Problem 1

**Finite difference approximation (3 points).** Let $f : \mathbb{R} \to \mathbb{R}$ be a smooth function, and let $x_1 < x_2 < x_3 < x_4$ be four increasing values.

(a) Derive a finite difference (FD) approximation for $f''(x_2)$ that is as accurate as possible, based on the four values of $f_1 = f(x_1), \ldots, f_4 = f(x_4)$. Calculate an expression for the dominant term in the error.

(b) Write a program to test the FD approximation on the function

$$f(x) = e^{-x} \tan x. \tag{1}$$

Consider step sizes of $H = 10^{-k/100}$ for $k = 100, 101, \ldots, 300$. For each $H$, set $x_1 = 0$ and $x_4 = H$. Choose $x_2$ and $x_3$ as uniformly randomly distributed random numbers over the range from 0 to $H$.[a] Make a log–log plot showing the absolute error magnitude $E$ of the FD approximation versus $H$. Use linear regression to fit the data to

$$E = CH^p \tag{2}$$

and determine $C$ and $p$ to three significant figures.[b]

(c) **Optional.**[c] Examine and discuss whether your value of the fitted parameter $C$ is consistent with the dominant error term from part (a).

[a] If $x_2 > x_3$, then swap the two values to ensure the ordering is preserved. If $x_2 = x_3$, then choose new random numbers.
[b] Since sample points for the FD approximation are randomly chosen, there will be small variations in the values of $C$ and $p$ that you compute.
[c] Optional questions are not graded.

(a) Using the method of undetermined coefficients as explained in class and in the textbook, we seek to find coefficients $a, b, c, d$ such that

$$f''(x_2) = af(x_1) + bf(x_2) + cf(x_3) + df(x_4) + E \tag{3}$$

where $E$ is the error term. Using Taylor expansions about $x_2$, we have

$$f(x_1) = f(x_2) + f'(x_2)(x_1 - x_2) + \frac{f''(x_2)}{2}(x_1 - x_2)^2 + \frac{f^{(3)}(x_2)}{6}(x_1 - x_2)^3 + \frac{f^{(4)}(\xi_1)}{24}(x_1 - x_2)^4, \tag{4}$$

$$f(x_3) = f(x_2) + f'(x_2)(x_3 - x_2) + \frac{f''(x_2)}{2}(x_3 - x_2)^2 + \frac{f^{(3)}(x_2)}{6}(x_3 - x_2)^3 + \frac{f^{(4)}(\xi_3)}{24}(x_3 - x_2)^4, \tag{5}$$

$$f(x_4) = f(x_2) + f'(x_2)(x_4 - x_2) + \frac{f''(x_2)}{2}(x_4 - x_2)^2 + \frac{f^{(3)}(x_2)}{6}(x_4 - x_2)^3 + \frac{f^{(4)}(\xi_4)}{24}(x_4 - x_2)^4, \tag{6}$$

where $\xi_i$ is some point between $x_i$ and $x_2$. Substituting these into the original equation and collecting terms gives

$$\begin{aligned}
f''(x_2) = {} & (a + b + c + d)f(x_2) && (7)\\
& + (a(x_1 - x_2) + c(x_3 - x_2) + d(x_4 - x_2))f'(x_2) && (8)\\
& + \left(\frac{a(x_1 - x_2)^2}{2} + \frac{c(x_3 - x_2)^2}{2} + \frac{d(x_4 - x_2)^2}{2}\right) f''(x_2) && (9)\\
& + \left(\frac{a(x_1 - x_2)^3}{6} + \frac{c(x_3 - x_2)^3}{6} + \frac{d(x_4 - x_2)^3}{6}\right) f^{(3)}(x_2) && (10)\\
& + \left(\frac{a(x_1 - x_2)^4}{24} f^{(4)}(\xi_1) + \frac{c(x_3 - x_2)^4}{24} f^{(4)}(\xi_3) + \frac{d(x_4 - x_2)^4}{24} f^{(4)}(\xi_4)\right). && (11)
\end{aligned}$$

To ensure that the approximation is exact for polynomials of degree up to 3, we require that the coefficients of $f(x_2)$, $f'(x_2)$, $f''(x_2)$, and $f^{(3)}(x_2)$ match on both sides, leading to the system of equations:

$$a + b + c + d = 0, \tag{12}$$

$$a(x_1 - x_2) + c(x_3 - x_2) + d(x_4 - x_2) = 0, \tag{13}$$

$$\frac{a(x_1 - x_2)^2}{2} + \frac{c(x_3 - x_2)^2}{2} + \frac{d(x_4 - x_2)^2}{2} = 1, \tag{14}$$

$$\frac{a(x_1 - x_2)^3}{6} + \frac{c(x_3 - x_2)^3}{6} + \frac{d(x_4 - x_2)^3}{6} = 0. \tag{15}$$

Solving this system yields the coefficients:

$$
\begin{aligned}
a &= \frac{2x_2 - x_3 - x_4}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)}, \\
b &= \frac{2x_2 - x_1 - x_4}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)}, \\
c &= \frac{2x_2 - x_1 - x_4}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)}, \\
d &= \frac{2x_2 - x_1 - x_3}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)}.
\end{aligned}
$$

The dominant term in the error is given by

$$
E = \left( \frac{a(x_1 - x_2)^4}{24} f^{(4)}(\xi_1) + \frac{c(x_3 - x_2)^4}{24} f^{(4)}(\xi_3) + \frac{d(x_4 - x_2)^4}{24} f^{(4)}(\xi_4) \right). \tag{16}
$$

So we have a third-order accurate finite difference approximation for $f''(x_2)$.

(b) See the attached code file `714Hw1.py` for the implementation of the finite difference approximation and the testing on the function $f(x) = e^{-x} \tan x$. The log-log plot of the absolute error magnitude $E$ versus $H$ is shown below, along with the results of the linear regression fit to $E = CH^p$ showing the values of $C$ and $p$ to three significant figures.

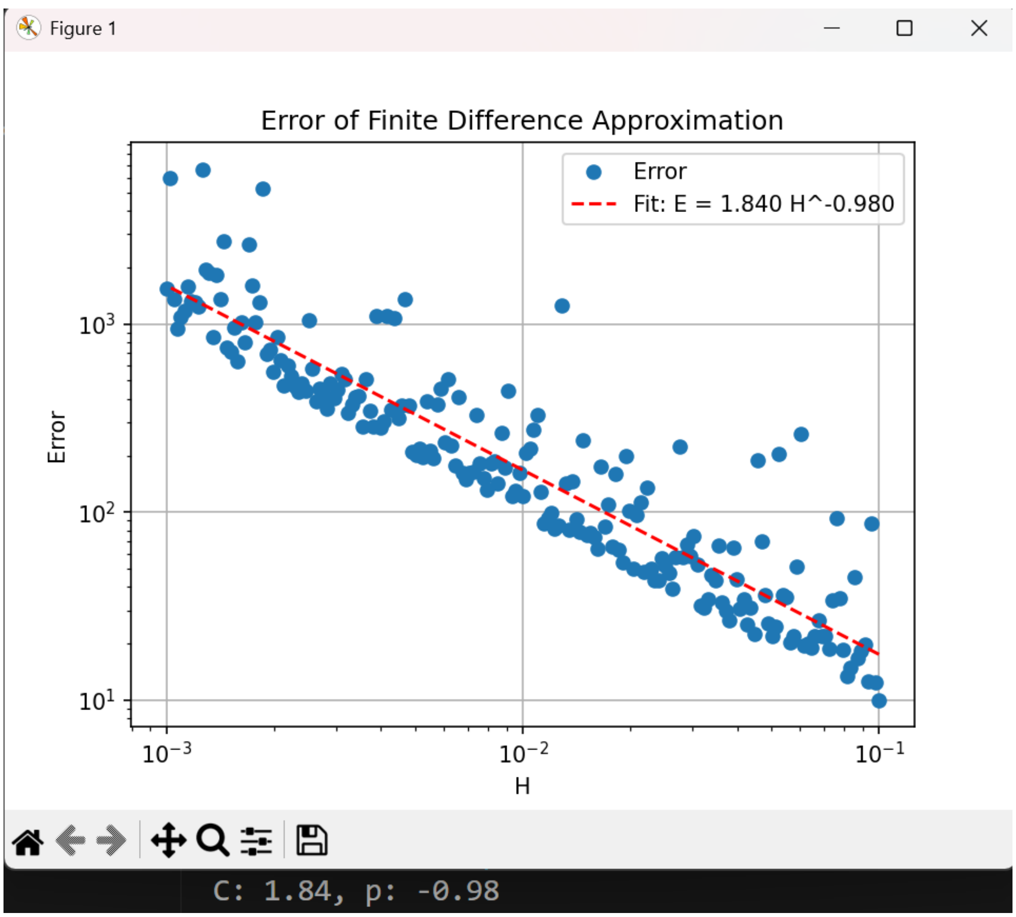


Figure 1: Log-log plot of absolute error magnitude $E$ versus step size $H$. The dashed line represents the linear regression fit to $E = CH^p$.

## Problem 2

**Mixed boundary value problem (5 points).** For a smooth function $u(x)$ and source term $f(x)$, consider the two point boundary value problem (BVP)

$$u'' + u = f(x) \tag{17}$$

on the domain $x \in [0, \pi]$, using the mixed boundary conditions

$$u'(0) - u(0) = 0, \qquad u'(\pi) + u(\pi) = 0. \tag{18}$$

(a) Use a mesh width of $h = \pi/n$ where $n \in \mathbb{N}$ and introduce grid points at $x_j = jh$ for $j = 0, 1, \ldots, n$. Construct a second-order accurate finite-difference method for this BVP. Write your method as a linear system of the form $AU = F$.

(b) Construct the exact solution $u(x)$ to the BVP when $f(x) = -e^x$.

(c) Verify that your method is second-order accurate by solving the BVP with $f(x) = -e^x$ using $n = 20, 40, 80, 160$. For each $n$, construct the error measure

$$E_n = \sqrt{h \sum_{j=0}^{n} q_j (U_j - u(x_j))^2} \tag{19}$$

where $U_j$ is the numerical solution at $x_j$. Here, $q_j = \frac{1}{2}$ when $j \in \{0, n\}$ and $q_j = 1$ otherwise. Present your results in a table, and comment on whether the trend in the errors is expected for a second-order method.

(a) Consider the following:
As given in the problem statement, we have the differential equation

$$u''(x) + u(x) = f(x), \quad x \in [0, \pi]$$

with boundary conditions

$$u'(0) - u(0) = 0,$$
$$u'(\pi) + u(\pi) = 0.$$

We discretize the domain using a uniform grid with mesh width $h = \pi/n$ and grid points $x_j = jh$ for $j = 0, 1, \ldots, n$. Let $U_j$ be the approximation of $u(x_j)$. We will use a second-order central difference approximation for the second derivative at interior points. So for each interior point $j = 1, 2, \ldots, n-1$, we have

$$u''(x_j) \approx \frac{U_{j-1} - 2U_j + U_{j+1}}{h^2}.$$

Substituting this into the differential equation gives

$$\frac{U_{j-1} - 2U_j + U_{j+1}}{h^2} + U_j = f(x_j), \quad j = 1, 2, \ldots, n-1.$$

so the matrix (for the interior points) will be a tridiagonal matrix of the form

$$A = \frac{1}{h^2} \begin{pmatrix} -2+h^2 & 1 & 0 & \cdots & 0 \\ 1 & -2+h^2 & 1 & \cdots & 0 \\ 0 & 1 & -2+h^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -2+h^2 \end{pmatrix}.$$

Next, we need to incorporate the boundary conditions. For the first boundary condition at $x = 0$, we can use a the second order accurate forward difference approximation for $u'(0)$:

$$u'(0) \approx \frac{-3U_0 + 4U_1 - U_2}{2h}.$$

Substituting this into the boundary condition gives

$$\frac{-3U_0 + 4U_1 - U_2}{2h} - U_0 = 0$$
$$\frac{U_0(-3 - 2h) + 4U_1 - U_2}{2h} = 0.$$

We can then incorporate this into the first row of the matrix $A$ and the first entry of the vector $F$. The first row of $A$ becomes

$$A_{0,:} = \frac{1}{2h}\begin{pmatrix} -3-2h & 4 & -1 & 0 & \cdots & 0 \end{pmatrix}, F_0 = 0.$$

For the second boundary condition at $x = \pi$, we can use a second-order accurate backward difference approximation for $u'(\pi)$:

$$u'(\pi) \approx \frac{3U_n - 4U_{n-1} + U_{n-2}}{2h}.$$

Substituting this into the boundary condition gives

$$\frac{3U_n - 4U_{n-1} + U_{n-2}}{2h} + U_n = 0$$
$$\frac{U_n(3+2h) - 4U_{n-1} + U_{n-2}}{2h} = 0.$$

We can then incorporate this into the last row of the matrix $A$ and the last entry of the vector $F$. The last row of $A$ becomes

$$A_{n,:} = \frac{1}{2h}\begin{pmatrix} 0 & \cdots & 0 & 1 & -4 & 3+2h \end{pmatrix}, F_n = 0.$$

Thus, we have that the linear system $AU = F$ is given by

$$AU = \begin{pmatrix} \frac{-3-2h}{2h} & \frac{4}{2h} & \frac{-1}{2h} & 0 & \cdots & 0 \\ \frac{1}{h^2} & \frac{-2+h^2}{h^2} & \frac{1}{h^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{h^2} & \frac{-2+h^2}{h^2} & \frac{1}{h^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{h^2} & \frac{-2+h^2}{h^2} \\ 0 & \cdots & 0 & \frac{1}{2h} & \frac{-4}{2h} & \frac{3+2h}{2h} \end{pmatrix}\begin{pmatrix} U_0 \\ U_1 \\ U_2 \\ \vdots \\ U_{n-1} \\ U_n \end{pmatrix} = \begin{pmatrix} 0 \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n-1}) \\ 0 \end{pmatrix} = F.$$

Since we used second-order accurate finite difference approximations for both the interior points and the boundary conditions, the overall method is second-order accurate.

(b) To construct the exact solution to the BVP when $f(x) = -e^x$, we first solve the corresponding homogeneous equation

$$u''(x) + u(x) = 0.$$

The characteristic equation is $r^2 + 1 = 0$, which has solutions $r = i$ and $r = -i$. Thus, the general solution to the homogeneous equation is

$$u_h(x) = C_1 \cos x + C_2 \sin x,$$

where $C_1$ and $C_2$ are constants to be determined by the boundary conditions.

Next, we find a particular solution to the non-homogeneous equation. We can use the method of undetermined coefficients and try a particular solution of the form

$$u_p(x) = Ae^x,$$

where $A$ is a constant to be determined. Substituting this into the differential equation gives

$$Ae^x + Ae^x = -e^x,$$

which simplifies to

$$2Ae^x = -e^x.$$

Dividing both sides by $e^x$ (which is never zero), we find

$$2A = -1 \implies A = -\frac{1}{2}.$$

Thus, a particular solution is

$$u_p(x) = -\frac{1}{2}e^x.$$

The general solution to the non-homogeneous equation is then

$$u(x) = u_h(x) + u_p(x) = C_1 \cos x + C_2 \sin x - \frac{1}{2}e^x.$$

Now, we apply the boundary conditions to determine $C_1$ and $C_2$. The first boundary condition is

$$u'(0) - u(0) = 0.$$

We first compute $u'(x)$:

$$u'(x) = -C_1 \sin x + C_2 \cos x - \frac{1}{2}e^x.$$

Evaluating at $x = 0$ gives

$$\begin{aligned} u(0) &= C_1 - \frac{1}{2}, \\ u'(0) &= C_2 - \frac{1}{2}. \end{aligned}$$

Applying the first boundary condition:

$$\begin{aligned} u'(0) - u(0) = 0 &\implies (C_2 - \tfrac{1}{2}) - (C_1 - \tfrac{1}{2}) = 0 \\ &\implies C_2 - C_1 = 0 \\ &\implies C_1 = C_2. \end{aligned}$$

For the second boundary condition at $x = \pi$:

$$\begin{aligned} u(\pi) &= C_1 \cos \pi + C_2 \sin \pi - \tfrac{1}{2}e^\pi = -C_1 - \tfrac{1}{2}e^\pi, \\ u'(\pi) &= -C_1 \sin \pi + C_2 \cos \pi - \tfrac{1}{2}e^\pi = C_2(-1) - \tfrac{1}{2}e^\pi = -C_2 - \tfrac{1}{2}e^\pi. \end{aligned}$$

The boundary condition is $u'(\pi) + u(\pi) = 0$, so:

$$\begin{aligned} (-C_2 - \tfrac{1}{2}e^\pi) + (-C_1 - \tfrac{1}{2}e^\pi) &= 0 \\ -(C_1 + C_2) - e^\pi &= 0 \\ C_1 + C_2 &= -e^\pi. \end{aligned}$$

But from above, $C_1 = C_2$, so $2C_1 = -e^\pi$, hence $C_1 = C_2 = -\frac{1}{2}e^\pi$.

Therefore, the exact solution is

$$u(x) = -\frac{1}{2}e^\pi \cos x - \frac{1}{2}e^\pi \sin x - \frac{1}{2}e^x.$$

(c) See the attached code file `714Hw1.py` for the implementation of the finite difference method and the error calculation.

The results for $n = 20, 40, 80, 160$ are presented in the following table:

| $n$ | $E_n$ |
|---|---|
| 20 | $2.3788752785e-01$ |
| 40 | $6.0669230276e-02$ |
| 80 | $1.5289901217e-02$ |
| 160 | $3.8359976081e-03$ |

Observing the error values, we can see that as $n$ doubles, the error $E_n$ decreases by approximately a factor of 4. This is consistent with the expected behavior of a second-order accurate method, where the error is proportional to $h^2$. Since $h = \pi/n$, halving $h$ (doubling $n$) should reduce the error by a factor of 4, confirming that our finite difference method is indeed second-order accurate.

*Note: I also checked the second order accuracy visually by making a log-log plot of the error versus h and seeing that the slope was approximately 2.*

Problem 3

**Nonlinear BVP (4 points).** Consider the nonlinear BVP

$$u''(x) - 80\cos u(x) = 0 \tag{20}$$

with the boundary conditions $u(0) = 0$ and $u(1) = 10$. Define the mesh width $h = \frac{1}{n}$ for $n \in \mathbb{N}$, and introduce gridpoints $x_j = jh$ for $j = 0, 1, \ldots, n$.
Let $U_i$ be the approximation of $u(x_i)$. From the boundary conditions, $U_0 = 0$ and $U_n = 10$. Let $U = (U_1, U_2, \ldots, U_{n-1})$ be the vector of unknown function values, and write $F(U) = 0$ as the nonlinear system of algebraic equations from the finite-difference approximation, where $F : \mathbb{R}^{n-1} \to \mathbb{R}^{n-1}$. The components are

$$F_1(U) = \frac{U_2 - 2U_1}{h^2} - 80\cos U_1, \tag{21}$$

$$F_i(U) = \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} - 80\cos U_i \qquad \text{for } i = 2, \ldots, n-2, \tag{22}$$

$$F_{n-1}(U) = \frac{10 - 2U_{n-1} + U_{n-2}}{h^2} - 80\cos U_{n-1}. \tag{23}$$

a) Calculate the Jacobian $J_F \in \mathbb{R}^{(n-1)\times(n-1)}$ for the function $F$ and describe its structure.

b) Use Newton's method to solve the BVP for $n = 100$, using the Jacobian matrix from part (a). Write $U^k$ to indicate the $k$th Newton step, and start with the initial guess $U^0 = 0$. Terminate Newton's method when the relative step size $\|\Delta U^k\|_2 / \|U^k\|_2$ is less than $10^{-10}$. Plot the solution $U$ over the interval $[0, 1]$, and report the value of $U_{50}$ to three significant figures.

(a) We calculate the Jacobian matrix $J_F$ of the function $F$. The Jacobian matrix is defined as

$$(J_F)_{ij} = \frac{\partial F_i}{\partial U_j}.$$

We compute the partial derivatives for each component of $F$:

- For $i = 1$:

$$\frac{\partial F_1}{\partial U_1} = -\frac{2}{h^2} + 80\sin U_1,$$
$$\frac{\partial F_1}{\partial U_2} = \frac{1}{h^2},$$
$$\frac{\partial F_1}{\partial U_j} = 0 \quad \text{for } j > 2.$$

- For $2 \le i \le n-2$:

$$\frac{\partial F_i}{\partial U_{i-1}} = \frac{1}{h^2},$$
$$\frac{\partial F_i}{\partial U_i} = -\frac{2}{h^2} + 80\sin U_i,$$
$$\frac{\partial F_i}{\partial U_{i+1}} = \frac{1}{h^2},$$
$$\frac{\partial F_i}{\partial U_j} = 0 \quad \text{for } j < i-1 \text{ or } j > i+1.$$

- For $i = n-1$:

$$\frac{\partial F_{n-1}}{\partial U_{n-2}} = \frac{1}{h^2},$$
$$\frac{\partial F_{n-1}}{\partial U_{n-1}} = -\frac{2}{h^2} + 80\sin U_{n-1},$$
$$\frac{\partial F_{n-1}}{\partial U_j} = 0 \quad \text{for } j < n-2.$$

Thus, the Jacobian matrix $J_F$ has the following tridiagonal structure:

$$J_F = \begin{pmatrix} -\frac{2}{h^2} + 80\sin U_1 & \frac{1}{h^2} & 0 & \cdots & 0 \\ \frac{1}{h^2} & -\frac{2}{h^2} + 80\sin U_2 & \frac{1}{h^2} & \cdots & 0 \\ 0 & \frac{1}{h^2} & -\frac{2}{h^2} + 80\sin U_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{2}{h^2} + 80\sin U_{n-1} \end{pmatrix}.$$

(b) For implementation of Newton's method, look at the attached code file `714Hw1.py`. The function `problem3` implements the finite difference approximation and Newton's method to solve the BVP. The solution is plotted over the interval $[0, 1]$,

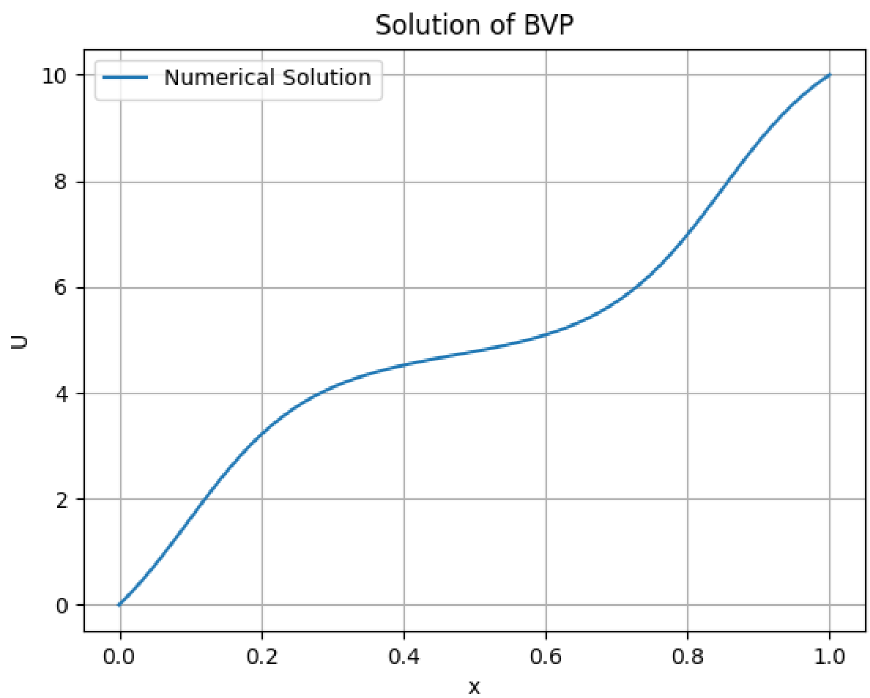


Figure 2: Plot of the solution $U$ over the interval $[0, 1]$.

and the value of $U_{50}$ is reported as `U[50] to three significant figures:  4.78`.

Problem 4

**FD in a triangular domain (8 points).** Let $T$ be a domain in the shape of an equilateral triangle with vertices at $(0,0)$, $(1,0)$, and $(\frac{1}{2}, s)$ where $s = \frac{\sqrt{3}}{2}$. For $n \in \mathbb{N}$, define $h = \frac{1}{n}$, and introduce grid points

$$\mathbf{x}_{i,j} = (h(i + \tfrac{1}{2}j), hsj)) \tag{24}$$

for $0 \le i \le n$, $0 \le j \le n-i$. An example grid for $n = 7$ is shown in Fig. **??**. The grid points on the boundary $\partial T$ correspond to $i = 0$, $j = 0$, or $i + j = n$. All other points are defined as a interior points.

(a) Let $u : T \to \mathbb{R}$ be a smooth function, and write $u_{i,j} = u(\mathbf{x}_{i,j})$. For an interior point $\mathbf{x}_{i,j}$, consider the finite difference approximation

$$\nabla_3^2 u_{i,j} = \alpha u_{i,j} + \beta u_{i+1,j} + \gamma u_{i,j-1} + \delta u_{i-1,j+1}. \tag{25}$$

Derive the values of $\alpha$, $\beta$, $\gamma$, and $\delta$ so that

$$\nabla_3^2 u_{i,j} = \nabla^2 u(\mathbf{x}_{i,j}) + hW + O(h^2) \tag{26}$$

and determine the form of $W$ in terms of partial derivatives of $u$. By considering the function $u(x,y) = x^3$ show that $\nabla_3^2 u_{i,j}$ is a first-order accurate approximation for $\nabla^2 u(\mathbf{x}_{i,j})$, but it is not second-order accurate.

(b) Using your result from part (a), or otherwise, determine the constants $c_0, c_1, \ldots, c_6$ such that the finite difference approximation

$$\begin{aligned}\nabla_6^2 u_{i,j} = {} & c_0 u_{i,j} + c_1 u_{i+1,j} + c_2 u_{i,j-1} + c_3 u_{i-1,j+1} \\ & + c_4 u_{i-1,j} + c_5 u_{i,j+1} + c_6 u_{i+1,j-1}\end{aligned} \tag{27}$$

satisfies $\nabla_6^2 u_{i,j} = \nabla^2 u(\mathbf{x}_{i,j}) + O(h^2)$.

(c) Write a program to solve the equation

$$\nabla^2 u = f \tag{28}$$

on the domain $T$ using the boundary conditions that $u(\mathbf{x}) = 0$ for $\mathbf{x} \in \partial T$. Using the method of manufactured solutions, determine the value of $f(x,y)$ such that the solution will be

$$u^{\text{ex}}(x,y) = \Big((2y - \sqrt{3})^2 - 3(2x-1)^2\Big) \sin y. \tag{29}$$

Consider values of $n = 10, 20, 40, 80, 160$, and compute the error measure

$$E_n = \sqrt{\frac{s}{2n^2} \sum_{j=1}^{n-1} \sum_{i=1}^{n-j-1} (u_{i,j} - u^{\text{ex}}(\mathbf{x}_{i,j}))^2}. \tag{30}$$

Make a log–log plot of $E_n$ versus $n$, and use linear regression to fit the data to $E_n = Cn^{-p}$, reporting your values of $C$ and $p$ to three significant figures.

(d) Repeat part (c) for the finite difference approximation given in Eq. (27).

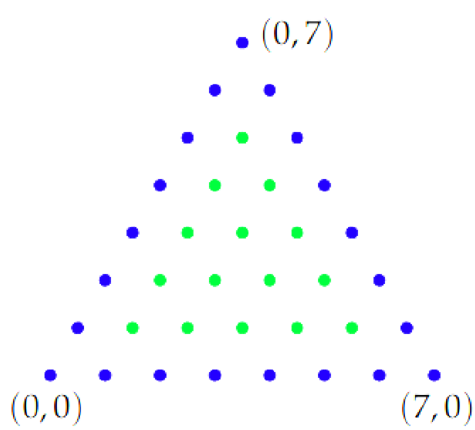


Figure 1: Example of the triangular domain $T$ and numerical grid for the case of $n = 7$. The corner points are labeled with their grid indices $(i, j)$. Blue circles denote the boundary points, and green circles denote the interior points.

(a) First, Taylor expand each neighbor about the point $\mathbf{x}_{i,j}$:

$$\begin{aligned} u_{i+1,j} &= u_{i,j} + hu_x + \frac{h^2}{2}u_{xx} + \frac{h^3}{6}u_{xxx} + O(h^4), \\ u_{i,j-1} &= u_{i,j} - hsu_y + \frac{h^2s^2}{2}u_{yy} - \frac{h^3s^3}{6}u_{yyy} + O(h^4), \\ u_{i-1,j+1} &= u_{i,j} - hu_x + hsu_y + \frac{h^2}{2}(u_{xx} + s^2u_{yy} - 2su_{xy}) \\ &\quad - \frac{h^3}{6}(u_{xxx} - s^3u_{yyy} + 3s^2u_{yyx} - 3su_{yxx}) + O(h^4). \end{aligned}$$

Then, with substitution, we arrive at

$$u(\mathbf{x}+e) = u + \nabla u \cdot e + \frac{1}{2}e^T He + \frac{1}{6}\sum_{i,j,k}\frac{\partial^3 u}{\partial x_i \partial x_j \partial x_k}e_ie_je_k + O(\|e\|^4),$$

where $H$ is the Hessian matrix of second derivatives of $u$ at $\mathbf{x}$ and $e$ is the vector of displacements. Then we can see that $\alpha+\beta+\gamma+\delta = 0$ to eliminate the zeroth order term. Similarly, we can set the coefficients of the first derivatives to zero to eliminate the first order terms:

$$\frac{1}{2}\sum_{\phi\in\{\alpha,\beta,\delta\}}\phi e_{\phi,x}^2 = 1, \quad \frac{1}{2}\sum_{\phi\in\{\alpha,\beta,\delta\}}\phi e_{\phi,y}^2 = 1, \qquad \sum_{\phi\in\{\alpha,\beta,\delta\}}\phi e_{\phi,x}e_{\phi,y} = 0.$$

From these equations we can recover the coefficients:

$$\begin{aligned} \alpha &= -\frac{4}{h^2}, \\ \beta = \gamma = \delta &= \frac{4}{3h^2}. \end{aligned}$$

Thus we can see that we have:

$$\nabla_3^2 u_{i,j} = -\frac{4}{h^2}u_{i,j} + \frac{4}{3h^2}(u_{i+1,j} + u_{i,j-1} + u_{i-1,j+1} + u_{i-1,j} + u_{i,j+1} + u_{i+1,j-1}).$$

Next we find the leading error term $W$. We can see that the linear terms cancel out ($\beta e_1 + \gamma e_2 + \delta e_3 = 0$ when $\beta = \gamma = \delta$ and $e_1 + e_2 + e_3 = 0$), so the next non-zero terms come from the cubic terms. If we let $e_k = hv_k$, then collecting the cubic terms gives:

$$\nabla_3^2 u_{i,j} = \Delta u_{i,j} = hW(\mathbf{x}_{i,j}) + O(h^2),$$

where

$$W = \frac{1}{6}u_{xxx} - \frac{1}{2}u_{xyy}.$$

To show that this is a first-order accurate approximation but not second-order accurate, we can consider the function $u(x,y) = x^3$. Then we have $u_{xxx} = 6$ and all other third derivatives are zero. So, $W = \frac{1}{6}\cdot 6 - \frac{1}{2}\cdot 0 = 1$. Thus, the leading error term is $hW = h \cdot 1 = h$, which shows that the approximation is first-order accurate. However, since $W$ is non-zero, the approximation cannot be second-order accurate.

(b) Using all six neighbors, we can set up a system of equations to determine the coefficients $c_0, c_1, \ldots, c_6$. We want to eliminate the zeroth and first order terms, and match the second order terms. First, we can notice symmetry in the coefficients:

$$c_1 = c_4, \quad c_2 = c_5, \quad c_3 = c_6.$$

Then, we can set up the following equations:

$$\begin{aligned} c_0 + \sum_{i=1}^{6} c_i &= 0, \\ \sum_{i=1}^{6} c_i e_{i,x}^2 &= 2, \\ \sum_{i=1}^{6} c_i e_{i,y}^2 &= 2, \end{aligned}$$

Then, letting $c_1 = c_4 = a$, $c_2 = c_5 = b$, and $c_3 = c_6 = c$, we have

$$\begin{aligned} 2a + \frac{b+c}{2} &= \frac{2}{h^2}, \\ 2s^2(b+c) &= \frac{2}{h^2} \implies b + c = \frac{1}{h^2 s^2}. \end{aligned}$$

Solving these equations and using $s^2 = \frac{3}{4}$, we find that $a = \frac{2}{3h^2}$, $b = \frac{2}{3h^2}$, and $c = \frac{2}{3h^2}$. Then we can see that all the neighbors have the same coefficient i.e., $\frac{2}{3h^2}$, and we can find $c_0$:

$$c_0 = -\sum_{i=1}^{6} c_i = -6 \cdot \frac{2}{3h^2} = -\frac{4}{h^2}.$$

Finally, we have

$$c_0 = -\frac{4}{h^2}, \quad c_1 = c_2 = c_3 = c_4 = c_5 = c_6 = \frac{2}{3h^2}.$$

We can also notice that all the cubic terms cancel out due to symmetry, so the leading error term is $O(h^2)$, confirming that this is a second-order accurate approximation.

(c) See the attached code file `714Hw1.py` for the implementation of the finite difference method and error calculation. Thhe figure below shows the log-log plot of $E_n$ versus $h$ for the first-order accurate method from part (a). Using linear regression, we find that $C = 0.088$ and $p = -0.000803$ of 3 significant figures.

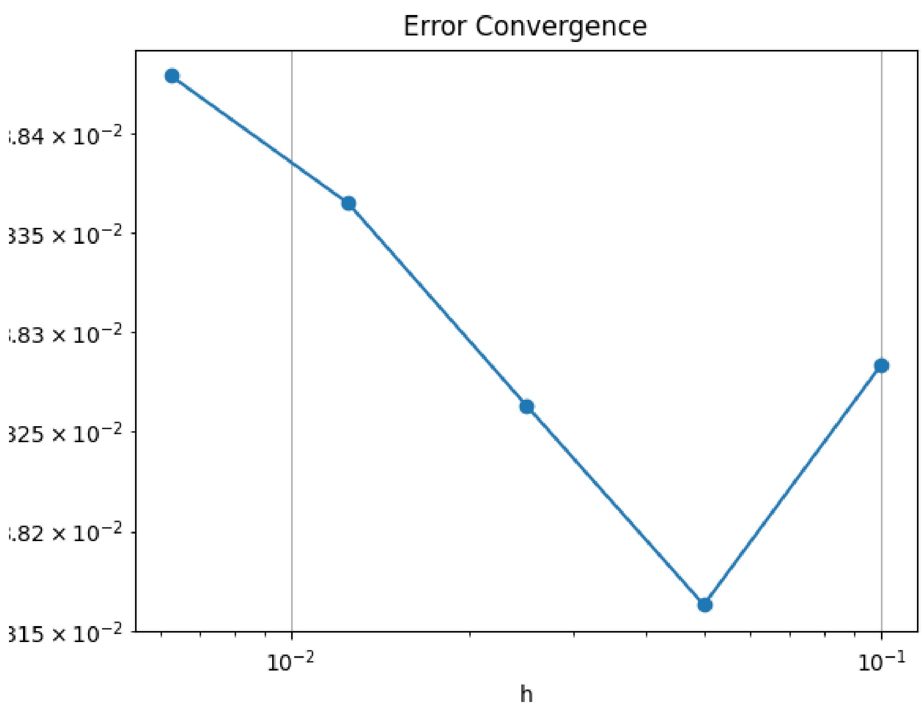


Figure 3: Log-log plot of $E_n$ versus $h$ for the first-order accurate method.

(d) For implementation of the second-order accurate method from part (b), see the attached code file `714Hw1.py`. (I basically used all of the same functions expect for the finite difference approximation function) The figure below shows the log-log plot of $E_n$ versus $h$ for the second-order accurate method. Using linear regression, we find that $C = 0.088$ and $p = -0.000803$ of 3 significant figures.

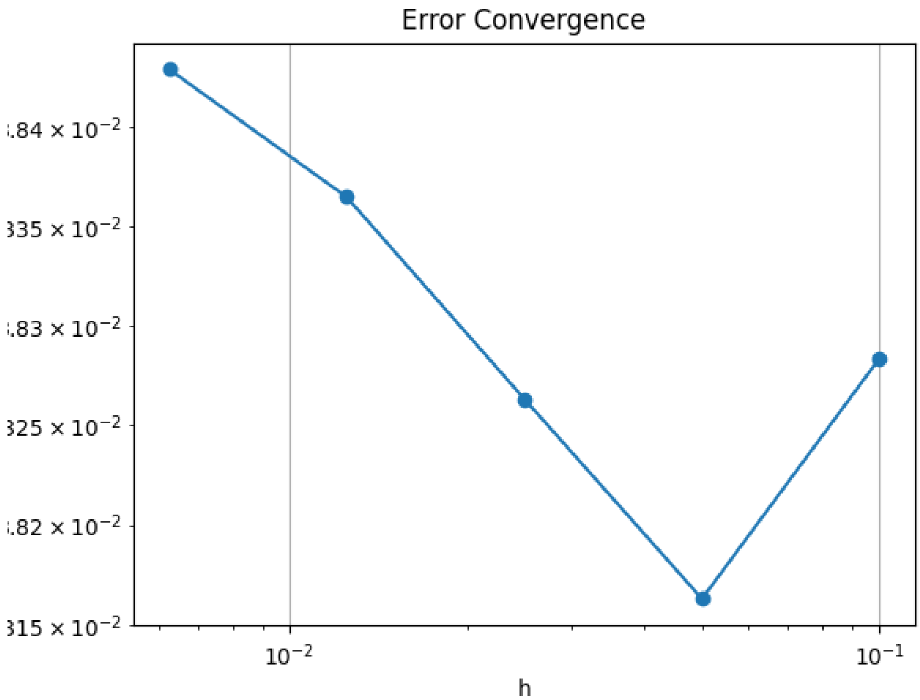


Figure 4: Log-log plot of $E_n$ versus $h$ for the second-order accurate method.

*Note:* I am not confident in my answers for problem 4 as I lost my notes for that problem and was in a rush to finish the write up so I just tried to re-derive everything from scratch. Sorry!