Response to Interactive comment of Prof. Dr. Kaus (referee) on
"Implementing nonlinear viscoplasticity in ASPECT: benchmarking and applications to 3D subduction modeling" by Anne Glerum et al.

We thank Prof. Dr. Kaus for his detailed review and reproducing one of the benchmarks, which allowed us to greatly improve the manuscript. Below we address his points, and changes in the actual manuscript are indicated in bold.

General remarks:
1. Plasticity implementation
The manuscript mainly deals with several benchmarks that demonstrate that the plasticity implementation gives similar results as that of other geodynamic codes. Yet, the implementation of plasticity itself is not described in very much detail, which is why the title of the paper is somewhat misleading (maybe drop "implementing" from the title.
We have dropped "implementing" from the title.

*The plasticity implementation itself is also rather primitive (using only Picard iterations, for example), and some points deserve more discussion: If you use a viscoplastic rheology, for example, you need an initial guess, which is usually done by taking a viscous-only step. You seem to use a user-specifiable constant viscosity for this, which can change from field to field. This can work (if you tune parameters just about right), but in my experience, viscoplastic models are extremely sensitive to this choice; if you use a too large value of viscosity, it won't converge to a physically sensitive solution because the initial guess for pressure (which is approximately equal to 4*EII*eta where EII is the strain rate and eta the viscosity) may be massively larger than what is*
physically meaningful. A background strain rate of 1e-15 1/s and initial viscosity of 1e24 Pas, for example, would result in a dynamic pressure of 4 GPa. Yet, in the setup of Figure 4, the lithostatic pressure at the base of the model should be on the order of 270 MPa, and for a frictional material (with friction angle 30 degree) the maximum pressure should be no more than about twice lithostatic (see Petrini & Podlachikov, 2001, among others). Thus, the dynamic pressure in this case is significantly larger than the physically admissible pressure, which can cause problems with convergence of the viscoplastic solution. So even while running the same model setup with ASPECT, different users could end up with
totally different results simply because they had a different choice of initial viscosities. Since you implement this in a community code, it is important that you take care that users with less background in computational geodynamics don't produce physically meaningless results (I realize that this cannot be fully excluded, but you can at least try to minimize the chance for this to happen).
One way to do this different is to incrementally increase the boundary velocity (as described in Kaus 2010), during the first timestep. In practice, I found this is non-trivial to implement for more complicated model setups; moreover, it requires a large amount of initial iterations. A technique that is easier and more general (and which I have used since), is to compute the admissible upper and lower bounds of pressure (that are usually a function of the lithostatic pressure), under the assumption of a homogeneous setup for a frictional material under compression & extension. The derivation of the admissible pressures is
given in (Petrini & Podladchikov 2000) for a case with zero cohesion - you can easily extend that to a case with cohesion. During the first iteration step, or the first timestep, a pressure cutoff is applied within the yield function routine, which will ultimately limit the viscosities that the plasticity algorithm gives to reasonable. During subsequent iterations and subsequent timestep, this pressure cutoff is no longer applied. An advantage of this method is that it takes away 'user-tuning' of the initial viscosities. It is implemented in both MVEP2

and LaMEM (both available as open-source on bitbucket). If you don't want to implement this in ASPECT, I would at least appreciate a longer discussion on the choice of reference viscosities and/or background strain rate and how new (or less experienced) users can detect non-sense results.

In ASPECT, during the first iteration step, we have the lithostatic pressure as initial pressure guess. Looking at your pressure cutoff in the yield function routine, this lithostatic pressure will lie within the pressure bounds calculated. Then prescribing the initial strain rate based on the velocity boundary conditions ensures the system starts in a reasonable state with a depth-dependent viscosity. The weak seed subsequently ensures localization of deformation and growing of the shear bands stemming from the seeds towards the surface.

If, on the other hand, we prescribe an initial high viscosity, the pressure resulting from the first nonlinear iteration might indeed be higher in magnitude than the lithostatic one. For example, for the 30 degrees compression case with an initial viscosity of 1e25 Pa s, the biggest pressure obtained after the first iteration is 452 MPa, which is still below the upper pressure limit which for this case would be 610 MPa. Even higher pressures would be 'corrected' by the viscosity cutoff, which will in fact set viscosity at the constant value $\mu_{\max}$ (but not in the seed).

Figure 1 shows the shear bands resulting from 1000 nonlinear iterations but starting from a different initial viscosity for the medium. For low internal friction angles (in this case 10 degrees), there is no difference in the final shear band angle. However, for higher internal angles of friction (30 degrees), there is some variation in shear band angle for both compression and extension, even though the pressures obtained in the first iteration are within the pressure limits. This variation is at most 3 degrees and occurs for those runs only that are less well converged anyway (see Fig. 6 of the manuscript).

To the parameter table of each viscoplastic benchmark we have added a footnote on how an estimate of the initial viscosity or strain rate can be made.

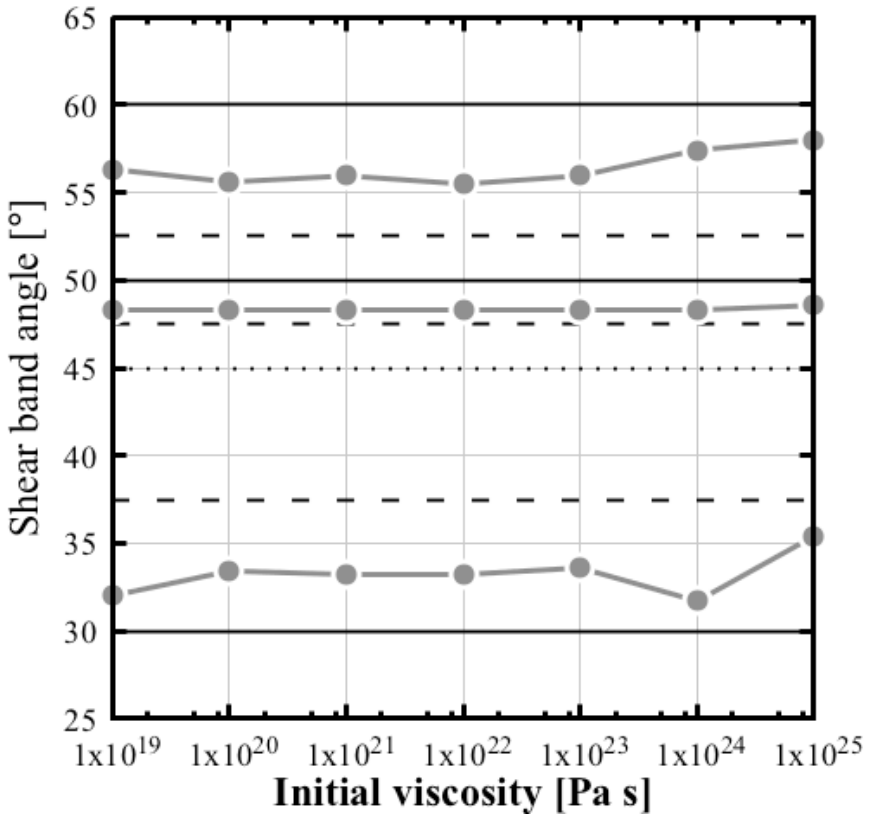

Figure 1 Shear band angles measured for different values of initial viscosity of the viscoplastic medium of the brick benchmark. Elemental resolution is 512x128. Two sets of models are performed in extension: one for an internal friction angle of 10 degrees and one of 30 degrees. In compression, a 30-degree friction angle is used.

2. Appendix A/B: Subduction benchmark:
The benchmark setup you discuss has a 90 degree 'notch' and is the one which gave the worst results in the Schmeling benchmark. This may well be related to the 90 degree initial subduction angle which is very far removed from the angle the slab wants to make. A much better setup is case 3 considered in that paper, for which we also have laboratory experiments and for which the various codes had better agreement. It would be very interesting to see the effect of the viscosity averaging methods for this setup as well (I expect the discrepancies between the models to be much less severe). Can you add that?
To test the effect of the initial subduction angle, we have run a model based on case 3 of Schmeling et al. (2008) with ASPECT 1.5: we kept everything the same, except for the geometry of the slab tip (see adapted Figure 20 of the revised manuscript). The simulations indeed show much less divergence in the slab tip depth evolutions (see Figure 2 below that we also added to the paper). For example, instead of more than 95 My difference in minimum and maximum time for the slab tip to sink 300 km in case 1, the difference is only about 19 My. Trends in evolution with averaging method or mesh refinement level are still the same.

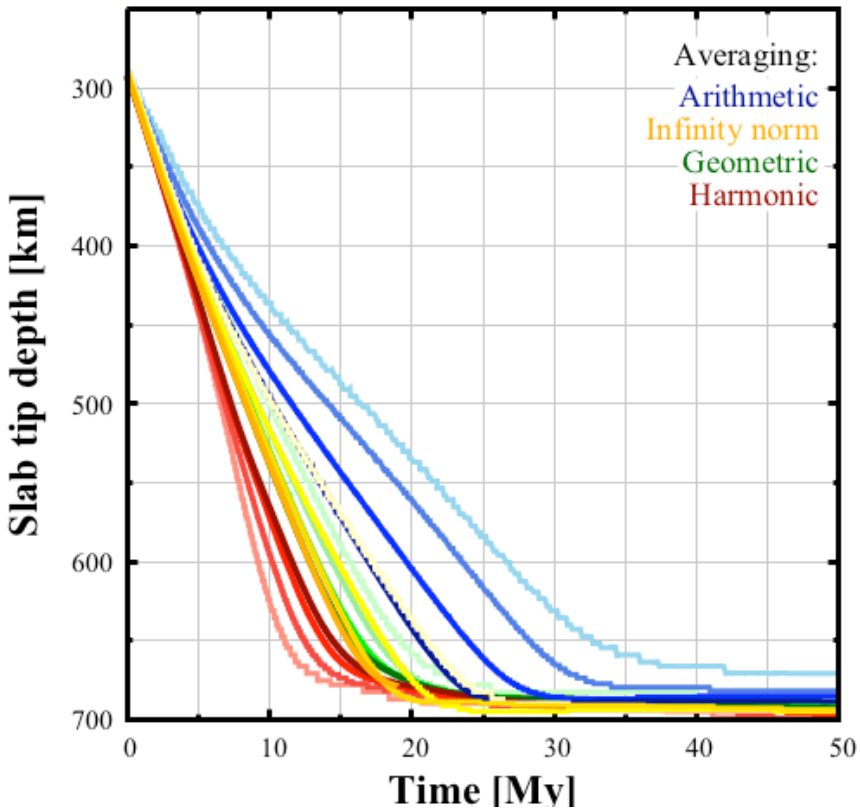

Figure 2 Slab tip depth over time for case 3 of Schmeling et al. (2008) for four different averaging methods of the contribution of the compositional fields to viscosity. Colors indicate the averaging method, while one color goes from light to dark with local resolution, which varies from 256 × 64 elements to 2,048×512 elements. Minimum resolution is always 128×32 elements.

3. 3D Viscoplastic models

Model setups 1 and 2 show that ASPECT can handle more complicated setups. Yet, from a science point of view, the initial geometry of the two cases is so different that it is difficult to discern what the effect of the adjacent plate is. You cite the paper of Schellart and Moresi (2013) in stating that the adjacent plate does not affect the geometry of the subducting plates. Yet on page 23 you also state that your rheology differs from their model. To make your conclusions more robust, it would be good if you can add at least one additional simulation with an adjacent plate for say the simple model setup (model 1). This would support your conclusions that the differences between models 1 and 2 are mainly caused by rheology and not by the adjacent plate. Along similar lines, you mention that the plate viscosity of model 2 is an order of magnitude larger than that in model 1. From Fig. 19, it seems that the asthenosphere also has a lower viscosity, such that the overall slab-mantle viscosity contrast increases. I agree with you that a systematic study of these differences is probably beyond the current paper (even more since each 3D simulation takes several weeks). Yet, what would be interesting is to better understand whether the same effects are observed with 2D simulations as well, using exactly same setup but without adjacent plates. That should be computationally much faster and will at least give the reader some insights into how important 3D effects are for these kinds of subduction scenarios. It would be great

We agree that the initial geometries of the two 3D cases are quite different. Therefore, we have run model 1 with a uniform adjacent plate (AP) and a transform zone (TZ) of the same uniform viscosity as the mantle until the slab tip is draping the bottom boundary of the domain. Both the AP and TZ are of the same thickness as the subducing plate. This model corresponds well with the findings of Schellart and Moresi (2013) in that the geometry of the subducting plate does not change in the presence of an adjacent plate, although in our case the subduction process is somewhat slowed down (TZ width is not given by Schellart and Moresi, 2013; another width could affect the subduction process differently.).

As the reviewer stated, a systematic study of the differences between model 1 and the thermo-mechanically coupled viscoplastic model 2 is beyond this paper. We have run a 2D simulation of model 2 that shows similar slab-mantle viscosity contrasts as the 3D model. Although subduction is faster for the 2D case than for the 3D case, this effect is also seen for model 1.

For more extensive investigations into rheological effects on subduction processes, see for example Andrews and Billen (2009) and Garel et al. (2014). Also, we invite the interested reader to investigate viscosity contrasts and other rheological controls on the subduction evolution with the input files of model 1 with the adjacent plate and the 2D thermo-mechanically coupled model that we have added to the repository belonging to this paper.

We have added the sentence to Section 4.3:
A test with a uniform viscosity adjacent plate for our model 1 corroborates this.

4. Required wall-clock time and Picard vs. Newton iterations
I really appreciate the honesty of the authors by reporting actual wall-clock times of the simulations, which gives interested readers a feeling for the computational costs involved in running ASPECT simulations. To be honest, the results left me a bit shocked. If a 3D free subduction simulation takes up to 6 weeks on 260 processors, with a maximum equivalent resolution of 640x128x128 elements, it essentially implies that it is nearly impossible to perform systematic science with this code (think about the carbon footprint that this simulation has...). It is ofcourse difficult to make a one to one comparison to other codes (also since you did not report the number of timesteps in the tables - would be great if you can add this information). Yet, I'm routinely running 3D models with about the same maximum resolution (but without AMR) with LaMEM in a day or so on 64-128 cores. This thus at least suggests that there are massive differences between different codes and in terms of the science per CPU-hour such difference matter. To get a somewhat better estimate of how timings differ and what the importance of Newton iterations is in this, I redid the 2D slab detachment benchmark (section 3.4) with LaMEM using your largest resolution (256x256 elements) on 8 cores. As LaMEM is currently a 3D only code, the LaMEM simulation was done with 2 elements in the y-direction and employed 3 rather than 2 velocity components per node (such that the total degrees of freedom of the 256x256x2 simulations were with 590'848 slightly larger than the 456'400 DOFs used in the ASPECT simulation). In the LaMEM simulation, I reconstruct the slab thickness from the interpolated phase proportions on the staggered grid. This is likely to be slightly less precise than the marker-line approach used in your manuscript. Nevertheless, results are quite comparable to the

Schmalholz solution: --FIG--
*Rather than taking 28 cores and 16 hours to compute the full simulation, the 295 timesteps of the LaMEM simulation were computed on 8 cores and took about 62 minutes. LaMEM thus seems to require (28*16*60)/(8*62)~55 times less CPU-time than ASPECT (assuming ASPECT employed a similar number of timesteps).*
Part of this discrepancy may be caused by LaMEM employing a Newton nonlinear solver, rather than a Picard iterations. To understand how much this accounts for, I reran the simulation with Picard-only iterations and show the convergence behavior of timestep 10: --FIG—
This demonstrates that Newton iterations are (in this case and for this timestep) around a factor 2 faster (note that we start every timestep with picard iterations before switching to the Newton solver). Evidently, for a tighter tolerance criteria the discrepancy between the two method is larger whereas it is less for a more relaxed nonlinear solver tolerance. Overall, the full simulation with Picard-only took around 153 minutes. So, whereas Newton can explain part of the discrepancy between the required wall-clock time for a full simulation, significant differences remain.
The input files for this setup, together with plotting routines, analysis tools and logfiles of the two simulations described above, are uploaded to the LaMEM repository under /input_models/DetachmentBenchmark.
It would be interesting to see how future optimizations of ASPECT (and of LaMEM) will reduce these timings and how this is in other geodynamic codes. The time-to-do-science is an important factor as well in computational geodynamics, that is unfortunately rarely documented for realistic cases (an exception being Pourhiet et al. 2017 for a different plasticity setup). I thus appreciate reporting these numbers – it would be great if you can report on how the latest ASPECT release affected the timings.
Thank you for taking the time to run the detachment benchmark and provide us with wall time comparisons. The model run originally reported on in the paper was indeed performed with an older version of ASPECT (svn revision 1812) and underlying libraries. Also, time stepping settings were such that 1448 timesteps were taken. We have rerun the model with the latest ASPECT release (version 1.5, https://github.com/geodynamics/aspect/tree/aspect-1.5) and developer deal.II version (commit 1c58789f74fc4c7fd8ec82705ea24aeac8cedf84, https://github.com/dealii/dealii) and time stepping settings to match your 295 timesteps by 289 timesteps, leads to a wall time of 347 minutes on 28 cores. Using a cheaper Stokes solver (with a single V-cycle preconditioner) further reduces the wall time to 235 min (see Figure 3 for necking evolutions). Different compositional field averaging methods reduce wall times even further (this effect was previously shown in Table 10 of the paper), but fail to reproduce the necking curve of Schmalholz (2008). So does averaging of the material properties over each element (for a discussion on this averaging, see Heister et al. 2017). A run on the supercomputer Cartesius (bullx B720 nodes with 2x12 Intel Xeon E5-2690 v3 CPUs, Connect-IB InifiniBand) shows that the broader bandwidth and more modern hardware in general result in a wall time that is about 2.75 times faster. On 8 cores a run then takes 172 minutes, which is similar to the LaMEM wall time of 153 minutes.

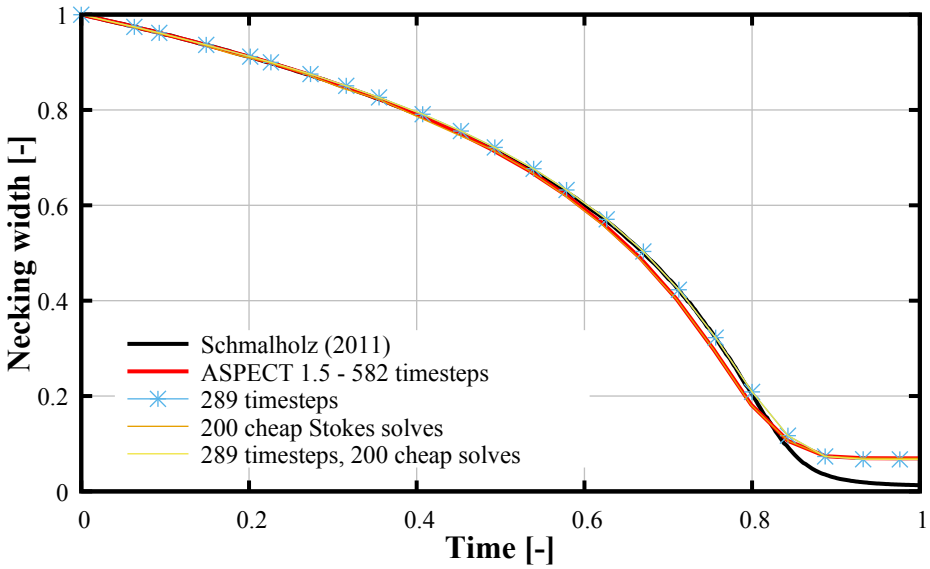

Figure 3 Necking evolution for different model parameters. Note that results for the same number of timesteps are identical.

We now report the necking evolution and wall time for the cheap Stokes solver run on our local cluster (235 min). Note that we have not performed wall time optimization for the other run times reported. Also we've added a section to the overall discussion elaborating on the different parameters affecting and possibly reducing wall time.

To Section 2.1.2 we have added:
A cheap Stokes solve option is available in which the preconditioner employs one V-cycle only. The number of such FGMRES iterations before switching to the more expensive preconditioner is set to 0 in this paper, unless stated otherwise.

To the results of the indentor benchmark (Section 3.1.2) we added:
When using 200 cheap Stokes iterations for the smooth punch, results are not changed, but wall time is about 1.6 times longer. Using harmonic averaging of the material properties as discussed in Heister et al. (2017) increases the velocity error for the smooth punch to ~1%, but reduces wall time about 4.7 times. Loosening the linear Stokes solver tolerance by 1 order of magnitude to 10−8 reduces the wall time of the rough punch by a factor of 1.6, while keeping the velocity error < 1%.

To the overall discussion we added:

Nonlinear rheologies also affect the linear solver by introducing large viscosity gradients. Different strategies to reduce the increased computational time and under/overshooting of the numerical approximation of the viscosity/pressure gradient are available in ASPECT. For one, one can reduce the linear tolerance (while making sure the results do not change significantly), as was shown in Section 3.1. Secondly, a cheap Stokes solver can be employed, although this does not help for each model set-up (compare Sections 3.1 and 3.4). Thirdly, averaging the contributions of the compositional field to the viscosity and other material properties in a

specific point reduces the sharpness of viscosity boundaries, making the problem easier to solve, but with the choice of averaging method affecting the model evolution (Section3.4 30 and Appendix tab:schmelingmodel). Lastly, averaging of material properties such as viscosity and density over each element reduces pressure oscillations (Heister et al., 2017), but can also influence the model evolution as was shown in Section 3.4.

Minor points:

p2. l14. It's LaMEM and the reference is wrong (should be Kaus et al., 2016)
Our apologies, we have changed the capitalization and the bibtex category.

p3. l23. Please clarify whether you employed tracers here or not
We have removed this sentence as we did not employ tracers except for the tracking of the necking evolution of the detachment benchmark.

p4. l6/7: Please clarify whether these are the PETSc SNES options.
They are not, we used ASPECT with Trilinos and not PETSc.

p5. l7: Do you also use a zero initial guess for pressure or a lithostatic value?
The initial guess for pressure considers a lithostatic pressure profile based on the model settings for density and gravity along the center of the domain. This pressure is also used in the computation of the initial residual.

p5. eq(9): As far as I am aware most geodynamic codes employ the same yield-criteria in 2D as in 3D (so eq.8). That has the advantage that if you do pseudo-2D calculations with the 3D code (using say 1 element in the 3rd dimension) you retrieve the 2D formulation. In your case, for typical values of the friction angle (30 degree), the 3D formulation deviates a few percent from the 2D one.
The 3D formulation is a text book Drucker--Prager formulation (e.g. Davis and Selvadurai 2002), a circumscribing cone of the hexagonal cone of the Mohr--Coulomb yield surface. In 2D incompressible plane strain, the Mohr--Coulomb and Drucker--Prager yield surfaces are identical. In case a pseudo--2D model with the same formulation as 2D is required, the user can easily select the formulation in Eq. (8) manually in the material model plugin.

p10. l2: Different than in Kaus (2010), you don't apply strain weakening in your setup. You do mention that later, but a comment at this stage would clarify things already.
We added the following sentence to Section 3.2.1:

Strain softening of the cohesion and angle of internal friction of the medium is not incorporated.

p10. l2 I would also appreciate a brief discussion on the choice of μ_{init} on the model results.
See also our reply to your general remark 1. To the manuscript we added the following in Section 3.2.2:

Varying the initial viscosity of the viscoplastic medium from μ_{min} to μ_{max} for a uniform mesh of 512 × 128 elements leads to the same shear band angles for well-behaved residual runs (see black lines in Fig. 6), while for higher internal angle of friction runs, a variation of maximally 3◦ is found.

p12: You performed these simulations without AMR. What is the effect of using AMR on the shear band angles (if any)?
We have performed several sets of runs both in compression and extension using AMR. Instead of doing 1000 nonlinear iterations on one mesh refinement level, we did 333 iterations on 3 different levels, with the finest level being of the same order as the fixed mesh refinement level. After 333 iterations, mesh refinement is performed based on the norm of the strain rate, viscosity or velocity. Differences in final shear band angles arise from the different areas that are refined for the different strategies and the user-set fractions of cells that should be refined. For uniform meshes, variation in angles with mesh size is already seen, and this variation can be carried on to finer meshes with AMR. The differences in shear band angle are shown in Figure 4: differences can be up to 5 degrees, but the angles remain within the theoretical Arthur--Coulomb range.

We've added a sentence to the results section of the brick benchmark (Section 3.2.2):

To estimate the effect of adaptive mesh refinement on the shear band angles, we ran additional tests with 3 × 333 245 nonlinear iterations at increasing refinement levels, with refinement based on gradients in the velocity, viscosity or strain rate and different fractions of cells that are refined. These simulations indicate a maximal variation of 5◦ degrees compared to results for a uniform mesh of 512 × 128 elements.

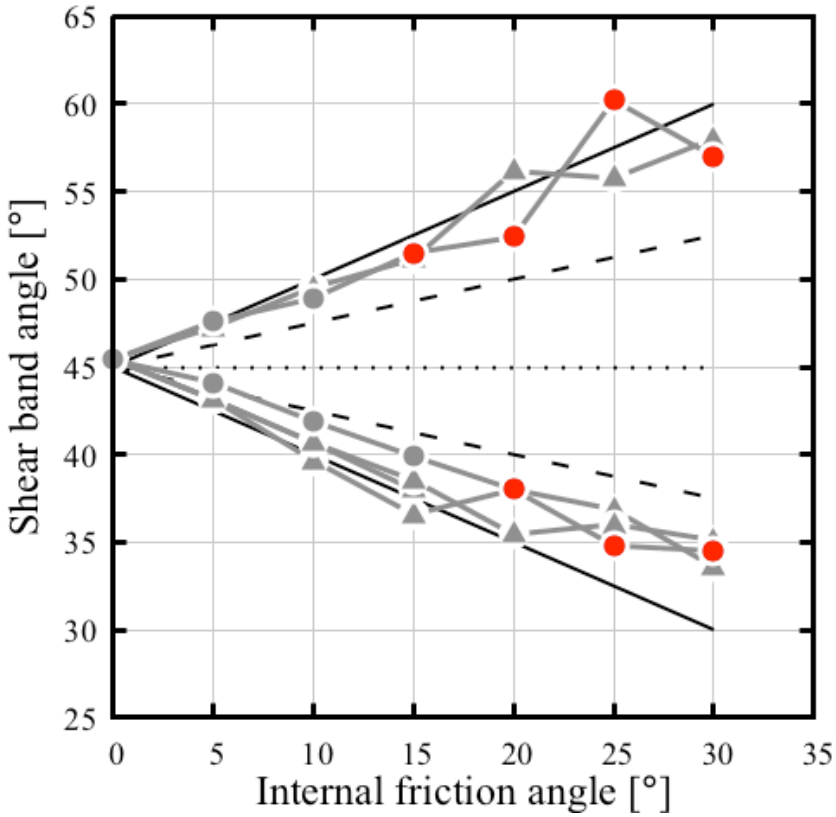

Figure 4 Shear band angles for different internal angles of friction. Round symbols indicate runs with a uniform mesh resolution of 512x128 elements, while triangular symbols represent runs with a base mesh resolution of 128x32 elements and two levels of adaptive mesh refinement, amounting to the same local mesh resolution.

p17: Fig 13: Did you mix the labelling of the x and y-axes of the figure?
Yes. This has been corrected.

p17. l 4: The "first" three benchmarks (as the detachment benchmark is not plastic)
Done.

p24, l21: You mention a benchmark of ASPECT that employs a different viscoplastic formulation. Can you explain better what the difference is? Do they not use a similar yield stress formulation and plastic viscosity?
In response to the first reviewer, we no longer mention this benchmark in the discussion. The viscoplastic rheology used by Tosi et al. (2015) consists of a harmonic average of a temperature- and depth-dependent linearized Arrhenius law and a nonlinear part that is the sum of a constant effective viscosity and a plastic viscosity calculated as $\frac{\sigma_y}{\sqrt{\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}}}$.

p25, l4/5: The first one to show the effect of a nonzero dilation angle in the geodynamics community was, as far as I am aware, the paper of Gerya & Yuen (2007) - see their figure 7.
We added a reference to their paper:

*For example**, Gerya and Yuen (2007) included dilatant materials and** Choi and Petersen (2015) argue that numerical models should incorporate an initially associated plastic flow rule that evolves into a non-associated flow rule with increased slip to assure persistent Coulomb shear band angles while avoiding unlimited dilatation.*

p25, l14/15: In my experience, adding elasticity significantly improves the convergence behavior of simulations with plastic failure (even though it does not solve all issues), and because of that it is worthwhile to incorporate. You are welcome to try MVEP2 or LaMEM to verify this.
See the next point.

p25, l16/17: Newton iterations are crucial for fast convergence - you can add LaMEM and TerraFERMA to the list here. Yet, a pure viscoplastic rheology remains difficult to impossible to converge (as explained by Spiegelman et al., 2016).
We have added LaMEM and TerraFERMA to the list and rephrased the paragraph as follows:

*Incorporating more realistic nonlinear rheologies such as described in this paper creates the necessity for additional nonlinear iterations within a single time step. Also, we have seen that at higher mesh resolutions, more of such iterations are required 15 to converge the solution. This greatly increases model run time and therefore it is important to implement a more efficient nonlinear solving strategy than the Picard iterations currently used by ASPECT. The more **sophisticated** Newton **solver** (see for example Popov and Sobolev, 2008; May et al., 2015; Rudi et al., 2015; Kaus et al., 2016; Wilson et al. 2017) **will help achieving faster convergence.***

Convergence behavior has also been suggested to improve from including elasticity (Kaus 2010), but especially dynamic pressure dependent plasticity remains difficult to converge for both Picard iterations and Newton solvers (Spiegelman et al. 2016).

p26, sect. 6: Can you attach all scripts used to generate the benchmarks and figures to this paper, together with detailed instructions in the exact version of ASPECT you used to create the models? It seems likely that future code changes may give slighty different results; this way the interested reader has a reference point to reproduce your results.
We have put all the input files, scripts to generate them and the necessary plugins to ASPECT1.5 in a GitHub reposity (https://github.com/anne-glerum/paper-aspect-plasticity-subduction-data). This repository also includes the postprocessing and plotting scripts for the graphs.

p27, l7: Why is the infinite norm computationally more expensive? Is that because you effectively end up with larger jumps in viscosity between adjacent elements, and you use iterative rather than direct solvers?
Yes, the maximum norm prohibits any smoothing due to the gradual transition from one composition to the other, leading to larger viscosity jumps between quadrature points, which is harder on the solver.

p39, Table8: In model 1, I am a bit puzzled about the relationship between the B-parameter and the initial viscosity. These models are linearly viscous (apart from the crust), so why is μ_{init} *not simply 1/(2*B) as suggested by eq. 6?*
The prefactor was calculated to mitigate a correction for uniaxiality for pure shear measurements of $\frac{1}{2}3^{\frac{n+1}{2}}B$ that our material model can take into account. However, this correction was not applied and the intended constant viscosities are off by a factor of 1.5 from the initial viscosities.

Additional changes:

The wall time for the indentor benchmark was quoted for the smooth indentor only, which was much smaller than for the rough indentor. We now report the wall time for both with ASPECT 1.5. Also, we changed the measurements of the velocity and pressure in Fig. 3, as it is now possible to extract solution variables at specific points based on the finite element solution instead of through ParaView.

The Stokes solver tolerance of the sandbox experiment was actually 1e-6 instead of the initially reported 1e-5.

References

Andrews, E. R. and Billen, M. I.: Rheologic controls on the dynamics of slab detachment, Tectonophysics, 2009.

Garel, F. et al.: Interaction of subducted slabs with the mantle transition-zone: a regime diagram from 2-D thermo-mechanical models with a mobile trench and an overriding plate, Geochemistry, Geophysics, Geosystems, 2014.

Heister, T., Dannberg, J., Gassmöller, R., and Bangerth, W.: High accuracy mantle convection simulation through modern numerical methods
– II: realistic models and problems, Geophysical Journal International, 210, 833–851.

Kronbichler, M., Heister, T., and Bangerth, W.: High accuracy mantle convection simulation through modern numerical methods, Geophysical Journal International, 191, 12–29, 2012.

Schellart, W. P. and Moresi, L.: A new driving mechanism for backarc extension and backarc shortening through slab sinking induced toroidal and poloidal mantle flow: Results from dynamic subduction models with an overriding plate, Journal of Geophysical Research, 118, 1–28, 2013.

Schmalholz, S. M.: A simple analytical solution for slab detachment, Earth and Planetary Science Letters, 304, 45–54, 2011.

Schmeling, H. A. et al.: A benchmark comparison of spontaneous subduction models-Towards a free surface, Physics of the Earth and Planetary Interiors, 171, 198–223, 2008.

Tosi, N., Stein, C., Noack, L., Hüttig, C., Maierova, P., Samual, H., Davies, D. R., Wilson, C. R., Kramer, S. C., Thieulot, C., Glerum, A., Fraters, M., Spakman, W., Rozel, A., and Tackley, P. J.: A community benchmark for viscoplastic thermal convection in a 2-D square box, Geochemistry, Geophysics, Geosystems, 16, 2175–2196, 2015.