AMATH 585
Shabab Ahmed
Take home final
Due Tuesday, March 17, 2020

Due to Canvas by 11:00pm PDT on the due date, at the latest.

No late papers accepted, so aim to get it in earlier!

To submit, see `https://canvas.uw.edu/courses/1352870/assignments/5301318`

100 points possible.

Problem 1.

In Homework 2 you solved a linear beam equation, valid for small deformations. If the deformations are larger, then the equation must be nonlinear. In some cases an equation of this form can be used:

$$\begin{aligned} &au''''(x) - b(u'(x))^2(u''(x))^2 = f(x) \quad \text{for } 0 \le x \le 1 \\ &u(0) = \alpha_0, \;\; u'(0) = \alpha_1, \quad u(1) = \beta_0, \;\; u'(1) = \beta_1. \end{aligned}$$

where $a,\ b,\ \alpha_0,\ \alpha_1,\ \beta_0,\ \beta_1$ are all specified constants.

(a) If we discretize with a uniform grid using $h = 1/(m+1)$, suggest a nonlinear system of m equations that could be solved for $[U_1,\ U_2,\ \ldots,\ U_m]$ (the interior grid values) to obtain an approximate solution that is second order accurate.

Make sure the boundary conditions are also handled appropriately. Use the "First approach" described in hw2_solutions.html since this is less messy and should still be second order accurate.

(b) If we wanted to solve this system using Newton's method, we would need the Jacobian matrix for the nonlinear system developed in (a). You don't need to compute the full matrix, but do compute the diagonal elements J_{ii} for $i = 1,\ 2,\ \ldots,\ m$.

Solution:

Consider the equation

$$\begin{aligned} &au''''(x) - b(u'(x))^2(u''(x))^2 = f(x) \quad \text{for } 0 \le x \le 1 \\ &u(0) = \alpha_0, \;\; u'(0) = \alpha_1, \quad u(1) = \beta_0, \;\; u'(1) = \beta_1. \end{aligned}$$

where $a,\ b,\ \alpha_0,\ \alpha_1,\ \beta_0,\ \beta_1$ are all specified constants.

Suppose we discretize it with a uniform grid using $h = \frac{1}{m+1}$. We will use the following second order approximations:

$$u'(x_j) = \frac{1}{2h}(u(x_{j+1}) - u(x_{j-1}),$$

$$u''(x_j) = \frac{1}{h^2}(u(x_{j-1} - 2u(x_j) + u(x_{j+1})$$

and

$$u^4(x_j) = \frac{1}{h^4}(u(x_{j-2}) - 4u(x_{j-1}) + 6u(x_j) - 4u(x_{j+1}) + u(x_{j+2}))$$

We approximate $u(x_j)$ by U_j. So we know $U_0 = \alpha_0$ and $U_{m+1} = \beta_0$. However, we also have to deal with the left boundary and we do this by using a one sided approximation to DU_0 given by:

$$-\frac{1}{2h}(3U_0 - 4U_1 + U_2) - \alpha_1 = 0 \implies -\frac{1}{2h}(-4U_1 + U_2) - \alpha_1 - \frac{3\alpha_0}{2h} = 0$$

using the fact $U_0 = \alpha_0$. So, $G_1(U) = -\frac{1}{2h}(-4U_1 + U_2) - \alpha_1 - \frac{3\alpha_0}{2h}$

Now, we deal with the right boundary using another one sided approximation to the derivative:

$$\frac{1}{2h}(3U_{m+1} - 4U_m + U_{m-1}) - \beta_1 = 0 \implies \frac{1}{2h}(-4U_m + U_{m-1}) - \beta_1 + \frac{3\beta_0}{2h} = 0$$

using that $U_{m+1} = \beta_0$. So, $G_m(U) = \frac{1}{2h}(-4U_m + U_{m-1}) - \beta_1 + \frac{3\beta_0}{2h}$

For $j = 2, 3, \dots, m-1$ we use the centered second order accurate approximations mentioned above to get:

$$G_j(U) = \frac{a}{h^4}(U_{j-2} - 4U_{j-1} + 6U_j - 4U_{j+1} + U_{j+2}) - b[\frac{1}{2h}(U_{j+1} - U_{j-1})]^2[\frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1})]^2 - f(x_j)$$

Then, the system to be solved is $G(U) = 0$ where $G : \mathbb{R}^m \to \mathbb{R}^m$ and components are given by expressions above.

The centered fourth derivative approximation used is second order accurate and the truncation error is given by $\frac{bh^2}{6}u(6)(x) + \mathcal{O}(h^8)$. Also the truncation error for the derivative and the second derivative are also second order and so they have the form: $Ch^2 + \mathcal{O}(h^4)$. In this case, the truncation error will be squared and so they will take the form of C_1h^2 + higher order terms since there will be terms like $(u''(x_j)$ in the expression being squared. We then multiply the truncation errors, so they take the form C_2h^2 + higher order terms because some constant terms and h^2 terms from the squared expression multiply together. Then we add it truncation error of fourth derivative approximation. Therefore, the truncation error we get will be of the form Eh^2 for some constant E and so the system will be second order accurate.

b)

$$J_{ii} = \frac{dG_i(U)}{dU_i}$$

$i = 1$:

$$J_{11} = \frac{dG_1(U)}{dU_1} = \frac{2}{h}$$

$i = m$:

$$J_{mm} = \frac{dG_m(U)}{dU_m} = -\frac{2}{h}$$

For $i = 2, 3, \dots, m-1$:

$$J_{ii} = \frac{dG_i(U)}{dU_i} = \frac{6a}{h^4} - b[\frac{1}{2h}(U_{j+1} - U_{j-1})]^2 \times 2[\frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1})] \times (-2)$$

$$= \frac{6a}{h^4} + \frac{4b}{4h^4}(U_{j+1} - U_{j-1})^2(U_{j-1} - 2U_j + U_{j+1})$$

$$= \frac{6a + b}{h^4}(U_{j+1} - U_{j-1})^2(U_{j-1} - 2U_j + U_{j+1})$$

Problem 2.

Suppose A is a 2×2 singular matrix that is symmetric and has one postive eigenvalue, for example

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

is one such matrix. Then A is symmetric positive *semi-definite* ($u^T Au \geq 0$ for all nonzero u) but is not positive definite. If we define the functional

$$\phi(u) = \frac{1}{2}u^T Au - u^T f$$

as in Section 4.3 then levels sets of $\phi(u)$ are no longer ellipses.

(a) What is the geometry of these level sets in general?

(b) Let $z \in \mathbb{R}^2$ be a null vector of A and suppose $f \in \mathbb{R}^2$ is a vector satisfying $z^T f = 0$. Then the system $Au = f$ has infinitely many solutions. Show that the steepest descent method applied to $\phi(u)$ from any initial guess $u^{[0]}$ converges to *some* solution of the linear system in a single iteration. Hint: It might be easier to explain this using the result of (a) than by computing $u^{[1]}$ explicitly. That's fine as long as you give a convincing argument.

Solution:

a)

In this case, $\phi(u)$ will not have an unique minimum point. So, there will be several $\phi(u*)$ for which $\nabla\phi(u) = 0$. We can think of the level sets being straight lines in 2D because one of the eigenvalues will be zero. So, the level sets would be straight vertical lines in general.

b)

Let z be a null vector so $Az = 0$. Suppose $z^T f = 0$. We start on one of the level sets, which are straight lines. In the steepest descent algorithm we move in the direction ϕ is decreasing most rapidly. So, we go opposite direction of $\nabla\phi(u) = Au - f$. If we start with an intial condition, we will move in the direction orthogonal to contour lines. The line orthogonal to contour lines will be in the direction of one of the minimum values. This is because the minimum will be in the middle of all level sets which are straight vertical lines. This is going to be a vertical line of minimum points. If we are on a straight vertical line, direction orthogonal would be horizontal which would take us to the vertical line of minimum points (by taking negative of gradient of $\phi(u)$, direction of most descent) and we would hit one of the minimum points. So, we converge in one step.

Problem 3.

Consider the *first order ODE*, with a single boundary condition,

$$\begin{aligned} &u'(x) = f(x), \quad 0 \le x \le 1, \\ &u(0) = \alpha. \end{aligned}$$

This boundary value problem has a unique solution $u(x) = \alpha + \int_0^x f(t)\,dt$. Using $u'(x_j) \approx (U_j - U_{j-1})/h$ on a uniform grid, we might attempt to approximate it by solving a system of the form

$$\frac{1}{h}\begin{bmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & -1 & 1 \end{bmatrix}\begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_m \end{bmatrix} = \begin{bmatrix} \alpha/h \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_m) \end{bmatrix}.$$

(a) Determine the exact solution to this linear system and show that U_j approximates $\alpha + \int_0^{x_j} f(t)\,dt$ with a "Riemann sum".

(b) Suppose we apply Gauss-Seidel, sweeping through the grid from left to right in the natural order (i.e. $j = 0,\ 1,\ \ldots,\ m$). Explain why one iteration is sufficient to converge to the exact solution of this linear system, for *any* choice of initial data $U^{[0]}$.

(c) Suppose we instead sweep from right to left in Gauss-Seidel (for $j = m,\ m-1,\ \ldots 0$). What is the iteration matrix G for this method on the system above? What are the eigenvalues of the G matrix? What are the matrices $G^2,\ G^3,\ \ldots$? (There is a simple pattern.)

(d) Suppose we start with initial guess $U^{[0]} = 0$ (the zero vector). Does this backward Gauss-Seidel method converge in a single step? In a finite number of steps? If so, how many?

(e) In Section 4.2.1 we saw that the spectral radius $\rho(G)$ tells us something about the rate of convergence of the method. Comment on what's going on in this example based on your answers to (c) and (d).

You are welcome to write a short code to try things out, but you are not required to implement this.

Solution:

a) The solution is given by $u(x) = \alpha + \int_0^x f(t)dt = \alpha + F(t)|_0^x = \alpha + F(x) - F(0)$ where $f(x) = F'(x)$ using the Fundamental Theorem of Calculus.

We will show that U_j approximates $\int_0^{x_j} f(t)dt$.

Notice: $U_0 = \alpha$.

Then:

$$U_1 - U_0 = f(x_1)hU_1 = f(x_1)h + \alpha$$

$$U_2 - U_1 = hf(x_2) \implies U_2 = hf(x_2) + U_1 = hf(x_2) + hf(x_1) + \alpha$$

$$U_3 - U_2 = hf(x_3) \implies U_3 = hf(x_3) + U_2 = hf(x_3) + hf(x_2) + U_1 = hf(x_3) + hf(x_2) + hf(x_1) + \alpha = \alpha + \sum_{i=1}^{3} hf(x_i)$$

Following this pattern we can see that

$$U_j - U_{j-1} = hf(x_j) \implies U_j = hf(x_j) + U_{j-1} = hf(x_j) + hf(x_{j-1}) + U_{j-2}$$

and we expand U_{j-2} giving us U_{j-3} and so on. We keep unpacking to get the following:

$$U_j = \alpha + \sum_{i=1}^{j} hf(x_i)$$

So U_j approximates $\alpha + \int_0^{x_j} f(t)dt$ with a "Riemann sum" because if we let $h \to 0$ we will get $U_j = \alpha + \int_0^{x_j} f(t)dt$. We are not going till m so we do not get integral from 0 to 1 but we only get integral from 0 to x_j because we are using the first j bins for Riemann Sum. The number that i goes upto determines the interval, for example, if we take the interval [a,b] and divide it into n, $n = \frac{b-a}{\delta x}$. In our case we are summing upto j and our $\delta x = h \implies jh = b - a$. But we are going from 0 ($a = 0$) and so $jh = b$ and the way we discretize it, we have that $b = x_j$. Therefore, our result makes sense. **b)**

Let $U^{[0]}$ be any choice of initial data.

Suppose we apply Gauss-Seidel, sweeping through the grid from left to right in the natural order.

Using the technique of matrix splitting we can write

$$A = \frac{1}{h}\begin{bmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & -1 & 1 \end{bmatrix}$$

as $A = M - N$ where M consists of the diagonal and strictly lower triangular part of A and N is the strictly upper triangular part of A. However, notice that A has no upper triangular part and so N must be a matrix will all zero entries. Also, M is actually the matrix A because A only has non zero entries in the places captured by M

Then, $G = M^{-1}N$ will also be a matrix with all zero entries. Therefore, when we use it to update our error we get:

$$e^{[1]} = Ge^{[0]} = \mathbf{0}$$

So, we must have reached the exact solution of the linear system. Therefore, the system converges in one iteration. We can also see this in the process of updating the initial data:

$$U^{[1]} = GU^{[0]} + M^{-1}f = M^{-1}f = A^{-1}f$$

which is exactly the solution of the linear system and we reached it in one iteration.

c)

Suppose we instead sweep from right to left, that is use backward Gauss-Seidel. We again use the technique of splitting the matrix A, but this time we M and N differently. We let M to be the diagonal and strictly upper triangular part of A. However, since A does not have a strictly upper triangular part we have that M is just the diagonal part of A. We also let N be the strictly lower triangular part of A.

Therefore,

$$M = \frac{1}{h}\begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}$$

and

$$N = \frac{1}{h}\begin{bmatrix} 0 & & & & & \\ -1 & & & & & \\ & -1 & & & & \\ & & -1 & & & \\ & & & \ddots & & \\ & & & & -1 & 0 \end{bmatrix}$$

.

So,

$$M^{-1} = h\begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}$$

because it is just a diagonal matrix and the inverse is a diagonal matrix with the reciprocals of the diagonal entries of M.

Finally, $G = M^{-1}N = hN$. That is,

$$G = \begin{bmatrix} 0 & & & & & \\ -1 & & & & & \\ & -1 & & & & \\ & & -1 & & & \\ & & & \ddots & & \\ & & & & -1 & 0 \end{bmatrix}$$

.

This can be thought of as a lower triangular matrix with all diagonal entries equal to zero and so the eigenvalues of G must all be zero.

$$G^2 = \begin{bmatrix} 0 & & & & & \\ -1 & & & & & \\ & -1 & & & & \\ & & -1 & & & \\ & & & \ddots & & \\ & & & & -1 & 0 \end{bmatrix}\begin{bmatrix} 0 & & & & & \\ -1 & & & & & \\ & -1 & & & & \\ & & -1 & & & \\ & & & \ddots & & \\ & & & & -1 & 0 \end{bmatrix}.$$

$$= \begin{bmatrix} 0 & & & & & \\ 0 & & & & & \\ 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & 0 & 0 \end{bmatrix}$$

$$G^3 = \begin{bmatrix} 0 & & & & & \\ 0 & & & & & \\ 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & & & & & \\ -1 & & & & & \\ & -1 & & & & \\ & & -1 & & & \\ & & & \ddots & & \\ & & & & -1 & 0 \end{bmatrix}.$$

$$= \begin{bmatrix} 0 & & & & & \\ 0 & & & & & \\ 0 & & & & & \\ -1 & & & & & \\ & \ddots & & & & \\ & & -1 & 0 & 0 & 0 \end{bmatrix}$$

The pattern continues. For G^k, the 1st column of the $(k+1)$th row will contain the only non zero entry of the form $(-1)^k$ of the 1st column. The second column will only have the same non zero entry at the $(k+2)$th row. These non-zero entries follow a diagonal until they reach the last row. All other entries are non zero. Therefore, the first $(m-k)$ columns will have only one non zero entry of the form $(-1)^k$ and for the jth column this will be located at the $(k+j)$th row fo r $j = 1, 2, \ldots, m-k$. Continuing like this for k, we will have that G^k is the zero matrix for $k = m$.

d)

Suppose we start with the initial guess $U^{[0]}$ the zero vector.

Then,

$$e^{[0]} = f - AU^{[0]} = f$$

since $U^{[0]}$ is the zero vector.

It does not converge in one step because $U^{[1]} = GU^{[0]} + M^{-1}f = hf$. Then, $AU^{[1]} = Ahf \neq f$. Then, we know that $e^{[k]} = G^k e^{[0]}$.

Therefore, $e^{[k]} = G^k f$. Since, f is non zero for $e^{[k]}$ to be zero we need $G^k = 0$. From our observation in c) we know that $G^m = 0$, So, $e^{[m]} = 0$ and the backward Gauss Seidel will converge in finitely many steps. In particular, it will converge in m steps.

e)

Section 4.2.1 tells us that the spectral radius $\rho(G)$ tells us the something about the rate of convergence. The section tells us that,

$$\|e^{[k]}\|_2 \leq \kappa(R)\rho(G)^k \|e^{[0]}\|_2$$

and since $\rho(G) = 0$ in this we should expect this to converge in one step. We see in our case that it does not converge in one step. However, the analysis done in 4.2.1 is based on the fact that the matrix is diagonalizable. However, A in our case is clearly not because all eigenvalues are zero and so we cannot split it into right eigenvectors and a diagonal matrix of eigenvalues because this diagonal matrix will just be a zero matrix. Therefore, this analysis does not really hold. However, the spectral radius is still giving some indication because the spectral radius of G is less than 1 and we see that the error decreases to zero in a finite number of steps. We do not see the error growing out of bounds.

Problem 4.

Consider the Conjugate gradient algorithm on page 87 for some symmetric positive definite matrix $A \in \mathbb{R}^{m\times m}$. Suppose we happen to choose the initial guess u_0 in such a way that the initial residual $r_0 = f - Au_0$ is an eigenvector of A. Show that in this case the method converges to the true solution of the linear system in one iteration.

Solution:

We consider the CG algorithm on page 87 for some symmetric positive definite matrix $A \in \mathbb{R}^{m\times m}$.

Suppose, we happen to choose the initial gues u_0 in such a way that the initial residual $r_0 = f - Au_0$ is an eigenvector of A. So, we know, there exists a constant λ such that $Ar_0 = \lambda r_0$. We want to show that method converges to the true solution of the linear system in one iteration. Let the subscripts denote the number of iteration.

This means that u_1 will be the true solution of the system. Recall, we are solving $Au = f$ and so that would mean $Au_1 = f$. In turn, this would mean that the residual is a zero vector since $r_1 = f - Au_1 = f - f = \mathbf{0}$.

Given, $p_0 = r_0$:

$$w_0 = Ap_0 = Ar_0 = \lambda r_0$$

since r_0 is an eigenvector of A.

Then,

$$\alpha_0 = \frac{r_0^T r_0}{p_0^T w_0} = \frac{r_0^T r_0}{r_0^T(\lambda r_0})} = \frac{r_0^T r_0}{\lambda(r_0^T r_0)} = \frac{1}{\lambda}$$

since $r_0^T r_0 = \|r\|_2^2$ is just going to be a constant we can cancel it out of the fraction.

So,

$$u_1 = u_0 + \alpha_0 p_0 = u_0 + \frac{1}{\lambda} r_0$$

Multiplying by A on the left:

$$Au_1 = A(u_0 + \frac{1}{\lambda} r_0) = Au_0 + \frac{1}{\lambda} Ar_0 = (f - r_0) + \frac{1}{\lambda}\lambda r_0 = f - r_0 + r_0 = f$$

since $Ar_0 = \lambda r_0$ and $r_0 = f - Au_0 \implies Au_0 = f - r_0$. Therefore, $Au_1 = f$ we have that the system converges in one iteration.

We can also check that with $u_1 = u_0 + \frac{1}{\lambda} r_0$ gives us that $r_1 = r_0 - \alpha_0 w_0 = r_0 - \frac{1}{\lambda}\lambda r_0 = r_0 - r_0 = 0$.

Problem 5.

Consider the BVP

$$u''(x) = f(x), \qquad \text{for } 0 \le x \le 1$$

with boundary conditions

$$\gamma_0 u(0) + \gamma_1 u'(0) = \sigma, \qquad u(1) = \beta.$$

At $x = 1$ a standard Dirichlet BC is specified, but at $x = 0$ we now have a "mixed" or "Robin" boundary condition, assuming $\gamma_0,\ \gamma_1,\ \sigma$ are all specified constants, as is β. For some physical problems this is the correct type of boundary condition, e.g. in a heat conduction problem it corresponds to a case in which the heat flux at $x = 0$ is related to the temperature at this point.

(a) Set up a tridiagonal linear system $Au = f$ that could be solved to model this, with the following properties:

- $u = [u_0,\ u_1,\ \dots,\ u_m]$ contains the unknown boundary value u_0 but not the known value $u_{m+1} = \beta$ (assuming as usual that $x_j = jh$ for $j = 0,\ 1,\ \dots,\ m+1$ on a grid with $h = 1/(m+1)$).
- The method is second order accurate.

Follow the strategy of the second approach on page 31 to obtain the first equation in your linear system (i.e. introduce a ghost point u_{-1} and then eliminate it from the two equations that involve this unknown). Write out the matrix and right hand side of your system.

(b) Determine the local truncation error of your method, $\tau = [\tau_0,\ \tau_1,\ \dots,\ \tau_m]$. We expect $\tau_j = C_j h^2 + o(h^2)$ so determine the constants C_j in terms of derivative(s) of the exact solution $u(x)$ (by doing Taylor series expansions, assuming $u(x)$ is sufficiently smooth).

Solution:

a) Consider the BVP

$$u''(x) = f(x), \qquad \text{for } 0 \le x \le 1$$

with boundary conditions

$$\gamma_0 u(0) + \gamma_1 u'(0) = \sigma, \qquad u(1) = \beta.$$

We want to set up a tridiagonal linear system $Au = f$ with $u = [u_0, u_1, \dots, u_m]$.

We will use the centered approximation for the derivative and the standard second order centered approximation for the second derivative. We approximate the solutions $u(x_j)$ by u_j. At x_0, we get:

$$u_0' \approx \frac{1}{2h}(u_1 - u_{-1})$$

Therefore, the boundary conditon can be written as:

$$\gamma_0 u_0 + \frac{\gamma_1}{2h} u_0' = \sigma \implies \gamma_0 u_0 + \frac{\gamma_1}{2h}(u_1 - u_{-1}) = \sigma \tag{1}$$

We also have,

$$u''(0) = \frac{1}{h^2}(u_{-1} - 2u_0 + u_1) = f(x_0) \implies u_{-1} - 2u_0 + u_1 = f(x_0)h^2 \tag{2}$$

So, we have two equations involving u_0 and we can use them to remove the ghost point.

First, we multiply (1) by $\frac{2h}{\gamma_1}$:

$$\frac{\gamma_0 2h}{\gamma_1}u_0 + u_1 - u_{-1} = \frac{2h}{\gamma_1}\sigma \tag{3}$$

Adding (3) + (2):

$$\begin{array}{c}
\dfrac{\gamma_0 2h}{\gamma_1}u_0 + u_1 - u_{-1} = \dfrac{2h}{\gamma_1}\sigma \\[2ex]
u_{-1} - 2u_0 + u_1 = f(x_0)h^2 \\[2ex]
\hline \\
(\dfrac{\gamma_0 2h}{\gamma_1} - 2)u_0 + 2u_1 = f(x_0)h^2 + \dfrac{2h}{\gamma_1}\sigma
\end{array}$$

For $j = 1, 2, \dots, m-1$:

$$\frac{1}{h^2}(u_{j-1} - 2u_j + u_{j+1}) = f(x_j) \tag{4}$$

$$\implies u_{j-1} - 2u_j + u_{j+1} = h^2 f(x_j)$$

For $j = m$, we can use the fact that we know the boundary condition $u_{m+1} = \beta$.

$$\frac{1}{h^2}(u_{m-1} - 2u_m + u_{m+1}) = f(x_m) \implies u_{m-1} - 2u_m = f(x_m)h^2 - u_{m+1} = f(x_m)h^2 - \beta$$

All of these equations give us the matrix A,

$$A = \begin{bmatrix}
(\frac{\gamma_0 2h}{\gamma_1} - 2) & 2 & 0 & \dots & 0 & \\
1 & -2 & 1 & & & \\
 & 1 & -2 & -1 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & 1 & -2 & 1 \\
 & & & & 1 & -2
\end{bmatrix}$$

and the right hand side,

$$f = \begin{bmatrix} f(x_0)h^2 + \frac{2h}{\gamma_1}\sigma \\ \\ f(x_1)h^2 \\ \vdots \\ f(x_{m-1})h^2 \\ \\ f(x_m)h^2 - \beta \end{bmatrix} = h^2 \begin{bmatrix} f(x_0) + \frac{2\sigma}{h\gamma_1} \\ \\ f(x_1) \\ \vdots \\ f(x_{m-1}) \\ \\ f(x_m) - \frac{\beta}{h^2} \end{bmatrix}$$

Instead of moving h^2 to A we keep the system in this way.

b)

We will determine the local truncation error of the method. For the interior points the local truncation error (LTE) is defined by replacing u_j in Eq (4) with the true solution $u(x_j)$.

Then,

$$\tau_j = \frac{1}{h^2}(u(x_{j-1} - 2u(x_j) + u(x_{j+1})) - f(x_j)$$

for $j = 1, 2, \dots, m$. Since, we assume $u(x)$ is sufficiently smooth, then by Taylor expansion we can get

$$\tau_j = \frac{1}{h^2}(u(x_j) - hu'(x_j) + \frac{h^2}{2}u''(x_j) - \frac{h^3}{6}u'''(x_j) + \frac{h^4}{24}u''''(x_j) - \frac{h^5}{120}u^5(x_j) + \dots - 2u(x_j) + u(x_j) + hu'(x_j) +$$

$$\frac{h^2}{2}u''(x_j) + \frac{h^3}{6}u'''(x_j) + \frac{h^4}{24}u''''(x_j) + \frac{h^5}{120}u^5(x_j) + \dots) - f(x_j)$$

.

So,

$$\tau_j = \frac{1}{h^2}(h^2 u''(x_j) + \frac{h^4}{12}u^4(x_j) + \frac{h^6}{360}u^6(x_j) + \dots) - f(x_j) = u''(x_j) + \frac{h^2}{12}u^4(x_j) + \frac{h^4}{360}u^6(x_j) + \mathcal{O}(h^6)$$

Using our original BVP, we know that $u''(x_j) = f(x_j)$.

Then, $\tau_j = \frac{h^2}{12}u^4(x_j) + \frac{h^4}{360}u^6(x_j) + \mathcal{O}(h^6)$. We can rewrite this as $\tau_j = \frac{h^2}{12}u^4(x_j) + o(h^2)$ since $m(h) = \frac{h^4}{360}u^6(x_j) + \mathcal{O}(h^6) = h^2(\frac{h^2}{360}u^6(x_j) + \mathcal{O}(h^4))$. Then, $\frac{m(h)}{h^2} = c_1h^2 + c_2h^4 + \dots$ for some constants c_1 and c_2. So as $h \to 0$, $\frac{m(h)}{h^2} \to 0$ as h will get smaller and smaller and $m(h)$ is going to be ultimately negligible compared to h^2 as $h \to 0$ since for $h < 1$, $h^2 > h^4$.

Therefore, $\tau_j = \frac{h^2}{12}u^4(x_j) + o(h^2)$

Recall: At the left boundary we used two the second order centered approximation for the second derivative and the derivative. Now, we want to evaluate the truncation error.

The centered second derivative gives a truncation error of: $\frac{h^2}{12}u^4(x_0) + o(h^2)$ as we have seen in the analysis above. What about contribution to the error from the first derivative approximation. We use the same method as above, substituting the actual solution to Eq(1) in this case.

So, truncation error from first derivative approximation:

$$\gamma_0 u(x_0) + \frac{\gamma_1}{2h}(u(x_1) - u(x_{-1})) - \sigma$$

Taylor expanding:

$$\gamma_0 u(x_0) + \frac{\gamma_1}{2h}(2hu'(x_0) + \frac{h^3}{3}u'''(x_0) + \frac{h^5}{60}u^5(x_0) + \mathcal{O}(h^7)) - \sigma$$

$$= \gamma_0 u(x_0) + \gamma_1 u'(x_0) + \frac{\gamma_1 h^2}{6}u'''(x_0) + \mathcal{O}(h^4) - \sigma$$

Since, $u(x_0)$ solves the robin boundary condition we have that $\gamma_0 u(x_0) + \gamma_1 u'(x_0) = \sigma$. Therefore, the truncation error for the boundary condition is

$$\frac{\gamma_1 h^2}{6}u'''(x) + \mathcal{O}(h^4) = \frac{\gamma_1 h^2}{6}u'''(x) + o(h^2)$$

using the same analysis from above about the little o notation.

Therefore,

$$\tau_0 = \frac{h^2}{12}u^4(x_0) + o(h^2) + \frac{\gamma_1 h^2}{6}u'''(x) + o(h^2) = \frac{h^2}{12}u^4(x_0) + \frac{\gamma_1 h^2}{6}u'''(x) + o(h^2)$$

$$= h^2(\frac{u^4(x_0)}{12} + \frac{\gamma_1 h^2 u'''(x)}{6}) + o(h^2)$$

So,

$$\tau_j = C_j h^2 + o(h^2)$$

for

$$C_j = \frac{u^4(x_j)}{12}$$

for $j = 1, 2, \ldots, m$ and

$$C_0 = \frac{u^4(x_0)}{12} + \frac{\gamma_1 h^2 u'''(x)}{6}$$

.

Problem 6. (With corrected boundary conditions)

(a) Implement the method you derived in the previous problem (in Python or Matlab). It is ok to base this on code you have previously written for homework problems, and/or the class Jupyter notebooks.

(b) Test it on the problem

$$\begin{aligned} &u''(x) = 4, \qquad 0 \le x \le 1, \\ &2u(0) + 3u'(0) = 1, \qquad u(1) = 2, \end{aligned}$$

which has exact solution $u(x) = 2x^2 + x - 1$.

Explain why you expect the exact solution to your linear system to agree with exact solution of the ODE when evaluated at the grid points, and confirm that this is true for your implementation.

(c) Also test it on the problem

$$\begin{aligned} &u''(x) = -18x + 4, \qquad 0 \le x \le 1, \\ &2u(0) + 3u'(0) = 1, \qquad u(1) = -1, \end{aligned}$$

which has exact solution $u(x) = -3x^3 + 2x^2 + x - 1$. In this case check that your method is second order accurate by producing a log-log plot of the errors (similar to what was done in the notebook BVP1), using $m + 1 = 50,\ 100,\ 200,\ 400,\ 800$.

Solution:

a)

We expect the solution to the system to agree to the the exact solution of the ODE when evaluated at the grid points because the lowest derivative of the solution $u(x)$ evaluated at the grid points involved in the truncation error is the third derivative. This is true even at the boundary. Since, the exact solution is a second degree polynomial we know that $u^j(x) = 0$ for $j \ge 3$. Therefore, the local truncation error at the grid points should equal zero and therefore the solution to the system must match the exact solution of the ODE.

This is exactly what we see in our implementation. Figure (1) contains the implementation using $m = 4$ and Figure(2) contains the implementation using $m = 99$. The right panel on both figures contains the error at each grid point. We can see that the error is identically zero for both cases which means that solution of the ODE evaluated at the grid points is the exact solution to the system.

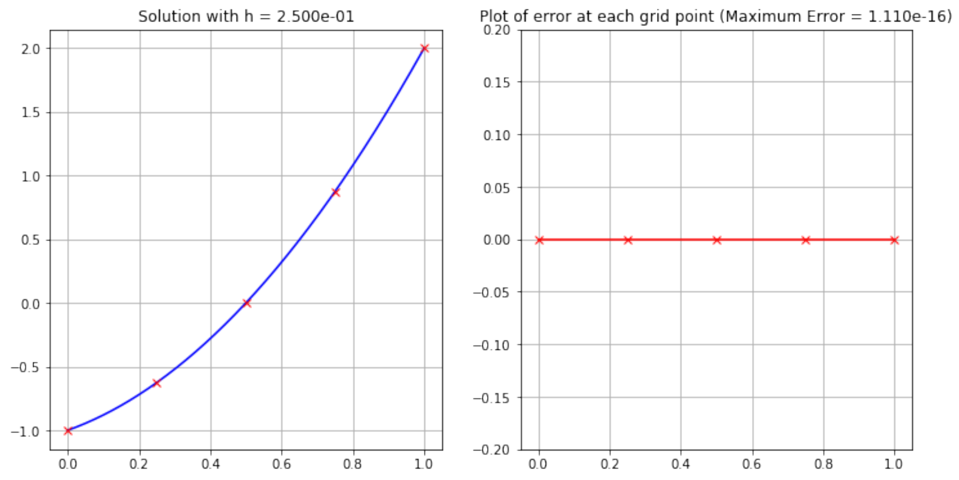

Figure 1: Solution and plot of errors using $m = 4$

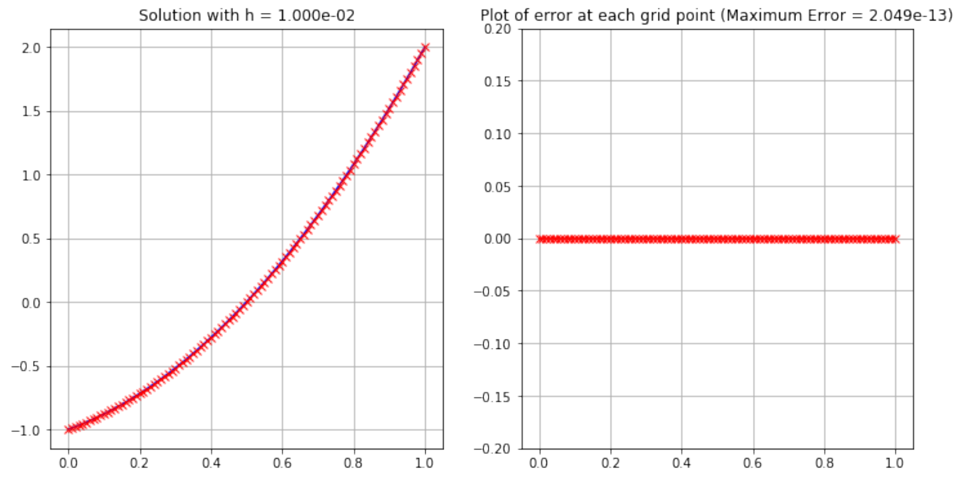

Figure 2: Solution and plot of errors using $m = 99$

b) We do not expect the exact solution to the system to agree with the exact solution of the ODE evaluated at the grid points because have a degree 3 polynomial and the third derivative will not be zero.

Figure (3) contains the solution obtained for a particular $m = 25$. We also plot the errors at each grid point on the right panel. Since, the truncation error at the left boundary involved the third the error is the highest at this point. The truncation error moves towards zero as we swipe left to right across the grid points. The issue is at the left boundary and this point is made even more clear for $m = 39$. We see that most grid points agree well except near the left boundary in Figure (4).

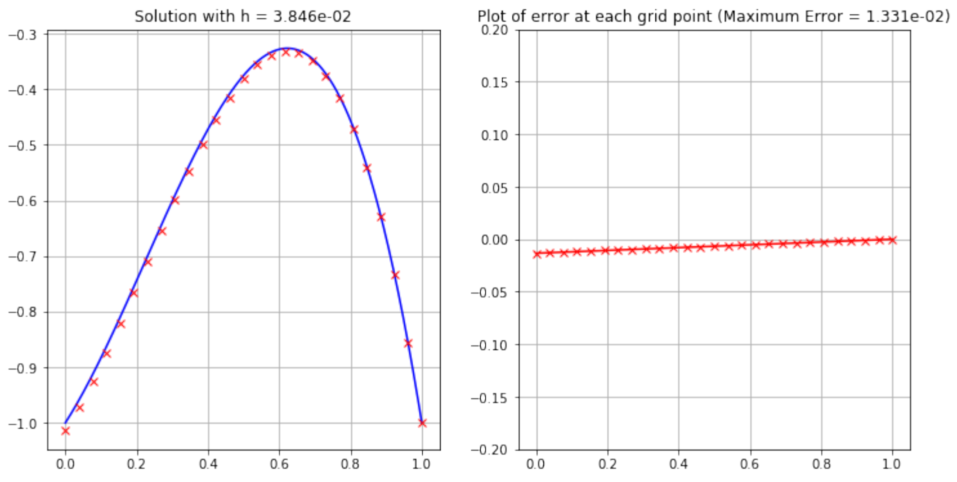

Figure 3: Solution and plot of errors using $m = 25$

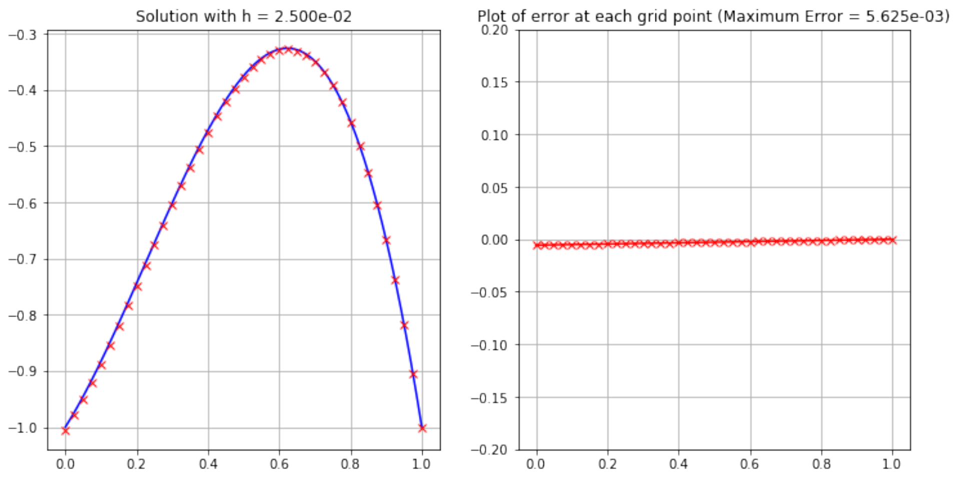

Figure 4: Solution and plot of errors using $m = 39$

Instead, we plot the log log of the error plot versus m to check if we achieve second order accuracy. We use values of $m + 1 = 50, 100, 200, 400, 800$. Figure (5) contains the log log plot of the errors. We include a "reference line" of slope 2. We expect the errors to have this slope for a second-order accurate method. We can see that the observed errors have the same slope as the reference line and so we do in fact have a second order accurate method.

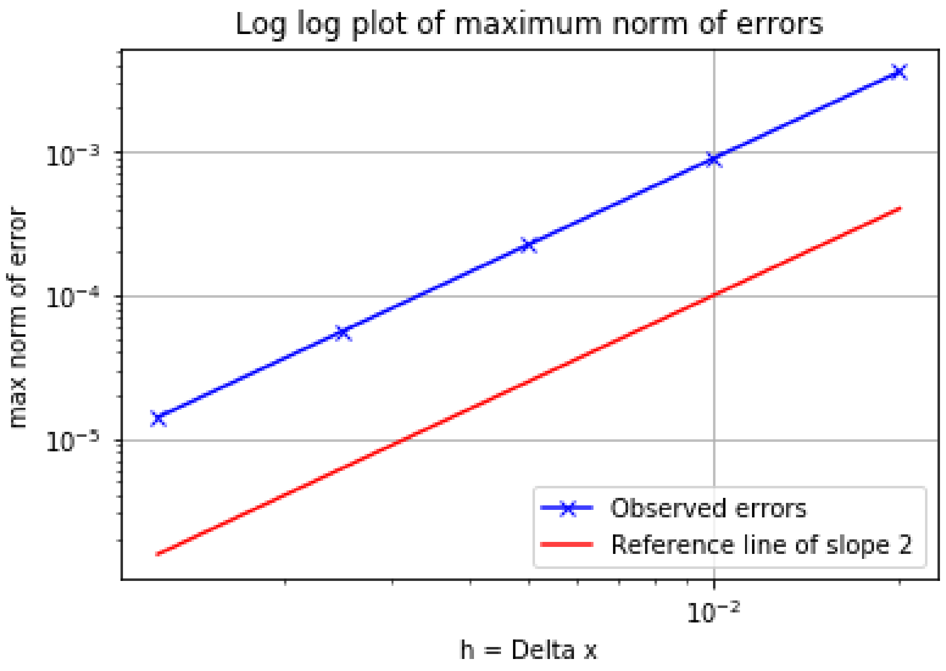

Figure 5: Log log plot of the errors