Boundary Integral Methods, Assignment 1

Emanuel Ström

September 2021

Exercise 1

We want to show that

$$\int_\Gamma \mu(Q)\frac{\partial}{\partial n_Q}\log|Q-P|\,\mathrm{d}S_Q = \int_\Gamma \mu(\tau)\Im\left\{\frac{\mathrm{d}\tau}{\tau-z}\right\}. \quad (1)$$

We intend to simplify the LHS of (1). First, with some notational abuse we can write the normal derivative on the form

$$\frac{\partial}{\partial n_Q}\log|Q-P|\mathrm{d}S_Q = \nabla\log|Q-P|\cdot\mathrm{d}Q^\perp, \quad (2)$$

where the normal vector $\mathrm{d}Q^\perp = n_Q\mathrm{d}S_Q$ is a 90° rotation of the tangential vector $\mathrm{d}Q$, and ∇ is the gradient operator. First, we calculate the gradient:

$$\nabla\log|Q-P| = \frac{Q-P}{||Q-P||^2}. \quad (3)$$

Putting (2) and (3) together, the LHS of (1) now looks like

$$\int_\Gamma \mu(Q)\frac{(Q-P)\cdot\mathrm{d}Q^\perp}{||Q-P||^2}. \quad (4)$$

Lastly, we transition to complex notation. First, let τ denote the complex representation of Q, and let z denote the complex representation of P. Given that Γ is parameterised counter clockwise, the complex equivalent of rotating $\mathrm{d}Q$ by 90° is to divide by i. Furthermore, it easily verifiable that one may obtain the scalar product of two vectors A and B by taking their complex representations a, b, and calculating the product $\Re\{\overline{a}b\}$, where $\overline{a}$ is the complex conjugate of a. Collecting the results, we get

$$\int_\Gamma \mu(\tau)\Re\left\{\frac{\overline{(\tau-z)}\frac{\mathrm{d}\tau}{i}}{|\tau-z|^2}\right\} = \int_\Gamma \mu(\tau)\Im\left\{\frac{\mathrm{d}\tau}{\tau-z}\right\}, \quad (5)$$

where we used that $|a|^2 = a\overline{a}$ and that $\Re\{\frac{a}{i}\} = \Im\{a\}$ for any $a \in \mathbb{C}$.

Exercise 2

Since $z_p \notin \Omega$, the function $F(z) = \frac{z^2}{z-z_p}$ is analytic in Ω. Hence, F is harmonic. In particular, $U(z) = \Im\{F(z)\}$ is harmonic and satisfies the boundary condition. By uniqueness, U is the solution.

Exercise 3

The numerical error spikes at the boundary. This is likely due to having a finite number of basis functions.

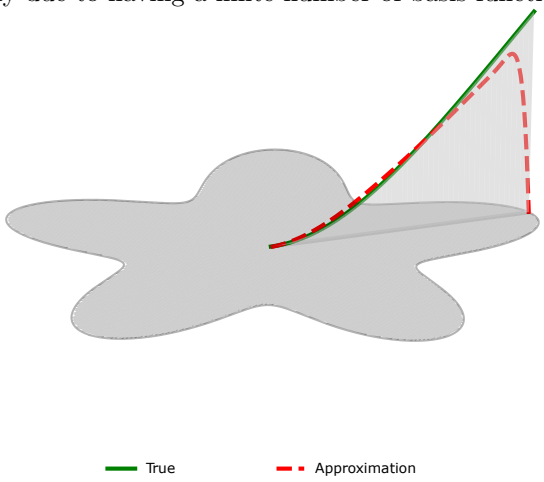

Figure 1: Comparison of the true solution (green) and numerical solution obtained with boundary integral methods (red, dashed). $N = 50$ basis functions were used for this figure.

To illustrate, suppose there is a n such that $\tau(t_n) = 0$, $\tau'(t_n) = ae^{i\phi}$ and $\mu(\tau(t_n)) = \mu_n$. Furthermore, suppose $z = be^{i\theta}$. Then, we have

$$\mu_n \Im\left\{\frac{\tau'(t_n)\Delta t_n}{\tau(t_n) - z}\right\} = \mu_n \frac{a}{b}\sin(\theta - \phi)\Delta t_n. \tag{6}$$

If z approaches $\tau(t_n)$ along a straight line from inside the domain, $\sin(\theta - \phi)$ will be nonzero, and thus the term will tend to infinity. The coarser the discretisation, the further away from the boundary this effect is noticeable.

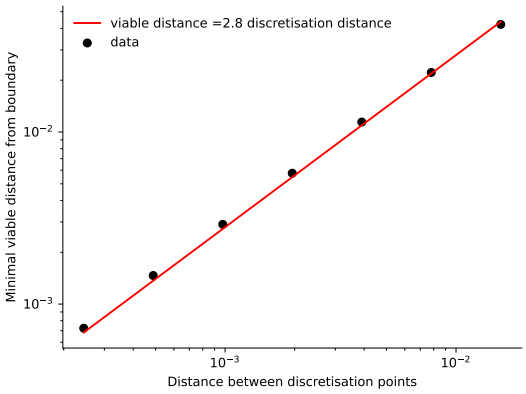

Figure 2: Distance from boundary where the relative error is lower than 0.2, for different discretisation resolutions.

From Figure 2 it is clear that the distance from the boundary where the error is sufficiently low depends linearly on the distance between the basis functions.

Exercise 4

The condition number tends towards a finite limit of 4.6. In contrast, the condition number of second order finite difference systems scales with the number of discretisation points. All but two eigenvalues seem to be real valued. Furthermore, they are all situated near 1 and as the number of basis functions is increased, new eigenvalues appear closer and closer to 1 (As figure 3 shows, the standard deviation of the eigenvalues tends to zero as $1/\sqrt{x}$). According to the introductory text by Ojala, this is important for the GMRES algorithm.

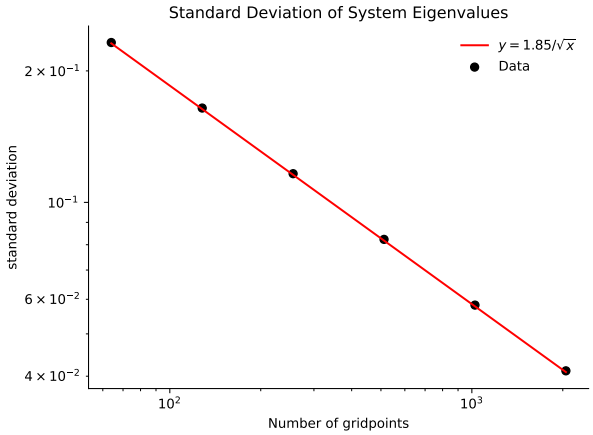

Figure 3: Standard deviation of the eigenvalues as a function of the number of grid points. The mean is close to 1 for all above points.

Exercise 5

Solving the system to full accuracy, the number of iterations scale linearly with the number of discretisation points. If we increase the tolerance to machine epsilon, it seems that the required number of iterations becomes constant. This means that the time complexity of solving the system with GMRES is the same complexity as one step of GMRES. Each step of GMRES requires $\mathcal{O}(N^2)$ evaluations, where N is the number of basis functions. With the fast multipole method, the time complexity should then decrease to $\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$ depending on the implementation.

Exercise 6

By the Leibniz integral rule, we are allowed to move the derivative inside the integral. Let P be on the exterior, we will later take the limit as P tends to the boundary.

$$\frac{\partial}{\partial n_P}\int_\Gamma G(P,Q)\rho(Q)\mathrm{d}S_Q = \int_\Gamma n_P \cdot \nabla_P G(P,Q)\rho(Q)\mathrm{d}S_Q,$$

where the gradient ∇_P is taken with respect to P and n_P is any unit vector. Transitioning to complex notation, we define the complex representation n_z of n_P. Then, with

the same reasoning and notation as exercise 1, we get

$$\frac{\partial U(z)}{\partial n_z} = \frac{1}{\pi}\int_\Gamma \rho(\tau)\Re\left\{\frac{n_z}{\tau - z}\right\}|\mathrm{d}\tau|, \tag{7}$$

where we have absorbed the factor $\frac{1}{2}$ into the density ρ. Lastly, letting z tend to a point on the boundary such that $n_z = \frac{z'}{i|z'|}$ becomes the boundary normal in z, the jump equation combined with the Neumann boundary conditions yield the following integral equation:

$$-\rho(z) + \frac{1}{\pi}\int_\Gamma \rho(\tau)\Im\left\{\frac{z'}{\tau - z}\right\}\frac{|\mathrm{d}\tau|}{|z'|} = f(z). \tag{8}$$

Here, $\hat{z} = \frac{z'}{|z'|}$ is the unit vector tangent to z on the curve Γ. The single layer potential allows us to use the Sokhotski-Plemelj theorem to obtain the jump equation which results in a well-conditioned system. I suspect that using double layer potential would not result in a system that is as well-conditioned.

Exercise 7-8

Since boundary integral methods are easy to adapt to different boundaries, let's choose something a bit more unconventional and cool than the standard square:

$$\tau(t) = \sin(2t) + i(2\sin(t) - 1), \qquad t \in [0, 2\pi]$$

The boundary condition is $f(z) = \frac{z^2}{z - z_p}$ with $z_p = 1.3i$. The error in the interior domain is approximately 10^{-10} (see Figure 4) which is off by a factor of 10^6 from the smooth boundary.

Exercise 9

Figure 7 shows the error for a 16-ply refined mesh. It shows the same magnitude as the star-shaped boundary. To investigate the error as a function of the parameter n, we look at the mean absolute error in the domain indicated in Figure 5

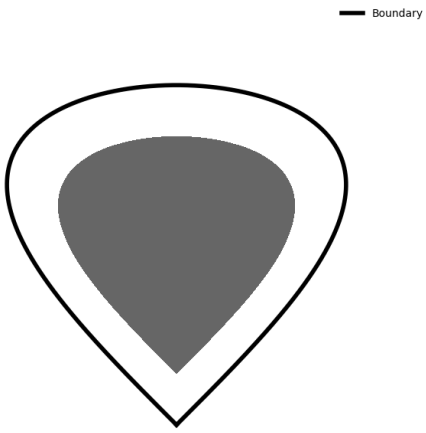

Figure 5: (Grey) Domain used to calulate mean absolute error.

Figure 6 shows that the error decays exponentially as a function of n. Note that since the refinement is only carried out on the corners of segments on the boundary, so the number of gridpoints increases linearly with the number of refinements.

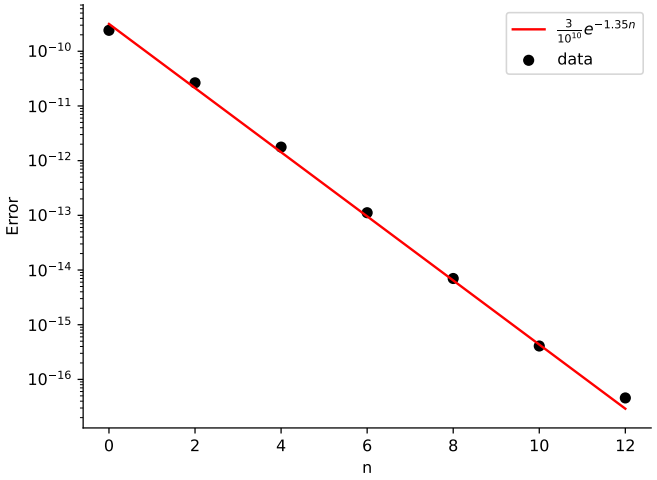

Figure 6: Mean absolute error as a function of the number of refinements applied to the discretisation.

The spectrum of the system matrix still accumulates at 1. The standard deviation tends to zero as $1/x^{0.14}$ which is

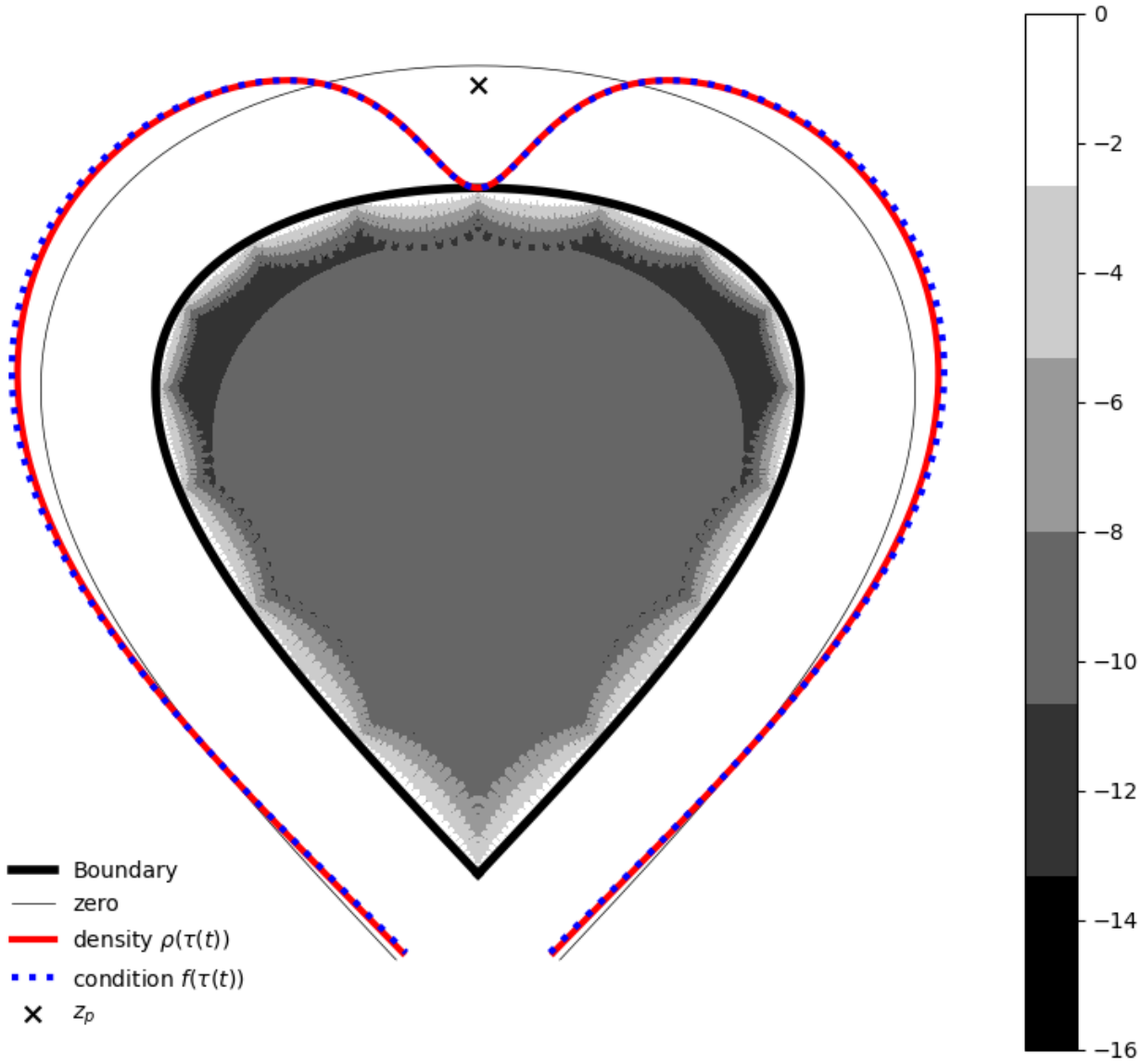

Figure 4: Error for a 16-ply refined mesh. The red line is the graph of $\rho(\tau(t))$, wrapped around the boundary such that $\rho(\tau(t)) = 0$ is at a constant distance from the boundary (gray line). The boundary condition is displayed for reference.

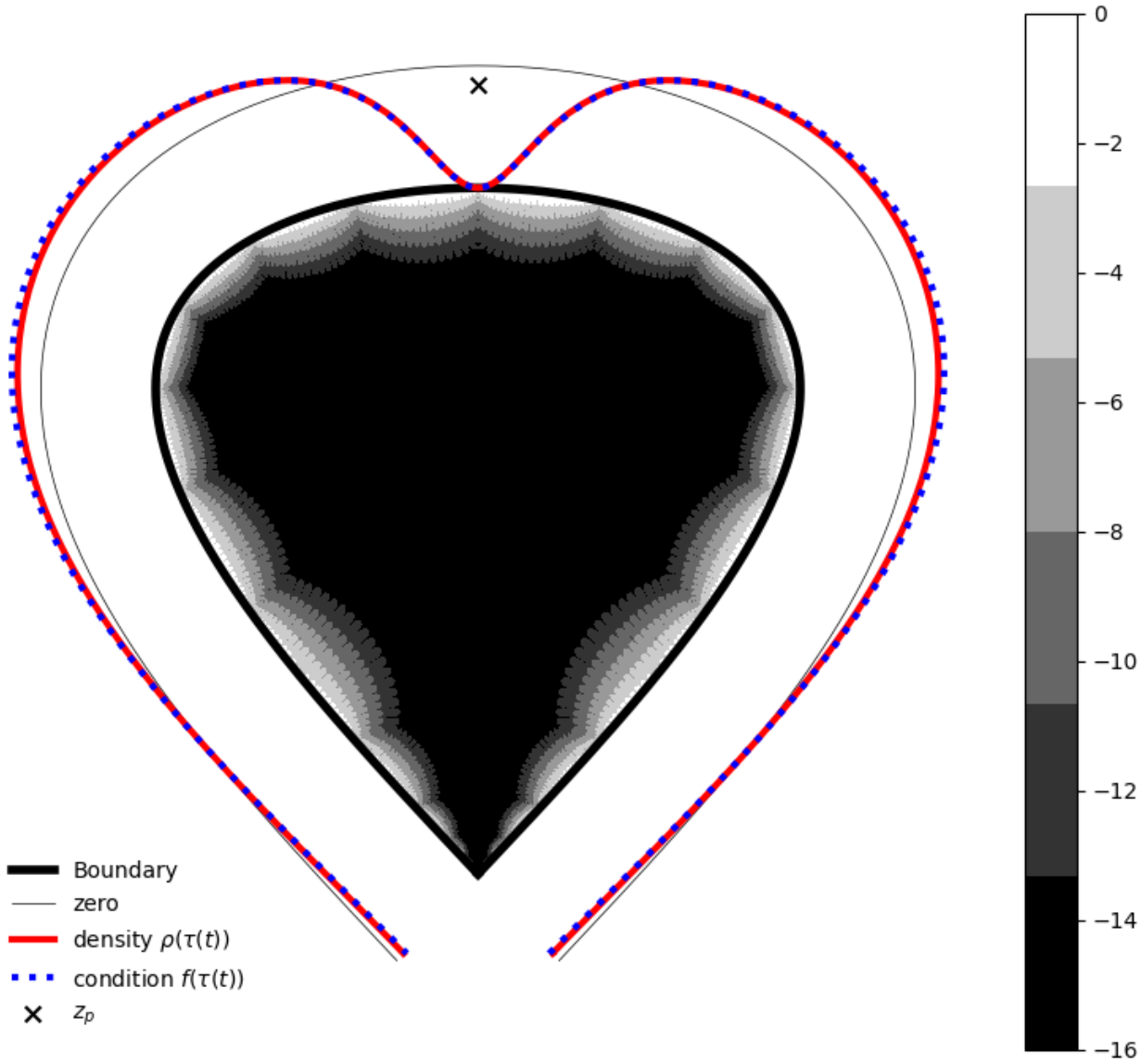

Figure 7: Error for a 16-ply refined mesh. The red line is the graph of $\rho(\tau(t))$, wrapped around the boundary such that $\rho(\tau(t)) = 0$ is at a constant distance from the boundary (gray line). The boundary condition is displayed for reference.

slower than when refining the grid for the smooth boundary. This might suggest that the eigenvalues are more spread out for non-smooth boundaries. However, the condition number still seems to converge to around 4 so the system should be stable. Hence, GMRES will converge fairly quickly.

Exercise 10

Ignoring the innermost six segments of a 20-ply refined grid, the density behaves approximately like $t^{1.35}$ close to the corner, as seen in Figure 8. This is in agreement with the idea that the density has non-polynomial properties. Setting $z_p = -1.3 + 1.3i$ instead of $1.3i$ results in a behaviour that is closer to $t^{0.7}$, as seen in Figure 9. It seems that the symmetry of the problem resulting from $z_p = 1.3i$ results in a "nicer" density.

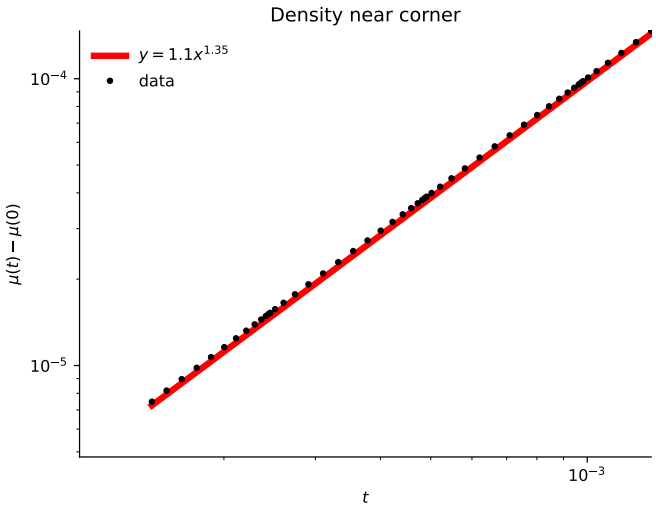

Figure 8: Density $|\mu(t_n) - \mu(t_{10})|$ for gridpoints $n = 20, 25, \ldots, 250$. $t = 0$ corresponds to the sharp corner of the domain in Figure 7.

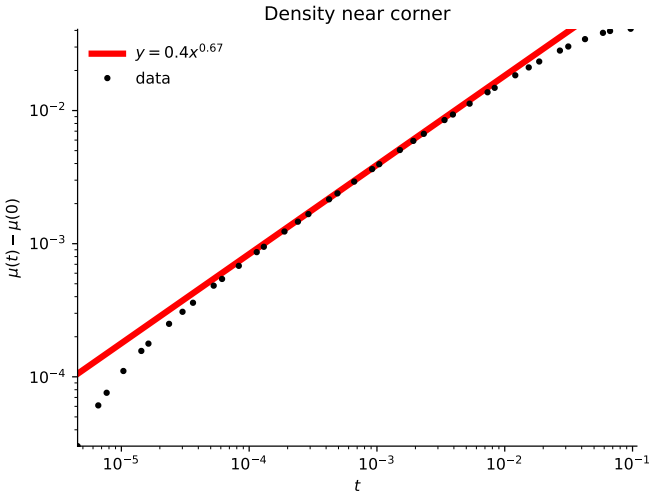

Figure 9: Density $|\mu(t_n) - \mu(t_{10})|$ for gridpoints $n = 20, 25, \ldots, 250$ with the setting discussed in exercise 7–9 but a shifted singularity $z_p = -1.3 + 1.3i$. $t = 0$ corresponds to the sharp corner of the domain in Figure 7.

Exercise 11

Changing to a piecewise smooth boundary, we go from a boundary in C^∞ to C^0. Atkinson show that the integral operator is compact under the assumption that the boundary is in C^2 or better (see Atkinson, p316). This result relies on deriving well-defined limits of the kernel $K(s,t)$ as $s \to t$ and it is not obvious that this limit exists on the corners. Having a compact integral operator is nice because it is sufficient to guarantee a well-behaved spectrum with eigenvalues accumulating at 1. However, from the experiments above it is clear that the system is still well-posed. The main difficulty seems to instead be that convergence is slower. Because of the corners, the trapezoid method does not yield spectral accuracy anymore, so we need more advanced quadrature, making sure to stay away from the cornerpoints. The situation is worsened further as the density exhibits non-polynomial behaviour close to the corners. This is a problem since numerical integration methods like Gauss-Legendre quadrature performs best when integrating polynomials. One way to tackle this problem is to dyadically refine the grid close to corners, which helped a lot as seen in Figue 7.

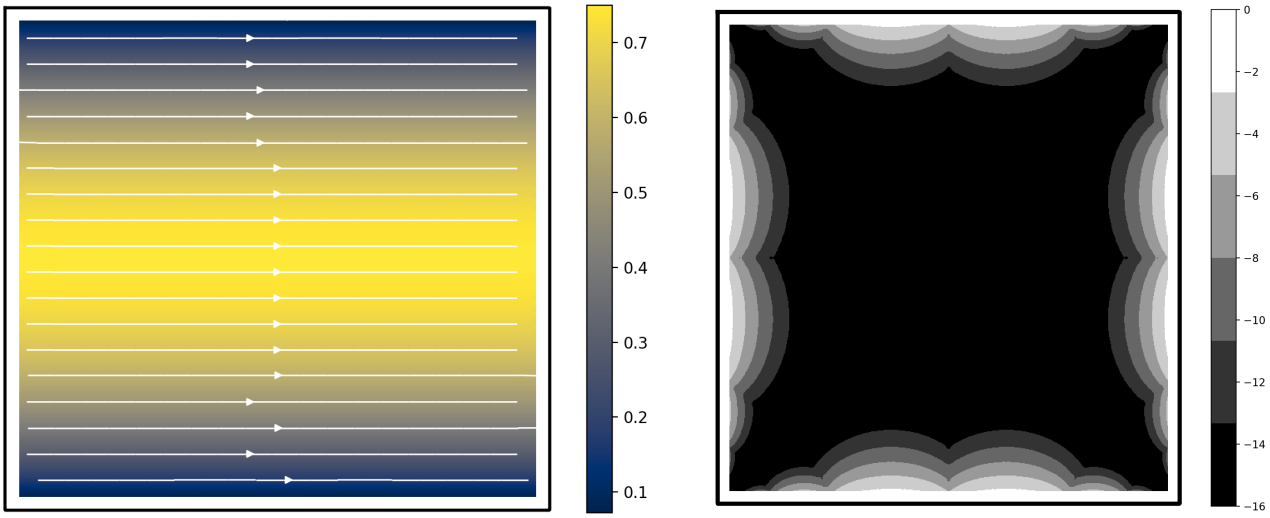

Figure 10: Flow through pipe. The inlet and outlet boundary conditions are quadratic functions with zeros at the corners. The left plot shows a stream plot of the velocity field. The background is colored according to the magnitude of the flow field (speed). The right plot shows the magnitude of the error. We have used a 40-ply refined grid with Gauss-Legendre quadrature.

Exercise 12

For the pipe flow in Figure 10, the boundary condition is $u(x, y) = \frac{3}{4}(1 - y^2) + 0i$, corresponding to a total flow rate of

$$\int_{-1}^{1} \frac{3}{4}(1 - y^2)\mathrm{d}y = 1$$

volume units per unit time through the pipe. We can still obtain machine precision like in the case of the Poisson equation, but it seems to require slightly more refinement. For the lid-driven cavity, we use a similar condition $u(x, y) = 0 + \frac{3}{4}(1 - y^2)i$ on the left boundary, and non-slip conditions everywhere else. In this case, the flow is instead parallel to the boundary. Furthermore, to make things a bit more interesting we look at a tapered pipe, which forces some high frequency details near the corners (see Figure 11).

Final Notes

For Figure 11, I have looked at the solution to Stokes equations with flow that is tangental to the boundary. The boundary is a Fourier series that has been fitted to hand drawn data. The data is obtained by extracting all points in a drawn black-white image whose pixel values exceed some threshold, and using the resulting point cloud to fit a Fourier series by minimising the mean square error locally on different points of the curve. The pixel mask has been obtained by formulating a discriminator function as follows. Suppose $\tau\colon [0, 1] \to \mathbb{C}$ is a smooth parametrisation of the boundary, and define an extension that expands or contracts the boundary perpendicular to its tangent:

$$\tau(t, s) := \tau(t) + \frac{s}{i}\tau'(t), s \in \mathbb{R}.$$

Note that fixing s and viewing $\tau(t, s)$ as a curve, we only have a simple curve (no loops) for small s. We then define a discriminator $d\colon \mathbb{C} \to \mathbb{R}$ by placing basis functions along an inner and outer expansion of the boundary:

$$d_{\mathrm{d}s}(z) := \int_0^1 \frac{1}{2\mathrm{d}s}\left(\frac{1}{|\tau(t, +\mathrm{d}s) - z|^2} - \frac{1}{|\tau(t, -\mathrm{d}s) - z|^2}\right)|\tau'(t)|\mathrm{d}t.$$

The idea behind this expression is that by placing positive basis functions on the outer boundary and negative basis

functions on the inner boundary, we will get a discriminator that is positive outside the boundary and negative inside the boundary. Taking the limit as $\mathrm{d}s \to 0$ and simplifying, we get the expression

$$d(z) = \int_0^1 \frac{\Im\left\{(\tau(t) - z)\overline{\tau}'(t)\right\}}{|\tau(t) - z|^4} |\tau'(t)| \mathrm{d}t.$$

We can now check if z is in the interior by verifying that $d(z) < 0$. The argument is heuristic but it seems to work in practice and produces a nice limiting expression. It might be possible to show that the integral operator above is some Greens function of a PDE operating on a density such that the solution reaches its maximum of 0 at the boundary (for example, we know harmonic functions reach their maxima on the boundary). To get a mask that pads the boundary to avoid difficulties with singularities, we take

$$m_D(z) = \begin{cases} 1 & \text{if } -D < d(z) < 0 \\ 0 & \text{otherwise} \end{cases}$$

for some big $D > 0$. Then, m is the indicator function for determining if a point z lies inside the mask. The discriminator d can now be discretised with the same quadrature as we use to solve the PDE.

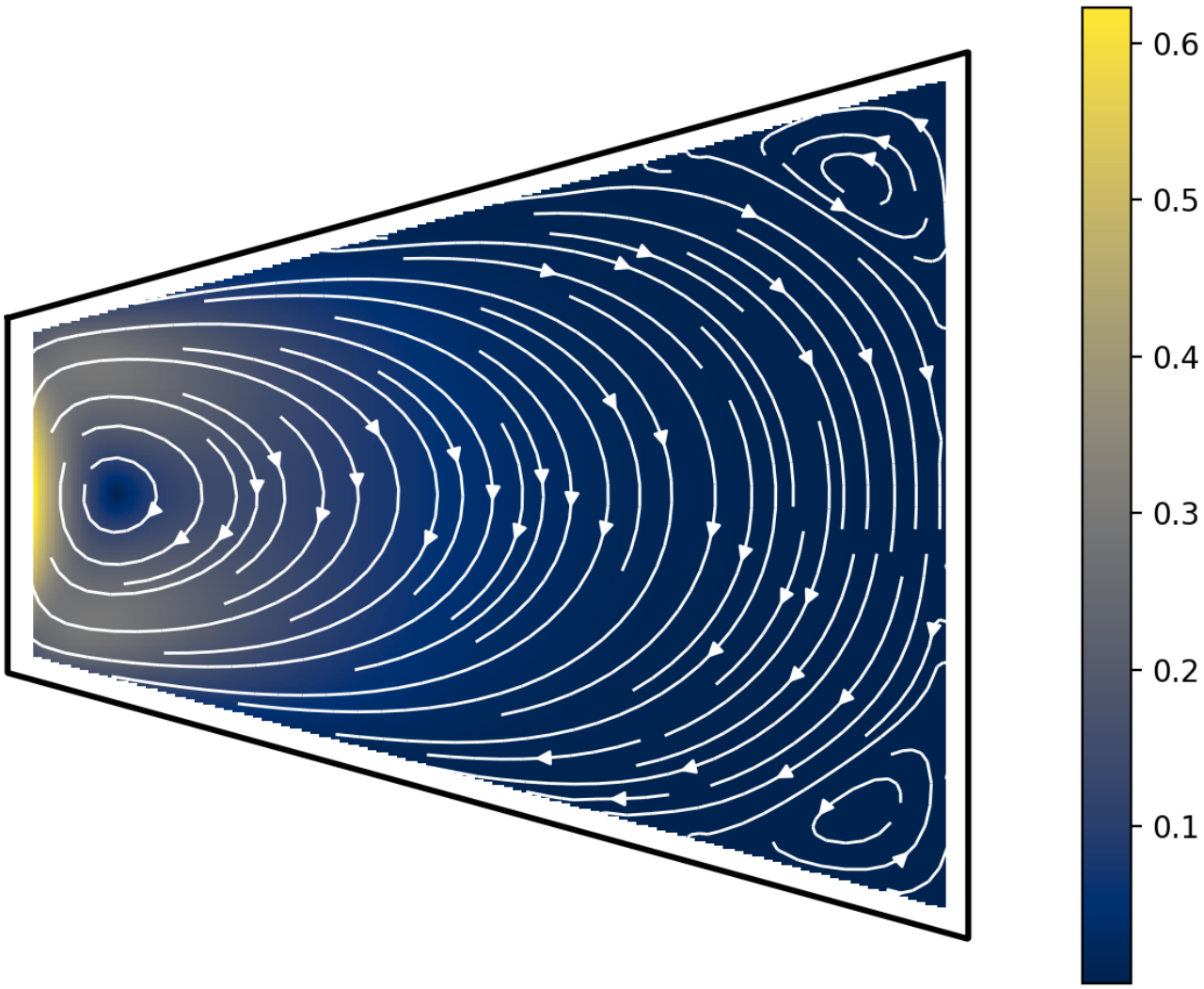

Figure 11: Lid driven cavity flow. The "lid" is positioned along the left boundary, and the boundary condition is a quadratic function with zeros at the corners. The scalar field shows the magnitude of the flow field (speed). We have used a 16-ply refined grid with Gauss-Legendre quadrature.

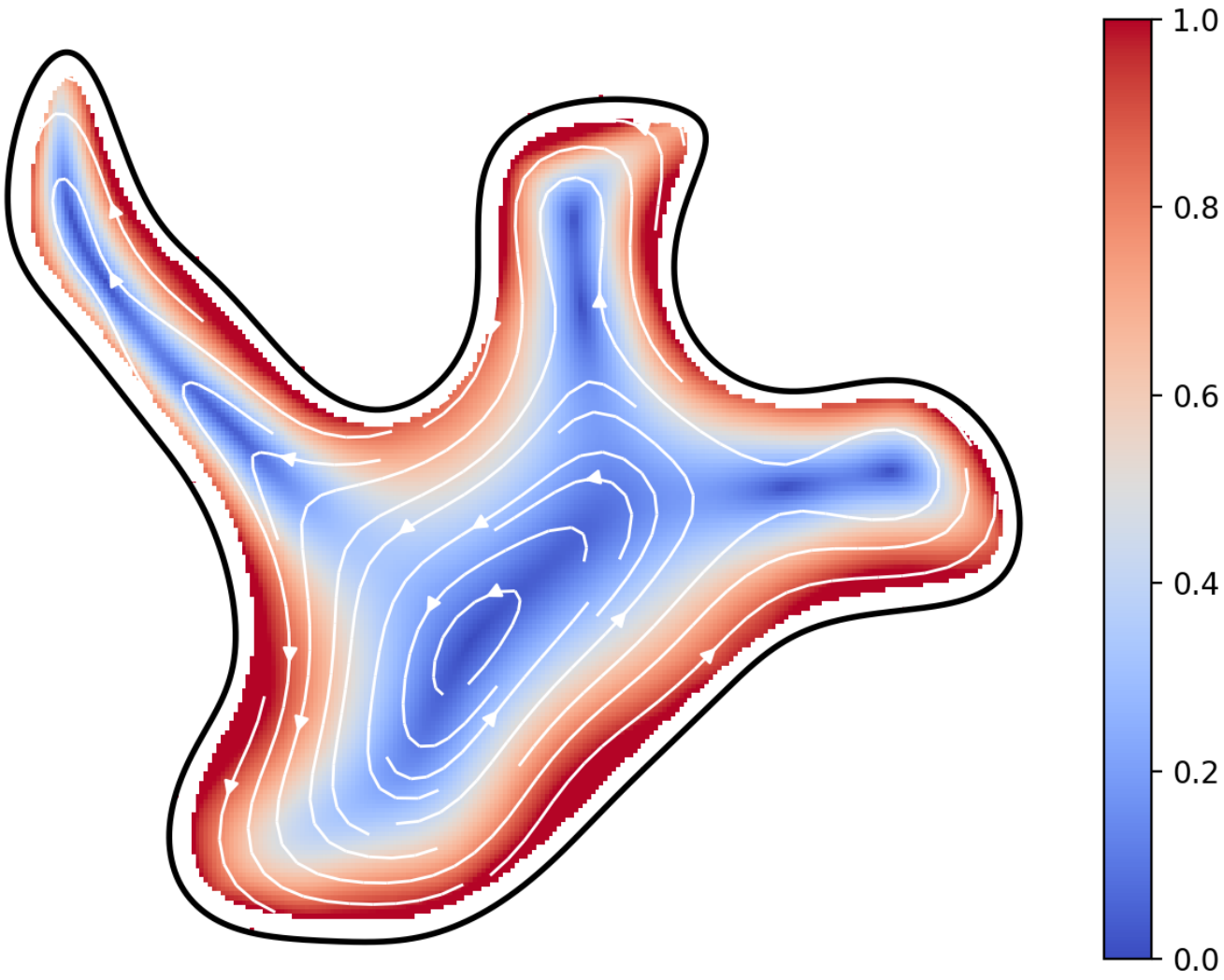

Figure 12: Stokes Flow with dirichlet boundary condition of speed 1.0 tangental to the boundary. The solution is obtained with 16 panels with 16-point Gauss-Legendre quadrature in each panel. The scalar field shows the magnitude of the velocity field.