Newton Solver Stabilization for Stokes Solvers in Geodynamic Problems

Fraters M. R. T. (1), Bangerth W. (2), Thieulot C. (1), Glerum A. (1,3), Spakman W. (1,4)

(1) Department of Earth Sciences, Utrecht University, Netherlands, (2) Department of Mathematics, Colorado State University, U.S.A, (3) GFZ German Research Center for Geosciences, Potsdam, Germany, (4) Centre of Earth Evolution and Dynamics (CEED), University of Oslo, Norway

Universiteit Utrecht

Colorado State University

NWO Netherlands Organisation for Scientific Research

Abstract

1. We have implemented a Newton solver for ASPECT (Kronbichler et al., 2012)
2. We only use solver libraries to solve linear systems (no Trilinos NOX or PETSC SNES)
3. We have implemented line search and oversolving-prevention ourselves
4. The Jacobian is not always Symmetric Positive Definite (SPD)
5. We force the Jacobian to be SPD in a cheap and optimal way
6. This allows for more complex rheologies to be used with a Newton solver

Problem Statement

We are interested solving the Stokes equations:

$$-\nabla\cdot\left[2\eta\left(\dot{\varepsilon}(\boldsymbol{u})-\frac{1}{3}(\nabla\cdot\boldsymbol{u})\mathbf{1}\right)\right]+\nabla p=\rho\boldsymbol{g} \qquad \text{in } \Omega,$$

$$\nabla\cdot(\rho\boldsymbol{u})=0 \qquad \text{in } \Omega,$$

The weak form of the Newton linearisation:

$$\begin{pmatrix} J_k^{uu} & J_k^{up} \\ J_k^{pu} & 0 \end{pmatrix}\begin{pmatrix}\delta U_k \\ \delta \bar{P}_k\end{pmatrix}=\begin{pmatrix}F_k^u \\ F_k^p\end{pmatrix}$$

where for the incompressible case the Jacobian elements are (ignoring the pressure scaling):

$$\left(J_k^{uu}\right)_{ij}=(A_k)_{ij}+\left(\varepsilon(\varphi_i^u),2\left(\frac{\partial\eta(\varepsilon(\boldsymbol{u}_k),p_k)}{\partial\varepsilon}:\varepsilon(\varphi_j^u)\right)\varepsilon(\boldsymbol{u}_k)\right),$$

$$\left(J_k^{up}\right)_{ij}=B_{ij}^T+\left(\varepsilon(\varphi_i^u),2\left(\frac{\partial\eta(\varepsilon(\boldsymbol{u}_k),p_k)}{\partial p}\varphi_j^p\right)\varepsilon(\boldsymbol{u}_k)\right),$$

$$\left(J_k^{pu}\right)_{ij}=B_{ij}.$$

This is in general neither Symmetric, nor Positive Definite, which can be very bad for solvers.

Restoring Symmetry for Preconditioning

Approximate Jacobian with $J_k^{up}\approx B^T$ and

$$\left(J_k^{uu}\right)_{ij}\approx(A_k)_{ij}+\left(\varepsilon(\varphi_i^u),\left(\frac{\partial\eta(\varepsilon(\boldsymbol{u}_k),p_k)}{\partial\varepsilon}:\varepsilon(\varphi_j^u)\right)\varepsilon(\boldsymbol{u}_k)\right)+\left(\varepsilon(\varphi_j^u),\left(\frac{\partial\eta(\varepsilon(\boldsymbol{u}_k),p_k)}{\partial\varepsilon}:\varepsilon(\varphi_i^u)\right)\varepsilon(\boldsymbol{u}_k)\right)$$

$$=(A_k)_{ij}+\left(\varepsilon(\varphi_i^u),E^{\text{sym}}(\varepsilon(\boldsymbol{u}_k))\varepsilon(\varphi_j^u)\right)$$

where $E^{\text{sym}}=\frac{1}{2}\left(E_{mnpq}+E_{pqmn}\right)$ and $E(\varepsilon(\boldsymbol{u}_k))_{mnpq}=\left[2\varepsilon(\boldsymbol{u})_{mn}\frac{\partial\eta(\varepsilon(\boldsymbol{u}),p)}{\partial\varepsilon_{pq}}\right]$.

Restoring Positive Definiteness

The positive definiteness of the Jacobian is determined by tensor H:

$$J^{uu}=\left(\varepsilon(\varphi_i^u),2\eta(\varepsilon(u))\varepsilon(\varphi_j^u)\right)+\left(\varepsilon(\varphi_i^u),E^{\text{sym}}(\varepsilon(u))\varepsilon(\varphi_j^u)\right)$$

$$=\left(\varepsilon(\varphi_i^u)\underbrace{\left[2\eta(\varepsilon(u))I\otimes I+E^{\text{sym}}(\varepsilon(u))\right]}_{=:H}\right.$$

We can force tensor H to be SPD by scaling E^{sym} with a factor α:

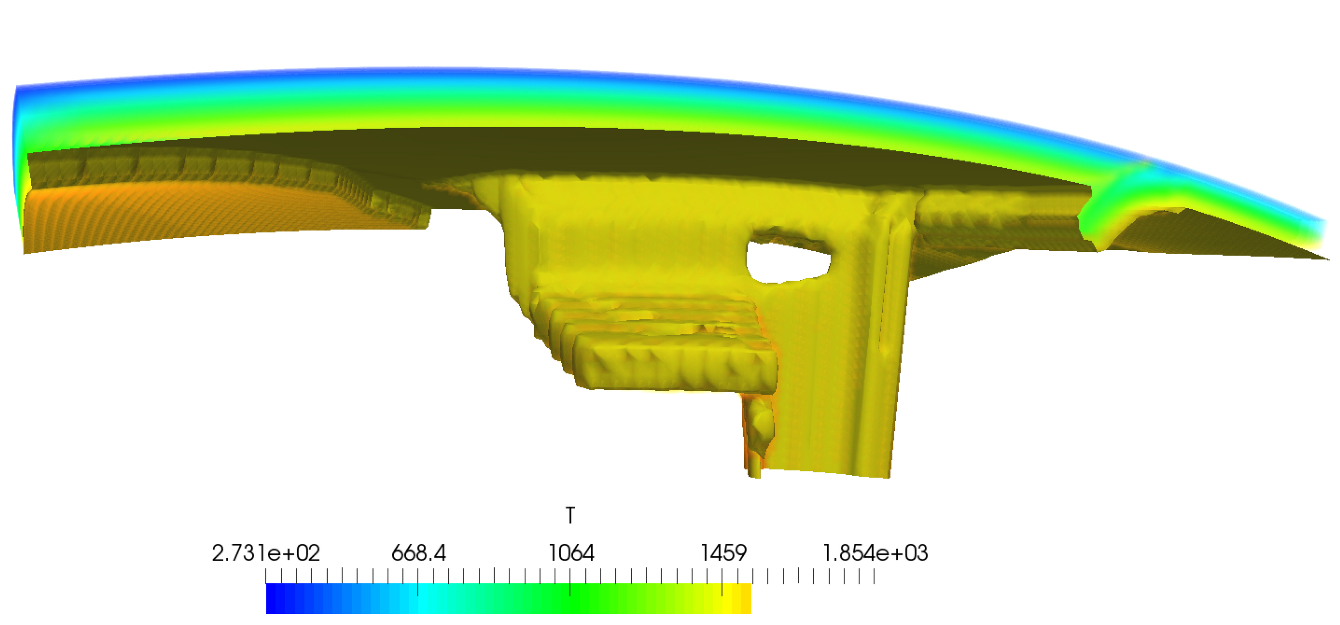
$$H^{\text{spd}}=2\eta(\varepsilon(u))I\otimes I+\alpha E^{\text{sym}}(\varepsilon(u))$$

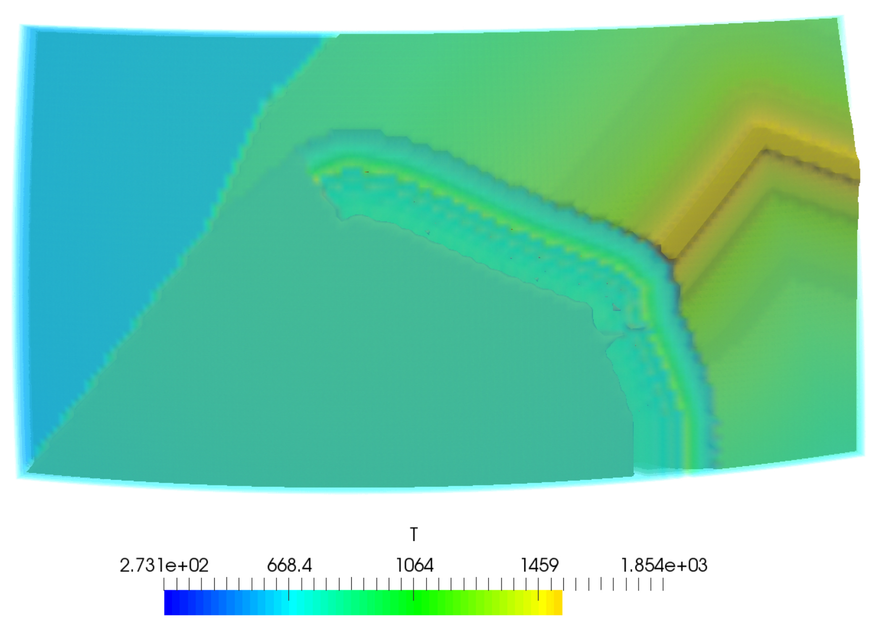
with $0<\alpha\leq 1$. If $\alpha=0$ the Jacobian will always be SPD, but to have optimal convergence we want α to be as large as possible (max one). It can be proven (Fraters et al., in prep) that the optimal value for α is (where $a=\varepsilon(\boldsymbol{u})$and $b=\frac{\partial\eta(\varepsilon(\boldsymbol{u}),p)}{\partial\varepsilon}$):

$$\alpha=\begin{cases}1 & \text{if } \left[1-\frac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\|<2\eta(\varepsilon(\boldsymbol{u}))\\ \dfrac{2\eta(\varepsilon(\boldsymbol{u}))}{\left[1-\frac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\|} & \text{otherwise.}\end{cases}$$

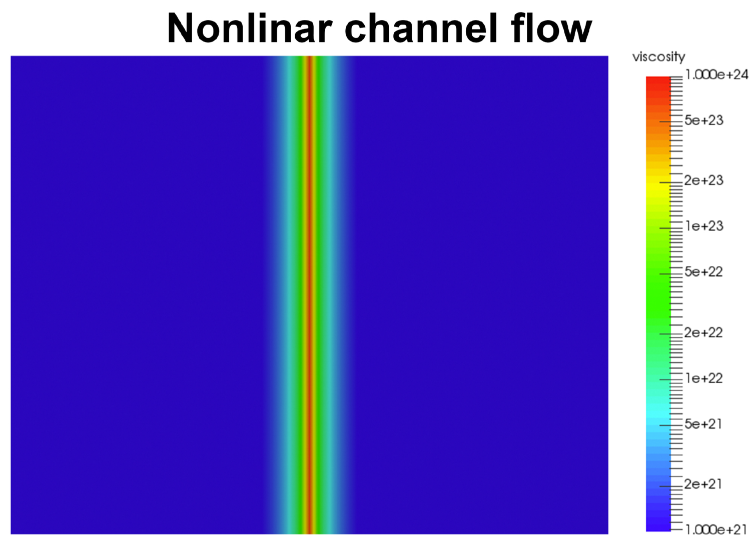
Demonstrating the World Generator

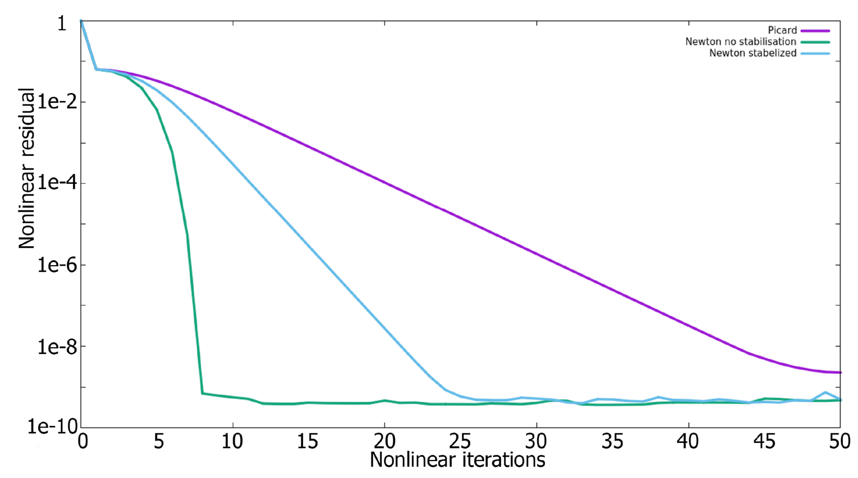
We are now testing the Newton solver on complex 3D geodynamic settings, such as constructed by our World Generator (Fraters et al., in prep):

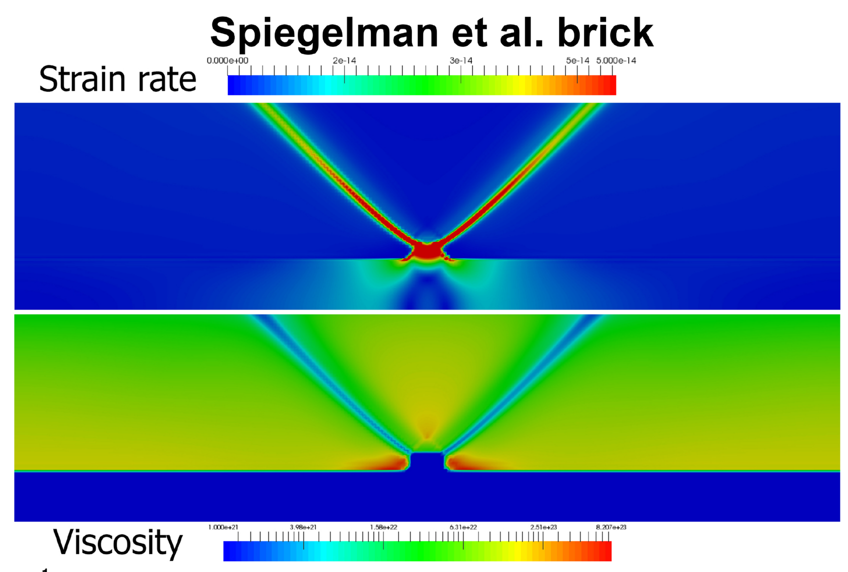


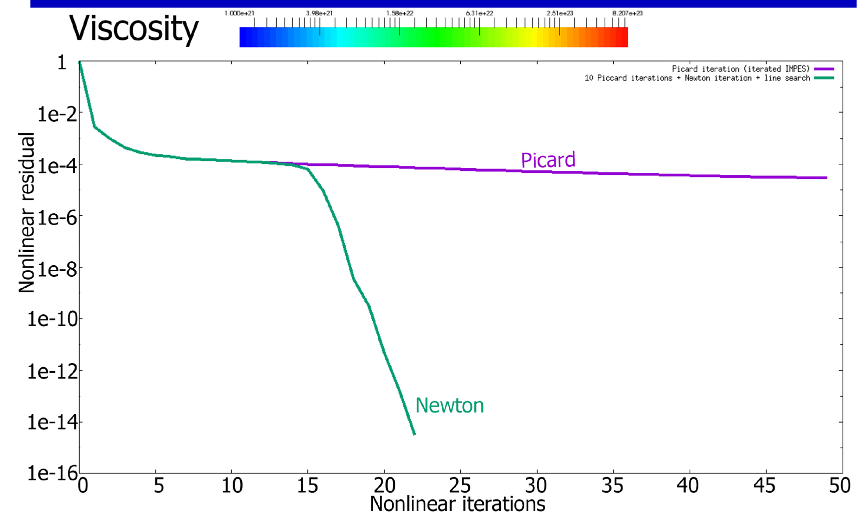


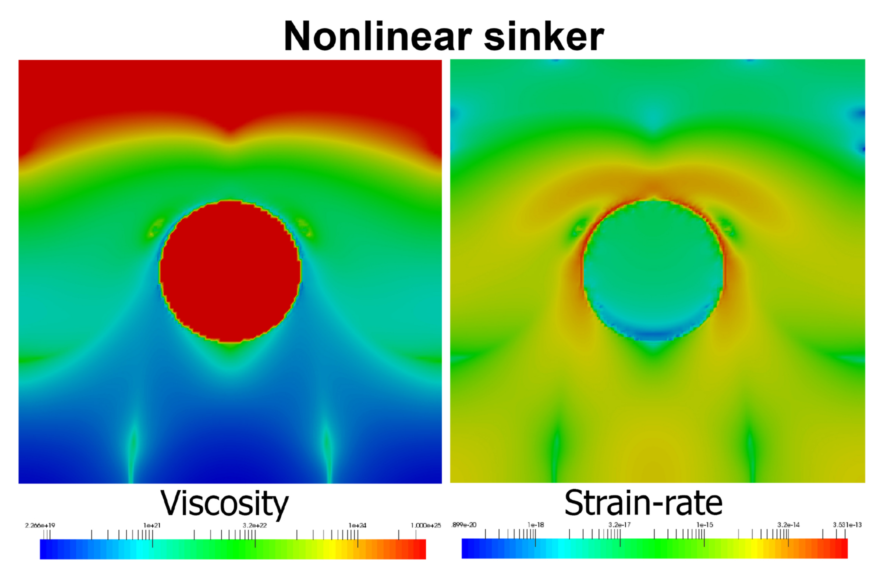
Benchmarking

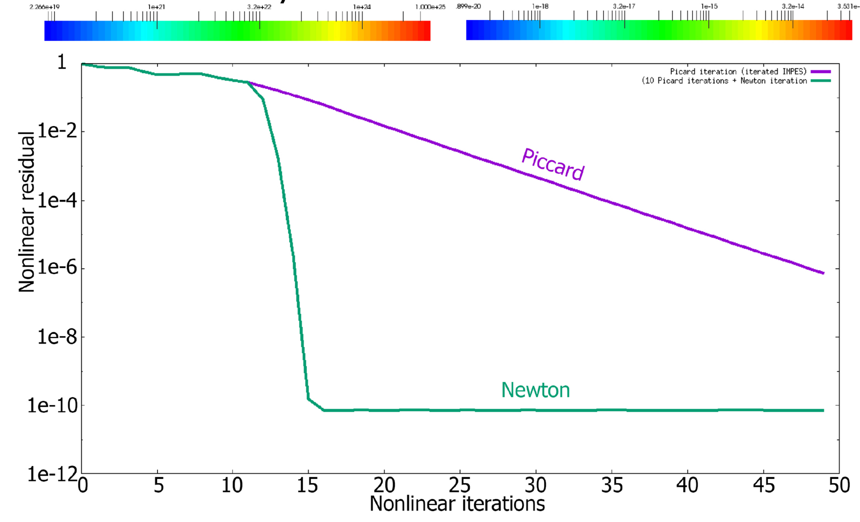












Preliminary Conclusions

The Newton solver without stabilisation is:

- fast;
- prone to numerical breakdowns;
- very senstitive to the preciece tweaking of parameters such as minimun linear tolerance, line search iterations, etc.

The Newton solver with stabilisation is:

- faster than Picard, slower than without stabilisation
- almost immune to numerical breakdowns.
- very insensitive to tweaking of these parameters.