

# ASPECT Hackathon 2023

<b>Meeting places</b>	<b>3</b>
<b>Land acknowledgement</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>Timeline</b>	<b>4</b>
<b>Participants and areas of interest</b>	<b>5</b>
<b>Resources</b>	<b>10</b>
<b>Things to do before the start of the hackathon</b>	<b>10</b>
Installing deal.II on Mac	10
Installing deal.II on Linux	10
Installing deal.II on Windows	10
Visual Studio Code Tutorial	11
Git Tutorial	11
Debugging tutorial	12
<b>Report on projects the participants worked on</b>	<b>14</b>
Removing deprecated deal.II version checks	14
Updates to the Peierls creep formulation	14
Phase transition cookbook	15
Make DebugRelease the new default build type	16
Remove LaTeX documentation	16
Improve default installation instructions and install SUNDIALS	16
SUNDIALS in candi	16
Sea level postprocessor tested	17
Co-Author Graph	18
An example for water release, migration and absorption	20
Continuous Integration testing on MacOS M1	22
A plugin for a random initial temperature perturbation	22
A cookbook that uses tomography data to develop mantle convection models	22
A cookbook on subduction initiation	23
A cookbook on deformation of inclusions	24
A cookbook on 3D rift interaction	24
A (preliminary) cookbook on Mars topography	25
A cookbook on 3D lower crustal flow	26
A cookbook for 2D corner flow subduction (van Keken 2008)	26
Use a phase diagram parametrization to simulate reactive fluid transport	27
A simple 2D cookbook for a dehydrating subducting slab	28
Finalizing VE(V)P implementation updates	28
Include the initial topography in the lithostatic pressure profile for traction boundary conditions	29

Add strain rate tensor output on the surface of the domain	29
Add lookup table functionality for viscoplastic phases	29
An improved microstructural evolution algorithm for the LPO particle plugin	30
Initial coupling between ASPECT and the FastScape C++ library	30
<b>Add plasticity, lateral viscosity profile, and P-T dependent conductivity to the Entropy Plugin</b>	<b>32</b>
Make assembly of the advection equations more flexible	32
Solve coupled ODEs with SUNDIALS	32
Default Field Types implemented	32
Add CPO main post-processor	33
Transform and rotate functions for full stiffness tensor, Voigt matrix and vector-tensor utilities	33
Levi-Cevita tensor for the tensors utilities	33
Add astyle installation instructions to the manual	33
Add a CPO postprocessor which can compute the elastic tensor (work in progress)	33
<b>A material for simulating reactive fluid transport</b>	<b>33</b>
<b>Fixed a bug in the strain rheology module</b>	<b>34</b>
<b>Functionality for averaging material properties between nonlinear iterations</b>	<b>34</b>
<b>Allowing multiple traction models for a boundary</b>	<b>34</b>
An anisotropic viscosity material model (work in progress)	34
<b>General code updates</b>	<b>35</b>
A compositional viscosity in the Steinberger material model	35
Education survey	36
<b>Statistics about ASPECT's growth during the hackathon</b>	<b>38</b>

# Meeting places

This is a hybrid workshop. Virtual participants will meet in the following places online:

- **Zoom Meeting room** - daily rounds @9A  
<https://ucdavis.zoom.us/j/92250758840?pwd=VThXdHRnLy9haE9xM29FU3V4dTdpUT09>
- **Slack Workspace for asynchronous communication (cig-aspect.slack.com):**  
[https://join.slack.com/t/cig-aspect/shared\\_invite/zt-1y kz5bstz-76~4k6w1GYcimhOkVMTX\\_g](https://join.slack.com/t/cig-aspect/shared_invite/zt-1y kz5bstz-76~4k6w1GYcimhOkVMTX_g)  
Ask to join channel: hackathon-2023 (need to message one of the organizers)
- **Hackathon log (this document):**  
[https://docs.google.com/document/d/1O-t1IRgllXc\\_rUHB\\_VbpB02rKtSR38jcDMTEdY4oBaY/edit?usp=sharing](https://docs.google.com/document/d/1O-t1IRgllXc_rUHB_VbpB02rKtSR38jcDMTEdY4oBaY/edit?usp=sharing)
- **Facebook:**  
<https://www.facebook.com/groups/1438286143140671> (Cedric & Lorraine admin)
- **Spotify playlist:** (Lorraine owner)  
ASPECT 2023  
<https://open.spotify.com/playlist/3kbTCScaYOkXdHB8Eh6gwr?si=90c0ab2bf6b64ec0>
- **YouTube playlist:**  
<https://youtube.com/playlist?list=PLdy04DoEepEzpKnMYERibFI45SxA1anlJ>
- **Google drive for photos**  
<https://drive.google.com/drive/folders/1Uc8ntKJiiMto3aDkzyWCiN-Q12ZXsSxH?usp=sharing>
- **Shared folder for presentations etc.**  
[https://drive.google.com/drive/u/0/folders/1LPozTCy2\\_aQMPa5MYWPf2U7CjKKjreTv](https://drive.google.com/drive/u/0/folders/1LPozTCy2_aQMPa5MYWPf2U7CjKKjreTv)

## Land acknowledgement

We acknowledge the indigenous people and land in which we are gathered. Lincoln City, Oregon has been home to the Siletz, Confederated Tribes of the Grand Ronde, Cayuse, Umatilla, Walla Walla, Confederated Tribes of the Siletz Indians peoples who have provided stewardship of this land over many centuries. We are honored and grateful to be here today on their traditional lands.

# Introduction

To further develop the geodynamic modeling code ASPECT and to grow and foster its user community, 22 users and developers of ASPECT worked in-person and virtually over two weeks in July 2023.

Below is the timeline and a description of the individual contributions.

## Timeline

Day	Scheduled items (in Pacific time; US East: +3h; Central Europe: +9h)
Thursday, 07/06	Evening: Arrival
Friday, 07/07	9 am: Project discussion, Planning meeting
Saturday, 07/08	9 am: Daily rounds 10 am: Rene: Using ASPECT on geodynamics.org After lunch: Rene: Debugging tutorial
Sunday, 07/09	9 am: Daily rounds 10 am: Lorraine Publishing data products with ASPECT
Monday, 07/10	9 am: Daily rounds 10 am: Changes for ASPECT 3.0 2 pm: Alternatives to slack: We will decide on Zulip or Element/Matrix after the hackathon.
Tuesday, 07/11	Day off
Wednesday, 07/12	9 am: Daily rounds 11 am: Anne+Michael: Fastscape coupling 2 pm: Wolfgang: Nonlinear solver packages
Thursday, 07/13	9 am: Daily rounds 10 am: Menno: Updates to World Builder 2 pm: Cedric: Fieldstone
Friday, 07/14	9 am: Daily rounds 10 am: Slack decision? 2 pm: John:
Saturday, 07/15	10 am: Departure for all



## Participants and areas of interest

Name, affiliation, email	Time zone (virtual only)	Goals and interests for this hackathon
Rene Gassmoeller, University of Florida, rene.gassmoeller@mailbox.org		<ol style="list-style-type: none"> <li>1. Help others achieve their goals</li> <li>2. Review pull requests</li> <li>3. Remove LaTeX documentation</li> <li>4. Make GMG default solver</li> <li>5. Change a lot of parameters (particles)</li> <li>6. Create installation documentation local.cfg</li> </ol>
Lorraine Hwang UC Davis ljhwang@ucdavis.edu		<ol style="list-style-type: none"> <li>1. Logistics</li> <li>2. ASPECT Networks</li> <li>3. Notebooks</li> <li>4. Baking</li> </ol>
Wolfgang Bangerth Colorado State University bangerth@colostate.edu		<ol style="list-style-type: none"> <li>1. Review pull requests</li> <li>2. Help others</li> <li>3. Write documentation</li> </ol>
Juliane Dannberg University of Florida judannberg@gmail.com		<ol style="list-style-type: none"> <li>1. Help others</li> <li>2. Review pull requests</li> <li>3. Stress boundary conditions</li> <li>4. Axisymmetric geometry</li> <li>5. Entropy model for several compositions</li> <li>6. Upper mantle melting model</li> <li>7. Cookbook for changing volatile solubility</li> </ol>
Timo Heister heister@clemson.edu	US, Eastern Standard Time	<ol style="list-style-type: none"> <li>1. Review pull requests</li> <li>2. Help others</li> <li>3. Infrastructure work (testing, cmake, etc.)</li> <li>4. Linear solvers (multigrid, Schur complement)</li> </ol>
Menno Fraters UC Davis menno.fraters@tutano.ta.com		<ol style="list-style-type: none"> <li>1. Help others</li> <li>2. Review pull requests</li> <li>3. Add documentation about how to use the Newton solver</li> <li>4. Add documentation about how to use the world builder which could double as a subduction cookbook.</li> <li>5. Add CPO related code</li> <li>6. Explorer the option of an Euler pole boundary plugin</li> </ol>

		<ol style="list-style-type: none"> <li>7. Slab2 integration</li> <li>8. Discuss addition of melt and GMG to Newton solver</li> </ol>
<p>John Naliboff New Mexico Tech john.naliboff@nmt.edu</p>	<p>US., Mountain Standard Time</p>	<ol style="list-style-type: none"> <li>1. Help others achieve their goals</li> <li>2. Review pull requests</li> <li>3. Finish existing PRs</li> <li>4. Update continental extension cookbook to use recent features</li> <li>5. Add a section to the manual about lithospheric, rheology and material model structure</li> <li>6. PRs related to VEP implementation (Anne, Bob, others) <ol style="list-style-type: none"> <li>a. Averaging viscosity between time steps</li> <li>b. Local iterative scheme for stresses between deformation mechanisms (diffusion-dislocation, creep-VE, VE - plasticity)</li> </ol> </li> <li>7. Help with features for modeling complex fluid transport and reactions (Daniel, Juliane, Ryan, others) <ol style="list-style-type: none"> <li>a. Coupling porosity to water fugacity in olivine flow laws (existing PR)</li> <li>b. Create new composite material model to couple fluid flow with the visco plastic model</li> <li>c. Implement grain-size evolution in the VP model and couple it to permeability</li> </ol> </li> <li>8. Tools to extract data from 3D global simulations for use in regional simulations</li> </ol>
<p>Anne Glerum</p>	<p>CEST</p>	<ol style="list-style-type: none"> <li>1. VEP implementation PR (Bob, John)</li> <li>2. FastScape PR</li> <li>3. Subduction cookbook</li> <li>4. Lithospheric melt model (Bob)</li> <li>5. Help and review pull requests</li> </ol>
<p>Arushi Saxena University of Florida saxena.arushi@ufl.edu</p>		<ol style="list-style-type: none"> <li>1. Merge the open PR to develop mantle flow models based on tomography and plate boundary geometries to best-match the surface observables.</li> <li>2. Merge a local World builder branch used with the global mantle flow models into main.</li> <li>3. Work on the open issues in the ASPECT github repository.</li> </ol>
<p>Maaïke Weerdesteijn UiO Oslo m.f.m.weerdesteijn@geo.uio.no</p>		<ol style="list-style-type: none"> <li>1. Find cause for free surface instabilities in a spherical domain with boundary traction</li> <li>2. Potentially a cookbook for regional GIA modeling in a box geometry</li> </ol>

		<ol style="list-style-type: none"> <li>3. Integrate sea level pp, talk about iterative structure</li> <li>4. Learn about VEP recent developments</li> </ol>
Cedric Thieulot c.thieulot@uu.nl		<ol style="list-style-type: none"> <li>1. Merge existing prm files for paper replication → cookbook</li> <li>2. Look into axisymmetric formulation</li> <li>3. Herschel-Bulkley model?</li> </ol>
Daniel Douglas New Mexico Tech daniel.douglas@student.nmt.edu	US, Mountain Standard Time	<ol style="list-style-type: none"> <li>1. Create a material model which advects volatiles</li> <li>2. Couple rheology to grain size</li> <li>3. Investigate adding the van Keken 2008 corner flow benchmark</li> </ol>
Yijun Wang		<ol style="list-style-type: none"> <li>1. Create the anisotropic viscosity material model using Hill parameters. (Potentially implement it using both dislocation and diffusion creep)</li> <li>2. Add parameters to read coefficients and intercepts for calculating Hill parameters</li> <li>3. Add a particle property that is similar to Bingham average but returns both eigenvectors and eigenvalues separately. Access this property using compositional fields in the parameters</li> </ol>
Bob Myhill		<ol style="list-style-type: none"> <li>1. Help others.</li> <li>2. Work on getting VEP model bug tested and ready for merge (with JN, AG, RG, others)</li> <li>3. Implement four component hydrous melting model for the mantle transition zone (with JD)</li> <li>4. Composite VEP model (depends on #2)</li> <li>5. Discuss multicomponent entropy method (with Ranpeng, JD)</li> <li>6. Discuss grain scale geodynamic models (with CT, JD, JN, other interested parties)</li> <li>7. Discuss Newton iteration on log values rather than real values (diffusion-dislocation, new VEP, Peierls, ...).</li> </ol>
Ryan Stoner		<ol style="list-style-type: none"> <li>1. Make new pull request for material model plugin for rheology lookups</li> <li>2. Volatile transport material model</li> <li>3. Rheology with fugacities</li> </ol>
Michael Pons		<ol style="list-style-type: none"> <li>1. Integrating Fastscape C++ share library in ASPECT, change in Cmakelist.txt</li> <li>2. Write initial fastscape.cc .h for FS C++ use</li> <li>3. Run test (working for a short time then problem of memory)</li> </ol>

		<ol style="list-style-type: none"> <li>4. Helping Chameera with the initial ASPECT - FS installation.</li> <li>5. Presentation on the ASPECT-Fastscape coupling (Slides available in the share repository)</li> </ol>
Haoyuan Li		<ol style="list-style-type: none"> <li>1. Finish the cookbook for phase transitions (layered)</li> <li>2. Improve on the Peierls creep rheology</li> <li>3. Test new implementations</li> <li>4. Expand the phase diagram cookbook to a phase &amp; deformation mechanism cookbook.</li> <li>5. (others: questions I accumulated).</li> </ol>
Ranpeng Li		<ol style="list-style-type: none"> <li>1. Add plasticity, lateral viscosity limit, p-t dependent conductivity to current entropy plug-in</li> <li>2. Make the entropy plug-in a material model</li> <li>3. Start on multi-component entropy method</li> </ol>
Chameera Silva (CERI) UofM		<ol style="list-style-type: none"> <li>1. Working on allowing to set up multiple traction models for a single boundary {eg. initial lithostatic pressure + traction function }</li> <li>2. Coupled aspect with fastscape fortran (with Michael)</li> <li>3. Ran couple of continental extension examples with coupled code</li> <li>4. Testing updated continental extension cookbook with new solver setup</li> </ol>
Srivatsan Vedavyas		<ol style="list-style-type: none"> <li>1. Working on D-Rex++, an improved microstructural evolution plugin for deforming olivine-enstatite aggregates.</li> </ol>
Mohamed Guiza		Education survey
Poulami Roy		<p>Add compositional dependence on viscosity in the Steinberger material model.</p> <ol style="list-style-type: none"> <li>1. Introduced a parameter named as “Viscosity prefactors” which is a dimensionless quantity and is multiplied with original Steinberger material model viscosity based on volume fraction.</li> <li>2. Run test with multicomponent parameter file and check if each composition can have different relative viscosity than the reference profile.</li> <li>3. Make pull request.</li> </ol> <p>Started initial discussion with Menno of introducing the Perovskite (Bridgmanite) anisotropy model inside the CPO module.</p>



# Resources

## Things to do before the start of the hackathon

- Create an account on <https://github.com> if you don't have one yet.
- Put your name into the table ["Participants and areas of interest"](#) above and fill in your goals and interests for the hackathon.
- Join the Slack workspace.
- Install the latest developer version of ASPECT (using deal.II version 9.4.0 or or 9.5.0 or the current deal.II development version).

## Installing deal.II on Mac

Download and install the Mac OS X "dmg" package from <https://dealii.org/download.html> .

## Installing deal.II on Linux

If you are running Ubuntu, and if you have *root* rights on this machine, then issue the following commands:

```
sudo apt-get install -y software-properties-common
sudo add-apt-repository ppa:ginggs/deal.ii-9.4.0-backports
sudo apt-get update
sudo apt-get install -yq --no-install-recommends libdeal.ii-dev
```

If you are running on any other Linux variant, the following commands will install deal.II and all of its dependencies from scratch by compiling them from source:

```
git clone https://github.com/dealii/candi.git
cd candi
./candi.sh --packages="once:astyle once:hdf5 once:p4est once:trilinos dealii" -j 2 -y
. ~/deal.ii-candi/configuration/enable.sh      # note the dot at the beginning of the line
```

## Installing deal.II on Windows

Follow the instructions linked to from here, installing first the Windows Subsystem for Linux (WSL) and then deal.II within it:

<https://github.com/dealii/dealii/wiki/Getting-deal.II#windows>

## Visual Studio Code Tutorial

- VS Code is an Integrated Development Environment (IDE) that simplifies programming
- It is free, powerful, and used by the majority of open-source software developers
- If you are already comfortable with a different IDE stick to it, if you do not use an IDE so far, please install VS code
- It is simpler for us to explain and help you if most of us use the same IDE
- How to get: <https://code.visualstudio.com/>
- Documentation: <https://code.visualstudio.com/docs>
- Necessary/Useful extensions for this hackathon:
  - <https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools>
  - <https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens>
  - <https://marketplace.visualstudio.com/items?itemName=davydden.dealii-prm>
  - <https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare>

## Git Tutorial

- Git commands cheat sheet: <https://education.github.com/git-cheat-sheet-education.pdf>
  - Github workflow: <https://guides.github.com/introduction/flow/>
  - Git tutorial: <https://swcarpentry.github.io/git-novice/>
1. Explain and set up Git:
    - a. Git install: <https://carpentries.github.io/workshop-template/#git>
    - b. <https://swcarpentry.github.io/git-novice/01-basics.html>
    - c. <https://swcarpentry.github.io/git-novice/02-setup.html>
    - d. Config git name: ``git config --global user.name "Vlad Dracula"`
    - e. Config git email: ``git config --global user.email "vlad@tran.sylvan.ia"`
  2. Explain Github Workflow:
    - a. <https://guides.github.com/introduction/flow/>
    - b. Ensure you have forked ASPECT's repository (you should own your\_username/aspect on github)
    - c. Ensure proper remotes are set up (remote 'upstream' pointing to geodynamics/aspect, remove 'origin' pointing to your\_username/aspect)
  3. Setup ASPECT in VS Code
  4. Walkthrough (these are terminal commands, I will walk you through the IDE instead)
    - a. Create Branch
      - i. `'git checkout main'`
      - ii. `'git pull upstream main'`
      - iii. `'git checkout -b remove_unused_option'`
    - b. Make changes to files that contain a clause DEAL\_II\_VERSION\_GTE. This preprocessor directive is used to determine which deal.II version is used. ASPECT now requires 9.4.0 so every check for 9.4.0 or 9.3.2 is superfluous (10.0.0 was renamed to 9.5.0 so that is superfluous as well). These files are:
      - i. `include/aspect/compat.h` (Moh)

- ii. source/particle/generator/interface.cc (Yijun)
  - iii. source/particle/property/grain\_size.cc (Ranpeng)
  - iv. source/particle/property/elastic\_stress.cc (Poulami)
  - v. source/particle/world.cc (Chameera)
  - vi. source/postprocess/particles.cc (Srivatsan)
  - vii. source/postprocess/visualization.cc (Maaike)
  - viii. source/material\_model/melt\_global.cc (Michael from Poulami account)
  - ix. source/simulator/core.cc (Maaike)
  - x. source/simulator/assemblers/interface.cc (Yijun)
  - xi. source/simulator/stokes\_matrix\_free.cc (Moh)
  - c. After making the practice change: Create commit:
    - i. 'git add FILE'
    - ii. 'git commit -m 'Removed a deprecated option''
  - d. Push and open PR
    - i. 'git push origin remove\_unused\_option'
    - ii. Open PR on github (CTRL-Click on shown link)
  - e. Wait for review
  - f. Address review (repeat steps b,c,d)
  - g. Now show how to change documentation via the web interface.
  - h. Success!
5. Now repeat the steps in 4. on your own. Pick a section of the manual in doc/sphinx/user that interests you. Find a sentence or description or formula to improve. Then repeat 4 for as often as you want or can!

## Debugging tutorial

Content of .vscode/launch.json:

```
{
  // Use IntelliSense to learn about possible attributes.
  // Hover to view descriptions of existing attributes.
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
  "configurations": [
    {
      "name": "C++ Launch",
      "type": "cppdbg",
      "request": "launch",
      "program": "${workspaceFolder}/build/aspect",
      "args": ["cookbooks/convection-box/convection-box.prm"],
      "environment": [{ "name": "config", "value": "Debug" }],
      "cwd": "${workspaceFolder}"
    }
  ]
}
```





# Report on projects the participants worked on

## Removing deprecated deal.II version checks

(Pretty much everyone)

We removed outdated checks for deal.II versions that are no longer supported by ASPECT. Specifically, the current ASPECT version requires deal.II 9.4, so any checks in the code base that conditionally work for older versions can be removed.

## Updates to the Peierls creep formulation

(Haoyuan Li, Bob Myhill, Daniel Douglas, John Naliboff)

1. Use log value of the gradient (stress, strain rate) in the “exact” scheme  
Using the logarithm of the gradient values instead of  $(d(\text{stress})/d(\text{strain rate}))$ , the idea is the logarithm values are in a safer range than the values themselves (e.g. -15 vs  $1e-15$ ). Also, there is no  $A$  in the expression of  $d(\log\_stress)/d(\log\_strain\ rate)$ , so there won't be an issue if  $A$  is too small (at least not from the gradient and iterations).  
I have done 2 tests with Python where I used both implementations of gradient and set the same tolerance (relative residue,  $1e-8$ ). In both tests, the iterations using the logarithm values converge faster (iterations 8 vs 4, and iterations 11 vs 4).  
(Among the two, the function `peierls_visc_from_edot_newton` is implemented with the logarithm value,  $n$  is the number of iterations.)
2. Add the option to apply a strict cutoff on the minimum stress from the Peierls creep mechanism.
3. Future plan: Change iteration method in other material models as well.

```

126 # test 1.1: nolog, medium T
127 P = 1.7882e9
128 T = 1330.38
129 edot = 1e-15
130 etap, sigma, diff, n = flf.peierls_visc_from_edot_newton_nolog('MK10',P,T, edot, 1e-8)
131 assert(abs((etap - 2.273489862826893e+22)/2.273489862826893e+22) < 1e-5)
132 assert(n == 8)
133 # test 1.2: log, medium T
134 P = 1.7882e9
135 T = 1330.38
136 edot = 1e-15
137 etap, sigma, diff, n = flf.peierls_visc_from_edot_newton('MK10',P,T, edot, 1e-8)
138 assert(abs((etap - 2.2734917295040908e+22)/2.2734917295040908e+22) < 1e-5)
139 assert(n == 4)
140 pass
141 # test 2.1: nolog, high T
142 P = 1.7882e9
143 T = 3500.0
144 edot = 1e-15
145 etap, sigma, diff, n = flf.peierls_visc_from_edot_newton_nolog('MK10',P,T, edot, 1e-8)
146 assert(abs((etap - 1.0219201878630164e+19)/1.0219201878630164e+19) < 1e-5)
147 assert(n == 11)
148 # test 2.2: log, high T
149 P = 1.7882e9
150 T = 3500.0
151 edot = 1e-15
152 etap, sigma, diff, n = flf.peierls_visc_from_edot_newton('MK10',P,T, edot, 1e-8)
153 assert(abs((etap - 1.0219201878630164e+19)/1.0219201878630164e+19) < 1e-5)
154 assert(n == 3)

```

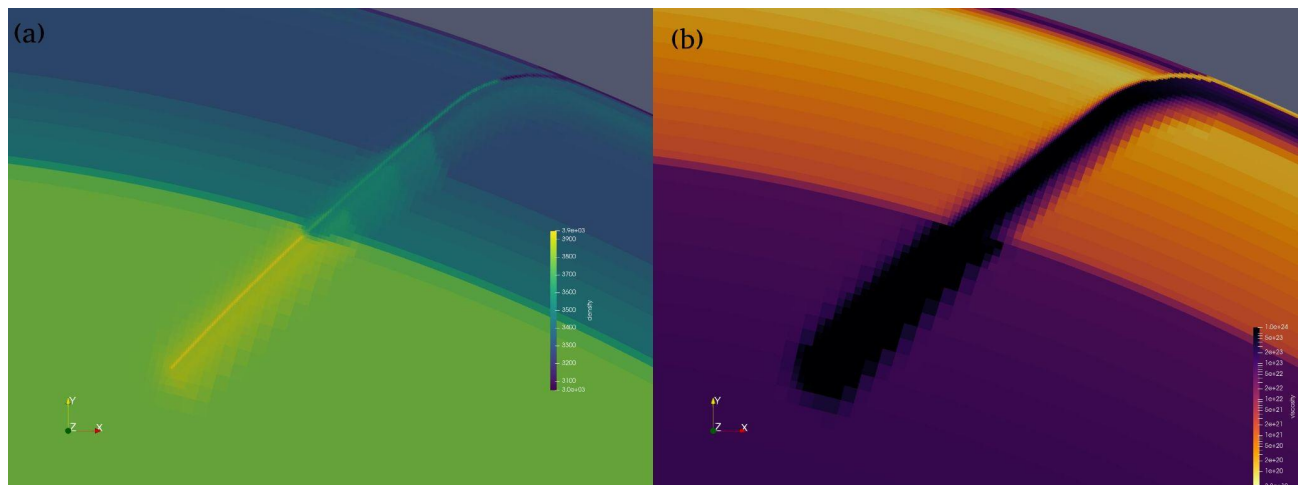
## Phase transition cookbook

(Haoyuan Li, Rene Gassmoeller)

We have summarized the usage of the phase transition interfaces in the visco plastic material model in this cookbook.

We first show how to set up the mantle phase transitions on a single composition (pyrolite) and then expand it to end member compositions (pyrolite, harzburgite, basalt). At the end, we show a setup with an initial subducting slab, with density and viscosity variations due to phase transitions.

(Left: Density. Right: Viscosity.)



## Make DebugRelease the new default build type

(Rene Gassmoeller)

ASPECT will now by default compile two executables in the build folder, one in DEBUG model and one in OPTIMIZED (“Release”) mode. This will make sure every user has access to both modes (and knows about both modes) and makes using ASPECT more convenient, as well as simplifying debugging.

## Remove LaTeX documentation

(Rene Gassmoeller)

Now that the transition of ASPECT's documentation to Sphinx and an online documentation is complete, I have removed the no longer necessary .tex source files of the old manual. I also updated the design of the webpage to a newer version of the Sphinx book theme.

## Improve default installation instructions and install SUNDIALS

(Rene Gassmoeller)

When compiling ASPECT using the candi program, candi used to compile a number of libraries that are not necessary for ASPECT and over the course of the past few years have caused some users a number of compile issues. In addition users would not know that there are optimization options available in candi. I have updated our installation instructions to include a custom configuration file for candi that only installs the necessary libraries, and the future dependency SUNDIALS including the recommended optimization flags. This should speed up user installations and avoid installation problems.

## SUNDIALS in candi

(Timo Heister)

SUNDIALS is now also enabled by default in candi.

## Various fixes for changes in deal.II master

(Timo Heister)

Many recent deprecations in deal.II require changes inside ASPECT. These have been implemented.

## Cookbook/Benchmark test fixes

(Timo Heister)

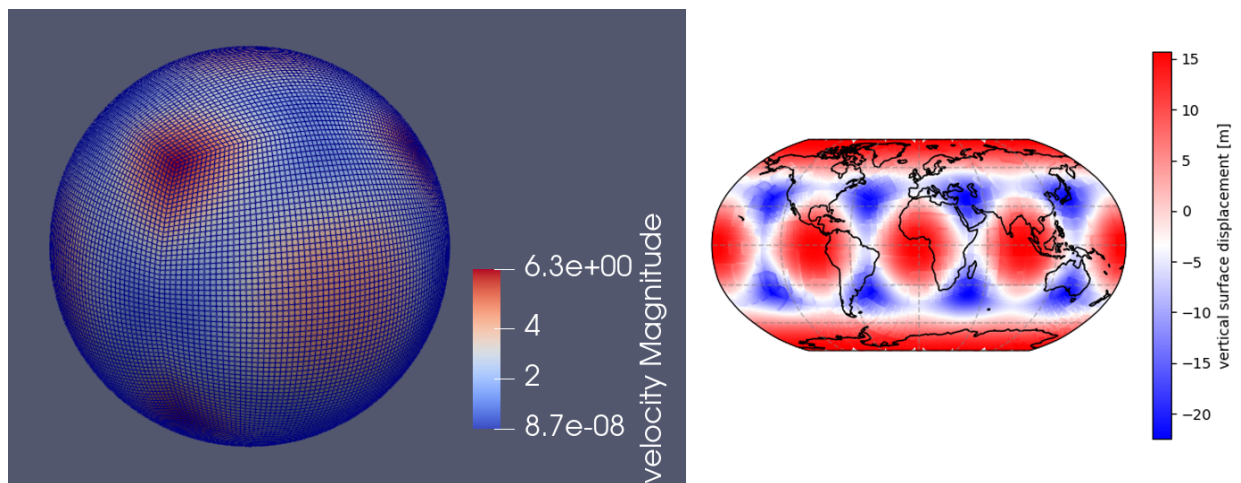
We ran all cookbooks and benchmarks and fixed various warnings and errors.

## Sea level postprocessor tested

(Maaïke Weerdesteijn)

The sea level postprocessor computes the sea level for glacial isostatic adjustment modeling. When ice melts and enters the ocean, the ocean water needs to be redistributed in a gravitationally consistent way. With the updated surface loading (ocean and ice) the free surface deformation needs to be computed iteratively before moving to the next time step. This postprocessor is intended for use with a deforming top surface (free surface) and the geoid postprocessor. After every step it computes the sea level based on the topography, ocean basin, ice melt, perturbed gravitational potential of the Earth model (still left to do: and gravitational potential of the ice load), relative to a reference datum (initial radius for a spherical shell geometry model). Currently, the iterative structure is not implemented. It is now tested (see the `sea_level_postprocessor` test) for a single time step and no iteration.

Note that when using a 3D spherical shell and free surface, the model first tries to find an equilibrium and thereby deforms (see Figure): the mesh nodes lay on a sphere, but the quadrature points may not. This (undesired) mesh deformation is larger for coarser meshes, and the result is a mesh-size-dependent unphysical deformation. The way to negate this numerical error is to first run your model without any boundary traction for X years (depending on mesh resolution). Then run another model with the boundary traction and subtract the undesired deformation from these free surface deformation.

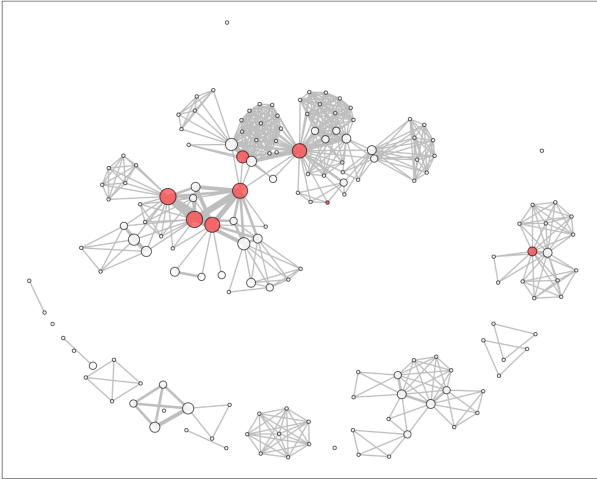


# Co-Author Graph

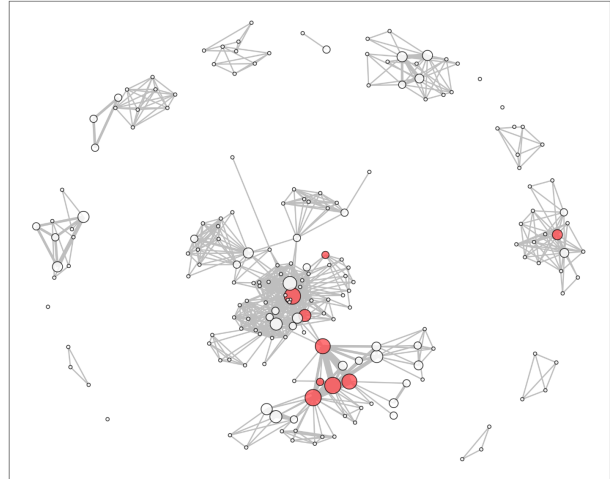
(Lorraine Hwang)

Below is the updated 2023 edition showing the relationship between authors of papers that cite ASPECT. In the past year, we have added 37 more authors.

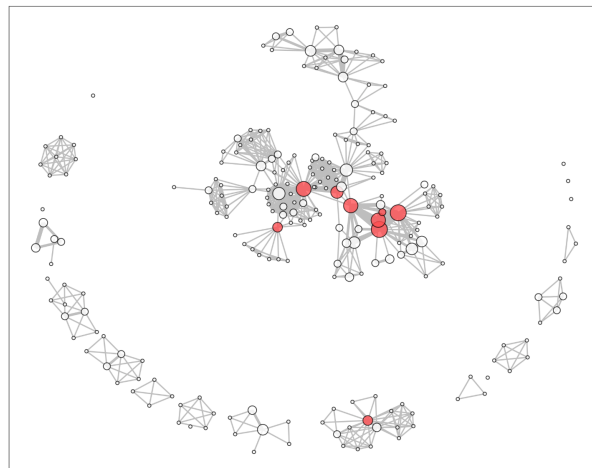
ASPECT Co-author Relationships



ASPECT Co-author Relationships



ASPECT Co-author Relationships



## ASPECT Co-author Relationships

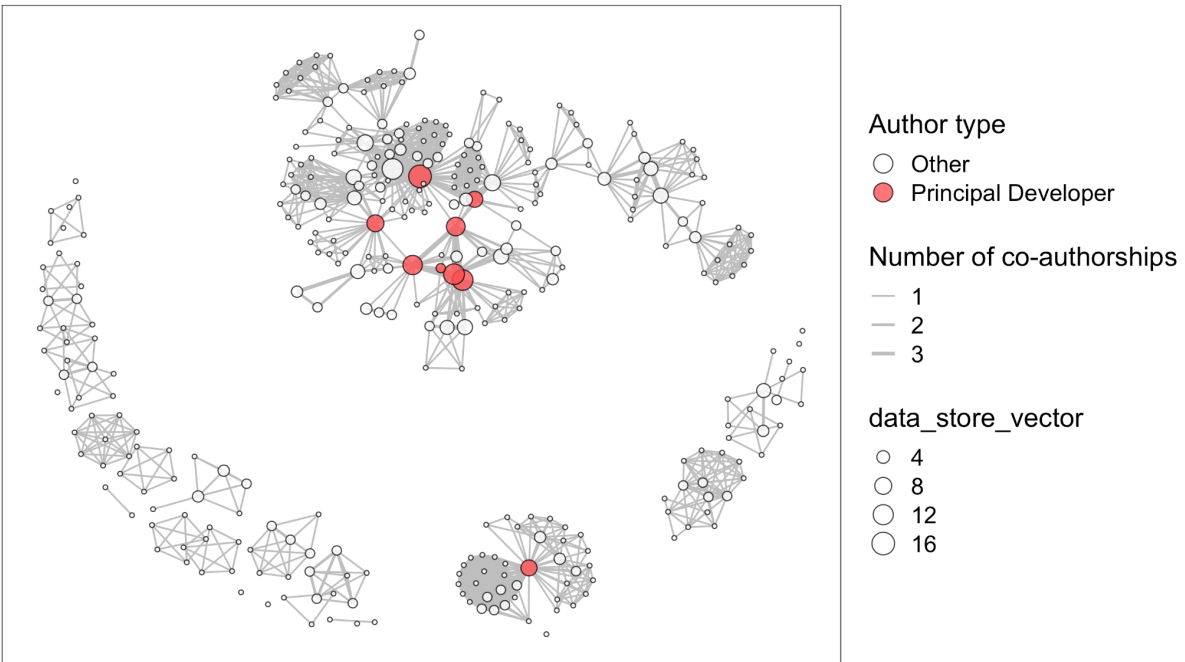


Figure: *Network graphs for 2020 (top left), 2021 (top right), 2022 (middle), and 2023 (bottom).  
The legend is the same for all plots.*

And the fine details ...

## ASPECT Co-author Relationships



Figure: 2023 network graph with author names. Legend is the same as above.

## An example for water release, migration and absorption

(Juliane Dannberg)

The implementation of the two-phase flow equations in ASPECT does not only describe the flow of a silicate melt through porous rocks, but can also be used to model the flow of other fluids,



such as water. Water in the Earth's mantle can either be bound in minerals, or it can be present as a fluid phase that can migrate relative to the solid rock. The fraction of this free water then represents the porosity of the material. Therefore, the terms describing melting and freezing of a silicate melt in the equations would instead describe the release and reabsorption of water into the rock, which are governed by the water solubility. I added a test case that shows how such a model for water solubility can be implemented in a material model in ASPECT, and demonstrates that the mass of water is conserved as water is released, migrates, and is being reabsorbed.

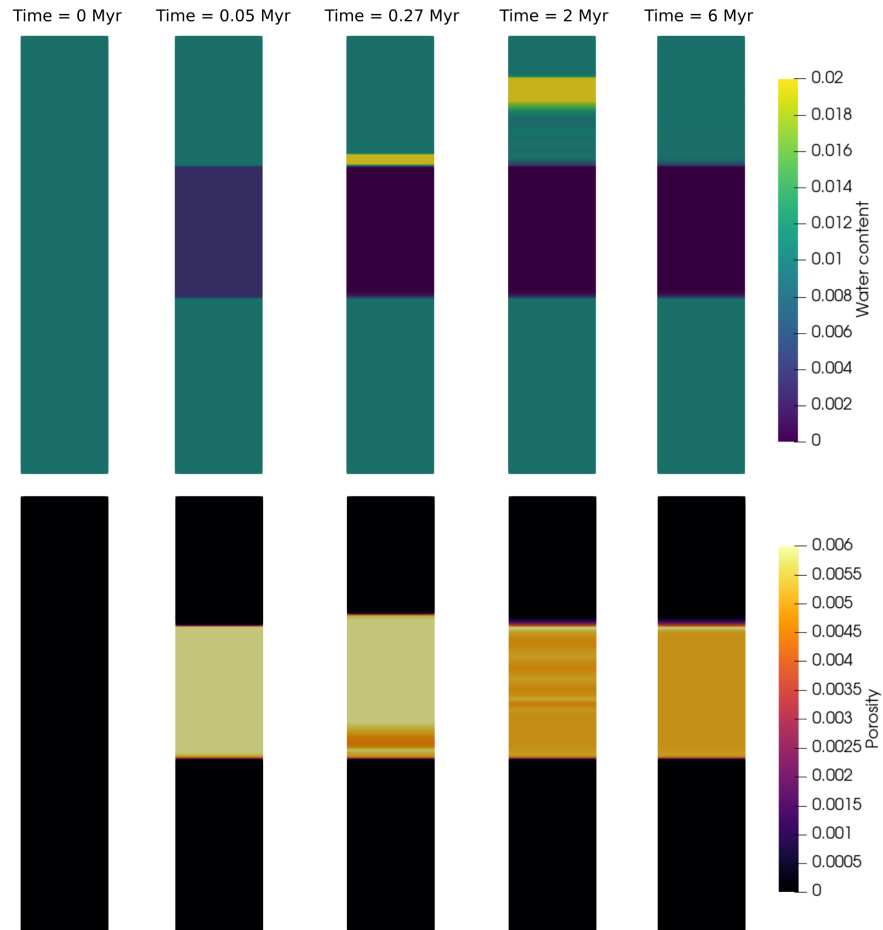


Figure: *Evolution of the bound water content (top) and the free water (bottom). Initially, all water is bound. Since the middle layer has a zero solubility, water is released and starts migrating upward with respect to the solid, initially leading to a higher bound water content when it reaches the top layer. As the model reaches steady state, the (bound) water content in the top and bottom layers are equal (1%) and the water content in the middle layer is 0.5%.*

## Continuous Integration testing on MacOS M1

(Timo Heister)

Every pull request is now also tested on MacOS on an M1 Mac Mini.

## A plugin for a random initial temperature perturbation

(Juliane Dannberg)

I added a plugin that allows adding a number of randomly placed Gaussian perturbations to the initial temperature. This can be useful in cases where one wants to initiate the temperature field with some noise that is independent of the model resolution and instead has a given wavelength.

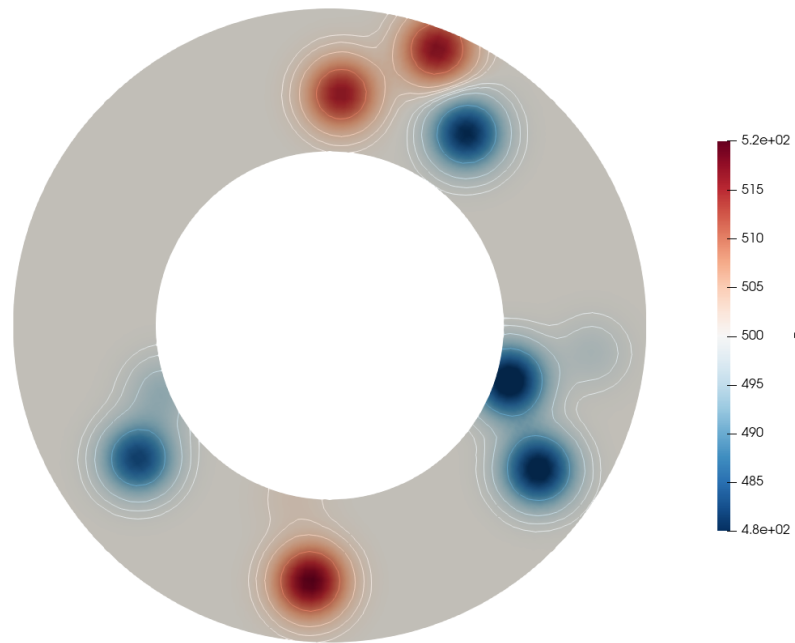


Figure: *Ten Gaussian temperature perturbations with an amplitude between  $\pm 25$  K in random locations in a spherical shell geometry.*

## A cookbook that uses tomography data to develop mantle convection models

(Arushi Saxena, Juliane Dannberg, Rene Gassmoeller)

We added a cookbook on how to set up global instantaneous models based on recent geophysical constraints with a heterogeneous density and viscosity distribution and weak plate boundaries prescribed using different plate boundary configurations. The setup can be modified

by the user to change the input geophysical constraints such as tomography, temperatures, viscosity/density scaling relative to the S-wave anomalies, plate boundary geometry, or to change the balance between the mantle forces such as the amount of friction between the plate boundaries or at the base of the plates, use of viscous cratons, or slabs.

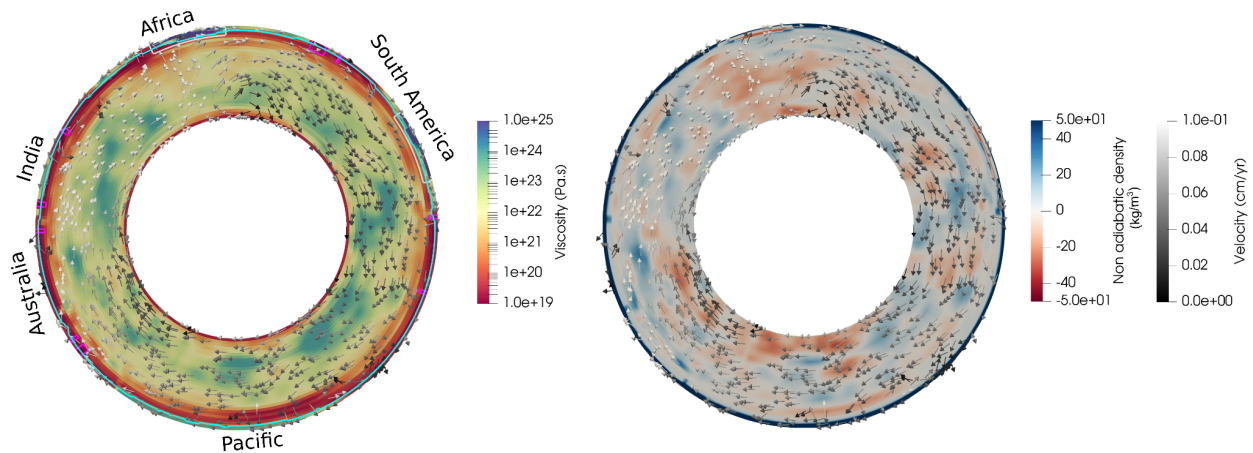
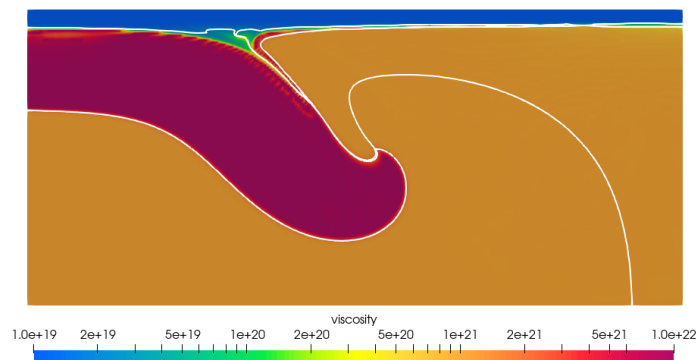


Figure: *Instantaneous mantle flow in the model overlying the heterogeneous lateral and radial viscosity distribution (left) and the non-adiabatic densities (right). The heterogeneities are based on an input tomography and an initial temperature distribution near the surface.*

## A cookbook on subduction initiation

(Cedric Thieulot)

The setup is based on Matsumoto & Tomoda, J. Phys. Earth (1983). It is a very simple setup (only Newtonian materials and free slip boundary conditions). I provide two prm files: One relying on compositional fields and one on particle-in-cell.

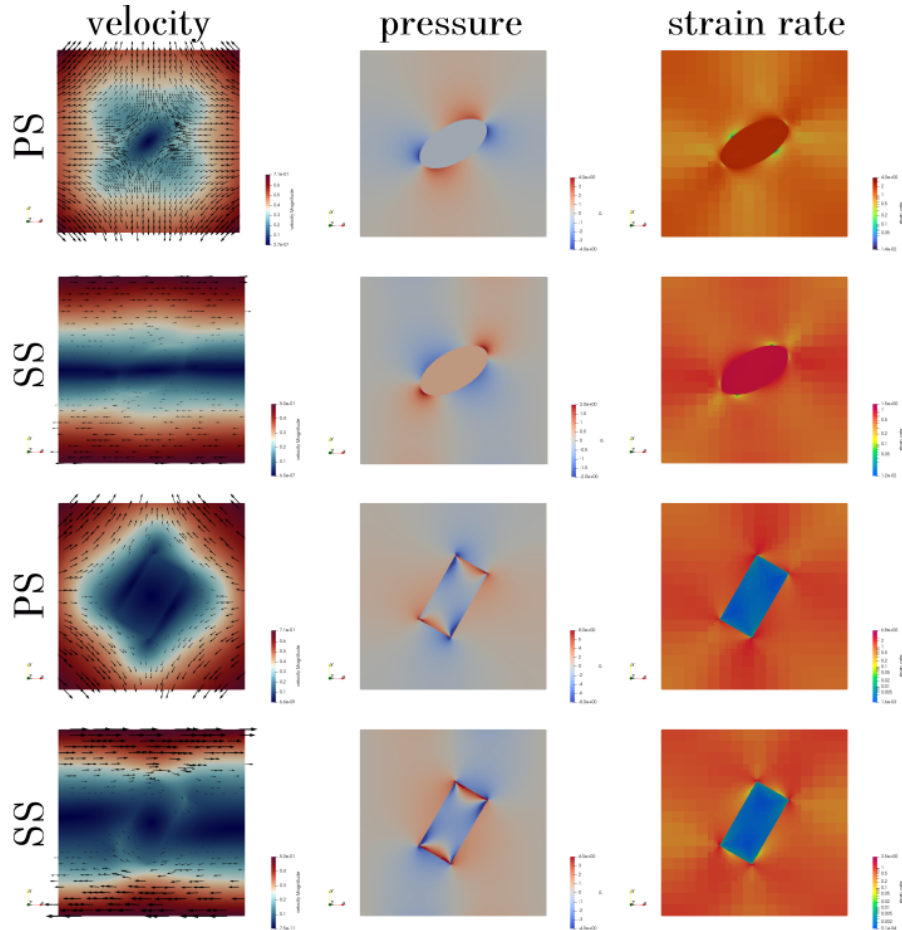


A visualization of the dynamic solution can be found here: [https://youtu.be/W\\_P6dxrjV2E](https://youtu.be/W_P6dxrjV2E).

## A cookbook on deformation of inclusions

(Cedric Thieulot)

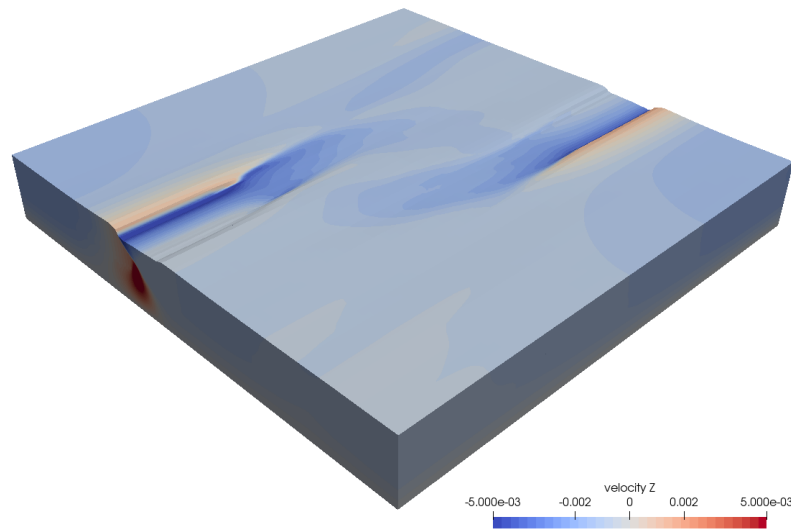
The setup is based on Halter, Macherel, & Schmalholz, Journal of Structural Geology (2022). Elliptical and rectangular inclusions are submitted to either pure shear or simple shear deformation. The matrix is described by either Newtonian or power-law viscosities.



## A cookbook on 3D rift interaction

(Cedric Thieulot)

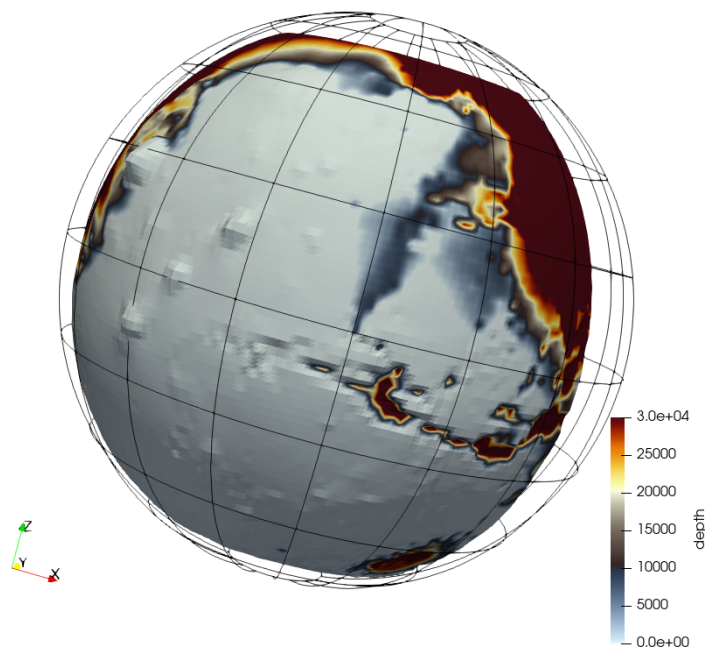
The setup is similar to Allken, Huismans, & Thieulot, G3 (2012). It is a 3D crustal model. The top half is visco-plastic while the bottom half is Newtonian. At the base of the upper crust, two weak zones are prescribed which initiate the formation of two rifts that ultimately interact with each other.



## A (preliminary) cookbook on Mars topography

(Cedric Thieulot, Bart Root)

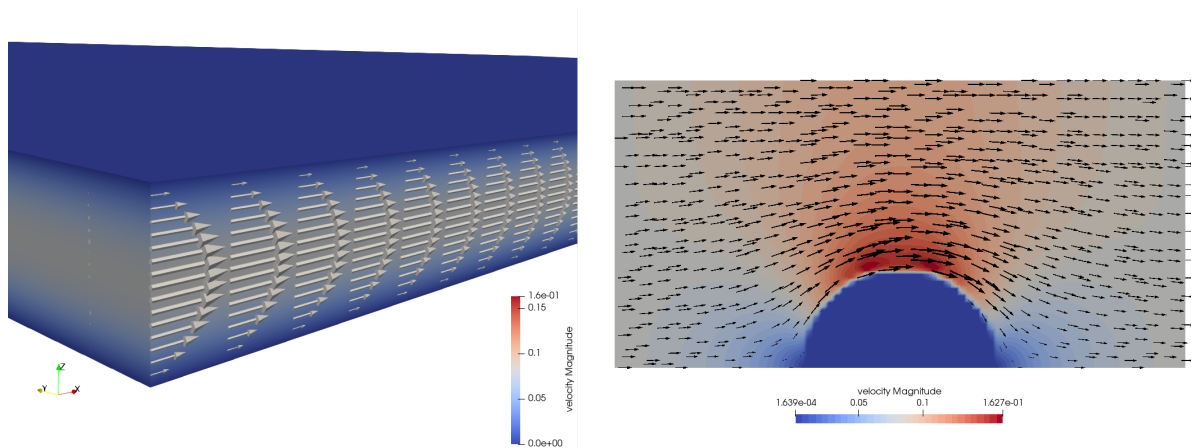
Bart Root (Delft University) provided me with a 1° resolution Mars topography model based on the MOLA data. This cookbook is a starting point for a Mars model. In future updates, we will add a density and viscosity profile.



## A cookbook on 3D lower crustal flow

(Cedric Thieulot)

This setup is based on Clark, Bush, & Royden, GJI (2005). In this conceptual paper the authors argue that a Poiseuille-type lower crustal flow around an obstacle is responsible for the topography observed in Eastern Tibet around the Sichuan Basin.



## A cookbook for 2D corner flow subduction (van Keken 2008)

(Daniel Douglas, Cedric Thieulot, Wolfgang Bangerth)

We tackled a classic model setup for simulating subduction. The unique geometry and constraints used in [van Keken et al., 2008](#) requires a generic quadrilateral mesh that aligns with the interface between the subducting slab, the mantle wedge, and the overriding plate. The principal contribution of this cookbook is the implementation of a coarse mesh that tracks all of these features. Further work would require comparing the results of the ASPECT models to the original study.

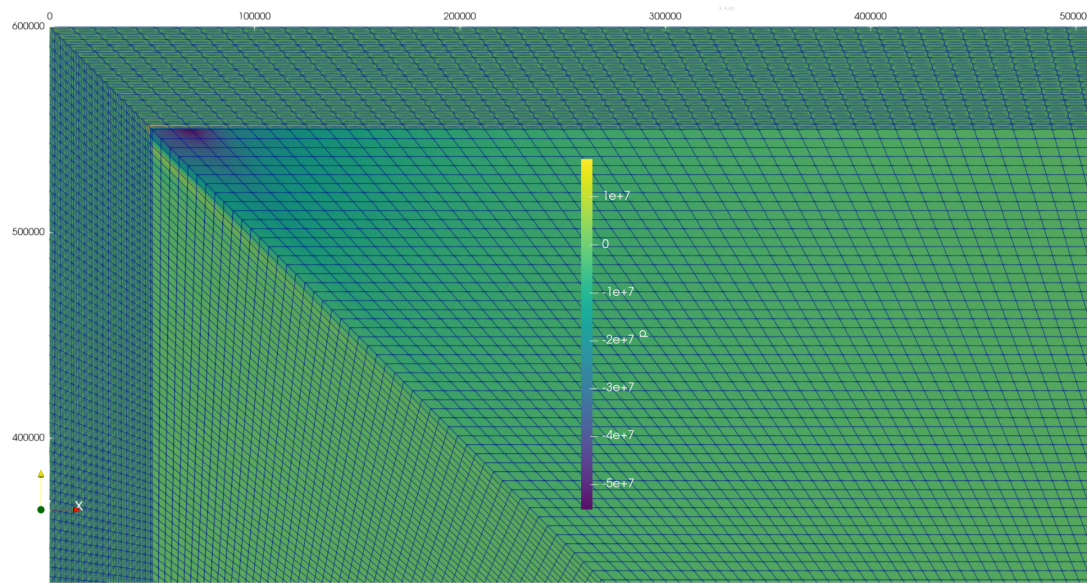


Figure: *The continuous pressure solution with an overlain mesh.*

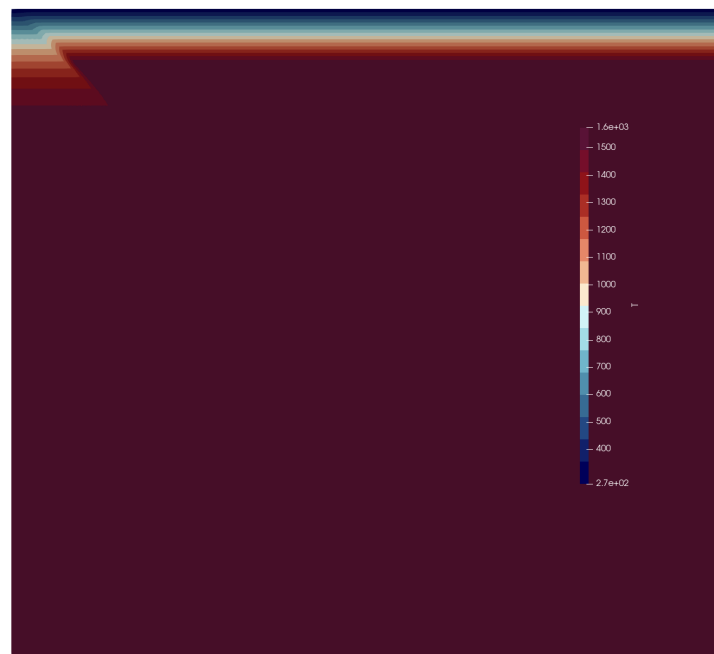


Figure: *The evolution of the thermal regime.*

Use a phase diagram parametrization to simulate reactive fluid transport

(Daniel Douglas)

I modified the reactive fluid transport model to simulate fluid-rock interactions based on a parameterized phase diagram from a study by [Tian et al., 2018](#). This study parameterized four different bulk rock compositions: mid-ocean ridge basalts, gabbro, sediments, and peridotite.



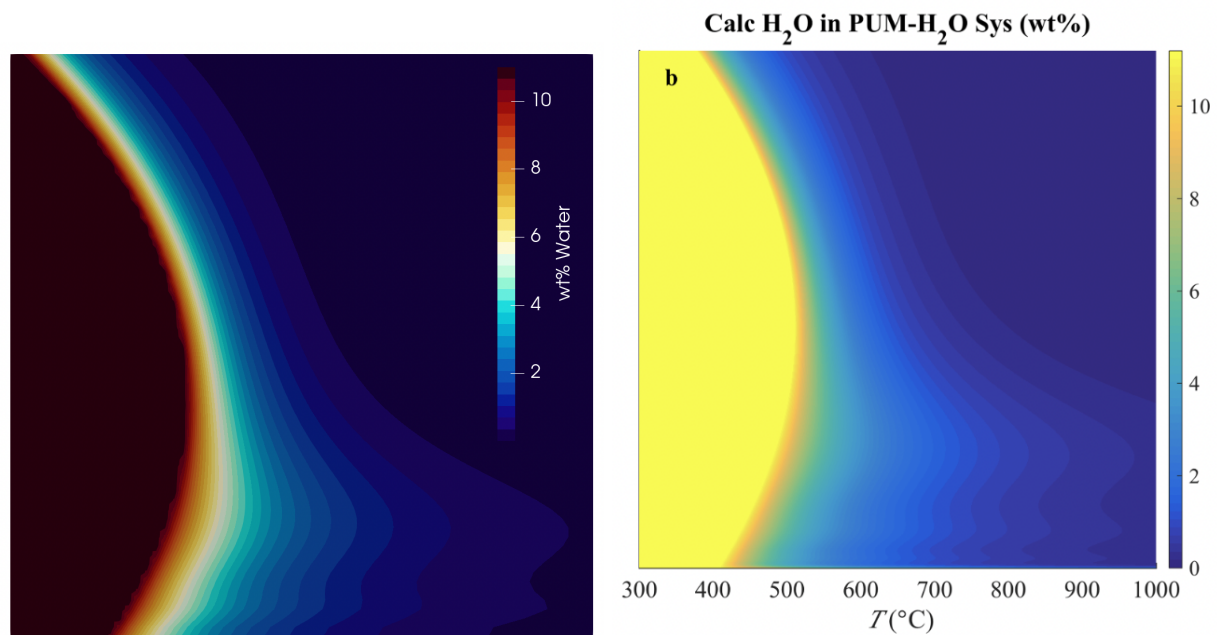


Figure: *Phase diagram in ASPECT (left) compared to the parametrized phase diagram from Tian et al., 2018 for upper mantle peridotite.*

## A simple 2D cookbook for a dehydrating subducting slab

(Daniel Douglas)

A cookbook for the tian parametrization that was described above. The setup looks at the upper 20km of the subducting oceanic crust and lithospheric mantle and prescribes an initial bound water content that gets released based on the P-T conditions of the slab.

## Finalizing VE(V)P implementation updates

(A. Glerum, R. Myhill, J. Dannberg, J. Naliboff, R. Gassmoeller)

We fixed the convergence behavior of several benchmarks for our updated implementation of visco-elasto-viscoplasticity. The dependence of the convergence on the maximum number of nonlinear iterations is now no longer allowed. We tested the stress-strain rate relationship on and off yield. We also adapted the shear heating term to ignore elastic energy, so it does not contribute to heating. The implementation of a plastic damper was checked and switched on. Finally, a check was implemented to make sure that material models support elasticity when it is switched on, and the additional parameter 'Include viscoelasticity' was removed from the visco-plastic material model.



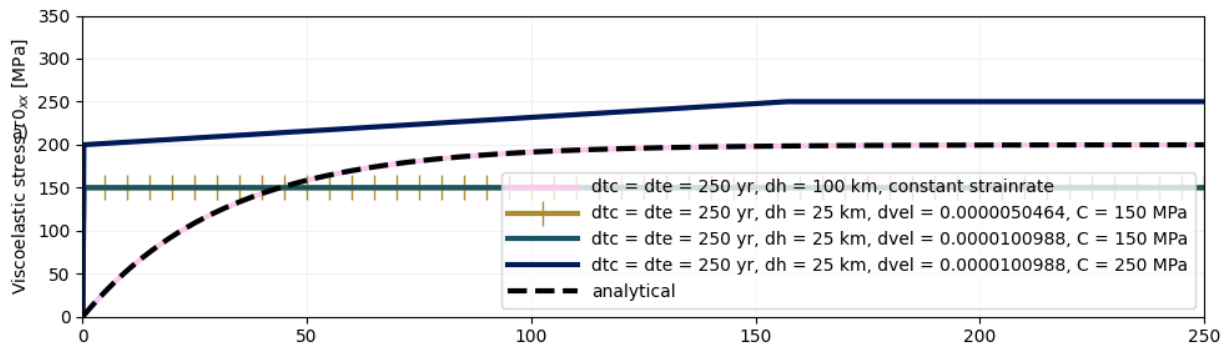


Figure: *xx*-component of the viscoelastic shear stress for increasing strain rates over time and different yield stresses. Rescaling the stress back onto the yield envelope ensures the stress does not exceed the yield stress.

## Include the initial topography in the lithostatic pressure profile for traction boundary conditions

(A. Glerum)

Instead of the maximum initial topography over the domain, the initial lithostatic pressure profile now includes the initial topography at the reference location.

## Add strain rate tensor output on the surface of the domain

(A. Glerum)

There is now also a visualization plugin that outputs the components of the strain rate tensor on the surface of the domain only, instead of on the whole volume of the domain.

## Add lookup table functionality for viscoplastic phases

(Ryan Stoner)

ASPECT uses phase functions to allow for phase transitions based on pressure and temperature. To expand the phase transitions that can be used, I added lookup table functionality to set phase functions based on the dominant phases in a lookup table. A user can now provide a list of phases of interest and associated parameters (density, flow law parameters), and those phases of interest are then used in the viscoplastic material model. The primary advantage of this approach is that rheologies can now be set based on lookup tables.

## An improved microstructural evolution algorithm for the LPO particle plugin

(Srivatsan Vedavyas, Menno Fraters)

We added a new algorithm to the existing LPO particle plugin that can (hopefully) predict the grain size evolution in a more realistic fashion. This algorithm is an add-on to the existing DREX algorithm that can predict the lattice preferred orientation (LPO). The improvements made upon the existing algorithm is that all the values used in the algorithm are dimensional, which means that the grain size predicted now will be dimensional. The grain size evolution is constrained using experimentally derived models of dynamic recrystallization. Pyroxene rheology is also accounted for in this newer implementation. The code has been compiled successfully, but the different parameters need to be benchmarked against experimental studies in order to be fine tuned. It is still a work in progress (requires benchmarking and maybe subsequent modification) but once completed will allow for a better microstructural evolution yielding dimensional grain size distribution that can be coupled with the particle plugin that exists in ASPECT.

## Initial coupling between ASPECT and the FastScape C++ library

(Michael Pons, Anne Glerum, Menno Fraters)

The existing coupling between ASPECT and FastScape, developed by Derek Neuharth and Anne Glerum, utilizes the Fortran version of FastScape. However, this Fortran version is no longer being actively developed, and indeed the coupling uses a version that is slightly modified from the final Fortran library. Our objective was to couple ASPECT with the actively developed C++ version of FastScape, which will incorporate an unstructured grid and be parallelized. This will make ASPECT–FastScape work with large scale spherical models.

We have integrated the Fastscape C++ library to the contrib folder and added the paths to the CMakeLists.txt file. We have developed the initial FastScapeCC (.cc, .h) interface which enables surface velocity to be communicated from ASPECT to Fastscape and vice versa while the topography is being altered by diffusion and Stream Power law erosion.

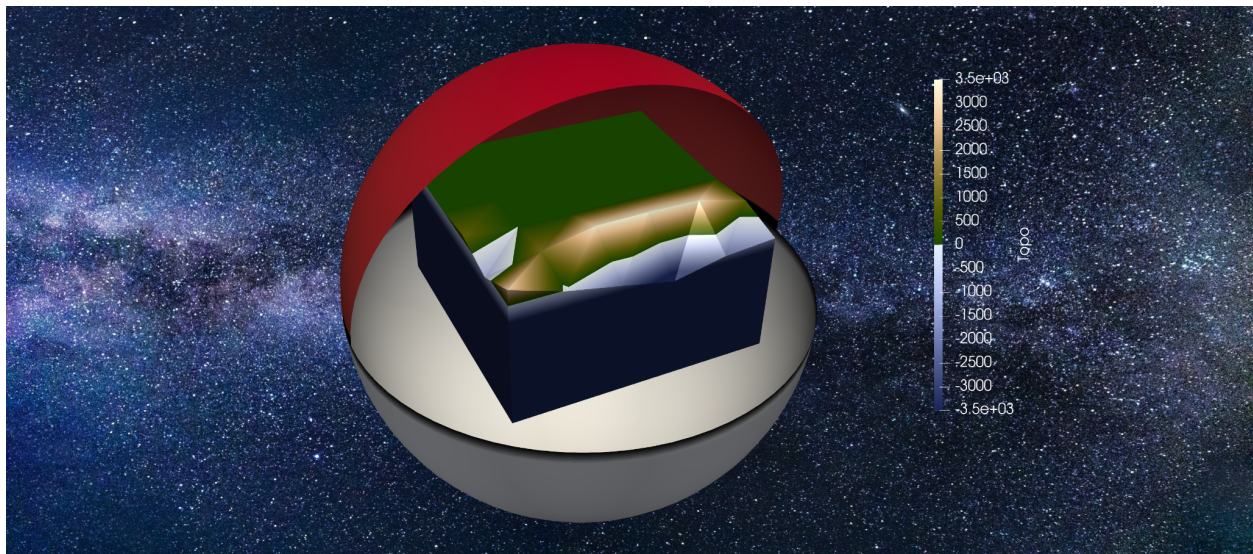


Figure: Low-resolution model illustrating mesh deformation resulting from diffusion and erosion processes. The Pokeball represents the ultimate goal of running FastScape on a spherical earth, while the real model inside illustrates the current state of coupling (reality) after one ASPECT timestep. A video of the evolution of this model can be found at [https://www.youtube.com/watch?v=\\_6RaNKX4Dss](https://www.youtube.com/watch?v=_6RaNKX4Dss).

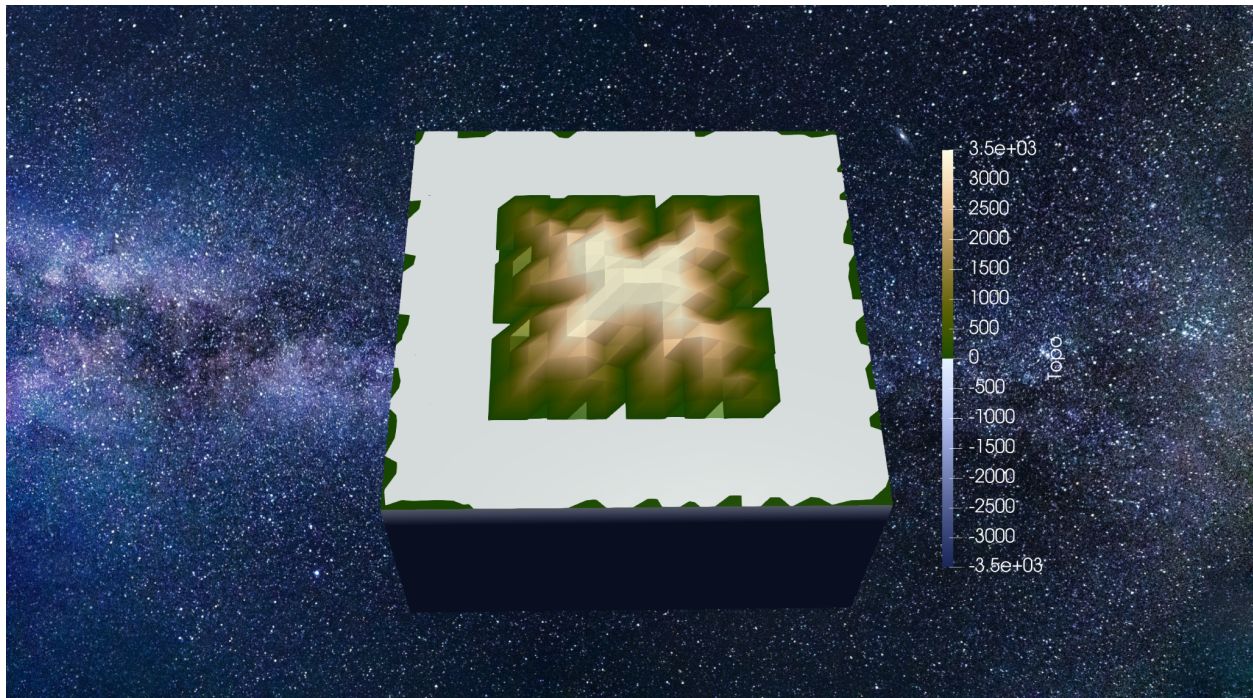


Figure: A higher resolution model after 57kyr illustrating mesh deformation resulting from diffusion and erosion processes.

## Add plasticity, lateral viscosity profile, and P-T dependent conductivity to the Entropy Plugin

(Ranpeng Li & Julianne Dannberg)

The entropy plugin in the benchmark folder now includes a plasticity rheology, which can be set by cohesion and friction coefficient. It can also use a viscosity prefactor profile for calculating the lateral viscosity variation, set the lateral viscosity variation limit to a factor from the reference profile, and make the thermal conductivity p-T dependent. We added a test case “entropy\_plasticity”.

## Make assembly of the advection equations more flexible

(Julianne Dannberg)

I made ASPECT’s advection assemblers more flexible by allowing each advection field (either the temperature or one of the compositional fields) to use a different list of assemblers. In addition, assemblers are now only called for the fields where they are needed instead of all assemblers being called for all fields and then just not assembling any terms for the fields they are not needed for. This is more efficient, less error-prone, and allows us to solve different equations for different fields more easily.

## Solve coupled ODEs with SUNDIALS

(Julianne Dannberg)

ASPECT can be used with SUNDIALS (using the SUNDIALS interface of deal.II), and we can use SUNDIALS to solve coupled ordinary differential equations we have in ASPECT, such as in the operator splitting routine. I adapted the computation of these reaction terms to use SUNDIALS instead of the forward Euler scheme with a constant time step size that we used before.

## Default Field Types implemented

(Bob Myhill, Rene Gassmoeller)

ASPECT now uses the ‘Field Types’ parameter to identify which compositional fields correspond to different properties of the simulation (e.g., chemical composition, grain size, porosity, entropy). This will make complex material models and coupling easier to implement in ASPECT.

## Add CPO main post-processor

(Menno Fraters)

The Crystal-Preferred Orientation (CPO) now has a postprocessor which can output the particle information.

## Transform and rotate functions for full stiffness tensor, Voigt matrix and vector-tensor utilities

(Menno Fraters)

There are now tensor utility functions which can convert between the full stiffness tensor, the Voigt matrix, and the Voigt tensor. There are also functions which can be used to rotate these.

## Levi-Cevita tensor for the tensors utilities

(Menno Fraters)

The Levi-Cevita tensor is now available in the Utilities namespace.

## Add astyle installation instructions to the manual

(Menno Fraters)

Manual installation instructions for astyle were not available in the manual, but only on the wiki. Now they are also in the manual and CONTRIBUTING.md. By adding this information to CONTRIBUTING.md, and directly including this file into the contributing page of the manual, the information is provided only once, but available in all relevant places.

## Add a CPO postprocessor which can compute the elastic tensor (work in progress)

(Menno Fraters)

The Crystal-Preferred Orientation (CPO) can be used to compute an effective elastic tensor. I added (soon hopefully) a plugin which can compute this elastic tensor. This elastic tensor can then be used for further postprocessing or in an external program to compute the seismic anisotropy of a model.

## A material for simulating reactive fluid transport

(John Naliboff, Juliane Dannberg, Ryan Stoner, Daniel Douglas)

We added a new material model for simulating reactive fluid transport that is composited on top of material models simulating solid deformation. The material model is designed to be expanded to include different models (user selected) for fluid-rock interaction.

## Fixed a bug in the strain rheology module

(John Naliboff, Anne Glerum, Rene Gassmoeller)

We fixed a bug that led to small errors in the strain weakening and healing reaction updates.

## Functionality for averaging material properties between nonlinear iterations

(John Naliboff, Anne Glerum, Bob Myhill, Julianne Dannberg)

We developed functionality within the material model class for averaging the material properties between nonlinear iterations. This functionality may enable improved nonlinear solver behavior. Testing of this functionality is still in progress.

## Allowing multiple traction models for a boundary

(Chameera Silva, Rene Gassmoeller)

We updated the aspect boundary traction, where we can now apply a combination of multiple traction models for any particular boundary. Boundary traction models are now organized in a manager class that allows assigning multiple boundary traction plugins to each boundary. Existing user plugins that use the function `this->get_boundary_traction()` will need to be modified to use `this->get_boundary_traction_manager()`. A test case was added to prove the new functionality works as expected.

## An anisotropic viscosity material model (work in progress)

(Yijun Wang, Ágnes Király)

I am working on adding an anisotropic viscosity material model. This material model will take 36 coefficients and 6 intercepts as inputs from the parameter file to calculate the Hill parameters (F, G, H, L, M, N) and compute the rank-4 viscosity tensor following Signorelli et al. (2021). This viscosity tensor is stored as the stress strain director to be used in Stokes equations and shear heating equations and the material model viscosity is updated with the effective viscosity. The stress strain director from the previous time step is saved in prescribed fields to be used in computing the anisotropic stress for the current time step. The anisotropy of the model can be visualized with anisotropic stress postprocessor in ParaView using TensorGlyph.



## General code updates

(Wolfgang Bangerth)

As with any code base, there are always places that could be improved or, equally importantly, that could be changed to replace whatever approach is used locally by a commonly used idiom used in many other places of the code base. Similarly, with evolving C++ standards, some code can now be expressed more clearly or concisely by using more modern C++. Many patches went into the code base during the hackathon to clean up these corners and to keep ASPECT on a sustainable path towards the future.

## A compositional viscosity in the Steinberger material model

(Poulami Roy, Juliane Dannberg)

We updated the Steinberger material model with viscosity prefactors. The viscosity prefactor is a dimensionless quantity and is multiplied to the original Steinberger radial and lateral viscosity variation based on a volume fraction. This gives different relative viscosities of different compositional fields.

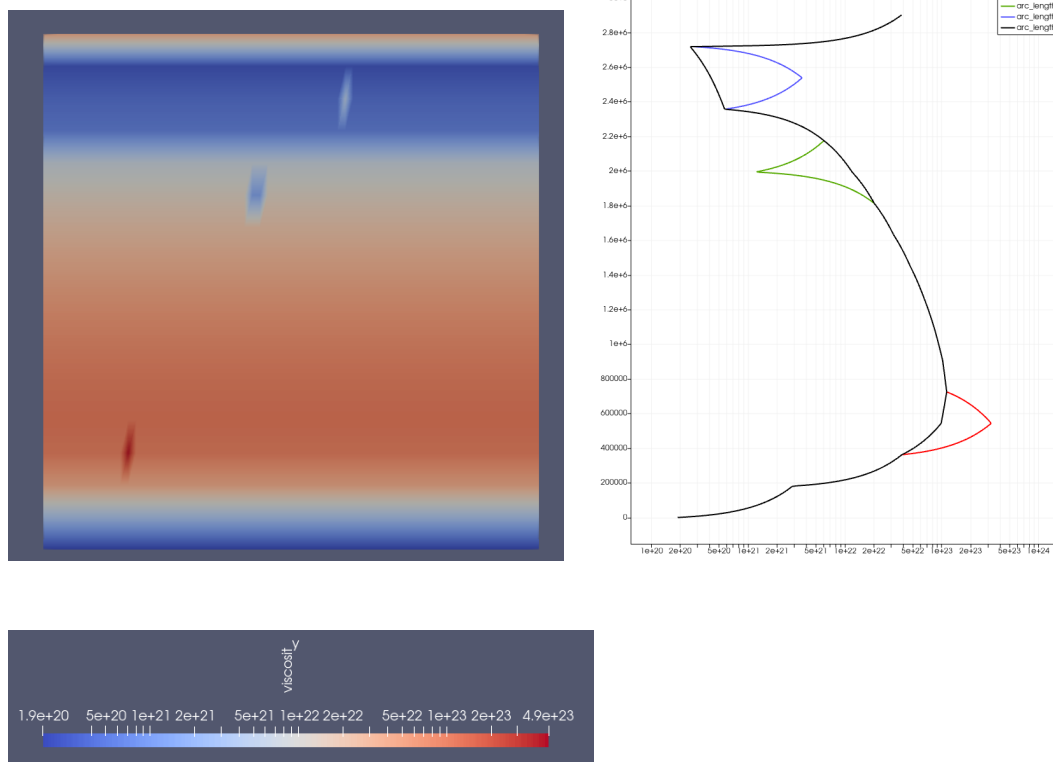


Figure: *Three different compositional fields with different viscosities. The reference viscosity profile is based on the original Steinberger and Calderwood [\[2006\]](#) formulation, shown by the*

black curve. Three other viscosity profiles (blue, green and red curves) are obtained by multiplying the viscosity prefactors to the reference viscosity profile.

## Education survey

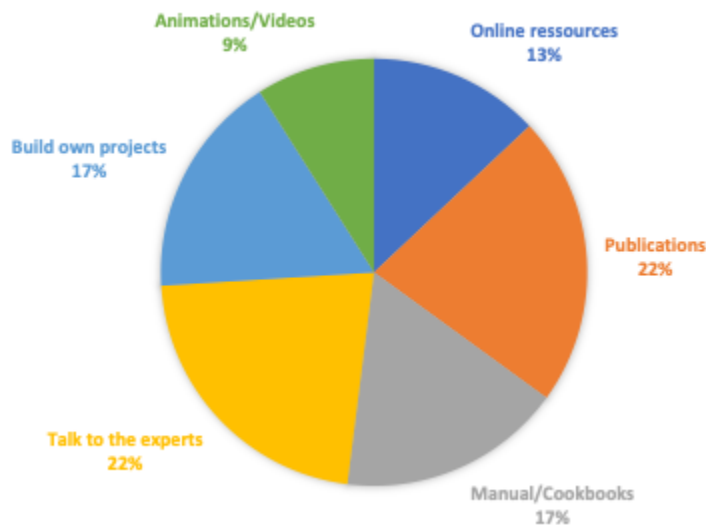
(Mohamed Guiza)

### Survey about learning computational tools

**Question:** When you started using/teaching a new computational tool (e.g., a code like ASPECT or a programming language), what was your learning/teaching approach? What were the issues you/the learners faced? How did you solve those issues?

#### Learning methods:

- Online resources (3/23=13%)
- Publications – published case studies/projects (5/23=22%)
- Manual and cookbooks (4/23=17%)
- Talk to the experts (i.e., emails, forums, knock on doors) (5/23=22%)
- Build your own case studies/project from scratch (4/23=17%)
- Animations/Videos/illustrations (2/23=9%)



#### Issues and Challenges:

- Lack of sufficient and relevant examples
- Supervisor not proficient in the tool
- Using while learning a tool can be challenging
- Terminology
- No prior experience
- Inappropriate hardware

#### Solutions:

- Collaboration with experts
- Tailored projects



- More online resources and examples

**Interesting views:**

- You do not need to be proficient in the entire tool (e.g., C++), but focus on components useful for your usage.

# Statistics about ASPECT's growth during the hackathon

The following contains a number of statistics about how much ASPECT has grown during the hackathon (July 5-22, 2023 to account for late merges):

- Number of source files in ASPECT before/after: 651 -> 661 + 10
- Lines of code in ASPECT before/after: 168,730 -> 172,659 + 3,929
- Number of merged pull requests before/after: 3,803 -> 3,925 + 122
- Commits in github before/after: 10,522 -> 10,722 + 200
- Number of tests before/after: 1,030 -> 1,049 + 19
- Lines of documentation in ASPECT before/after: 19,639 -> 19,686 + 47

For comparison, these were the statistics for last year's (2022) hackathon:

- Number of source files in ASPECT before/after: 629 -> 637 + 8
- Lines of code in ASPECT before/after: 158,537 -> 162,241 + 3,704
- Number of merged pull requests before/after: 3,330 -> 3,532 + 202
- Commits in github before/after: 9,468 -> 9,908 + 440
- Number of tests before/after: 963 -> 986 + 23
- Lines of documentation in ASPECT before/after: N/A -> 16,753 +16,753

Comparing the number of commits and pull requests suggests that the 2023 hackathon was less productive than the one in the previous year. However, one has to take into account that the 2022 hackathon contained the conversion of all of ASPECT's documentation into a markdown format, which necessitated many small fixes and relatively simple updates, scattered over a large number of commits and pull requests. The 2023 hackathon instead saw the completion of some larger projects (visible in the large numbers of lines of code merged) and the addition of many new cookbooks.

The difference between the second number in the second table (at the end of the 2022 hackathon) and the first number in each column of the first table above it (at the start of the 2023 hackathon) illustrates the level of development that happened over the course of the year between the hackathons. As in previous years, somewhere between one quarter and one half of the ASPECT development happens during hackathon weeks.

For completeness, the statistics above were generated with the following commands:

- `find include/ source/ | egrep '\.(h|cc)$' | wc -l`
- `cat `find include/ source/ | egrep '\.(h|cc)$' | wc -l`
- `git log --format=oneline | grep "Merge pull request" | wc -l`
- `git log --format=oneline | grep -v "Merge pull request" | wc -l`
- `ls -l tests/*prm | wc -l`
- `cat `find doc/sphinx | egrep '\.(md)$' | wc -l`