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. We found that 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
7. Our Newton solver also works for compressible models

Problem Statement

We are interested solving the Stokes equations:

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

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

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:

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

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

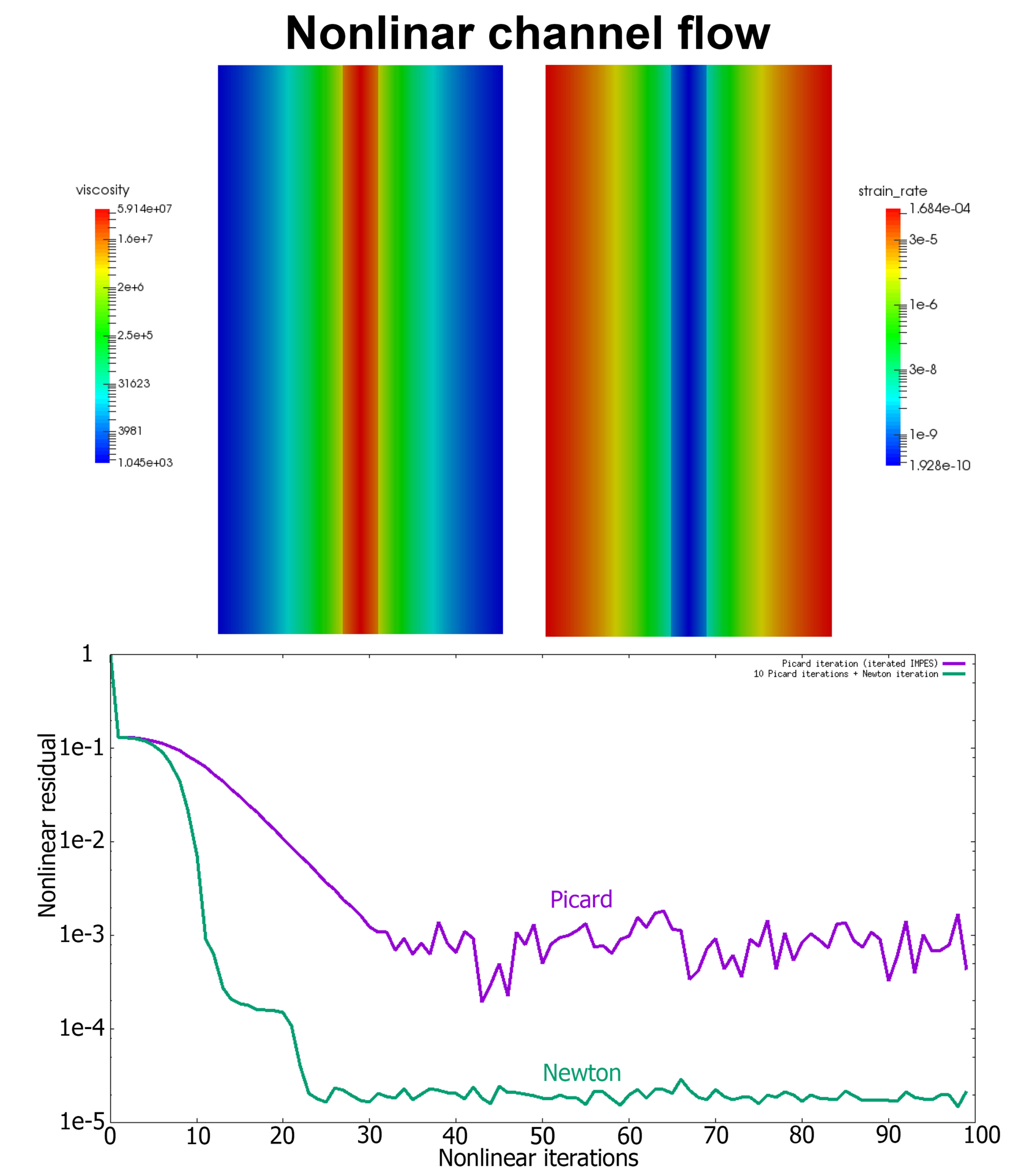
We can force tensor H to be SPD by scaling E^{sym} with a factor α:

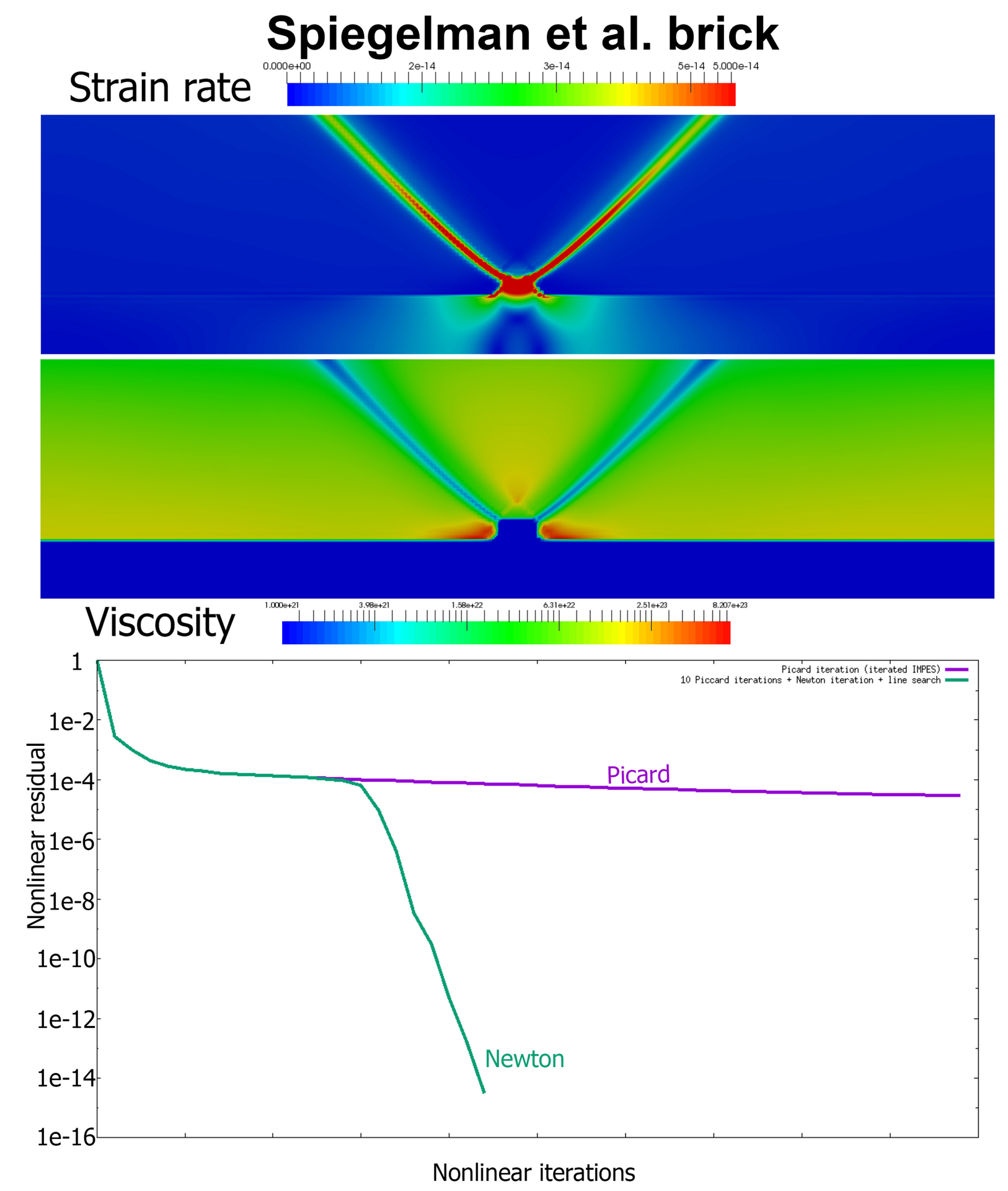
$$H^{\text{spd}} = 2\eta(\varepsilon(u)) I \otimes I + \alpha E^{\text{sym}}(\varepsilon(u))$$

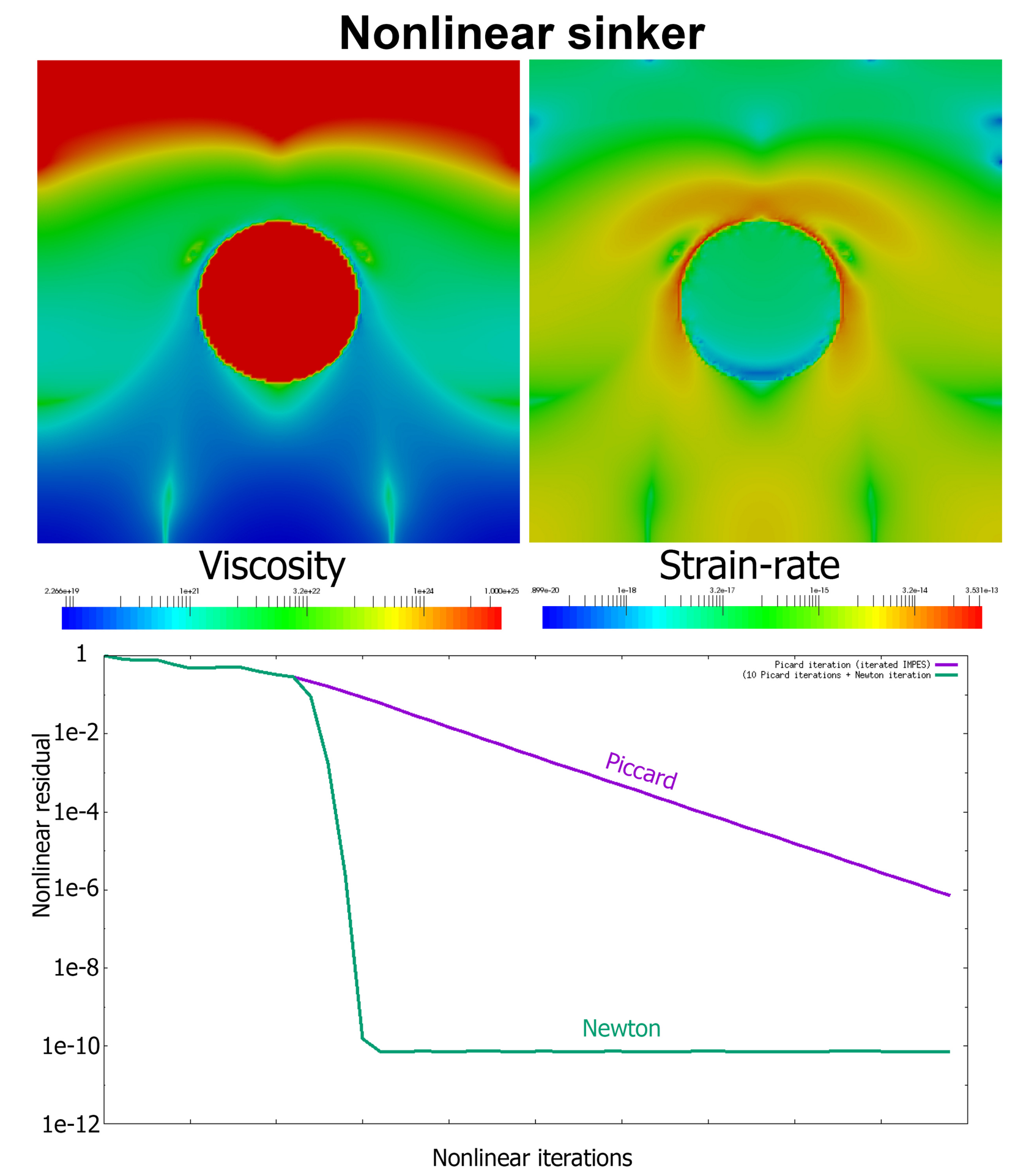
with $0 < \alpha \leq 1$. If $\alpha = 0$ the Jacobian will always be SPD, but to have optimal convergence we want α to be as large as possible (max one). It can be proven (Fraters et al., in prep) that the optimal value for α is (where $a = \varepsilon(\boldsymbol{u})$and $b = \frac{\partial \eta(\varepsilon(\boldsymbol{u}),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

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

Benchmarking the Newton Method



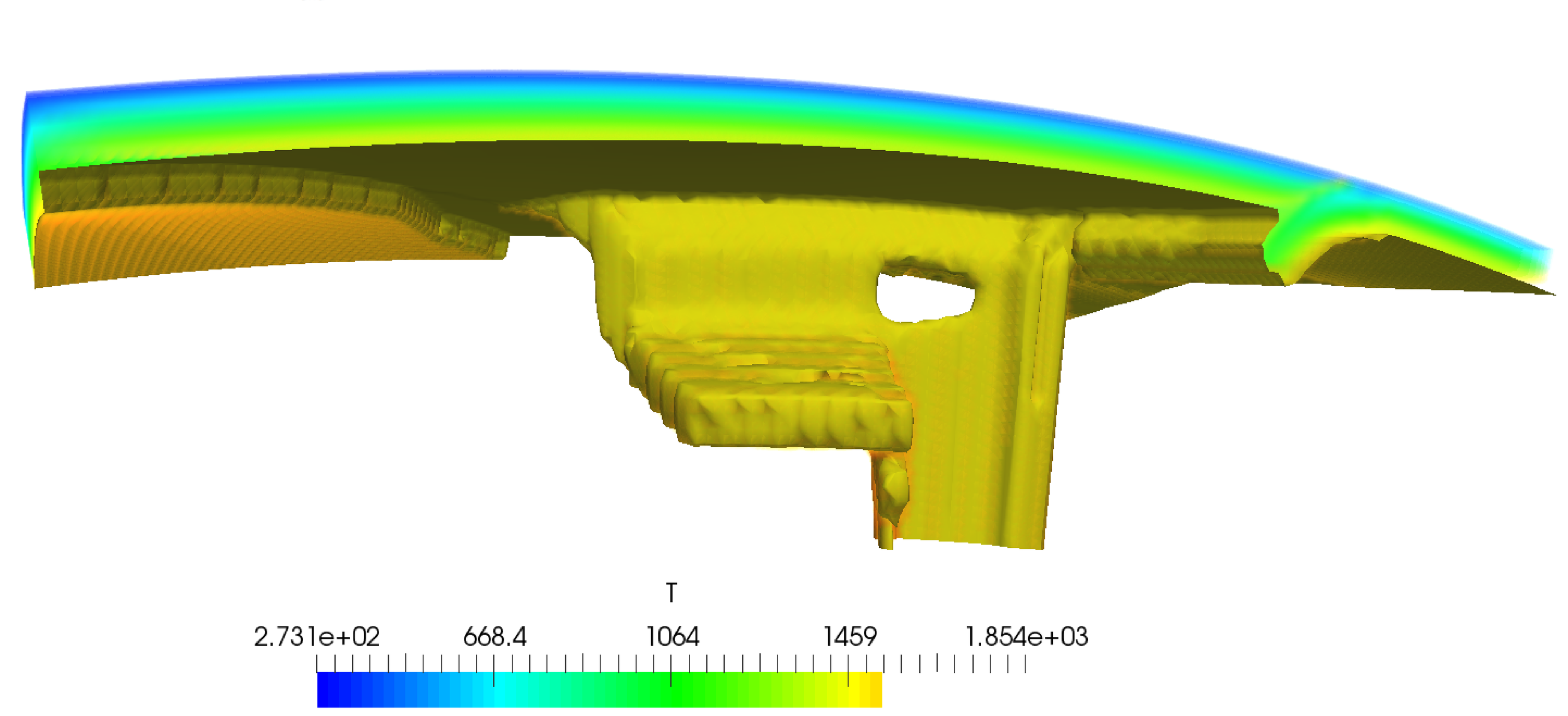


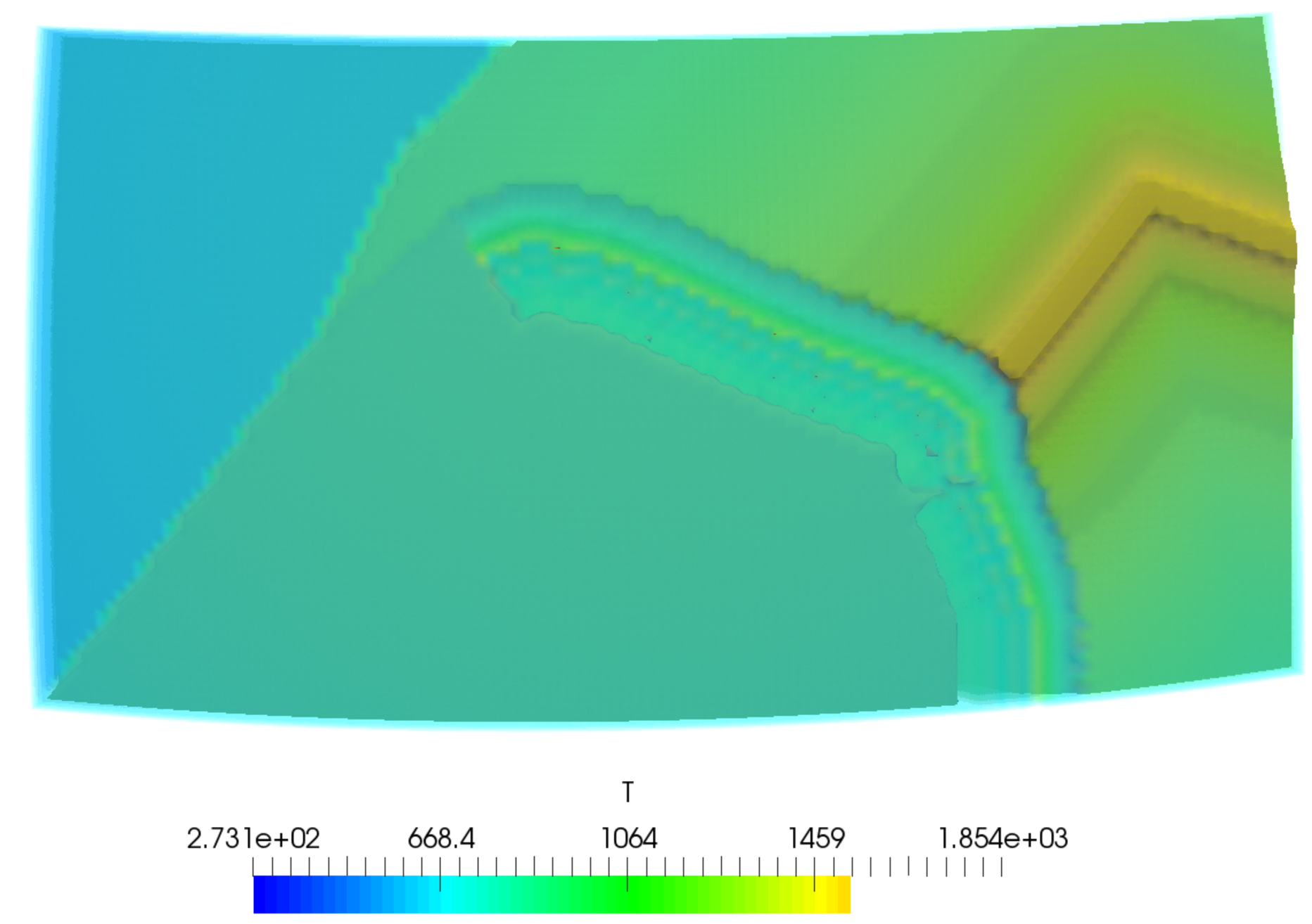


Demonstrating the World Generator

The next step is to apply the Newton solver to complex 3D geodynamic settings, such as constructed by our World Generator (Fraters et al., in prep), which generates initital conditions with little input:

```
subsection Initial conditions
  set Model name = world generator
  subsection World generator
    set Minimum parts per degree = 5
    set Minimum distance points = 1e-5
    set Refinement threshold radius = 50

    set Surface objects = \ # Name obj          | Module name       | Interpolation    | Coordinates of the cornerpoints of the polygon
                            North America       | Continental Plate |                  | -83:13,-82:14,-76.5:21,-75:24,-73:26,-72:27,-83:27; \
                            South America       | Continental Plate |                  | -62.5:13,-62.5:15.5,-57:16,-57:13; \
                            North Atlantic      | Oceanic Plate     |                  | -76.5:21,-75:24,-73:26,-72:27,-57:27,-57:13,-62.5:13,-62.5:15.5,-63.5:18.5,-64:20,-71:23,-72.5:23.5,-74:23; \
                            Caribbean Ocean     | Oceanic Plate     |                  | -83:0,-82:14,-76.5:21,-74:23,-72.5:23.5,-71:23,-64:20,-63.5:18.5,-62.5:15.5,-62.5:14,-62.5:0; \
                            South Trench        | Subducting plate  | Monotone Spline  | -63.5:18.5,-62.5:15.5,-62.5:13; \
                            North Trench        | Subducting plate  | Monotone Spline  | -74:23,-73.5:23.25,-72.5:23.5,-71:23.25,-67.75:21.875,-66.9375:21.53125,-64.5:20.5,-64:20,-63.74:19.75,-63.5:18.5

    subsection Continental plate
      set Parameters = \ # Name object | Temp. mod. | Comp. mod.
                         North America | Linear     | Layers; \
                         South America | Linear     | Constant

      subsection Linear
          set Temperature parameters = \ # Name object | depth | Bottom temperature
                                          North America | 150e3 | ;\
                                          South America | 100e3 |
      end
    end

    subsection Oceanic plate
      set Parameters = \ # Name object   | Temp. mod.  | Comp. mod. |  Coordinates of ride points
                         North Atlantic  | Plate model | None       | -64:20,-61:23,-60:22,-57:25; \
                         Caribbean Ocean | Plate model | None       | -90:0,-91:1

      subsection Plate model
        set Temperature parameters = \ # Name Object    | depth | B. temp. | spreading vel.
                                        North Atlantic  | 95e3  |          | 1.3e-2; \
                                        Caribbean Ocean | 95e3  |          | 1.3e-3
      end
    end

    subsection Subducting plate
      set Parameters = \ # Name object | Temp. mod.  | Comp. mod. | intpl. | Start depth | max depth seg. | length segments                     | thickness segments                                       |  angle segments                | ref. point
                         North Trench  | Plate model | None       |        | 0           | 1000e3         | 150e3,200e3,50e3,50e3,100e3,400e3   | 100e3,100e3,100e3:100e3,100e3:100e3,100e3,100e3          | 0:30,30:60,60:90,90,90,90      | -65:80; \
                         South Trench  | Plate model | None       |        | 0           | 1000e3         | 150e3,180e3                         | 100e3                                                    | 0:30,30:60                     | -50:15

      set Plate detail parameters = \ # Name object | between coords | length segments                   | thickness segments                          | angle segments                    | ref. point
                                        North Trench  | 1:3            | 150e3,200e3,50e3,50e3,100e3,400e3 | 100e3,100e3,100e3,100e3,100e3,100e3         | 0:30,30:60,60:90,90,90:20,20:10   | -65:80;
                                        North Trench  | 5:6            | 150e3,200e3,50e3,50e3,100e3,400e3 | 100e3,100e3,100e3:0,0,0:100e3,100e3         | 0:30,30:60,60:90,90,90,90         | -65:80; \

      subsection Plate model
        set Temperature parameters = \ # Name object   | B. temp | density | plate vel.
                                        North Trench   |         | 3300    | 1.3e-2; \
                                        South Trench   |         | 3300    | 1.3e-2
      end
    end
  end
end
```





Available features: different plate types, faults and variations in curvature and thickness within slabs.