PROJECT 1: FINITE DIFFERENCE METHODS

PRESTON MALEN

February 2023

Consider the ordinary differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\kappa(x)\frac{\mathrm{d}}{\mathrm{d}x}u(x)\right) = f(x), \qquad x \in [0,1], \tag{1}$$

with homogeneous Dirichlet boundary conditions, $u(0) = u(1) = 0$, and where scalar diffusion coefficient κ is given by,

$$\kappa(x) = 2 + \sum_{\ell=1}^{5} \frac{1}{\ell+1}\sin(\ell\pi x).$$

The goal of this exercise will be to numerically compute solutions to this problem.

(a) Define the operator,

$$\widetilde{D}_0 u(x_j) = \frac{u(x_j + h/2) - u(x_j - h/2)}{h}, \qquad h = 1/(N+1), \qquad x_j \coloneqq jh,$$

for a fixed number of points $N \in \mathbb{N}$. Then with u_j the numerical solution approximating $u(x_j)$ for solving the $d = 1$ version of (1), consider the scheme,

$$\widetilde{D}_0\left(\kappa(x_j)\widetilde{D}_0 u_j\right) = f(x_j), \qquad j \in [N]. \tag{2}$$

Show that, for smooth u and κ, this scheme has second-order local truncation error.

(b) Construct an exact solution via the *method of manufactured solutions*: posit an exact (smooth) solution $u(x)$ (that satisfies the boundary conditions!) and, compute f in (1) so that your posited solutions satisfies (1).

(c) Implement the scheme above for solving (1), setting f to be the function identified in part (b), so that you know the exact solution. Show that indeed you achieve second-order convergence in h (say in the $h^{d/2}$-scaled vector ℓ^2 norm) . (To "show" this, plot on a log scale the error as a function of a discretization parameter, such as h or N, and verify that the slope of the resulting line is what is expected.)

Solutions

(a) To calculate the local truncation error, we just need to expand the left-hand side of (2) and show we get a residual of h^2. So first, we consider the product $\kappa(x_j)\widetilde{D}_0 u_j$:

$$\kappa(x_j)\widetilde{D}_0 u_j = \kappa(x_j)\left(\frac{u(x_j + h/2) - u(x_j - h/2)}{h}\right) \tag{3}$$

Now, we apply the given difference operator $\widetilde{D}_0$ to (3). For brevity, we will define the following:

$$\begin{aligned}
\kappa_+ &= \kappa(x_j + h/2) \\
\kappa_- &= \kappa(x_j - h/2) \\
D_+ &= \kappa_+\left(\frac{u(x_j + h) - u(x_j)}{h}\right) \\
D_- &= \kappa_-\left(\frac{u(x_j) - u(x_j - h)}{h}\right)
\end{aligned}$$

So in summary, we have the expression:

$$f(x_j) = \widetilde{D}_0\left(\kappa(x_j)\widetilde{D}_0 u_j\right) = \frac{D_+ - D_-}{h} \tag{4}$$

After some painful calculation, we get:

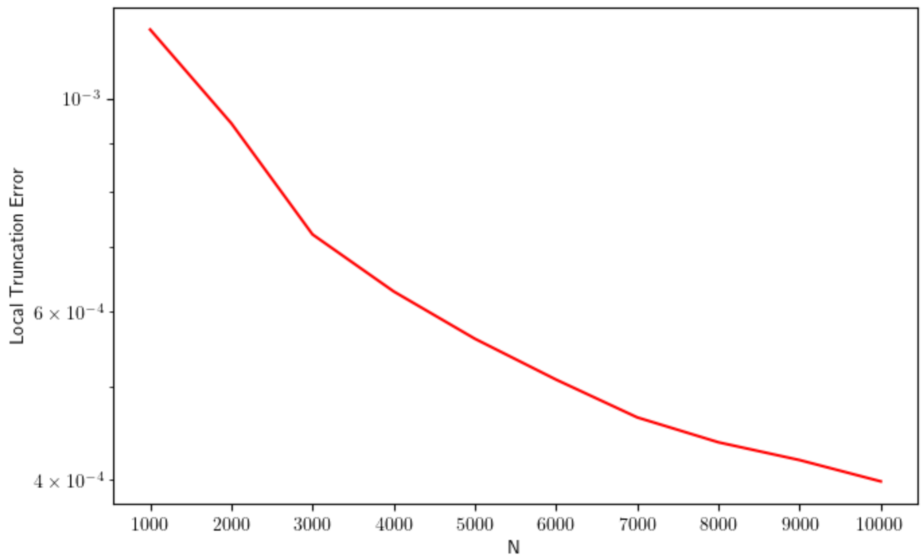
$$h^2 f(x_j) = -u(x_j)(k_+ + k_-) + k_+ u(x_j + h) + k_- u(x_j - h)$$

So the given scheme does in fact have a local truncation error of h^2.

(b) Consider the function $u(x) = x(x-1)$. It is clear that u has roots at 0 and 1 so the boundary conditions are satisfied. We will now compute $f(x)$ explicitly using the given parameters:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}x}\left(\kappa(x)\frac{\mathrm{d}}{\mathrm{d}x}u(x)\right) &= \frac{\mathrm{d}}{\mathrm{d}x}\left[\left(2 + \sum_{\ell=1}^{5}\frac{1}{\ell+1}\sin(\ell\pi x)\right)\cdot(2x-1)\right] \\
f(x) &= 4 + \sum_{\ell=1}^{5}\frac{(2x-1)(\cos(\ell\pi x))(\ell\pi) + 2\sin(\ell\pi x)}{\ell+1}
\end{aligned}$$

(c) First, we choose a suitable N. For this problem, it was sufficient to choose $N \in [10^3, 10^4]$ in increments of 10^3. Graphed below, we have N along the $x-$axis, and the local truncation error of the scheme on the $y-$axis. Note also that the local truncation error is graphed on a log scale.



Using SciPy's `linalg` class, we found acceptable convergence and runtime using an iterative conjugate gradient method with default parameters. It is then clear that we do in fact achieve second order convergence in h using the given scheme.