Dear editor,
we have processed the second round of comments.

Mattew Knepley's review:

> The prior reviews have been comprehensive, and the authors has dealt well with the suggested changes. This paper fills a long-standing gap in the literature, and will be quite helpful to young researchers starting computational investigations of mantle flow. I will make two brief comments.
> First, I think the manuscript would benefit from a fuller explanation of the calculation of pressure convergence. In Fig. 5, higher order convergence of the pressure is reported. If the integral of pressure over the domain is constrained to vanish (as seems to be the case), this constraint must be enforced by the solver. However, the solver will enforce this in the discrete sense, meaning the computed solution will be orthogonal to the constant vector. This is O(h) different from L_2 orthogonality to the constant function, and thus in order to measure higher order convergence of the pressure, we must offset the computed solution by this difference. Carefully explaining this will help readers to reproduce these important results.

That's an entirely legitimate point. We have added a new paragraph at the end of section 3.1 that illustrates the problem, along with the solution:
``We end this section by noting that in many of the set ups we use in Section~\ref{sec:benchmarks}, the boundary conditions we impose lead to a problem in which the pressure is only determined up to an additive constant. The same is then true for the linear system one has to solve after discretization. As a consequence, we can only meaningfully compute quantities such as $\|p-p_h\|_{L_2}$ if both the exact and the numerical solution are \textit{normalized}; a typical normalization is to ensure that their mean values are zero. \aspect{} enforces this normalization after solving the linear system.''

> Second, the concluding remark that the Q_1-P_0 method is more expensive than Q_2-Q_1 is not fully resolved because a quantitative comparison of the increased solver cost for the former with the increased assembly cost for the latter (the goal being solutions of comparable accuracy) is not done. This is beyond the scope of the current paper, but the authors might be interested in a methodology for these kinds of quantitative comparisons: https://arxiv.org/abs/1802.07832

We appreciate the suggestion of comparing other measures of efficiency. In practice, assembly typically costs a relatively fixed fraction of the cost of the linear solver (except in the case of the Q1xP0, where the cost of the solver does not scale linearly with the problem size). As a consequence, we think that no additional insight is possible by such a comparison. Of course, it is conceivable that other codes are faster or slower than ASPECT at assembly, but one might surmise that the difference is a more or less constant factor that does not affect the conclusions we draw.

Susanne Buiter's comment:

>Thank you for the revision of your manuscript. The revised version has been seen by two reviewers who are
both positive (as i am). I would like to ask you to take the suggestions of reviewer #1 into account.
These are only minor and i trust this will not take you much time.
In addition, let me try to answer your query in the response to reviewer #1 of the first submitted version
regarding Fig 5 in section 5.2: i can see both arguments for or against including the FGMRES iterations in
Fig. 5. As it allows a comparison between the two benchmarks (SolCx and Donea & Huerta), it has my preference to
show the number of FGMRES iterations.

We have added Fig. 6 which shows the number of outer FGMRES iterations of the
Stokes solver as a function of the mesh size
and added the following sentences: ``
Fig.~\ref{bench:solcx3} shows the number of outer FGMRES iterations of the Stokes
solver as a function of the mesh size.
We find this time that this number is nearly constant with increasing resolution
for all four elements. Unsurprisingly the
$Q_1\times P_0$ element requires more iterations than all the others but by less
than a factor 2. The quadratic elements
require the same number of iterations while the stabilized $Q_1\times Q_1$
requires only half their number:
this is surprising but the conclusions from the previous paragraph remain about it
being the least accurate of all
four elements here.''

I hereby hope to have adequately answered the comments.

Best regards,
Cedric & wolfgang