1 Convergence results

ℓ	Dofs $\boldsymbol{u}_h/p_h$	$\Vert\boldsymbol{u}-\boldsymbol{u}_h\Vert_{L^2(\Omega)}$	order	$\Vert\boldsymbol{u}-\boldsymbol{u}_h\Vert_{H^1(\Omega)}$	order	$\Vert p-p_h\Vert_{L^2(\Omega)}$	order
1	375/27	1.1189e-01	0.00	5.1107e-01	0.00	1.3055e-01	0.00
2	2187/125	1.9008e-02	3.02	1.7464e-01	1.83	1.5402e-02	4.18
3	14739/729	2.7787e-03	3.02	5.2700e-02	1.88	2.6362e-03	3.00
4	107811/4913	5.5815e-04	2.42	1.6718e-02	1.73	4.5891e-04	2.75

Table 1: Fluid convergence

ℓ	Dofs $\boldsymbol{b}_h/r_h$	$\Vert\boldsymbol{b}-\boldsymbol{b}_h\Vert_{L^2(\Omega)}$	order	$\Vert\boldsymbol{b}-\boldsymbol{b}_h\Vert_{H(\text{curl},\Omega)}$	order	$\Vert\boldsymbol{r}_h\Vert_{L^2(\Omega)}$
1	98/27	1.8167e-05	0.00	3.9061e-05	0.00	1.9044e-14
2	604/125	1.3020e-05	0.55	2.3989e-05	0.80	2.8631e-08
3	4184/729	7.6478e-06	0.82	1.2832e-05	0.97	7.7598e-10
4	31024/4913	4.0170e-06	0.96	6.4902e-06	1.02	1.3356e-09

Table 2: Magnetic convergence

2 Preconditioning iterations

ℓ	Dofs	it_{NL}	it_{av}
1	527	2	22.0
2	3041	1	7.0
3	20381	1	3.0
4	148661	1	2.0

Table 3: Magnetic convergence

3 Issues

1. From tables 1 and 2 the convergence results seems to approach the expected rates but due to the fact I am having issues getting the larger problems (see the next items) to run I cannot check them. I have tried using a stricter nonlinear tolerance but I get exactly the same rates.

2. One of the main problems at the moment is the calculation of the Stokes initial guess. Due to the Neumann boundary conditions, you are not able to use the mass matrix in the Stokes preconditioner in a standard fashion. Apparently (after chatting with my Dad) for Neumann problems you need to modify the mass matrix to become a “Neumann” mass matrix but I can’t find how I would do this.... Dominik, have you heard of this before?

3. Due to this I am limited with respect to the dimension of the matrices that I can solve. For some reason, when I use kaldi (the CS server) to try and run the next level it seems to stall. I don't really know why this happens....

4. From my understanding, the reason why we are looking at the Hartmann flow is that it is a slightly more realistic example? However, due the exact solution the initial guess seems to solve the problem pretty well so that we only need to do a few nonlinear iterations. Dominik, from what you remember is this what you observed with your implementation of this model?

5. The solution seems fairly simple:

   $$\boldsymbol{u} = (u(y,z),0,0),\ \boldsymbol{b} \approx (0,1,0),\ p \approx -Gx \text{ and } r = 0,$$

   since the magnitude of the Fourier series

   $$|b(y,z)| \approx 1\text{e-}6.$$

   From my experiments, table 3, we see that the preconditioned iterations are very low and maybe this is due the the solution of the problem?

6. For all the above reasons I'm not 100% sure that it is worth pursuing this solution further. At the moment, it seems that the 3D smooth solution is harder for the preconditioner to handle than the 3D Hartmann flow.

4 Hartmann flow solution

In this example, we consider the 3D unidirectional flow in the rectangular duct given by $\Omega = (0, L) \times (-y_0, y_0) \times (-z_0, z_0)$ with $y_0, z_0 \ll L$ under the constant transverse magnetic field $\boldsymbol{b}_D = (0, 1, 0)$. We take $\boldsymbol{f} = \boldsymbol{g} = \boldsymbol{0}$ and consider solutions of the form

$$\begin{aligned} \boldsymbol{u}(x, y, z) &= (u(y, z), 0, 0), & p(x, y, z) &= -Gx + p_0(y, z), \\ \boldsymbol{b}(x, y, z) &= (b(y, z), 1, 0), & r(x, y, z) &\equiv 0. \end{aligned}$$

We enforce the boundary conditions

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{0} && \text{for } y = \pm y_0 \text{ and } z = \pm z_0, \\ (p\underline{\boldsymbol{I}} - \nu \nabla \boldsymbol{u})\boldsymbol{n} &= p_N \boldsymbol{n} && \text{for } x = 0 \text{ and } x = L, \\ \boldsymbol{n} \times \boldsymbol{b} &= \boldsymbol{n} \times \boldsymbol{b}_D && \text{on } \partial\Omega, \\ r &= 0 && \text{on } \partial\Omega, \end{aligned}$$

with $p_N(x, y, z) = -Gx - \frac{\kappa b(y,z)^2}{2} + 10$. The functions $u(y, z)$ and $b(y, z)$ are given by the Fourier series

$$u(y, z) = -\frac{G}{2\nu}(z^2 - z_0^2) + \sum_{n=0}^{\infty} u_n(y) \cos(\lambda_n z) \quad \text{and} \quad b(y, z) = \sum_{n=0}^{\infty} b_n(y) \cos(\lambda_n z),$$

where

$$\begin{aligned} \lambda_n &= \frac{(2n + 1)\pi}{2z_0}, \\ u_n(y) &= A_n \cosh(p_1 y) + B_n \cosh(p_2 y), \\ b_n(y) &= \frac{\nu}{\kappa}\left(A_n \frac{\lambda_n^2 - p_1^2}{p_1} \sinh(p_1 y) + B_n \frac{\lambda_n^2 - p_2^2}{p_2} \sinh(p_2 y)\right), \\ p_{1,2}^2 &= \lambda_n^2 + \mathrm{Ha}^2/2 \pm \mathrm{Ha}\sqrt{\lambda_n^2 + \mathrm{Ha}^2/4}, \\ A_n &= \frac{-p_1(\lambda_n^2 - p_2^2)}{\Delta_n} u_n(y_0) \sinh(p_2 y_0), \\ B_n &= \frac{p_2(\lambda_n^2 - p_1^2)}{\Delta_n} u_n(y_0) \sinh(p_1 y_0), \\ \Delta_n &= p_2(\lambda_n^2 - p_1^2) \sinh(p_1 y_0) \cosh(p_2 y_0) - p_1(\lambda_n^2 - p_2^2) \sinh(p_2 y_0) \cosh(p_1 y_0), \\ u_n(y_0) &= \frac{-2G}{\nu \lambda_n^3 z_0} \sin(\lambda_n z_0). \end{aligned}$$

We note that $p_0(y, z)$ and $-\frac{\kappa b(y,z)^2}{2}$ are exact up to a constant and $p(x, y, z) = p_N(x, y, z)$.

In our experiments, we use $L = 10$, $y_0 = 2$, $z_0 = 1$, $\nu = \kappa = 1$, $\nu_m = 1\mathrm{e}4$ and $G = 0.5$.