Department of Scientific Computing
Written Preliminary Examination
Summer 2017

May 22–25, 2017

Instructions:

- Solve only 10 of the 11 questions as completely as you can.
- All questions are weighted equally.
- All parts of a question are weighted equally unless stated otherwise.
- If a particular question requests code, please email it (with your Student ID) to the person responsible for the question and Dr. Xiaoqiang Wang (wwang3@fsu.edu). The person responsible is listed at the beginning of each question.
- If you use web sources, please list them clearly.
- The exam is due back to Karey Fowler no later than 1:00 pm on Thursday, May 25, 2017; no exceptions allowed.
- If you have any questions related to this exam as you work on it, please send an e-mail to the person responsible, *and* Dr. Xiaoqiang Wang (wwang3@fsu.edu).
- Write your Student ID on each of your answer sheets. Do not write your name on your answer sheets. When turning in your exam, include a cover page with your name and Student ID.

Q1. **Linear Algebra** (Dr. Peterson)

In this problem we want to approximate the dominant eigenvalue of $A\vec{x} = \lambda\vec{x}$.

1. Apply the Power Method to each of the matrices

$$A_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix} \qquad A_2 = \begin{pmatrix} 3 & 3 & 0 \\ 3 & 5 & 0 \\ 0 & 0 & 7 \end{pmatrix} \qquad A_3 = \begin{pmatrix} 3 & 1 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

to find the dominant eigenvalue, if possible. Use $\vec{x}^0 = (1,1,1)^T$ as an initial guess and iterate until the normalized difference in successive approximations to the dominant eigenvalue is $< 10^{-6}$ or a maximum of 100 iterations. Scale the eigenvector approximations so that the $\|\vec{x}^k\|_\infty = 1$. Use the Rayleigh quotient to approximate the eigenvalue but implement this in an efficient manner. Tabulate complete results for the first 10 iterations (or less if it reaches the convergence tolerance before this) and for the final iteration. Give the numerical rate of convergence at each step and compare with the theoretical rate. Discuss results. **(45 points)**

2. Repeat (a) for matrix A_1 using an initial guess of

$$(-1 + \sqrt{0.6}, -.4(5 + \sqrt{15}), 2).$$

Explain your results and compare with those in (a). Would your result be different if you were using exact arithmetic? Why or why not? **(10 points)**

3. In the previous two questions you used the Rayleigh quotient to approximate the eigenvalue from the given approximate eigenvector. Show that the Rayleigh quotient forms the least squares best approximation to the eigenvalue. **(10 points)**

4. In this problem we want to explain why a shifted inverse power method has typically better convergence properties than just the shifted power method. Give (you don't have to prove) the theoretical rates of convergence for both of these methods. Now suppose you have a 5×5 symmetric matrix with distinct eigenvalues -1.5, 0.2, 0.5, 0.8 and 3 and we want to approximate the dominant eigenvalue which we know is around 2.5 so we can use this as a shift. Compare the convergence rates for the shifted power method and shifted inverse power method to find the dominant eigenvalue of the matrix. Use this example to explain why a shifted inverse power method is typically faster than just the shifted power method. **(20 points)**

5. Describe a method based on the shifted inverse power method in which the shift, μ_k, is updated after each iteration. Carefully explain why you do not have an unstable algorithm because it would seem that as the shift $\mu_k \to \lambda$ then the matrix $(A - \mu_k I)^{-1}$ becomes singular. **(15 points)**

Q2. **Integration and Fourier Series** (Dr. Shanbhag)

Trapezoidal rule works unexpectedly well for smooth periodic functions. Consider the integral of the even analytical function, $f(x) = e^{\cos x}$,

$$I = \int_0^{2\pi} f(x)\, dx. \tag{1}$$

1. For periodic functions like $f(x)$, show that the multipoint trapezoidal rule with N subintervals simplifies to,

$$I_N = \frac{2\pi}{N} \sum_{j=1}^{N} f(x_j), \tag{2}$$

   where $x_j = 2\pi j/N$ for $j = 0, 1, ..., N$. Note that $f(x_0) = f(x_N)$, due to periodicity. **(10 points)**

2. Use trapezoidal rule with N between 2 and 10, and plot the error $E = I_N - I$ as a function of N. You may use a built-in quadrature routine to estimate I to an accuracy of 10^{-10} or greater. Show that the convergence of the trapezoidal rule for this problem is much stronger than the expected $E \sim 1/N^2$. **(25 points)**

3. Periodic and even functions like $f(x)$ above can be represented by a simplified Fourier sum,

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos kx, \tag{3}$$

   where the coefficients a_k are given by,

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos kx\, dx. \tag{4}$$

   Note that $I = \pi a_0$. Substitute eqn 3 in eqn 2, and show that,

$$I_N - I = \frac{2\pi}{N} \sum_{k=1}^{\infty} a_k \sum_{j=1}^{N} \cos(2\pi kj/N).$$

   (20 points)

4. For equispaced $x_j = 2\pi j/N$, it can be shown that

$$\sum_{j=1}^{N} \cos kx_j = \sum_{j=1}^{N} \cos(2\pi kj/N) = \begin{cases} N & \text{if } k = mN \\ 0 & \text{otherwise.} \end{cases},$$

   where m is an integer, and $k = mN$ indicates that k is a multiple of N. Use this relationship to show that,

$$I_N - I = 2\pi \sum_{m=1}^{\infty} a_{mN}.$$

   (15 points)

5. Thus, the convergence of the trapezoidal method is intimately tied to the convergence of cosine series. With $N = 3$ and 4, plot the dependence of a_{mN} versus m for $m = 1, 2, ..., 10$, and show that coefficients decay exponentially fast (use a log-linear plot). You may use an inbuilt solver to numerically evaluate the coefficients. **(30 points)**

Q3. Approximation (Dr. Shanbhag)

A classical problem in data analysis involves fitting a sum of exponentials to a time series of uniformly sampled observations. Here, let us suppose we are given N observations (t_i, f_i), where $t_i = i\Delta t$ for $i = 0, 1, ..., N-1$. We want to fit the data to a sum of two exponentials,

$$\hat{f}(t) = a_1 e^{b_1 t} + a_2 e^{b_2 t}. \tag{5}$$

The general nonlinear regression problem to determine $\{a_j, b_j\}$ becomes difficult as the number of exponentials in the sum increases. A number of quasi-linear methods have been developed to address this. In the question, we will explore one of these methods, and determine the fitting parameters.

1. First, generate a synthetic dataset (t_i, f_i) with true $a_1^* = a_2^* = 1.0$, $b_1^* = -2.0$, $b_2^* = -0.2$ using eqn 5. Use $t_0 = 0$, $\Delta t = 1$, and $N = 20$. Attach a plot of the synthetic dataset. Use this dataset for the numerical calculations below. **(15 points)**

2. If b_1 and b_2 are known, then we can determine a_1 and a_2 by linear least squares. Set $u_1 = e^{b_1 \Delta t}$ and $u_2 = e^{b_2 \Delta t}$. Recognize that $e^{b_i t_j} = e^{b_i j \Delta t} = u_i^j$. Hence from eqn 5, we can set:

$$\begin{aligned} f_0 &= a_1 u_1^0 + a_2 u_2^0 \\ f_1 &= a_1 u_1^1 + a_2 u_2^1 \\ \vdots &= \vdots \\ f_{N-1} &= a_1 u_1^{N-1} + a_2 u_2^{N-1} \end{aligned} \tag{6}$$

   Write a program to determine a_1 and a_2, given the data, b_1 and b_2. **(15 points)**

3. Consider the polynomial $p(z)$, which has u_1 and u_2 as its roots, $p(z) = (z - u_1)(z - u_2) = z^2 - d_1 z - d_2 = 0$. Express u_1 and u_2 in terms of d_1 and d_2. **(10 points)**

4. Now we seek to take linear combinations equations in eqn 6 with the goal of eliminating a_j. For example, consider the first three equations. If we multiply the first eqn by d_2, the next by d_1, and the third by -1 and sum them up.

$$\begin{aligned} d_2 f_0 &= a_1 d_2 + a_2 d_2 \\ d_1 f_1 &= a_1 u_1 d_1 + a_2 u_2 d_1 \\ -1 f_2 &= -a_1 u_1^2 - a_2 u_2^2. \end{aligned}$$

   We get $-F_2 + d_1 F_1 + d_2 F_0 = -a_1(u_1^2 - d_1 u_1 - d_2) - a_2(u_2^2 - d_1 u_2 - d_2) = 0$, since $p(u_i) = 0$.

   We can pick the next set of three equations, and repeat the process (multiply by d_2, d_1, and -1 before summing up). Show that we end up with the following linear system: **(20 points)**

$$\begin{bmatrix} f_1 & f_0 \\ f_2 & f_1 \\ \vdots & \vdots \\ f_{N-2} & f_{N-3} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} f_2 \\ f_3 \\ \vdots \\ f_{N-1} \end{bmatrix}$$

Determine d_1 and d_2, and hence u_1 and u_2. From this, find the estimated b_1 and b_2. **(25 points)**

5. Once you know b_1 and b_2 find a_1 and a_2 by linear least squares solution of eqn. 6. **(15 points)**

Q4. **Probability and Statistics** (Dr. Shanbhag)

Consider a discrete random variable y, with probability distribution,

$$\pi(y=0) = \pi(y=1) = 1/2.$$

Now, consider a continuous random variable, x, with a conditional probability distribution,

$$\pi(x|y) = \begin{cases} \mathcal{N}(\mu=1,\sigma), & \text{if } y=0 \\ \mathcal{N}(\mu=2,\sigma), & \text{if } y=1 \end{cases},$$

where $\mathcal{N}(\mu,\sigma)$ denotes a normal distribution with mean μ and variance σ^2.

1. If $\sigma = 2$, find and plot the marginal distribution of x, which is given by,

   $$\pi(x) = \sum_y \pi(x|y)\pi(y).$$

   (40 points)

2. Use Bayes theorem to find the probability $\pi(y=1|x=1)$, again assuming $\sigma = 2$. **(30 points)**

3. Plot the posterior probability distribution $\pi(y|x)$ for $\sigma = 1$ and $\sigma = 4$. Discuss how and why $\pi(y|x)$ changes as σ increases. **(30 points)**

Q5. **Finite Difference** (Dr. Quaife)

Consider the one-dimensional PDE

$$\begin{aligned} -u''(x) &= f(x), \qquad x \in (0,1), \\ u(0) &= 1,\ u(1) = 1. \end{aligned}$$

(a) Using the standard second-order central difference formula with N interior points, show that the matrix A in the resulting linear system $A\vec{u} = h^2\vec{f}$ is

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix},$$

where $h = (N+1)^{-1}$. Be sure to discuss how the boundary conditions are being satisfied. **(15 points)**

(b) We, of course, are interested in inverting A. For $N = 8$, compute $B = A^{-1}$ (you can use any language) and write down the result. If you factor out a common factor of $(N+1)^{-1}$ from each term of the matrix, then the matrix should only contain integers. **(15 points)**

(c) Consider partitioning B from part (b) into blocks as

$$B = \frac{1}{(N+1)} \begin{bmatrix} \begin{bmatrix} B_{11}^{(2)} & B_{12}^{(2)} \\ B_{21}^{(2)} & B_{22}^{(2)} \end{bmatrix} & B_{12}^{(1)} \\ B_{21}^{(1)} & \begin{bmatrix} B_{33}^{(2)} & B_{34}^{(2)} \\ B_{43}^{(2)} & B_{44}^{(2)} \end{bmatrix} \end{bmatrix},$$

where a block superscripted with (k) is a square matrix with $N/2^k$ columns and rows. Comment on the symmetry of the off-diagonal blocks B_{ij}^k, $i \neq j$. **(15 points)**

(d) What is the rank, k, of $B_{12}^{(1)}$ and $B_{12}^{(2)}$? Use the SVD to write these two matrices as

$$B_{12}^{(1)} = U_{12}^{(1)}\Sigma_{12}^{(1)}V_{12}^{(2)} \quad \text{and} \quad B_{12}^{(2)} = U_{12}^{(2)}\Sigma_{12}^{(2)}V_{12}^{(2)},$$

where the matrices Σ are $k \times k$. **(25 points)**

(e) Based on your result from part (d), explain what structure of B you would expect to observe for larger values of N (you may assume N is a power of 2). Use a picture to describe your idea. **(20 points)**

(f) Explain how the structure in part (e) can be used to efficiently apply A^{-1} to any vector. What complexity would you expect to observe (you do not need to implement the method)? How does this compare to a naive approach of applying A^{-1}? **(10 points)**

Q6. **Optimization/Linear Programming** (Dr. Quaife)

Consider the linear cost function

$$c(\mathbf{x}) = \mathbf{f}^T\mathbf{x} = 4x_1 + 2x_2 + x_3 + 4x_4 - 2x_5$$

Use the simplex method to minimize $c(\mathbf{x})$ with the constraints

$$\begin{aligned} 2x_1 + 3x_2 - 2x_3 + x_4 + 2x_5 &= 4 \\ -2x_1 + x_2 + 2x_4 + 3x_5 &= 2 \\ x_i \geq 0, \quad i = 1, \ldots, 5&. \end{aligned}$$

Carry out all the calculations by hand, and document your writeup with the algorithmic steps of the simplex method.

Q7. **Partial Differential Equations** (Dr. Quaife)

Consider the two-dimensional PDE

$$u_{xx} + u_{xy} = f(x, y), \quad (x, y) \in (0, 1)^2, \tag{7}$$

with homogeneous Dirichlet boundary conditions. We will be considering a finite difference method with spacing $\Delta x = \Delta y = h$.

(a) Show that

$$\frac{u(x+h, y+h) + u(x-h, y-h) - u(x+h, y-h) - u(x-h, y+h)}{4h^2}.$$

approximates $u_{xy}(x, y)$ with second-order accuracy. **(25 points)**

(b) Consider discretizing the PDE (7) with the stencil from part (a) for the u_{xy} term and the standard second-order central difference for the u_{xx} term. Write down the linear system that must be solved if $N = 16$ interior points are used in the discretization. **(20 points)**

(c) Draw the undirected adjacency graph for the linear system in part (b). That is, draw a graph where each node corresponds to one of the 16 discretization points, and the nodes i and j are connected if and only if $a_{ij} \neq 0$ or $a_{ji} \neq 0$. **(10 points)**

(d) In algebraic multigrid, the grid points need to be separated into two sets—coarse grid points C and fine grid points F. One way to sort the points is to use the coloring scheme. For this problem, this amounts to performing the following steps:

(i) For each node k in the adjacency graph, count the total number of points, λ_k, connected to it.

(ii) Choose a point that has the largest value of λ_k and place it in C.

(iii) Place each point connected to node k in F.

(iv) For every unassigned point, increase its value of λ for each new fine point that it is connected to.

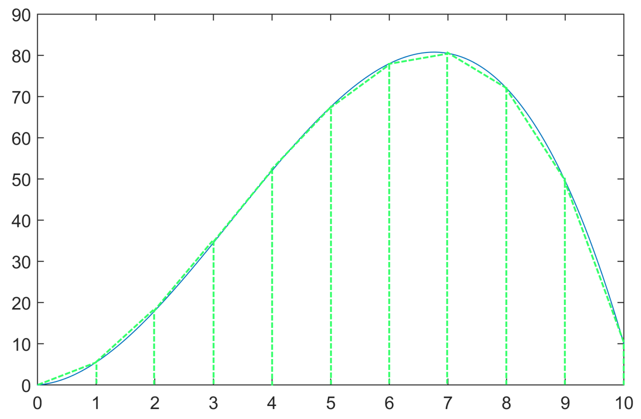
(v) Go to step (ii).

Apply this coloring scheme, by hand, to the adjacency graph from part (c). **(35 points)**

(e) Contrast your result from part (d) to the coarse grid that geometric multigrid would apply to the PDE $u_{xx} + u_{yy} = f(x, y)$. Explain why there is a difference. **(10 points)**

Q8. **Parallel Programming** (Dr. Huang)

Integral $\int_a^b f(x)dx$ can be approximately computed using the trapezoid rule, which is illustrated in the figure. We evenly divide the function into n subintervals with the nodes $\{x_0$, x_1, ..., $x_n\}$ where $x_0 = a$ and $x_n = b$. The width is $\Delta x = \frac{b-a}{n}$. The area of the trapezoidal over the interval $[x_i, x_{i+1}]$ is $A_i = \frac{\Delta x}{2}(f(x_i) + f(x_{i+1}))$. $\int_a^b f(x)dx$ is then approximated by the sum of the areas, that is, $\int_a^b f(x)dx = \sum_{i=0}^{n-1} A_i$.



Write an MPI program to integrate $f(x) = x + 5x^2 - 0.5x^3$ (shown in the picture) over the interval $[0, 10]$ using the trapezoidal rule. In your program, the interval is evenly divided to N_p subintervals. N_p is the number of processes. The process i ($i = 0, 1, ..., N_p - 1$) is in charge of the interval $[x_i, x_{i+1}]$ and computes the area A_i. Note that the process i *only* evaluates the function $f(x)$ at x_i and gets $f(x_{i+1})$ from the process $i + 1$. In such a way, the computational burden of each process can be approximately halved, if the evaluation of $f(x)$ is time consuming. This algorithm indicates that the processes send data to each other in a ring-like fashion, except for the last process which calculates both $f(x_{N_p-1})$ and $f(x_{N_p})$. Write an MPI program implementing this algorithm. Submit your code and briefly describe how to run it. I will compile and run your program. Run your program with 10 processors. Report the integral and the areas $\{A_i\}$ calculated by each process.

Q9. **Fast Fourier Transform** (Dr. Meyer-Baese)

For a prime factor FFT the following 2D DFT is used:

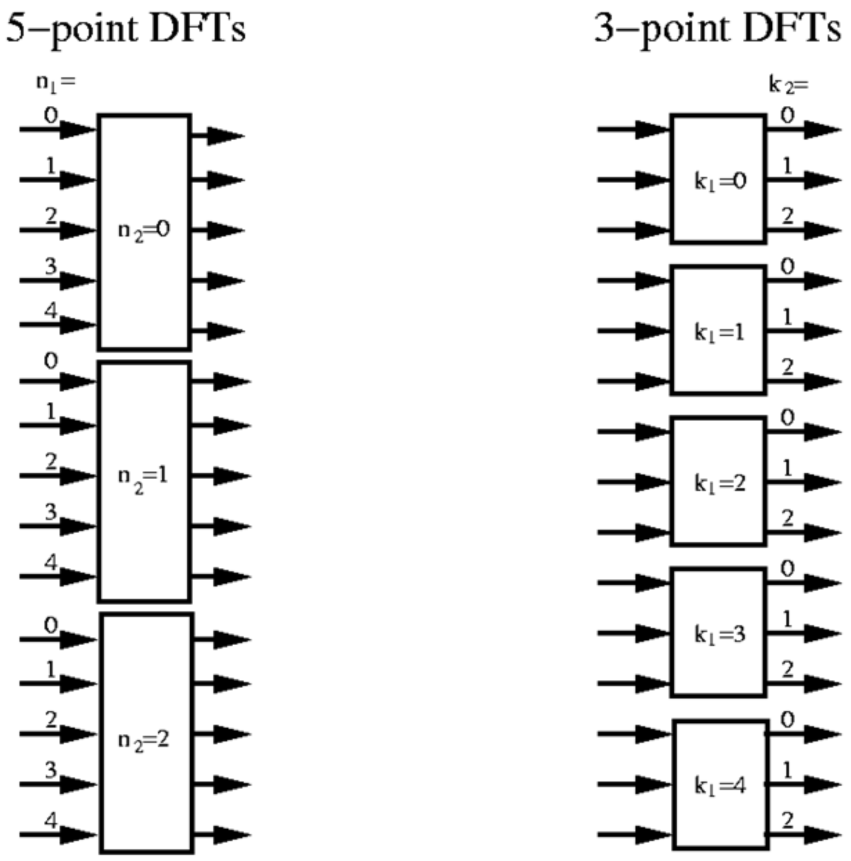
$$X[k_1,k_2] = \sum_{n_2=0}^{N_2-1} W_{N_2}^{n_2k_2} \left(\sum_{n_1=0}^{N_1-1} x[n_1,n_2] W_{N_1}^{n_1k_1} \right)$$

1. Complete the following table for the index map for a $N = 15$ with $N_1 = 5$ and $N_2 = 3$ FFT with: $n = 3n_1 + 5n_2$ mod 15 and $k = 6k_1 + 10k_2$ mod 15. **(40 points)**

n2	**n1**				
	0	**1**	**2**	**3**	**4**
0					
1					
2					

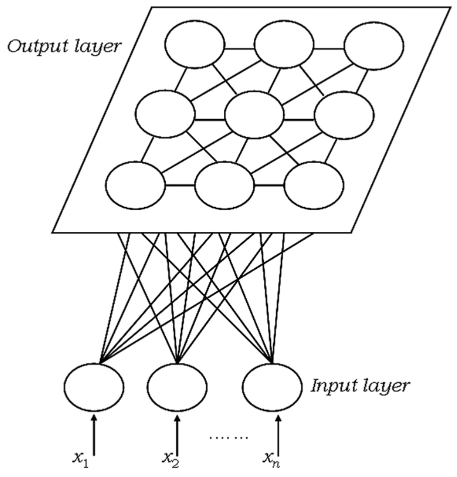
k2	**k1**				
	0	**1**	**2**	**3**	**4**
0					
1					
2					

2. Complete the SFG (for $x[n], X[k]$ and connection between first and second stage) for the FFT: **(60 points)**



Q10. **Data Structure** (Dr. Meyer-Baese)

Assume we have a self-organized map (SOM) or Kohonen map as given below with the input layer having n neurons x_i and the lattice (hidden layer) having N^2 neurons (called also receptive fields). The architecture of the SOM is given in the following Figure:



As neighborhood functions we consider with r being the radius:
(a) An inverse multiquadrics with $c > 0$, $\varphi(r) = 1/\sqrt{r^2 + c^2}$
(b) A Gaussian function with the variance σ, $\varphi(r) = \exp(-r^2/2\sigma^2)$
Please answer the following questions. Libraries cannot be used. You can use any language you like for the pseudocode (C++, Java, Python, Fortran, Matlab).

1. Assume that neighborhood function φ_i is both a Gaussian and an inverse multi-quadrics function. Express the weights changes as a function of x at the input layer as a matrix equation. Please explain all notation and reasoning. Use the given SOM network above as an example. If the execution of φ_i counts as one operation, how many operations are required to find a solution? **(30 points)**

2. Now consider a network with an input layer of size 10, with N^2 neurons of Gaussian type. How many operations are required to solve the network? Devise a notation and write down the matrix equation. Write a program to calculate the neural net. Assume that the weights are uniformly distributed between -1 and 1. Choose values of x_i and σ_i to be random numbers between 0 and 1. Numerically calculate the computer time it takes to compute the outputs for $N = 5; 10; 15$. What is the scaling of time with respect to N? **(35 points)**

3. Assume that the neighborhood function is a Gaussian function. Can we also give a matrix formulation of the SOM for the weights? In the event that the matrix formulation is not appropriate, provide an alternative approach to computing the weights? Write some pseudocode. **(20 points)**

4. Now, assume that we have a neural lattice with a very large number of neurons and with varying variance over time. Describe a data structure appropriate to the task.

Please provide pseudocode that describes that data structure using object-orientated coding. **(15 points)**

Q11. **Stability and Convergence of Numerical PDEs** (Dr. Plewa)

Consider an explicit numerical solution of a one-dimensional heat diffusion problem, `http://people.sc.fsu.edu/~jburkardt/f77_src/fd1d_heat_explicit/fd1d_heat_explicit.html`.

The above website provides implementations in C, C++, Fortran 77, and Fortran 90, including the initial and boundary conditions for a smooth test problem. You can either use one of the above codes or implement your own version of the algorithm using one of the above languages.

Compile the code and build an executable. Obtain, analyze, and verify order of convergence of the solver using the following approach. Recall that for time-dependent problems, the solution error contains spatial, temporal, and mixed (spatio-temporal) terms. For example, using a suitable solution error norm, e.g. L_1, one can write

$$L_1 = A(\Delta x)^\alpha + B(\Delta t)^\beta + C(\Delta x \Delta t)^\gamma.$$

Your task is to estimate the coefficients A, B, C, and exponents (rates of convergence) α, β, and γ for the above code and test problem implementation. To this end, you need to modify the code and perform a series of experiments methodically varying mesh resolution (in space and/or time).

To determine coefficients and powers in the above model equation use a least-squares fit to data (using MATLAB for this purpose is allowed). Make sure your spatial convergence experiments are performed when the code runs stably at all resolutions (i.e., the CFL condition is always satisfied) and the solution is obtained in an asymptotic regime (loosely speaking the solution is well-resolved and the above error model applies).

For spatial experiments, you may want to start with a coarse mesh containing perhaps only a dozen or so mesh cells, then double the mesh resolution until you obtained a few solutions in the asymptotic regime allowing for extracting A and α. For temporal experiments, choose a spatially well-resolved model and vary the time step. Your space-time study requires more creative thinking. (Hint: It has to be performed last.)

Document your experiments by showing a table with mesh resolution and corresponding error norms. Document your analysis by graphing data and model fits. Discuss the results in the context of formal order of the solver.

Grade contribution:

1. Spatial convergence study **(50 points)**
2. Temporal convergence study **(30 points)**
3. Spatio-temporal convergence study **(20 points)**