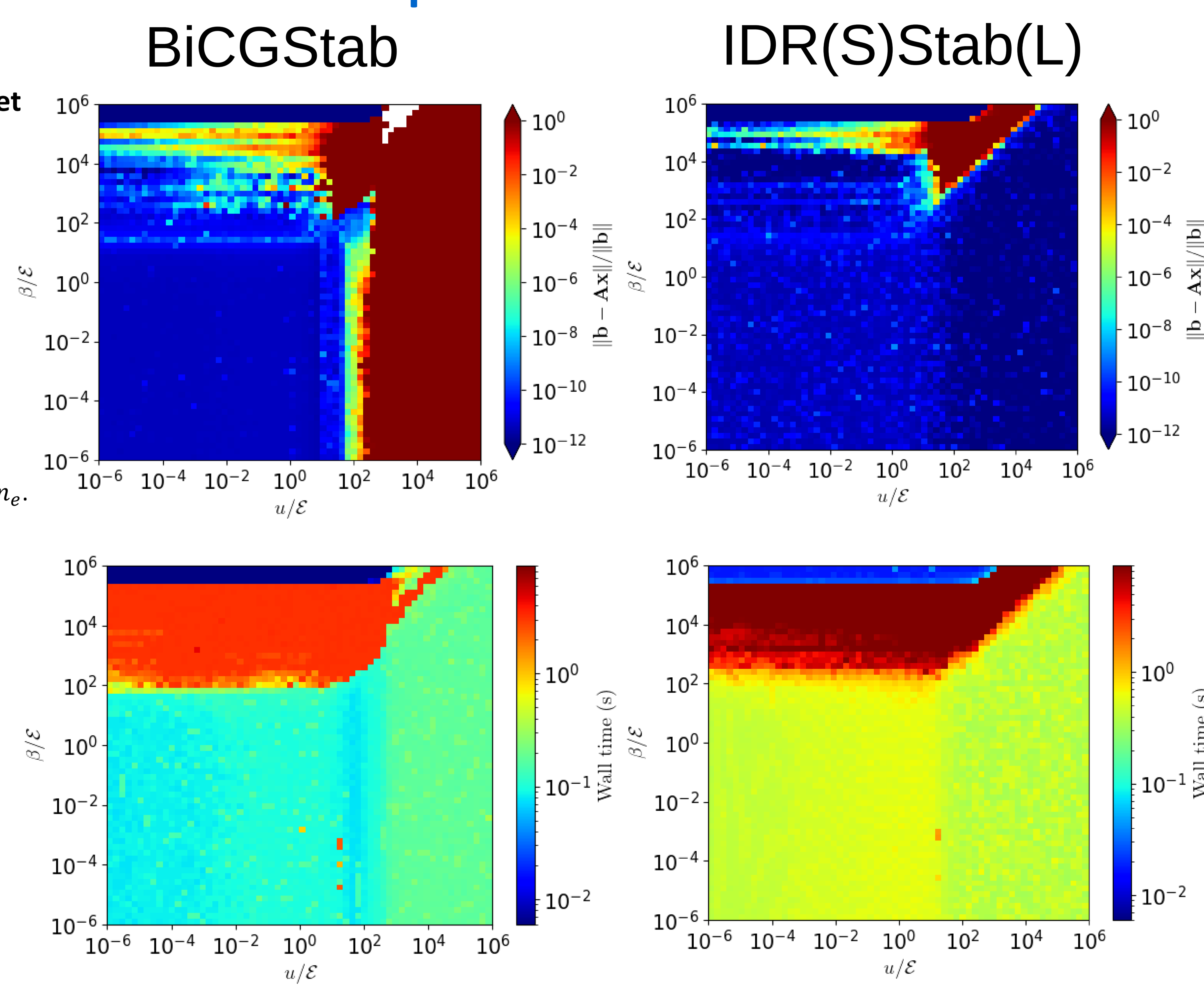
Tolerance: 1e-12, max iterations: 2000 IDR(S)Stab(L), 13000 BiCGStab

Total: 113 linear systems

BiCGStab converges for 37

IDRStab converges for 47

4. 2D model problem results



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 Ministery 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.