CS 714

HOMEWORK 1

Noah Cohen Kalafut
Computer Science Doctoral Student
University of Wisconsin-Madison
nkalafut@wisc.edu
https://github.com/Oafish1/CSC-714

October 6, 2020

1 Problem A

1.1 Part a

We know three main things

$$\mu u(x) = \frac{u(x + h/2) + u(x - h/2)}{2} \tag{1}$$

$$\delta u(x) = u(x + h/2) - u(x - h/2) \tag{2}$$

$$u(x \pm h) = e^{\pm hD} u \tag{3}$$

Through substitution with equations 2 and 3, we can calculate

$$\delta u(x) = u(x + h/2) - u(x - h/2) = e^{\frac{hD}{2}} u - e^{\frac{-hD}{2}} u$$
$$\delta = e^{\frac{hD}{2}} - e^{\frac{-hD}{2}}$$

Applying further operations to both sides provides

$$\delta^2 = e^{hD} - 2e^0 + e^{-hD} = e^{hD} + e^{-hD} - 2$$
$$1 + \frac{1}{4}\delta^2 = 1 + \frac{e^{hD} + e^{-hD} - 2}{4} = \frac{e^{hD} + e^{-hD} + 2}{4}$$
$$\left(1 + \frac{1}{4}\delta^2\right)^{-\frac{1}{2}} = \frac{2}{\sqrt{e^{hD} + e^{-hD} + 2}}$$

Using a similar method to δ, using equations 1 and 3, we can determine that

$$\mu = \frac{e^{\frac{hD}{2}} + e^{\frac{-hD}{2}}}{2}$$

Notice

$$\left(e^{\frac{hD}{2}} + e^{\frac{-hD}{2}}\right)^2 = e^{hD} + 2e^0 + e^{-hD} = e^{hD} + e^{-hD} + 2$$
$$e^{\frac{hD}{2}} + e^{\frac{-hD}{2}} = \sqrt{e^{hD} + e^{-hD} + 2}$$

So,

$$\left(1 + \frac{1}{4}\delta^2\right)^{-\frac{1}{2}} = \frac{2}{\sqrt{e^{hD} + e^{-hD} + 2}} = \frac{2}{e^{\frac{hD}{2}} + e^{\frac{-hD}{2}}} = \mu^{-1}$$

Then, we can see that

$$\mu\left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}} = \mu\mu^{-1} = 1$$

1.2 Part b

The odd powers of δ require samples of u not provided on the grid (i.e. $\ldots x-\frac{3h}{2}, x-\frac{h}{2}, x+\frac{h}{2}, x+\frac{3h}{2}\ldots$).

1.3 Part c

We have previously proven

$$hD = \delta - \frac{\delta^3}{24} + \frac{3\delta^5}{640} - \frac{5\delta^7}{7168}\cdots \tag{4}$$

$$1 = \mu\left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}} \tag{5}$$

We also know $\mu\delta = hD_c$ and δ^2 are on the grid. Multiplying equations 4 and 5 provides

$$\begin{aligned}
hD &= \mu\left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}}\left(\delta - \frac{\delta^3}{24} + \frac{3\delta^5}{640} - \frac{5\delta^7}{7168}\cdots\right)\\
&= \left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}}\left(\mu\delta - \frac{\mu\delta^3}{24} + \frac{3\mu\delta^5}{640} - \frac{5\mu\delta^7}{7168}\cdots\right)\\
&= \left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}}\mu\delta - \left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}}\frac{\mu\delta^3}{24} + \left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}}\frac{3\mu\delta^5}{640}\cdots
\end{aligned}$$

Note that each individual term can now be calculated on the grid, thereby providing a proper scheme after truncation

$$\left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}}\mu\delta = \left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}} * \mu\delta$$

$$\left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}}\frac{\mu\delta^3}{24} = \frac{1}{24} * \left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}} * \mu\delta * \delta^2$$

$$\left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}}\frac{3\mu\delta^5}{640} = \frac{3}{640} * \left(1+\frac{1}{4}\delta^2\right)^{-\frac{1}{2}} * \mu\delta * \delta^4$$

$$\cdots$$

2 Problem B

We are given

$$u(x) = x_+^n = \begin{cases} x^n & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$D_c u(-\epsilon) = \frac{u(-\epsilon+h) - u(-\epsilon-h)}{2h}$$

Note that

$$u'(x) = \begin{cases} nx^{n-1} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

2.1 Part a

The scheme will be consistent for the ℓ_∞ norm if and only if $\lim_{h\to 0}||\tau^h||_\infty = 0$.

2.1.1 General Cases

We can start by observing

$$D_c u(-\epsilon + mh) = \frac{(-\epsilon + (m+1)h)_+^n - (-\epsilon + (m-1)h)_+^n}{2h}$$

In the case that $m > 1$, since $0 < \epsilon < h$, we know D_c does not straddle the origin. So, $x_+^n = x^n$ and $\frac{\delta}{\delta x} x_+^n = nx^{n-1}$. Expanding $(-\epsilon + (m \pm 1)h)_+^n$ provides

$$(-\epsilon + (m \pm 1)h)_+^n = (-\epsilon + mh)^n \pm hn(-\epsilon + mh)^{n-1} + \frac{h^2}{2} n(n-1)(-\epsilon + mh)^{n-2} \pm \dots$$

$$(-\epsilon + (m+1)h)_+^n - (-\epsilon + (m-1)h)_+^n = 2h \left(n(-\epsilon + mh)^{n-1} + \frac{h^2}{6} \frac{n!}{(n-3)!} (-\epsilon + mh)^{n-3} + \dots \right)$$

$$D_c u(-\epsilon + mh) = n(-\epsilon + mh)^{n-1} + \frac{h^2}{6} \frac{n!}{(n-3)!} (-\epsilon + mh)^{n-3} + \frac{h^4}{6} \frac{n!}{(n-5)!} (-\epsilon + mh)^{n-5} + \dots$$

Since $D_c u(x)$ is a first-order derivative approximation, we can determine $f(-\epsilon + mh) = u'(-\epsilon + mh) = n(-\epsilon + mh)_+^{n-1}$ and

$$\tau_j = D_c u(-\epsilon + mh) - f(-\epsilon + mh) = \frac{h^2}{6} \frac{n!}{(n-3)!} (-\epsilon + mh)^{n-3} + \frac{h^4}{6} \frac{n!}{(n-5)!} (-\epsilon + mh)^{n-5} + \dots$$

$$\tau_j = O(h^2)$$

The case of $m < 0$ is trivial, since both $f(x)$ and $D_c u(-\epsilon + mh)$ would be 0, making $\tau_j = 0$ and $||\tau||_\infty = 0$.

2.1.2 Contentious Cases

With the general cases out of the way, the contentious cases $m = 0, 1$ can be handled. Note that $u(x)$ is infinitely differentiable but not continuous in these cases due to a discrepancy at the n$^{\text{th}}$ derivative.

In the case that $m = 0$, where D_c straddles the origin, we can say

$$D_c u(-\epsilon) = \frac{(-\epsilon + h)_+^n - (-\epsilon - h)_+^n}{2h} = \frac{(-\epsilon + h)_+^n}{2h} = \frac{(-\epsilon + h)^n}{2h}$$

Since $h > \epsilon > 0$, ϵ is dependent on h and

$$D_c u(-\epsilon) \approx O(\frac{\epsilon^n + h^n}{2h}) \approx O(h^{n-1})$$

Here, since $-\epsilon < 0$, $f(-\epsilon) = 0$ and

$$\tau_j = D_c u(-\epsilon) - f(-\epsilon) = D_c u(-\epsilon)$$

$$\tau_j \approx O(h^{n-1})$$

Similarly, in the case that $m = 1$, where D_c also straddles the origin,

$$D_c u(-\epsilon + h) = \frac{(-\epsilon + 2h)_+^n - (-\epsilon)_+^n}{2h} = \frac{(-\epsilon + 2h)_+^n}{2h} = \frac{(-\epsilon + 2h)^n}{2h}$$

Since $h > \epsilon > 0$,

$$^1 D_c u(-\epsilon + h) \approx O(\frac{2^n h^n}{2h}) \approx O((2h)^{n-1}) \approx O(h^{n-1})$$

Here, since $-\epsilon + h \geq 0$, $f(-\epsilon + h) = n(-\epsilon + h)^{n-1}$ and

$$f(-\epsilon + h) \approx O(nh^{n-1}) \approx O(h^{n-1})$$

$$\tau_j = D_c u(-\epsilon) - f(-\epsilon) \approx O(h^{n-1}) - O(h^{n-1}) \approx O(h^{n-1})$$

Empirically, it makes sense that $n = 0, 1$ are problem cases as they are not continuous either in general or at the first derivative.

[1] We only really care about convergence as $h \to 0$ here, so the 2^{n-1} has no bearing.

2.1.3 Conclusion

For all m, for $n \geq 2$, $\tau_j \approx O(h^p)$ for some $p > 0$.

Thus, for all m, for $n \geq 2$, D_c is consistent and is $\begin{cases} O(h^2) & n > 3 \\ O(h^{n-1}) & n \leq 2 \end{cases}$ order accurate in the ℓ_∞ norm.

2.2 Part b

Let's examine each case in the ℓ_1 norm.

When $m > 1$, $\tau_j \approx O(h^2)$. Consistent for all $n \geq 0$.

When $m < 0$, $\tau_j = O(0)$. Consistent for all $n \geq 0$.

When $m = 0, 1$, $\tau_j \approx O(h^{n-1})$. Consistent for all $n \geq 1$.

Notice, the ℓ_1 norm is defined as $h \sum_j |\tau_j|$. This means that a term corresponding to $\tau_j \approx O(h^p)$ would be $O(h^{p+1})$ in the ℓ_1 norm, potentially allowing $||\tau||_1 \to 0$ as $h \to 0$. This is why $\tau_j \approx O(h^{n-1})$ is consistent for all $n \geq 1$.

For any n, it would make sense that the order of accuracy of ℓ_1 would be proportional to the term with the lowest power of h.

So, for all m, for $n \geq 1$, D_c is consistent and is $\begin{cases} O(h^3) & n > 3 \\ O(h^n) & n \leq 2 \end{cases}$ order accurate in the ℓ_1 norm.

2.3 Part c

At $n = 3$ in the ℓ_∞ norm, the error is $\frac{h^2}{6} \frac{n!}{(n-3)!} = h^2$ for h sufficiently small. This seems just right. For $n = 2$, the first derivative increases linearly, so D_c will be totally accurate since $(x - h, u(x - h))$ and $(x + h, u(x + h))$ form a line segment tangent to x^2. For $n = 3$, however, this is not the case, and the first derivative increases proportional to $O(x^2)$.

3 Problem C

This problem has Neumann boundary conditions on the top and bottom in the form of insulation. There are Dirichlet boundary conditions on the left and right, in the form of some pattern of electrical potential $f(y)$ on the left and grounding on the right.

3.1 Part a

We can use the second-order finite difference discretization for both x and y as a starting point.

$$\Delta u(x_i, y_j) = (\frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2})u(x_i, y_j) \approx \frac{u(x_{i-1}, y_j) - 2u(x_i, y_j) + u(x_{i+1}, y_j)}{h_x^2} + \frac{u(x_i, y_{j-1}) - 2u(x_i, y_j) + u(x_i, y_{j+1})}{h_y^2}$$

Choosing $h_x = h_y$ allows us to simplify

$$-\widetilde{\Delta} u(x_i, y_j) = \frac{4u(x_i, y_j) - u(x_{i-1}, y_j) - u(x_{i+1}, y_j) - u(x_i, y_{j-1}) - u(x_i, y_{j+1})}{h^2} \approx 0 \tag{6}$$

Notice that equation 6 implies that $u(x, y)$ is the average of its surrounding samples; this will be used in the next problem.

Dirichlet boundary conditions are easy to implement, as they provide a known value at the point in question. For these, we are given the conditions

$$u(0, y) = f(x), u(1, y) = 0$$

Neumann boundary conditions are harder, as we do not have a known value. As was done in section 2.12 of [1], we can take the centered-difference discretization for the first derivative of $u(x, 0)$ and set it to 0, in accordance with our

Neumann boundary condition. This will provide a second-order accurate approximation of the derivative with respect to y.

$$\frac{\delta}{\delta y}u(x,0) = 0 \approx \frac{u(x,y_1) - u(x,y_{-1})}{2h} \tag{7}$$

We can then solve for $u(x,y_{-1})$

$$u(x,y_{-1}) = u(x,y_1)$$

And plug it into equation 6 for y_0

$$\begin{aligned} -\widetilde{\Delta}u(x_i,y_0) &= 4u(x_i,y_0) - u(x_{i-1},y_0) - u(x_{i+1},y_0) - u(x_i,y_{-1}) - u(x_i,y_1) \\ &= 4u(x_i,y_0) - u(x_{i-1},y_0) - u(x_{i+1},y_0) - 2u(x_i,y_1) \end{aligned}$$

The same can be done for the boundary condition $\frac{\delta}{\delta y}u(x,1) = 0$.

We can then summarize everything in matrix form. For A, I_A of size $n \times n$ and B, I_B of size $(n+2) \times (n+2)$, where n is the number of non-boundary samples for set x or y, let

$$A = \frac{1}{h^2}\begin{bmatrix} 0 & -1 & 0 & \\ -1 & 0 & -1 & \\ 0 & -1 & 0 & \ddots \\ & & \ddots & \end{bmatrix}, B = \frac{1}{h^2}\begin{bmatrix} 4 & -1 & 0 & \\ -1 & 4 & -1 & \\ 0 & -1 & 4 & \ddots \\ & & \ddots & \ddots \end{bmatrix}, A \otimes I_B + I_A \otimes B = \begin{bmatrix} B & I & 0 & \\ I & B & I & \\ 0 & I & B & \ddots \\ & & \ddots & \ddots \end{bmatrix}$$

Then

$$\frac{1}{h^2}\begin{bmatrix} B & I & 0 & 0 & \\ I & B & I & 0 & \\ 0 & I & B & I & \\ 0 & 0 & I & B & \ddots \\ & & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} u(x_1,y_0) \\ u(x_1,y_1) \\ \vdots \\ u(x_1,y_{n+1}) \\ \hline u(x_2,y_0) \\ u(x_2,y_1) \\ \vdots \\ u(x_2,y_{n+1}) \\ \hline u(x_n,y_0) \\ u(x_n,y_1) \\ \vdots \\ u(x_n,y_{n+1}) \end{bmatrix} = \begin{bmatrix} -\widetilde{\Delta}u(x_1,y_0) \\ -\widetilde{\Delta}u(x_1,y_1) \\ \vdots \\ -\widetilde{\Delta}u(x_1,y_{n+1}) \\ \hline -\widetilde{\Delta}u(x_2,y_0) \\ -\widetilde{\Delta}u(x_2,y_1) \\ \vdots \\ -\widetilde{\Delta}u(x_2,y_{n+1}) \\ \hline -\widetilde{\Delta}u(x_n,y_0) \\ -\widetilde{\Delta}u(x_n,y_1) \\ \vdots \\ -\widetilde{\Delta}u(x_n,y_{n+1}) \end{bmatrix} + \begin{bmatrix} +f(y_0) + u(x_1,y_1) \\ +f(y_1) \\ \vdots \\ +f(y_{n+1}) + u(x_1,y_n) \\ \hline +u(x_2,y_1) \\ 0 \\ \vdots \\ +u(x_2,y_n) \\ \hline +0 + u(x_n,y_1) \\ +0 \\ \vdots \\ +0 + u(x_n,y_n) \end{bmatrix} \tag{8}$$

Notice, B takes a weighted sum of the original point and the two neighbors on the y axis. I takes the sum of the two x axis neighbors. We can also drop the $\frac{1}{h^2}$ term in practice since the Laplacian is supposed to be 0.

Also notice, the boundary conditions are implemented on the right-hand side of equation 8. The Neumann boundary conditions could have also been implemented by adding -1 to cells $B_{1,2}$ and $B_{n+2,n+1}$ – which is useful when considering the iterative approach. This also could have been done by using a second-order accurate forward-approximation on the first and last rows of B for the Neumann boundary condition. This would also require sorting by y rather than x in the second matrix above – which can be done anyway by switching the dimensions of B and A if someone so desired. However, the same result is acquired in any case.

3.2 Part b

First, an array was constructed with the Dirichlet boundary conditions. Note, $f(x) = f(2\pi - x)$.

```
% Iterative approach
U = zeros(n+2, n+2);
% Dirichlet BC
for i = 1:n+2
    U(i, 1) = f((i-1) * h);
end
```

For each iteration, the solver then replaced each cell $(x_1, y_0 \dots x_n, y_{n+1})$ with the average of its neighbors according to the Gauss-Seidel method.

```
% Gauss-Seidel
for it = 1:iter
    % Neumann BC
    for j = 2:n+1
        U(1, j) = (1/4)*(2*U(2, j) + U(1, j+1) + U(1, j-1));
        U(n+2, j) = (1/4)*(2*U(n+1, j) + U(n+2, j+1) + U(n+2, j-1));
    end

    % Main iteration loop
    for i = 2:n+1
        for j = 2:n+1
            U(i, j) = (1/4)*(U(i+1, j) + U(i-1, j) + U(i, j+1) + U(i, j-1));
        end
    end
end
```

Notice that the Neumann boundary conditions are accounted for by replacing $u(x_i, y_-1)$ with $u(x_i, y_1)$ and $u(x_i, y_{n+2})$ with $u(x_i, y_n)$.

Alternatively, the for loop can be replaced with a while loop that ends when the change in error or change in error dips below a certain threshold.

By running this method until convergence for very fine h and using the result to compute error for larger h, figure 1 can be created.

Figure 1: Maximum Norm of the Error vs h

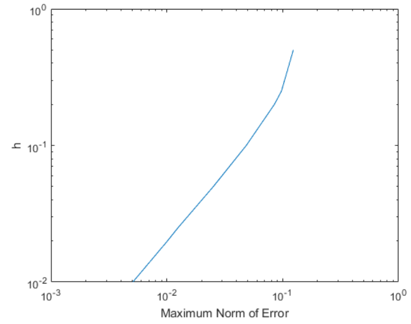
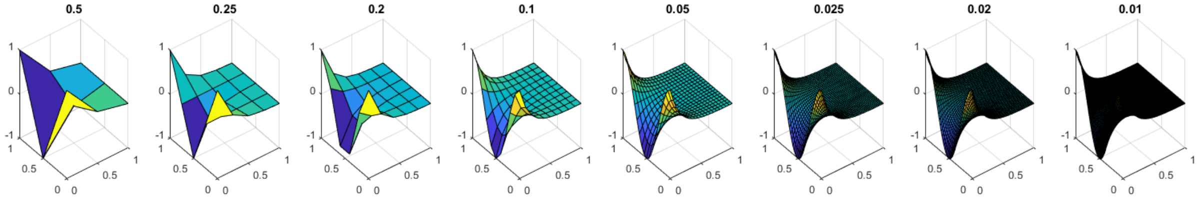


Figure 2: Sample calculations of $u(x, y)$ for certain h



3.3 Part c

All of the components used in the scheme are second-order accurate (and, therefore, consistent). It stands to reason that the scheme would also then be consistent. More formally, given that u is infinitely diffirentiable, the taylor expansion of equation 6 provides $-\widetilde{\Delta} u(x_i, y_j) = O(h^2)$. For the Neumann boundary, equation 7 is also second-order accurate.

3.4 Part d

Let A, B have eigenvalues, eigenvectors λ, Λ and μ, M, respectively.

We know $-\widetilde{\Delta} = A \oplus B = A \otimes I_B + I_A \otimes B$.

Consider

$$\begin{aligned}(A \oplus B)(\lambda \otimes M) =& \begin{bmatrix} A_{1,1}I_B + B & A_{1,2}I_B & \dots \\ A_{2,1}I_B & A_{2,2}I_B + B & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \Lambda_1 M \\ \Lambda_2 M \\ \vdots \end{bmatrix} \\ =& \begin{bmatrix} B\Lambda_1 M + A_{1,1}\Lambda_1 M + A_{1,2}\Lambda_2 M + \dots \\ B\Lambda_2 M + A_{2,1}\Lambda_1 M + A_{2,2}\Lambda_2 M + \dots \\ \vdots \end{bmatrix} \\ =& \begin{bmatrix} \mu\Lambda_1 M + (A\Lambda)_1 M \\ \mu\Lambda_2 M + (A\Lambda)_2 M \\ \vdots \end{bmatrix} = \begin{bmatrix} \mu\Lambda_1 M + \lambda\Lambda_1 M \\ \mu\Lambda_2 M + \lambda\Lambda_2 M \\ \vdots \end{bmatrix} \\ =&(\mu + \lambda)(\Lambda \otimes M)\end{aligned}$$

This can be proven for all combinations $\lambda + \mu$, of which there are $n(n+2)$.

For a more formal proof that also confirms that there are no additional eigenvalues, we can use a modified proof from Theorem 42 of [2].

Suppose A^h and B^h are diagonalizable. Then there exist some R, ν, S, ξ such that

$$\begin{aligned} A^h &= R\nu R^{-1} \\ B^h &= S\xi S^{-1} \end{aligned}$$

Note that ν, ξ are matrices with the eigenvalues of A^h, B^h on the diagonal.

Then,

$$\begin{aligned} &(R \otimes S)^{-1}(A^h \oplus B^h)(R \otimes S) \\ =&(R^{-1} \otimes S^{-1})(A^h \otimes I_B + I_A \otimes B^h)(R \otimes S) \\ =&(R^{-1} \otimes S^{-1})(A^h \otimes I_B)(R \otimes S) + (R^{-1} \otimes S^{-1})(I_A \otimes B^h)(R \otimes S) \\ =&(R^{-1}A^hR \otimes S^{-1}I_BS) + (R^{-1}I_AR \otimes S^{-1}B^hS) \\ =&(\nu \otimes I_B) + (I_A \otimes \xi) \end{aligned}$$

Which implies that $(\nu \otimes I_B) + (I_A \otimes \xi)$ has the eigenvectors of $A^h \oplus B^h$ on the diagonal. Notice that the diagonal of $(\nu \otimes I_B) + (I_A \otimes \xi)$ is comprised of all possible $\mu + \lambda$.

So,

$$A \oplus B \text{ has eigenvalues } \lambda_1 + \mu_1, \lambda_1 + \mu_2, \dots, \lambda_1 + \mu_j, \lambda_2 + \mu_1, \dots, \lambda_i + \mu_j. \tag{9}$$

3.5 Part e

Since both A^h and B^h are tridiagonal Toeplitz matrices, we can determine their eigenvalues [3].

$$A^h \text{ has eigenvalues } \lambda_p = \frac{1}{h^2}\left(-2\cos(\frac{p\pi}{n+1})\right) \text{ for } p = 1, 2, \dots, n.$$

$$B^h \text{ has eigenvalues } \mu_q = \frac{1}{h^2}\left(4 - 2\cos(\frac{q\pi}{n+3})\right) \text{ for } q = 1, 2, \dots, n+2.^2$$

[2] B^h is of size $(n+2) \times (n+2)$

Suppose, for the sake of contradiction, that $-\widetilde{\Delta}$ has some eigenvalue less than or equal to 0.

Then, using statement 9,

$$\lambda(-\widetilde{\Delta}) = \lambda_p + \mu_q = \frac{1}{h^2}\left(4 - 2\cos(p\pi h) - 2\cos(q\pi h)\right) \leq 0 \text{ for some } p, q$$

This implies $\cos(\frac{p}{n+1}\pi) + \cos(\frac{q}{n+3}\pi) \geq 2$, which is only possible if $\frac{p}{n+1}, \frac{q}{n+3}$ are even integers.

Since $p \in (0, n+1)$ and $q \in (0, n+3)$, this is impossible.

Thus, all eigenvalues of $-\widetilde{\Delta}$ are positive and non-zero.

Further, recall $h = \frac{1}{n+1}$

$$\lambda_{min}(-\widetilde{\Delta}) = (n+1)^2\left(4 - 2\cos(\frac{\pi}{n+1}) - 2\cos(\frac{\pi}{n+3})\right) \tag{10}$$

Notice that, as $h \to 0$, $\lambda_{min}(-\widetilde{\Delta}) \to \infty$.

Then, we know that all eigenvalues of $-\widetilde{\Delta}$ are non-zero and $\lambda_{min}(-\widetilde{\Delta})$ goes to infinity as $h \to 0$.

So, there must exist some C independent of h for which $\lambda_{min}(-\widetilde{\Delta}) \geq C > 0$.

3.6 Part f

For readability, let's say $Z = A \oplus B$.

In accordance with section 4.2 of [4], from $Z = M - N = D - L - U$, we can take

$$M = D - L = \frac{1}{h^2}\begin{bmatrix} B & 0 & 0 & 0 & \\ I & B & 0 & 0 & \\ 0 & I & B & 0 & \\ 0 & 0 & I & B & \ddots \\ & & & \ddots & \ddots \end{bmatrix}, N = U = -\frac{1}{h^2}\begin{bmatrix} 0 & I & 0 & 0 & \\ 0 & 0 & I & 0 & \\ 0 & 0 & 0 & I & \\ 0 & 0 & 0 & 0 & \ddots \\ & & & \ddots & \ddots \end{bmatrix}$$

The iterative method $Mu^{[k+1]} = Nu^{[k]} + u''$ is then equivalent to the Gauss-Seidel method. We can then derive

$$\begin{gathered} e^{[k+1]} = G^k e^{[0]} \\ G = M^{-1}N = (D-L)^{-1}U \end{gathered}$$

using the same method as in [4].

So, we can find the spectral radius $\rho = \rho(G)$.

For the Jacobi method, this is relatively easy. We get $G = D^{-1}(D - Z) = I - D^{-1}Z = I - \frac{h^2}{4}Z$ which has eigenvalues $\lambda(G) = 1 - \frac{h^2}{4}\lambda(Z)$, providing the spectral radius $\rho(G) = 1 - \frac{h^2}{4}\lambda_{min}(-\widetilde{\Delta})$ from equation 10.

However, for the Gauss-Seidel method, this is much less straight-forward. Take a look at the following adapted proof from section 13.2.3 of [5].

The eigenvalues, eigenvectors λ, Λ of G must satisfy

$$\begin{gathered} (D-L)^{-1}U\Lambda = \lambda\Lambda \\ \left((D-L)^{-1}U - \lambda\right)\Lambda = 0 \\ (U - (D-L)\lambda)\,\Lambda = 0 \end{gathered}$$

Which we can split into components by row

$$\Lambda_{j+1,k} + \Lambda_{j,k+1} - 4\lambda\Lambda_{j,k} + \lambda\Lambda_{j-1,k} + \lambda\Lambda_{j,k-1} = 0$$

Note that, although Λ is a vector, it represents a 2D space. Hence, the double subscripts. This makes visualization easier in the next steps. You'll notice that this is very similar to our initial FD scheme.

We can substitute $\Lambda_{j,k} = \lambda^{(j+k)/2} r_{j,k}$ as is done in [6] in order to attempt finding a general solution.

$$\lambda^{(j+k+1)/2} r_{j+1,k} + \lambda^{(j+k+1)/2} r_{j,k+1} - 4\lambda\lambda^{(j+k)/2} r_{j,k} + \lambda\lambda^{(j+k-1)/2} r_{j-1,k} + \lambda\lambda^{(j+k-1)/2} r_{j,k-1} = 0$$
$$\lambda^{1/2} r_{j+1,k} + \lambda^{1/2} r_{j,k+1} - 4\lambda r_{j,k} + \lambda\lambda^{-1/2} r_{j-1,k} + \lambda\lambda^{-1/2} r_{j,k-1} = 0$$
$$4\lambda r_{j,k} = \lambda^{1/2} \left(r_{j+1,k} + r_{j,k+1} + r_{j-1,k} + r_{j,k-1} \right)$$
$$\lambda^{1/2} r_{j,k} = \frac{1}{4} \left(r_{j+1,k} + r_{j,k+1} + r_{j-1,k} + r_{j,k-1} \right)$$

As is stated in equation 9.2.17c of [6], we can propose the solution that

$$r_{j,k} = \sin\left(\frac{pj\pi}{n+1}\right) \sin\left(\frac{qk\pi}{n+3}\right)$$
$$\lambda^{1/2} = 1 - \sin^2\left(\frac{p\pi}{2(n+1)}\right) - \sin^2\left(\frac{q\pi}{2(n+3)}\right)$$
$$p = 1, 2, \ldots, n; q = 1, 2, \ldots, n+2$$

Thus, the spectral radius can be written as

$$\begin{aligned} \rho(G) &= \left(1 - \sin^2\left(\frac{\pi}{2(n+1)}\right) - \sin^2\left(\frac{\pi}{2(n+3)}\right)\right)^2 \\ &\approx \left(1 - \frac{\pi^2}{4(n+1)^2} - \frac{\pi^2}{4(n+3)^2} + O(h^4)\right)^2 \\ &\approx 1 - \frac{\pi^2}{2(n+1)^2} - \frac{\pi^2}{2(n+3)^2} + O(h^4) \end{aligned} \tag{11}$$

Since M is lower diagonal with a non-zero diagonal, M^{-1} exists and is real. So, G is normal. Then, we know from [4] that

$$G = R^{-1} \Gamma R$$

$$||e^{[k]}|| \leq \rho^k ||e^{[0]}|| \tag{12}$$

If we want to approximate the solution to a specified error ϵ, we can say

$$||e^{[k]}|| \leq \epsilon ||e^{[0]}||$$
$$\epsilon \approx \rho^k$$

It follows that

$$k \approx \frac{\log(\epsilon)}{\log(\rho)} \tag{13}$$

Combining equations 11 and 13 gives an approximation of the number of iterations needed to achieve accuracy under any specified error ϵ.

3.6.1 Analysis

As can be seen from figure 3, for all h and ϵ not especially large (in other words, all practical cases), the actual number of iterations needed is generally lower than the predicted value $\frac{\log(\epsilon)}{\log(\rho)}$. This makes sense, as 12 was designed to get an upper-bound of the error. The estimation not only takes the worst case evaluation of Γ, ρ, but our calculation of k in equation 13 assumes the initial error vector $1_{n*(n+2)}$ when ϵ is used as our error threshold.[3] These two things combined mean that the estimation uses the worst-case dampening on the largest error possible – something that will rarely be the case for regular boundary conditions.

Figure 3: Iteration Prediction Error and $\frac{\text{True Iterations}}{\text{Predicted Iterations}}$ under the ℓ_∞ norm

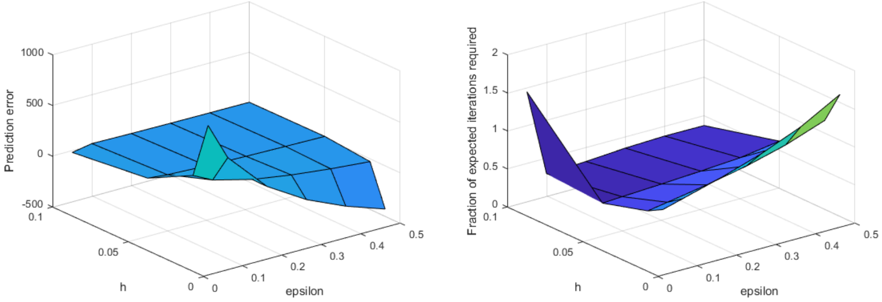
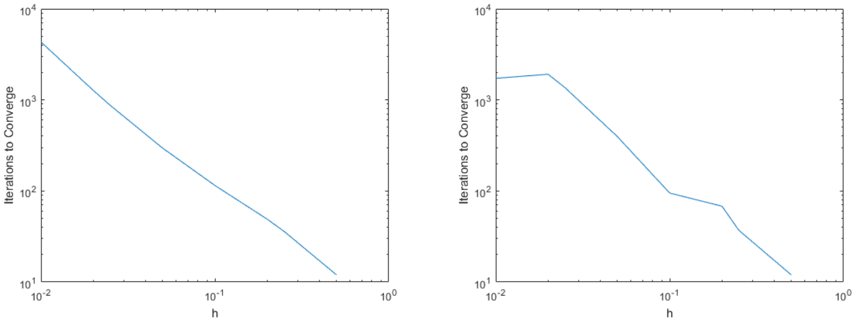


Figure 4: Iterations to converge for given h on a log-log scale. $f(y)$ and $\hat{f}(y)$, respectively.

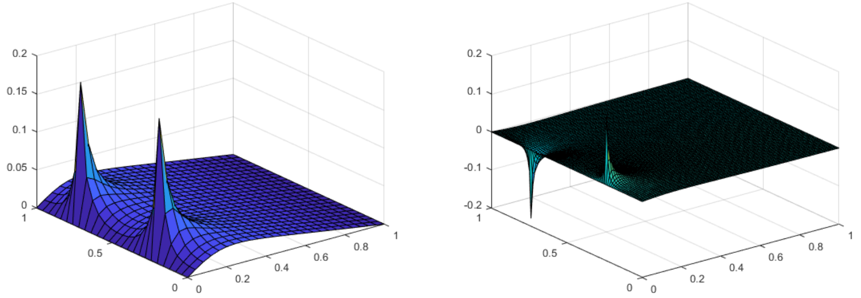


3.7 Part g

As seen in figure 4, the order of convergence seems to remain the same at $O(\frac{1}{h^2})$. Γ does not change with f or $\hat{f}$ since it only relies on the iteration matrix. Therefore, the spectral radius remains the same and so does the order of convergence.

This is a difficult problem to analyze experimentally. The undefined magnitude current (∇u) where $(x, y) = (0, {}^1\!/\!{}_4)$ or $(0, {}^3\!/\!{}_4)$ frequently creates problems such as the one seen in figure 5j, where a grid's error when compared to a fine estimation might not decrease with h. However, it remains that the approximation still seems to converge to the true solution (which is still an approximation) at the same rate as its counterpart.

Figure 5: Difference between $h = .04$ and $h = .02$, $h = .01$ and $h = .005$ approximations, respectively

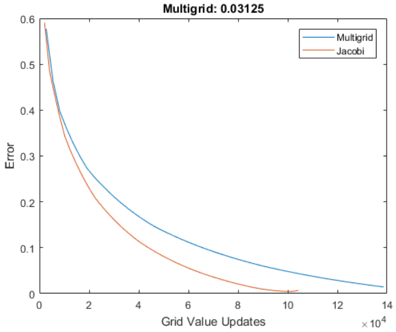


[3] Where 1_m is the vector of length m consisting of only ones.

4 Problem D

This implementation of V-cycle seems to converge slightly slower than unweighted Jacobi, as shown in figure 6. This is likely a problem with the solutions for $\widetilde{e}$ not having enough of an effect on the original matrix, as the result in figure 6 looks approximately the same as underrelaxed Jacobi.

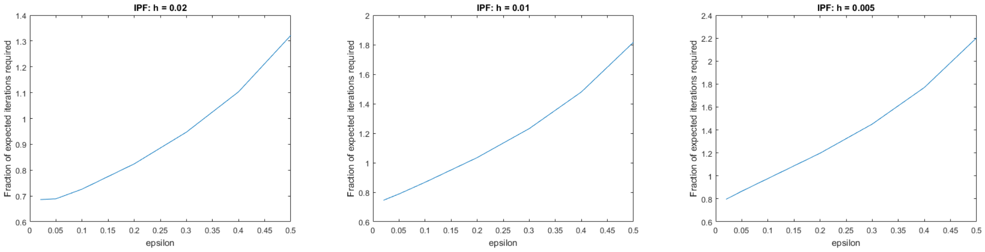
Figure 6: Difference between $h = .04$ and $h = .02$ approximations



5 Problem E

Examining the error of the approximation with respect to a numerical solution on the same grid approximated to convergence (update norms within a certain threshold) provided figure 7.

Figure 7: $\frac{\text{True Iterations}}{\text{Predicted Iterations}}$ under the ℓ_∞ norm for $h = .02, .01, .005$



It can be observed that the relation found in equation 11 and 13 still holds for ϵ sufficiently small. It is worth noting that $\frac{\text{True Iterations}}{\text{Predicted Iterations}}$ is generally higher in this case even for small ϵ, likely because many elements in $e^{[0]}$ begin larger in magnitude (especially near $x = 0$) due to the more extreme boundary condition.

Notice that higher epsilon values take more iterations than predicted. This could be a product of the boundary conditions as well. $\widehat{f(y)}$ is not continuous at $y = {}^1\!/\!_4, {}^3\!/\!_4$. This can be problematic when computing $\frac{\delta}{\delta y^2}$ near the boundary $x = 0$. Since f is $\mathcal{C}^\infty$, this is not a problem in the normal case.

References

[1] Randall J. LeVeque. *2. Steady States and Boundary Value Problems*, chapter 2, pages 13–58.

[2] Bobbi Jo Broxson. *The Kronecker Product*, page 35. 2006.

[3] Randall J. LeVeque. *C. Eigenvalues and Inner-Product Norms*, chapter C, pages 269–283.

[4] Randall J. LeVeque. *4. Analysis of Matrix Splitting Methods*, chapter 4, pages 69–110.

[5] Randall J. LeVeque. *13. Multigrid Methods*, chapter 13, pages 407–449.

[6] Joseph E. Flaherty. *Solution Techniques for Elliptic Problems*, page 35. 2012.