ASPECT Hackathon 2020

Meeting places

Zoom Rooms (open 24/7, common password: 142784):

- Group room 1 (Rheology): https://ufl.zoom.us/j/96328571830 (email juliane.dannberg@ufl.edu for questions)
- Group room 2 : https://ufl.zoom.us/j/98963981950 (rene.gassmoeller@mailbox.org)
- Group room 3: https://clemson.zoom.us/j/91183804485 (email heister@clemson.edu)
- Lounge / Main meeting room (/ European room! During europe daytime): https://ucdavis.zoom.us/j/94512508603 (email mrfraters@ucdavis.edu)
- **Slack Workspace (cig-aspect.slack.com):** https://join.slack.com/t/cig-aspect/shared_invite/zt-gdn947lt-xRu8SpWkzI2rgnfOLCOpYw
- **Hackathon log (this document):** https://docs.google.com/document/d/1PhW6c4nxJ_CdRJawra1YtjA_5o89fzS4G5tXEaKZSS4/edit?usp=sharing
- **When2Meet to indicate availability:** https://www.when2meet.com/?9448005-HS3wD
- **Blackboard link (optional meeting room with shared whiteboard and unlimited breakout rooms):** https://eu.bbcollab.com/guest/2d869c82cd8743b1abd92da09d0cf88e (email bob.myhill@bristol.ac.uk if not working)
- **Facebook:** https://www.facebook.com/groups/1438286143140671 (Cedric & Lorraine admin)

Things to do before the start of the hackathon

- Create an account on https://github.com if you don't have one yet.
- Install the latest developer version of ASPECT (this requires deal.II version 9.2.0 or the current deal.II development version). You can follow the instructions here: https://geodynamics.org/cig/events/calendar/2020-aspect-hackathon/technical-requirements/
- Put your name into the table "Participants and areas of interest" below and fill in your goals and interests for the hackathon
- Fill out the When2Meet link above with the times that you expect to be working on ASPECT during the time of the hackathon. Please note: the core hours will be 9am to noon Pacific time (18:00 to 21:00 Central European Summer Time). Confirm that you have your timezone selected (upper left drop down) and ignore times that do not fall into Mon-Fri. Try to keep this information updated during the hackathon.
  This will help us to form smaller groups (based on interests and time zones) where we mix new users, more experienced users, and principal developers, so that everyone knows where they can get help when they have a question.
- Join the Slack workspace: https://join.slack.com/t/cig-aspect/shared_invite/zt-gdn947lt-xRu8SpWkzI2rgnfOLCOpYw

Introduction

To further develop the geodynamic modeling code ASPECT and to grow and foster its user community, 25 users and developers of ASPECT worked virtually over two weeks in August 2020.

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

Timeline

Day	**Scheduled items** **(in Pacific time; Central Europe: add 9 hours)**
Monday, 08/03	9 am: Welcome, Introduction, Reiterating technical prerequisites, What is a hackathon (Timo), Code of Conduct (Lorraine) 10 am: Introductions 11 am: Git Pull Requests (Rene)
Tuesday, 08/04	10 am: Daily rounds 11 am: IDEs and workflow demos (Rene and Timo)
Wednesday, 08/05	10 am: Daily rounds 11 am: Discussion of visco_plastic/rheology module structure + current (hackathon) and long-term plans 1 pm: Discussion of topography related issues (ZOOM ROOM 3) 2 pm: Discussion of adding melt into visco_plastic
Thursday, 08/06	10 am: Daily rounds 11 am: Numerical properties of the free surface scheme
Friday, 08/07	9 am: Logo discussion 10 am: Daily rounds 11 am: how ASPECT plugin systems work 4 pm: Happy hour in Lounge
Monday, 08/10	9 am: Debugging showcase 10 am: Daily rounds 11 am: BC for diffusing the strain rate/strain
Tuesday, 08/11	10 am: Daily rounds 11 am: Elastic time stepping
Wednesday, 08/12	9am: GMG discussions with guest Conrad 10 am: Daily rounds

	10:30 am: Usage analytics 11 am: Teaching Resources 12 am: Plasticity implementation
Thursday, 08/13	9 am: New Logos 10 am: Daily rounds 11 am: Dynamic topography / Free surface
Friday, 08/14	10 am: Daily rounds 11 am: Logistics, Announcements, The Future

Participants and areas of interest

Name, affiliation, email	**Time zone**	**Goals and interests for this hackathon**
Rene Gassmoeller, UC Davis, rene.gassmoeller@mailbox.org	US East Coast	1. Help others achieve their goals 2. Review pull requests 3. Cleanup duty (cookbook folder) 4. Entropy equation solver
Lorraine Hwang UC Davis ljhwang@ucdavis.edu	PDT UTC+7(?)	1. Logistics 2. Reporting 3. Baking 4. ASPECT Networks 5. Planning
Wolfgang Bangerth Colorado State University bangerth@colostate.edu	US Mountain Time	1. Review pull requests 2. Help others 3. Write documentation 4. Output for surface quantities 5. Play with the DiscreteTime class 6. Jacobi preconditioner for surface equations 7. Convert stress/shear stress/strain to output tensors 8. Some constexpr stuff 9. Run the extension benchmark with tensor viz
Juliane Dannberg University of Florida judannberg@gmail.com		1. Help others 2. Review pull requests 3. If there is time, start a repository with resources for teaching
Timo Heister heister@clemson.edu	US, Eastern	1. Review pull requests 2. Help others

	Standard Time	3. Infrastructure work (testing, cmake, etc.) 4. Linear solvers (multigrid, Schur complement) 5. Geometry representation 6. Stabilization schemes
Menno Fraters UC Davis menno.fraters@outlook.com	US, Pacific Standard Time	1. Help others 2. Review pull requests 3. (Finally) finish the isotherms plugin 4. Add LPO related code 5. Add defect correction Picard iteration solvers 6. Discuss addition of melt and GMG to Newton/DC Picard solver
John Naliboff New Mexico Tech john.naliboff@nmt.edu	US., Mountain Standard Time	1. Help others achieve their goals 2. Review pull requests 3. Finish existing nearly completed pull requests - particles + elasticity, peierls creep. Dynamic friction 4. Additional VEP examples (with Bob) 5. Plasticity stabilization 6. Compressible visco-elastic-plastic rheology 7. Work on two-phase flow + plasticity implementation 8. Get a plan for CRUST 1.0 - use libdap?
Anne Glerum	CEST	1. Free surface stability (timestepping & diffusion) 2. Free surface + particles 3. Compressible visco-elasto-plastic rheology 4. Extract solution on surface mesh (Wolfgang) 5. Finish subduction cookbooks (Menno & Magali & Haoyuan 6. Help and review pull requests
Jacky Austermann Lamont-Doherty Earth Observatory jackya@ldeo.columbia.edu	US, Eastern Standard Time	1. Adjoint equations for stokes flow, go from 'hacked in' version to 'mergeable version' 2. Viscoelasticity / GIA: help with benchmarking / implementation of sea level equation if useful 3. Start on an extensional basin related project - figure out what's there and what needs implementing (talk to Anne & John :) 4. Learn about ongoing global geodynamics, dynamic topography, geoid projects 5. Help and review
Robert Citron UC Davis ricitron@gmail.com	US, Pacific Standard Time	1. Model viscoelastic relaxation of impact basins on planetary bodies (similar to this) 2. Insert initial crater topography in 2D and 3D, and check if the mesh accurately tracks deformation 3. Benchmark relaxation of crater topography, crustal thickness, and temperature distribution

Arushi Saxena University of Florida saxena.arushi@ufl.edu	US, Central Standard Time	1. Setup instantaneous 3D mantle models employing earth-like temperature and viscosity distribution and compare the modeled surface strain-rates and velocities with the observed GPS data. 2. Work on the open issues in the ASPECT github repository. 3. Participate (and contribute?) in the plasticity formulation with John.
Magali Billen	US, Pacific Standard Time	1. Help get the phase transition pull request done. 2. Set-up a 2D visco-elastic subduction model with an initial slab geometry from world-builder 3. Add a half-space derived temperature for initial slab temperature 4. Learn about melt module 5. Get better at Git
Esther Heckenbach GFZ Potsdam hecken@gfz-potsdam.de	CEST	1. Discuss and work on rate-and-state friction 2. Time-stepping for 1. 3. Work on open issues
Maaike Weerdesteijn UiO Oslo m.f.m.weerdesteijn@geo.uio.no	CEST, but I can be flexible and work late European hours	1. Viscoelasticity / GIA a. Surface loads on a 3D spherical shell b. Sea level equation implementation c. Include load gravitational attraction d. Include time-varying ocean basins
Antoniette Grima University of Texas agrima.ig@utexas.edu	GMT +1	1. Free surface and dynamic topography in subduction zones (2D and 3D) 2. Set-up a 2D visco-elastic subduction model with an initial slab geometry 3. Setup/add infinitely sharp/discontinuous phase transitions in a multimineralic slab 4. Continue the pull request of phase transition 5. Get more familiar with: a. Git b. Modifying and benchmarking code
Haoyuan Li hylli@ucdavis.edu	US, Pacific Standard Time	1. Set-up a 2D visco-elastic subduction model with an initial slab geometry. 2. Continue the pull request of phase transition 3. Learn more 4. Want some advise on setting up tests and plugins; how to parse in parameters

		5. Looking for scripts and experience on either visit or paraview
Kiran Chotalia University of Florida kiran.chotalia@ufl.edu	BST (GMT+1)	1. Periodic boundary conditions in 2D spherical shell section to examine lower mantle mixing 2. Getting familiar with a. How to adapt code b. Testing adaptations c. Raising issues with Git
Erin Heilman erin.heilman@utexas.edu	US CDT	1. Strain weakening in viscoplastic 2D cylindrical 2. Work on strain healing implementation 3. Get more comfortable with modifying the code and understanding the libraries
Bob Myhill University of Bristol bob.myhill@bristol.ac.uk	BST (GMT+1)	1. Continue viscoelastoplastic models w/ free surface (w/ John) 2. Phase ID from thermo lookup (w/ Juliane, Kiran) 3. Discuss compositional freedom for material models (for reactive flow)
Cedric Thieulot c.thieulot@uu.nl	CEST but I can be flexible and work late	1. Improved plasticity -> John 2. Adding cookbook(s) 3. Discussion on free surface
Marie Kajan University of Florida	US EDT	1. Free surface 2. Dynamic topography and geoid 3. Work on open issues
Daniel Douglas University of Hawaii daniel92@hawaii.edu	HST	1. Dynamic Topography 2. Peierls Creep Mechanism 3. Free Surface processes
Fiona Clerc MIT/WHOI fclerc@mit.edu	EDT	1. Melt migration in VEP rheology (esp. mantle depths) 2. If above works, possible to recover features of dike initiation/propagation ? 3. Check melt migration and VE implementation using manufactured solution 4. Interested in group doing VE subduction model
Jonathan Perry-Houts	PDT	General interests: 1. Continental tectonics 2. Volcanism & melt transport Hackathon plans: 1. Two-phase flow + plasticity 2. Lower crust

Kodi Neumiller	MDT	1. Work on format converter (netcdf -> sph specifically) 2. Create a plugin to read netcdf ascii data and pass in the data directly to the appropriate function 3. Improve url reader within ASPECT 4. Get a better understanding on certain parts of Aspect code 5. Discuss URL reader

We will sometimes want to meet in loosely organized groups centered around certain topics:

Rheology **Group 1** **https://ufl.zoom.us/j/96328571830** Password 142784	**Subduction, phase transitions** **Group 2** https://ufl.zoom.us/j/98963981950 Password 142784	**Surfaces, free surfaces, dynamic topography, GIA** **Group 3** https://clemson.zoom.us/j/91183804485 Password 142784
John N.	Menno	Wolfgang
Bob M.	Antoniette	Anne
Erin H	Magali	Daniel D.
Jonathan P-H	Rene	Maaike
Fiona C.	Haoyuan	Marie
	Kiran	Robert
Arushi	Kodi	Jacky
Juliane	Timo	Lorraine
Esther		
Cedric		

Proposed Topics for Presentations / Discussions

If you would like to give a talk on a specific area of research or interest related to the Hackathon, please list it below! We will find an appropriate time for these talks throughout the Hackathon during the listed online hours (10 am - 1 pm PT).

- Overview of VEP rheology in ASPECT and future plans? (John Naliboff)
- Data extraction and analysis with python?
- Discussion about new ASPECT logo (Rene)
- Discussion about measuring the impact of ASPECT (Rene, usage statistics)
- Discussion on coupling nonlinear solvers (Menno, Timo)
- Discussion about the URL reader inside ASPECT (Kodi)
- Indentation
- BC for diffusing the strain/strain rate
- GMG conversation (Timo, visitor: Conrad, Wed/Thur/Fri week 2)

Resources

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. https://swcarpentry.github.io/git-novice/01-basics/index.html
    b. https://swcarpentry.github.io/git-novice/02-setup/index.html
    c. Config git name: `git config --global user.name "Vlad Dracula"`
    d. 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
    c. Ensure proper remotes
3. Walkthrough
    a. Create Branch
        i. 'git checkout master'
        ii. 'git pull upstream master'
        iii. 'git checkout -b remove_dealii_compatibility_fix'
    b. Make changes for DEAL_II_VERSION_GTE in one of:
        i. source/geometry_model/spherical_shell.cc Magali
        ii. source/mesh_refinement/volume_of_fluid_interface.cc (5 times)
        iii. source/particle/world.cc (4 times) Robert
        iv. source/postprocess/memory_statistics.cc (Antoniette and Kiran)
        v. source/postprocess/particles.cc (2 times) Daniel Douglas
        vi. source/postprocess/visualization.cc (2 times) Erin
        vii. source/simulator/assembly.cc (2 times) (Antoniette)
        viii. source/simulator/checkpoint_restart.cc (4 times) Magali
        ix. source/simulator/core.cc (3 times) Maaike
        x. source/simulator/entropy_viscosity.cc (2 times) Kiran
        xi. source/simulator/stokes_matrix_free.cc (8 times)Esther
        xii. source/volume_of_fluid/assembler.cc (3 times)Esther
        xiii. source/volume_of_fluid/reconstruct.cc (2 times) (Antoniette)
        xiv. source/volume_of_fluid/solver.cc (Rene)
    c. Create commit
        i. 'git add FILE'
        ii. 'git commit -m 'Removed a now unnecessary compatibility fix'
    d. Push and open PR
        i. 'git push origin remove_dealii_compatibility_fix'
        ii. Open PR on github (CTRL-Click on shown link)
    e. Wait for review

f. Address review (repeat steps b,c,d)
g. Success!

4. Now repeat the steps in 3. on your own. Pick a section of the manual 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

Update ASPECT to the new deal.II version

(Timo Heister, Rene Gassmöller, ...)

We removed support for deal.II 9.1 and removed the remaining compatibility code in ASPECT.

Remove deprecated MaterialModelInputs::cell

(Timo Heister)

We removed deprecated functionality.

Reorganize scripts

(Timo Heister)

We used to have all scripts inside ./doc/. Instead, they are now in

./contrib/utilities/ for indent, update_prm_files.sh, update_source_files.sh
./contrib/release/ for update_copyright, release-tasklist, bump_version.sh
./contrib/docker/ for the docker files (from ./docker)

AsciiDataLookup improvements

(Timo Heister)

The loading of data into AsciiDataLookup class has been refactored and separated from the file reading. This made it more robust (wrong column values will now be detected) and more flexible (data does no longer need to come from a txt file existing on disk). Various unit tests were added.

Unity builds

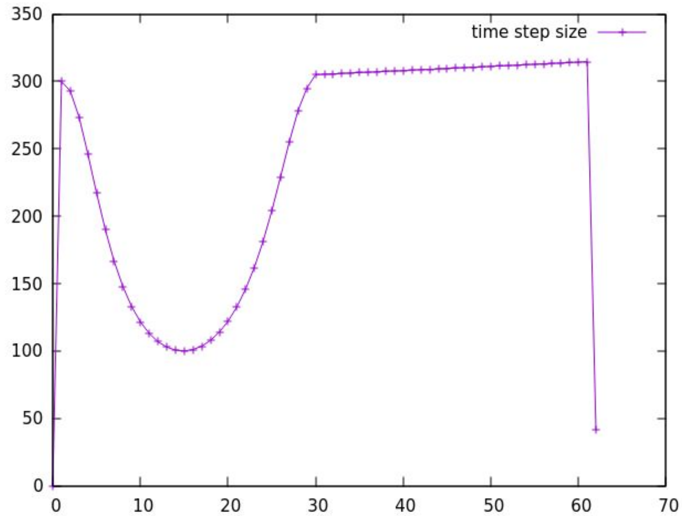
(Timo Heister, Rene Gassmoeller)

ASPECT now uses unity builds by default, speeding up compilation by 3x if CMake version 3.16 or newer is used. This required various smaller fixes throughout the library.

Time stepping control

(Timo Heister)

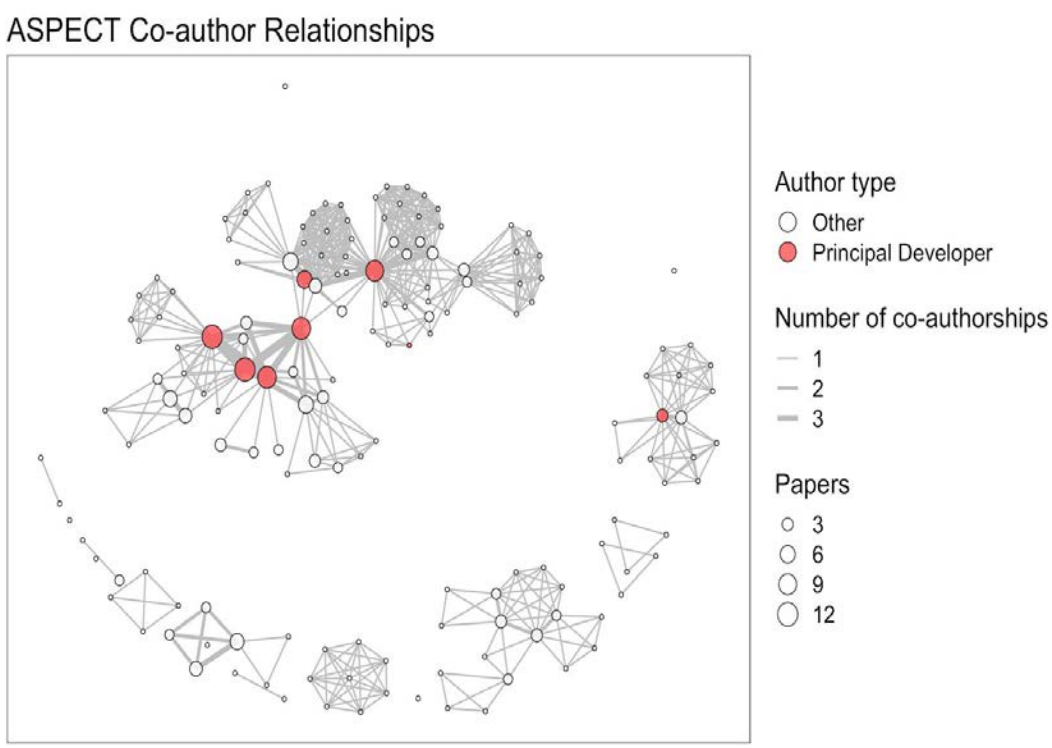
After refactoring and extending the logic to compute time step sizes, the user can now (optionally) precisely control the time step size, as seen in this example:



Update co-author networks graphs

(Lorraine Hwang)

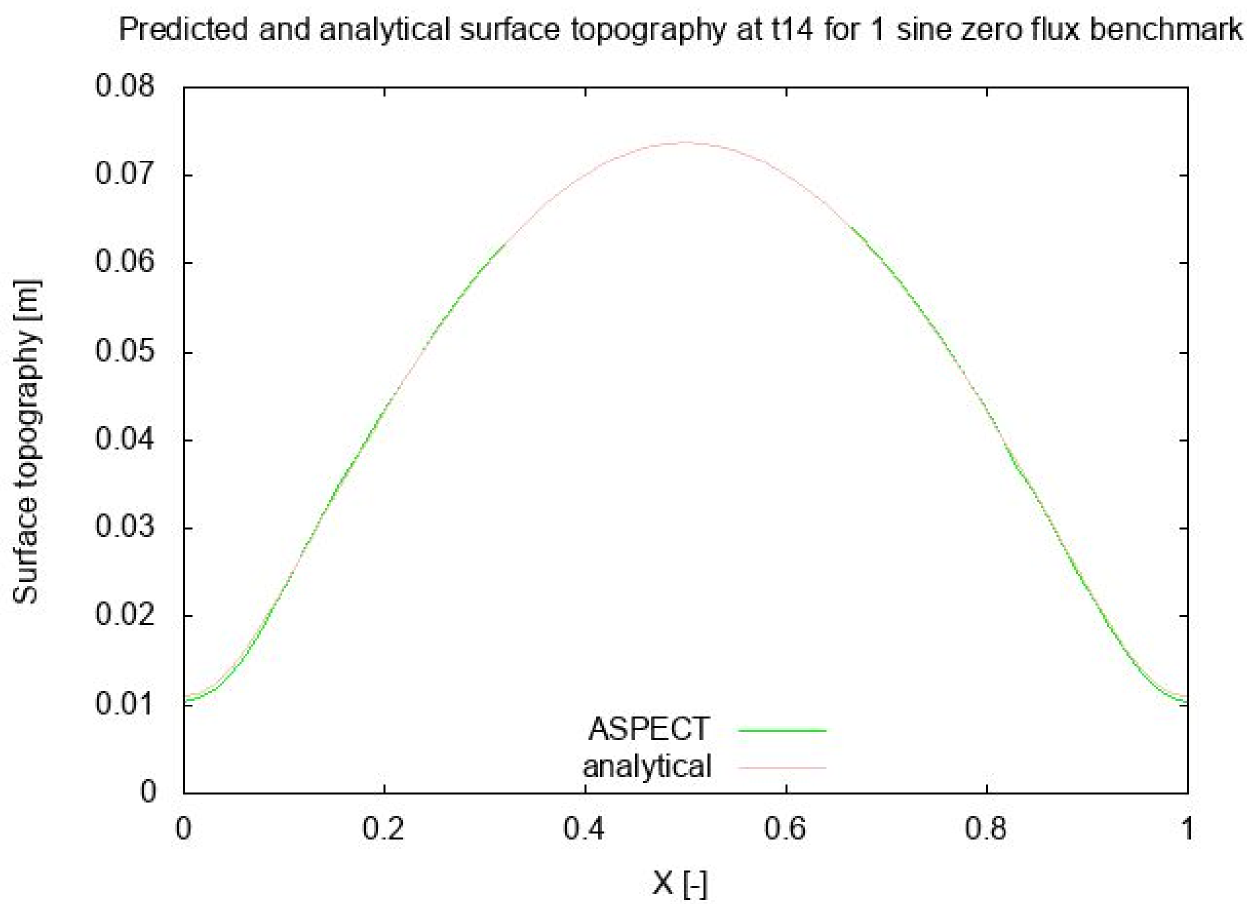
The following graph shows the relationship between co-authors of ASPECT-related publications. It covers 73 references and <156 authors ("Others" included with 3 publications):



Add diffusion mesh deformation plugin

(Anne Glerum)

The diffusion mesh deformation plugin applies an amount of user-defined diffusion to the topography of the surface boundary mesh. It can be used in combination with initial topography, and/or with other mesh deformation plugins like the free surface. When applied every timestep, the hillslope diffusion algorithm can be seen as a first order implementation of a simplified, diffusion-based erosion and deposition model. From a practical perspective, it acts to smooth sharp changes in topographic gradient, which can prevent distortion of the mesh cells. The figure shows an initial hill-like topography that diffuses over time without any other distributions to the surface topography.

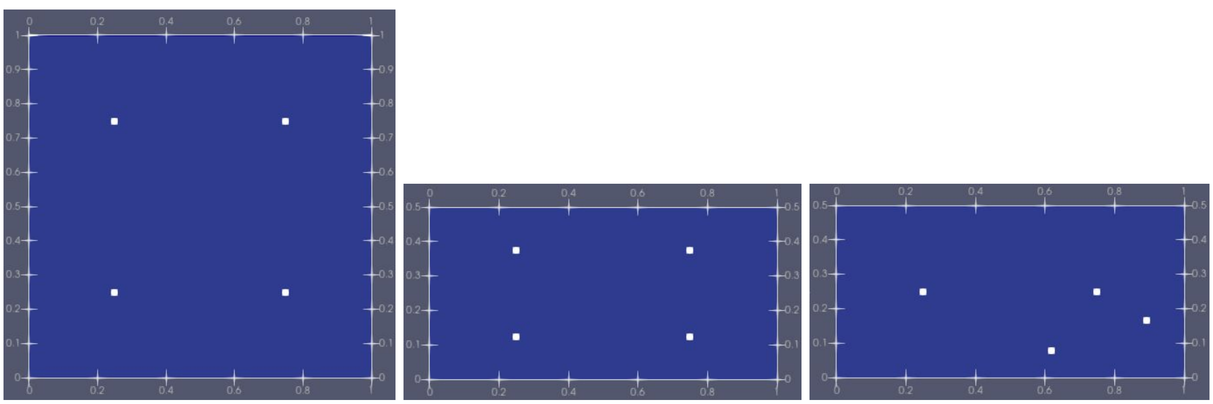


Reorder particle advection and apply additional particle sort when used in combination with mesh deformation

(Anne Glerum, Robert Citron, Rene Gassmoeller)

So far the use of particles with deformation of the mesh was by default disabled because it led to crashes and was untested. In particular, it was unknown whether the reference location of each particle was correct at the moment of compositional field assembly and solving, at which time data from active particles could be required. Mesh deformation modifies the mapping between the real cell and the reference cell. After mesh deformation, the cells were deformed, but while absolute particle positions were not changed (as particles were advected at the end of each timestep), their reference locations change in this process, but were (wrongly) not updated. Now, particles are sorted directly after mesh deformation, updating their reference location. Also, instead of advecting the particles at the end of each timestep, they are advected before the composition is solved for, such that their properties are in sync when required for assembly of the compositional field systems.

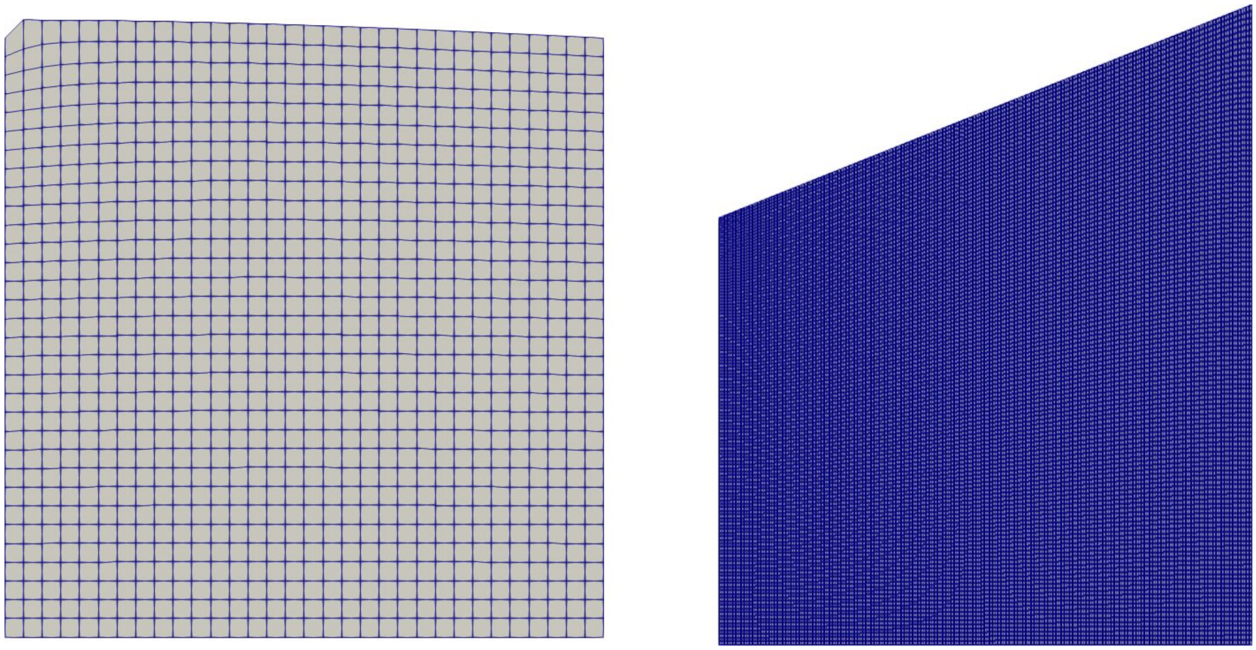
The figures below show tests as follows: (left) default behavior of four particles generated in an undeformed reference cell; (middle) four particles generated in a reference cell initially deformed by -0.5 with ascii data; and (right) an instance where the initial cell is deformed with a boundary function that moves the upper boundary downwards by 0.5 after one timestep, resulting in the two particles originally generated at y=0.75 to be removed when they are outside the cell and regenerated within the reference cell at random locations, while the two particles originally at y=0.25 remain unchanged.



Extension of mesh deformation plugins to allow initial deformation

(Rene Gassmoeller, Anne Glerum, Robert Citron)

The interface for the mesh deformation so far only supported the deformation of the mesh over time, but not to specify an initial (deformed) state. The initial state was often specified using the initial_topography plugin system, which actually deformed the initial coarse mesh (while the mesh deformation system acts using a modified mapping to simulate the deformed mesh). Using both in combination created awkward interactions and many bugs related to not properly adjusted meshes, or wrongly set initial conditions. We extended the mesh deformation interface to allow an initial deformation state to be specified. This can handle all the cases the initial_topography plugins can, is more general, and integrates better into the existing mesh deformation. In the long run the new system can replace all existing plugins (although all existing ones have to be ported individually to the new system). We started porting some plugins. The ascii data plugin has been ported.



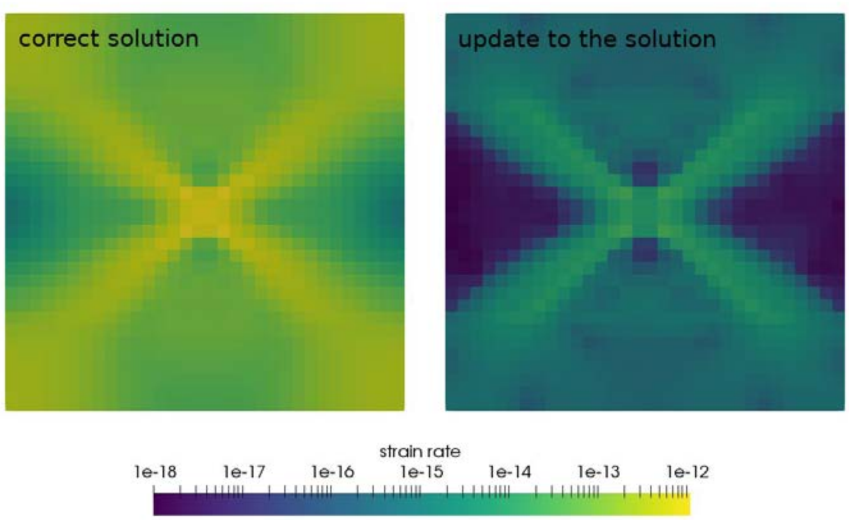
The two images above show two examples of initially deformed meshes created using the new interface.

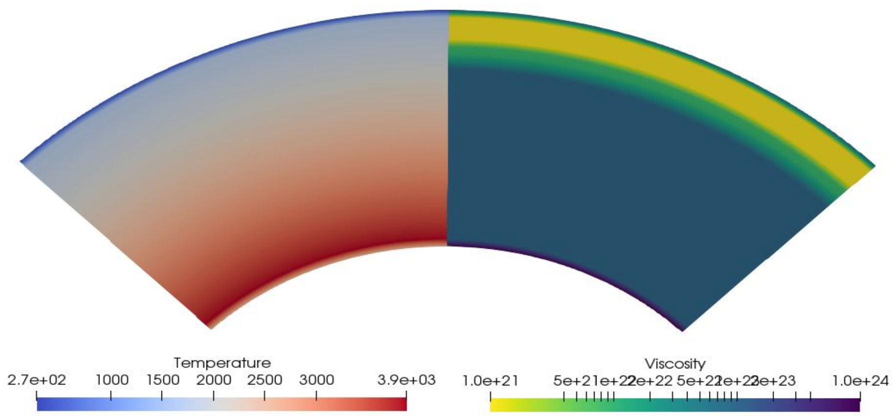
Bugfixes

(Juliane Dannberg, ...)

We fixed several bugs:

- Nonlinear iterations are now postprocessed correctly in the Newton solver schemes.
- The latent heat material models now use reasonable viscosity profiles in models with adiabatic heating.





Peierls Creep Approximation

(Magali Billen, John Naliboff and Bob Myhill)

We derived a general form of the Peierls creep approximation so users can choose their favorite flow law from the literature. We implemented this approximate formulation to be used in the visco_plastic material model.

Visco-elastic slab with Phase Transitions Cookbook

(Magali Billen and Haoyuan Li)

We created a simple 2D subduction cookbook using visco-plastic-elastic rheology, a new phase transition model to put in both density and viscosity phase transitions, and building the initial temperature and composition structure using WorldBuilder. The snapshots below show the initial temperature, composition, phase transitions and viscosity. In putting this model together, we identified one unexpected behavior of the phase transition model when there is a phase transition with non-zero Clapeyron slope near the surface where the temperature cools rapidly toward the surface. We will address this in a future pull request. We also started exploring how to add flow-through boundary conditions on the sidewalls as a way to reduce the size of the model. Working closely with the code maintainers made it possible to quickly learn about each of the input parameters and put this model together.

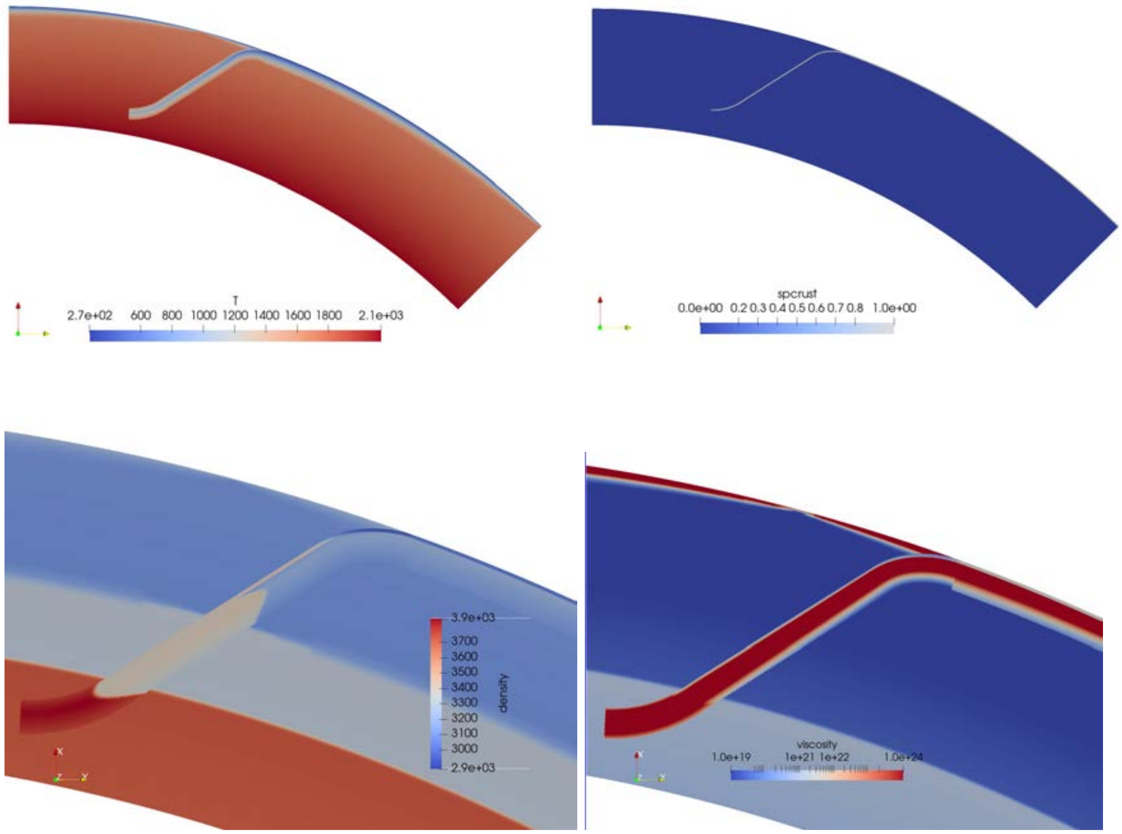
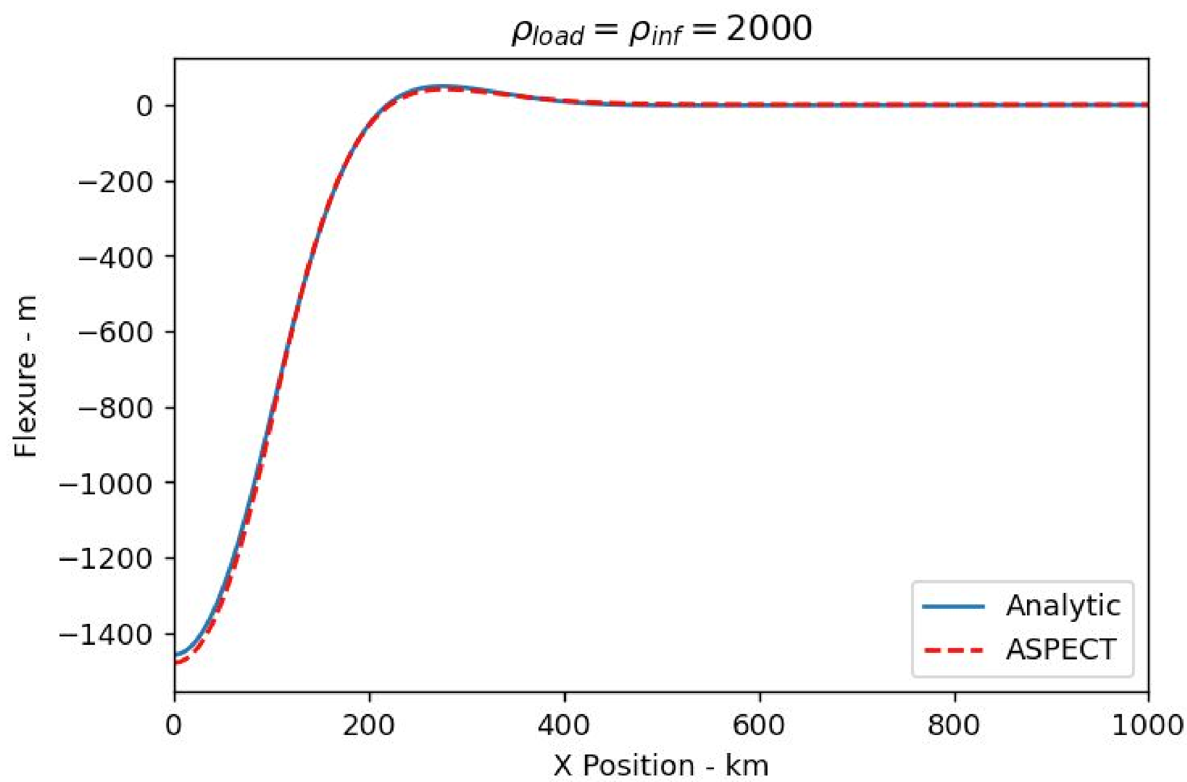


Figure: Initial conditions for visco-elastic model with phase transitions. Top left: Temperature with an overriding plate, a subducting plate and a slab. Top right: Crustal layer in the subducting plate and slab. Bottom left: Composition-dependent density with phase transitions. The basalt-to-eclogite transition occurs in the crustal layer around 80 km, while the olivine-wadsleyite (410 km) and ringwoodite to bridgmanite plus magnesiowustite (660 km) occurs in the surrounding mantle and slab. Bottom right: viscosity structure with layers. The phase transitions are also used to change the viscosity of the crustal layer and all the compositions at 660 km.

Flexure dependent traction cookbook

(Daniel Douglas)

I worked on creating a boundary traction plugin which updates the traction based on deformation of the free surface from a reference height. The new code adds a traction that is equal to rho*g*w, where rho is a density input by the user, g is gravity and w is the deflection of the free surface from its undeformed state. I still need to implement the cookbook into ASPECT.



Geoid postprocessor with free surface

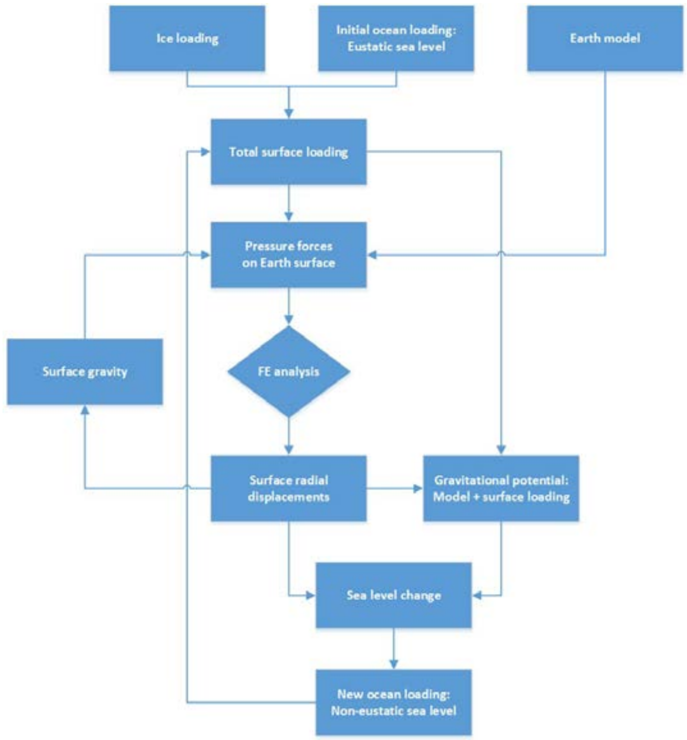
(Maaike Weerdesteijn, Rene Gassmoeller, Jacky Austermann)

We worked on getting the geoid postprocessor functioning with a free surface instead of only dynamic topography. The user can decide if the surface and/or CMB topography needs to be taken into account for the geoid calculation. If a free surface is present, it will use the actual topography of the deformed model, if there is no free surface, it will use the topography based on the stresses at the boundary (as computed by the dynamic topography postprocessor).

Sea level postprocessor reading topography and geoid

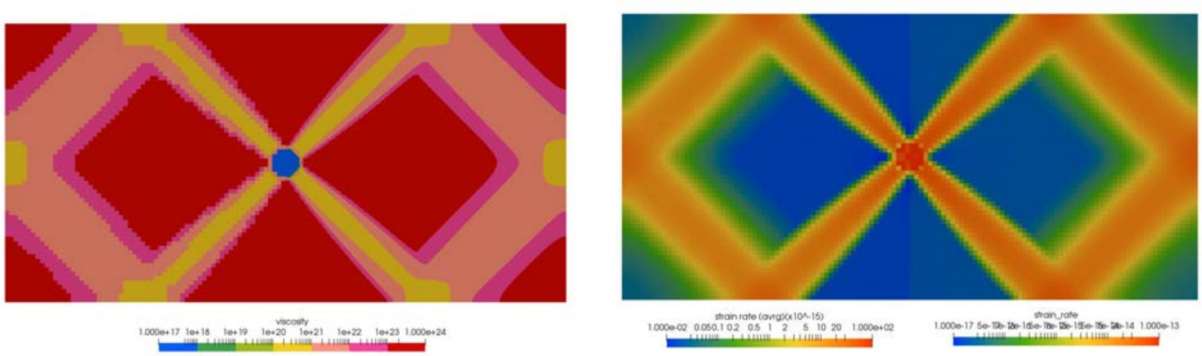
(Maaike Weerdesteijn, Rene Gassmoeller, Jacky Austermann)

We created a new sea level postprocessor which reads topography and geoid data for the relative sea level computation. More contributions need to be added (new water mass from melting ice, from ice loading input data, an ocean basin). And I need to use similar computations as in the geoid postprocessor to calculate the gravitational attraction of the surface loading. Eventually this sea level postprocessor will be used for glacial isostatic adjustment (GIA) modeling. Shown below is a schematic overview of the GIA implementation.



(Elasto-)visco-plastic experiment from Duretz et al., G3, 2018.

(Cedric Thieulot, John Naliboff)
We experimented with (elasto-)visco-plastic viscosities and the examples that can be found in Duretz et al., G3, 2018. Domain is 4x2km, inclusion has a 100m radius. Pure shear boundary conditions are applied.



Left half of plots obtained with dedicated python code, right half obtained with ASPECT.
Very good agreement after 200 nonlinear iterations. Visco-plastic rheology.

Added a capability to allow the gravity to be time dependent in the gravity function plugin

(Jacky Austermann)

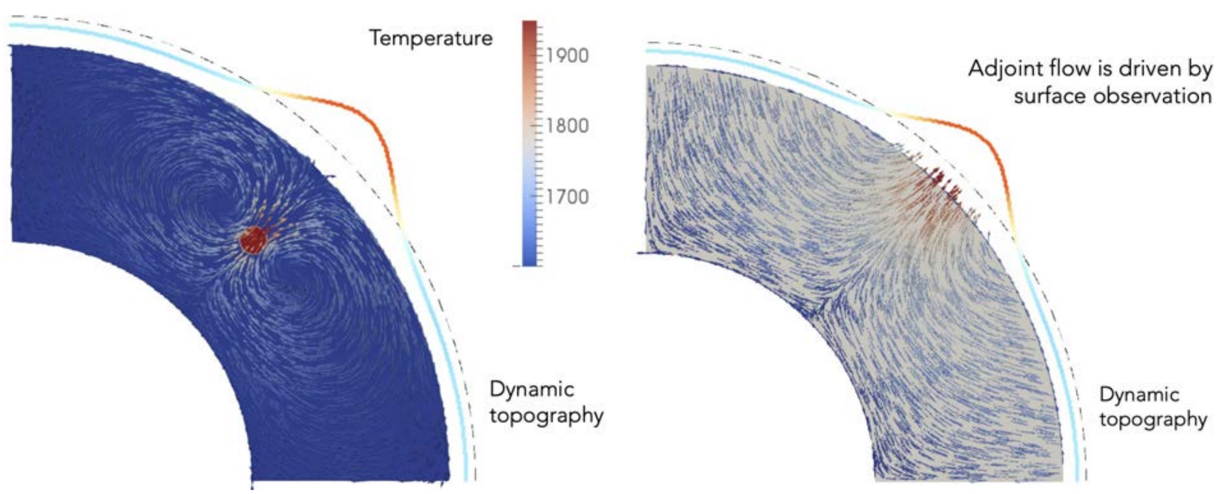
When choosing the gravity function plugin you can now include time in the function description and gravity will then vary as a function of time.

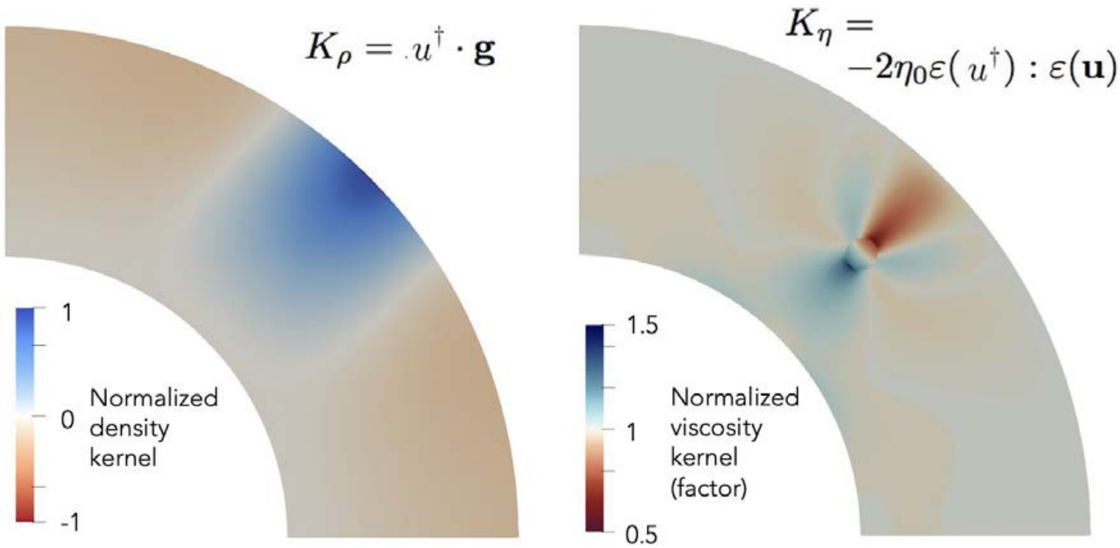
Improve and benchmark the implementation of the sensitivity kernels that are calculated by solving the adjoint equations

(Jacky Austermann)

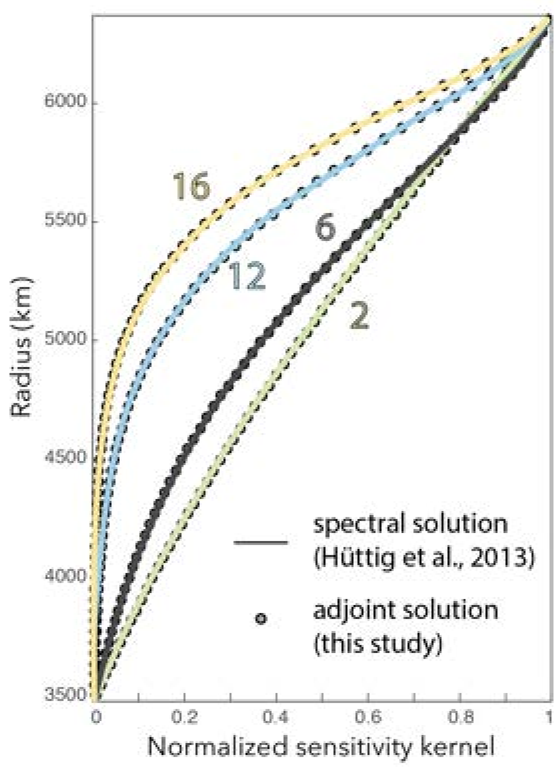
I had previously implemented a new solver scheme that solves the adjoint equations. During this hackathon I restructured my implementation so that it can be included in the developer branch. This restructuring included moving the assembly of the adjoint right hand side to a separate assembler (assembler/adjoint.cc) and the function that computes updates to the density and viscosity field into helper_functions along with a series of smaller cleanups.

Here is an example of a forward and adjoint flow field (top row) as well as the respective density and viscosity sensitivity kernel (bottom row). In this example I'm simulating a rising hot anomaly in a 2D spherical shell section. While the calculation is in 2D, it directly extends to 3D. The sensitivity kernels can be used to update the initial viscosity (or density) to improve a fit to dynamic topography surface observations.





I further worked on benchmarking my code. One benchmark that is possible since it's commonly done in spectral codes is to calculate the sensitivity of surface topography to density perturbations at different depths and different spherical harmonic degree. The figure below shows a spectral solution (solid line) and the adjoint solution (markers) for different spherical harmonic degrees (number next to each line). While the solution agrees very well within the mantle, there are still inconsistencies at the surface cell of the domain (not shown below). During the hackathon I worked on trying to better understand why this issue arises.



Several minor convenience features

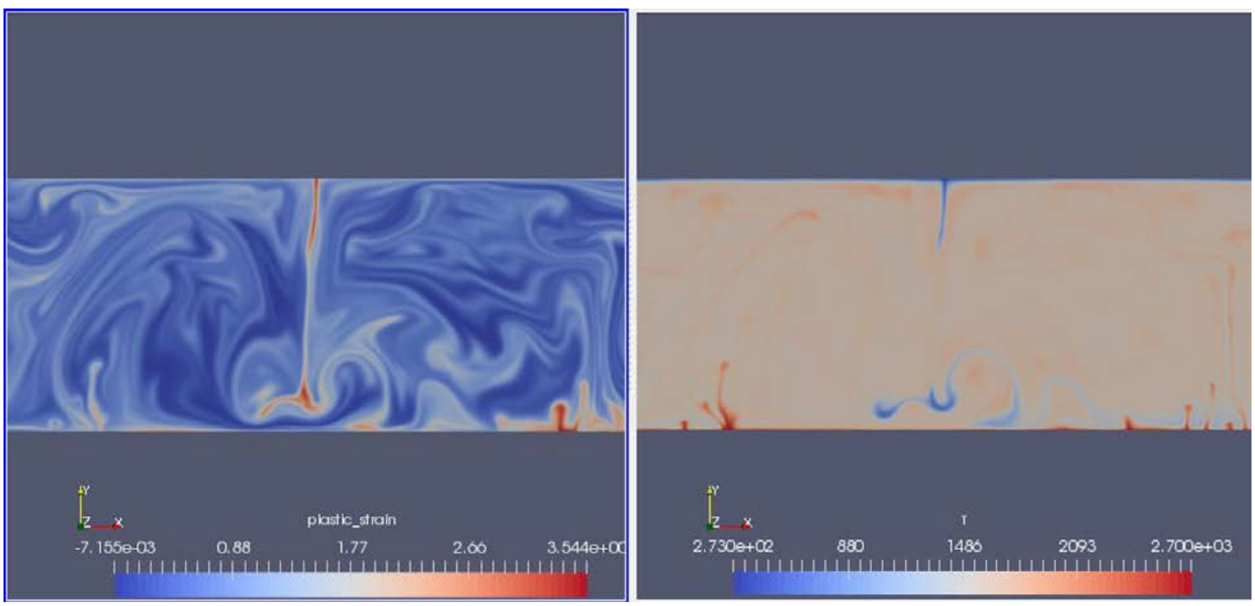
(Jonathan Perry-Houts)

- Automatically search for the correct astyle version in the indentation script. In the case where a user has multiple astyle binaries installed, they might have version 2.04 under a different name. In that case, checking `which astyle` will fail. Added ability to search all directories in user's $PATH for executables with 'astyle' in their name until the correct version is found.
- Syntax highlighting for .prm files in the Vim editor.

Added an option for Strain healing to strain dependent rheology

(Erin Heilman)

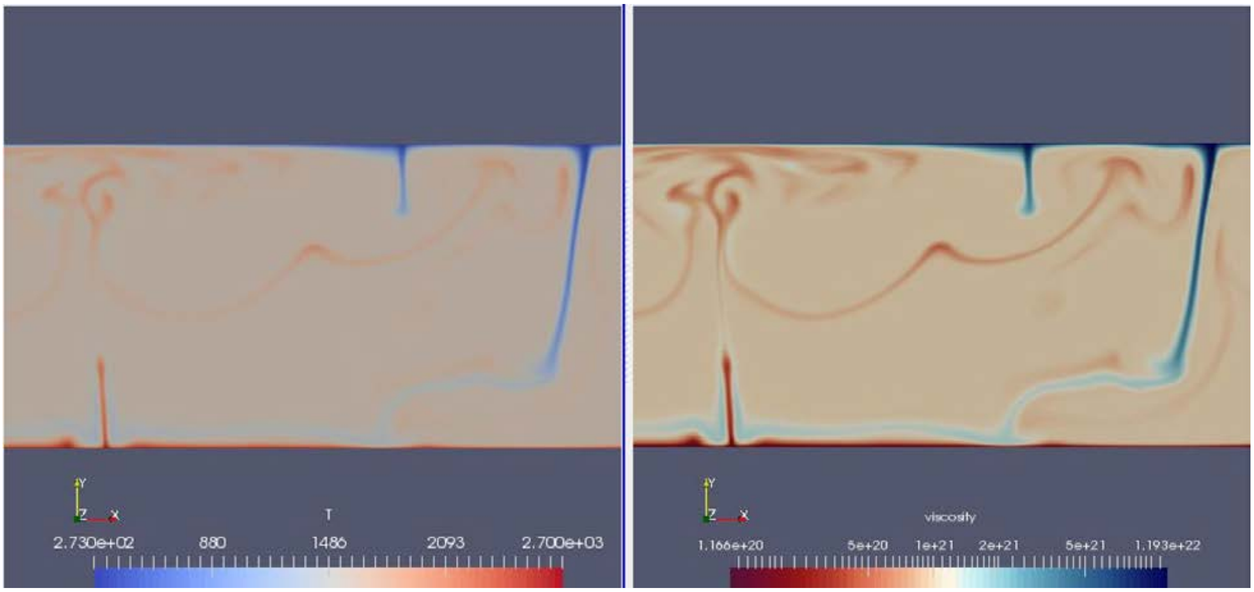
Added a temperature dependent strain healing case, equations from Fuchs and Becker 2019, to the strain dependent rheology material model. The figure shows accumulated plastic strain next to the temperature to highlight the temperature dependence of the strain healing, with strain accumulated at downwellings.



Added Frank-Kamenetskii viscosity to the rheology module

(Erin Heilman)

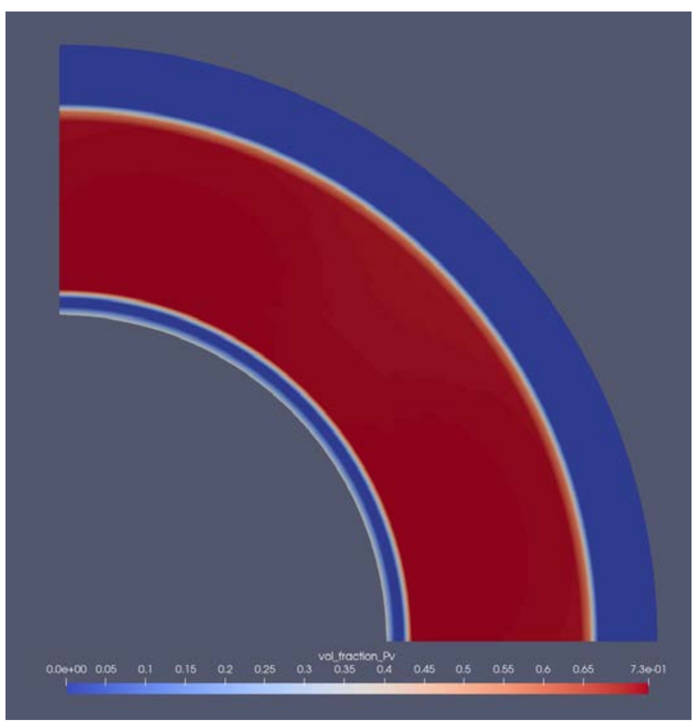
I added an implementation of a temperature-dependent Frank-Kamenetskii viscosity approximation to the visco plastic material model. The figure shows temperature next to the viscosity of the model highlighting the temperature dependence of viscosity with the new Frank Kamenetskii approximation. The inputs for the Frank Kamenetskii viscosity control the magnitude of the output viscosity.



Added phase volume fractions to the PerpleX lookup class

(Bob Myhill)

The PerpleX Lookup module can now accept phase volume fractions, for use in the material models or simply to visualize after the model runs. The image below shows the distribution of magnesium-silicate perovskite (by volume fraction) in a simplified pyrolitic world, as predicted by the Stixrude and Lithgow-Bertelloni (2011) thermodynamic database.



A python file is provided in the contributions folder to allow users to easily create input files in the correct format.

Modularized composite viscosity calculations

(Bob Myhill, John Naliboff, Cedric Theulot)

The diffusion and dislocation creep rheology modules now have a new function to calculate the strain rate and first stress-derivative of the strain rate. This will facilitate the development of complex composite rheologies.

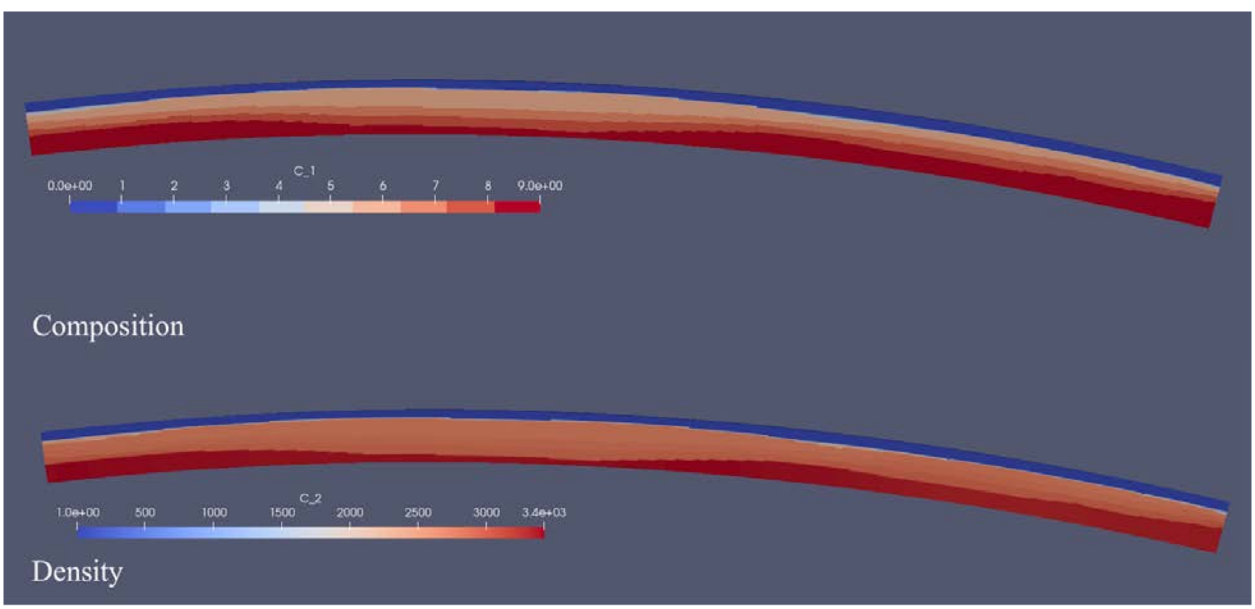
New cookbook using Crust 1.0 densities

(Bob Myhill, John Naliboff)

The AsciiDataLayered module added at the 2019 Hackathon allows the user to define compositional fields based on uneven boundaries in the domain. One dataset structured in this way is Crust 1.0 (Laske et al., 2013), available for download here: https://igppweb.ucsd.edu/~gabi/crust1.html

The ASPECT manual now has a new cookbook showing the user how to use the Crust 1.0 dataset in ASPECT, and how to read the density field into the material model density output.

The output generated by the new cookbook was used to generate the images below.



Added diffusion strain rate in the visco-plastic material model

(Arushi Saxena, Juliane Dannberg, Jonathan Perry-Houts)

Faulting in the lithosphere is usually approximated by the shear bands (regions where strain is localized) in the numerical models. However, the thickness of these shear bands are highly sensitive to the mesh resolution of the domain (de Borst et al., 1993), and the localized shear bands make the solution of the model difficult to converge. To overcome these issues in the numerical models, diffusion or averaging of the local field (e.g., strain rate invariant) is utilized.
We added diffusion of strain rate components into the existing visco-plastic material model. The following is a plot of strain rate invariant at different mesh resolutions for a benchmark numerical model (a weak block in the center of the domain to localize deformation), illustrating the effects on shear band widths after diffusing the strain rate components.

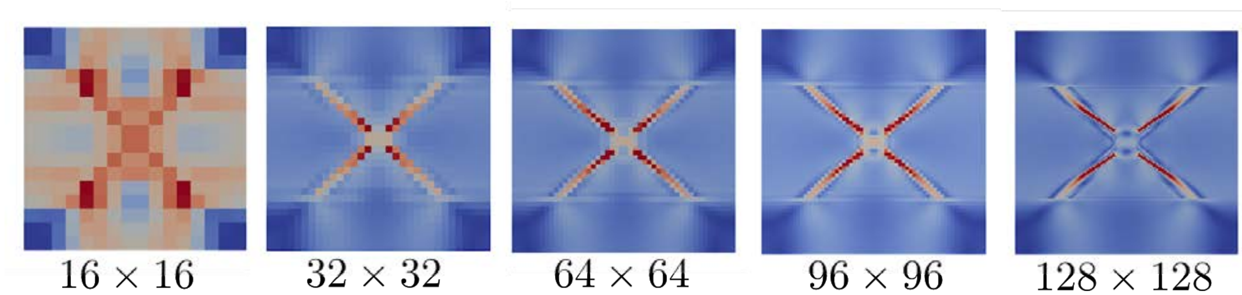


Fig.: Strain rate invariant for models with varying mesh resolution.

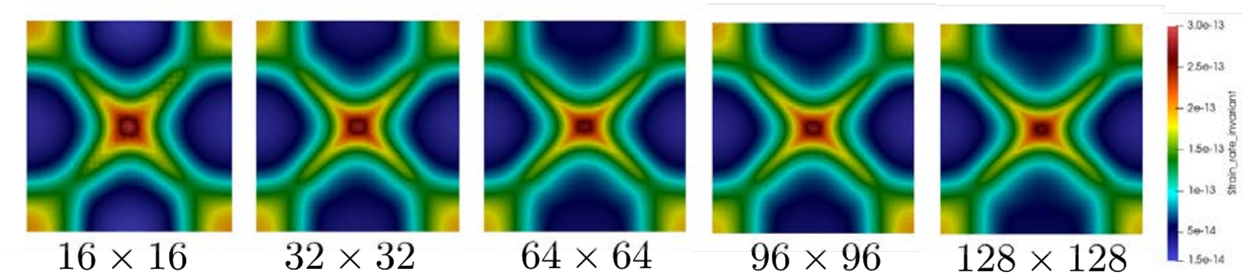
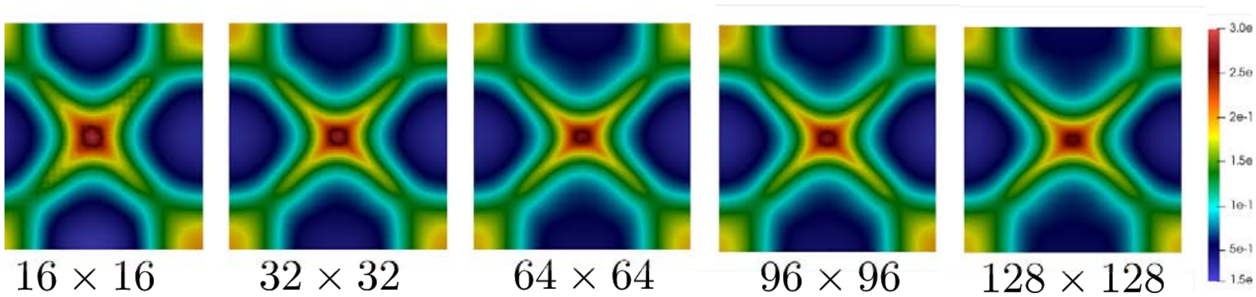
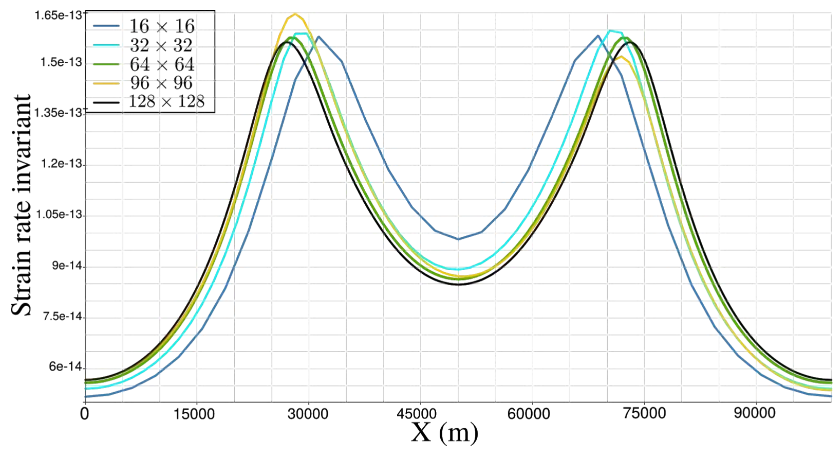


Fig.: Strain rate invariant after diffusing the strain rate components for models with varying mesh resolution.

A plot through the line y=(11/16)*(length along Y) shows the diffused strain rate's thickness for varying mesh resolutions. It can be seen from this plot that besides the coarsest refinement level (16x16), the thickness of the shear bands (assumed by the half-widths) are very similar. The difference can be seen at the resolution of 96x96 because this refinement level is equivalent to refining mesh with three elements four times, which leads to asymmetric peaks and thickness in the shear bands. It should also be noted that the shear band angles change with the resolutions, i.e., curves shift laterally, but the overall thickness at resolutions higher than 16x16 does not vary with increasing refinement level. This is expected as the diffusion method does not lead to correct shear band angles.





Strain rate along y=(11/16)*(total length along Y)

Added plugin for surface velocity residual in the visualization

(Arushi Saxena)

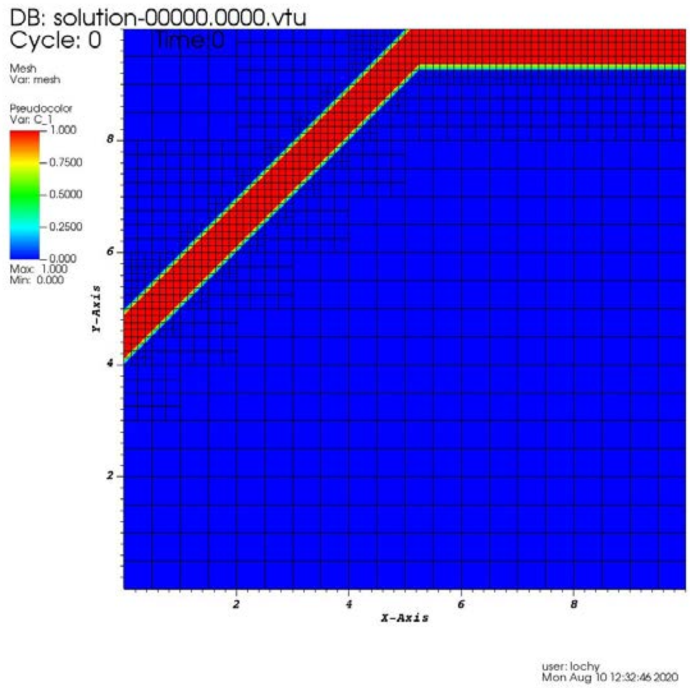
I added a visualization post-processor that generates output showing the residual at the top surface of the model domain. The residual is computed at each point as the difference between the modeled velocities and the input velocities for each vector component. The user has an option to choose the input data as ascii data (e.g, GPS velocities), or a velocity field computed from the GPlates velocity model program.

Added mesh refinement plugin based on isosurfaces

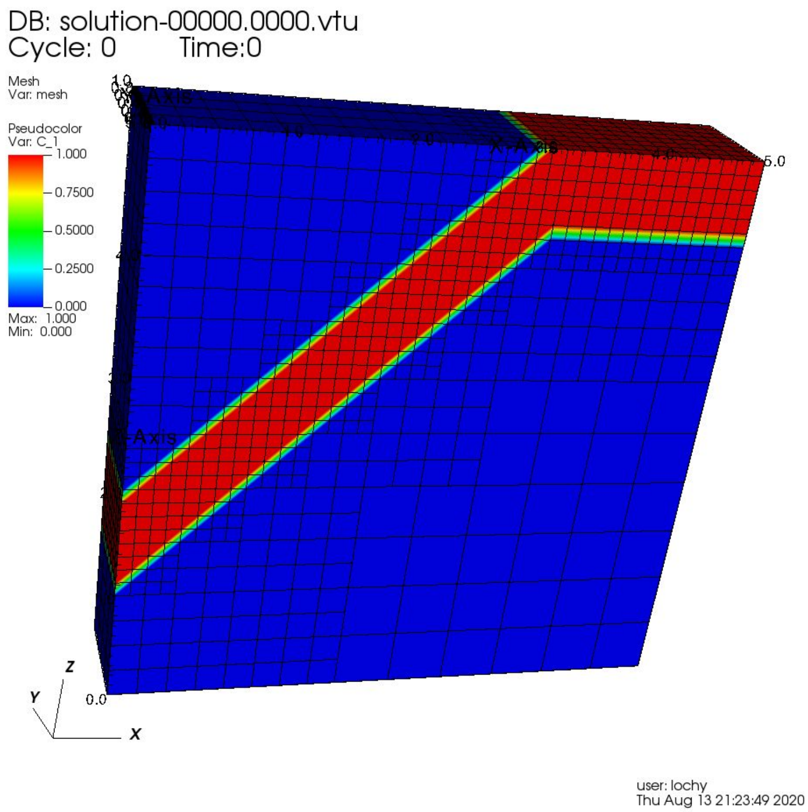
(Menno Fraters and Haoyuan Li)

Menno once had an isotherm plugin where the temperature of points is compared to a lower and upper limit to decide whether it should be refined. We adapted it a little to fit the general need to refine the grid based on the values of fields. This is basically based on the old plugin and allows for entry of temperature, compositions, and a combination of them. In the future, we

will include other fields like viscosity and background composition. But as a start, we keep it small right now.
Here is an example to refine the mesh around a slab using this plugin where the slab is refined to 2 levels higher than ambient. In this way, it works like the minimum refinement function. The difference is that it is based on field value instead of position, and there is also a maximum refinement level. In this way, you could either specify a composition by setting the limits on that composition as 0.5-1.0 or exclude one by setting the limits as 0.0-0.5.



The same can also be done for the following 3D version:



The input files for these cases can be found here:

- https://lhy11009.github.io/LearnAspect/isolines_slab_2D/case.prm
- https://lhy11009.github.io/LearnAspect/isolines_slab_3D/case.prm

Add defect correction Picard methods

(Menno Fraters)

There are now defect correction Picard methods added to ASPECT. They use much of the same code as the Newton solvers, but do not use any derivatives. The same could be accomplished in the past by setting some parameters of the Newton solver correctly, but the new scheme is easier to use and does open up the possibility of making these new solvers the default solvers in the future.

Increased performance of defect correction solvers (Newton and Picard)

(Menno Fraters)

The defect correction and Newton methods have been sped up considerably (~25 percent in some cases). This was mostly achieved by limiting the number of times the preconditioner and other matrices are rebuilt.

Move the world builder into a library linked by ASPECT

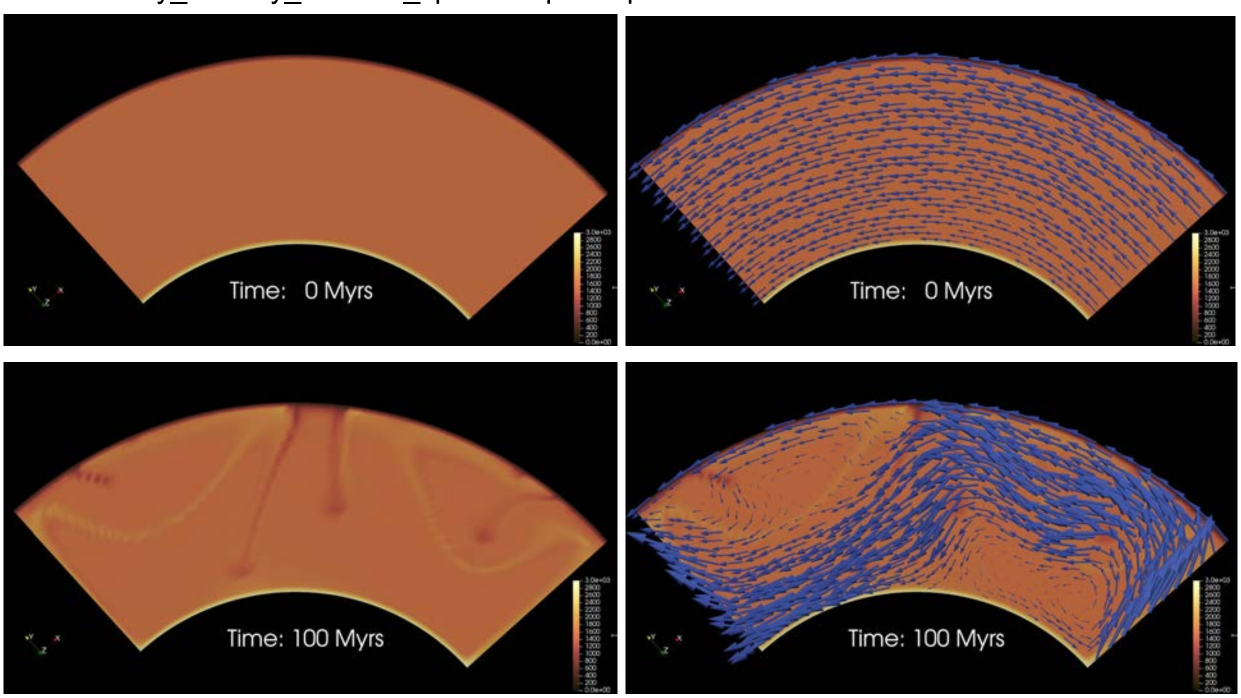
(Menno Fraters)

We worked on changing the building of the Geodynamic World Builder library from ASPECT. Previously, the source files of the World Builder were simply considered part of ASPECT and linked directly into the executable; now, they are combined into a library of their own that ASPECT can then link against. The advantage of this is that the build time can be decreased and that potentially the testers can be coupled. We discussed several options to do this and have proposed an implementation.

Periodic boundary conditions for quarter shell spherical geometry

(Kiran Chotalia, Timo Heister, and Rene Gassmoeller)

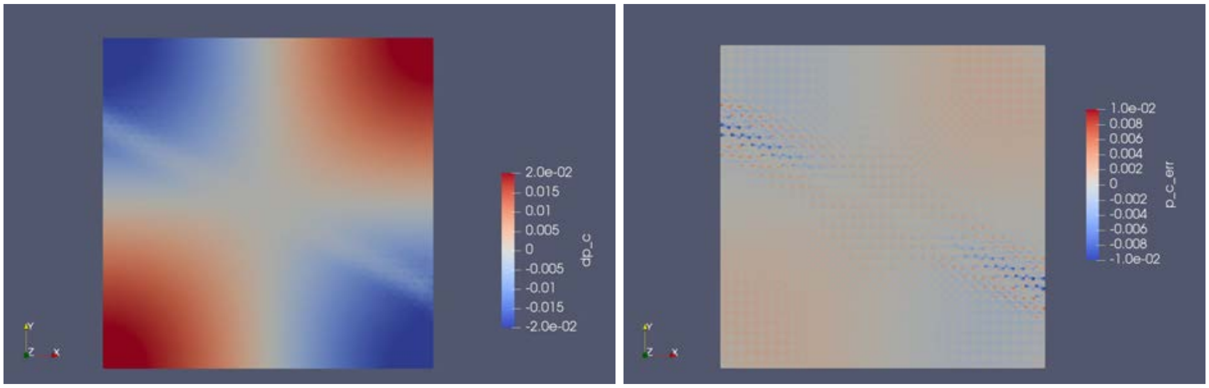
Periodic boundary conditions were previously only implemented for the 'box' geometry. The new 'Phi periodic' option allows periodic boundary conditions for the quarter spherical shell geometry in the phi direction i.e., through the sides of the model. The test case shown below is based on the boundary_velocity_function_spherical.prm input file.



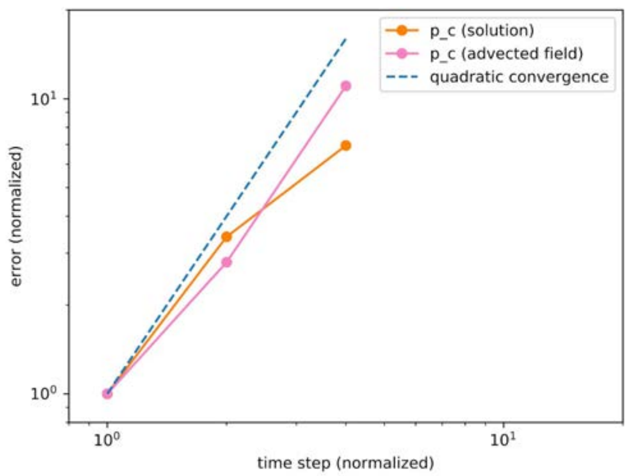
Manufactured solution to test addition of elastic compaction term in melt migration equations

(Fiona Clerc, Juliane Dannberg, Timo Heister)

In order to model the migration of melt through a viscoelastic medium, the compaction pressure is modified by a time-dependent term capturing the elastic response of the solid “de/compressing”. We test the addition of this term using a manufactured time-dependent sinusoidal solution for the compaction pressure, as shown in the figures below (Left: elastic term; Right: errors).



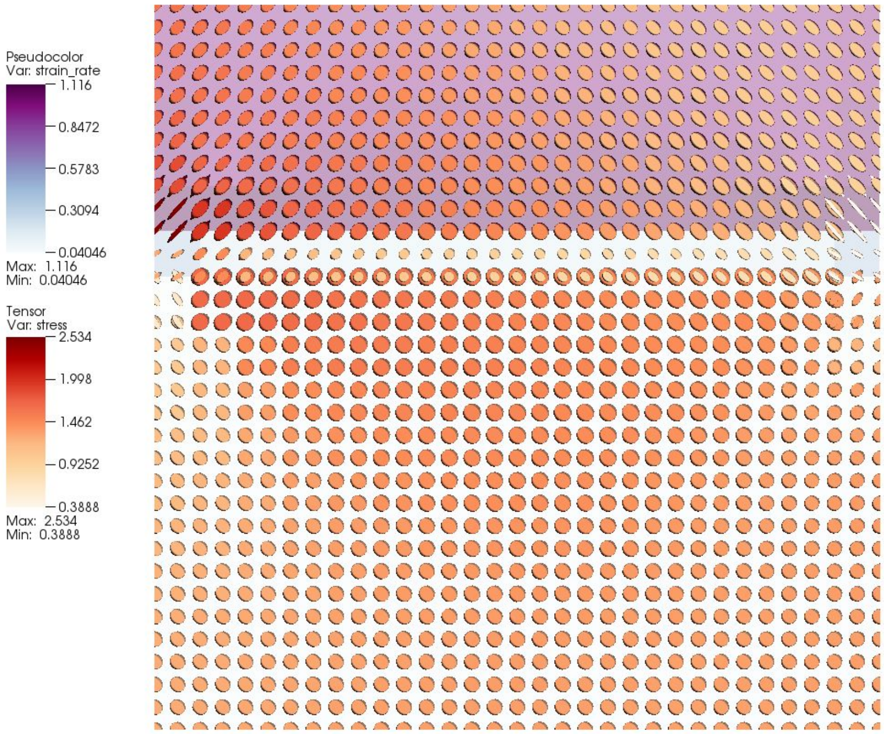
The errors nearly converge quadratically:



Output stress, shear stress, and strain rate as tensors

(Wolfgang Bangerth)

Previously, when selecting “stress”, “shear stress”, and “strain rate tensor” in the list of variables to be output into the visualization files, these quantities were output as 3 or 6 scalar fields representing the independent components of these (symmetric) tensors, but one had to manually re-assemble them into a tensor in Visit or Paraview to actually visualize them as tensors. This has now been fixed, but currently requires the development version of deal.II. With these fixes, one can now generate images such as the following that comes from running the “layered flow” benchmark:



Modernizing some parts of the code base

(Wolfgang Bangerth)

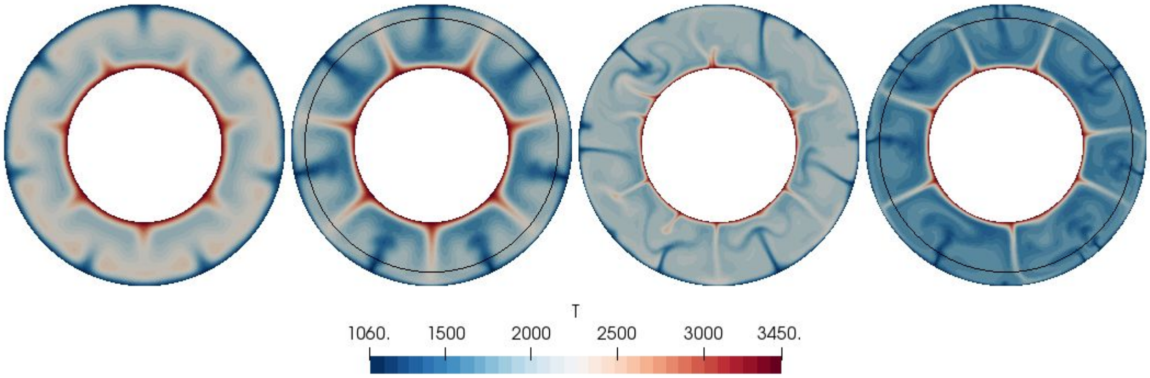
The switch to using C++11 a while ago has enabled the use of some language constructs that are more concise or easier to read (i.e., syntactically different but semantically equivalent). However, by and large few places in our code base have been converted.
During the hackathon, two specific aspects have been addressed: The use of “typedef” declarations has been replaced by the more modern “using” declaration, and a number of places have been converted to using the more concise “range-based for loop” style.

Added Bunge et al., Nature, 1996 mantle convection cookbook

(Cedric Thieulot, Bob Myhill)

We have recreated the setup of the famous 1996 Nature publication by Bunge et al. Although the original article showcases results obtained in a 3D hollow sphere, we here run the models in an annular domain of inner radius Ri = 3480 km and outer radius Ro = 6370 km.
The surface temperature is set to T=1060 K and the bottom temperature to T=3450 K. The gravity vector is radial and its magnitude is g = 10 m/s2. There are four viscosity profiles (two isoviscous, two showcasing a jump at 660 km depth.

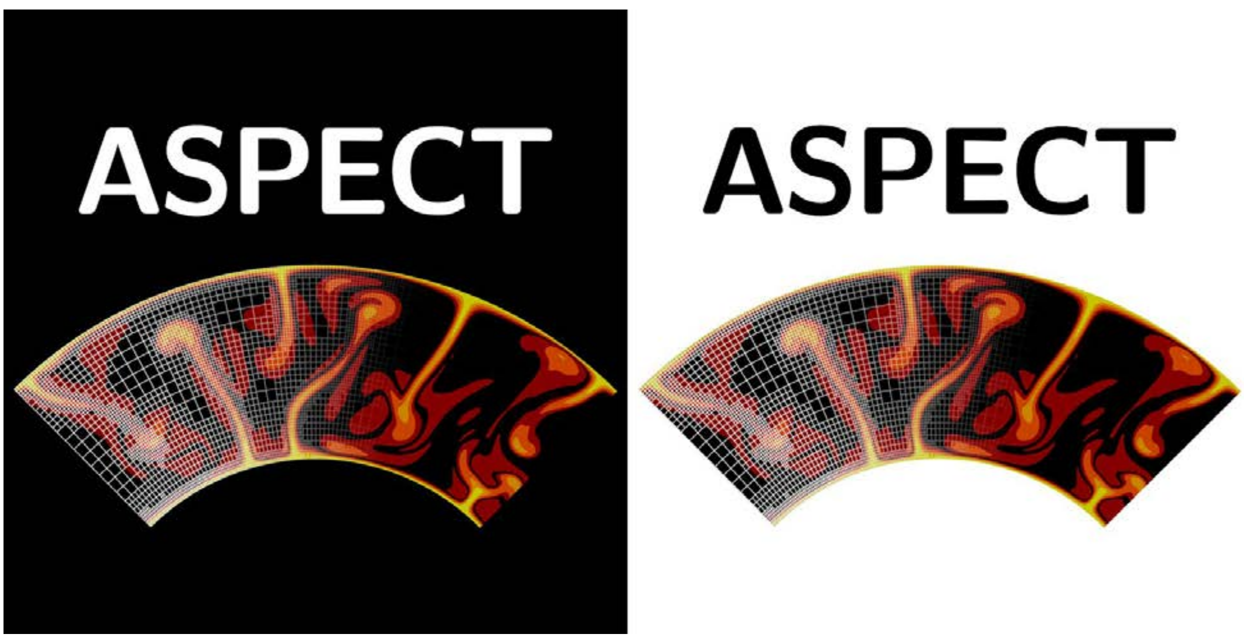
The resulting temperature fields after 5 billion years of convection are shown here under:



Similar to the results obtained by Bunge et al, models in which the lower mantle is more viscous than the upper mantle are distinctly colder than their isoviscous equivalents, with more clearly defined upwellings. You can find a movie of how the temperature evolves over this time period at https://youtu.be/5SPCU1sFGGc.

Created, discussed and decided on the new ASPECT Logo

(Rene Gassmoeller, Juliane Dannberg, Timo Heister)



We concluded the discussion about changing ASPECT's logo to a new, reproducible, more recognizable, and more modern logo. We discussed different variations of our original idea. A general discussion produced a new idea for a fading mesh, which after further experiments was settled on as our new logo, as shown above.

Created a new class to read NetCDF ascii data directly

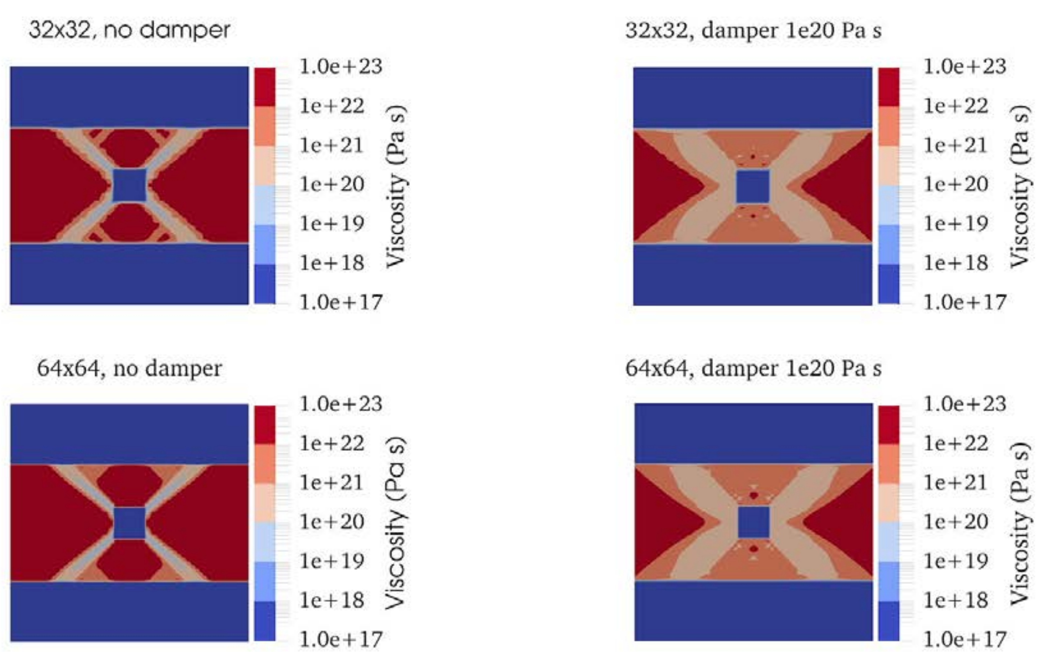
(Kodi Neumiller)

I created a new class, Seismic_pertubation, that will read the ascii values of a NetCDF file from a given URL. Once the ascii data is pulled from the URL it can be passed in to AsciiDataLookup where the data can be used directly. This will make it possible to read NetCDF files in place of .sph files for things such as the S20RTS and S40RTS.

Add a plastic damper stabilization scheme to the DruckerPrager rheology module

(Cedric Thieulot, John Naliboff)

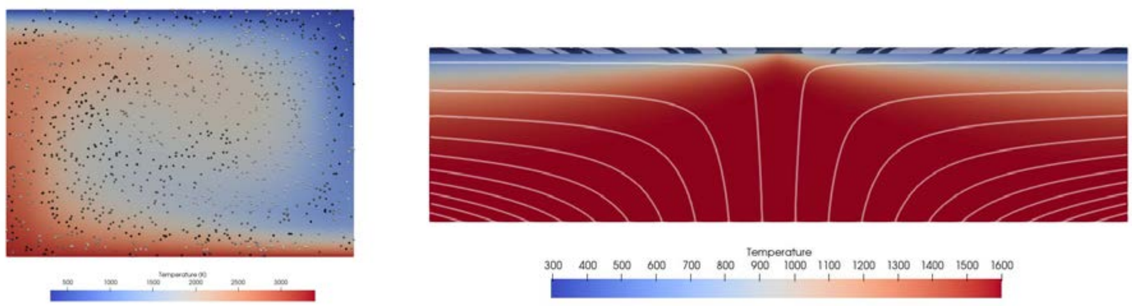
We added and tested a plastic damper to the Drucker Prager yield stress, which provides a length scale for the width of plastic shear bands. The plastic damper is specified by the user as a “damper” viscosity, which is converted to a stress after multiplying the viscosity by the plastic strain rate. A lower damper viscosity produces narrower shear band widths. The figure below shows a comparison between visco plastic models with or without a damper (1e20 Pa s). The resolution is either 32x32 elements or 64x64 elements. The model is undergoing compression and shear bands localize at the corners of a weak viscous inclusion in the model center. The results show that even at coarse resolution, a sufficiently high damper provides mesh-independent shear band patterns whereas the simulations without a damper show a behavior that depends on the mesh size.

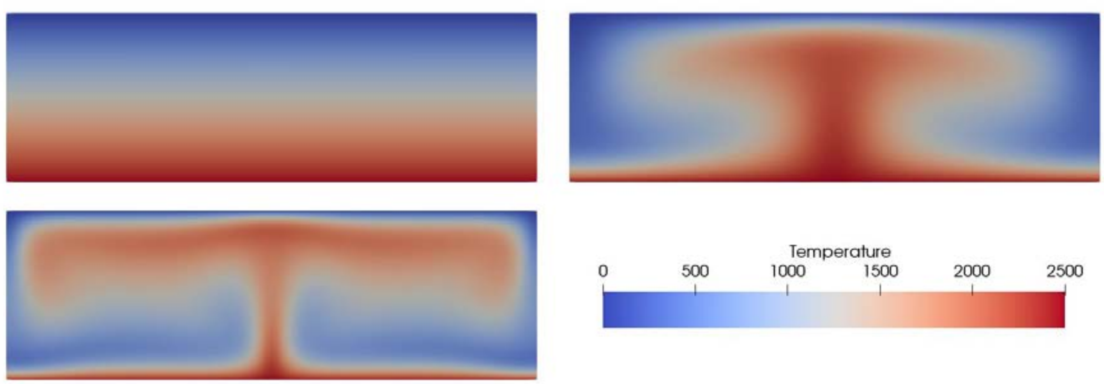


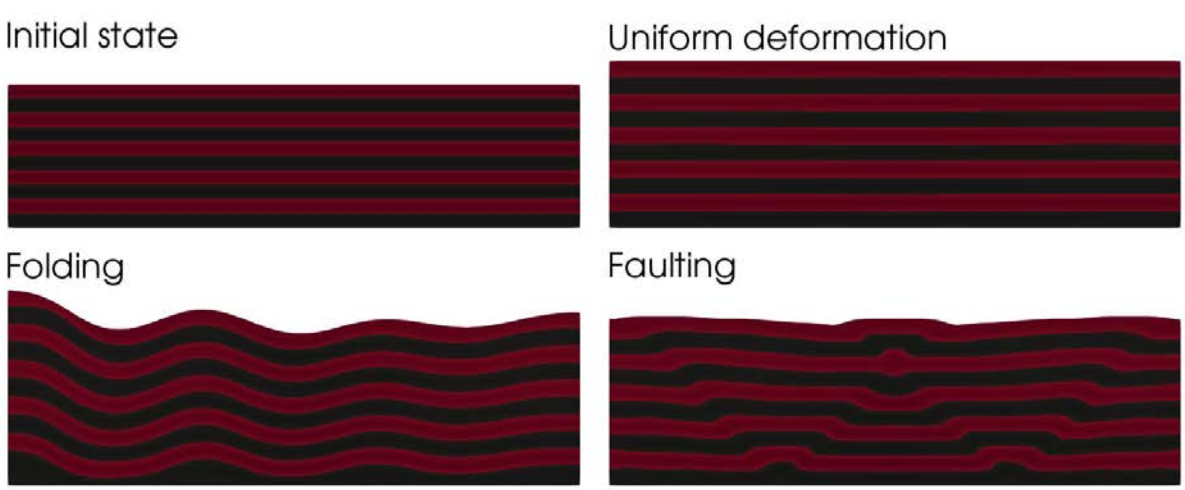
Added teaching materials

(Juliane Dannberg)

I added some cookbooks to the manual that can be used to teach an Introduction to geophysics class using ASPECTC models. Some example setups are shown in the figures below. Specifically, these show: (i) A simple starter model with particles, (ii) a mid-ocean ridge model that illustrates the creation of magnetic “stripes” on the seafloor, (iii) a setup that demonstrates which parameters influence the balance between convection and conduction of heat, and (iv) a model that shows how the rheology controls the mode of deformation in the lithosphere and crust.



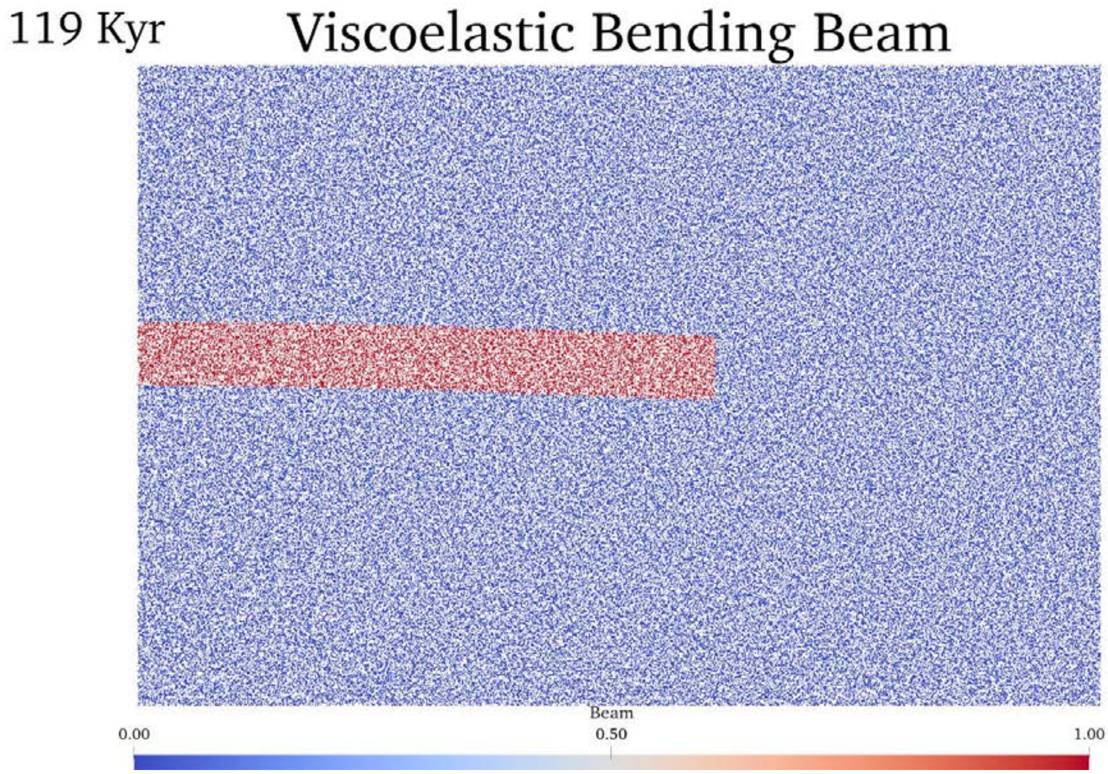




Added the ability to track viscoelastic stresses with particles

(John Naliboff)

I added a new particle property, which tracks viscoelastic stress accumulation on tracers. This functionality uses existing code in the elastic rheology module to calculate viscoelastic stress accumulations, which simply requires passing over the standard MaterialModelInputs at the position of each particle. This implementation has been tested for multiple benchmarks within the ASPECT repository where active particles are used to track viscoelastic stresses and in some cases lithologies. The example below shows the viscoelastic bending beam benchmark, with the beam composition (between 0-1) delineated by particles.

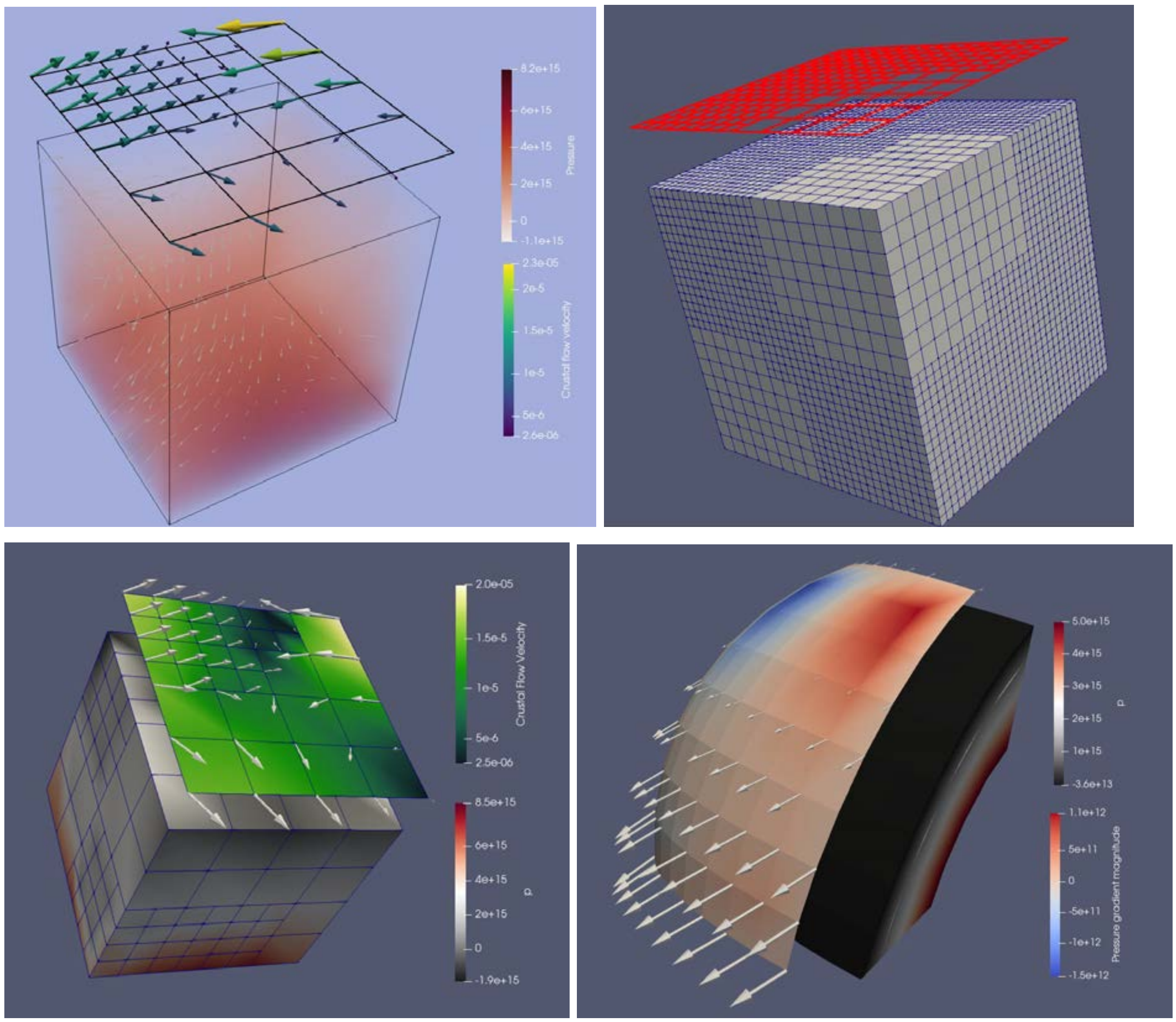


Coupled surface process models

(Jonathan Perry-Houts)

Solving a separate set of differential equations, in a lower spatial dimension, along the boundary of an ASPECT model would be helpful for a number of different applications. I'm currently interested in calculating lower crustal flow driven by mantle-generated pressure gradients. This requires first creating a separate mesh for the surface problem. The surface finite element system will likely require different MPI partitioning than the main model, and therefore communicating values from one domain to the other is computationally challenging. deal.II does not yet support these cases.

During the hackathon, I developed a proof of concept implementation for this technique, using a parallel::shared::Triangulation, for which all MPI processes know about the entire surface mesh, but not necessarily the whole volume mesh. Mapping values between the domains is done by temporarily gathering all surface values on all MPI processes, and selectively discarding values that are not needed. This is sufficient for small problems, but could become prohibitively memory intensive for very large models.

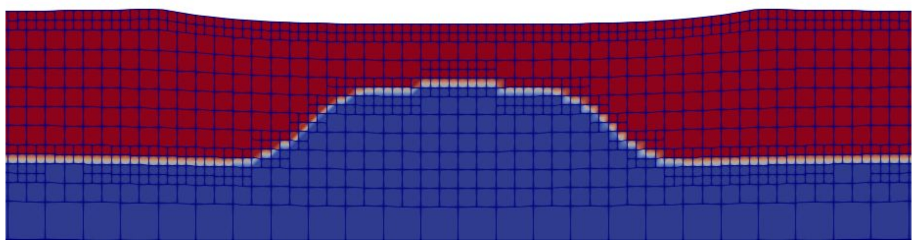


Tested application to impact crater viscoelastic relaxation

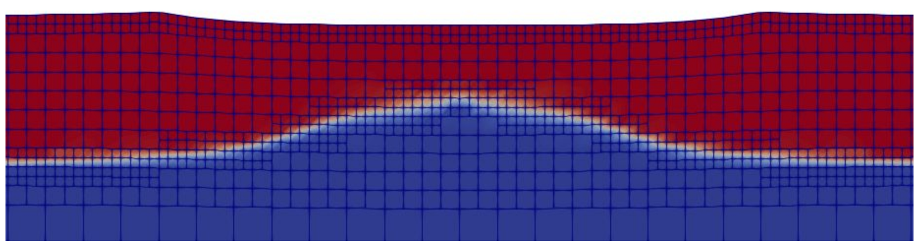
(Robert Citron)

I tested the application of ASPECT to modeling the viscoelastic relaxation of impact basins. ASPECT displayed good performance in modeling the crustal flow beneath the impact basin, with the sharp contrast between the thin basin crust and the thicker surrounding crust being smoothed out over time. Relaxation of impact craters (and other topography) is of wide interest to the planetary science community, and could make the use of ASPECT of interest to that field. Most recent impact crater evolution models decouple the post-impact viscoelastic and thermal evolution, but ASPECT can model the combined evolution of these processes. One issue is that because ASPECT does not have axisymmetric geometry a realistic impact crater relaxation model requires full 3D geometry.

Initial crater:



Relaxed crater:

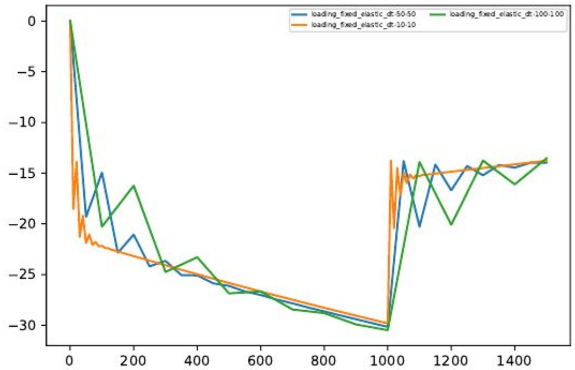


Added comments about elastic and computational time steps in the surface-traction benchmark

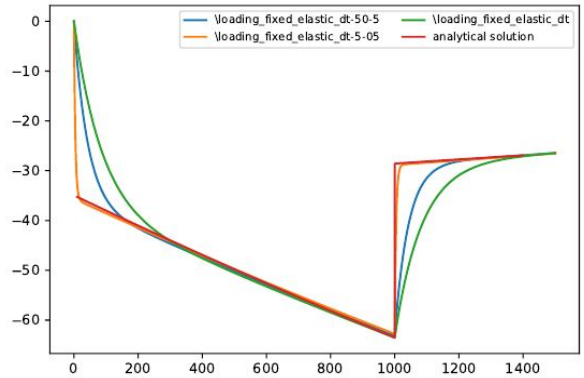
(Esther Heckenbach)

I tested the influence of the size of the elastic and computational time steps for the viscoelastic case of the surface traction benchmark and added my findings to the description.

Having the computational time step equal to the elastic timestep results in oscillating surface heights (part a of the figure below). Having the computational time step smaller than the elastic one smoothes the curve. Decreasing the elastic (and therewith computational) time step size lets the surface motion approach the sudden movement of the analytical solution (part c).



a) $dt_{el} = dt_{comp}$.
green: 100 years
blue: 50 years
orange: 10 years



c) $dt_{el} = 10 \cdot dt_{comp}$.
green: 100 and 10 years
blue: 50 and 5 years
orange: 5 and 0.5 years
red: analytical solution

Computation of edot_ii moved into the *Utilities* namespace

(Esther Heckenbach)

I moved the function to compute the second invariant of the deviatoric strain rate from visco_plastic into utilities, to make visco_plastic more readable.

Starting with a rheology for frictional behavior

(Esther Heckenbach)

This is work in progress. I set up a structure where the friction angle is computed in a separate file outside visco_plastic. The user can choose between different options like dynamic friction (after van Dinther et al., 2013), rate and state friction (after Sobolev and Muldashev, 2017), or non-dependent frictional behavior, as it is used in Aspect until now.

List of hackathon related ASPECT animations

A number of movies and animations were produced at the hackathon. They can be found at the following locations:

Last year's winner:
https://www.youtube.com/watch?v=7fmSfNetG3c&feature=youtu.be

Cedric: My entries
https://youtu.be/0YeyM5mkUas (not so serious)
https://youtu.be/YIN9Dcq31x0 (not so serious)
https://youtu.be/5SPCU1sFGGc (goes with cookbook I submitted today)

Rene: My entry
https://youtu.be/R92wJfKhJ2I

Juliane: My entry
https://youtu.be/KeHNhWLL7ws

Kiran: My entry
https://youtu.be/LrhCCHNU-B4 (most definitely not working)
https://youtu.be/1IMk1xj77uc (working!)

Anne: My not so serious entry
https://youtu.be/EiHppvAAk98

Timo:
https://youtu.be/Yj4zc8wwhMw

Jonathan P-H: Lower crustal flow in a flexurally/isostatically supported channel.
https://drive.google.com/file/d/1Ih4Uu2LoGeRfowuqilwtmopsrBPoHIZQ/view

Robert:
https://youtu.be/F-90h24DH7U (Impact crater relaxation)

Bob:
https://youtu.be/jqN1_oPg5P0 (Prescribed velocities)

Statistics about ASPECT's growth during the hackathon

The following contains a number of statistics about how much ASPECT has grown during the hackathon:

Number of source files in ASPECT before/after:	572 -> 587	+ 15
Lines of code in ASPECT before/after:	142,784 -> 145,228	+ 2,444
Number of merged pull requests before/after:	2596 -> 2739	+ 143
Commits in github before/after:	7873 ->8151	+ 278
Number of tests before/after:	805 -> 831	+ 23

For comparison, these were the statistics for last year's (2019) hackathon:

Number of source files in ASPECT before/after:	520 -> 540	+ 20
Lines of code in ASPECT before/after:	128,121 -> 134,857	+ 6,300
Number of merged pull requests before/after:	2067 -> 2206	+ 139
Commits in github before/after:	6638 -> 7011	+ 373
Number of tests before/after:	681 -> 707	+ 26

The comparison of these numbers suggests that the 2020 hackathon was not quite as productive as the one in the previous year – likely owing to the disruption of running it online and across many time zones rather than having everyone in one place – but in the grand scheme of hackathons we have held over the years, this year was not bad and the number of new source files, pull requests, commits, and tests is all broadly comparable to previous years.

The difference between the second number in the second table (at the end of the 2019 hackathon) and the first number in each column of the first table above it (at the start of the 2020 hackathon) illustrates the level of development that happened over the course of the year between the hackathons. As in previous years, somewhere around one quarter to one third 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