

# ASPECT Hackathon 2022

<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>8</b>
<b>Things to do before the start of the hackathon</b>	<b>8</b>
Installing deal.II on Mac	8
Installing deal.II on Linux	8
Installing deal.II on Windows	8
Visual Studio Code Tutorial	9
Git Tutorial	9
<b>Report on projects the participants worked on</b>	<b>11</b>
Documentation has been converted to markdown/html	11
Fix indentation of post-C++11 code	12
Fix computation of <code>e_dot_ij</code>	12
Co-Author Graph	13
Anisotropic Viscosity using Euler angles on compositional fields	14
Anisotropic viscosity using CPO data on particles	15
Anisotropic viscosity using dislocation strain rate	16
Output physical units along with solutions	16
Accelerate the the gravity point values postprocessor	16
Describe documentation generation in cmake	17
Add lower mantle melting/melt transport material model	17
Add support for compressed input data	17
Add support for sharing data between processes	18
Add new advection darcy field method	18
Add test for fixing temperature in a compositional field	18
Add benchmark for lithospheric flexure with infill material	18
Add subduction cookbook for fixing temperature inside slab composition	19
Add strain rate particle property	19
Code cleanup: Remove the reference viscosity	20
Models with an equilibrium grain size	20
Quadratic least squares particle interpolation limiter	20
Adding a tutorial on nonlinear Stokes solvers to the manual	21
Adding CPO core plugins to ASPECT	21

Discussed World Builder algorithms	21
Add Kaus (2010) and Gerya (2019) documentation to existing .prm files	22
Add Allken et al. (2012) cookbook	23
Adding Duretz et al. (2020) elastoviscoplastic benchmark	24
Enable GMG+free surface+elasticity	25
Boundary traction in spherical geometry/coordinates	25
Testing open boundaries in spherical geometries with free surface	25
Loading netCDF data into StructuredData	26
<b>Statistics about ASPECT's growth during the hackathon</b>	<b>27</b>

# Meeting places

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

- **Main Zoom Meeting room (open 24/7)**  
<https://ucdavis.zoom.us/j/98014710513?pwd=YVYwcEVLSFMzR1hya244aDIFTVVYUT09>
  - Meeting ID: 980 1471 0513
  - Passcode:725310(This is Lorraine's account. Let her know if something is not working.)
- **Slack Workspace for asynchronous communication (cig-aspect.slack.com):**  
[https://join.slack.com/t/cig-aspect/shared\\_invite/zt-gdn947lt-xRu8SpWkzI2rgnfOLCOpYw](https://join.slack.com/t/cig-aspect/shared_invite/zt-gdn947lt-xRu8SpWkzI2rgnfOLCOpYw)
- **Hackathon log (this document):**  
<https://docs.google.com/document/d/1z07jsXcuAul8HuAcOt-HHR7K28zvuqGuNLtValxDSpM/edit?usp=sharing>  
<https://bit.ly/2022-hack-log>
- **Facebook:**  
<https://www.facebook.com/groups/1438286143140671> (Cedric & Lorraine admin)
- **Spotify playlist:** (Lorraine owner)  
<https://open.spotify.com/playlist/6hgoGdSfne9VGLZWCc7y7e?si=95e13ea0f6a74e71>
- Google drive for photos  
<https://drive.google.com/drive/folders/1ecr2TkrBe8Y1a5kVKI3QF2Udew5dLw34>

# Land acknowledgement

We acknowledge the indigneous people and land in which we are gathered. Cody, Wyoming has been home to the Eastern Shoshone, Cheyenne, Apsaalooké (Crow), and Očhéthi Šakówiŋ 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, 17 users and developers of ASPECT worked in-person and virtually over two weeks in May 2022.

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

## Timeline

<b>Day</b>	<b>Scheduled items (in Mountain time; US East: +2h; Central Europe: +8h)</b>
Sunday, 5/15	6:30p Welcome, Introductions, Code of Conduct (Lorraine) 7:00p Dinner
Monday, 05/16	9 am: Project discussion, Planning meeting
Tuesday, 05/17	9 am: Daily rounds 11.30 am: Chris Mills Sphinx documentation conversion process
Wednesday, 05/18	- DAY OFF
Thursday, 05/19	9 am: Daily rounds
Friday, 05/20	9 am: Daily rounds
Saturday, 05/21	9 am: Daily rounds
Sunday, 05/22	9 am: Daily rounds
Monday, 07/23	9 am: Daily rounds 8 pm: EGU practice presentations (Arushi, Yijun)
Tuesday, 07/24	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. Migrate documentation</li> <li>4. Unified plugin/manager base class?</li> </ol>
Lorraine Hwang UC Davis ljhwang@ucdavis.edu		<ol style="list-style-type: none"> <li>1. Logistics</li> <li>2. Writing</li> <li>3. ASPECT Networks</li> <li>4. Documentation</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> <li>4. Output for surface quantities</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. Lower mantle melting model</li> <li>4. Make example for running a model that uses the equilibrium grain size for the rheology</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@tutanta.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> <li>7. Slab2 integration</li> <li>8. Discuss addition of melt and GMG to Newton solver</li> </ol>

John Naliboff New Mexico Tech john.naliboff@nmt.edu	US., Mountain Standard Time	<ol style="list-style-type: none"> <li>1. Help others achieve their goals</li> <li>2. Review pull requests</li> <li>3. Add a section to the manual about lithospheric, rheology and material model structure</li> <li>4. Help finish test and implement VEP pull request</li> </ol>
Anne Glerum	CEST	<ol style="list-style-type: none"> <li>1. VEP implementation PR (Bob, John, Esther)</li> <li>2. FastScape PR</li> <li>3. MaterialModelInputs properties PR</li> <li>4. Lithospheric melt model (Bob)</li> <li>5. Help and review pull requests</li> </ol>
Arushi Saxena University of Florida saxena.arushi@ufl.edu		<ol style="list-style-type: none"> <li>1. Test instantaneous 3D mantle models to best-match the surface observables, and create additional plugins required for it. Also test the World builder performance in these models.</li> <li>2. Work on the open issues in the ASPECT github repository.</li> </ol>
Maaïke Weerdesteijn UiO Oslo m.f.m.weerdesteijn@geo.uio.no	CEST, but I can be flexible and work late European hours	<ol style="list-style-type: none"> <li>1. Viscoelasticity / GIA <ol style="list-style-type: none"> <li>a. Surface loads on a 3D spherical shell</li> <li>b. Sea level equation implementation</li> <li>c. Include load gravitational attraction</li> <li>d. Include time-varying ocean basins</li> </ol> </li> </ol>
Cedric Thieulot c.thieulot@uu.nl	CEST but I can be flexible and work late	<p>Add (at least) 2 EVP cookbooks</p> <p>Add extra parameters to allow T-depdt ductile weakening</p> <p>Herschel-Bulkley model?</p>
Agi Kiraly agnes.kiraly@geo.uio.no		<ol style="list-style-type: none"> <li>1. Add anisotropic viscosity to LPO/CPO development of olivine (Menno) -&gt; The anisotropic stress is calculated on the particles and stored as particle properties (done) Need material model that uses these particle properties for calculating AV when deformation mechanism is dislocation creep.</li> </ol>
Daniel Douglas New Mexico Tech daniel.douglas@student.nmt.edu	US, Mountain Standard Time	<ol style="list-style-type: none"> <li>1. Help with git</li> <li>2. Create pull request for plate cooling model temperature plugin</li> <li>3. Get feedback on time dependent heating plugin I wrote/create pull request</li> <li>4. Create pull request for traction plugin/cookbook</li> </ol>
Paul Bremner MSFC	US, Central Standard	<ol style="list-style-type: none"> <li>1. Model calibration</li> <li>2. Static-lid dynamics</li> <li>3. Surface processes</li> </ol>

paul.m.bremner@nas a.gov	Time	<ol style="list-style-type: none"> <li>4. Gravity</li> <li>5. Rheology</li> </ol>
Chris Mills		<ol style="list-style-type: none"> <li>1. Convert manual to sphinx documentation</li> <li>2. Learn!</li> </ol>
Mack Gregory		<ol style="list-style-type: none"> <li>1. Implement limiter algorithm for linear least squares and quadratic least squares particle interpolators</li> </ol>
Yijun Wang		<ol style="list-style-type: none"> <li>1. Make sure the anisotropic stress particle property is correct</li> <li>2. Work on anisotropic viscosity material model</li> <li>3. Test if the Enstatite part of anisotropic stress calculation works</li> </ol>
Jiaqi Zhang jiaqi2@clemson.edu	US, Eastern Standard Time	<ol style="list-style-type: none"> <li>1. GMG Stokes solver</li> <li>2. Free-surface boundaries</li> </ol>

# Resources

## Things to do before the start of the hackathon

- Create an account on <https://github.com> if you don't have one yet.
- Make sure you have Zoom installed and can use it in a reasonable environment to participate using audio/video in our discussions.
- 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 9.3.2 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-9.3.2-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/index.html>
    - c. <https://swcarpentry.github.io/git-novice/02-setup/index.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 forked repositories (you should own your\_username/aspect on github)
    - c. Ensure proper remotes (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 master'`
      - ii. `'git pull upstream master'`
      - iii. `'git checkout -b remove_unused_option'`
    - b. Make changes for ASPECT\_USE\_PETSC in one of the files (remove the option for ASPECT\_USE\_PETSC == true):
      - i.
    - c. Create commit
      - i. `'git add FILE'`
      - ii. `'git commit -m 'Removed a deprecated option'`

- d. Push and open PR
    - i. 'git push origin remove\_petsc\_option'
    - ii. Open PR on github (CTRL-Click on shown link)
  - e. Wait for review
  - f. Address review (repeat steps b,c,d)
  - g. Success!
  - h. Alternative to fix: Look for DEAL\_II\_VERSION\_GTE(9,2,0)
5. Now repeat the steps in 3. on your own. Pick a section of the manual in doc/manual/manual.tex that interests you. Find a sentence or description or formula to improve. Then repeat 3. and make your changes to the file doc/manual/manual.tex.

# Report on projects the participants worked on

## Documentation has been converted to markdown/html

(Everyone)

The ASPECT manual was written in latex since the beginning of the project. We have desired to convert this to an HTML-based format for a while, and this was largely achieved at the hackathon: The manual has been converted to a markdown format which can be hosted online at ReadTheDocs.io by way of [Sphinx](#) documentation generator. Before the hackathon, we converted the latex version of the manual (including the cookbooks) to markdown using [Pandoc](#), and then split the file into different pages and directories corresponding to manual sections and chapters. These Markdown files were then converted into [MyST](#) over the course of the hackathon. We also recreated the automatically generated parameter documentation section of the tex manual using the parameters.json produced by running ASPECT, and then using a [Python script](#) to create Markdown files. Many dozens of patches were merged during the hackathon to address issues with the conversion from latex to markdown, such as poorly formatted equations, latex macros that are not supported by MyST, and many other problems with the initial conversion.

As of the end of the hackathon, the documentation is live at [aspect-documentation.readthedocs.io](https://aspect-documentation.readthedocs.io). Some of the cookbooks and benchmarks pages as well as some parameter descriptions have not yet been fully converted to MyST and suffer from formatting issues as a result, but the bulk of the manual, including the parameter documentation, is complete.

ASPECT 2.4.0-pre

Search the docs ...

**ASPECT**

- User Guide
- Introduction
- Geodynamic modeling assumptions and numerical methods in ASPECT
- Installation
- Running ASPECT
- Cookbooks
- Benchmarks
- Extending and contributing to ASPECT
- Finding answers to more questions
- Parameter Documentation
  - Global parameters
  - Adiabatic conditions model
  - Rayleigh convection model

ReadTheDocs v: latest

### About ASPECT

ASPECT is a code to simulate problems in thermal convection. Its primary focus is on the simulation of processes in the Earth's mantle, but its design is more general than that. The primary aims developing ASPECT are:

- Usability and extensibility:** Simulating mantle convection is a difficult problem characterized not only by complicated and nonlinear material models but, more generally, by a lack of understanding which parts of a much more complicated model are really necessary to simulate the defining features of the problem. This uncertainty requires a code that is easy to extend by users to support the community in determining what the essential features of convection in the Earth's mantle are.
- Modern numerical methods:** We build ASPECT on numerical methods that are at the forefront of research in all areas – adaptive mesh refinement, linear and nonlinear solvers, stabilization of transport-dominated processes. This implies complexity in our algorithms, but also guarantees highly accurate solutions while remaining efficient in the number of unknowns and with CPU and

## Fix indentation of post-C++11 code

(Wolfgang Bangerth)

We have a bunch of code of the kind

```
std::vector<std::pair<int,double> > my_vector;
```

In the pre-C++11 era, we needed to have a space between the closing template brackets, i.e., “> >”. This was necessary because otherwise the C++ parser would read the two separate closing brackets as the single “>>” token (the right shift operator). C++11 changed this by stating that one need not have the space any more, and indeed we have many many places in the code base where we don’t need the space any more. The downside is that the ‘astyle’ indentation program – in the version we use – does not know C++11 or later, and consequently gets very confused. It then produces nonsensical indentation because it thinks that the two closing parentheses are in fact a right-shift operator.

I’ve implemented a patch that works around this by changing all occurrences of “>>” to “> >”, letting ‘astyle’ run on this, and then change things back. The result is that we can drop the space (and the patch does this in all places where it is present) and still get reasonable indentation.

(Reference: <https://github.com/geodynamics/aspect/pull/4575> )

## Fix computation of e\_dot\_ii

(Wolfgang Bangerth)

Our code base had a number of places where we use

```
sqrt( abs( second_invariant( deviator( eps ) ) ) ) )
```

to compute what geodynamicists call the “second invariant of strain” (which is separate from what mathematicians call the “second invariant” and which is implemented by the `second_invariant()` function). But this is wrong, it should have been

```
sqrt( -second_invariant( deviator( eps ) ) ) )
```

The code used the `abs()` function to do the negation and also deal with situations where round-off yields a small but negative result. This is now replaced by

```
sqrt( max( -second_invariant( deviator( eps ) ), 0.0 ) )
```

This has the right sign and deals with the issue of small negative values by simply truncating them.

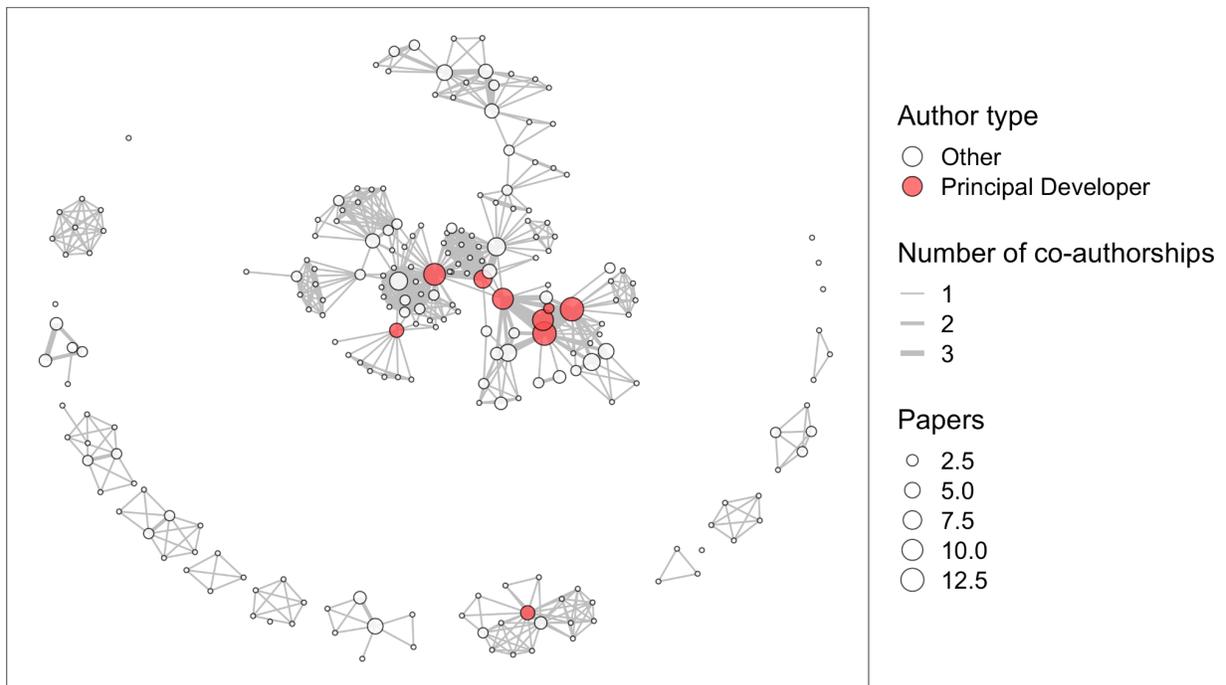
(Reference: <https://github.com/geodynamics/aspect/pull/4557> )

# Co-Author Graph

(Lorraine Hwang)

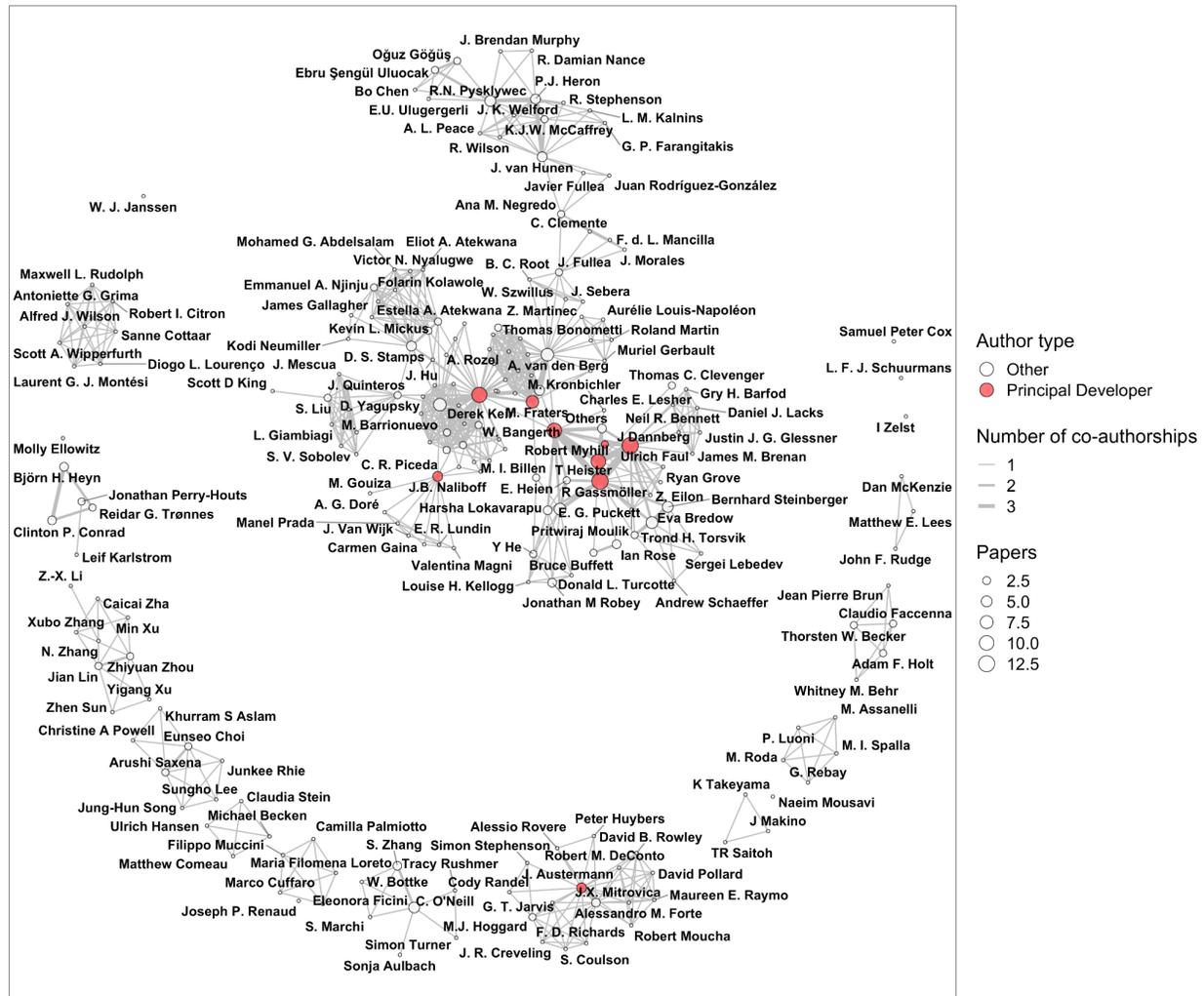
Below is the updated, 2022 edition showing the relationship between authors of papers that cite ASPECT:

## ASPECT Co-author Relationships



Here's a messier version for those wanting to dig into the details:

## ASPECT Co-author Relationships



## Anisotropic Viscosity using Euler angles on compositional fields

(Ági Király)

The first step of implementing Olivine LPO (or CPO) related anisotropic viscosity was achieved by describing olivine texture by three compositional fields related to the three Euler angles of the orientation of the olivine texture. This method assumes one olivine grain per cell, and does not deal with the evolution of the grain's orientation. Anisotropic viscosity is calculated based on the micromechanical model described by Hansen et al. (2016), and calculates the viscosity tensor similarly to the methods described by Kiraly et al. (2020). I also added a postprocessor for saving the anisotropic stress tensor, which is the actual stress the model needs for achieving the given strain rate.

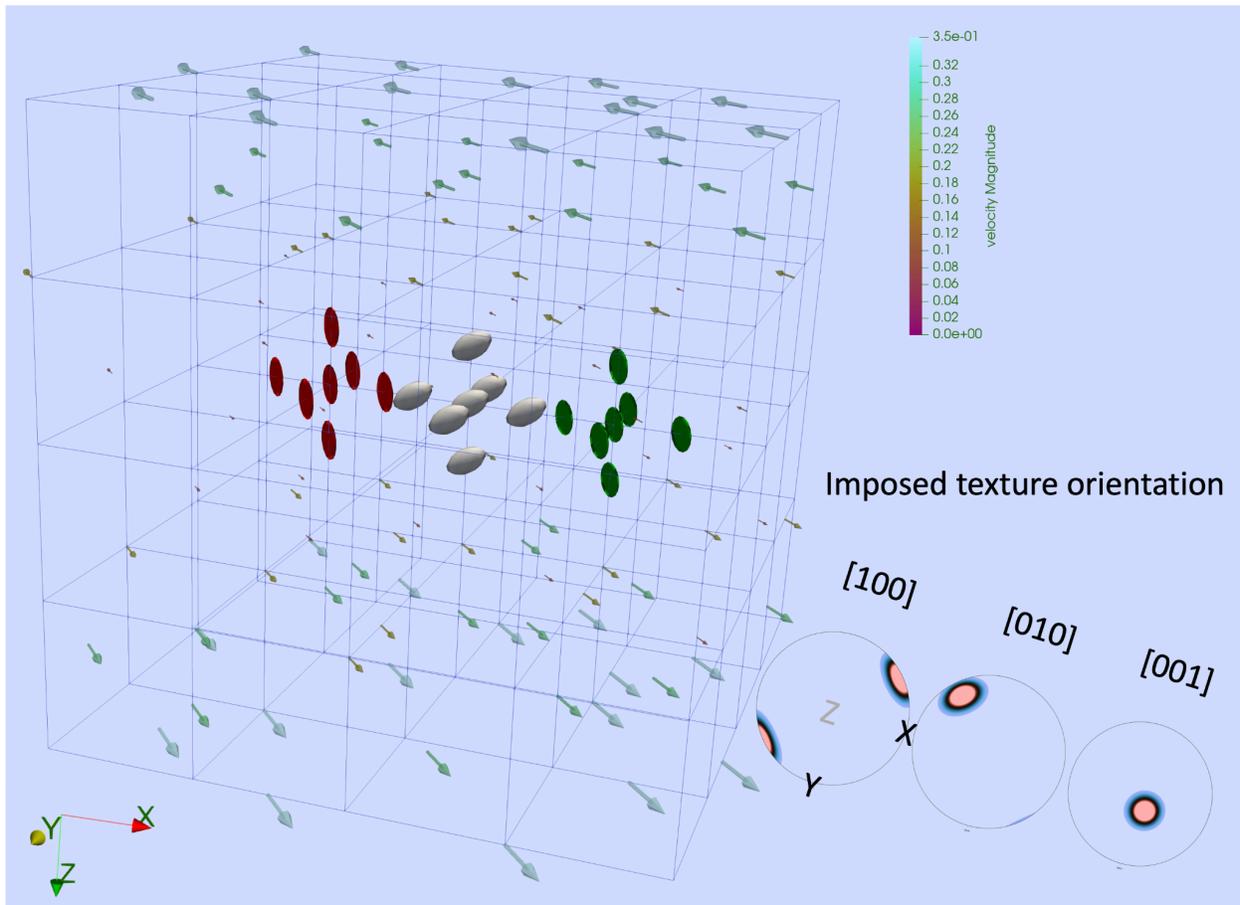


Figure: Results with using anisotropic viscosity in a shear box. Arrows show the input velocity field. Red discs represent the strain rate tensor, green disks the isotropic stress tensor, while the white ones the anisotropic stress tensor. The orientation of the olivine grains (shown in the right bottom corner) can be described by (10, 20, 30) Euler angles. In this case the anisotropic stress has a significant amount of normal stresses as well as shear stresses primarily on the xz component, but also on the xy component (while the strain rate and the isotropic stress have only the xz and yz components that are non 0).

## Anisotropic viscosity using CPO data on particles

(Ági Király)

The next step in the implementation of Olivine CPO-related anisotropic viscosity is to actually use the CPO data (based on Fraters and Billen, 2021). To do so, we implemented a particle property that calculates the anisotropic stress from the current strain rate and the current texture (CPO) on each particle. Using the equivalent strain rate as amplitude for 5 independent strain rates we calculate five more stresses that are required for building the full rank-4 viscosity tensor. We store these 6 stresses (36 components) on the particles, and interpolate them onto compositional fields. In the material model, we load the compositional fields and use them for the [calculation of the viscosity tensor](#). I am currently running models using this material model in a shear box and compare them to our model in MATLAB (Király et al., 2020).

Next steps: test it in different setups; implement anisotropic viscosity with diffusion-dislocation/visco-plastic rheologies, where the dislocation creep part acts anisotropically.

## Anisotropic viscosity using dislocation strain rate

(Yijun Wang)

Continuing with what Agi Kiraly has done above, we worked on computing the dislocation strain rate in the anisotropic viscosity material model, and use this dislocation strain rate for anisotropic viscosity calculations in this material model, and in the particle property where we compute the anisotropic stress tensor required to achieve the current strain rate. Now that the dislocation creep part is working, we need some tests to see that it is behaving as we expected.

## Output physical units along with solutions

(Wolfgang Bangerth, Andrew Hollyday)

As part of taking care of leftovers from last year's hackathon, this is a patch that collects from all visualization postprocessors what physical units the quantity has that is being output. This information is then written as meta-data into the VTU files along with the actual data to be visualized.

Currently, no other output format that deal.II can write supports this kind of information, but having the information collected and at hand would make it possible to also output physical unit descriptors for other file formats.

(Reference: <https://github.com/geodynamics/aspect/pull/4637> )

## Accelerate the the gravity point values postprocessor

(Wolfgang Bangerth)

The gravity point values postprocessor computes information from the solution field, but also does some analytical computations that are then written into the output file. These analytical computations happen only on process zero, creating an inefficiency and bottleneck in parallel computations. This has been addressed now by letting all processors participate in these computations.

(Reference: <https://github.com/geodynamics/aspect/issues/4485> )

## Describe documentation generation in cmake

(Wolfgang Bangerth)

Generating the documentation – both the PDF version of the manual as well as running doxygen – has always lived outside the usual build system written in cmake. Rather, it used a hand-written Makefile. At least parts of this have now been converted into proper cmake scripts. Among the advantages is that temporary files and intermediate inputs are no longer placed in the source directory, but in the build directory.

(Reference: <https://github.com/geodynamics/aspect/pull/4656> and <https://github.com/geodynamics/aspect/pull/4794> )

## Add lower mantle melting/melt transport material model

(Juliane Dannberg, Rene Gassmüller, Bob Myhill)

This pull request adds the material model used in Dannberg, J., Myhill, R., Gassmüller, R., & Cottaar, S. (2021): “The morphology, evolution and seismic visibility of partial melt at the core–mantle boundary: implications for ULVZs”, *Geophysical Journal International*, 227(2), 1028-1059. The model implements a simplified version of the melting model of Boukare et al. (<https://doi.org/10.1002/2015JB011929>) for the lowermost mantle. The model parameterizes the composition (which includes the components MgO, FeO and SiO<sub>2</sub>) as a mixture between two endmember compositions (one iron-bearing and one magnesium-bearing). The equation of state considers three phases: bridgmanite, ferropericlase, and melt (each with their individual compositions). It can be used to model melting and melt migration in the lowermost mantle.

(Reference: <https://github.com/geodynamics/aspect/pull/4703>)

## Add support for compressed input data

(Rene Gassmoeller)

ASPECT's StructuredData input classes have been augmented with support for compressed files in the gzip format (.gz). This can save about 80% of storage space for input data that is part of the repository, and makes it faster to download ASPECT. It can also speed up the process of loading input data on systems with slow hard-drives.

(Reference: <https://github.com/geodynamics/aspect/pull/4717>)

## Add support for sharing data between processes

(Wolfgang Bangerth)

ASPECT's StructuredData class can now share data between MPI processes located on the same physical machine. This is useful when a simulation requires large data tables because these now only have to be stored once per machine, rather than once per MPI process. On current machines that may have 32, 64, or even more processor cores, this can make a substantial difference and may allow working with data sets that would be too large to keep in memory once per core.

(Reference: <https://github.com/geodynamics/aspect/pull/4086> )

## Add new advection darcy field method

(Daniel Douglas)

Add a new advection field method which allows for compositional fields to be advected according to Darcy's law, instead of using the Stokes velocity.

## Add test for fixing temperature in a compositional field

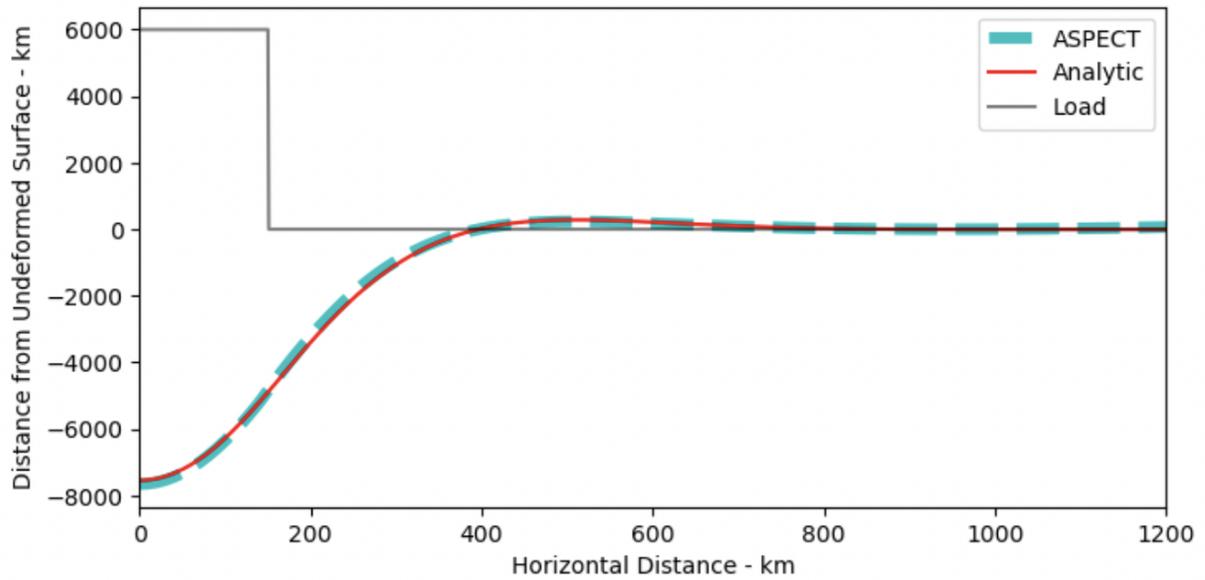
(Daniel Douglas)

Created a test case where a user specified compositional field(s) have temperature fixed to the temperature it was initialized with.

## Add benchmark for lithospheric flexure with infill material

(Daniel Douglas)

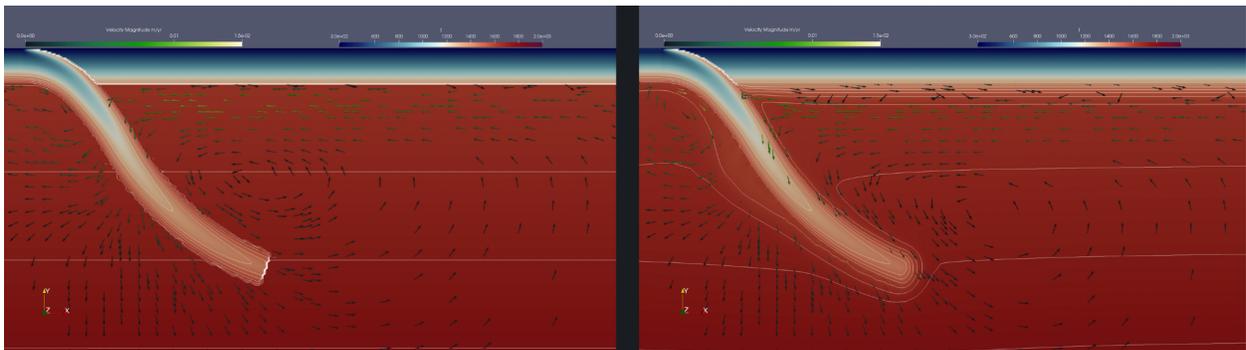
This is old code that I used for my masters project that I never merged. I created and documented a benchmark case for using this plugin to model lithospheric flexure.



## Add subduction cookbook for fixing temperature inside slab composition

(Daniel Douglas)

Created a cookbook demonstrating an application for the fixed temperature in a compositional field plugin I implemented (see above).



## Add strain rate particle property

(Agi Kiraly)

I added a particle property to track the strain rate over time on the particles. The test for this has a few particles in the middle of a simple shear box.

(Reference: <https://github.com/geodynamics/aspect/pull/4690>).

## Code cleanup: Remove the reference viscosity

(Juliane Dannberg)

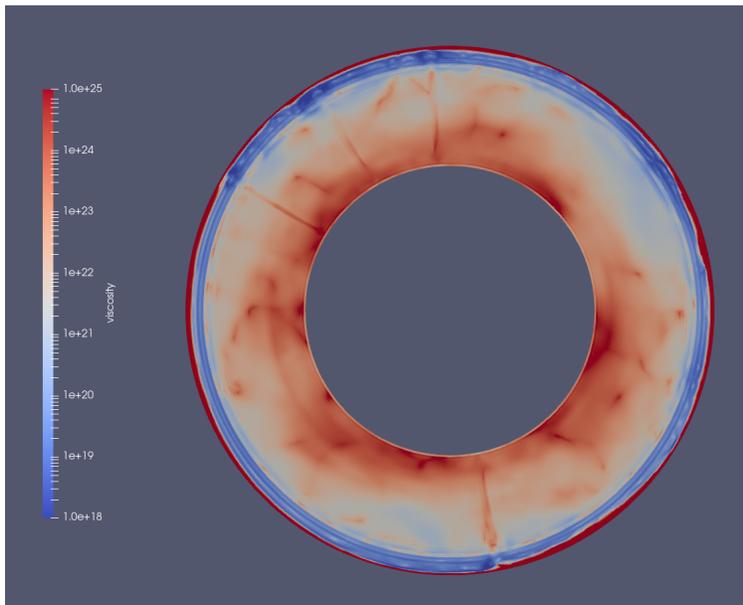
In previous ASPECT versions, material models were required to provide a reference viscosity that was used to scale the pressure when solving the equations. Instead, ASPECT now computes the average viscosity in the model in a given time step, so this parameter is no longer necessary. Accordingly, the reference viscosity has been removed from the material model interface and many of the individual material models.

(Reference: <https://github.com/geodynamics/aspect/pull/4722> and <https://github.com/geodynamics/aspect/pull/4733>)

## Models with an equilibrium grain size

(Juliane Dannberg)

ASPECT can now run models with a combined diffusion/dislocation rheology that takes into account the equilibrium mineral grain size.



## Quadratic least squares particle interpolation limiter

(Mack Gregory)

ASPECT can now use a particle interpolation scheme that reaches the velocity and pressure convergence expected on Q2xQ1 sSokes element without increasing the number of particles per cell with refinement, and without overshoots and undershoots of expected values.

(Reference: <https://github.com/geodynamics/aspect/pull/4600>)

## Adding a tutorial on nonlinear Stokes solvers to the manual

(Menno Fraters)

There are quite a few ways to solve nonlinear Stokes equations in ASPECT. I have added a section to the methods part of the new sphinx manual on the different ways of doing that, what the trade offs are and how they work.

(Reference: <https://github.com/geodynamics/aspect/pull/4611>)

## Adding CPO core plugins to ASPECT

(Menno Fraters)

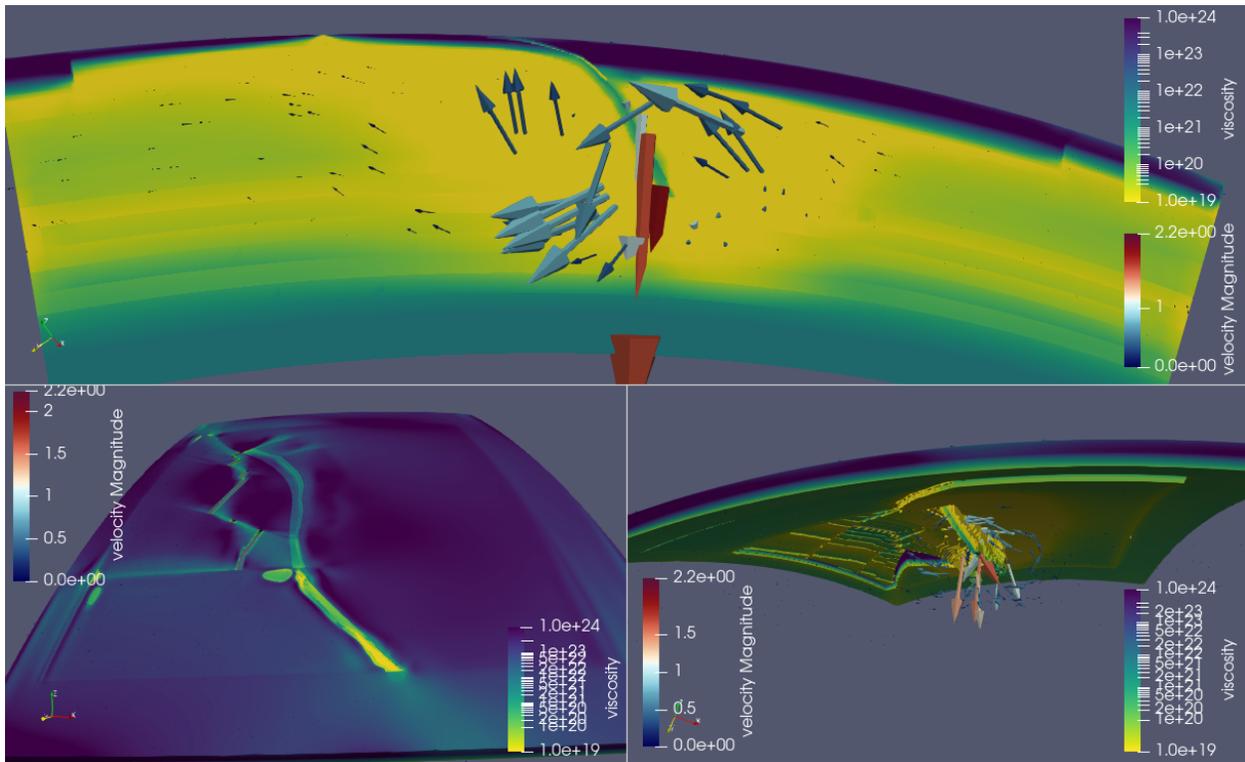
I made significant progress on getting the CPO core plugin to be able to merge it into the main branch of ASPECT.

(Reference: <https://github.com/geodynamics/aspect/pull/3903>)

## Discussed World Builder algorithms

(Menno Fraters, Wolfgang Bangerth)

We had a discussion on several of the algorithms used inside the Geodynamic World Builder (GWB), which is used in ASPECT. We discussed the KD-tree which is used for variable thickness area features, nonlinear spline solvers and slab2 integration algorithms.



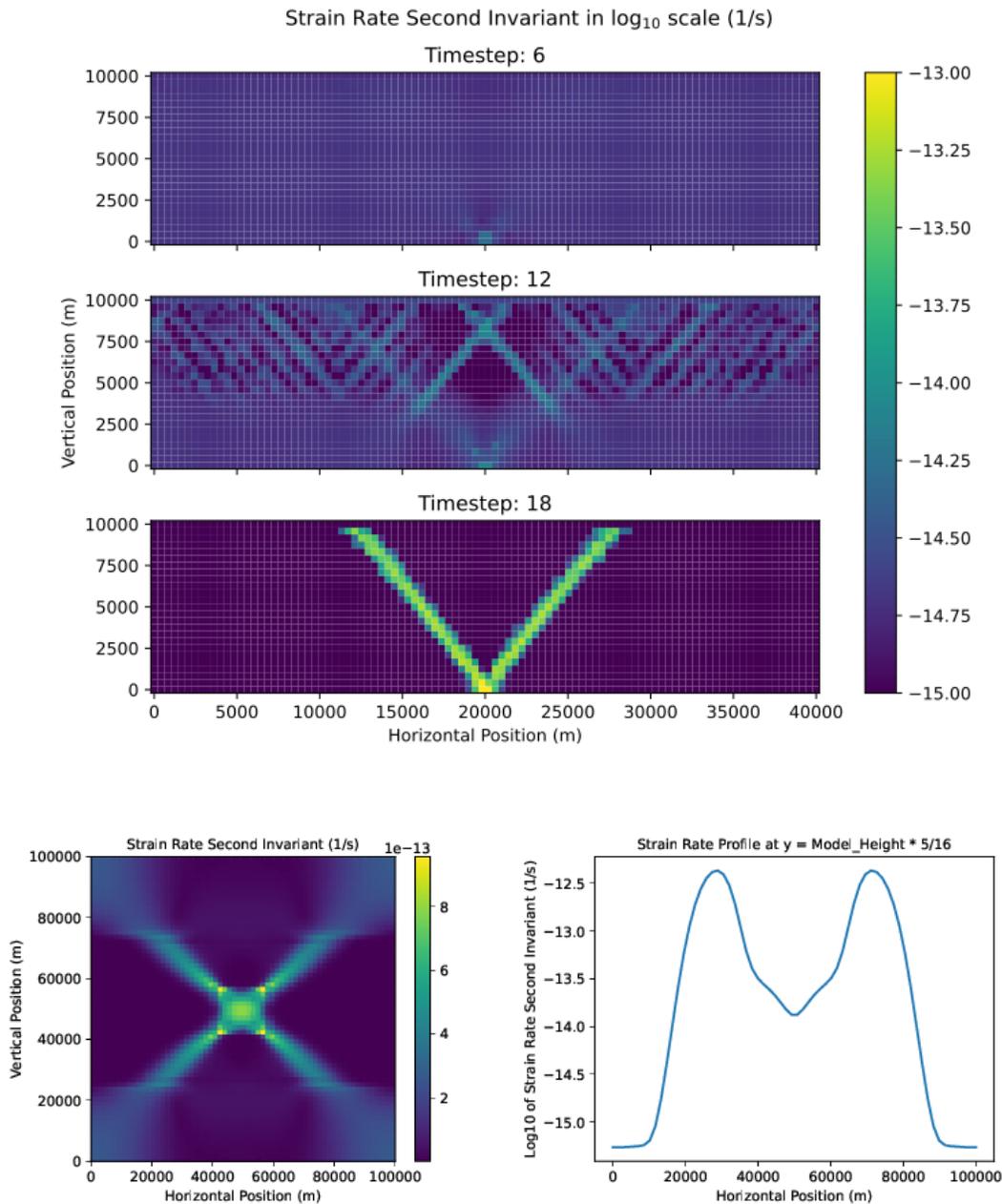
(Cascadia subduction model in ASPECT made with the GWB.

(Reference: <https://github.com/geodynamics/aspect/pull/3903>)

Add Kaus (2010) and Gerya (2019) documentation to existing .prm files

(Marcel Saaro & Cedric Thieulot)

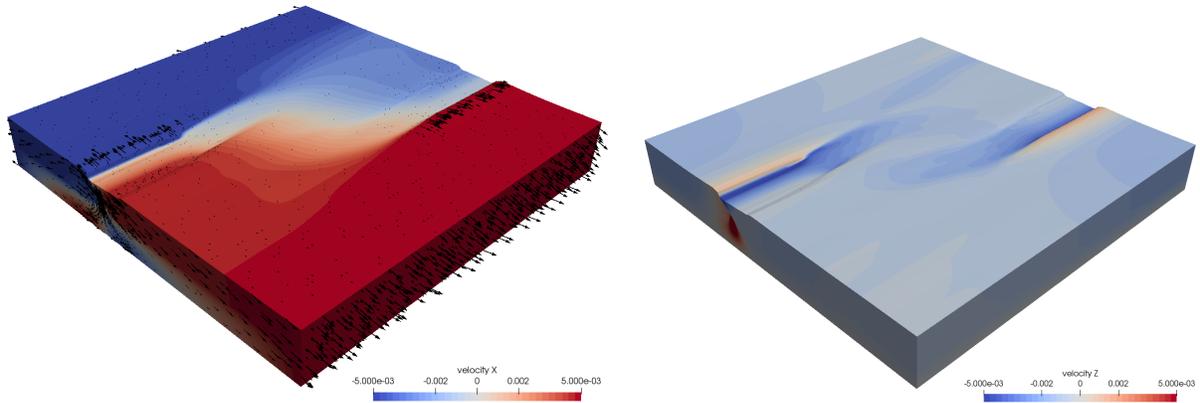
These .prm files had been merged but were never documented in the manual. Also, the provided python scripts to produce figures and cross sections were improved.



## Add Allken et al. (2012) cookbook

(Cedric Thieulot)

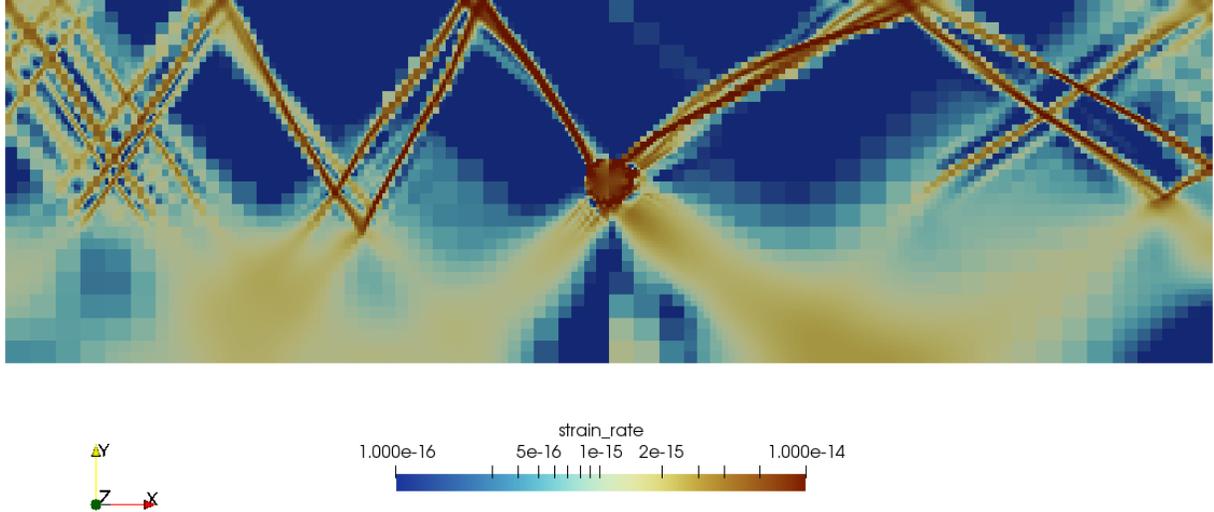
This is a 3D crustal model consisting of a plastic upper crust and a linear viscous lower crust. Two plastic weak zones are placed at the bottom of the upper crust and generate two rift systems, which in time progress towards one another and ultimately interact (or not), depending on their initial spacing and the viscosity of the lower crust as shown in the paper.



## Adding Duretz et al. (2020) elastoviscoplastic benchmark

(Cedric Thieulot)

This is a thermo-mechanically coupled setup. The domain is subjected to kinematic boundary conditions, which cause a pure shear stress state. The domain is 100x30km and the temperature field is such that in the bottom third the deformation is primarily following dislocation creep rheology while it is visco-plastic in the top part. The weak inclusion in the middle serves to initiate localized deformation. The left half of the figure is the extensional case while the right half is the compressional one.



## Enable GMG+free surface+elasticity

(Anne Glerum, Timo Heister, John Naliboff, Jiaqi Zhang)

The block GMG Stokes solver now works for problems with free-surface boundaries and elasticity.

(Reference: <https://github.com/geodynamics/aspect/pull/4754>)

## Boundary traction in spherical geometry/coordinates

(Maaïke Weerdesteijn, Rene Gassmoeller)

In the boundary traction plugin that uses a function you could define the boundary traction in spherical coordinates (by setting coordinate system to spherical), but there was no conversion from spherical to Cartesian coordinates happening. This is now added. You also have to set a new parameter 'Use spherical unit vectors' to true. Tests are added for using Cartesian and spherical coordinates in combination with a free surface.

## Testing open boundaries in spherical geometries with free surface

(Maaïke Weerdesteijn)

Open boundaries in spherical geometries (tested with the chunk geometry) with free surface (no boundary traction applied on top surface) cause strain rate anomalies at the corner of the bottom boundary and lateral boundary. These anomalies affect the free surface topography.

This also occurs at the corners of different mesh resolution regions. In the spherical shell geometry one cannot apply open boundaries. There is an inconsistency in conditions between box, spherical shell, and chunk geometry in the function which finds whether a point is in the domain.

(Reference: <https://github.com/geodynamics/aspect/issues/4715>)

## Loading netCDF data into StructuredData

(Timo Heister, Paul Bremner)

Instead of loading .txt ASCII files, we are adding support to load structured data from netCDF files. The pull request is currently under review, and the broader goal remains work in progress.

# Statistics about ASPECT's growth during the hackathon

The following contains a number of statistics about how much ASPECT has grown during the hackathon (May 16 - May 27):

- 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

For comparison, these were the statistics for last year's (2021) hackathon:

- Number of source files in ASPECT before/after: 615 -> 625 + 10
- Lines of code in ASPECT before/after: 152,700 -> 155,981 + 3,281
- Number of merged pull requests before/after: 2954 -> 3081 + 127
- Commits in github before/after: 8704 -> 8950 + 246
- Number of tests before/after: 890 -> 914 + 24

The comparison of these numbers suggests that the 2022 hackathon was more productive than the one in the previous year – a welcome quantitative confirmation of our long-held belief that holding hackathons in person is preferable to virtual versions. Compared to the nine hackathons we have held over the years, this hackathon ranks in the top half in terms of productivity.

The difference between the second number in the second table (at the end of the 2021 hackathon) and the first number in each column of the first table above it (at the start of the 2022 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 through 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/user | egrep '\.(md)$' | wc -l`