
SiStER

Overview of the **SiStER** code

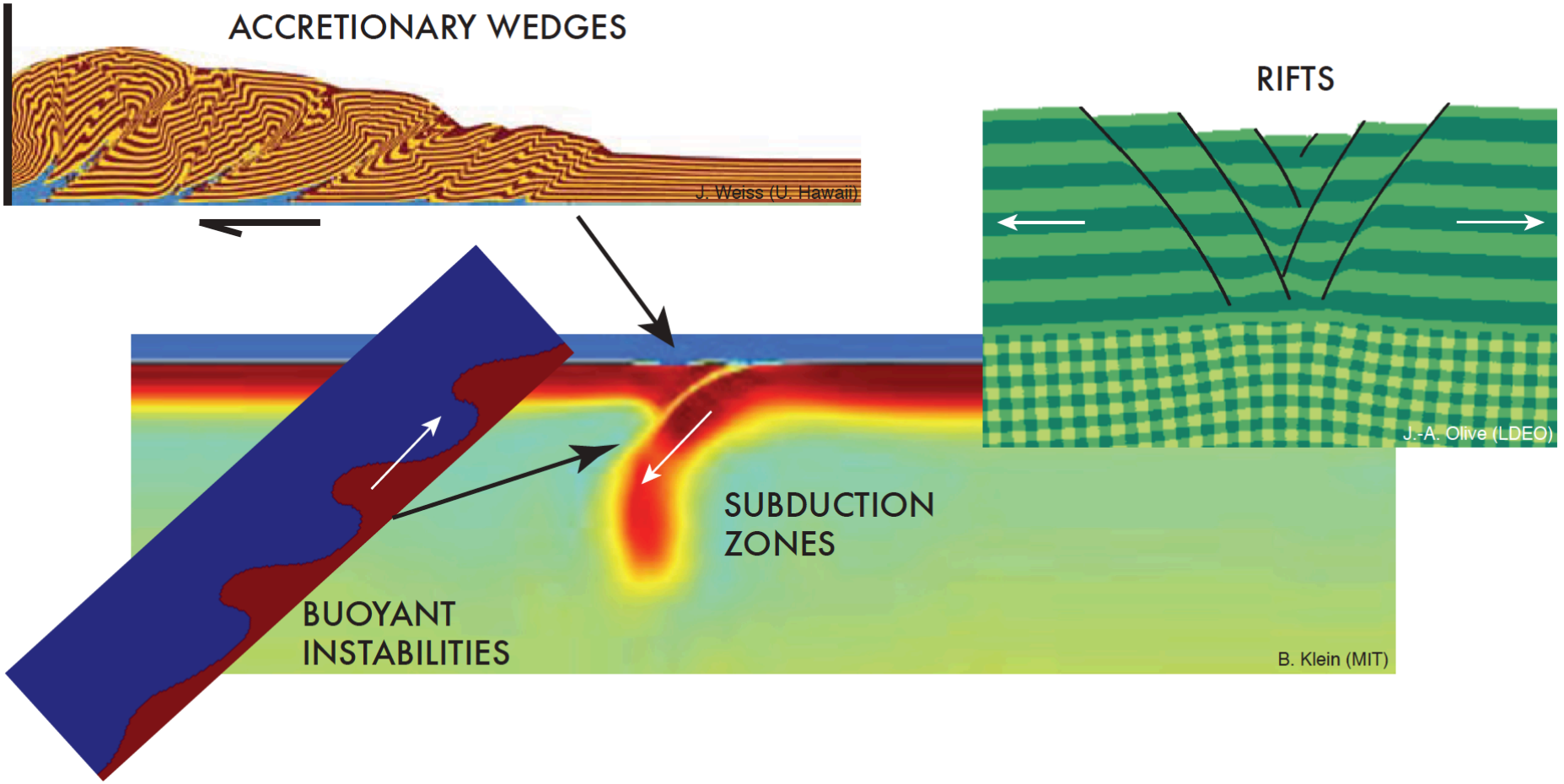
Simple Stokes solver with Exotic Rheologies

Jean-Arthur Olive (LDEO / Columbia University)

In collaboration with E. Mittelstaedt (U. Idaho), B.Z. Klein (MIT),
S. Howell (U. Hawaii), M.D. Behn (WHOI), and G. Ito (U. Hawaii)

Objectives & Challenges

Modeling lithosphere and mantle deformation with continuum mechanics: *Stokes flow with large strains, strain localization, non-linear rheologies, sharp contrasts in material properties, complex BCs.*



Governing equations

- Conservation of mass and momentum

$$\partial_k v_k = 0$$

$$\partial_j \sigma'_{ij} - \partial_i P + \rho g = 0$$

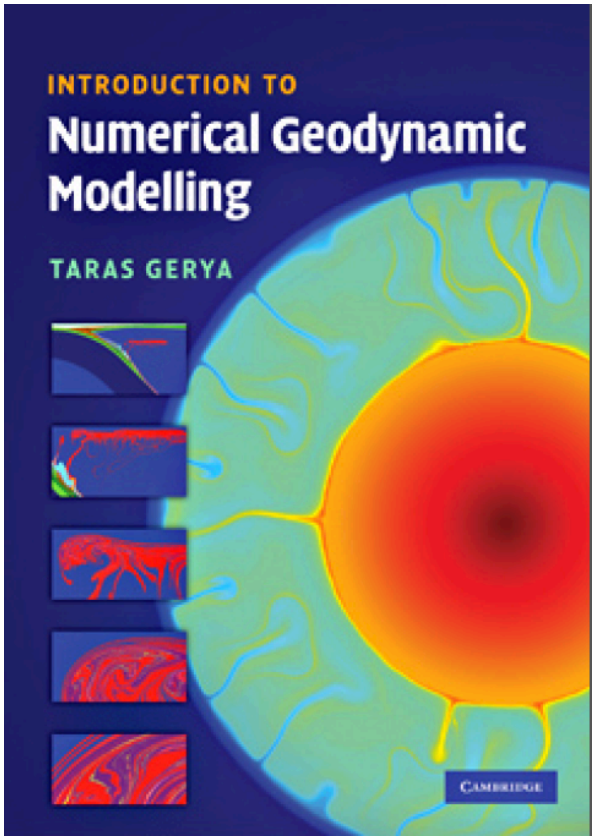
- Non linear viscosity law + elasticity + plasticity

- Conservation of energy

$$\frac{\partial T}{\partial t} + v\nabla T = \nabla \cdot \left(\kappa \nabla T \right)$$

Numerical methodology / Citations

- Finite Difference / Particle-in-Cell method. Implementation largely based on *Gerya* [2010].



- Methodology partly described in *Olive et al.* [2016, GJI]

Geophysical Journal International

Geophys. J. Int. (2016) **205,** 728–743
Advance Access publication 2016 January 27
GJI Geodynamics and tectonics

doi: 10.1093/gji/ggw044

The role of elasticity in simulating long-term tectonic extension

Jean-Arthur Olive,[1,*] Mark D. Behn,[2] Eric Mittelstaedt,[3] Garrett Ito[4] and Benjamin Z. Klein[5]

Finite-difference discretization

Steady-state Stokes flow on Eulerian grid:

Conservative FD scheme on a staggered grid:

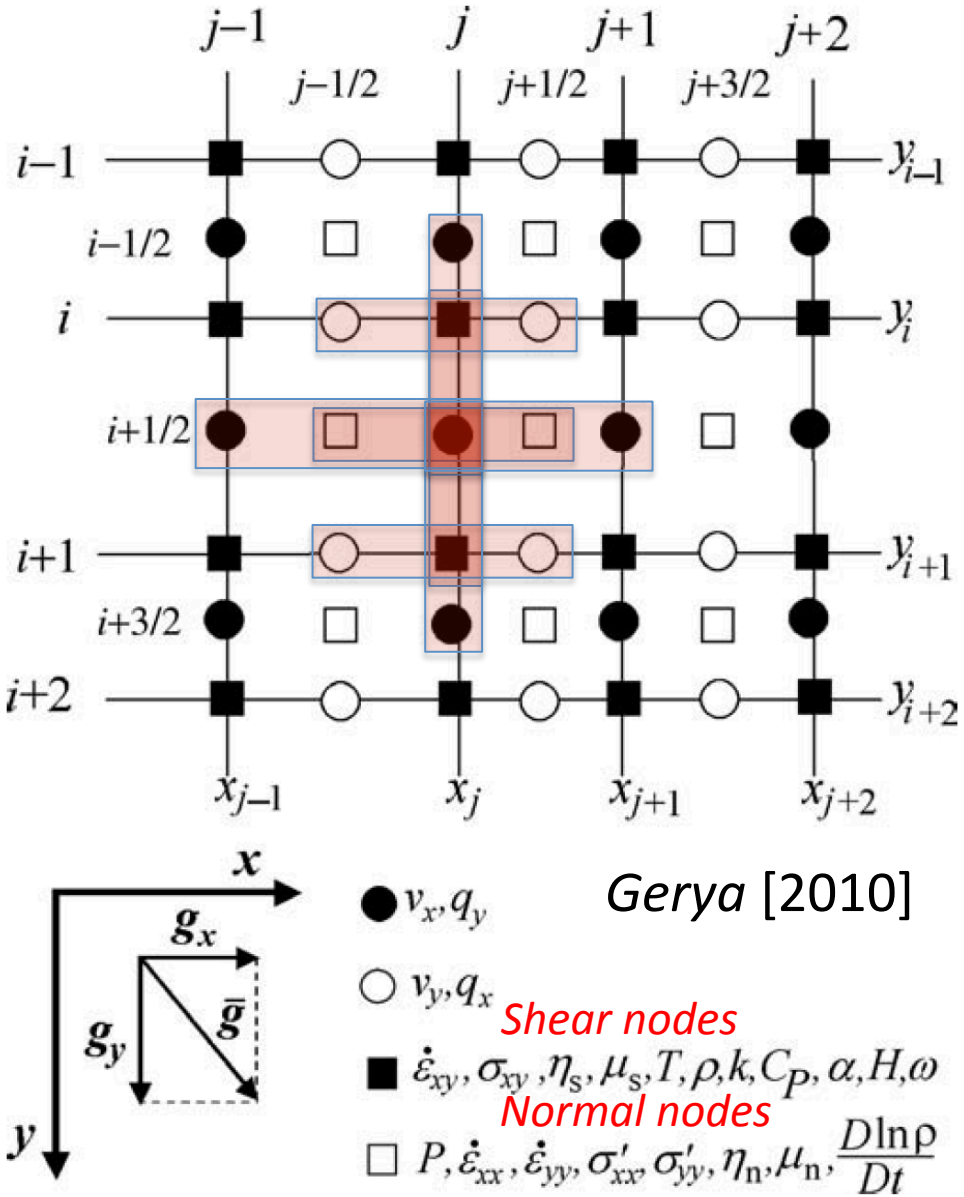
$$\nabla \cdot \left(\eta \left(\nabla v + \nabla v^T \right) \right) - \nabla P + \rho g = RHS$$

$$\nabla \cdot v = 0$$

Leads to linear system:

$$JX = \begin{pmatrix} K & G \\ G^T & 0 \end{pmatrix} \begin{pmatrix} v \\ P \end{pmatrix} = \begin{pmatrix} rhs \\ 0 \end{pmatrix}$$

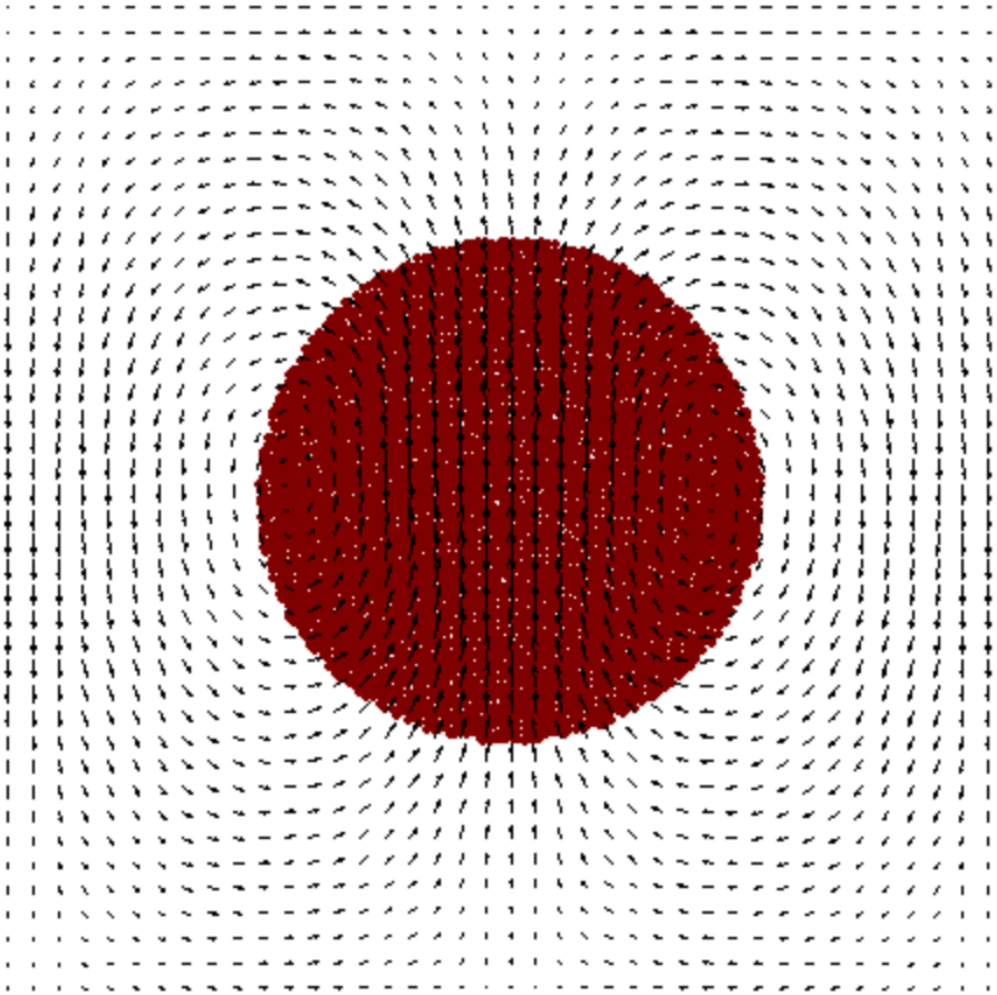
Solved with MATLAB's "backslash" solver.



Gerya [2010]

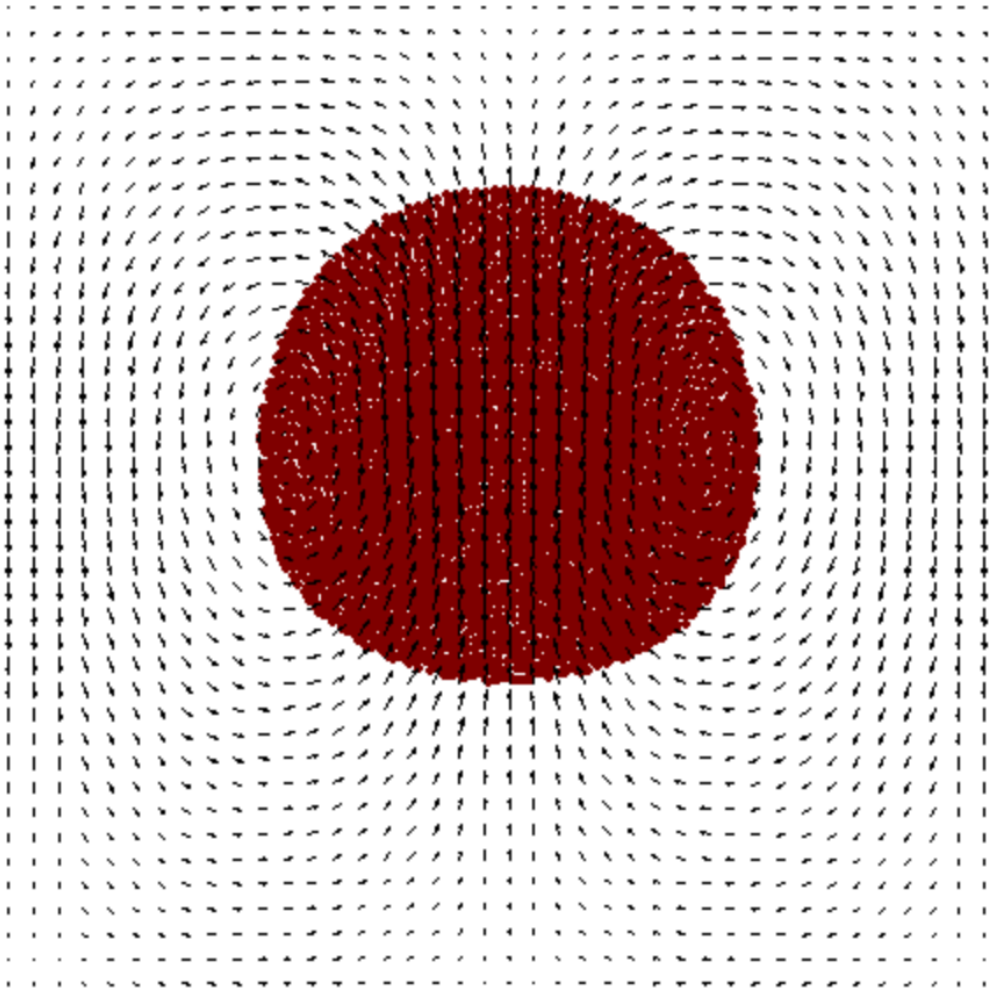
Particles handle material advection

- low density sphere in dense, low viscosity fluid:

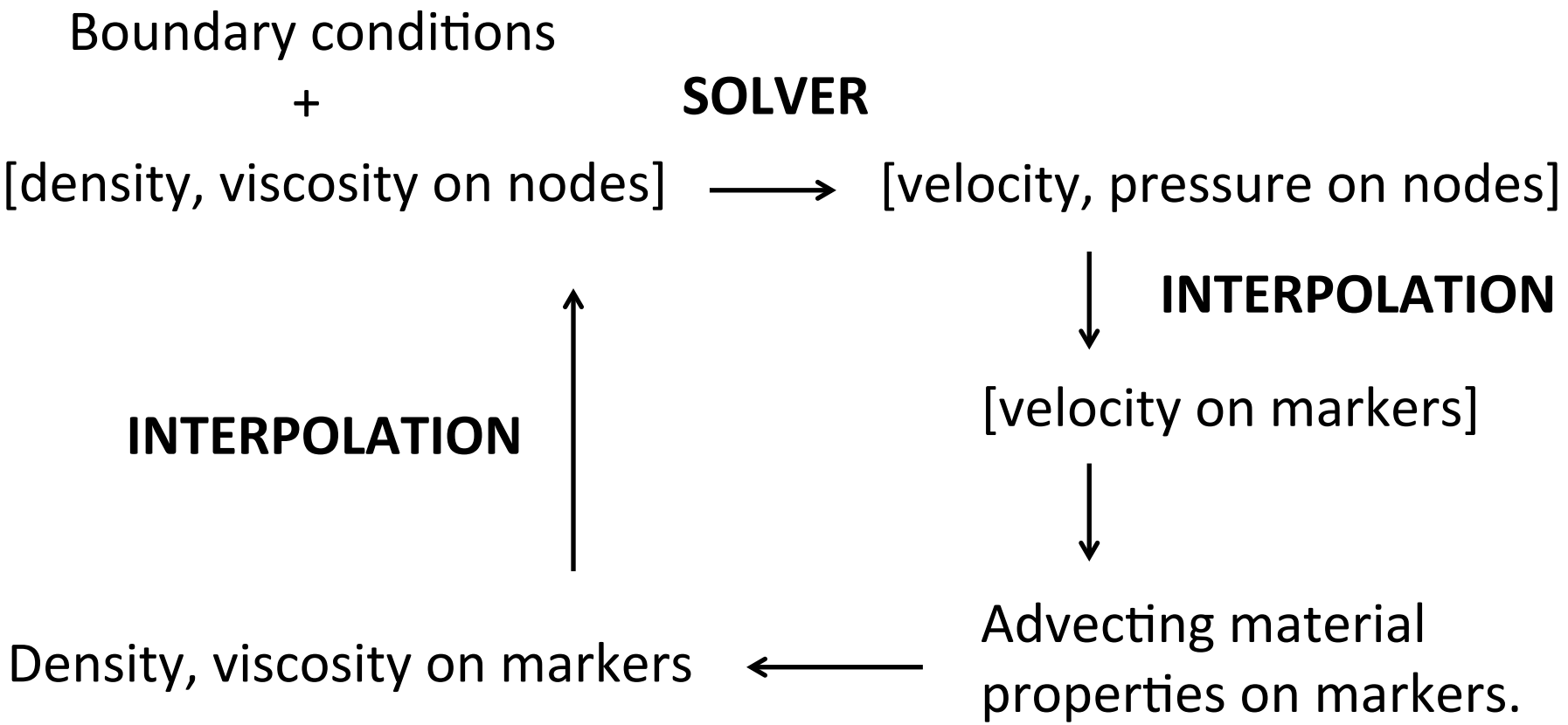


Particles handle material advection

- low density sphere in dense, low viscosity fluid:



Basic code structure



Handling non-linear creep laws

The formulation of viscosity strictly pertaining to non-linear creep laws writes:

$$\eta = \left(\frac{1}{\eta_{DIFF}} + \frac{1}{\eta_{DISC}} + \text{ other terms } \right)^{-1}$$

This distinguishes the contributions from diffusion and dislocation creep.

Each creep law can be parameterized as: $\eta = A^{-\frac{1}{n}} \dot{\epsilon}_{II}^{\frac{1-n}{n}} e^{\frac{E}{nRT}}$

Picard iterations (**SiStER_run_Picard_iterations**) consist of:
1/ Solving the flow equations using the latest viscosity map
2/ Calculating the strain rate from the latest solution
3/ Re-evaluating the viscosity from the new strain rate map
Repeat until the solution stops changing (within specified tolerance)

Implementation of visco-elasticity

Maxwell **visco-elastic** rheology:

$$\dot{\varepsilon}_{ij} = \frac{1}{2G}\frac{D\sigma_{ij}}{Dt} + \frac{1}{2\eta}\sigma_{ij}$$

ELASTIC VISCOUS

$$\tau_{Max} = \frac{\eta}{G}$$

Maxwell time scale:

FD approximation

$$\sigma_{ij}^{(t)} = 2\,Z\,\eta\,\dot{\varepsilon}_{ij} + (1-Z)\,\sigma_{ij}^{(t-\Delta t)}$$

with $Z = \frac{G\Delta t}{\eta + G\Delta t}$

Implementation in a viscous solver:
[*Moresi et al.*, 2003]

$$\nabla \cdot (2\,Z\,\eta\,\dot{\varepsilon}_{ij}) - \nabla p + \rho g = -\nabla \cdot ((1-Z)\sigma_{ij}^{(t-\Delta t)})$$

For VISCOUS BEHAVIOR
set very high G so that:
$G\Delta t \gg \eta$, i.e., **$\Delta t \gg \tau_{Max}$**
(and $Z \approx 1$)

For ELASTIC BEHAVIOR
impose high viscosity.

Implementation of plasticity

Plastic (localizing) rheology

Mohr-Coulomb plasticity is implemented by weakening viscosity to cap σ'_{II} at σ_{yield}:

$$\eta = \left(\frac{1}{\eta_{REF}} + \frac{1}{\eta_{PLAS}} \right)^{-1} \quad \text{with} \quad \eta_{PLAS} = \frac{\sigma_{yield}}{2\dot{\varepsilon}_{II}}$$

Yield stress prescribed through cohesion and friction

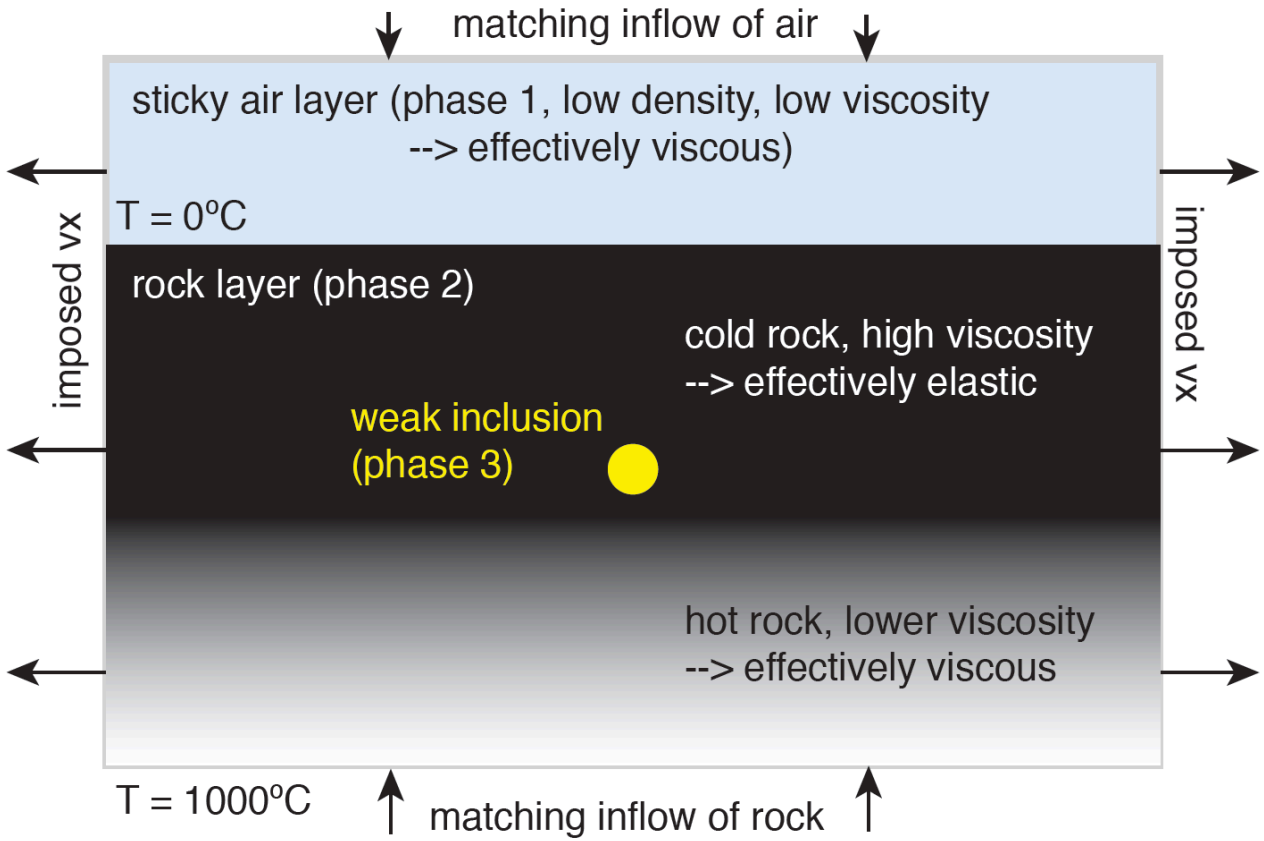
$$\boldsymbol{\sigma_{yield}} = \sin\Phi \ \mathbf{P} + C \cos\Phi$$

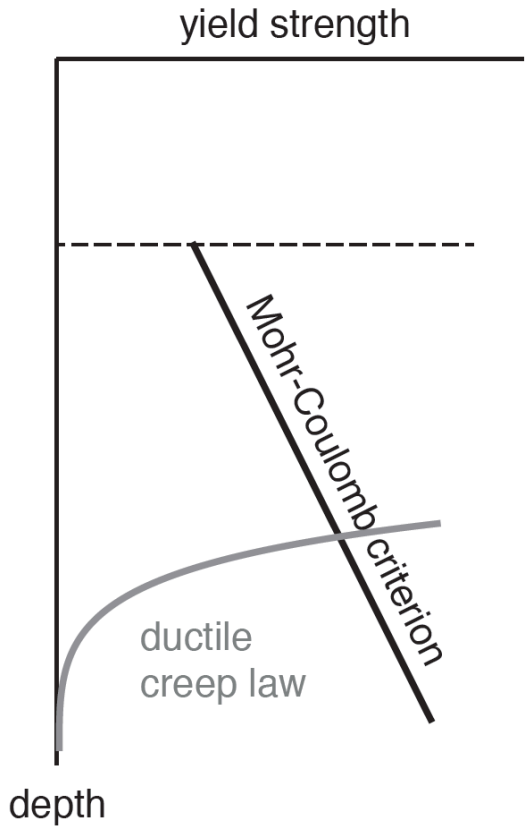
Strain localization is promoted by **dropping cohesion** linearly with accumulated plastic strain ε_P. Full weakening occurs when fault offset reaches $h_C \approx 3\ \Delta x\ \varepsilon_{P\ CRIT}$ *Lavier et al.* [2000]

Accumulated plastic strain **heals** on a prescribed time scale

Example case: rifting example

Run **SiStER_MAIN**, then enter **SiStER_Input_File_continental_rift**





Input file structure

Runs for 100 iterations, will output a file every 10 time steps

```
% DURATION OF SIMULATION AND FREQUENCY OF OUTPUT %%%%%%%%%%%%%%%%%%%%%%%%%%
Nt=100; % max number of time iterations
dt_out=10; % output files every "dt_out" iterations
```

Top-left corner of the domain at (0,0), x>0 to the right, y>0 down

```
% DOMAIN SIZE AND GRIDDING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xsize=90e3;
ysize=30e3;
% gridding- from 0 to GRID.x(1), grid size is GRID.dx(1)
% from GRID.x(1) to GRID.x(2), grid size is GRID.dx(1) etc...
% same for y
GRID.dx(1)=2000;
GRID.x(1)=30e3;
GRID.dx(2)=400;
GRID.x(2)=60e3;
GRID.dx(3)=2000;
GRID.dy(1)=2000;
GRID.y(1)=9e3;
GRID.dy(2)=400;
GRID.y(2)=22e3;
GRID.dy(3)=2000;
```

Box size 90 km × 30 km

All units SI

Horizontal grid size = 2km between 0 and 30 km, 400m between 30 and 60 km, 2 km between 60 and 90 km.

Vertical grid size = 2km between 0 and 9 km, 400m between 9 and 22 km, 2 km between 22 and 30 km.

Input file structure

Seeding markers to advect material properties:

```
% LAGRANGIAN MARKERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Mquad=6; % number of markers in the smallest quadrant
Mquad_crit=3; % minimum number of markers allowed in smallest quadrant (for reseeding)
```

The initial marker density is defined such that a quarter of the smallest cell (defined above) has 6 markers, i.e., the smallest cell contains 24 markers. Bigger cells will contain more markers such that the density of markers (number of markers per m^2) is constant.

As markers move around, gaps may form in the marker distribution. New markers will be added to fill in the gaps in cells that have quadrants with less than 3 markers.

Input file structure

Setting up the initial geometry of the phases (material types)

Here there are 3 phases in total: a sticky air layer, a rock layer and a weak rock inclusion

```
30
31      % GEOMETRY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32
33 -    Nphase=3; % number of phases
34
35      % phase 1
36 -    GEOM(1).type=1; % 1 = layer (then specify top and bot) or 2 = circle % 1 = layer (then specify top and bot) or 2 = circle
37 -    GEOM(1).top=0;
38 -    GEOM(1).bot=10e3;
39
40      % phase 2
41 -    GEOM(2).type=1; % 1 = layer (then specify top and bot) or 2 = circle % 1 = layer (then specify top and bot) or 2 = circle
42 -    GEOM(2).top=10e3;
43 -    GEOM(2).bot=30e3;
44
45      % phase 3
46 -    GEOM(3).type=2; % 1 = layer (then specify top and bot) or 2 = circle % 1 = layer (then specify top and bot) or 2 = circle
47 -    GEOM(3).x0=xsize/2;
48 -    GEOM(3).y0=20e3;
49 -    GEOM(3).rad=1e3;
50
```

First phase (air) is a horizontal layer (type 1) spanning the whole width of the box, between y=0 (top) and y=10 km (bottom).

Input file structure

```
30
31      % GEOMETRY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32
33 -    Nphase=3; % number of phases
34
35      % phase 1
36 -    GEOM(1).type=1; % 1 = layer (then specify top and bot) or 2 = circle % 1 = layer (then specify top and bot) or 2 = circle
37 -    GEOM(1).top=0;
38 -    GEOM(1).bot=10e3;
39
40      % phase 2
41 -    GEOM(2).type=1; % 1 = layer (then specify top and bot) or 2 = circle % 1 = layer (then specify top and bot) or 2 = circle
42 -    GEOM(2).top=10e3;
43 -    GEOM(2).bot=30e3;
44
45      % phase 3
46 -    GEOM(3).type=2; % 1 = layer (then specify top and bot) or 2 = circle % 1 = layer (then specify top and bot) or 2 = circle
47 -    GEOM(3).x0=xsize/2;
48 -    GEOM(3).y0=20e3;
49 -    GEOM(3).rad=1e3;
50
```

Second phase (rock) is a horizontal layer (type 1) spanning the whole width of the box, between y=10 (top) and y=30 km (bottom).

Third phase (weak rock) is a circle (type 2) of radius 1 km centered at (45 km, 20 km).

Phase definitions overprint each other in the order they are defined- make sure they cover the entire domain!

Input file structure

Material properties must be specified for each phase defined above, in the following format (shown here for phase 1)

```
% MATERIAL PROPERTIES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% creep laws of the form: pre^(-1/n)*epsII^((1-n)/n)*exp(E/(nRT))
% harmonically averaging diffusion creep, dislocation creep
% (and plastic creep to simulate brittle failure)

% phase 1
MAT(1).phase=1;
% density parameters
MAT(1).rho0=0.01;
MAT(1).alpha=0;
% elasticity
MAT(1).G=1e18;
% diffusion creep parameters
MAT(1).pre_diff=.5/1e18;
MAT(1).Ediff=0;
MAT(1).ndiff=1;
% dislocation creep parameters
MAT(1).pre_disc=.5/1e18;
MAT(1).Edisc=0;
MAT(1).ndisc=1;
% plasticity
MAT(1).mu=0.6;
MAT(1).Cmax=40e6;
MAT(1).Cmin=0.01e6;
MAT(1).ecrit=0.1;
```

Phase 1 will be indexed in marker array im with value 1

Reference density and thermal expansion, see **SiStER_get_density** function

Parameters for diffusion creep law

Parameters for dislocation creep law

See **SiStER_get_viscosity** function

Parameters for plasticity (faulting), see **SiStER_get_yield_stress** function

Input file structure

Material properties must be specified for each phase defined above, in the following format (shown here for phase 1)

```
% MATERIAL PROPERTIES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% creep laws of the form: pre^(-1/n)*epsII^((1-n)/n)*exp(E/(nRT))
% harmonically averaging diffusion creep, dislocation creep
% (and plastic creep to simulate brittle failure)

% phase 1
MAT(1).phase=1;
% density parameters
MAT(1).rho0=0.01;
MAT(1).alpha=0;
% elasticity
MAT(1).G=1e18;
% diffusion creep parameters
MAT(1).pre_diff=.5/1e18;
MAT(1).Ediff=0;
MAT(1).ndiff=1;
% dislocation creep parameters
MAT(1).pre_disc=.5/1e18;
MAT(1).Edisc=0;
MAT(1).ndisc=1;
% plasticity
MAT(1).mu=0.6;
MAT(1).Cmax=40e6;
MAT(1).Cmin=0.01e6;
MAT(1).ecrit=0.1;
```

NOTES ON PHASE 1 (sticky air)

Note the low density + no thermal dependence of density (MAT(1).rho0=0.01;)

Unrealistically high shear modulus ensures an effectively viscous behavior (MAT(1).G=1e18;)

Input file structure

Material properties must be specified for each phase defined above, in the following format (shown here for phase 1)

```
% MATERIAL PROPERTIES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% creep laws of the form: pre^(-1/n)*epsII^((1-n)/n)*exp(E/(nRT))
% harmonically averaging diffusion creep, dislocation creep
% (and plastic creep to simulate brittle failure)

% phase 1
MAT(1).phase=1;
% density parameters
MAT(1).rho0=0.01;
MAT(1).alpha=0;
% elasticity
MAT(1).G=1e18;
% diffusion creep parameters
MAT(1).pre_diff=.5/1e18;
MAT(1).Ediff=0;
MAT(1).ndiff=1;
% dislocation creep parameters
MAT(1).pre_disc=.5/1e18;
MAT(1).Edisc=0;
MAT(1).ndisc=1;
% plasticity
MAT(1).mu=0.6;
MAT(1).Cmax=40e6;
MAT(1).Cmin=0.01e6;
MAT(1).ecrit=0.1;
```

NOTES ON PHASE 1 (sticky air)

To enforce a constant Newtonian viscosity η_0 that does not depend on temperature, set $n = 1$, $E=0$, and prefactor $= 0.5/\eta_0$ in both dislocation and diffusion creep laws

Input file structure

Material properties must be specified for each phase defined above, in the following format (shown here for phase 2)

```
79
80      % phase 2
81 -    MAT(2).phase=2;
82      % density parameters
83 -    MAT(2).rho0=2700;
84 -    MAT(2).alpha=0;
85      % elasticity
86 -    MAT(2).G=30e9;
87      % diffusion creep parameters
88 -    MAT(2).pre_diff=.5/1e40;
89 -    MAT(2).Ediff=0;
90 -    MAT(2).ndiff=1;
91      % dislocation creep parameters
92 -    MAT(2).pre_disc=1.0e-3;
93 -    MAT(2).Edisc=167000*2.25;
94 -    MAT(2).ndisc=2;
95      % plasticity
96 -    MAT(2).mu=0.6;
97 -    MAT(2).Cmax=40e6;
98 -    MAT(2).Cmin=0.01e6;
99 -    MAT(2).ecrit=0.1;
100
```

NOTES ON PHASE 2 (rock)

Very low prefactor in diffusion creep to select only dislocation creep

Non-Newtonian creep law with T-dependence

Plasticity parameters follow the formalism of *Lavier et al.* [2000, JGR]

Input file structure

Material properties must be specified for each phase defined above, in the following format (shown here for phase 3)

```
101
102     % phase 3
103 -   MAT(3).phase=3;
104     % density parameters
105 -   MAT(3).rho0=2700;
106 -   MAT(3).alpha=0;
107     % elasticity
108 -   MAT(3).G=30e9;
109     % diffusion creep parameters
110 -   MAT(3).pre_diff=.5/1e40;
111 -   MAT(3).Ediff=0;
112 -   MAT(3).ndiff=1;
113     % dislocation creep parameters
114 -   MAT(3).pre_disc=1.0e-3;
115 -   MAT(3).Edisc=167000*2.25;
116 -   MAT(3).ndisc=2;
117     % plasticity
118 -   MAT(3).mu=0.6;
119 -   MAT(3).Cmax=0.01e6;
120 -   MAT(3).Cmin=0.01e6;
121 -   MAT(3).ecrit=0.1;
122
```

NOTES ON PHASE 3 (weak rock)

Identical to phase 2,
but with very low initial cohesion
to promote rapid yielding of this weak seed

Input file structure

More parameters…

```
% ADDITIONAL PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PARAMS.YNElast=1; % elasticity on (1) or off (0)
PARAMS.YNPlas=1; % plasticity on (1) or off (0)
PARAMS.epsII_from_stress=1; % get strain rate from stresses (1, default) or from velocity field (0)
PARAMS.tau_heal=1e12; % healing time for plasticity (s)
PARAMS.gx=0; % gravity along x
PARAMS.gy=9.8; % gravity along y
PARAMS.fracCFL=0.5; % distance by which a marker is allowed to move over a time step, expressed as a fraction
PARAMS.R=8.314; % gas constant
PARAMS.etamax=1e25; % maximum viscosity
PARAMS.etamin=1e18; % minimum viscosity
PARAMS.Tsolve=1; % yes (1) or no (0) solve for temperature
```

Simulation incorporate elasticity and plasticity (brittle strain localization)

Input file structure

More parameters…

```
% ADDITIONAL PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PARAMS.YNElast=1; % elasticity on (1) or off (0)
PARAMS.YNPlas=1; % plasticity on (1) or off (0)
PARAMS.epsII_from_stress=1; % get strain rate from stresses (1, default) or from velocity field (0)
PARAMS.tau_heal=1e12; % healing time for plasticity (s)
PARAMS.gx=0; % gravity along x
PARAMS.gy=9.8; % gravity along y
PARAMS.fracCFL=0.5; % distance by which a marker is allowed to move over a time step, expressed as a fraction
PARAMS.R=8.314; % gas constant
PARAMS.etamax=1e25; % maximum viscosity
PARAMS.etamin=1e18; % minimum viscosity
PARAMS.Tsolve=1; % yes (1) or no (0) solve for temperature
```

Strain rate is calculated using the current stress and the stress-strain relation, which limits the number of interpolation. The alternative (set value to 0) is to calculate it directly from the velocity solution.

Input file structure

More parameters…

```
% ADDITIONAL PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PARAMS.YNElast=1; % elasticity on (1) or off (0)
PARAMS.YNPlas=1; % plasticity on (1) or off (0)
PARAMS.epsII_from_stress=1; % get strain rate from stresses (1, default) or from velocity field (0)
PARAMS.tau_heal=1e12; % healing time for plasticity (s)
PARAMS.gx=0; % gravity along x
PARAMS.gy=9.8; % gravity along y
PARAMS.fracCFL=0.5; % distance by which a marker is allowed to move over a time step, expressed as a fraction
PARAMS.R=8.314; % gas constant
PARAMS.etamax=1e25; % maximum viscosity
PARAMS.etamin=1e18; % minimum viscosity
PARAMS.Tsolve=1; % yes (1) or no (0) solve for temperature
```

Accumulated plastic strain (in weak yielded zones, i.e., faults) heals exponentially on a time scale specified here, in seconds [*Lavier et al.*, 2000, JGR]. This way, old, abandoned faults where no strain localization is sustained can disappear.

Input file structure

More parameters…

```
% ADDITIONAL PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PARAMS.YNElast=1; % elasticity on (1) or off (0)
PARAMS.YNPlas=1; % plasticity on (1) or off (0)
PARAMS.epsII_from_stress=1; % get strain rate from stresses (1, default) or from velocity field (0)
PARAMS.tau_heal=1e12; % healing time for plasticity (s)
PARAMS.gx=0; % gravity along x
PARAMS.gy=9.8; % gravity along y
PARAMS.fracCFL=0.5; % distance by which a marker is allowed to move over a time step, expressed as a fraction
PARAMS.R=8.314; % gas constant
PARAMS.etamax=1e25; % maximum viscosity
PARAMS.etamin=1e18; % minimum viscosity
PARAMS.Tsolve=1; % yes (1) or no (0) solve for temperature
```

Gravity in m/s^2

(both x and y components, i.e., cases with sloping gravity are possible)

Input file structure

More parameters…

```
% ADDITIONAL PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PARAMS.YNElast=1; % elasticity on (1) or off (0)
PARAMS.YNPlas=1; % plasticity on (1) or off (0)
PARAMS.epsII_from_stress=1; % get strain rate from stresses (1, default) or from velocity field (0)
PARAMS.tau_heal=1e12; % healing time for plasticity (s)
PARAMS.gx=0; % gravity along x
PARAMS.gy=9.8; % gravity along y
PARAMS.fracCFL=0.5; % distance by which a marker is allowed to move over a time step, expressed as a fraction
PARAMS.R=8.314; % gas constant
PARAMS.etamax=1e25; % maximum viscosity
PARAMS.etamin=1e18; % minimum viscosity
PARAMS.Tsolve=1; % yes (1) or no (0) solve for temperature
```

The time step (used for marker advection) is set such that markers cannot move by more than PARAMS.fracCFL × the smallest grid size, at every time iteration (= CFL condition for advection).

Input file structure

More parameters…

```

% ADDITIONAL PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PARAMS.YNElast=1; % elasticity on (1) or off (0)
PARAMS.YNPlas=1; % plasticity on (1) or off (0)
PARAMS.epsII_from_stress=1; % get strain rate from stresses (1, default) or from velocity field (0)
PARAMS.tau_heal=1e12; % healing time for plasticity (s)
PARAMS.gx=0; % gravity along x
PARAMS.gy=9.8; % gravity along y
PARAMS.fracCFL=0.5; % distance by which a marker is allowed to move over a time step, expressed as a fraction
PARAMS.R=8.314; % gas constant
PARAMS.etamax=1e25; % maximum viscosity
PARAMS.etamin=1e18; % minimum viscosity
PARAMS.Tsolve=1; % yes (1) or no (0) solve for temperature
```

Hard limits on the viscosity, enforced in **SiStER_get_viscosity**
Here it cannot drop below 1e18 Pa.s, or exceed 1e25 Pa.s.

Input file structure

More parameters…

```
% ADDITIONAL PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PARAMS.YNElast=1; % elasticity on (1) or off (0)
PARAMS.YNPlas=1; % plasticity on (1) or off (0)
PARAMS.epsII_from_stress=1; % get strain rate from stresses (1, default) or from velocity field (0)
PARAMS.tau_heal=1e12; % healing time for plasticity (s)
PARAMS.gx=0; % gravity along x
PARAMS.gy=9.8; % gravity along y
PARAMS.fracCFL=0.5; % distance by which a marker is allowed to move over a time step, expressed as a fraction
PARAMS.R=8.314; % gas constant
PARAMS.etamax=1e25; % maximum viscosity
PARAMS.etamin=1e18; % minimum viscosity
PARAMS.Tsolve=1; % yes (1) or no (0) solve for temperature
```

If set to 1, the temperature field will evolve by advection (in the solid flow field) and diffusion.

If set to 0, it will only be advected- the heat equation will not be solved.

Input file structure

Thermal parameters

```
136      % initial temperature profile, polynomial with depth
137      % T = a0 + a1*y+a2*y^2+a3*y^3+amp*sin(2*pi*X/lam)
138      % (make sure it matches the BCs)
139 -    PARAMS.a0=0;
140 -    PARAMS.a1=0;
141 -    PARAMS.a2=0;
142 -    PARAMS.a3=1000/(30e3)^3;
143 -    PARAMS.amp=0; % amplitude of sinusoidal perturbation
144 -    PARAMS.lam=1; % wavelength of sinusoidal perturbation
145 -    PARAMS.ynTreset=1; % if ==1, reset T=T0 where im==1 (sticky layer)
146 -    PARAMS.T0=0;
147      % reference values for the constant diffusivity thermal solver
148      % (kappa = kref / (rhoref*cpref))
149 -    PARAMS.rhoref=MAT(2).rho0;
150 -    PARAMS.kref=3;
151 -    PARAMS.cpref=1000;
```

The initial temperature structure follows this functional form:

T = a0 + a1 y + a2 y^2 + a3 y^3 + amp*sin(2*pi x/lam)

See **SiStER_Initialize.m** – Make sure it is not incompatible with the imposed boundary conditions! And do not forget that y=0 is the top of the domain, not of the rock layer (it includes the sticky layer)

Input file structure

Thermal parameters

```
136      % initial temperature profile, polynomial with depth
137      % T = a0 + a1*y+a2*y^2+a3*y^3+amp*sin(2*pi*X/lam)
138      % (make sure it matches the BCs)
139 -    PARAMS.a0=0;
140 -    PARAMS.a1=0;
141 -    PARAMS.a2=0;
142 -    PARAMS.a3=1000/(30e3)^3;
143 -    PARAMS.amp=0; % amplitude of sinusoidal perturbation
144 -    PARAMS.lam=1; % wavelength of sinusoidal perturbation
145 -    PARAMS.ynTreset=1; % if ==1, reset T=T0 where im==1 (sticky layer)
146 -    PARAMS.T0=0;
147      % reference values for the constant diffusivity thermal solver
148      % (kappa = kref / (rhoref*cpref))
149 -    PARAMS.rhoref=MAT(2).rho0;
150 -    PARAMS.kref=3;
151 -    PARAMS.cpref=1000;
```

With PARAMS.ynTreset = 1, the temperature field will be automatically set to PARAMS.T0 throughout the sticky layer. This is a way to enforce T0 at the air-rock interface.

Input file structure

Thermal parameters

```
136      % initial temperature profile, polynomial with depth
137      % T = a0 + a1*y+a2*y^2+a3*y^3+amp*sin(2*pi*X/lam)
138      % (make sure it matches the BCs)
139 -    PARAMS.a0=0;
140 -    PARAMS.a1=0;
141 -    PARAMS.a2=0;
142 -    PARAMS.a3=1000/(30e3)^3;
143 -    PARAMS.amp=0; % amplitude of sinusoidal perturbation
144 -    PARAMS.lam=1; % wavelength of sinusoidal perturbation
145 -    PARAMS.ynTreset=1; % if ==1, reset T=T0 where im==1 (sticky layer)
146 -    PARAMS.T0=0;
147      % reference values for the constant diffusivity thermal solver
148      % (kappa = kref / (rhoref*cpref))
149 -    PARAMS.rhoref=MAT(2).rho0;
150 -    PARAMS.kref=3;
151 -    PARAMS.cpref=1000;
```

The present version of the thermal solver uses a constant diffusivity that can be set here as kref / (rhoref×cpref)

This only gets used if PARAMS.Tsolve=1

Input file structure

Picard iterations parameters

```
152     % Picard iterations
153 -   PARAMS.Npicard_min=3; % minimum number of Picard iterations per time step
154 -   PARAMS.Npicard_max=50; % maximum number of Picard iterations per time step
155 -   PARAMS.conv_crit1=0.01;
156 -   PARAMS.conv_crit2=0.03;
157     % convergence is assumed if a fraction of the domain smaller
158     % than PARAMS.conv_crit2 * domain size has more than relative
159     % change of PARAMS.conv_crit1 in strain rate and velocity field
160
```

This run will always use at least 3 Picard iterations, but will stop after 50, regardless of the convergence criterion.

Input file structure

Picard iterations parameters

```
152     % Picard iterations
153 -   PARAMS.Npicard_min=3; % minimum number of Picard iterations per time step
154 -   PARAMS.Npicard_max=50; % maximum number of Picard iterations per time step
155 -   PARAMS.conv_crit1=0.01;
156 -   PARAMS.conv_crit2=0.03;
157     % convergence is assumed if a fraction of the domain smaller
158     % than PARAMS.conv_crit2 * domain size has more than relative
159     % change of PARAMS.conv_crit1 in strain rate and velocity field
160
```

Convergence is assumed when the strain rate and velocity field change by more than 1% in only 3% of the domain or less. This formulation helps when iterating over the formation of narrow shear bands.

See **SiStER_run_Picard_iterations**

Input file structure

Boundary conditions (pressure)

```

% BOUNDARY CONDITIONS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% pressure
PARAMS.p0cell=0; % pressure in the top-left corner of the domain (anchor point)

```

Pressure is set to 0 in the top-left corner of the domain. This is done to anchor the solution to a specific pressure and stabilize the inversion.

Input file structure

```
167
168     % flow
169
170     % boundary conditions
171     % entries in BC correspond to
172     % 1/ rollers? (1=yes, 0=no)
173     % 2/ type of velocity normal to boundary (0=constant)
174     % 3/ value of normal velocity
175
176 -   BC.top=[1 0 1.0563e-10];
177 -   BC.bot=[1 0 -1.0563e-10];
178 -   BC.left=[1 0 -3.1688e-10];
179 -   BC.right=[1 0 3.1688e-10];
180 -   PARAMS.BalanceStickyLayer=1; % if set to 1, the code will reset the inflow
181     % / outflow BCs to balance the inflow / outflow of sticky layer material,
182     % and rock separately, based on the position of the sticky layer / air
183     % interface
184
```

Boundary conditions (flow)

BCs are specified in this BC structure for the top, bottom, left and right edges of the domain. The first entry is 1 for free slip (d v_tangential / d normal = 0), 0 for no-slip (v_tangential = 0)

Input file structure

```
167
168     % flow
169
170     % boundary conditions
171     % entries in BC correspond to
172     % 1/ rollers? (1=yes, 0=no)
173     % 2/ type of velocity normal to boundary (0=constant)
174     % 3/ value of normal velocity
175
176 -   BC.top=[1 0 1.0563e-10];
177 -   BC.bot=[1 0 -1.0563e-10];
178 -   BC.left=[1 0 -3.1688e-10];
179 -   BC.right=[1 0 3.1688e-10];
180 -   PARAMS.BalanceStickyLayer=1; % if set to 1, the code will reset the inflow
181     % / outflow BCs to balance the inflow / outflow of sticky layer material,
182     % and rock separately, based on the position of the sticky layer / air
183     % interface
184
```

Boundary conditions (flow)

The second entry specifies the type of normal velocity BC –
it can only be 0 in the current version, which enforces a constant normal velocity (m/s), specified in the third entry (careful with sign conventions, and make sure inflow matched outflow!)

Input file structure

```
167
168     % flow      Boundary conditions (flow)
169
170     % boundary conditions
171     % entries in BC correspond to
172     % 1/ rollers? (1=yes, 0=no)
173     % 2/ type of velocity normal to boundary (0=constant)
174     % 3/ value of normal velocity
175
176 -   BC.top=[1 0 1.0563e-10];
177 -   BC.bot=[1 0 -1.0563e-10];
178 -   BC.left=[1 0 -3.1688e-10];
179 -   BC.right=[1 0 3.1688e-10];
180 -   PARAMS.BalanceStickyLayer=1; % if set to 1, the code will reset the inflow
181     % / outflow BCs to balance the inflow / outflow of sticky layer material,
182     % and rock separately, based on the position of the sticky layer / air
183     % interface
184
```

In this example, a velocity of 1 cm/yr is applied towards the right on the right edge, and – 1 cm/yr on the left edge.

The top and bottom velocities are set to balance this outflow.

Input file structure

```
167
168     % flow
169
170     % boundary conditions
171     % entries in BC correspond to
172     % 1/ rollers? (1=yes, 0=no)
173     % 2/ type of velocity normal to boundary (0=constant)
174     % 3/ value of normal velocity
175
176 -   BC.top=[1 0 1.0563e-10];
177 -   BC.bot=[1 0 -1.0563e-10];
178 -   BC.left=[1 0 -3.1688e-10];
179 -   BC.right=[1 0 3.1688e-10];
180 -   PARAMS.BalanceStickyLayer=1; % if set to 1, the code will reset the inflow
181     % / outflow BCs to balance the inflow / outflow of sticky layer material,
182     % and rock separately, based on the position of the sticky layer / air
183     % interface
184
```

Boundary conditions (flow)

Setting this parameter to 1 will balance the inflow of sticky air and rock separately, based on the position of the air-rock interface at any given time (See **SiStER_flow_solve**)

Input file structure

Boundary conditions (temperature)

```
186      % thermal
187
188      % entries in BCtherm correspond to
189      % 1/ type? (1=Dirichlet, 0=Neumann)
190      % 2/ value
191 -    BCtherm.top=[1 0];
192 -    BCtherm.bot=[1 1000];
193 -    BCtherm.left=[2 0];
194 -    BCtherm.right=[2 0];
```

Those will be used if PARAMS.Tsolve is set to 1.
If the first entry is 1 (Dirichlet), the temperature is set equal to the second entry. If the first entry is 2 (Neumann), the gradient of temperature is set equal to the second entry.

This example has T = 0 on the top, T=1000ºC on the bottom,
and no heat flux through the sides.

Visualizing the output

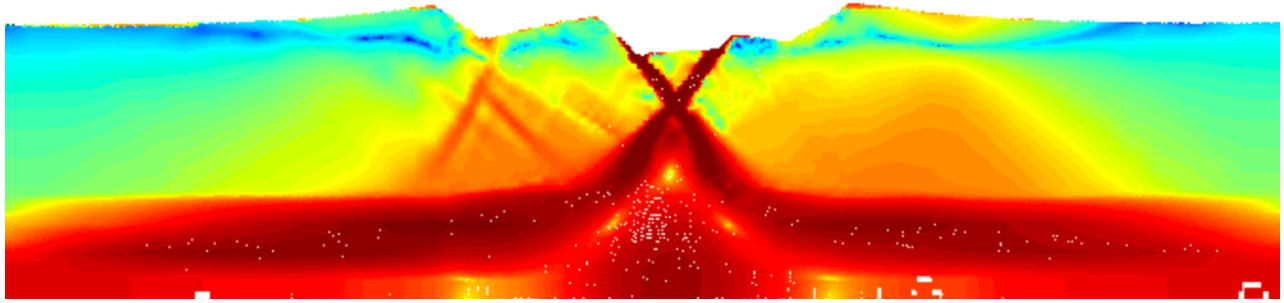
Output files contain (among other things):
time (in seconds)
Nodal arrays:
X, Y (coordinates of shear nodes)
vx, vy (velocities) and p (pressure)
Marker arrays:
xm, ym (marker coordinates)
im (phase) etam (viscosity) rhom (density) Tm (temperature)
ep (accumulated plastic strain) sxxm, sxym (deviatoric stresses)
epsIIm (strain rate)
Topography:
topo_x, topo_y (coordinates of a dense chain of markers tracking the sticky layer – rock interface)

More variables can be output- see l. 96 of **SiStER_MAIN**

Visualizing the output

To visualize marker arrays (here the strain rate at iteration 100, excluding the sticky air layer):

```
>> fastscatter(xm(im>1)/1e3,ym(im>1) /1e3,log10(epsIIm(im>1)),'markersize',1);
>> axis ij
>> axis equal
>> grid off
>> colorbar
>> caxis([-18 -13])
```



And the topography:

```
>> plot(topo_x,topo_y,'k.')
>> set(gca,'YDir','reverse')
```