Numerical Analysis of Boundary Value Problems

Karthik Iyer

This document is a collection of exercises and their solutions for a graduate level course that I took on 'Numerical Analysis of Boundary Value Problems'. The course covered the first four chapters of *Finite Difference Methods for Ordinary and Partial Differential Equations"* by R. J Leveque (RJL). The syllabus included numerical interpolation and differentiation, two-point boundary value problems for ODE's, multidimensional boundary value problems for PDE's and iterative solution of large linear systems. The programming exercises were all coded up in MATLAB. Any errors that remain are entirely mine. Please let me know if you spot any.

AMath 585, Winter '16
Homework 1

Problem #1.

Consider the BVP

$$y^{''} - y = sech(x), \quad -L/2 < x < L/2, \quad y(-L/2) = y(L/2) = 0, \quad L/2 = 20 \qquad (1)$$

This is a truncation of the same BVP on $\mathbb{R}$ which has the exact solution
$y_E(x) = -\frac{e^x}{2}\log(1+e^{-2x}) - \frac{e^{-x}}{2}\log(1+e^{2x})$.
This domain is large enough that the exact solution is only -4×10^{-8} at the boundaries, so the exact solution to (1) should be well approximated by the exact solution to the infinite domain problem. Make a finite difference discretization to this BVP using a uniform grid spacing h and a second order finite difference approximation for d^2/dx^2 as in class. Index the grid points $x_j = -L/2 + jh$, $j = 1, 2, ..., m$.

(a) Write down the finite difference approximation to (1) at each grid point x_j for a general h, distinguishing the cases $j = 1, m$ from other $j's$. Set the problem up as tridiagonal matrix problem $\mathbf{Ax} = \mathbf{f}$

Solution: The standard second order accurate finite difference scheme for approximating second order derivative gives us $D^2U_j = \frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1})$ for $1 \le j \le m$. In our case $U_0 = U_{m+1} = 0$. We thus get the following tri-diagonal matrix

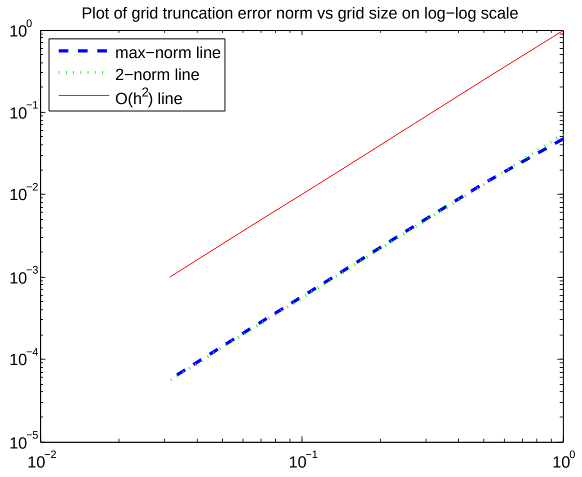
$$A = \begin{bmatrix} -\frac{2}{h^2} - 1 & 1/h^2 & & 0 \\ 1/h^2 & -\frac{2}{h^2} - 1 & 1/h^2 & \ddots \\ & \ddots & \ddots & 1/h^2 \\ 0 & & 1/h^2 & -\frac{2}{h^2} - 1 \end{bmatrix}$$

$Ax = f$ then takes the form where A is as above, x is the vector of unknowns, more specifically the value of $y(x)$ at the grid points $x_j = -L/2 + jh$, $j = 1, 2, ..., m$ and

$$f = \begin{bmatrix} sech(x_1) \\ sech(x_2) \\ \vdots \\ sech(x_m) \end{bmatrix}$$

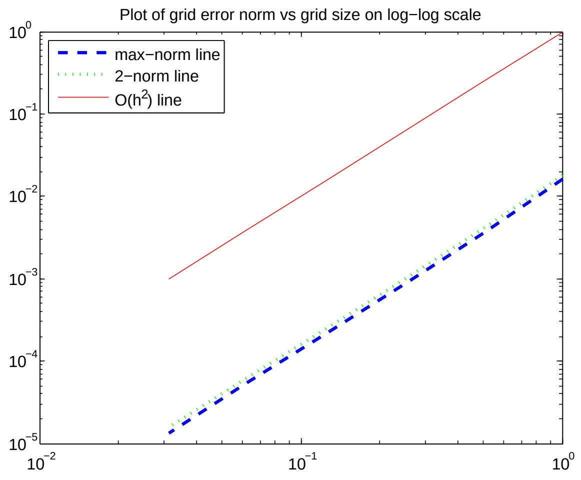
(b) For $h = 2^{-p}$, $p = 0, 1, 2, ..., 5$, calculate the grid 2 norm and the max norm of the truncation error vector computed at the grid points. Plot the error norms on a log-log plot vs h and use an $O(h^2)$ comparison line to show the error norms go to $O(h^2)$ as expected. In computing the truncation error, approximate the exact solution by $y_E(x)$.

Solution: For each value of h, we compute the vector $\tau = A\hat{U} - f$ where A is as in part(a), U is the vector of exact solution at the grid points and f is also as in part(a). We have attached a plot of grid norm of τ vs grid size on a log-log scale. We observe that for both the max and 2 norm, the plotted lines are parallel to h^2 line. This confirms the theory.



(c) For the same set of h calculate the solution in MATLAB. Find the 2-norm and the max norm of the error of the solution, again approximating the exact solution by $y_E(x)$. Plot these 2 error norms on a log-log plot vs h. Do they also go as $O(h^2)$?

Solution: For each value of h, we solve the tridiagonal system $Ax = f$ and get the vector of computed solution. We then calculate the grid max norm and grid 2-norm for error = computed solution - actual solution and plot it against the grid size on a log-log scale. Even in this case both the grid error norms behave like $O(h^2)$.



AMath 585, Winter '16
Homework 2

Consider the BVP and its finite difference approximation from Homework 1

Problem #1. Calculate the Green's function $G(x;\bar{x})$ for the infinite domain problem, which obeys $\frac{\partial^2 G}{\partial x^2} - G = \delta(x-\bar{x})$, $-\infty < x < \infty$. for an arbitrary value of the parameter $\bar{x}$.

Solution: Green's function solves the BVP $\frac{\partial^2 G}{\partial x^2} - G = \delta(x-\bar{x})$, $G(-\infty) = G(\infty) = 0$. The solution to the homogeneous problem with the same boundary conditions gives us two independent solutions $u_1(x) = c_1 e^x$ and $u_2(x) = c_2 e^{-x}$. Thus

$$G(x;\bar{x}) = \begin{cases} c_1 e^x & : x < \bar{x} \\ c_2 e^{-x} & : x \geq \bar{x} \end{cases}$$

Using continuity of $G(x;\bar{x})$ at $x = \bar{x}$ and that $\frac{\partial G}{\partial x}$ has a finite jump discontinuity at $x = \bar{x}$, we get

$$G(x;\bar{x}) = \begin{cases} \frac{-1}{2}\, e^{(x-\bar{x})} & : x < \bar{x} \\ \frac{-1}{2}\, e^{-(x-\bar{x})} & : x \geq \bar{x} \end{cases}$$

Problem #2. For the case $h = 0.5$, calculate the inverse of the $m \times m$ tridiagonal matrix which arises in the finite difference solution of the BVP, now over the large finite domain $-20 < x < 20$, so $m = 79$. In class we argued that $(A^{-1})_{ij}$ is an approximation to $hG(x_i;x_j)$. For the case $x_j = 5$ check this assertion by plotting both of these quantities at the grid points x_i, $i = 1, 2, ..., m = 79$.

Solution: For $h = 0.5$ we compute $B = A^{-1}$. We then find the index corresponding to $xj = 5$ in the vector of grid points. That index turns out to be 50. We compute $G(x;5)$ and plot $B(:,50)$ vs $h * G(x;5)$. In the attached plot in Figure [**1**] we can see that $B(:,50)$ and $hG(x_i,5)$ are almost the same. This verifies the assertion that $(A^{-1})_{ij}$ is an approximation to $hG(x_i;x_j)$.

Problem #3. Based on your answer to Problem 1, argue that in the limit $h \to 0$, the matrix infinity norm of A^{-1} should be 1 (which implies the FDA is stable in the infinity norm). If you use MATLAB to compute the matrix infinity norm of A^{-1} for the case in Problem 2, is it close to this value?

Solution: Let us fix the number of grid points m and the grid size h. Let $B = A^{-1}$. (B depends on h. We suppress the dependence of B on h now on) Since the jth column

$B(:,j)$ is a discrete approximation to $G(x;x_j)$, $B_{ij} = h * G(x_i;x_j)$

$$B_{ij} = \begin{cases} \frac{-h}{2}\, e^{(x_i - x_j)} & : i = 1,2,...,j \\ \frac{-h}{2}\, e^{-(x_i - x_j)} & : i = j, j+1, ..., m \end{cases}$$

Thus $\sum_{i=1}^{m} |B_{ij}| = \sum_{i=1}^{m} \frac{h}{2} e^{|x_i - x_j|}$.

As $h \to 0$ $m \to \infty$ and the above Riemann sum converges to $\int\limits_{\mathbb{R}} \frac{1}{2} e^{-|x-x_j|}\, dx$. (Note that we are dealing with an infinite domain problem. So when $h \to 0$ we also tacitly let $L \to \infty$) To get $||B||_\infty$, we now maximize over j, the choices for which keep on increasing as $h \to 0$. For any choice of x_j it is clear that $\int\limits_{\mathbb{R}} \frac{1}{2} e^{-|x-x_j|}\, dx = 1$. This implies that $||B||_\infty \to 1$ as $h \to 0$. Computing $||B|_\infty$ for the case $h = 0.5$ using MATLAB gives us exactly 1.

Problem #4. Following a similar approach to lecture, the eigenvalues of the finite difference second derivative operator D^2 on a domain of length not equal to 1 are: $\lambda_p = \frac{2}{h^2}\left[\cos(\frac{p\pi}{m+1}) - 1\right]$, $p = 1, 2, ..., m$. Based on this, what is the spectral radius (=matrix 2-norm) of A^{-1} constructed in Problem 2? For the same domain size, what value would the matrix 2-norm approach as $h \to 0$? What does this tell us about the stability of the FDA in the 2 norm?

Solution: For our problem, the finite difference operator is $A = D^2 - I$. Thus the m eigenvalues of A will be $\lambda_p - 1$, $p = 1, 2, ..., m$. Since A is symmetric, $A_2^{-1} = (\min_{1 \le p \le m} |\lambda_p - 1|)^{-1}$. As in the text we can use Taylor expansion and argue that the smallest eigenvalue of A in magnitude is $\lambda_1 - 1 = -\frac{\pi^2}{L^2} - 1 + O(h^2)$. This is clearly bounded away from 0 as $h \to 0$. This tells that the FDA method is stable in the 2-norm. Moreover the above estimate tells us $||A^{-1}||_2 \to \frac{L^2}{\pi^2 + L^2}$ as $h \to 0$. The spectral radius of A^{-1} constructed in problem 2 is evaluated to be 0.993870097387878 which is close to $\frac{L^2}{\pi^2 + L^2}$.

Problem #5. Return to the max-norm and the grid 2 norm of the solution and truncation errors found in HW1 for $h = 0.5$. For each of these 2 norms, calculate the ratio of the norm of the solution error to the norm of truncation error and check that is less than the corresponding matrix norm for A^{-1} (this bound was an important step in our theoretical argument about the convergence of the finite difference approximation)

Solution: Based on the codes submitted for HW1, for $h = 0.5$ we have $||\frac{\text{solution error}}{\text{truncation error}}||_2 = 0.300734087120016 < ||A^{-1}||_2 = 0.993870097387878$ and
$||\frac{\text{solution error}}{\text{truncation error}}||_\infty = 0.258771356111123 < ||A^{-1}||_\infty = 1$.

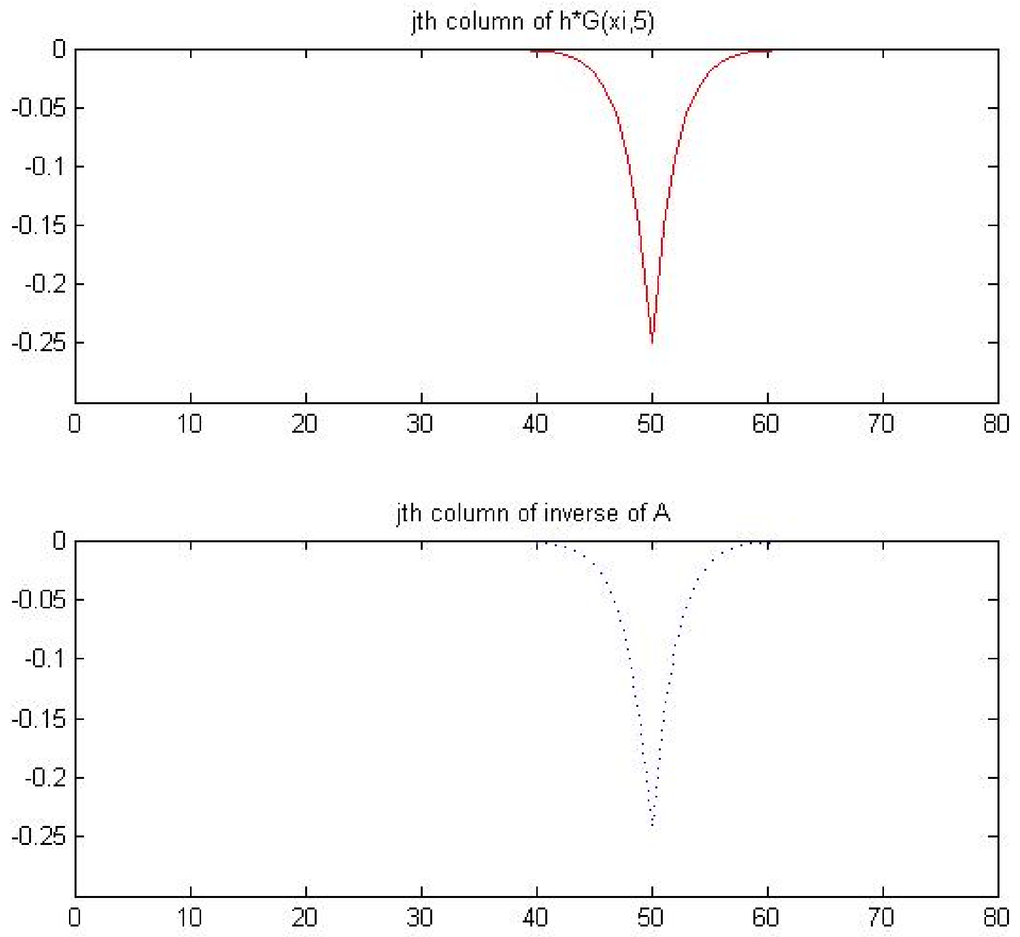


Figure 1: **Experimental verification of** $A_{ij}^{-1} = h * G(x_i; x_j)$ **for** $h = 0.5$ **and** $x_j = 5$**.**

AMath 585, Winter '16
Homework 3

Problem #1. You can solve the BVP of Homework 1 more efficiently as follows. Since the solution is a an even function of x, it must be identical to the solution of

$$\frac{d^2y}{dx^2} - y = sech(x), \quad y'(0) = 0\,; \;\; y(20) = 0 \tag{1}$$

For a given grid spacing h this requires only half as many as grid points as HW1. Discretize the above BVP using finite differences to second order accuracy. Implement this approach in MATLAB, and show that for $h = 1$ it gives identical solution (to roundoff error) as the finite difference approximation in HW1.

Solution: For the BVP (1) using finite difference approximation with $h = 1$ gives us $m = 19$ interior grid points. To obtain a second order approximation to $y'(0) = 0$ we use centered difference. This results in the following $(m+2) \times (m+2)$ finite difference matrix for the BVP (1).

$$A = \begin{bmatrix} -\frac{1}{h} - \frac{h}{2} & 1/h & & 0 & \\ 1/h^2 & -\frac{2}{h^2} - 1 & 1/h^2 & \ddots & \\ & \ddots & \ddots & 1/h^2 & \\ 0 & & 1/h^2 & -\frac{2}{h^2} - 1 & 1/h^2 \\ 0 & & 0 & 0 & 1 \end{bmatrix}$$

The given boundary conditions give us the following vector

$$f = \begin{bmatrix} 0 \\ sech(x_1) \\ sech(x_2) \\ \vdots \\ sech(x_m) \\ 0 \end{bmatrix}$$

where $x_1, x_2, ..., x_m$ are the interior grid points. We solve for U in $AU = f$ to get the solution at the interior grid points. We compare this solution with the solution to the Dirichlet problem computed in HW1 for grid size $h = 1$. Since the solution to the Dirichlet problem is defined even for negative values of x, we only consider the solution to the Dirichlet problem for positive x and compare that to the above computed solution to the mixed boundary value problem. As seen in Figure (1), the solutions are nearly identical.

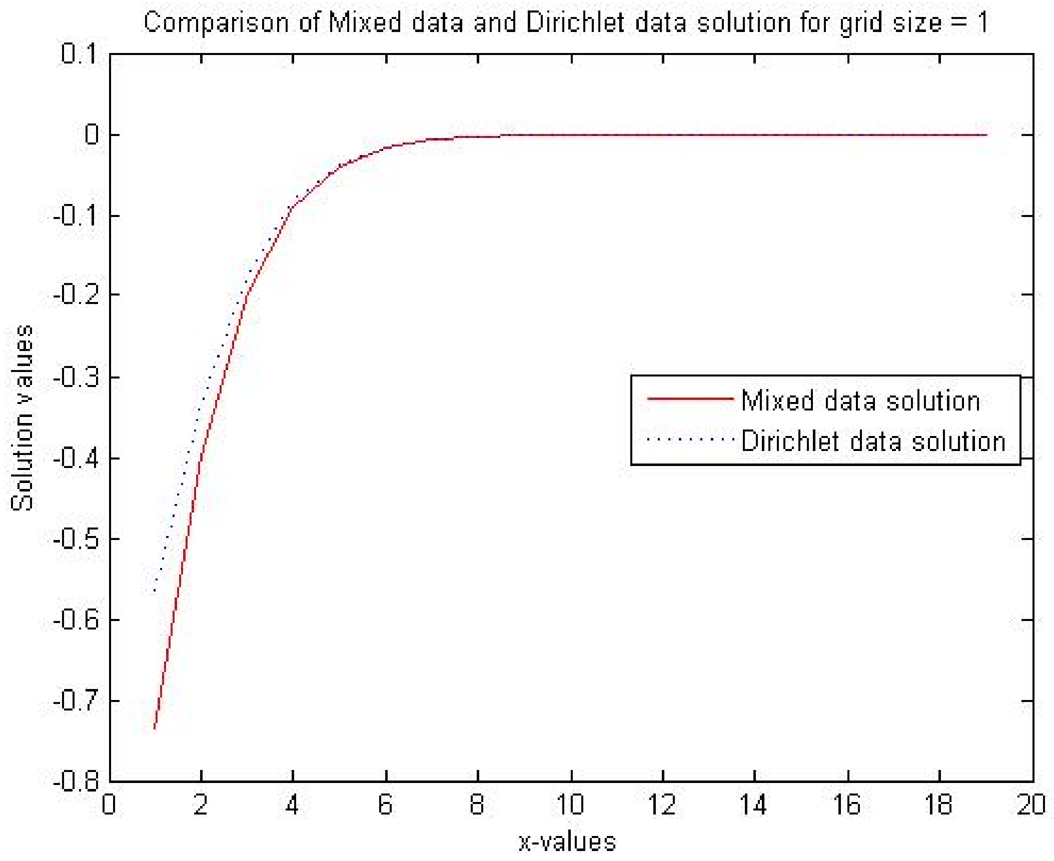


Figure 1: **Plots of Dirichlet data solution and mixed data solution on (0,20) for grid size h=1**

Problem #2. Consider the following non-linear BVP involving the steady state Burger's equation, a non-linear advection-diffusion equation often used to describe the structure of shock waves:

$$uu' - \epsilon u'' = 0 \quad u(0) = 1 \;\; u(1) = -1 \;\; \epsilon = 1$$

(a) Write down a 2nd order accurate finite difference discretization of this equation on a grid of spacing h with m interior grid points.

Solution: Following our approach for linear problems, we can use the usual second order accurate scheme for second order derivatives and the second order accurate centered difference for first order derivatives to get the following system of equations

$$\frac{1}{h^2}(U_{i-1} - 2U_i + U_{i+1}) - \frac{U_i}{2h}(U_{i+1} - U_{i-1}) = 0 \tag{2}$$

for $i = 1, 2, ..., m$ where $h = 1/(m+1)$ with $U_0 = 1$ and $U_{m+1} = -1$. This is a non linear system of equations of the form $G(U) = 0$ where $G : \mathbb{R}^m \to \mathbb{R}^m$.

(b) Use matrix Newton iteration to find and plot the approximate solution $\mathbf{U}^{(p)}$ for $h^{(p)} = 2^{-p}$ for $p = 2, 3, ..., 6$ as a column vector of length $m^{(p)} = 2^p - 1$.

Solution: Our objective function for running Newton's iteration has been defined in (2). The Jacobian for the system is

$$J = \begin{bmatrix} \left(-\frac{2}{h^2} - \frac{U_2 - U_0}{2h}\right) & \frac{1}{h^2} - \frac{U_1}{2h} & & 0 \\ \frac{1}{h^2} + \frac{U_2}{2h} & \left(-\frac{2}{h^2} - \frac{U_3 - U_1}{2h}\right) & \frac{1}{h^2} - \frac{U_2}{2h} & \ddots \\ & \ddots & \ddots & \frac{1}{h^2} - \frac{U_{m-1}}{2h} \\ 0 & & \frac{1}{h^2} + \frac{U_m}{2h} & \left(-\frac{2}{h^2} - \frac{U_{m+1} - U_{m-1}}{2h}\right) \end{bmatrix}$$

with $U_0 = 1$ and $U_{m+1} = -1$. We use matrix Newton iteration with initial guess $u_0(x) = x^2$ and find the approximate solution for each of the grid spacings $h^{(p)} = 2^{-p}$ for $p = 2, 3, ..., 6$. We have attached the plot in Figure (2).

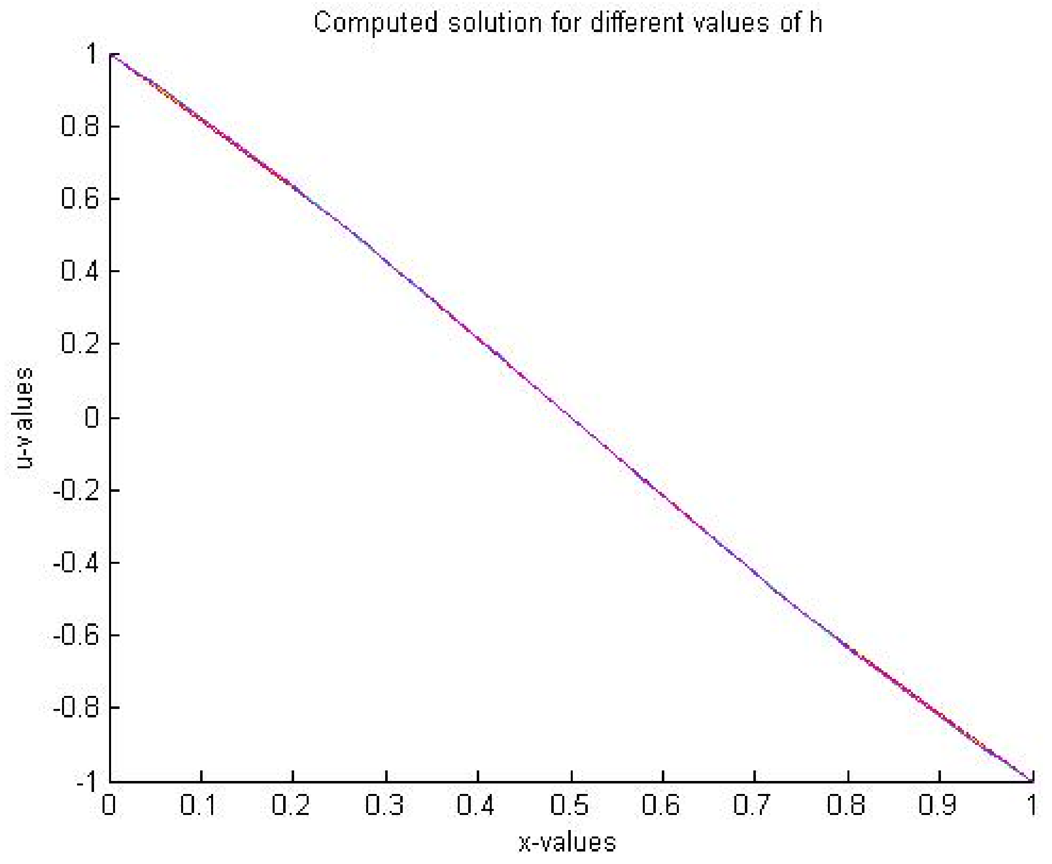


Figure 2: **Plots of solutions to $u'' - uu' = 0$ with $u(0) = 1$ and $u(1) = -1$ on [0,1] for different grid sizes.**

(c) As a measure of solution error, calculate the max norm $\delta^{(p)}$ of the the difference of $\mathbf{U}^{(p)}$ and $\mathbf{U}^{(p-1)}$ on the grid points that these solutions have in common, and use a log-log plot of $\delta^{(p)}$ vs $h^{(p)}$ to argue that your solution is $O(h^2)$ accurate

Solution: We compute the max norm $\delta^{(p)}$ of the the difference of $\mathbf{U}^{(p)}$ and $\mathbf{U}^{(p-1)}$ on the grid points that these solutions have in common for each value of $p = 2, 3, 4, 5, 6$. We have attached the log-log plot of the max norm of the errors vs the vector of the last 4 values of h in Figure (3). (We can take only the last 4 of the 5 values of h as the 'difference' vector solution is a vector of length 4).

As observed in the plot, on the log-log scale the error behaves linearly with slope 2, the order of accuracy. By existence of limits for Cauchy sequences the successive approximations converge to the approximate solution as $h \to 0$.

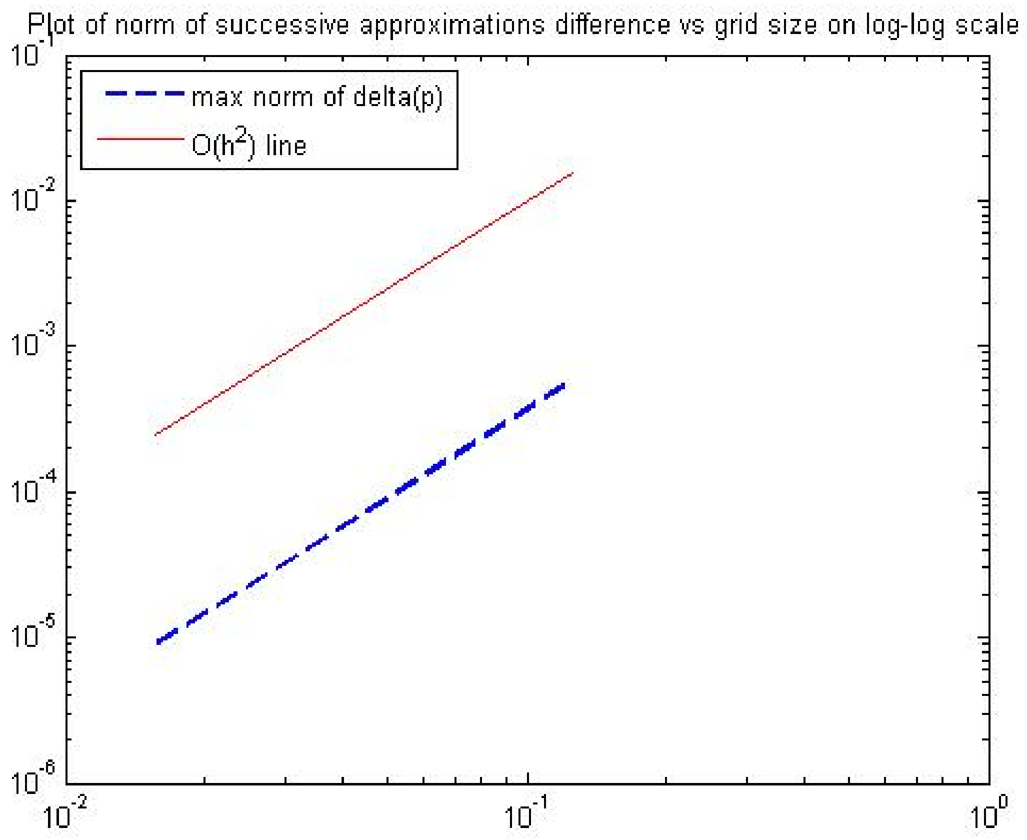


Figure 3: **Plot of norm of difference of successive approximations to the solution to $u'' - uu' = 0$ with $u(0) = 1$ and $u(1) = -1$ with different grid spacings on a log-log scale.**

(d) Repeat parts (b) and (c) for the case $\epsilon = 0.1$. How is the solution different? Are the errors measured by δ^p smaller or larger for a given grid spacing? Can you give a reason why you might have expected your result?

Solution: For $\epsilon = 0.1$ and the given grid spacings h we use Newton's method with initial guess $u_0(x) = x^2$ and plot the solutions to $\epsilon u'' - uu' = 0$ with $u(0) = 1$ and $u(1) = -1$ on $[0, 1]$. We get a plot as in Figure (4). We can see that as h decreases the solution has more curvature with slower change in the interior and sharper change near the end points. This can be seen if Figure (4) in the blue colored curve which corresponds to $h = 2^{-6}$. We expect to see highly discontinuous behavior as $\epsilon \to 0$, since in the limit we are over specifying the problem.

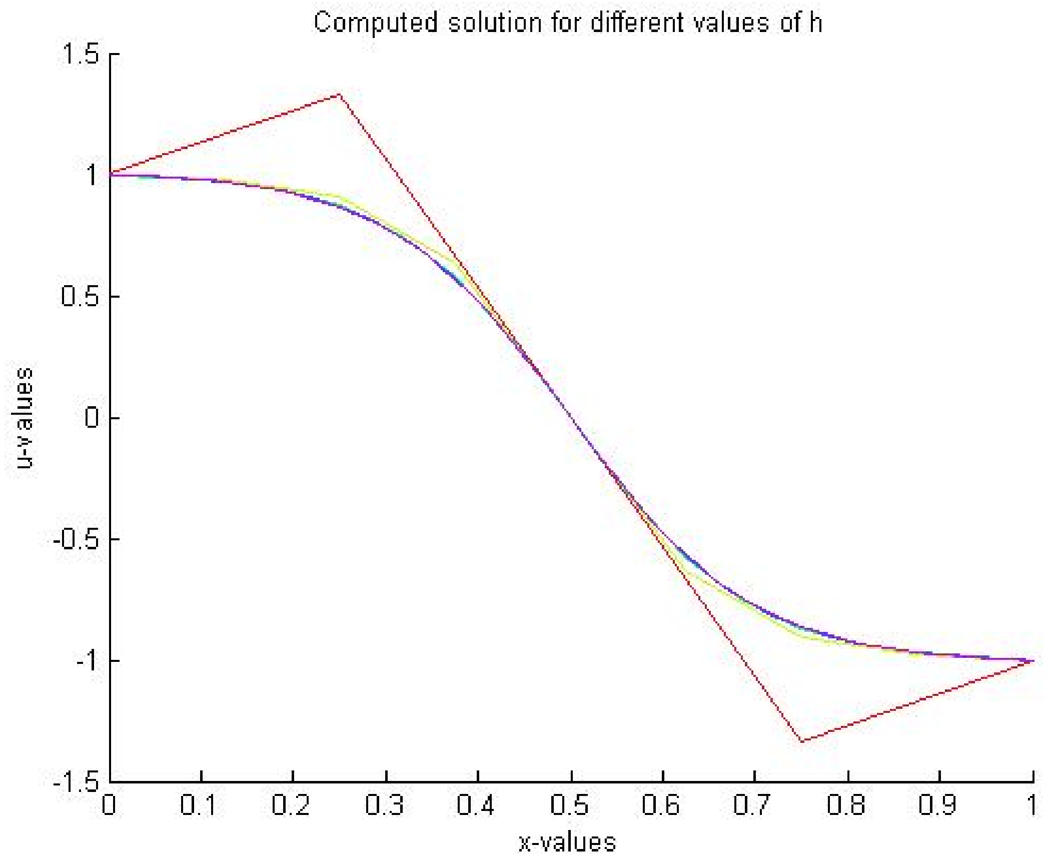


Figure 4: **Plots of solutions to** $0.1u'' - uu' = 0$ **with** $u(0) = 1$ **and** $u(1) = -1$ **on [0,1] for different grid sizes.**

For $\epsilon = 0.1$ the errors measured by $\delta^{(p)}$ turn out to be higher for a given grid spacing as compared to that for $\epsilon = 1$. However it seems that the errors behave better than $O(h^2)$ as seen in Figure (5). (The slope is larger.)

One likely explanation is that for $\epsilon = 0.1$ the initial few grid sizes are comparable to ϵ and as $h \to 0$ the size of ϵ is higher compared to h so that we get the true accuracy for the finite difference approximation. In this case it looks like the accuracy is better than $O(h^2)$.

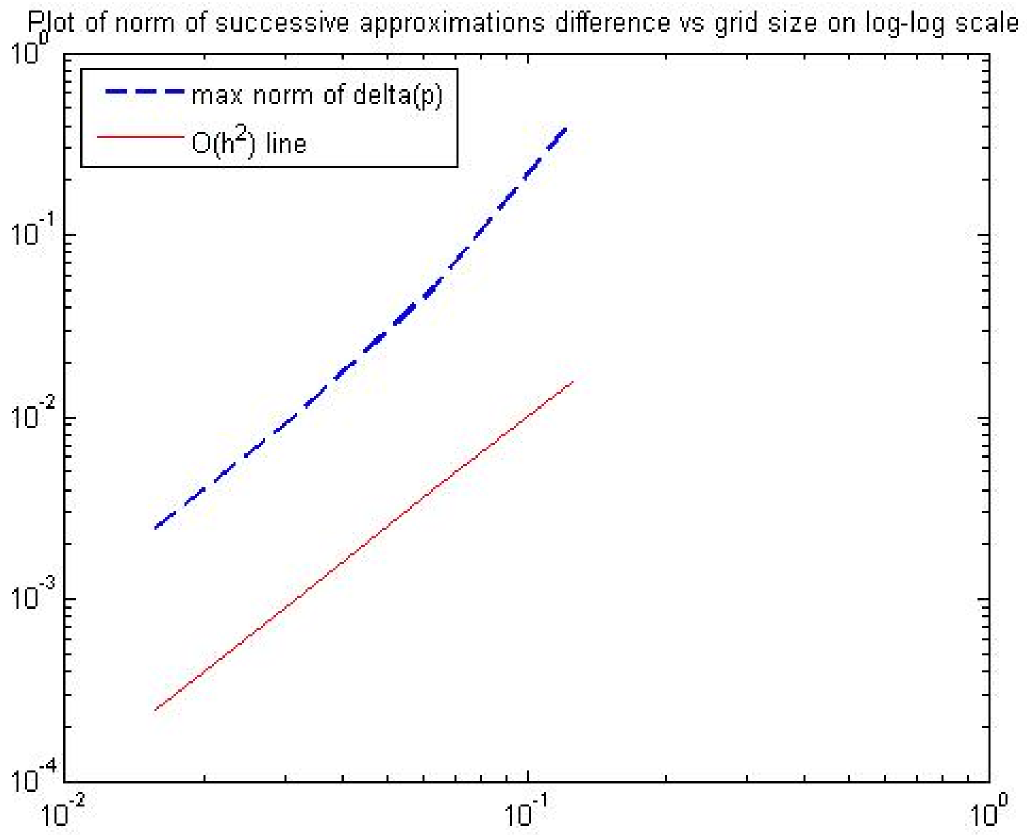


Figure 5: **Plot of norm of difference of successive approximations to the solution to $0.1u'' - uu' = 0$ with $u(0) = 1$ and $u(1) = -1$ with different grid spacings on a log-log scale.**

AMath 585, Winter '16
Homework 4

Consider the BVP considered in class

$$\epsilon u'' - u' = -1 \quad u(0) = 1 \quad u(1) = 3 \quad \epsilon = 0.01 \tag{1}$$

whose exact solution

$$u_\epsilon(x) = 1 + x + \frac{e^{x/\epsilon} - 1}{e^{1/\epsilon} - 1} \tag{2}$$

has a boundary layer near $x = 1$. We use the co-ordinate transformation

$$x = X(z) = \frac{1 - e^{z \log \epsilon}}{1 - \epsilon} \tag{3}$$

to generate a non uniform grid x_j across the domain from uniformly spaced grid points $z_j = jh$, $j = 0, 1, ..., m+1$ with a given grid spacing $h = \frac{1}{m+1}$.

Problem 1a) Assuming $h \ll 1$, use the derivative of (3) to show the spacing $h_{j-\frac{1}{2}} = x_j - x_{j-1}$ of the non uniform grid is only a fraction ϵ as large next to the right boundary as it is next to the left boundary. Explain why this grid configuration is particularly well suited to discretizing a problem whose solution is anticipated to have a right boundary layer with thickness $O(\epsilon)$.

Solution: $\frac{x_{m+1}-x_m}{x_1-x_0} = \frac{\frac{x_{m+1}-x_m}{h}}{\frac{x_1-x_0}{h}} \sim \frac{X'(1)}{X'(0)} = \epsilon$ for small values of h. This tells us that the spacing $h_{j-\frac{1}{2}} = x_j - x_{j-1}$ of the non uniform grid is only a fraction ϵ as large next to the right boundary as it is next to the left boundary. Thus the grid resolution is $O(\epsilon)$ at the right boundary point compared to the left boundary point. The idea is to have a coarse mesh outside the boundary layer and a fine mesh inside the boundary layer so that the number of grid points remains the same (so that the computational cost is still manageable) but some information about the actual solution in the boundary layer is obtained. Since the width of the right boundary layer for equation (1) is $O(\epsilon)$ we concentrate the grid with $O(\frac{1}{\epsilon})$ 'more' points near the right boundary layer so that the properties of the actual solution near the location of the boundary layer i.e the right boundary layer are captured by the numerical method. This is achieved by the mapping $X(z)$ which transforms an uniform grid in to a non-uniform grid with grid points being concentrated most at the place where the solution changes most rapidly i.e at the right boundary point.

Problem 1b) Plot $X(z)$ over the interval $0 < z < 1$. On this plot, also show as $x's$ the grid points x_j that correspond to $z_j = jh$ for $h = 0.1$, so that you can see how they are concentrated near the boundary.

Solution: Figure 1 shows the plot of z vs $X(z)$. We can see that $X(z)$ is concentrated near the boundary. If we drop vertical lines from the points marked on the graph to the horizontal axis ($X(z)$ axis in our case), we can see that the grid spacing on the horizontal axis decreases as we move toward $X(z) = 1$.

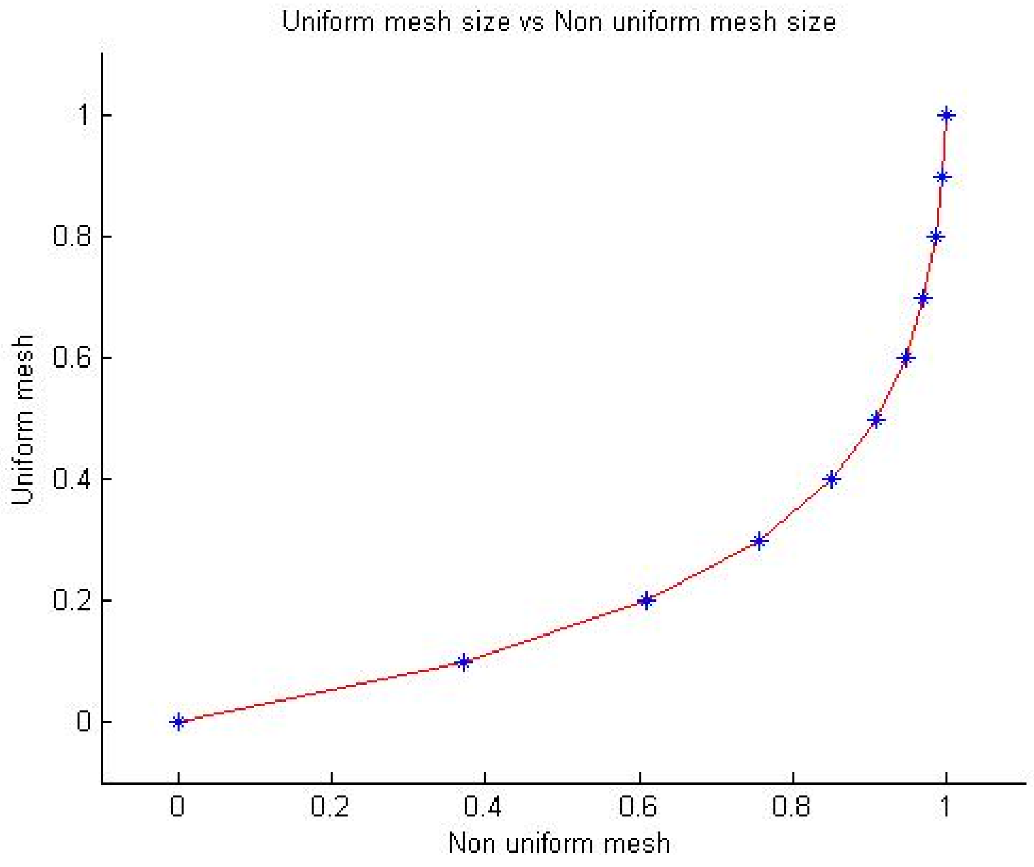


Figure 1: **Plot of uniform mesh vs non uniform mesh. The points of non uniform mesh are not equally spaced and are concentrated near the right boundary.**

Problem 2) Using methodology similar to page 51 of RJL's notes, construct a second order finite difference approximation to (1) on this non uniform grid. Solve the resulting sparse matrix equation and plot its solution at the grid points x_j on top of the exact solution (2). What is the max norm error of your solution? How does it compare to the max norm error of a finite difference approximation on a uniform grid in x with the same number of grid points (which you can estimate by eye from Fig 2.6b)?

Solution: We plot of the solution to (1) using finite difference approximation with non uniform mesh on top of the exact solution. We have used the MATLAB function **fdcoeffV** in RJL's notes to compute the coefficients for the finite difference approximation. The resultant plot is attached in Figure (2). The max norm error of the computed solution turns out to be 0.2160. This error is smaller to the max norm error of the finite difference approximation on a uniform grid in x.

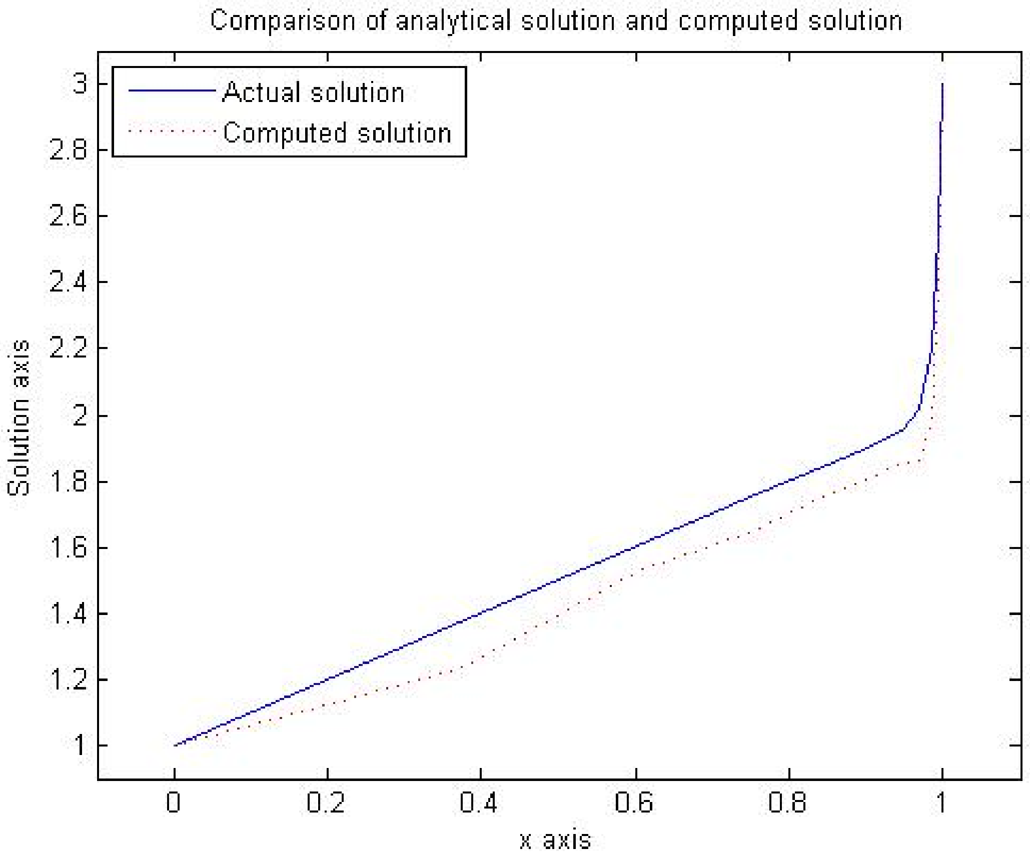


Figure 2: **Plots of analytical and computed solutions to** $\epsilon u'' - u' = -1$ **with** $u(0) = 1$ **and** $u(1) = 3$ **for** $m = 9$ **interior grid points.**

Problem 3) Implement the finite element method discussed in class on (3) using the same stretched grid x_j. Add the finite element solution to the plot of Problem 2. How does its max norm error compare with the FDA?

Solution: We set up the FEM on m interior grid points using the trial function $u_h(x) = c_0\phi_0(x) + \sum\limits_{j=1}^{m} c_j\phi_j(x) + c_{m+1}\phi_{m+1}(x)$ and test functions $\phi_i(x)$ with $1 \le i \le m$. We denote the interior grid points by x_i with $1 \le i \le m$, the left boundary point by $x_0 = 0$ and the right boundary point by $x_{m+1} = 1$. We let $h_i = x_i - x_{i-1}$. Here,

$$\phi_0(x) = \begin{cases} \frac{1}{h_1}(h_1 - x) & : 0 \leq x \leq x_1 \\ 0 & : otherwise \end{cases}$$

$$\phi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h_i} & : x_{i-1} \leq x \leq x_1 \\ \frac{x_{i+1} - x}{h_{i+1}} & : x_i \leq x \leq x_{i+1} \\ 0 & : otherwise \end{cases}$$

for $1 \leq i \leq m$ and

$$\phi_{m+1}(x) = \begin{cases} \frac{1}{h_{m+1}}(x - x_m) & : x_m \leq x \leq x_{m+1} \\ 0 & : otherwise \end{cases}$$

Plugging in the trial function u_h and the test functions ϕ_i in to (1) and integrating by parts gives us the following m equations with m unknowns cj for $1 \leq j \leq m$. Note that $c_0 = u(0) = 1$ and $c_{m+1} = u(1) = 3$.

$$\epsilon \int_0^1 \sum_{j=1}^{m} c_j \phi_j^{'} \phi_i^{'}(x) - \int_0^1 \sum_{j=1}^{m} c_j \phi_j \phi_i^{'} = \int_0^1 \phi_i - \epsilon \int_0^1 c_0 \phi_0^{'} \phi_i^{'} - \epsilon \int_0^1 c_{m+1} \phi_{m+1}^{'} \phi_i^{'}$$

$$+ \int_0^1 c_0 \phi_0 \phi_i^{'} + \int_0^1 c_{m+1} \phi_{m+1} \phi_i^{'}$$

We compute the above expression using the explicit form of ϕ_i for $0 \leq i \leq m+1$ and get $Ac = F$ where

$$A = \begin{bmatrix} \epsilon\left(\frac{1}{h_1} + \frac{1}{h_2}\right) & -\epsilon\frac{1}{h_2} + \frac{1}{2} & & 0 \\ -\epsilon\frac{1}{h_2} - \frac{1}{2} & \epsilon\left(\frac{1}{h_2} + \frac{1}{h_3}\right) & -\epsilon\frac{1}{h_3} + \frac{1}{2} & \ddots \\ & \ddots & \ddots & -\epsilon\frac{1}{h_m} + \frac{1}{2} \\ 0 & & -\epsilon\frac{1}{h_m} - \frac{1}{2} & \epsilon\left(\frac{1}{h_m} + \frac{1}{h_{m+1}}\right) \end{bmatrix}$$

$c = [c_1; c_2; ...c_m]^{'}$ and

$$F = \begin{bmatrix} \frac{1}{h_1} + \frac{1}{h_2} + c_0\left(\frac{1}{2} + \frac{\epsilon}{h_1}\right) \\ \frac{1}{h_2} + \frac{1}{h_3} \\ \vdots \\ \frac{1}{h_m} + \frac{1}{h_{m+1}} + c_{m+1}\left(-\frac{1}{2} + \frac{\epsilon}{h_{m+1}}\right) \end{bmatrix}$$

We solve the above system for $m = 9$ and $\epsilon = 0.1$ and plot the resultant solution alongside the solution to Problem 2. We have attached the plot in Figure 3. The max norm error for the finite element method for $m = 9$ turns out to be 0.02 which is much smaller than the max norm error for the finite difference method approximation computed in Problem 2. For $m = 9$ interior grid points we can see that the finite element method performs better than the finite difference method.

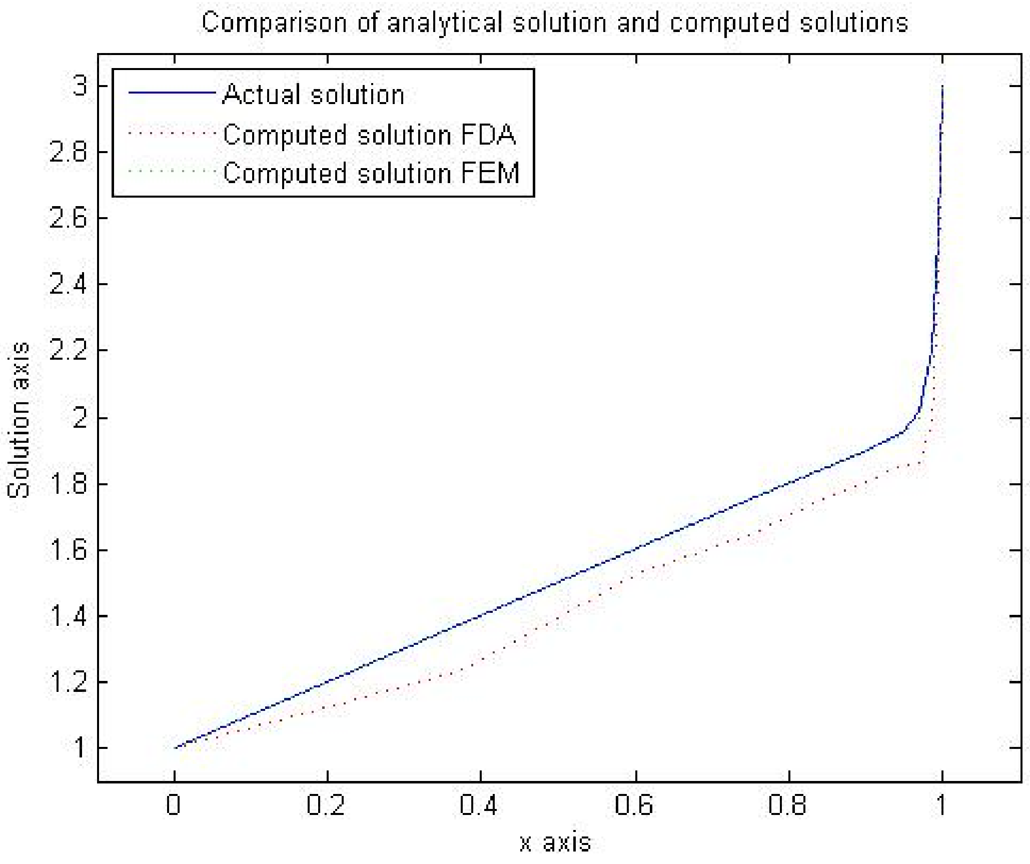


Figure 3: **Plot of analytical solution, approximate solution using FDA and approximate solution using FEM all on a non uniform grid with $m = 9$ interior grid points for $\epsilon u'' - u' = -1$ with $u(0) = 1$ and $u(1) = 3$.**

Problem 4) On a log-log plot, plot the max norm of the solution error for $h = 0.1, 0.04, 0.02, 0.01$ vs h for both the finite difference and finite element methods, as well as a comparison $O(h^2)$ line. If you have got everything right, both the methods should exhibit second order accuracy.

Solution: Figure (4) shows on a log-log plot the max norm of the solution error for $h = 0.1, 0.04, 0.02, 0.01$ vs h i.e for $m = 9, 24, 49, 99$ interior grid points respectively for both the finite difference and finite element methods, as well as a comparison $O(h^2)$ line.

For both methods we get second order accuracy as expected.

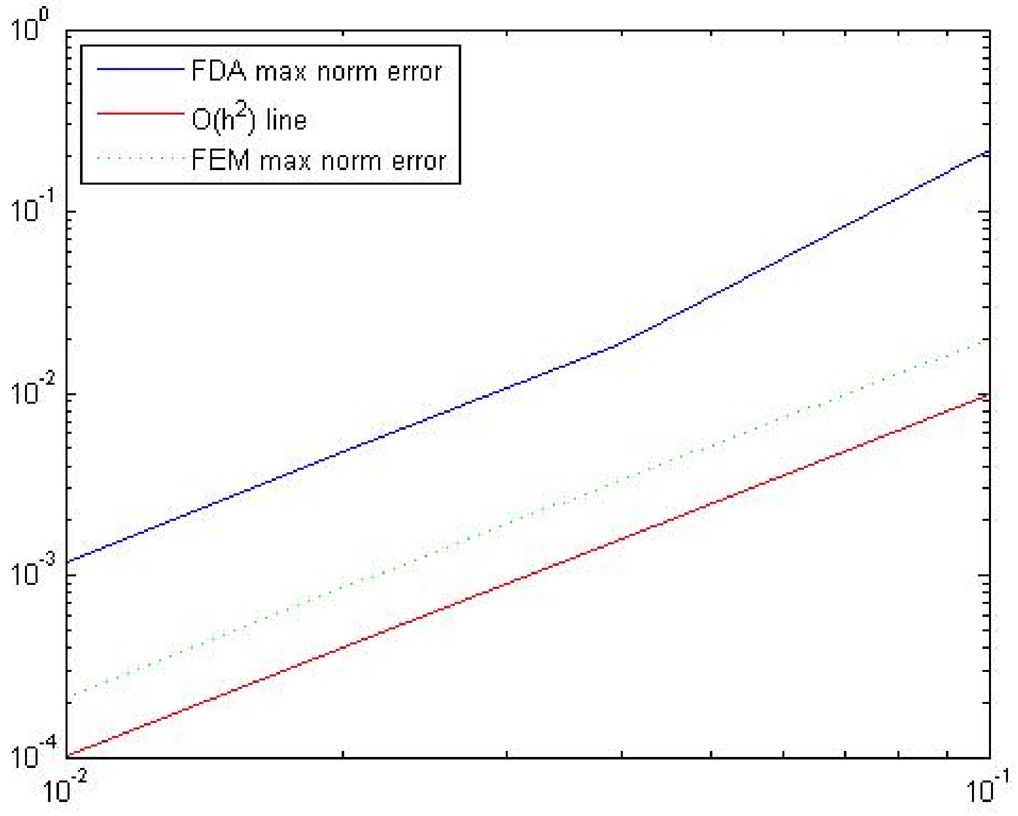


Figure 4: **Both the Finite Difference Approximation method and the Finite Element Approximation exhibit second order global accuracy.**

AMath 585, Winter '16
Midterm

Consider the BVP

$$u'' + u = \sin x \quad u(0) = 0 \quad u(\pi) = 0 \tag{1}$$

Problem 1a) Write down a second order finite difference approximation to this problem

Solution: We will construct a grid function consisting of values $U_0, U_1, ..., U_m, U_{m+1}$ where U_j is our approximation to $u(x_j)$. Here $x_j = jh$ are the m interior grid points and $h = \pi/(m+1)$ is the mesh width. From the boundary conditions we know that $U_0 = 0$ and $U_{m+1} = 1$ and so we have m unknown values $U_1, U_2, ..., U_m$ to compute. We use the usual centered difference approximation for $u''(xj)$ and obtain the following system of equations.

$$\frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1}) + U_j = \sin x_j \quad \text{for} \quad j = 1, 2, ..., m$$

Using $U_0 = 0$ and $U_{m+1} = 0$ we get a linear system of m equations and m unknowns, $AU = f$ where

$$A = \begin{bmatrix} -\frac{2}{h^2}+1 & \frac{1}{h^2} & & 0 \\ \frac{1}{h^2} & -\frac{2}{h^2}+1 & \frac{1}{h^2} & \\ & \ddots & \ddots & \frac{1}{h^2} \\ 0 & & \frac{1}{h^2} & -\frac{2}{h^2}+1 \end{bmatrix}$$

and

$$f = \begin{bmatrix} \sin(x_1) \\ \sin(x_2) \\ \vdots \\ \sin(x_m) \end{bmatrix}$$

Problem 1b) Using a local truncation error analysis, show that your finite difference approximation is consistent.

Solution: The local truncation error denoted by τ_j for $j = 1, 2, ..., m$ is given by

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

where u is the true solution to $u'' + u = \sin x$. Assuming u is smooth enough, we use Taylor series expansion to get

$$\tau_j = \left[u''(x_j) + u(x_j) + \frac{1}{12}h^2 u''''(x_j) + O(h^4)\right] - \sin(x_j)$$

Using (1) this becomes

$$\tau_j = \frac{1}{2}h^2 u''''(x_j) + O(h^4)$$

Using boundedness of u'''' on $[0, \pi]$ we get $\tau_j = O(h^2)$ as $h \to 0$. This shows that the finite difference approximation is consistent.

Problem 1c) Solve your finite difference approximation as a sparse tridiagonal matrix problem with $\frac{h}{\pi} = 2^{-p}$ for $p = 2, 3, 4, 5$. Based on the plots of solution vs h, what can you say about the convergence of finite difference approximation as h decreases.

Solution: The exact solution to $u'' + u = \sin x$ is $u = c_1 \sin x + c_2 \cos x - \frac{1}{2}x \cos x$. Plugging in the boundary conditions tells us that the BVP (1) has no solution. We compute the solution to the BVP numerically using the above finite difference approximation and plot the solutions for different grid spacing. We have attached the plot in figure (1). It is seen that as h decreases the maximum value of the solution blows up. This tells us that the finite difference approximation does not converge as $h \to 0$.

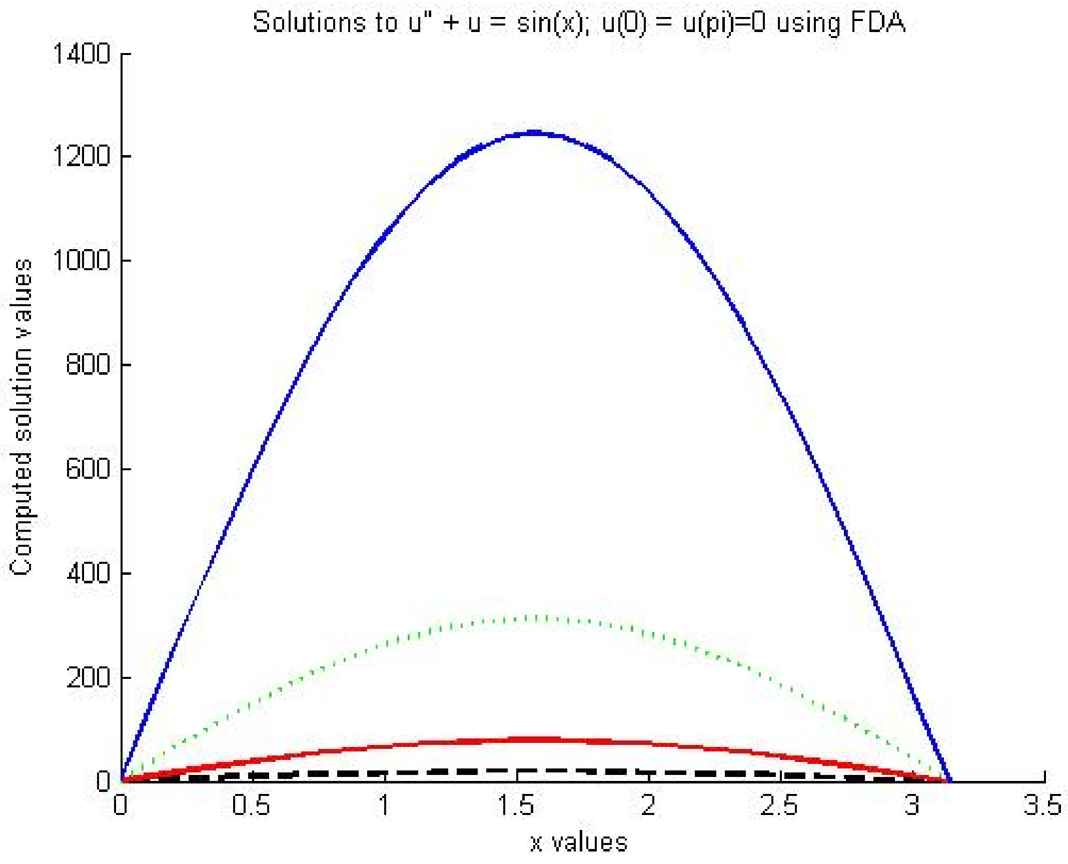


Figure 1: **Plots of computed solutions to $u'' + u' = \sin x$ with $u(0) = 0$ and $u(\pi) = 0$ for different mesh width h using FDA. The blue figure corresponds to $h = \pi 2^{-5}$, the green to $h = \pi 2^{-4}$, the red to $h = \pi 2^{-3}$ and the black to $h = \pi 2^{-2}$.**

Problem 1d) Analyze the 2 norm stability of this finite difference approximation and comment on its relation to your results of part (c).

Solution: The m eigenvalues for the matrix A of Problem 1a) are given by

$$\lambda_p = \frac{2}{h^2}(\cos(ph) - 1) + 1 \quad \text{for} \quad p = 1, 2, ..., m$$

Using Taylor expansion technique as used in RJL we see that $\lambda_1 = O(h^2)$. Thus an eigen value of A approaches 0 as $h \to 0$. Since A is symmetric, the eigen values of A^{-1} are simply the inverses of eigen values of A, so $||A^{-1}||_2 = (\min_{1 \le p \le m} |\lambda_p|)^{-1}$. Since $\lambda_1 \to 0$ as $h \to 0$ we can see that $||A^{-1}||_2$ blows up as $h \to 0$ and hence this discretization is unstable. This is to be expected since by Problem 1c), we do not have convergence for the finite difference approximation. Had the finite difference method been stable then since by Problem 1b) we have consistency, we should have gotten convergence which would have violated the non convergence seen in Problem 1c).

Problem 2) Return to the Burger's equation discussed in Problem 2d) of Homework 3

$$uu' - \epsilon u'' = 0 \quad u(0) = 1 \;\; u(1) = -1 \;\; \epsilon = 0.1 \tag{3}$$

Find and plot an approximate solution derived via the shooting method.. What value do you obtain for $u'(0)$.

Solution: We modify the MATLAB code provided in class for the shooting method. More specifically, we use MATLAB solver **ode45** to find the approximate solution at each iteration and then use secant method to update. Using shooting method for the non-linear BVP (3) gives us the plot attached in Figure (2). Since we know from a previous homework that the solution decreases slowly at $x = 0$ we start our initial guesses for $u'(0)$ with small negative values. On choosing $v^{(0)} = -0.02$ and $v^{(1)} = -0.05$ for $u'(0)$ as our initial guesses, we get the plot shown in Figure (2). Our approximate computed $u'(0)$ turns out to be -0.1280.

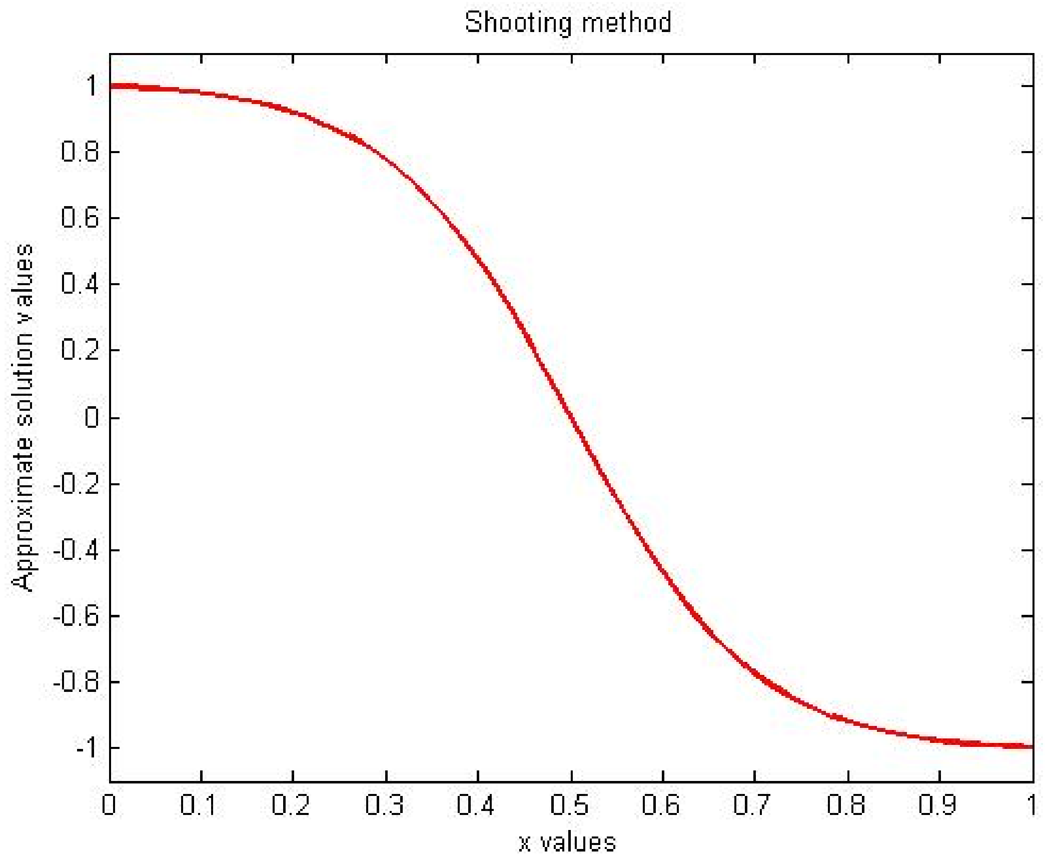


Figure 2: **Computed solution to** $\epsilon u'' - uu' = 0$ **with** $u(0) = 1$**,** $u(1) = -1$ **and** $\epsilon = 0.1$ **using Shooting method.**

Problem 3) Consider the problem $u'' - u = sech(x)$, $-\frac{L}{2} < x < \frac{L}{2}$. As in HW1 we are seeking an approximation of the solution to this ODE on an infinite domain which goes to 0 for large $|x|$. Our approach is to truncate the domain by taking $\frac{L}{2} = 20$ as in HW1, but now we assume periodic boundary conditions. As in HW1, we use the exact solution on the infinite domain as our reference exact solution.

Problem 3a) Implement a Fourier spectral method on this problem for $N = 2^p$ Fourier modes. Plot the solution for $p = 5$ (including the Fourier interpolating function between the grid points, using **interpft**)

Solution: We make use of the MATLAB code given in notes and make appropriate changes for the given boundary value problem. On implementing a Fourier spectral method for the boundary value problem with 2^5 Fourier modes and 2^7 interpolating grid points we get the solution as seen in Figure (3) We have plotted the computed solution, the interpolated computed solution (computed using **interpft**) and the actual solution all on the same plot. It can be seen that even with only 32 grid points we get an accurate solution.

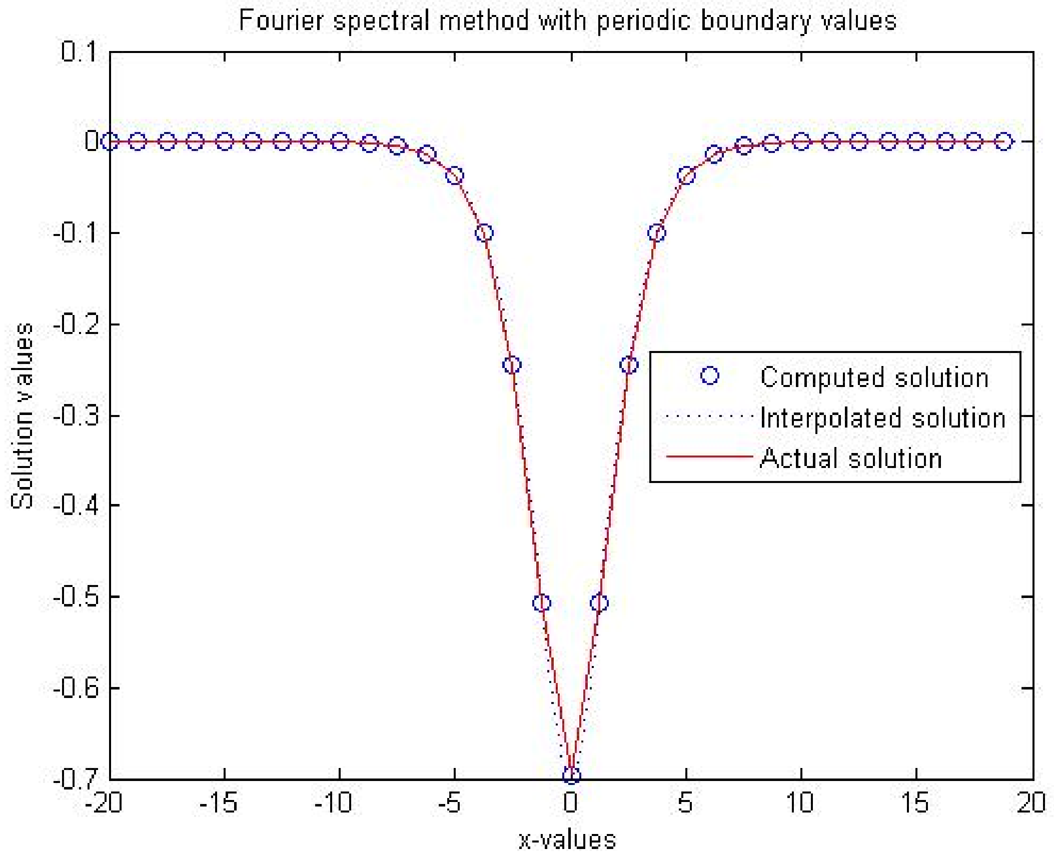


Figure 3: **Solution to $u'' - u = sech(x)$ on $(-20, 20)$ using spectral methods with periodic boundary conditions and 32 grid points and 128 interpolating grid points.**

Problem 3b) On a log-log plot, plot the max-norm of the solution error vs. the grid spacing h for $p = 3, 4, ..., 7$ and discuss the convergence of the method. Do we appear to achieve 'spectral' accuracy?

Solution: We have attached the log-log plot of the max norm of the difference of consecutive solutions at common grid points vs the grid spacing in Figure(4). It is seen that the Fourier spectral method for the given boundary value problem is definitely convergent. We can see that the max norm difference δ between consecutive solutions i.e between solutions evaluated for consecutive values of p at the common grid points decreases faster than any power of h. Thus we indeed achieve spectral accuracy.

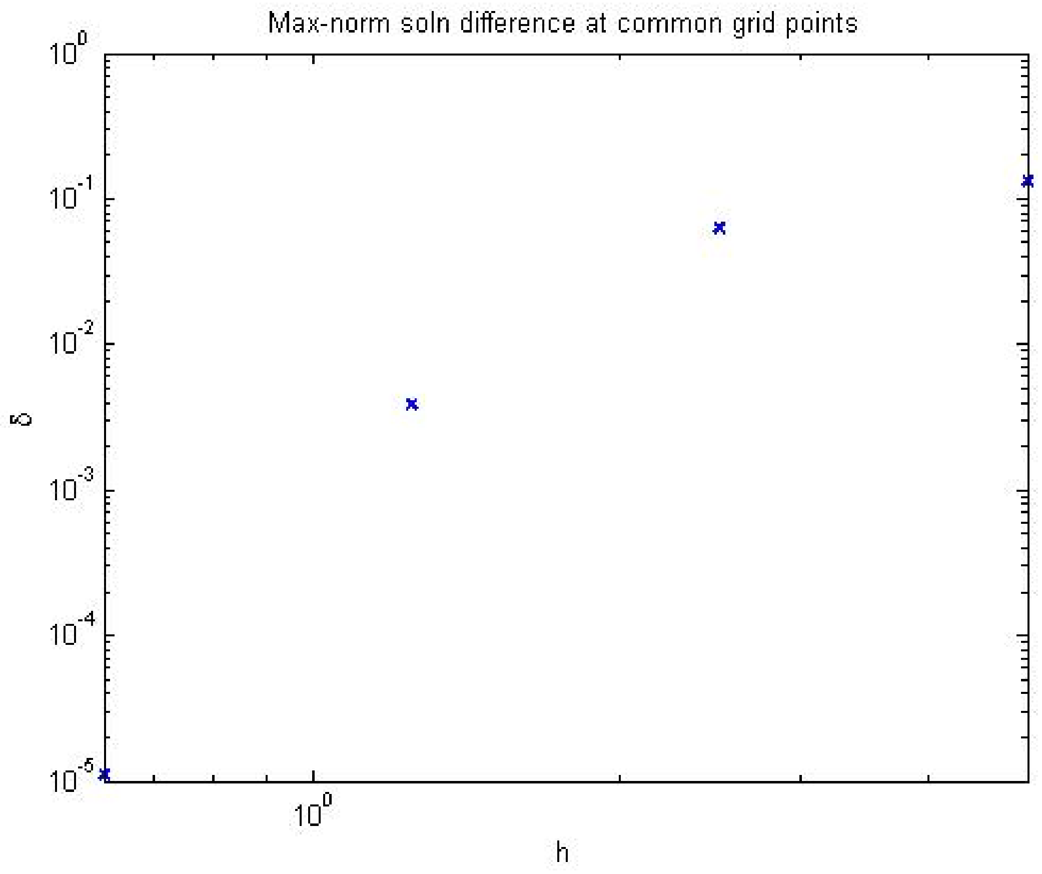


Figure 4: **Error convergence plot. It can be seen that we achieve spectral accuracy. δ dereases faster than any power of h.**

AMath 585, Winter '16
Homework 6

Problem 1 Consider the BVP

$$\epsilon u'' - u' = -1 \quad u(0) = 1 \quad u(1) = 3 \quad \epsilon = 0.01 \tag{1}$$

whose exact solution

$$u_\epsilon(x) = 1 + x + \frac{e^{x/\epsilon} - 1}{e^{1/\epsilon} - 1} \tag{2}$$

has a boundary layer near $x = 1$. Implement a Chebyshev spectral method for (1) using the derivative matrix function cheb.m in the MATLAB lecture scripts on the class web page. Plot the Chebyshev solution for $N = 32$ at and between the grid points. Make a log-log plot of the max-norm of the error for $N = 2^p$ for $p = 3, 4, 5, 6$ and comment on the rate of error convergence.

Solution: We use the helper function *cheb.m* from the class web page and implement a Chebyshev spectral method for the BVP (1) with $N = 32$. We also fit a polynomial of degree N (using *polyfit* and *polyval*) through twice the number of the original Chebyshev grid points. The results are attached in Figure (1). We can see that the polynomial interpolation of the computed solution is not smooth near the right boundary layer and does not seem to approximate the actual solution close to the right boundary point. This is probably due to inability of MATLAB's inbuilt *polyfit* and *polyval* to handle larger number of grid points.

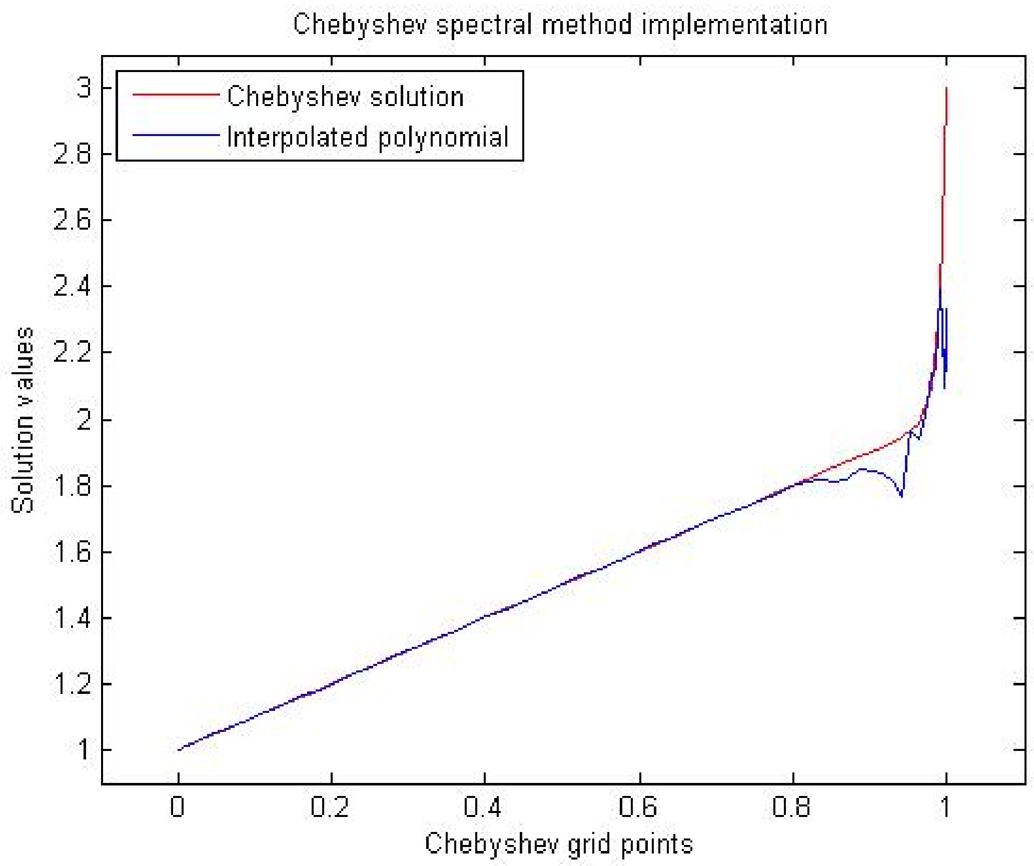


Figure 1: **Plot of Chebyshev solution for $N = 32$ and the interpolated polynomial of degree N between the Chebyshev grid points.**

If we use fewer number for Chebyshev grid points, say $N = 20$ and compute the Chebyshev solution and the interpolated polynomial (again through twice the number of Chebyshev grid points), we get a smoother and a more accurate interpolation. The corresponding plot is attached in Figure (2).

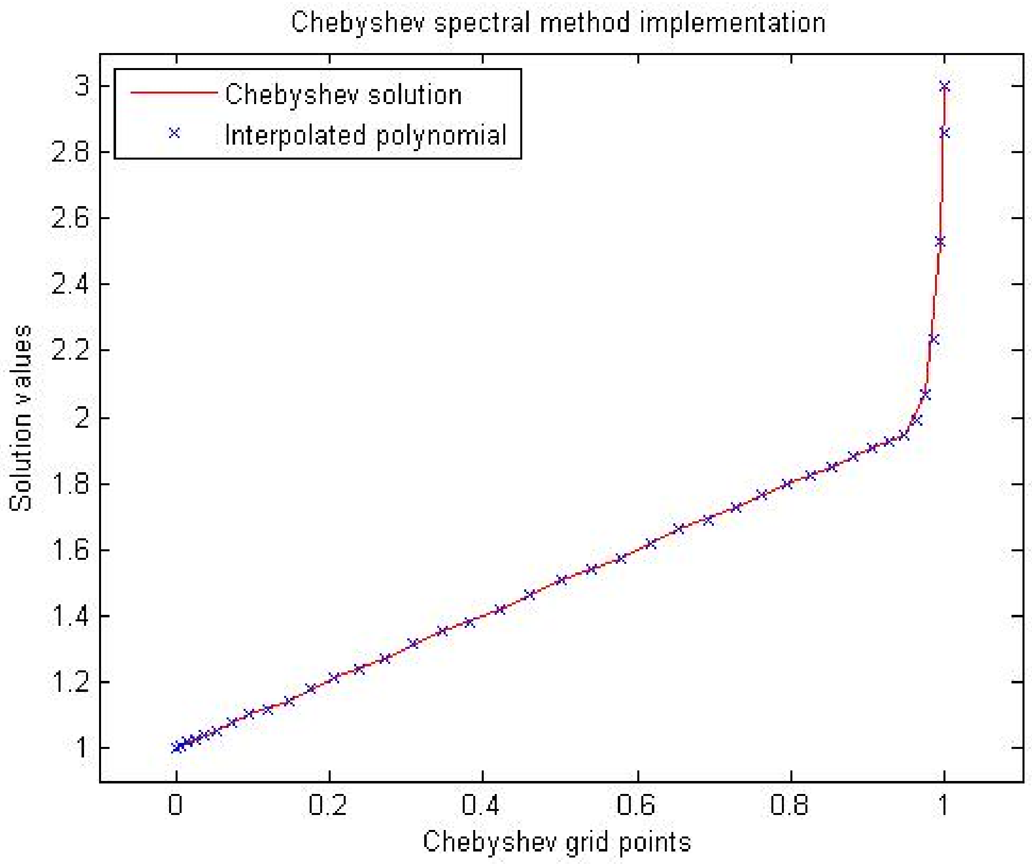


Figure 2: **Plot of Chebyshev solution for $N = 20$ and the interpolated polynomial of degree N between the Chebyshev grid points.**

Using the exact solution (2), we now compute the max norm of the error for $N = 2^p$ for $p = 3, 4, 5, 6$ at the Chebyshev grid points and plot the errors on a log-log plot (vs $h = 1/N$, uniform grid size). We have attached the plot in Figure (3). We can see that spectral accuracy is achieved, the errors go faster to 0 than any polynomial in h as $h \to 0$.

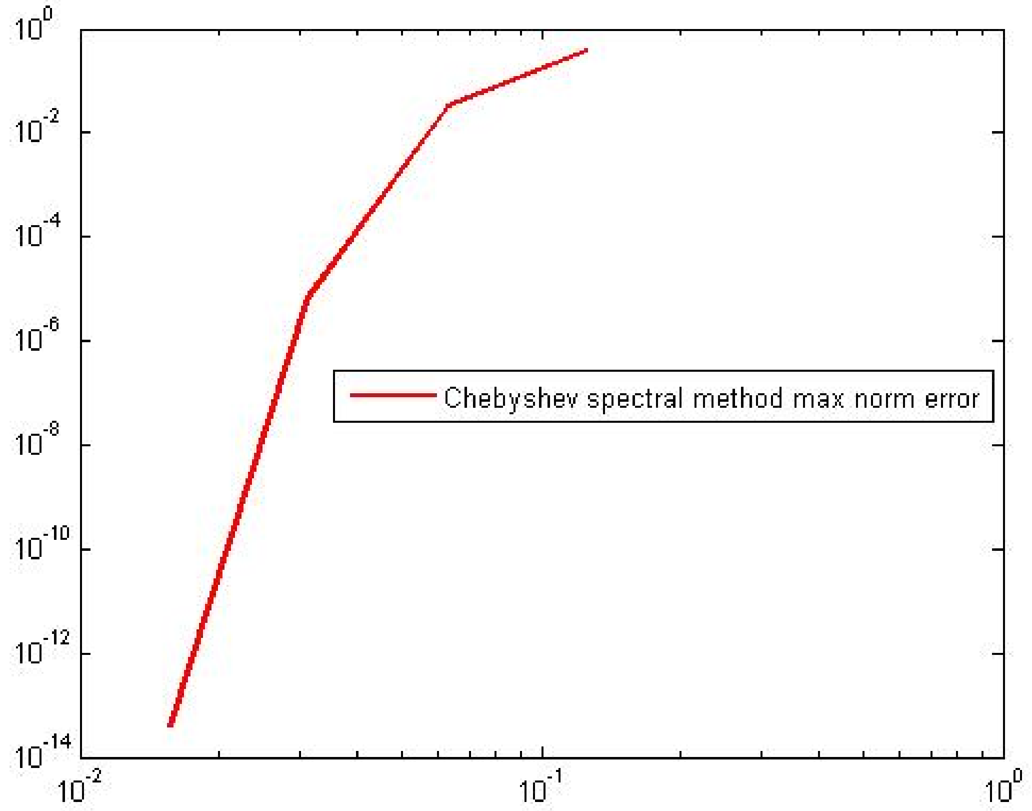


Figure 3: **Log-Log plot of max norm error for Chebyshev spectral method vs h. Spectral accuracy is achieved.**

Problem 2) Implement a sparse direct matrix solver using centered finite difference approximations to second derivatives to solve the Poisson equation on a square

$$u_{xx} + u_{yy} = \sin(\pi x)\sin(\pi y), \quad 0 < x, y < 1 \tag{3}$$

with homogeneous Dirichlet boundary conditions and a uniform grid spacing h in each direction. Make a log-log error plot of the grid 2 norm of the error $h = 2^{-p}$ for $p = 1 - 5$, that shows that your method is second order accurate. Make a surface plot of your finite difference solution $u(x, y)$ vs x, y for $h = 2^{-5}$. Note that the exact solution of (3) is $u_{ex} = (-2\pi^2)^{-1}\sin(\pi x)\sin \pi y)$ for $0 \leq x, y \leq 1$.

Solution: We set up the finite difference method for the given BVP (3) and implement a sparse direct matrix solver. For $h = 2^{-5}$, we have attached a plot of the finite difference solution $u(x, y)$ in Figure (4).

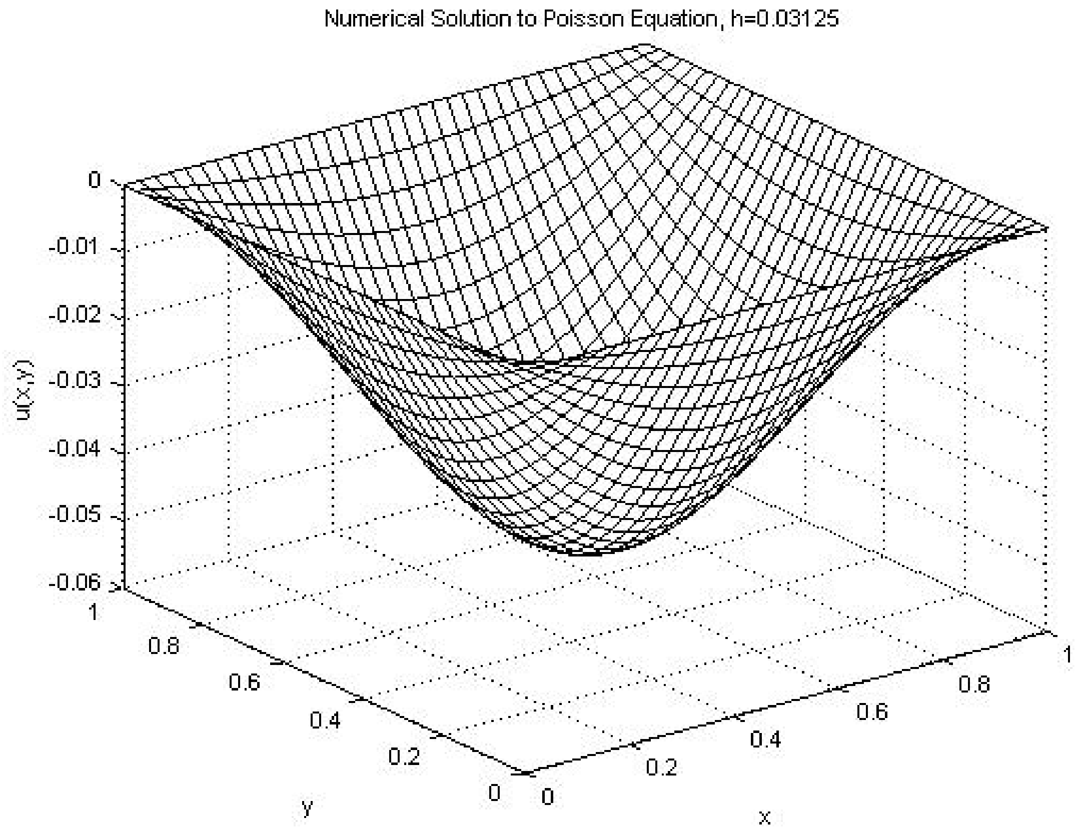


Figure 4: **Plot of numerical solution to BVP (3) with homogenous Dirichlet boundary conditons with grid spacing** $h = 2^{-5} = 0.03125$

Using the given exact solution we compute the grid 2 norm error for different grid spacings h and plot the grid 2 norm error vs h on a log-log scale. We have attached the plot in Figure (5).

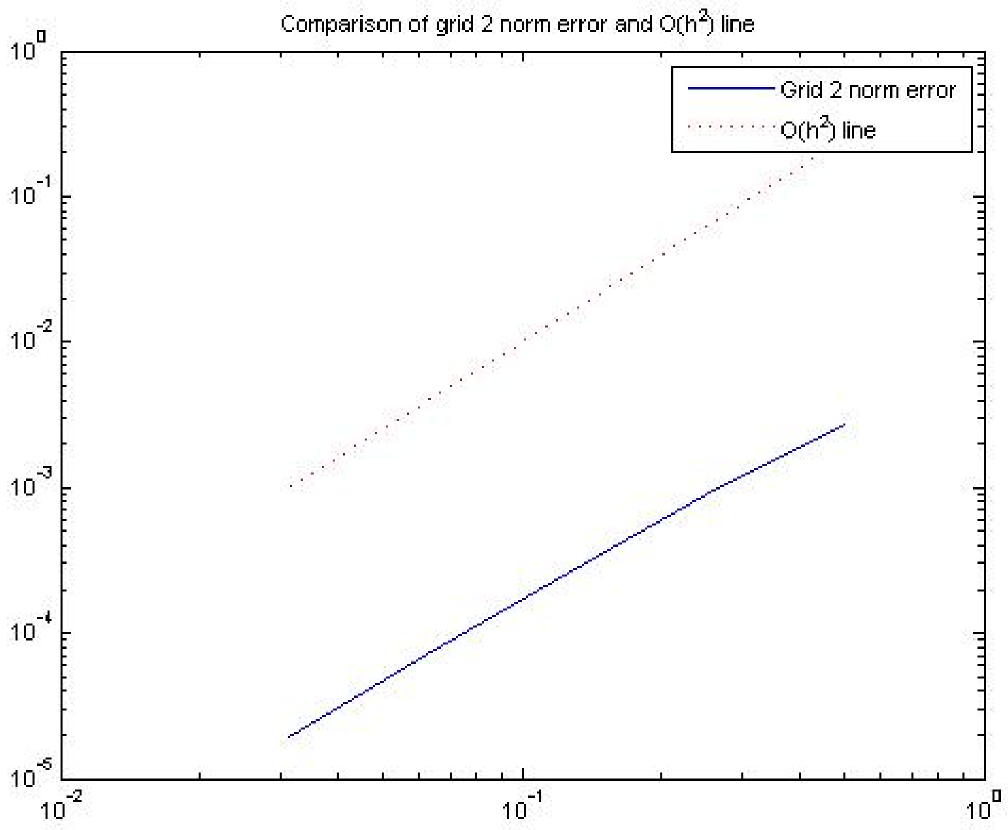


Figure 5: **Log-Log plot of the grid 2 norm error using finite difference for the BVP (3). We can see that we achieve second order accuracy.**

AMath 585, Winter '16
Homework 7

Consider the Poisson equation

$$u_{xx} + u_{yy} = \sin(\pi x)\sin(\pi y) \quad 0 < x, y < 1 \tag{1}$$

Problem 1 Using odd extensions in x and y, apply the Fourier spectral method to this equation. Calculate the max norm of the error of your solution for $h = 2^{-3}$; what do you notice and why?

Solution: We extend the driving function f in odd manner in both x and y. Call the extended driving function $\tilde{f}$. Note that $\tilde{f}$ is defined on $(0, 2) \times (0, 2)$. We solve the Poisson equation on $[0, 2] \times [0, 2]$ with zero Dirchlet boundary conditions and the driving function $\tilde{f}$. We note that the driving function $f = \sin(\pi x)\sin(\pi y)$ originally defined on $(0, 1) \times (0, 1)$ is such that its odd extension $\tilde{f}$ to $(0, 2) \times (0, 2)$ is equal to f on $(0, 2) \times (0, 2)$ i.e odd extension equals the natural extension of f. Note also that f is 1-periodic in x and y and smooth.

When we discretize the problem with $h = 1/8$ and $N = 8$, the grid points are chosen in such a way that the odd extension of f denoted by $\tilde{f}$ which is defined on twice the number of grid points (in each direction) exactly matches the natural extension of f and hence the odd extension does not introduce any unwanted discontinuities in the discretized problem. We thus expect to achieve spectral accuracy on restricting the extended computed solution $\tilde{u}_{comp}$ to $[0, 1] \times [0, 1]$.

When we implement a Fourier spectral method on the Possion problem (1) with zero Dirichlet boundary conditions, we get a plot as seen in Figure (1). On comparing with the exact solution, we see that the max norm error $= 1.040834085586084e - 17$ which is smaller than $\epsilon_{machine}$.

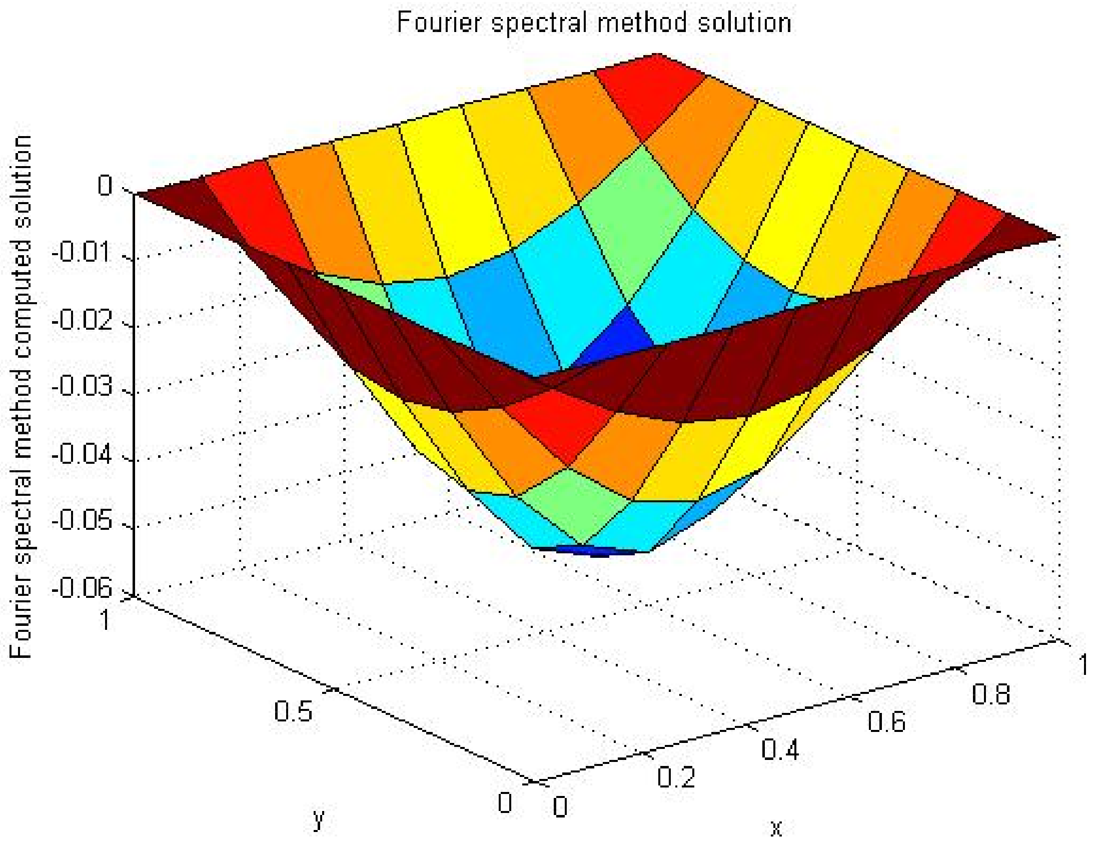


Figure 1: **Plot of the solution to the given BVP (1) with zero Dirichlet boundary data computed using Discrete Fourier transform. Spectral accuracy is achieved.**

Problem 2) Now implement the DFT-based solution technique for the finite difference approximation, starting with the example on the class web page. Show that for $h = 2^{-3}$, the solution is identical to that you obtained in a previous homework using a sparse LU solver.

Solution: We implement a DFT based solution technique for the second order finite difference approximation with $h = 1/8$. We make use of the MATLAB script *pois_FD_FFT.m* given on the class web page and change it appropriately for our problem. We compute the solution for this second order finite difference approximation using the discrete Fourier transform. We have attached a plot of the solution in Figure (2).

The solution turns out to be identical to the one obtained in HW6 using a sparse LU solver. (The max norm difference turns out to 6.245e-17 which is smaller than $\epsilon_{machine}$).

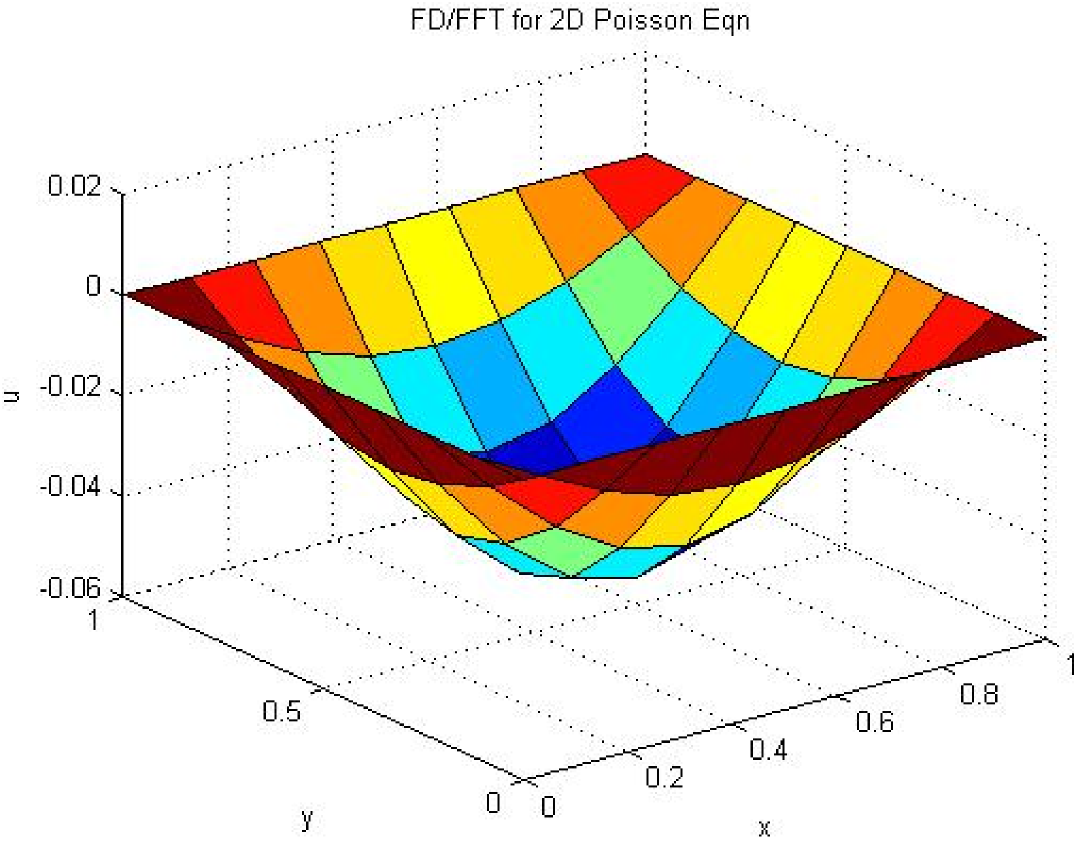


Figure 2: **Plot of the solution to the given BVP (1) with zero Dirichlet boundary data computed using Discrete Fourier transform on a second order Finite Difference method with grid spacing $h = 1/8$ in x and y directions. The solution turns out to be identical (within $\epsilon_{machine}$) to the one obtained by implementing a second order finite difference approximation to BVP (1).**

Problem 3) Consider a Jacobi iteration to solve this problem with the same $h = 2^{-3}$ at all grid points at iteration $k = 0$. Show that at iteration k, the solution has the form

$$u_{ij}^{(k)} = a_k \sin(\pi i h) \sin(\pi j h), \quad 1 \le i, j \le 7 \tag{2}$$

where $a_{k+1} = a_k \cos(\pi h) - \frac{h^2}{4}$ with $a_0 = 0$. Plot a_k vs k for the first 10 iterations and also calculate and overplot the fixed point of this iteration. Has the Jacobi iteration come close to convergence after 10 iterations.

Solution: At stage k, we have

$$u_{ij}^{(k+1)} = \frac{1}{4}\left(u_{i-1,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)} - h^2 f_{ij}\right) \tag{3}$$

We will prove (2) using induction on k. For $k = 1$, we thus need to prove that $u_{ij}^{(1)} = a_1 \sin(\pi i h) \sin(\pi j h)$ where $a_1 = -\frac{h^2}{4}$. Using (3) and the assumption that $u^{(0)} = 0$ we see that $u_{ij}^{(1)} = -\frac{h^2}{4} f_{ij} = -\frac{h^2}{4} \sin(\pi i h) \sin(\pi j h)$. The base case is thus proved.
Assume that (2) holds for $k = n$. We will prove that (2) holds for $k = n + 1$. By (3),

$$u_{ij}^{(n+1)} = \frac{1}{4}\left(u_{i-1,j}^{(n)} + u_{i+1,j}^{(n)} + u_{i,j-1}^{(n)} + u_{i,j+1}^{(n)} - h^2 f_{ij}\right) = \frac{1}{4}\left(a_n \sin(\pi(i-1)h)\sin(\pi j h)\right)$$
$$+\frac{1}{4}\left(a_n \sin(\pi(i+1)h)\sin(\pi j h) + a_n \sin(\pi i h)\sin(\pi(j-1)h) + a_n \sin(\pi i h)\sin(\pi(j+1)h)\right)$$
$$-\frac{h^2}{4}\sin(\pi i h)\sin(\pi j h)$$

Using trigonometric identities, we can simplify the above expression to get

$$u_{ij}^{(n+1)} = \sin(\pi i h)\sin(\pi j h)\left(\frac{a_n}{2}\cos(\pi h) + \frac{a_n}{2}\cos(\pi h) - \frac{h^2}{4}\right)$$

Since $a_{n+1} = a_n \cos(\pi h) - \frac{h^2}{4}$, $u_{ij}^{(n+1)} = a_{n+1} \sin(\pi i h) \sin(\pi j h)$ thereby proving (2).

It is easy to see that a_k must $\to \frac{-1}{2\pi^2}$ as $k \to \infty$ and $h \to 0$. In Figure (3) we have attached a plot of a_k vs k for the first 10 iterations along with with the line $y = -\frac{1}{2\pi^2}$. Since only a_k changes with k in the formula for the iterates u^k, it suffices to study the convergence of a_k to get information about the convergence of the Jacobi iterates. It can be seen from the Figure (3) that as k increases the rate of convergence of a_k to $-\frac{1}{2\pi^2}$ decreases. The Jacobi iteration has hence not come close to the actual solution after only 10 iterations. (We need more iterations to come close to the actual solution.)

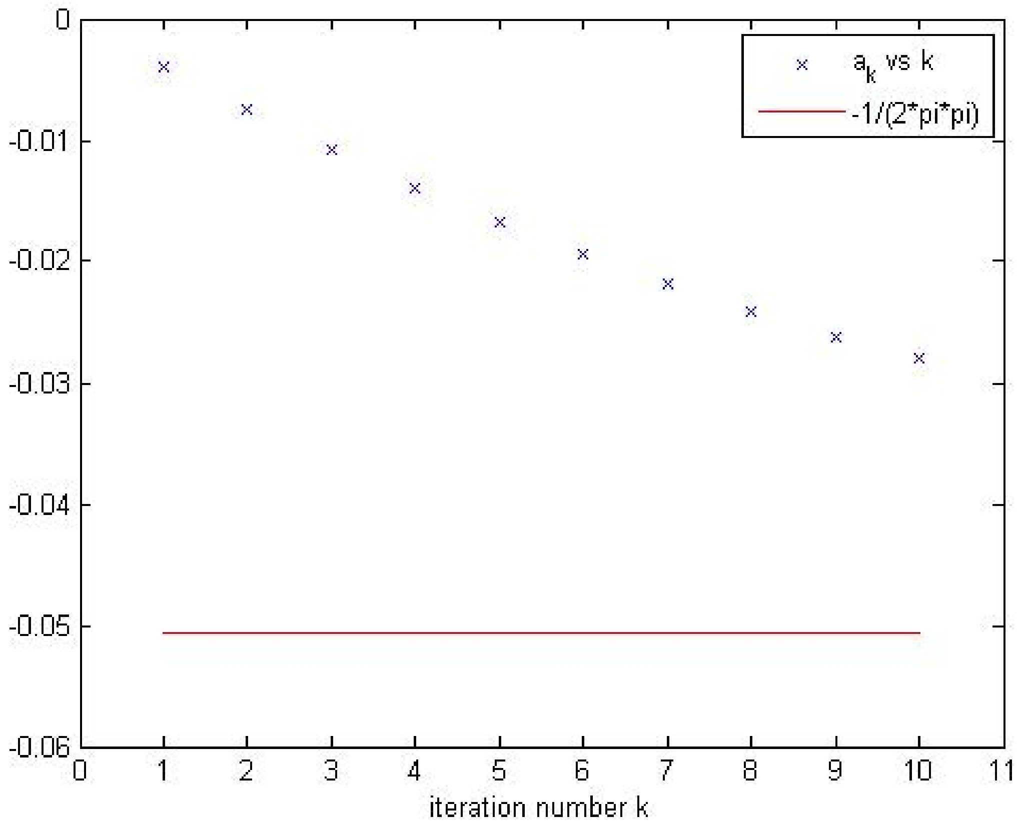


Figure 3: **Plot of a_k vs k for $k = 1 : 10$ and the line $y = -\frac{1}{2\pi^2}$.**

AMath 585, Winter '16
Homework 8

Consider the Poisson equation

$$u_{xx} + u_{yy} = 0 \tag{1}$$

in the right triangle bounded by the lines $x = 0$, $y = 0$ and $x + y = 1$ with boundary conditions $u(x, 0) = u(y, 0) = 0$ and $u(x, 1 - x) = \sin(\pi x)$

Problem 1a) Write down a second order accurate centered finite difference approximation for this BVP with uniform grid spacing $h = 1/m + 1$. How many unknowns are there?

Solution: We can use the usual 5 point stencil for the Laplacian to get the following finite difference scheme

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = 0 \tag{2}$$

where $1 \leq i \leq m - 1$ and $1 \leq j \leq m - i$. This gives us a total of $\frac{m(m-1)}{2}$ unknowns. It is easy to see that the finite difference scheme proposed above is second order accurate. (It is simply the usual second order accurate finite difference scheme for the Laplacian but on a non-square domain.)

Problem 1b) In the resulting matrix problem $Au = f$, argue why A is symmetric. Plot the sparsity structure of A for $h = 2^{-3}$ using *spy(A)*. Note that it is less simple than for a square domain.

Solution: We use the finite difference scheme (2) on the given right angled triangle and form the matrix A using the usual lexicographic ordering of the unknowns. Let $a_{l,k}$ denote the $l - k$ entry of A. We will just concentrate on off-diagonal entries i.e $k \neq l$.

Because we have rolled the unknowns in to a column vector with $\frac{m(m-1)}{2}$ unknowns using lexicographic ordering, we observe that $a_{l,k} = 1$ iff the l^{th} unknown is a neighbor (east, west, south or north) of the k^{th} unknown and is 0 otherwise. The property of being a neighbor is symmetric in k and l. This implies that if k^{th} and the l^{th} unknowns are neighbors of each other, $a_{k,l} = a_{l,k} = 1$, otherwise $a_{k,l} = a_{l,k} = 0$. In either case, $a_{l,k} = a_{k,l}$ thereby showing that A is symmetric.

(This argument works if we order the unknowns in a particular way. If we use a different ordering, the symmetric nature is still be preserved and will ultimately depend only on two unknowns being neighbors of each other. We can see that for non-diagonal entries, the "weights" attached to the neighbors are all the same, for this finite difference the weights are all 1. The symmetric nature of A ultimately depends on this observation.)

We have attached a plot in Figure (1) of the sparsity structure for A when $h = 2^{-3}$ i.e A is a 21×21 matrix. A is still a sparse matrix since only at most 5 entries in each row will be non-zero.

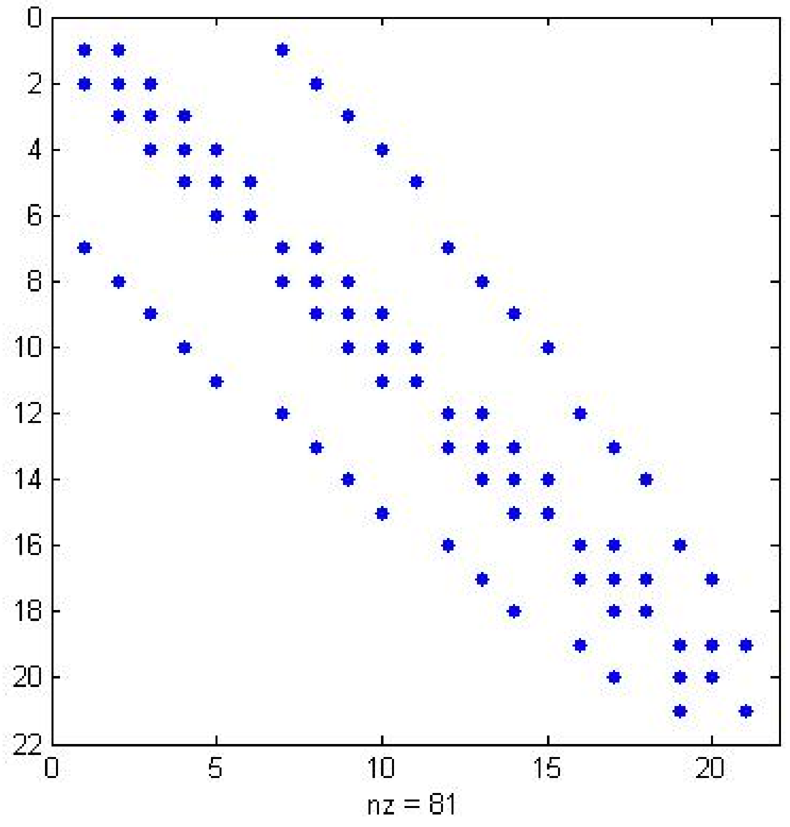


Figure 1: **Sparsity structure for the matrix A when $h = 2^{-3}$ i.e $m = 7$. The matrix A is a sqaure matrix of size $m(m+1)/2$ i.e 21×21 matrix with 81 non-zero entries.**

For $h = 2^{-p}$, $p = 3, 4, 5, 6$:
Problem 2a) Using *eig(A)*, calculate and log-log plot the eigen-values λ_n vs n, ordered in increasing order.

Solution: We compute the eigenvalues of A for each of the specified values of h. If n denotes the size of matrix A, ($n = \frac{(\frac{1}{h}-1)(\frac{1}{h}-2)}{2}$) , the eigenvalues of A are arranged in the order $0 > \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. Thus $|\lambda_n| = ||A||_2$. For $h = [2^{-3}\ 2^{-4}\ 2^{-5}\ 2^{-6}]$, $\lambda_n = [7.2620 \quad -7.8093 \quad -7.9519 \quad -7.9880]$.

We have attached a log-log plot in Figure (2) of λ_n vs n.

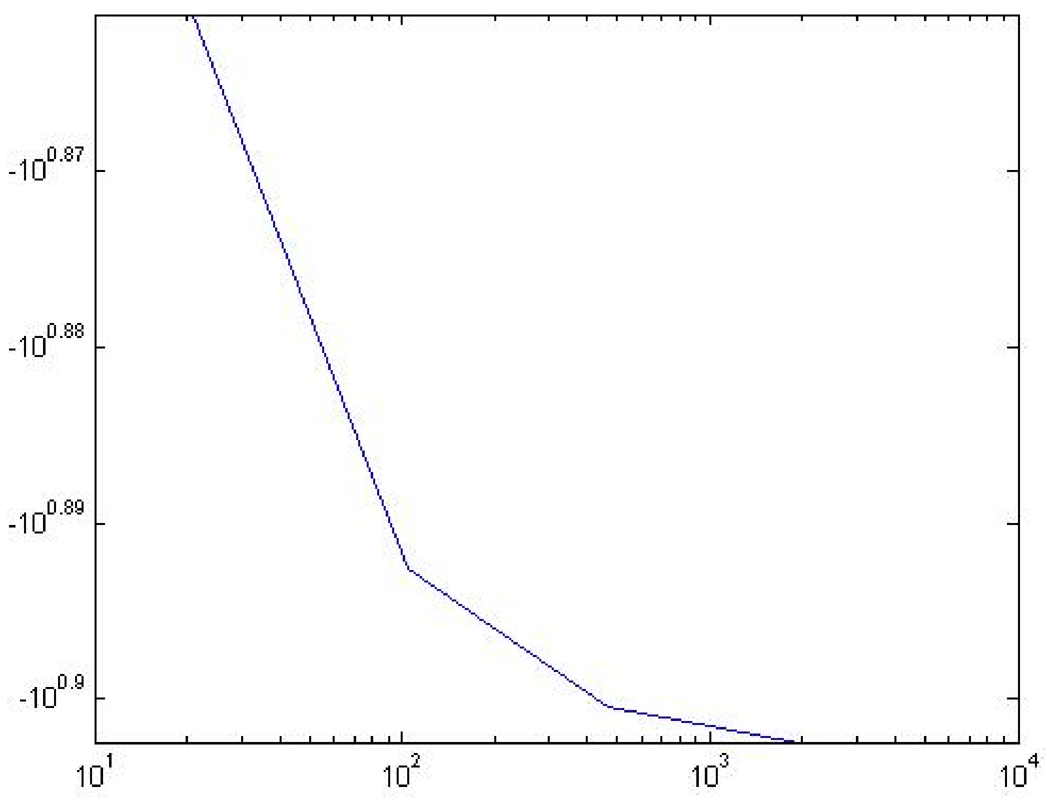


Figure 2: **Log-log plot of λ_n vs n where n is the size of the matrix A. Note that $\lambda_n < 0$**

Problem 2b) Semilogx plot λ_1 vs h, and estimate the value to which λ_1 converges as $h \to 0$. This is the smallest magnitude eigenvalue of

$$u_{xx} + u_{yy} = \lambda u \quad u = 0 \text{ on } \partial\Omega$$

Solution: We compute the smallest magnitude eigen value of A for each of the specifed values of h. The smallest magnitude eigenvalue for A for $h = 2^{-p}$, $p = 3, 4, 5, 6$ turn out to be $[-0.7380 \quad -0.1907 \quad -0.0481 \quad -0.0120]$ respectively. We have attached a semilogx plot of λ_1 vs h in Figure (3).

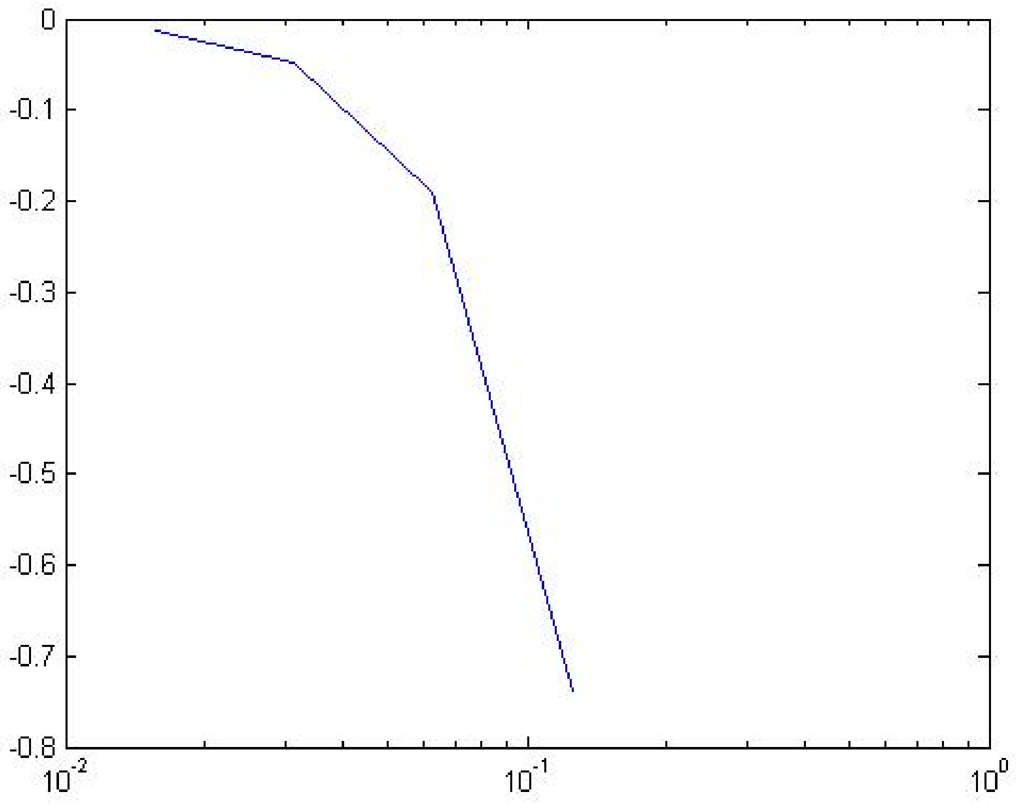


Figure 3: **Semilogx plot of λ_1 vs h where λ_1 is the smallest magnitude eigenvalue of A. Note that $\lambda_1 < 0$**

It is unclear where λ_1 converges. I am not sure what the smallest eigenvalue of the Laplacian on the given triangular domain is. By theory, λ_1 should converge to a non-zero value. (The laplacian is a negative definite symmetric operator and as such the smallest magnitude eigenvalue must be no-zero). Experimenting with smaller values of h indicates that λ_1 decreases as h decreases. It is not clear experimentally whether λ_1 stabilizes.

Problem 2c) Make a log-log plot of the condition number of A vs h. Add a fit line that shows that the condition number increases roughly as $O(h^{-2})$, as in a square domain.

Solution: We have attached a log-log plot of the condition number of A vs h in Figure (4) along with an $O(h^{-2})$ fit line.

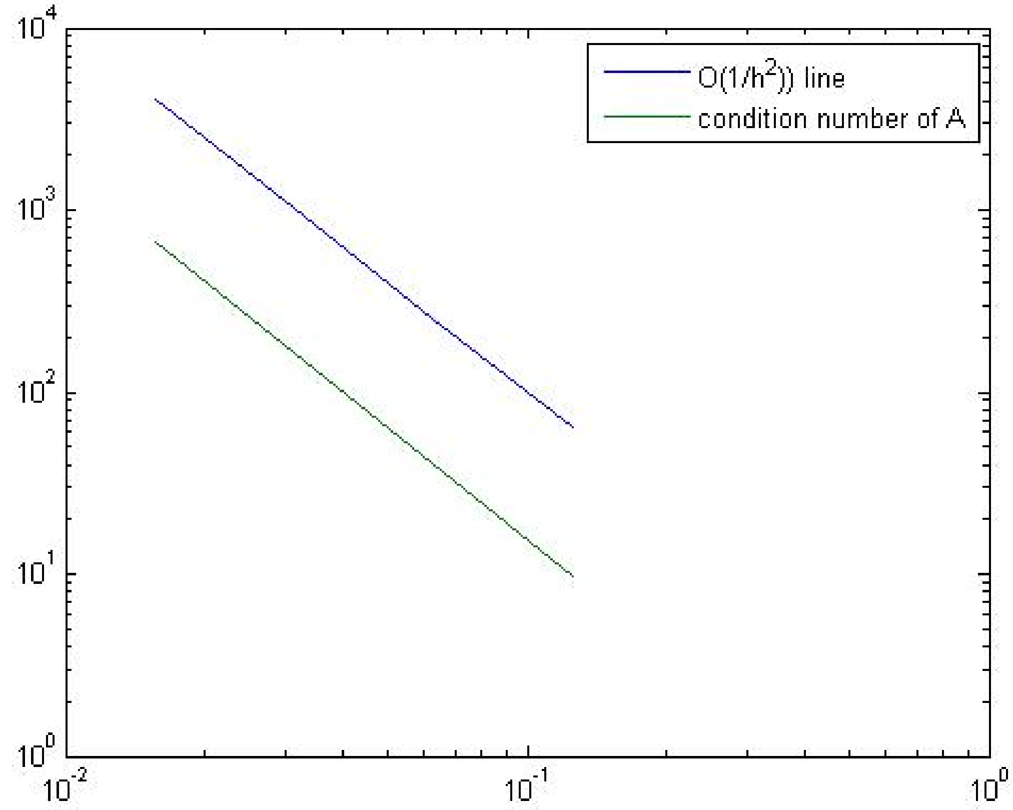


Figure 4: **Log-log plot of condition number of A vs h. It can be seen that the condition number increases as $O(h^{-2})$ as in the case of the square domain.**

For $h = 2^{-p}$, $p = 3, 4, 5, 6$

Problem 3a) Use the conjugate gradient method as implemented in the function $CG.m$ on the class web page to iteratively solve $Au = f$ to a relative tolerance of $0.1h^2$, starting with a guess $u_0 = 0$. Set the maximum allowed iterations to 100. Log-log plot the 2-norm of the relative residual $\delta = \frac{||\mathbf{r}||_2}{||\mathbf{f}||_2}$ vs the iteration number k. Roughly how does the number of iterations required for convergence scale with h.

Solution: We first form the RHS vector f using the given boundary conditions. Using the helper function $CG.m$ and the matrix A constructed in Problem (1), we solve $Au = f$. In Figure (5), we have attached a log-log plot of 2 norm of the relative residual vs the number of iterations required to reach the specified tolerance level i.e the iteration number k is first time where delta is less than $0.1h^2$.

It is observed that in each case k turns out to be $\frac{1}{h} - 1$.

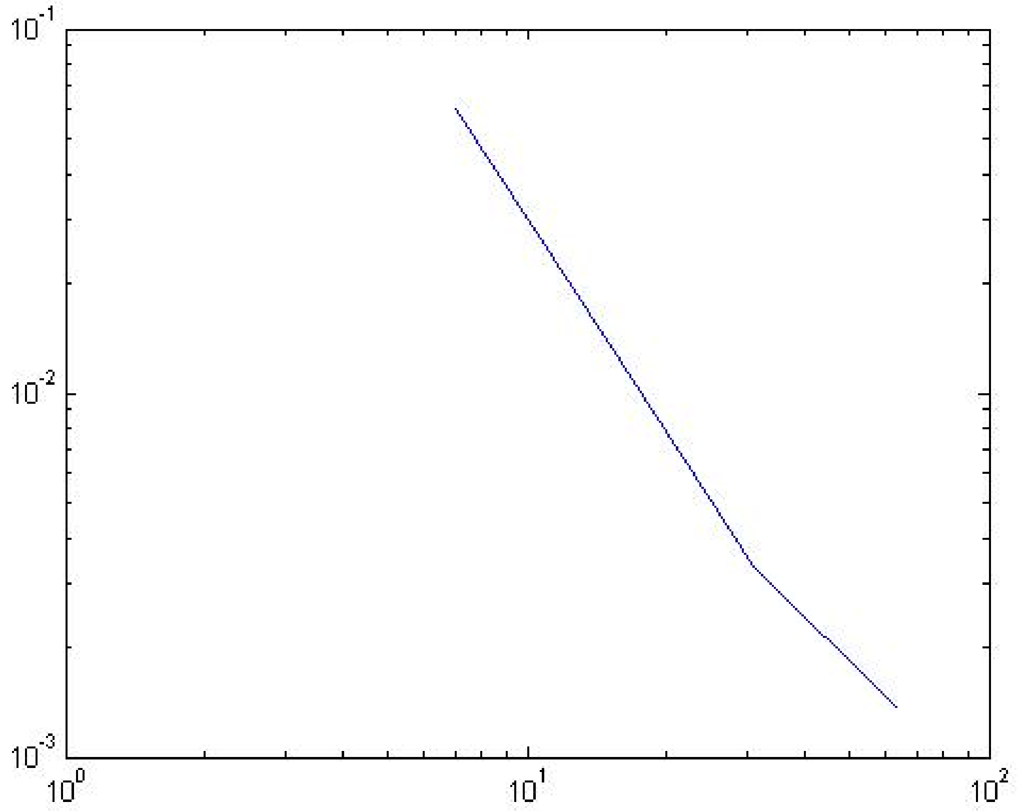


Figure 5: **Log-log plot of the 2 norm of the relative resdual vs the number of iterations k.**

Problem 3b) Make a surface plot of your solution on the triangular domain for $h = 2^{-4}$.

Solution: For $h = 2^{-4}$, $m = 15$. We first form the matrix A of size $m(m-1)/2$ and the RHS vector f. We run the conjugate gradient algorithm and get the vector of unknowns u. Using lexicographic ordering, we recast the $m(m-1)/2 \times 1$ vector u in to a square matrix U of size m. We populate the value of unknowns outside our domain with NaN for plotting purposes. We have attached the resultant surface plot in Figure (5).

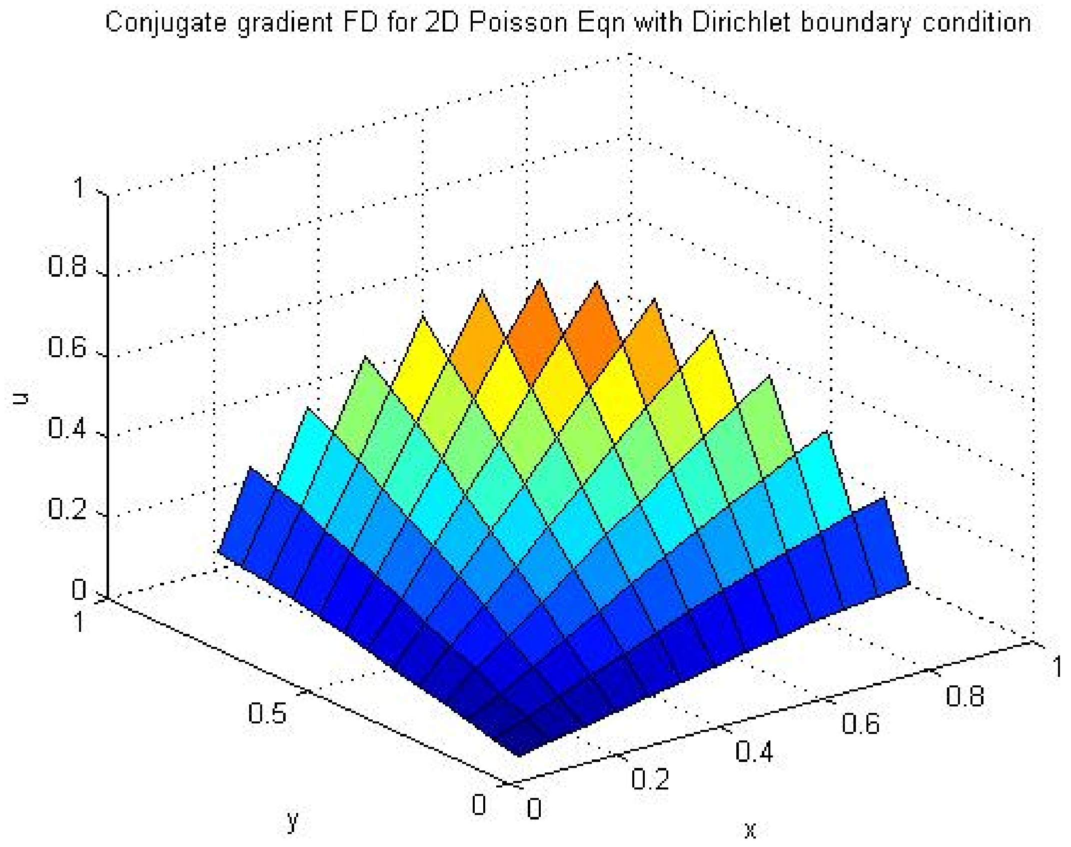


Figure 6: **Surface plot for the solution u inside the triangular domain for $h = 2^{-4}$ using conjugate gradient algorithm.**

AMath 585, Winter '16
Final Examination

Problem 1a) Consider the Laplace equation on the interior of a circle of radius 1. We use polar co-ordinates (r, θ). A Dirichlet boundary condition is imposed.

$$u(1,\theta) = \frac{\cos(\theta)}{0.3 + \cos^2(\theta)}, \quad 0 \le \theta \le 2\pi$$

By seperation of variables, the general bounded solution to the Laplace equation has the form

$$u(r,\theta) = \sum_{M=-\infty}^{\infty} U_M r^{|M|} e^{iM\theta}$$

Use a 64 point Discrete Fourier Transform (DFT) on the boundary condition to estimate the U_M, calculate and make a surface plot of the solution inside the circle. Estimate the accuracy of this solution by taking its max norm difference with a solution calculated with a 128 point DFT. As a check, this should be $O(10^{-9})$.

Solution: Since we are using a N point DFT on the boundary condition, our computed solution will be of the form

$$u_{comp} = \sum_{k=-(\frac{N}{2}-1)}^{\frac{N}{2}-1} U_k r^{|k|} e^{ik\theta}$$

Since for $k < 0$, we have $U_k = U_{N-|k|}$, we can rewrite the above expression as

$$u_{comp} = U_0 + \sum_{k=1}^{\frac{N}{2}-1} r^k (U_k \, e^{ik\theta} + U_{N-k} e^{-ik\theta}) \tag{1}$$

If the discrete Fourier Transform of $h(\theta) = u(1,\theta)$ is denoted by $\hat{h}$ where $\hat{h}$ is a $N \times 1$ complex vector, then we know that $U_0 = \frac{\hat{h}(1)}{N}$ and for $1 \le k \le \frac{N}{2} - 1$, $U_k = \frac{\hat{h}(k+1)}{N}$ and $U_{N-k} = \frac{\hat{h}(N-k+1)}{N}$. Using these relations and equation (1) we compute the approximate solution of the Laplace equation in the unit disk.

We have attached a surface plot for $N = 64$ in Figure (1). For plotting, we have taken a uniform θ grid of size $(N+1)$ with $0 \le \theta \le 2\pi$ and a uniform r grid of size 40 with $0 \le r < 1$.

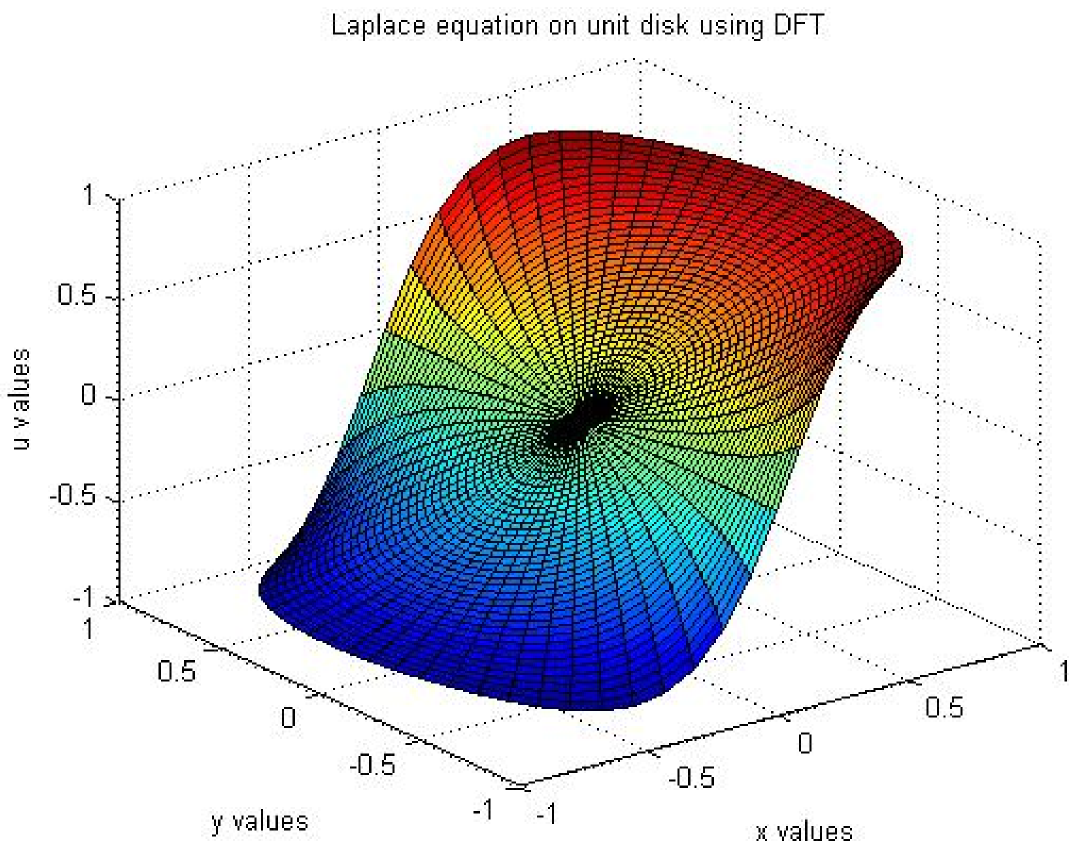


Figure 1: **Solution of the Laplace equation $u(r,\theta)$ on the unit disk with $u(1,\theta) = \frac{\cos(\theta)}{0.3+\cos^2(\theta)}$, $0 \le \theta \le 2\pi$. We use a 64 point DFT on the boundary condition**

We take the max norm difference with a solution calculated with a 128 point DFT and find that the max norm error is 2.5060×10^{-9} confirming the spectral accuracy of the solution.

Consider the Laplace equation:

$$u_{xx} + u_{yy} = 0 \text{ on } 0 < y < 1 \text{ and } y < x < x+1 \tag{2}$$

with zero boundary conditions on all sides of the parallelogram except on the side which is part of the line $y = x$ where we prescribe $u(y,y) = \sin(\pi y)$

Problem 2a) Write down a second order finite difference approximation for this BVP with uniform grid spacing $h = \frac{1}{m+1}$, implement it in terms of a sparse $m^2 \times m^2$ matrix A and plot the sparsity pattern of A for $h = \frac{1}{8}$ using MATLAB's *spy* function.

Solution: We can use the usual 5 point stencil for the Laplacian to get the following finite difference scheme

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = 0 \tag{3}$$

where $1 \leq i, j \leq m$. This gives us a total of m^2 unknowns. It is easy to see that the finite difference scheme proposed above is second order accurate. (It is simply the usual second order accurate finite difference scheme for the Laplacian but on a non-square domain.)

To get the matrix A we use the usual row wise ordering. For $h = \frac{1}{8}$ i.e $m = 7$, the sparsity pattern for A is attached in Figure (2).

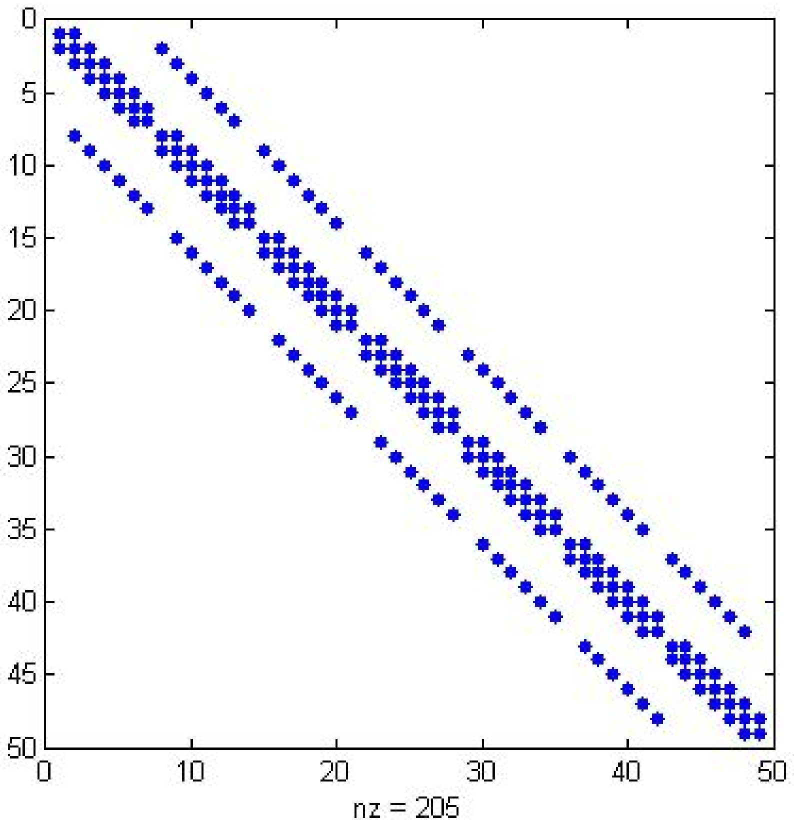


Figure 2: Sparsity pattern for A when $h = \frac{1}{8}$.

Problem 2b) For $h = 2^{-p}$, $p = 3, 4, 5, 6$, use the conjugate gradient method as implemented in the function $CG.m$ on the class webpage to iteratively solve $A\mathbf{u} = f$ starting with an initial guess of $\mathbf{u_0} = \mathbf{0}$. For $p = 3, 4, 5$, calculate the grid 2 norm of the difference between the solutions with grid spacings h and $h/2$ and log-log plot this vs h. As a check on your work, this plot should verify that your solution is approximately second order accurate.

Solution: For each $h = 2^{-p}$ with $p = 3, 4, 5, 6$ we form the matrix A. The RHS vector f is formed using the boundary conditions on the line $y = x$. We then run the conjugate gradient algorithm to find the vector of unknowns u. We recast this vector u into a square matrix of size m^2 where $h = \frac{1}{m+1}$.
To compute the error, we note that if A_1 is the matrix corresponding to grid size h and A_2 is the matrix corresponding to grid size $\frac{h}{2}$, then the value of the unknowns at the common grid points will given by A_1 and (using MATLAB's notation) $\tilde{A}_2 = A_2(2:2:end, 2:2:end)$. We then compute the grid 2 norm of $A_1 - \tilde{A}_2$.

Unfortunately, on plotting the log-log plot of the errors vs h, we only get an $O(h)$ fit as can be seen in Figure(3). (It is not clear why we don't get second order accuracy. I am fairly confident that the construction of A and f is correct and so is the logic for finding out values at common grid points.)

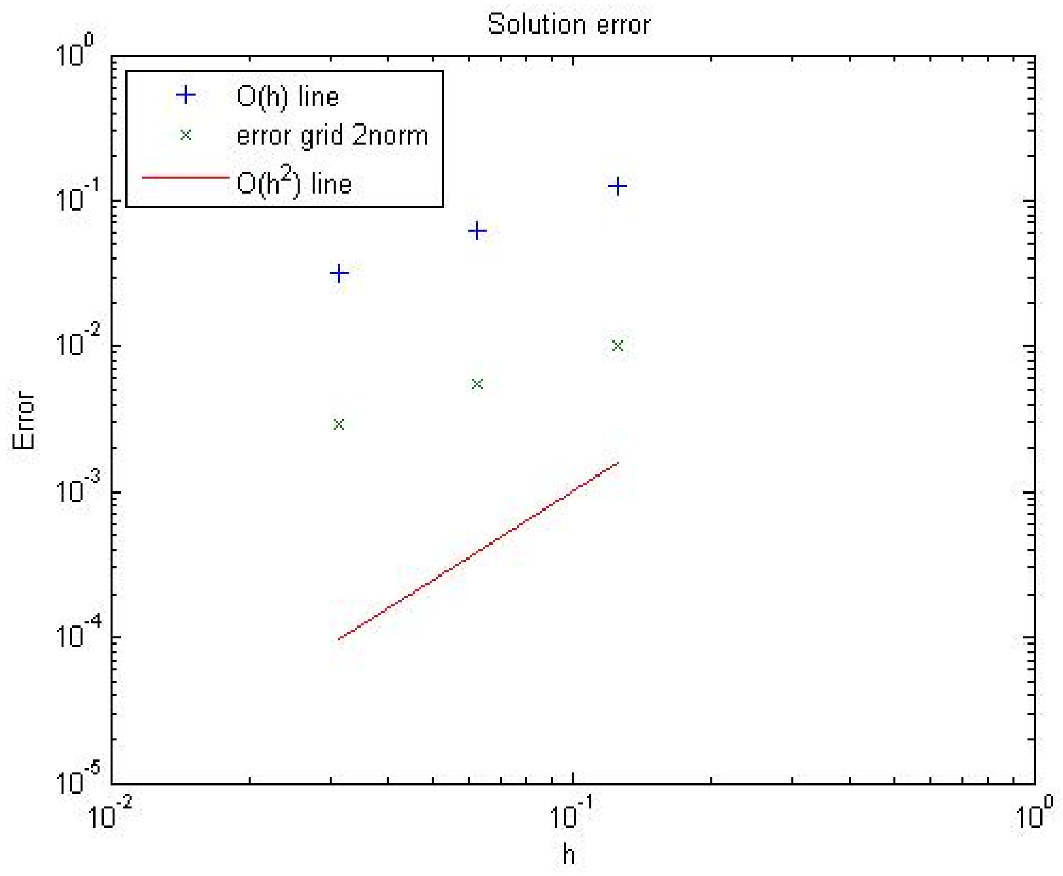


Figure 3: **Log-log plot of the grid 2 norm of the errors at the common grid points vs h**

Problem 2c) Make a surface plot of your solution on the given domain for $h = 2^{-4}$. Using this grid what is your estimate for $u(1, 0.5)$?

Solution: We compute the solution u at the grid points for $h = 2^{-4}$ and convert the solution in to a matrix of size 225×225 using MATLAB's inbuilt *vec2mat* function. We note that this conversion is consistent with our ordering i.e the ij^{th} entry of our solution matrix corresponds to the value of the solution at the ij^{th} grid point according to our ordering. To use MATLAB's *meshgrid* we first map the square $[0, 1] \times [0, 1]$ onto $\bar{\Omega}$ using the transformation $(u, v) \to (u + v, v)$ and then use *meshgrid* on the variables $(u + v, v)$. This enables us to the get the surface plot as attached in Figure (4). Using the figure, the approximate value of $u(1, 0.5)$ is 0.1468

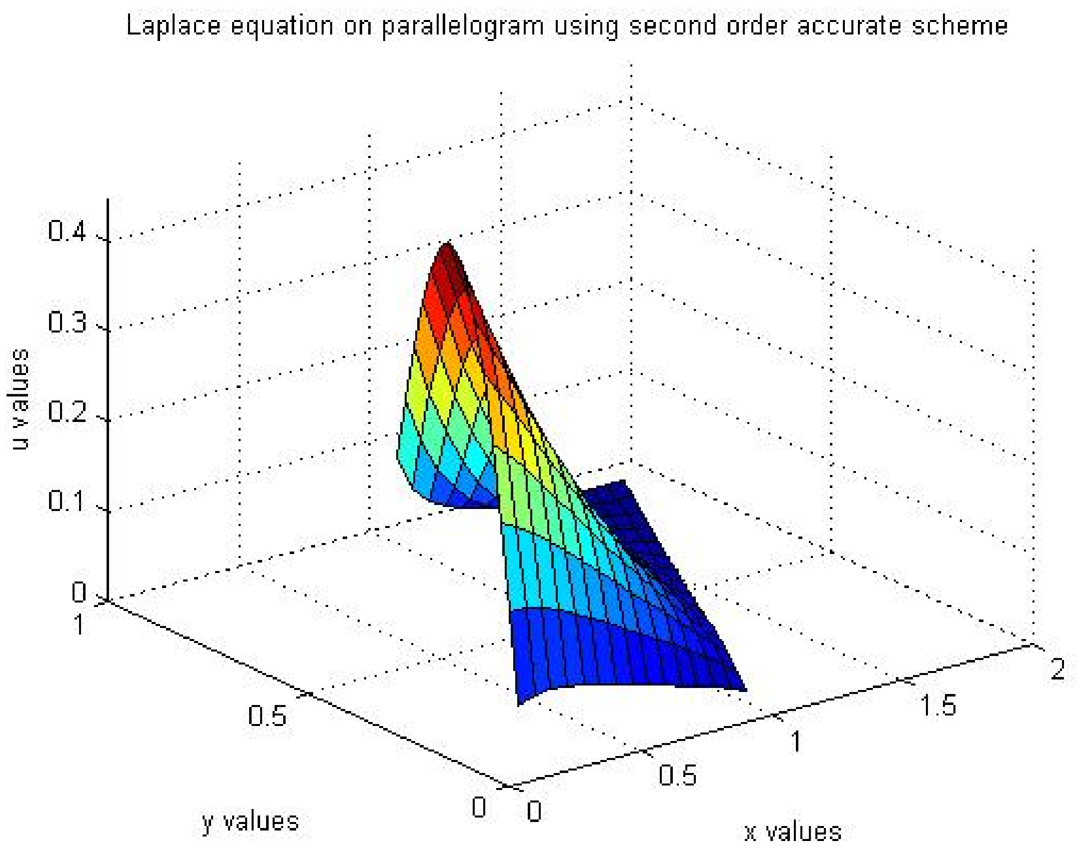


Figure 4: **Plot of the solution u on Ω using second order centered finite difference scheme for the Laplace equation on Ω with given boundary conditions on $\partial\Omega$ with $h = 2^{-4}$.**