Exercise 1 Write a $m$–step deferred correction code. First solve the midpoint rule $L_h u = f$ for the uncorrected second–order solution $u$ on an equidistant mesh of $N$ points on the interval $[a,b]$. Save the matrix $L_h$ in factored (or inverted) form for computing the errors. Compute the residuals

$$R_{j+1/2} = P_j'(x_{j+1/2}) + Q(x_{j+1/2})P_j(x_{j+1/2}) - f(x_{j+1/2})$$

with interpolation and differentiation polynomials $P_j(x)$ built on $2m$ mesh points, centered around $j + 1/2$ when possible, but necessarily uncentered at the endpoints of the interval. You may find the routine idwts from the web page or the routine fdstencil from LeVeque’s web page helpful in building such formulas. Use the factored matrix $L_h$ to solve the midpoint rule for the approximate errors $e$ and correct the solution to $u ← u - e$. Repeat $m - 1$ times to yield a solution of order $2m$. Test your method with $m = 2$ through 5 and $N = 2^5$ through $2^{10}$ on the BVP $y' + Q(x)y = f(x)$ on $0 ≤ x ≤ π$, $y(0) + y(π) = g$, where

$$Q(x) = \begin{bmatrix} 1 - 9 \cos 2x & 1 + 9 \sin 2x \\ -1 + 9 \sin 2x & 1 + 9 \cos 2x \end{bmatrix}$$

and $f$ and $g$ are chosen to make $y(x) = (\cos(5x^2), 2 + e^{-10x})^T$ the exact solution. For each precision $ε = 10^{-3}, 10^{-6}, 10^{-9},$ and $10^{-12}$ estimate which grid size $N$ and order $2m$ delivers a solution with max-norm error less than $ε$ at roughly the least cost in CPU time.

Exercise 2 Verify stability of the fourth-order rule from exercise 1 computationally. Compute the norm of the matrix $M_h^{-1}$ that takes data $(g, f_{1/2}, \ldots, f_{N-1/2})^T$ to solution values $(u_0, u_1, \ldots, u_N)^T$, for the $N$ values employed in exercise 1.