

Review of van Zelst et al. 101 Geodynamic modelling: How to design, carry out, and interpret numerical studies

Boris Kaus, Mainz 1.6.2021

This is a well-written paper that is likely to be very useful for people that are new to geodynamic modelling. Putting all of that together in one place is challenging and I congratulate the authors for doing a good job with it. Unsurprisingly it is rather long and feels more like a book (which partly explains why it took me so long to get the review back to you - apologies!).

This certainly deserved to be published in SE. Yet, I do have a number of comments which I believe would be good to address first. This can mostly be done with some rewriting, so I don't think it should take you a lot of time (yet, it may help to point new geodynamicists in the right direction).

1) Modelling tools

You discuss quite a bit about the open source modelling packages like ASPECT, Underworld or LaMEM. Yet, missing from this discussion are alternative approaches that are based on smaller (sometimes one-page) scripts. Those are not full-blown modelling packages but rather simpler scripts that solve a particular problem very well (and fast). The classical example of that in geodynamics is the MILAMIN code (<http://milamin.org>) which remains one of the fastest codes to solve the incompressible Stokes equations on 2D unstructured meshes, and may be very helpful for those interested to solve for example, problems with viscous inclusions or interacting crystals. Other examples are the M2Di scripts which are concise matlab and julia codes to solve viscoelastic problems, including regularized plasticity, available from <https://bitbucket.org/lraess/m2di/src/master/>. The most recent development in this direction is the ParallelStencil julia package (<https://github.com/omlins/ParallelStencil.jl#stencil-computations-with-math-close-notation>) which comes with many geodynamic examples.

Geodynamicists that are interested to work on technical developments as well, may find such scripts much easier to understand than the big software packages that can do it all. In fact, with the ParallelStencil julia package it is possible to write a very compact code that scales to very large parallel GPU-based supercomputers with almost no effort (provided you use a pseudotransient iterative solver approach). Other efforts (under development) allow calling PETSc and its staggered grid interface from julia. In my opinion such approaches may become increasingly important in the future as it allows PhD students to go from writing an experimental solver to a fully blown (parallel) production code in a rather straightforward manner. This will help to address new multiphysics problems, such as the coupling between reactions & deformation. Given the informative nature of your current paper it would be good if you can discuss these topics as well in your manuscript (and give some of the links above).

2) Parameter sensitivity/controlling parameters

Typical forward models used in many geodynamic applications indeed have a large amount of parameters (as you discuss around lines 116 and lines 845). Usually, such

sensitivity studies are done ‘by hand’ by modifying input parameters, making a model run and checking the difference with respect to a reference model. Yet, part of this can be done automatically by computing scaling exponents which directly show which of the parameters control the velocity at a certain point (as discussed in Reuber et al. 2018 Tectonophysics and used in Reuber et al. 2018, Front. Earth Science). In case adjoint methods are used to compute the gradients this is even computationally extremely efficient, and gives you the sensitivity to all model parameters at the same time. This would go a long way in determining which of those are of first order importance and which are not. This method only gives the sensitivity of the model results for a particular timestep/geometry but for many of the cases we looked at so far this sensitivity did not change drastically during a model simulation making this a quite powerful techniques (provided adjoints are available).

I suppose that the reason that this is not yet more widely applied is that it is not yet implemented in many of the codes currently in use, but I can well imagine that this may change soon. It would be good to highlight this as it is a very useful and automatic way to map and reduce the model parameter space (section 5.2.2) and will help to reduce the number of required simulations and thus the CO2 emissions of a study (line 1455).

3) Section 2.1.1: Mass conservation

Later in the manuscript (and around line 179/180 & lines 203-205) you discuss bulk compressibility, poisson ratio etc., but you don’t show how that should be added to the mass conservation equation in equations (same in line 452). It would clarify matters if you can add this.

4) Solution methods

You discuss different solution methods in section 3.4, but what is completely missing is a discussion about multigrid preconditioners. Users of any 3D geodynamic code will run into having to use multigrid at some point or the other and will wonder why it sometimes does not converge (and sometimes does). As people new to geodynamic modelling are the target audience of this paper, it would be good if you can add a paragraph to discuss this (and why, for many lithosphere dynamics problems, it is important to have a coarse grid that still “feels” the main viscosity structure of the model and is thus not too coarse).

What I believe is also important to discuss are pseudotransient solvers (there is much recent work by Raess et al.) as they result in compact solvers that scale particularly well on GPU’s systems (see link to the ParallelStencil julia package for sample codes).

5) Particle-in-cell methods

I agree that for lithosphere dynamics codes, the particle-in-cell method is the most popular one (and I don’t really understand why you say it is difficult to parallelize). Perhaps you can explain here why that is, which is in my opinion because it is the easiest method to take things like phase transitions, history variables like strain as well as large deformations in a simple manner. Many alternative approaches have been suggested over the years, and many of those are good if dealing with a limited amount of phases (e.g. bubbles interacting with crystals can be well approximated with level

sets). Yet, somehow none of these other methods withstood the test of time, perhaps because they are not general enough.

6) Multiphysics

Multiphysics is indeed an important avenue of future research (perhaps even one of the most important ones in geodynamics). Yet achieving this by “code coupling” of different, unrelated, codes really only works if there is only a loose coupling between the physics (section 3.8). This is perhaps the case when coupling models of surface processes and lithosphere deformation. Other problems, on the other hand, require a much stronger coupling on the solver level which implies that new solvers should be developed. An example are the two phase flow equations that describe magma migration, which roughly consists in a Stokes-like and a Darcy-like problem. Getting the solution in an efficient manner cannot be done by taking a Stokes code and a separate Darcy code and coupling that using batch scripts. Instead, the coupled set of equations needs to be solved in a tightly integrated manner. Our knowledge on the individual systems is still useful, as we can use multigrid preconditioners that work well for Stokes as part of this, for example. With this in mind, PETSc developed the multi-physics framework (see papers by Jed Brown) as well as the recently developed DMStag interface which allows you to add a Darcy-like code to an existing Stokes solver in a straightforward manner. I think it is important that you clarify this here, as there have been too many fruitless attempts in the past already to do loose coupling of different codes (which sounds intuitively easier but has very strong limitations for more tightly coupled problems).

7) Creep rheologies & modelling manuscripts

You spend quite some space describing how to write a modelling paper, which is certainly quite useful for people that are new to the community and/or to the paper-writing business. Yet, after having taught a class for over 10 years in Mainz in which students are supposed to reproduce published geodynamic modelling papers with a different code, I think the two most commonly made mistakes are not described here.

a) The first one is that in many papers, the model parameters are incorrectly listed in the tables, perhaps because the units of say dislocation creep laws are non-intuitive (the prefactor has units of the form of $\text{MPa}^{-(n)} \text{s}^{(-1)}$ where n is the powerlaw exponent; if $n > 1$, transferring this to $\text{Pa}^{-n} \text{s}^{-1}$ takes a bit of work). In addition, creep law experiments are usually given in terms of principal stresses, whereas geodynamic models need to have this in a tensor format, which involves correction factors (see the textbook of Gerya for a nice derivation). All these issues add up to make it quite difficult to fully reproduce the experiments unless the input scripts are attached. In most cases these are obvious typos, as employing the stated parameters results in wide spread drip tectonics, rather than stable subduction (as many generations of students in mainz have experienced), so it seems quite clear that it is actually correctly implemented in the code. Yet, whether the correction factor is taken into account or not is often less clear as it has a smaller effect. To me this points to a deeper underlying problem: there is currently no central database that collects all experimental creep rheology data in a format that is directly implementable in different geodynamic simulations. If we would

have that it would eliminate quite a few of those mistakes.

b) A second issue is that often not all parameters are listed and some info is missing (nor just in terms of the material parameters employed but also with respect to the numerical convergence criteria etc.). I suppose that the reason for this is that there are often so many model parameters that it is quite easy to overlook some. A potential way around that would be to develop scripts that automatically generates tables with model parameters from input scripts which would eliminate another source of mistakes (which is that it seems that often some parameters are forgotten to be described in detail). Ofcourse publishing the input script used to perform the simulations, as you point out, helps as well, but in that case it is still not fully guaranteed that the creep law used in one code are implemented in the same manner in a different code (while using the same correction factors).

I think it would be good if you can discuss these topics in the manuscript. To minimize the chance of mistakes in the future, developing standardized databases and automatized ways to create the parameter tables could be very helpful.

Additional comments

line-wise (mostly typos, with some longer comments)

I. 55: One of the reference textbooks on analytical solutions is Turcotte and Schubert, which would be good to list here. Similarly, the textbook of Neil Ribe (theoretical mantle dynamics) has lots of useful info as well.

I. 170: highly viscous fluid

I. 259: please define deviatoric stress in equations, and not just in words.

I. 263: you did not explain what D/Dt means (or give equations)

I. 269: The shear heating term should only involve the non-recoverable deviatoric strain rate components (that is, non-elastic) and not the full strain rate. I realize that you define strain rate only later in the paper and that for incompressible viscous rheologies this is equivalent. For compressible viscoelastic materials it is however not the same, so it seems important to point that out here.

I. 268: thin phase transitions? Not sure what you mean by that.

I. 349: It would be good if you can give the mathematical definition of how to go from total to deviatoric strain rate (which indeed simplifies to what you write here in the incompressible case).

I. 368: f_{H_2O} looks as if it could be several symbols; perhaps better to write as $f_{\{H_2O\}}$

I. 368; please indicate the units of all variables. In fact that is an issue throughout the manuscript and without specifying the units of parameters you employed, modelling results will not be reproducible and replicable...

I. 368/369: Can you add the units of all parameters?

I. 402: Plasticity can also be used for pressure-independent yield criteria. Examples are Griffith's criteria (tensile failure) or implementing an ultimate yield stress in geodynamic models. It's therefore better to remove 'pressure-dependent'

I. 410: equation 14 is actually not identical to equation 13, unless cohesion C has a different meaning in eq. 14 compared to eq. 13. We actually had a discussion about this before in Solid Earth, so please have a look at <https://se.copernicus.org/articles/11/1333/2020/se-11-1333-2020-discussion.html> to see an illustration of the difference. The reason that cosine and sine terms appear in eq. 13 instead of $\tan(\mu_f)$ is that this is the yield stress function.

I. 416/417: If lithostatic pressure is used to evaluate the yield criteria, the shear band orientation is always 45 degree and there is no difference between compression and extension. I realize that many large-scale convection codes use that assumption, and assuming lithostatic pressure is fine within the mantle. Yet, within the stronger lithosphere there can be quite strong deviations between lithostatic and dynamic P (up to factor 2 under compression). It's probably good to point that out to the readers.

I. 445. An easy-to-follow explanation of how to implement anisotropy in 2D codes is given in Kocher et al. Tectonophysics 421, 71–87.

I. 558: A big difference between FD and FE methods is that in FD, you solve the partial differential equation in a pointwise manner, whereas FE approximate the equations on average per element. It would be useful to add a comment on that here.

I. 562: Perhaps add a small remark on why finite elements should use a mixed formulation with higher order for velocity compared to pressure to get reliable results in the (near)-incompressible limit.

I. 584: 10^{24} Pas

I. 740: the method of manufactured solutions also works for nonlinear problems which is perhaps good to mention here.

I. 745: It is perhaps interesting to mention that community benchmarks is a process that typically takes several years...

I. 963: geomio is spelled "geomIO"

I. 1026: Implicit timestepping, in which advection becomes part of the nonlinear solution step, is an even better method to deal with the drunken sailor effect (was described in Popov & Sobolev 2008, even if only very briefly).

Fig. 10: Maybe it is good to mention that the reason the drunken sailor effect occurs is that the typical density difference between rocks and air is much much larger than the typical density difference within the Earth. Moving the Earth's surface by one meter therefore causes a much larger stress perturbation than moving the Moho by a meter and that is why the models have a tendency to become unstable.

Fig. 10b: Even when you can 'fix' the wrong pressure field in this case by smoothing there are other, more heterogeneous, setups where such smoothing does not fully remove the artefacts. It is thus clearly better to employ LBB stable elements (like $Q_{2P_{-1}}$) throughout, The other undesired side-effect of unstable elements such as Q_{1P_0} is that they require more iterations for a higher resolutions, if combined with iterative solvers. Stable elements fix that.

Fig. 10c: I am not sure why you claim that the higher-resolution model is better here. It seems that both models are performed with non-regularized plasticity so both are actually numerically non-convergent. On one hand this manuscript argues about the importance of reproducible and numerically trustworthy results. On the other hand you show a key example in geodynamics where this is not the case. In my opinion there is no way around using a form of regularized plasticity (together with sufficient resolution such that the plasticity length scale is captured) and it would thus be better to use an example of that within the current manuscript. There are some recent papers by Duretz et al. showing that this can be done, for example by using viscoplastic regularization, so it seems more appropriate to show examples of such computations in this figure.

I. 1047: Or better avoid unstable elements altogether and use stable ones (which also scale better on parallel computers using multigrid solvers)..

I. 1184: probabilistic

I. 1311: In the geosciences it is common practice to acknowledge the work that reviewers put in going through the manuscript and making suggestions. Not everyone in computational geodynamics follows this (unwritten) rule, however, which I know doesn't go well with many of my colleagues (including myself). So perhaps it is good to spell this out here.

Online supplement:

Byerlee's law: Byerlee's law was originally derived for small-scale laboratory samples and is in itself quite amazing in that it shows that the maximum stress that rocks can

withstand is nearly independent on the rock composition. What is even better is that it is nearly perfectly consistent with in-situ stress measurements in drill holes around the world (the classical reference would be Townend and Zoback, 2000). This shows that we can safely upscale Byerlee's law from small samples to the upper crust. For other geodynamically relevant parameters, such as the effective viscosity of rocks such upscaling does not appear to be that easy. It would be good to point that our here (or in the main manuscript).