Working Document

Test Cases for the Stokes Discretization and the Stokes Solver

Uli Ruede

September 25, 2013

Contents

1 Goals Motivation

1.1 Preliminary remarks

I am still concerned about the basic question *what discretization for the Stokes Solver can give best results?* This starts with asking "what should we consider good results?" What norms are relevant? Are there other quality criteria? Maybe the usual norms are not everything? First, we should primarily be interested in the *absolute errors*, that is the error including the relevant constants, not only their asymptotic behavior. Standard theory often leaves the constants unspecified, therefore we should explore the absolute errors with systematic tests: we need to define benchmarks that are characteristic for the class of problems that we will face. Here I want to propose a class of test problems of this type. I will pose the problem in a square box, but will try to mimic a (sharply) localized upwelling in the center.

1.2 Cost of a discretization

Eventually, we must go beyond the question of accuracy in isolation and must relate the quality of a discretization to its cost. However, techniques for assessing the cost are fundamentally unclear. Certainly we are not only interested in the number of unknowns, but what then should we consider beyond? One easily quantifiable complexity measure is the number of non-zeros in the matrix or –equivalently– the average number of non-zeros per unknown. The number of non-zeros is a first measure for the cost of a matvec operation, but eventually evaluating the cost is much more complex than this.

We cannot only think in terms of multiplying with the (stiffness/mass) matrix, but a realistic complexity measure must account for other operations, such as inter-grid-transfers for our multigrid implementation, scalar products and norms, and, of course, how efficient the solver will be. Here numerical characteristics, such as the rate of convergence will make out one of the relevant effects, but the speed of the implementation is eventually equally important. For example, a high quality discretization may lead to slow converge of our iterative methods, or to an unproportionally high cost for executing smoothing and interpolation. We will see a benefit only if the improved accuracy outweighs the cost.

Additionally, we must not forget that a large contribution to cost comes from the cost for setting up the matrices in the first place. In some cases, a matrix-free implementation may be feasible and this may be essential for constructing efficient algorithms, in other cases this may not be so.

When considered hardware effects, then depending on the data layout, the processing of a grid (typically: executing a matvec) can cost very differently, depending on implementation aspects, such as memory layout, possibilities to use vectorization, etc. We have seen e.g. that even in HHG, the data layout for tetrahedral meshes and the execution order of our smoothers prohibit vectorization. And ultimately, different discretizations may induce different communication requirements when thea problem is distributed across a large machine. We have recently discussed the communication requirements for a Vanka smoother, and have seen that this is not trivial to support. But this was still for simple discretizations, and potentially things become even worse when we move to more complex element types. There are many things to consider, many things are unclear.

1.3 What should we require of a discretization

Coming back to the question about the discretization: One of my concerns is the *resolution* that a scheme can provide with a given number of unknowns. I am taking here a view that is in spirit of signal processing: If the characteristic data and characteristic features of the solution are of a certain size (i.e. there is a highest wavelength that we want to resolve), how many mesh points (and how many elements) do we need to capture these features? I am not aware that this is systematically studied in the mainstream numerical analysis literature. At least our current literature study for Stokes has not pointed me to any such evaluations. The difference is that I ask about the behavior of a scheme in the pre-asymptotic stage. It is unclear to me how this question relates to the known results on the asymptotic behavior and I am unclear what is relevant for the geophysical problems that we plan to study.

For the discretizations under consideration for Terra-Neo I think we should look into this issue, because they are sometimes constructed with different number of mesh points for velocity and pressure. I would like to understand how this affects the ability of the schemes to resolve fine scale features. My starting point is the observation that a P2-P1 discretization for velocity/pressure would resolve the pressure field with only twice the mesh size as the velocity. Here, the velocity is discretely represented by a mesh of width h, but the pressure only on mesh with $2h$. Thus, if the pressure field can resolve features only on scale $2h$, how does this affect the abilities of the velocity field to represent features on scale h?

In the following, I would like to suggest a type of test case that is motivated by the Earth mantle convection scenario - but cooked down to a very simple problem on a square domain. I will suggest a special type of test problem, where we can construct analytical solutions so that we can evaluate the numerical schemes systematically for their (asymptotic) accuracy and also their quality in terms of “fine scale resolution”.

2 Benchmarks for the Stokes Problem

2.1 Construction of Analytical Solutions

We start by fixing the domain $\Omega = (0, \pi)^2$, to simplify working with Fourier-sin and cos-series. Next, we define the potential function

$$\Phi_k = \sin(kx)\sin(y), \tag{2.1}$$

so that for any k a divergence-free velocity field is given by

$$\begin{bmatrix} u_k \\ v_k \end{bmatrix} = \nabla^\perp \Phi_k = \begin{bmatrix} \sin(kx)\cos(y) \\ -\cos(kx)\, k \sin(y) \end{bmatrix}. \tag{2.2}$$

This flow field has zero normal velocities on $\partial\Omega$, but the tangential velocities are nonzero on the boundary, corresponding to special *driven cavity* flow. Clearly, the above flow field will be a solution of the Stokes equation, together with suitable right and side and pressure field. For our benchmark, we want to construct the right hand side such that it is representative of buoyancy forces. Consequently its horizontal component should be zero, only the vertical forces should be nonzero. This can be accomplished when we set the pressure field as

$$p_k = \frac{\left(k^2+1\right)\cos(kx)\cos(y)}{k}. \tag{2.3}$$

Note that for integer k, this pressure field p_k satisfies homogeneous Neumann conditions on $\partial\Omega$. Based on this, we compute the right hand side (i.e. the force field) for the momentum equations as

$$F_k = \begin{bmatrix} f_k^x \\ f_k^y \end{bmatrix} = \begin{bmatrix} -\Delta u_k \\ -\Delta v_k \end{bmatrix} + \nabla p_k = \begin{bmatrix} 0 \\ -\frac{\cos(kx)\sin(y)\left(k^4+2\,k^2+1\right)}{k} \end{bmatrix}. \tag{2.4}$$

With this, we are now prepared to study the behavior of different Stokes discretizations for flows that may be characteristic for buoyancy driven flows in Geophysics.[1]. Note that for $k = 2$, depicted in Fig. 2.1, we can already see a flow that develops kind of a rising plume in the center of the domain, due to uplifting forces that drive the flow.

I would now like to propose that we study the upwelling when the force field is less smooth and when it starts to exhibit more fine scale features. Based on the prototype solution $U_k = (u_k, v_k, p_k)$, we can construct such benchmark problems using superposition for various k, i.e. constructing special benchmark solutions as

$$U = \sum_k \alpha_k U_k, \quad F = \sum_k \alpha_k F_k. \tag{2.5}$$

[1] I am claiming something here that I am ignorant about – Peter, Marcus, and Jens should comment

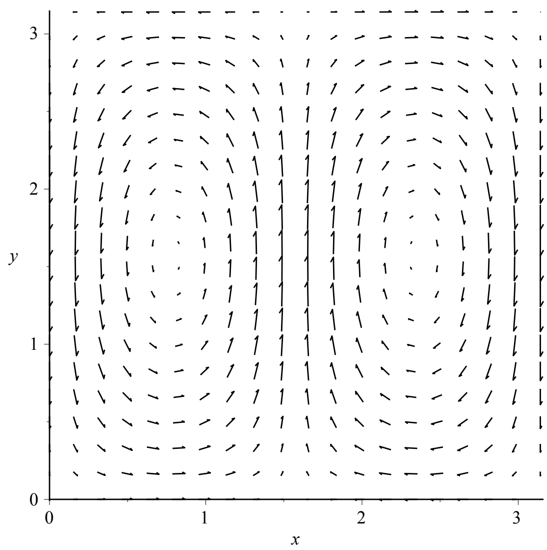

Figure 2.1: Flow field for $k = 2$ — showing two recirculation cells.

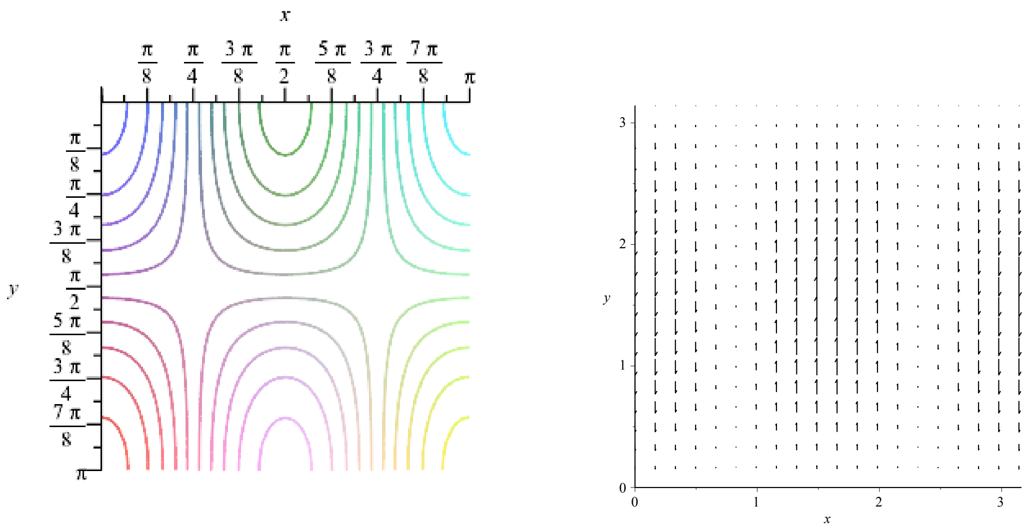

Figure 2.2: Pressure contours and force field for $k = 2$. Forces are only in the vertical direction.

In particular, this class of problems is set up such that the uplifting force f^y behaves as a function of y like a sin-function, but in the x direction we can represent any function as a Fourier-cosine series. In particular, we can even construct a dependence on x as a Dirac-delta distribution, by using its representation as Fourier series. This will then correspond to the uplifting forces concentrated in an infinitesimally narrow channel. Though this may not have relevance directly in our geophysical application, I believe that this limit may be a relevant test case. It allows us to study the ability of our discretizations to represent (Geophysically relevant?) small scale features on a finite mesh size.

Thus, as one particular case, I propose to use

$$\alpha_k = \begin{cases} \frac{k}{k^4+2\,k^2+1} & \text{for } k \text{ even} \\ 0 & \text{for } k \text{ odd,} \end{cases} \tag{2.6}$$

since this leads to a (yet unscaled) discretization of the Dirac function (in the x-direction) for the force. As an example, the first 8 (nonzero) terms become

$$\begin{aligned} u &= -\Big(\frac{2}{25}\sin(2\,x) + \frac{4}{289}\sin(4\,x) - \frac{6}{1369}\sin(6\,x) + \frac{8}{4225}\sin(8\,x) \\ &\quad - \frac{10}{10201}\sin(10\,x) + \frac{12}{21025}\sin(12\,x) - \frac{14}{38809}\sin(14\,x) + \frac{16}{66049}\sin(16\,x)\Big)\cos(y) \\ v &= \Big(\frac{4}{25}\cos(2\,x) - \frac{16}{289}\cos(4\,x) + \frac{36}{1369}\cos(6\,x) - \frac{64}{4225}\cos(8\,x) \\ &\quad + \frac{100}{10201}\cos(10\,x) - \frac{144}{21025}\cos(12\,x) + \frac{196}{38809}\cos(14\,x) - \frac{256}{66049}\cos(16\,x)\Big)\sin(y) \end{aligned} \tag{2.7}$$

This flow field is illustrated in 2.3, where one can see that it looks actually rather similar to the flow field in Fig. 2.1. This, however, is somewhat deceiving, since a closer look exhibits, they the upwelling is more concentrated, and the corresponding recirculation downwards more spread out. The corresponding force field is possibly more illustrative; it given by

$$\begin{aligned} f^x &= 0 \\ f^y &= \cos(2\,x)\sin(y) - \cos(4\,x)\sin(y) + \cos(6\,x)\sin(y) - \cos(8\,x)\sin(y) + \\ &\quad \cos(10\,x)\sin(y) - \cos(12\,x)\sin(y) + \cos(14\,x)\sin(y) - \cos(16\,x)\sin(y)\,, \end{aligned} \tag{2.8}$$

and is illustrated in Fig. 2.4.

Using these functions, we can now explore the quality of discretizations for the Stokes equations systematically, varying the smoothness of the solution (and the generating force field) from moderate k, e.g. $k = 2$, to representing a Dirac-type force field that is constructed analogous to the above expansion.

Based on this, we can explore the *resolution* capabilities of our discretization when driving k to the limits that can be resolved, i.e. when looking at isolated high frequency modes of U_k. That is, I propose to study U_k for high values of k — where high is to be interpreted relative to the mesh size. Generally, this means that $h \approx \pi/k$, but this must be defined appropriately when considering more complex finite elements. From a complexity view point (i.e. when considering storage cost for the solution), the relevant number is the number of DOFs (for each variable, but see also the remarks above). Therefore, for e.g. P2-P1 elements, I would define the complexity-relevant h as only half of the element size for the two velocity components,

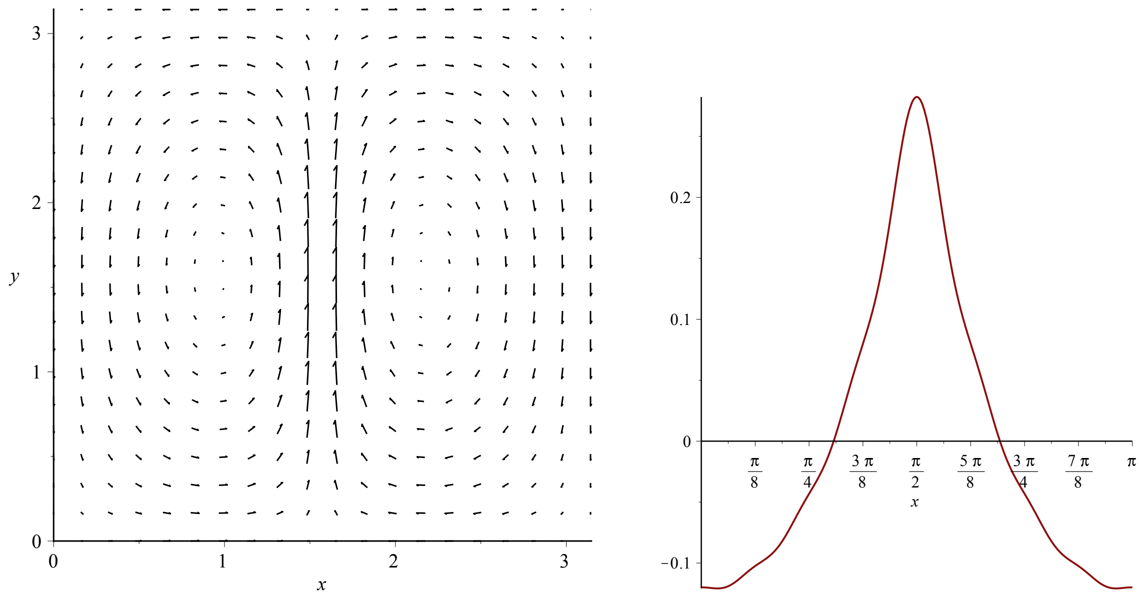

Figure 2.3: Flow field with the the eight terms of eq. (2.7) (left) and vertical velocity plotted along the line $y = \pi/2$ (right).

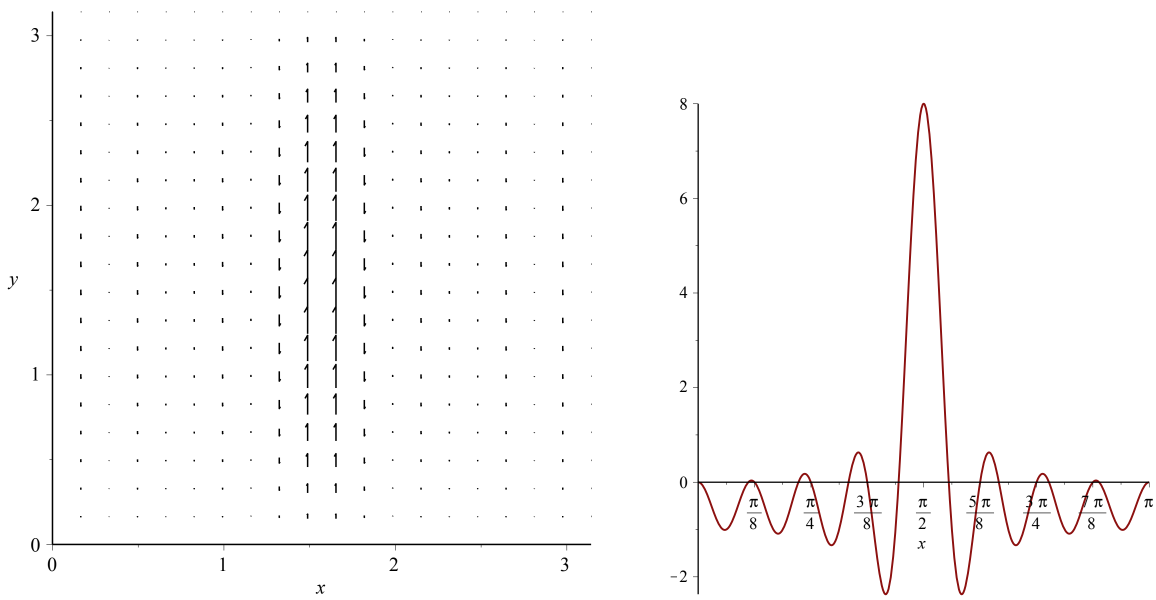

Figure 2.4: Force field for expansion with 8 terms (left) and vertical force plotted for $y = \pi/2$ (right).

and (to remain consistent) $2h$ for pressure. Of course, this is a matter of definition. In the end what counts is cost (for storage and processing) relative to the accuracy that we can obtain. Accuracy here is meant both asymptotically, i.e. keeping one k fixed and considering $h \to 0$, and as measure of the resolution power of the scheme, i.e. keeping h fixed and studying the error when $k \approx \pi/h$.

2.2 Interpretation and implications

The class of special solutions U_k that were constructed above show that we can expect a solution behavior with a special kind of anisotropy for the Mantle convection scenario. This may be useful for constructing problem specifically optimized discretizations. However, in general terms there is more to the decomposition in Fourier modes that can help interpreting and analyzing accuracy issues. This is what I would like to discuss next.

I would like to use this case to argue on the difference between of errors that are caused by

1. issues of fine scale features unresolvable on a finite grid, and
2. asymptotic errors, caused once we are in the asymptotic limit when a full resolution of the problem features is possible

We see that for the force that acts just in vertical direction and that is modeled by a Dirc-function in x (times a smooth function in y), the coefficients of the modes decay as $O(k^{-2})$ for the pressure and for the v-component of the velocity, and as $O(k^{-3})$ for the u-component of the velocity. This translates to the corresponding smoothness classes. But more than that, we can also see the effect of finite resolution on the approximation. The failure of finite size grids to represent the high frequency modes translates immediately into error bounds and a lower bound on the decay of the error. This is the behavior that we expect, as long as we cannot resolve the narrowness of an upwelling channel, and thus must model it mathematically by Dirac-type forces. There is more to say about this, depending on which error measures we use (more local or more global), but that is left to another discussion.

However, once we have reached a resolution that can resolve the width of a channel, we may switch to a different model with a bounded but discontinuous force. Assume that we have a fixed finite value for the vertical force inside the channel and (say) 0 outside the channel. The force is now mathematically not represented by a Dirac function any more, but it has become a combination Heaviside step functions (in x) (times something smooth in y). Consequently the decay of the Fourier modes will be one order faster, and thus the approximation error caused by truncating the Fourier series due to the poor resolution on grid h, will be one order better. Note that we now model the channel with finite width, but that we have not yet included an explicit description of the transition from inside to outside the channel.

If we eventual have resolved the domain so well that we can model the transition from outside the channel to inside the channel by an infinitely smooth transition function, then the Fourier modes will decay exponentially. In this case, the effect of truncating the Fourier series after a finite number of terms will be negligible compared to any polynomial convergence order. In this case, the error will be determined by having "wrong amplitudes" for the low frequency modes. This will give rise to polynomial converge order that our usual FE (or FD) analysis delivers.

In summary, as long as there are features relevant to the solution (caused by corresponding data), that we cannot resolve (here the width of an uprising channel), and that we must therefore model by e.g. Dirac functions or Heaviside jump functions (in the right and sides, i.e. the forces), then it is important that the grids can resolve the high frequency modes on a fine enough grid, since negleging them will lead to significant errors. In our case this is true for velocities and pressure. From this, I think there is reason to believe that we should have the same resolution for pressure and velocity, independent of the polynomial approximation order.

This is different from the situation when we are in the asymptotic regime and can model the forces by smooth functions. Then it is the polynomial order of the approximation that determines the error, because then error is demented by the error we make in large scale modes, and truncating the Fourier series, i.e. ignoring fine scale features has negligible effect.

2.3 Outlook

The representation of solutions to Stokes equation as Fourier series can also be used as starting point to study other interesting aspects of discretization and solver. Classically of course, mode analysis for analysing smoothers and other multigrid components, is based on Fourier series. This has been done in the literature, but I have not yet systematically checked what is out there for Stokes, which discretizations and smoother types have been studied, and what of this is relevant for us.

Beyond the predictive analysis of multigrid components, and as long as we prototypically consider regular grids, we can also try analyze the discrete versions of Stokes using Fourier techniques. I am unsure how far that would carry. For example, I would find it interesting whether one could somehow see/analyze the effect of various stabilization techniques in Fourier space. Note that Brandt (in the Guide) claims that the instability of P1-P1-type elements can simply be removed by suitable post processing. Such aspects might become visible using Fourier techniques.

The limits of Fourier techniques will be reached when we either move to more complex domains and grid structures, or when the operators become more complex, i.e. variable or even jumpy viscosity. Therfore the benchmark and its possible extensions cannot carry us alone through the project. Nevertheless, I believe that we should explore this line of thinking in more detail, to set a firm basis for the techniques that we need to develop. If we cannot solve Stokes as a special case (and possibly as a building block for more complex scenarios) efficiently, it is unlikely that our final solver for more complex models, including variable viscosity, variable density, and variable whatever, will be efficient.