se-20200-79-RC1
Response to Reviewer Comments

Patrick Sanan, for all authors

We deeply thank the reviewer, Marcin Dabrowski, for insightful and thorough comments. We reproduce these comments here and provide our own comments and responses inline. Line numbers in our responses refer to the revised manuscript.

> General comments
>
> The paper addresses problems relevant to the scope of SE and includes some interesting novel concepts regarding the numerical solution of 3D mechanical problems. The scientific methods and assumptions are sound and the presented conclusions are justified. The authors give credit to previous related work and they clearly delineated their contribution. The paper is well written and properly structured, the title is informative and clear, and the abstract provides a good summary of the work. Below I present my specific comments and technical corrections. I would encourage the author to include a more detailed presentation of the studied numerical setups within the main body of the manuscript, according to my detailed suggestions below.

> Specific comments
> For the Taylor-Hood element, the static elasticity in the mixed finite element formulation produces a symmetric indefinite system. It is maybe worth noting that for FEM discretization with piecewise discontinuous pressure field such as in the case of the Crouzeix-Raviart element family, the pressure mass matrix can be easily inverted on the element level and by performing block Guassian elimination a positive definite system can be obtained that allows for using the highly robust sparse Cholesky factorization.

In the discussion of the ineffectiveness of ILU preconditioning (line 289), we have added a citation to Dabrowski et al., 2008 to highlight this approach, which indeed needed to be mentioned.

> The author claimed that the previous incarnations of ILDL' were not necessarily robust(l.93-94). Could the authors just briefly mentioned the major improvements within the recent ILDL' implementation? What improvements exactly have made them robust in the recent years?

This mainly refers to ILDL preconditioning without the weighted-matching step, or just applying ILU or incomplete Cholesky factorizations directly. As in the Chow and Saad 1997 reference cited on line 284, ILU factorizations perform very unreliably. Hagemann and Schenk's 2006 paper presents some experiments which compare the effect of different pivoting strategies.

> What is exactly meant by "coefficient structure" (for example l. 115)? I guess that this is not just the sparsity pattern.

This is intended to refer to the functional form of the coefficients, that is how the material properties vary over the domain. Of interest in the context of solver robustness is whether these coefficients have large global

variation, large local variations, and geometrically simple (roughly, "low frequency") or complex distribution of these variations.

We've further clarified the use of this term on line 113.

> 1x1 and 2x2 blocks are mentioned in the context of pivoting for both LDL' and ILDL'. I am actually wondering whether the natural blocking inherent to the problem due to its dimensionality is retained during this operation? It is stated that fill-reducing reordering is performed block-wise. Which blocks are exactly meant here? I would guess that the ones related to the problem dimensional (say 3x3 blocks in the case of 3D problems). How is it ensure that the blocking due to the symmetric maximum weighted matching preprocessing is retained during the subsequent fill-in reducing reordering? I would suggest that this issue could be clarified in the manuscript.

The natural blocking ($2 + 1$ or $3 + 1$) is not directly retained during the factorization. All blocking and ordering at the level of the preconditioner is done with 2×2 and 1×1 blocks.

This has the advantage of requiring less information from the user, which is extremely desirable in the context of providing widely-applicable and robust methods. However, methods which take the specific saddle point structure of the problem into account do exist; for instance, the paper from Wubs and Thies (in the paper's references) takes into account what they call $\mathcal{F}$-matrix structure.

We have added a footnote (line 233) to emphasize which blocking is being discussed.

> So what is exactly used as the Schur complement preconditioner for large coefficient jumps? The author mention "a scaled pressure mass matrix" in this context. What (viscosity) scaling is exactly used? If there is not enough space for explaining it, maybe the authors could refer to some other work here.

This has been clarified in the text (line 345) to note the exact scaling, $-\left(\frac{1}{\mu} + \frac{1}{\lambda}\right)$. As suggested by Reviewer 2, we have added a reference (line 254) to Grinevich and Olshanskii's 2009 paper, which discusses this preconditioner. Also see the response below to the comments about the C term.

> The authors claim that sparse direct solution methods for indefinite systems using LDL' are expected to be highly competitive for 2D cases (l. 263). Is there any recent study showing their real performance (not just the theoretical scaling) that could be referred here?

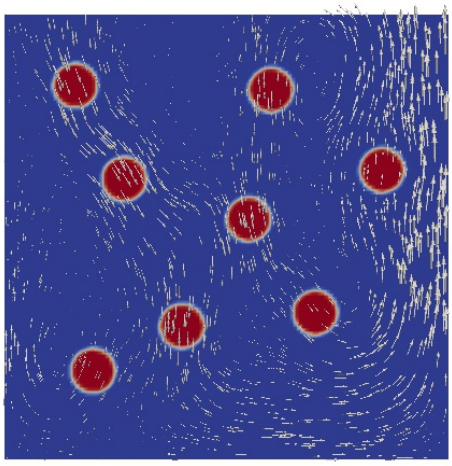
We include a 2D study, from an earlier draft of this paper, which hopefully gives a direct example of this, in Figure 1 of this response.

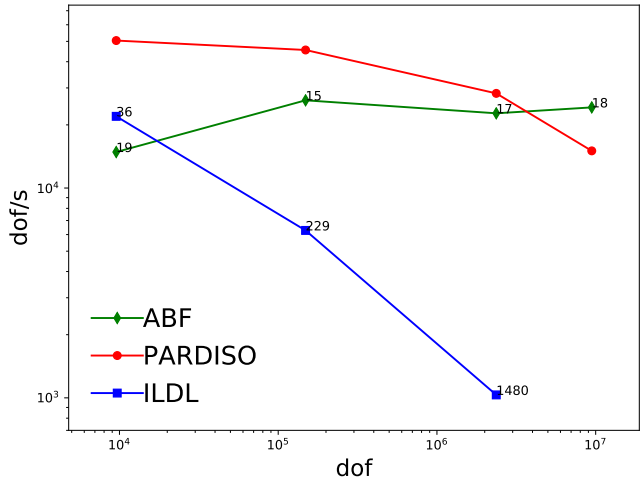
> The authors make statements that variable coefficients on the level of individual elements (non-grid aligned coefficient jumps) are, loosely speaking, harder to solve. In what sense? Solution accuracy or solution time, or maybe both?

With sharply-varying fields, higher order convergence is harder to obtain. In practice, solution times are likely longer, both due to this lower convergence order requiring a finer mesh, or by the fact that multigrid methods appear to converge less quickly.

> I would guess that referring to incomplete factorization preconditioners in l. 283, the authors specifically mean ILDL' rather than ILU or ICHOL, and that they perhaps make this statement in the context of geodynamics or, in general, geosciences.

Yes - this sentence was only meant to refer to incomplete factorization preconditioners for indefinite problems in geosciences (or even more generally in computational mechanics). We have updated it in the discussion around line 284.





	GMRES(60)/ILDL(1e-3)				PARDISO		FGMRES(30)/ABF			
Els.	Fill	Its.	Time [s]	Mem. [MB]	Time [s]	Mem. [MB]	Lvls.	Its.	Time [s]	Mem. [MB]
32^2	2.9	36	4.34E-01	27	1.89E-01	<1024	3	19	6.43E-01	<1024
128^2	4.4	229	2.37E+01	324	3.27E+00	404	5	15	5.68E+00	335.00
512^2	5.7	1480	2.29E+03	5879	8.37E+01	7059	7	17	1.04E+02	5292.00
1024^2					6.28E+02	31266	8	18	3.90E+02	20846.00

Figure 1: Experiments comparing solvers for a 2D stationary Stokes flow problem with dense (1.1×), viscous ($\eta_1 = 10^4, \eta_0 = 1$) inclusions, discretized with $\mathbb{Q}_2 - \mathbb{Q}_1$ finite elements. Extremely short runs do not always report accurate memory usage on the test system.

> The comparative study of Gould (2007) is mentioned in line 292 to justify the choice of PARDISO. I am wondering where there could be any more up-to-date performance studies for sparse direct solvers of symmetric indefinite systems.

Unfortunately, we couldn't find a good, more-recent study (though this would also be of great interest to us). We hope that the older study at least convinces the reader that we have chosen a highly-competitive direct solver, to provide a meaningful comparison with other options. We have slightly modified the statement in the paper (line 300), to imply that we choose it as a competitive solver in general, and that the cited paper offers some concrete, though not complete, evidence of this.

> I totally agree with the authors that the choice of norm is important for matrices characterized by large condition numbers such as in the case of the studied Stokes problem with strongly variable viscosities. In this respect, the authors choose to use the true residual 2-norm rather than the norm induced by the preconditioning. It is perhaps outside the scope of this study, but, in my view, given that the authors have access to highly accurate solutions obtained using the direct solver approach, it would be quite interesting to check and compare the solution error between GMRES(60)/ILDL & FGMRES(30)/ABF, say in the energy norm.

Choosing a norm is usually a compromise, and here we chose one that we thought would be the most representative of something like "actual solver performance", in terms of minimizing a quantity (true residual norm) which most people would agree corresponds well to minimization of the quantity of actual interest. However, as noted in the paper (line 308), in practice a different (quasi-)norm would likely be used, either because of computational expedience or because it better represented the application's idea of an accurate solution. We do think it's probably outside the scope of the paper to present the results in additional norms, but agree that it would be interesting to interrogate and should be, if a practitioner relies on a specific norm.

As a related point here, we have devoted considerable effort to making the experiments here reproducible, with publicly-available (and usable under a BSD-2 license) source code, to help address the problem that users will often want or need to further interrogate the experiments presented. Here, additional norms could be examined by running the application code (with the help of our reproduction supplement), and using PETSc command line options to monitor a different norm or (if the desired norm is not supported), even modifying the C code. While obviously the investment of time and effort to run any code is non-trivial, as readers we very much appreciate the ability to examine source code to answer the common question of how a description in a paper ultimately translates to the implementation, and if interested in extending the method, to be able to directly compare to an existing implementation. As scientific software becomes more and more complex and relies on larger and larger software stacks, the notion that reproducibility (and re-implementability) is implied by a technically-complete description of algorithms becomes less and less valid.

> Do the solve times reported in the tables for the ILDL' preconditioning include the time spent on computing the ILDL' preconditioner? Actually, it would interesting to see how this time compares to the time spent on iterations.

The solve times include both the setup and solve time. The motivation for this is as mentioned on on line 114 - for most of the applications we envision being relevant (3-dimensional nonlinear problems with large-enough memory footprints that direct solvers become problematic), the system is only solved once.

The setup time usually dominates the application time, very roughly requiring $50 - 90\%$ of the total solve time, for the experiments in this paper. If running the included code, one can observe this by using PETSc's logging feature (use `-log_view` as an option), and then can observe the amount of time spent in `PCSetUp`, where the factorization is computed.

We agree that this is under-reported in the paper (especially given the prevalence of reporting these times separately in computational science literature, whether or not is really relevant). Thus, we have added a

note on line 297 and a new column of setup times in Table 1.

> Regarding the numerical setup, I would claim that what really matters is the fraction of the inclusion. With increasing inclusion fraction, as in the case of the setup studied in Fig. 2, a natural transition towards porous media like systems occurs (technically speaking, I am wondering how well 100 inclusions can be resolved using a 32ˆ3 computational mesh). Such physical systems are characterized by strongly localized flows, which might be harder to solve compared to the suspension type of flow typically obtained for low concentration. It would be actually interesting to see how well the presented methods work when the gravity load is replaced by an ambient pressure gradient prescribed through boundary traction.

None of the experiments presented here includes a particularly high volume fraction of inclusions.

The intent of these benchmarks is not to stress the solvers by imposing a complex domain (as for instance would be the case for a high volume fraction of rigid inclusions, with only the interstitial space meshed). Rather, it is to explore the solver performance on a simple domain, but with coefficient structures which vary across non-grid-aligned discontinuities. The multiple sinker benchmark is convenient, but one could also have explored, say, a sinusoidal boundary between two materials, varying the coefficient jump across the boundary, and the frequency of the boundary.

> Technical corrections
>
> l. 11-14 This sentence seems a bit convoluted. I would actually guess that something might be missing here.

A typo (extra "to") has been fixed and this sentence (line 11) has been simplified and split into two.

> l. 22-23...the coefficient structure is made increasingly challenging – I would suggest formulating it more precisely; What "complex topologies" have been addressed in this study?

We have changed this sentence to specifically mention the multiple-inclusion scenario (line 24).

> l. 33 This is maybe not so critical, but compressible quasi-static linear elasticity is not exactly an example of a problem with a divergence free displacement field. In addition, it may indeed represent a saddle point problem, but in some numerical formulations it may be straightforwardly cast as a positive-definite problem.

This was indeed incorrectly expressed - to amend this, we have added a sentence to mention the interesting and computationally-relevant fact that systems requiring divergence-free flow/displacement fields can be seen as limiting cases of systems which penalize volume changes (line 30).

We have chosen to focus on incompressible or nearly-so examples in this work, as outside of this context, there is less motivation to introduce elasticity in mixed form (even at the continuous level).

> l.71...the nonzero entries of the factors are restricted to those for Aˆk – This could be stated a bit more precisely.

Fixed to "A^{k+1}" and wording clarified (line 71).

> l.72-73 I find the end part of this sentence unclear.

The has been reworded to be more concise (line 72), as the point of this passage is simply to point out that only a small number of parameters are required for various variants of ILU preconditioning.

l. 109 In contrast to the previous ILDL studies previous mentioned above...- please remove the second instance of “previous”

Fixed.

l. 134 One could consider using the transpose for one of the vectors in $n * \sigma * n$, etc in eq. 3 & 4.

We opt not to do this, to try to keep the notation uncluttered. We believe it is at least consistent notation, in the sense that $u \cdot v = u^T v$ for two vectors, $u \cdot A = A^T u$ for a vector and a tensor (thought of as a matrix) and similarly $T \cdot u = Tu$.

To make this section more clear, we have added a description of the boundary conditions as a partition of the boundary into free-slip and free surface (zero stress) regions (line 134).

l. 179 Is it really necessary to replace τ with $\mathrm{dev}(\sigma)$ in eq. 3? Given that the t and n vectors are perpendicular ($\langle t, n \rangle = 0$), $t' * \sigma * n = t' * (\tau - p * I) * n = t' * \tau * n - p * tn = t' * \tau * n$

This is true, and indeed this requires no special treatment in our code, so we have removed this statement.

l. 210 permutation (a map from rows to columns) – I would think that the permutation operates within the rows and within the columns, and not from rows to column.

This was confusingly written and we’ve removed the mention of the row-column map (line 209), as the interested reader is better served by reading about the details of the matching in the references. Briefly, the problem of permuting the matrix is re-cast as a matching problem: the matrix is interpreted as a weighted bipartite graph, where entries correspond to edges between rows and columns. The maximum weighted matching gives (for a nonsingular matrix) a subset of the edges such that each row and column is involved exactly once, which can thus be interpreted as a permutation. This subset is sought which maximizes an objective (the product of the entries).

l. 215 If one wishes to find a symmetric permutation, one can only change the order oft he diagonal entries. – If I am getting it right, a symmetric permutation preserves the symmetry of the matrix. I guess that with changing the order of the diagonal entries,the order of the entire rows and columns is also changes (not just the order of the diagonal entries). Anyway, could “non-symmetric” permutations be considered in the current context?

A symmetric permutation of a square matrix M is indeed of the form PMP^T, where P is a permutation matrix, and preserves the symmetry of the matrix while moving entire rows and columns.

We have not considered non-symmetric permutations in this context, but we do not categorically disregard them. The aim of the permutations is to provide a good pivoting strategy, and the current approach seems to scale close to optimally in practice while still retaining system symmetry (thus allowing one to store only L and D, as opposed to two triangular factors, and allowing the use of methods like QMR or MINRES which require symmetric systems). Thus, exploring non-symmetric permutations becomes less appealing.

However, the question of relaxing the symmetry requirements is a very interesting one! On the practical level, this would be highly desirable for applications, for instance finite difference schemes (including finite volume schemes on orthogonal grids) for the Stokes equations, which don’t produce a symmetric system.

Interesting future work could address the usage algorithms which can extend the approaches presented here (for instance, exploring the use of the multi-level ILU as implemented in ILUPACK) to non-symmetric, indefinite systems which arise in computational geosciences.

l. 293-4 Through a custom interface we use PARDISO (Kuzmin et al., 2013) – This looks a bit repetitive with respect to the previous sentence.

Modified to be more concise (line 299).

l. 298 The choice or norm allows is...- Please fix.

Fixed.

Table 1 - I would suggest that the volume fraction of the inclusions could be given. The viscosity is shown without the unit, and this problem could be easily solved by showing the viscosity ratio. Is the relative density dimensionless? Is it defined as $(\rho_{\text{incl}} - \rho_{\text{host}})/\rho_{\text{host}}$? Is it actually relevant given that the model is linear? I would suspect that changing the relative density should only result in a rescaled velocity, and it should, hopefully, produce no appreciable changes to the course of numerical iterations. Is "fill" defined as the ratio between the non-zero entries in the ILDL' factor with respect to the non-zero entries of the original matrix (the triangular part of it, including the diagonal)? Is it necessary to use the scientific notation when time is reported? Maybe giving the total dof count could be useful.

We haven't focused on volume ratio (though several are computed in the remainder of this response), because these ratios are low and we don't believe that this is a factor stressing the solver. We choose the multiple-inclusion problem as a benchmark not because of its direct relevance in application (where higher volume fractions are a key consideration) but because of its usefulness as an abstraction of difficult coefficient structure. See also our response to Reviewer 2's general comments.

However, the presentation wasn't clear enough to make the volume fraction obvious to the reader. As such, we've added the inclusion radius to the captions of Figures 1,3, and 4. This information was also available at the very end of our reproduction supplement, where we have updated the command-line options to include the default (0.1) inclusion radius.

We have changed the caption of Table 1 to mention only the viscosity ratio, and added a column for total DOFs. The relative density is dimensionless, and is defined as $\rho_{\text{incl}}/\rho_{\text{host}}$. Indeed, as this is a linear problem, most simple scalings of any of the parameters have little effect on the solver performance, which motivates the fact that we do not focus heavily on units or scalings in this paper, but on variations in the material coefficients.

"Fill" is defined as the number of nonzeros in the L factor in the LDL^T factorization, relative to the number of nonzeros in the strictly upper-triangular part of the matrix being factored. We have added a note in the paper to make this concrete (line 201). Fill is not reported with scientific notation, though drop tolerance is (which we thought was more readable).

Figure 1 - I would suggest a more detailed description of the numerical setups, both in the caption and in the main body of the manuscript. What is the volume fraction of the inclusions? What is meant by (Vel. scaled 1/3x)? Isn't it that the scaling of the quiver lengths is in no obvious way absolute? I think that it would be useful to show gridlines in the plots. I guess that the dashed line in the Peak Memory Footprint shows the maximum available RAM during the numerical tests, but it would be useful to explain it in the caption. The curve styles are not well visible in the legend. It could also be explicitly explained in the caption that ABF(a), ABF(b),...refer to setups a, b, c...(at a first glance it may look as if it were some variants of the solvers).

The volume fractions of the inclusions can be computed from the inclusion count and radius (which are specified in Section 1.7 in the reproduction supplement).

- Single sinker of radius 0.25 $\implies$ volume fraction of 0.06

- 3 sinkers of radius 0.1 $\implies$ volume fraction of 0.01
- 8 sinkers of radius 0.1 $\implies$ volume fraction of 0.03

These are of course low volume fractions, in the context of problems concerned with interstitial flow (where indeed, other models than a pure Stokes model may be appropriate, e.g. including a Darcy-type term). Our use of the multiple-inclusion problem is motivated by its usefulness as an abstraction of coefficient structure which can affect solver performance, as further discussed in our response to the general comments from the second reviewer.

The captions about the velocity scaling are meant to signify that the first two plots have the same scaling, and the second two a different one, but as pointed out, the absolute scaling of these velocities is not very meaningful, so we have removed these notes to reduce clutter.

Grid lines have been added to all graphs in the paper. The flow plots already have grid lines, though one has to zoom in to see them.

Notes have added to the plots specifying that the dashed black lines are indeed the maximum RAM available.

All legend entries have been modified to hopefully make the line styles more clear. The caption of Figure 1 has been changed to refer directly to the coefficient structures (a)-(d).

> l. 317-8...the ABF solver fails to converge. – It is not clear to me where this can be seen in Fig. 1 (I can't really see any missing data for ABF)

This was intended to mean that when using even more inclusions than are presented in Figure 1 are added, the performance continues to degrade, and thus there are missing ABF entries in Figure 2 (right). This has been clarified (line 326).

> Figure 2 - What is the volume fraction of the inclusions as their number is increase? Given that the numerical resolution is kept constant (32^3) I would guess that it is increased. In my opinion, this should be explicitly stated in the caption and also in the main body of the manuscript. In fig. 1 for 32^3 the overall solver performance in terms of dof/s fell in to the range between 5*10^3 and 10^4, which is consistent with the time reported in table 1. However, in fig. 2, even in the previously studied case of the viscosity ratio of 10^4, the performance is between 10^-2 and 10^-1. I would guess that this could be some technical mistake. In my opinion, it would be useful to show gridlines and maybe use a slightly large font for the legend entries.

The volume fraction indeed increases with the number of inclusions. The inclusion are of radius 0.05 (in the unit cube), and as such for the maximum of 140 inclusions, representing about 7% of the volume. We've added a note to this effect in the caption of Figure 2.

There was indeed an error in our plotting script (an error in the calculation for the total number of DOFs). We have fixed the error and made the y axes uniform between the two sub-plots in Figure 2.

The legends have been increased in size in Figure 2, and grid lines have been added to all graphs in the paper.

> l.324 "...varying to drop tolerance" – Please fix.

Fixed.

> l. 326 System scaling is mentioned in the footnote. Please explain what system scaling(physical, algebraic, ..) is exactly meant here.

This has been clarified (line 333) to refer to a (newly-numbered) equation in Section 3, describing the preprocessing performed before the drop tolerance is applied.

l.339.. and C is the term (depending on λ as in Eq.(9)). – I would guess that the outer brackets are not necessary here. Could the author hint what they actually use for the C term?

The parentheses were indeed a typo, and we agree that the presentation was unclear.

The matrix $-C$ arises from a term in the weak form like $-\int_\Omega q^T \frac{1}{\lambda} p dv$, hence is just another scaled pressure mass matrix, scaled with coefficient $-\frac{1}{\lambda}$, which is ultimately added to the mass matrix scaled with $-\frac{1}{\mu}$, that arises in the same way that the $-\frac{1}{\eta}$ term does for the Stokes problem (as an easy to compute yet spectrally equivalent approximation for the $-B^T K^{-1} B$ term in the Schur complement).

This passage has been reworded in the manuscript (line 343), and more explicit descriptions of the weightings for pressure mass matrices have been added elsewhere (e.g. line 253).

Readers who may be interested in reimplementing the method, or simply wanting to see the direct expression of the formulae in C code, can also examine the source code (at the time of this writing, see `femixedspace.c:2987` at `bitbucket.org/psanan/exsaddle`).

l. 340 Figure 4 shows a similar experiment using a scenario which is perhaps more typical in applications. – Please explain the boundary conditions used in this setup in the main body of the manuscript.

We have added this description and a short note in the initial description of the elasticity problem to highlight that in this case we use inhomogeneous boundary conditions; these are a simple modification of the free-slip conditions, adding terms to the righthand side to specify a given normal displacement as opposed to a zero normal displacement.

Figure 3 – Maybe the Lame parameters μ and λ could be scaled by $\rho * g * L$. A colorbar for the color-coded pressure and gridlines would be a nice addition to this figure.

We have not focused on scaling parameters, as global scalings of these do not affect linear solves, and only relative scalings affects the solvers considered here; indeed, it is one of the great advantages of the ILDL preconditioners, shared with the direct solvers, that they can largely automatically address scaling issues. In each case, the maximum pressure (red) is about 1, and the minimum is a small number (corresponding to a zero pressure at the top, free surface). We have added color bars for the pressure fields, and grid lines for the graphs. There are already grid lines (albeit faint) in the 3d plots.

Figure 4 – It is of small relevance to the studied topic, but the deformed wire mesh implies a substantial deformation that could hardly be accommodated elastically by any geomaterial. But maybe this could be treated as an exaggerated mesh deformation.The elastic moduli are given with no units.

This being a linear problem, it is indeed hopefully still a relevant (and easier to visualize) experiment, even when using an unrealistically-large deformation.

We have not emphasized the units or absolute values of the parameters, as while these are obviously of great importance in actual applications, the applicability and effectiveness of the solvers discussed in this paper are crucially dependent on relative coefficient variability, and essentially invariant to scalings.

Marcin Dabrowski