Problem Set 1

Math228B Numerical solutions to differential equations

Håkon Olav Torvik

UC Berkeley

(Dated: February 3, 2022)

Problem 1a

The 9-point and 5-point Laplacians are given as

$$
\begin{aligned}
\nabla_9^2 u_{i,j} =& \frac{1}{6h^2}[20u_{i,j} - 4u_{i-1,j} - 4u_{i+1,j} - 4u_{i,j-1} - 4u_{i,j+1} \\
&- 1u_{i+1,j+1} - 1u_{i+1,j-1} - 1u_{i-1,j+1} - 1u_{i-1,j-1}], \\
\nabla_5^2 u_{i,j} =& \frac{1}{h^2}[4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}]
\end{aligned} \tag{1}
$$

The dominant error term in the 5-point Laplacian is know, and given by

$$\frac{h^2}{12}(u_{xxxx} + u_{yyyy}),$$

with the remaining error goes as $\mathcal{O}(h^4)$.

In the the book, the true solution is applied to the 9-point laplacian before Taylor expanding. This gives the following expression for the 9-point Laplacian

$$
\begin{aligned}
\nabla_9^2 u_{i,j} &= \nabla^2 u + \frac{h^2}{12}\left(u_{xxxx} + 2u_{xxyy} + u_{yyyy}\right) + \mathcal{O}(h^4), \\
&= \nabla^2 u + \frac{h^2}{12}(u_{xxxx} + u_{yyyy}) + \frac{h^2}{6}u_{xxyy} + \mathcal{O}(h^4), \\
&= \nabla_5^2 u + \frac{h^2}{6}u_{xxyy} + \mathcal{O}(h^4).
\end{aligned}
$$

Problem 1b

Solving Laplaces' equation, $\nabla^2 u(x,y) = f(x,y)$, using the 9-point Laplacian approximation (1), it is re-written

$$\nabla_9^2 u_{ij} = f_{ij},$$

where

$$f_{ij} = f(x_i, y_j) + \frac{h^2}{12}\nabla^2 f(x_i, y_j).$$

Using the 5-point Laplacian on f here, the set of equations that needs to be solved are written

$$[20u_{i,j} - 4u_{i-1,j} - 4u_{i+1,j} - 4u_{i,j-1} - 4u_{i,j+1} - 1u_{i+1,j+1} - 1u_{i+1,j-1} - 1u_{i-1,j+1} - 1u_{i-1,j-1}] = 6h^2\left(f_{ij} - \frac{1}{12}[4f_{i,j} - f_{i-1,j} - f_{i+1,j} - f_{i,j-1} - f_{i,j+1}]\right).$$

The function `assemblePossion` from the course page is modified to implement this, in fuction `Poisson9(n, f, g)` on line 54 in my code submission file. `assemblePossion` is also a possible to call with same signature, in case the name was important. To test that I get 4th order convergence, I use the test-function from the course page with known exact solution and finds the maximum relative error for different grid sizes. I plot the log of the errors against the log of the step-length h, such that the slope λ is the convergence rate $\mathcal{O}(h^\lambda)$. This is done for both the 9-point stencil, as well as the 5-point stencil. As can be seen in Figure 1, the 9-point stencil is indeed 4th order accurate, while the 5-point is 2nd order.

problem 2a

The mapping

$$T = \begin{cases} \xi(x,y) = \frac{x}{B/2+Ay/H} = \frac{x}{\gamma'}, \quad \gamma'(x,y) \equiv B/2 + Ay/H, \\ \eta(x,y) = y/H, \end{cases}$$

is one transform from Ω to the unit square $\hat{\Omega}$. The the inverse transform is

$$T^{-1} = \begin{cases} x(\xi,\eta) = \xi\gamma, \quad \gamma(\xi,\eta) \equiv B/2 + A\eta, \\ y(\xi,\eta) = H\eta. \end{cases}$$

In Ω, Poissons equation to solve looks like

$$-(u_{xx} + u_{yy}) = 1,$$

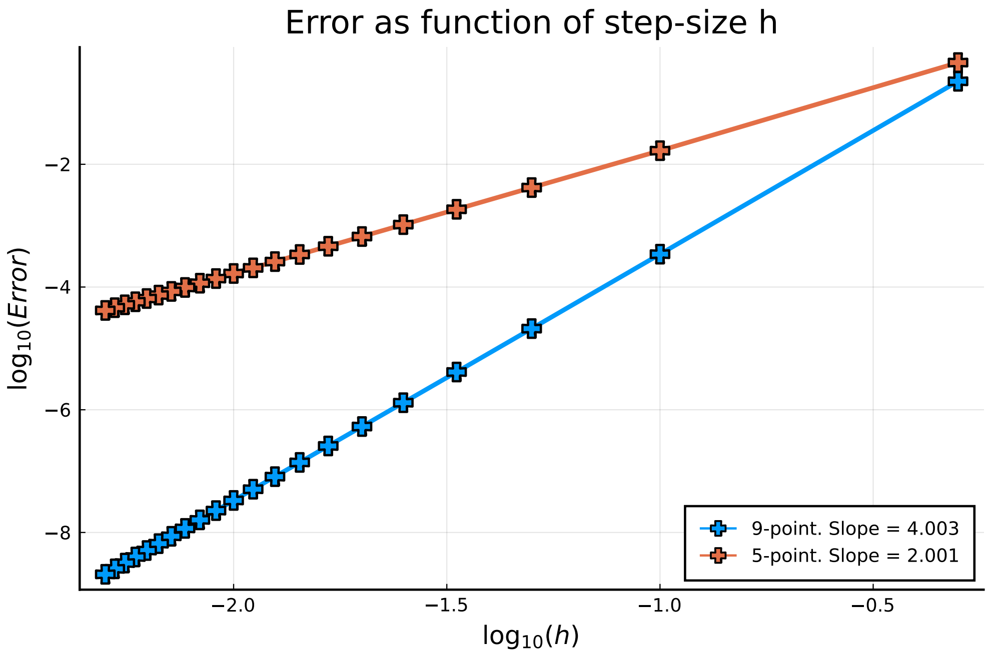

Figure 1: Log-log plot of relative error against step-length for 5-point and 9-point Laplacian stencil, to show the convergence rates of both. The 5-point goes as $\mathcal{O}(h^2)$, while the 9-point as $\mathcal{O}(h^4)$, which is as expected. The 2 points with highest step-length are excluded from the estimation of the slope, in order to avoid inaccuracies for large h.

while in $\hat{\Omega}$, it takes the shape

$$au_{\xi\xi} - 2bu_{\xi\eta} + cu_{\eta\eta} + du_{\eta} + eu_{xi} = -J^2 f, \tag{2}$$

where a, b, c, d, e, J are functions of the first and second derivatives of x and y w.r.t. ξ and η. These derivatives that do not evaluate to 0 are as follows:

$$x_\xi = \gamma \quad x_\eta = A\xi \quad y_\eta = H \quad x_{\eta\xi} = A$$

Using the definitions in the lecture slides, I find that the expressions for the functions of ξ and η are given by

$$\begin{aligned} \alpha &= -2bA & \beta &= 0 \\ a &= A^2\xi^2 + H^2 & b &= A\xi\gamma \\ c &= \gamma^2 & e &= 2A^2\xi \\ d &= 0 & J &= H\gamma \end{aligned}$$

The Dirichlet boundary conditions stay the same in the computational domain, that is

$$u = 0 \text{ for } P_1'P_2' \text{ and } P_2'P_3'.$$

The von Neumann condition at $P_1'P_4', \xi = 0$, is given with the normal derivative, and the normal vector is unchanged. Thus

$$\frac{du}{dn} = \frac{du}{d\xi} = 0$$

where I have used a 3-point boundary stencil to approximate the derivative.

At $P_3'P_4', \eta = 1$, the normal derivative is given by

$$\frac{\partial u}{\partial n} = \frac{1}{J}\left[(y_\eta n^x - x_\eta n^y)u_\xi + (-y_\xi n^x + x_\xi n^y)u_\eta\right]$$

where

$$(n^x, n^y) = \frac{1}{\sqrt{x_\xi^2 + y_\xi^2}}(-y_\xi, x_\xi) = (0, 1),$$

such that

$$\frac{\partial u}{\partial n} = \frac{1}{J}\left[-A\xi u_\xi + \gamma u_\eta\right] = \frac{-A\xi}{J}u_\xi + \frac{1}{H}u_\eta.$$

Problem 2b

With all the expressions in the previous secion, I now have all that is needed to discretize the computational domain and write down the systems of equations to solve the system.

First, using $\Delta\xi = \Delta\eta = h = 1/n$, we have $\xi \to \xi_i = ih, \eta \to \eta_j = jh$ for $i, j \in [0, n]$. Then, a function $g(\xi, \eta) \to g(\xi_i, \eta_j) = g(i, j) = g_{i,j}$. Setting $f_{ij} = 1$, and using 2nd order approximations for the derivatives in (2), the equations for the interiour points are written

$$\begin{aligned} -J_{i,j}^2 h^2 = &\big[a_{ij}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) - \frac{b_{i,j}}{2}(u_{i-1,j+1} + u_{i+1,j-1} - u_{i-1,j-1} - u_{i+1,j+1}) \\ &+ c_{ij}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) + \frac{d_{i,j}}{2}(u_{i,j+1} - u_{i,j-1}) + \frac{e_{i,j}}{2}(u_{i+1,j} - u_{i-1,j})\big] \end{aligned}$$

For the boundaries, we have

$$
\begin{aligned}
j = 0 \lor i = n \Rightarrow & u_{i,j} = 0 && \text{Derichlet at} P_1'P_2' \land P_2'P_3'. \\
i = 0 \Rightarrow & u_\xi \approx \frac{1}{h^2}\left[-1.5u_{i,j} + 2u_{i+1,j} - 0.5u_{i+2,j}\right] = 0, && \text{von Neumann at} P_1'P_4'. \\
j = n \Rightarrow & \frac{-A\xi_i}{J_{i,j}} \frac{1}{2h^2}\left[u_{i+1,j} - u_{i-1j}\right] + \\
& \frac{1}{H}\frac{1}{h^2}\left[-1.5u_{i,j} + 2u_{i,j-1} - 0.5u_{i,j-2}\right] = 0 && \text{von Neumann at} P_3'P_4'.
\end{aligned}
$$

3 of the corners are handled by the first case and set to 0, while the last is handled by the second case, using a left boundary stencil.

These equations can be written simply as $Au = b$. When we have the solution u, the flowrate $\hat{Q}$ in the computational domain, which is an intergral over the domain, can be approximated by the 2D-trapezoidal rule, a 2nd order method, as such: [1]

$$
\begin{aligned}
\hat{Q} = & \iint_{\hat{\Omega}} u_{i,j} d\xi d\eta \\
= & \frac{h^2}{4}\left[u_{0,0} + u_{0,n} + u_{n,0} + u_{n,n} + 2\left(\sum_i u_{i,0} + \sum_i u_{i,n} + \sum_j u_{0,j} + \sum_j u_{n,j}\right) + 4\sum_{i,j} u_{i,j}\right]
\end{aligned}
$$

Problem 2c

The function `buildA(L, B, H, n)` in my code constructs the matrix A, vector b as well as the matricies for the physical grid x and y. `channelflow(L, B, H, n)` solves $Au = b$, and plots the result. For $H = 1, L = 3, B = 0.5, n = 20$, the resulting contour and grid is shown in Figure 2. The figure shares a great resembelance with the figure in the problem set, so I am confident my implementation is at least mostly correct.

Problem 2d

For different values of B, I calculate the flowrate $\hat{Q}$ for increasing number of points in the discretization, to estimate the convergence rate. As I dont have the exact solution, I let $Q_{true} \equiv Q(n = 640)$, and calcuate the relative error for $n = 10 * 2^k, k \in [1 : 6]$. I plot the results in a log-log plot as in Problem 1b. This is shown in Figure 3, for $B = 0, 0.5, 1$ in separate plots. We see that in all cases, the convergence is somewhat better than 2nd order.

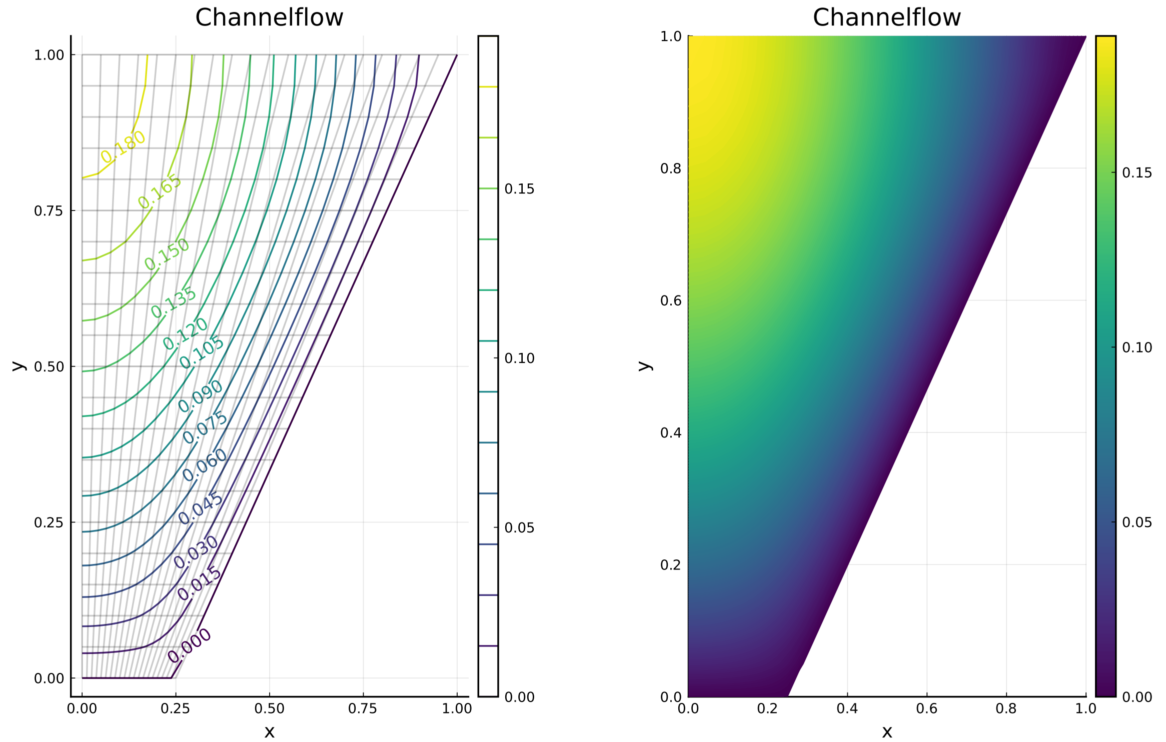

Figure 2: Left: unfilled contour with grid of flow in channel. Right: Filled contour of the same simulation.

There is no big difference for the different values of B, neither for the slope nor the value of the error.

The flowrate is similar for the case of $B = 0$ and $B = 1$, and highest for $B = 0.5$. This makes sense physically, as this corrosponds to the shape of the chanel with the largest cross-sectional area (among those 3 values. $B = L - H\sqrt{16/3} \approx 0.69$ is the global maximum.)

Problem 3

The PDE

$$u_t = \kappa u_{xx} - \gamma u$$

can be solved with the θ-parameterized scheme

$$U_j^{n+1} = U_j^n + \frac{k\kappa}{2h^2}\left[U_{j-1}^n - 2U_j^n + U_{j+1}^n + U_{j-1}^{n+1} - 2U_j^{n+1} + U_{j+1}^{n+1}\right] - k\gamma\left[(1-\theta)U_j^n + \theta U_j^{n+1}\right].$$

Setting $\gamma = 0$ gives the heat equation which we know has a local truncation error of $\mathcal{O}(h^2)$, so to study the local truncation error in this PDE, I ignore that part by setting $\kappa = 0$. We

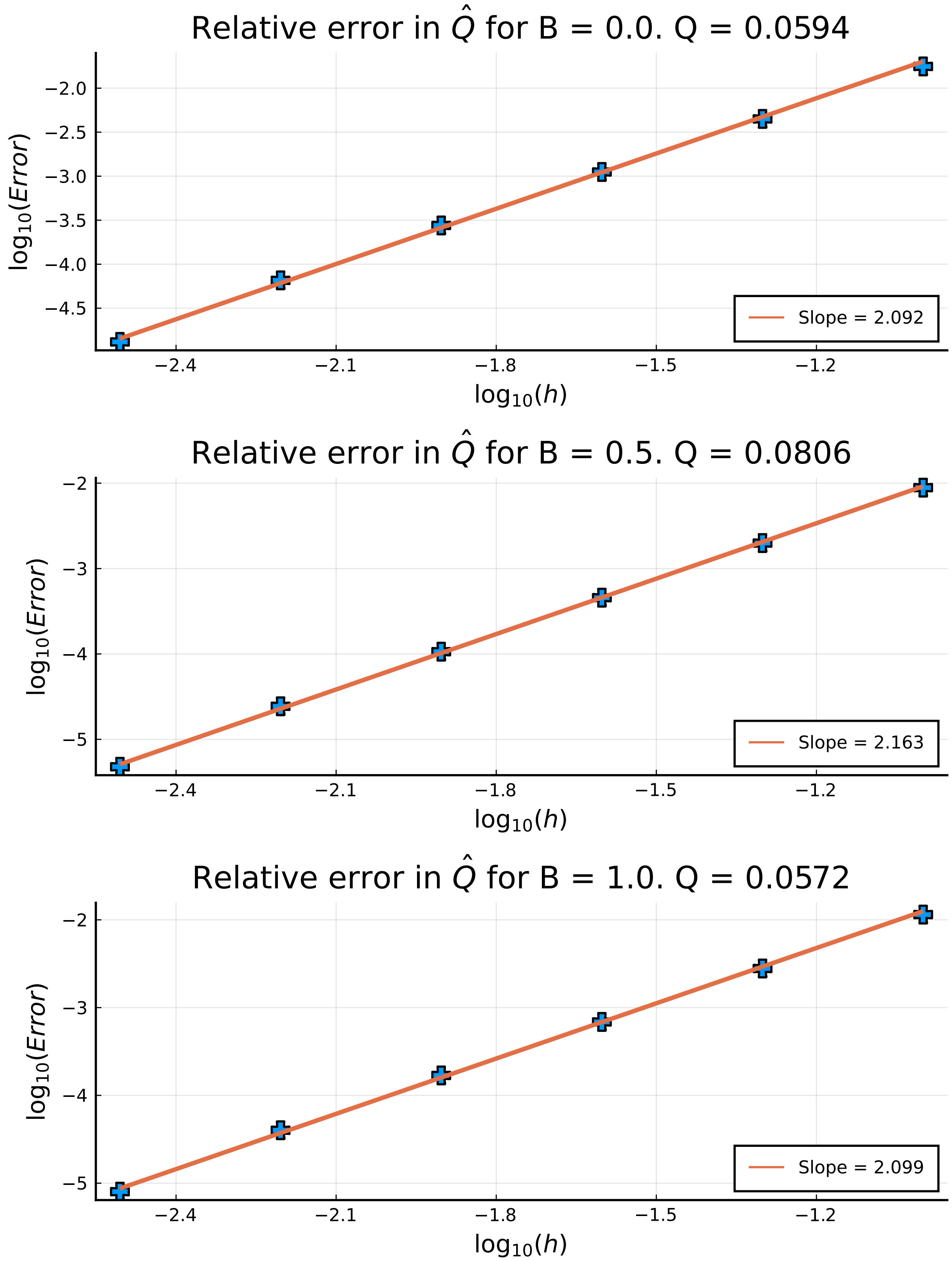

Figure 3: Convergence of error in $\hat{Q}$ for different values of B. The first point is excluded in the estimation of the slope to aviod large inaccuracies when h is big.

then get

$$\begin{aligned} U_j^{n+1} =& U_j^n - k\gamma\left[(1-\theta)U_j^n + \theta U_j^{n+1}\right] \\ (1+k\gamma\theta)U_j^{n+1} &= [1-k\gamma(1-\theta)]U_j^n. \\ (1+k\gamma\theta)U(x,t+k) &= [1-k\gamma(1-\theta)]U(x,t). \end{aligned} \tag{3}$$

Taylor expanding the left side as

$$U(x,t+k) \approx U + kU_t + \frac{k^2}{2}U_{tt} + \mathcal{O}(k^3).$$

From the original PDE, we know

$$\begin{aligned} u_t &= -\gamma u \\ u_{tt} &= (u_t)_t = (-\gamma u)_t = -\gamma u_t = \gamma^2 u \end{aligned}$$

Inserting this into (3), and subtracting the two sides from eachother gives the local truncation error

$$\begin{aligned} \tau &= (1+k\gamma\theta)\left(U - k\gamma U + \frac{k^2\gamma^2}{2}U + \mathcal{O}(k^3)\right) - [1-k\gamma(1-\theta)]U \\ &= (1+k\gamma\theta)\left(-k\gamma U + \frac{k^2\gamma^2}{2}U + \mathcal{O}(k^3)\right) + k\gamma U \\ &= k^2\gamma^2\theta U + \frac{k^2\gamma^2}{2}U + \mathcal{O}(k^3) \\ &\begin{cases} \mathcal{O}(k^3), & \theta = \frac{1}{2} \\ \mathcal{O}(k^2), & \text{otherwise} \end{cases} \end{aligned}$$

The global error is 1 order lower, and bringing back in the error for $\kappa \neq 0$, we get that the method is $\mathcal{O}(k^p + h^2)$ accurate, where $p = 2$ for $\theta = \frac{1}{2}$ and $p = 1$ otherwise.

---

[1] Math StackExchange, *Derivation of 2D Trapezoid Rule*, retrieved at the internet from *https://math.stackexchange.com/questions/2891298/derivation-of-2d-trapezoid-rule* at 02/02/2022