---

Department of Scientific Computing

# Written Preliminary Examination

Spring 2013

January 11–14, 2013

---

*Instructions:*

- Solve only 10 of the 12 questions as completely as you can.
- All questions are weighted equally.
- All parts of a question are weighted equally unless stated otherwise.
- If you use web sources, please list them clearly.
- The exam is due back to Maribel Amwake no later than 1:00 pm on Monday, January 14, 2012; 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. Sachin Shanbhag (sshanbhag at fsu dot edu). The person responsible is listed at the beginning of each question.
- 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**. **Optimization** (Dr. Navon)

Use the Augmented Lagrangian method to solve the equality constrained problem

$$\min f(x) = 6x_1^2 + 4x_1x_2 + 3x_2^2$$

$$s.t. \quad h(x) = x_1 + x_2 - 5 = 0$$

Construct the Augmented Lagrangian

$$L(x, \lambda, r_k) = f(x) + \lambda h(x) + r_k h^2(x)$$

where $r_k$ is the penalty parameter.
Start with $r_k = 1$ and $\lambda^{(1)} = 0$. The necessary conditions for the stationary point of $L$ yield $x_1^{(1)}$ and $x_2^{(1)}$ as a function of $r_k$ and $\lambda^{(1)}$.

For the next iteration update penalty by $r_{k+1} = 2r_k$ and the multiplier by $\lambda^{(k+1)} = \lambda^{(k)} + 2r_k h(x^k)$.

Carry out the procedure for one more iteration by substituting new values of $\lambda$ and $r_{k+1}$ and obtain values of $x_1^{(2)}$, $x_2^{(2)}$ and $h(x^{(2)})$.

---

**Q2**. **Fourier Analysis** (Dr. Meyer-Baese)

1. A periodic signal $g(t)$ is expressed by the following Fourier series:

$$g(t) = 5\sin 2t + 4\sin(4t - \pi/2) + 5\cos(7t - \pi/5) \tag{1}$$

   a) Sketch the exponential Fourier series spectra.

   b) Write the exponential Fourier series for $g(t)$.

2. From the definition $G(\omega) = \int_{-\infty}^{\infty} g(t)e^{-j\omega t}dt$ show that the Fourier transform of sinc(t-5) is rect$(\omega/2)e^{-j5\omega}$.

   Hint: rect is given as

$$\text{rect}(x) = \begin{cases} 0 & : & |x| > 0.5 \\ 0.5 & : & |x| = 0.5 \\ 1 & : & |x| < 0.5 \end{cases} \tag{2}$$

   and sinc(x) is given as sinc(x)=$\frac{\sin x}{x}$

---

**Q3**. **Ordinary Differential Equations**: (Dr. Shanbhag )

The elementary reactions:

$$A + B \rightarrow C + B$$
$$C \rightarrow D,$$

lead to the following kinetic equations which describe the variation of the concentration $c_i$ $(i = A, B, C, D)$:

$$\frac{dc_A}{dt} = -k_1 c_A c_B$$
$$\frac{dc_C}{dt} = k_1 c_A c_B - k_2 c_C$$
$$\frac{dc_D}{dt} = +k_2 c_C$$

Assume $c_A(0) = 1$, $c_B(0) = 1$, and $c_C(0) = c_D(0) = 0$. Note that $c_B$ does not change with time, and $c_D(t) = c_A(0) - c_A(t) - c_C(t)$. This means we only have to solve the first two equations. Let us set $\mathbf{c} = [c_A \quad c_C]^T$.

a) We can write the kinetic equations as:

$$\frac{d\mathbf{c}}{dt} = \mathbf{f}(t, \mathbf{c}).$$

Find the corresponding Jacobian matrix $\mathbf{J}_f(\mathbf{c})$. (30 points)

b) One method to diagnose stiffness is to consider the condition number of this Jacobian matrix. If $k_1 = 1$, and $k_2 = 0.5$, what is the condition number? What is the condition number if $k_1 = 1$, and $k_2 = 10,000$. Comment on the stiffness of the problem in the two cases. (20 points)

c) Use RK45 to solve the two cases above, and report the amount of time required to solve it. You may use a canned routine, and set the relative tolerance to $10^{-6}$. (30 points)

d) Comment on the following statement: "One simply cannot use forward Euler to solve a stiff initial value problem to a specified level of accuracy." (20 points)

---

**Q4**. **Partial Differential Equations** (Dr. Ye)

Consider the heat equation

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = \frac{\partial h}{\partial t}, \quad 0 < x, y < 1; t > 0$$
$$h(x, y, 0) = 0$$

with the boundary conditions

$$h(1, y, t) = h(x, 1, t) = 1$$
$$\frac{\partial h}{\partial x}(0, y, t) = \frac{\partial h}{\partial y}(x, 0, t) = 0.$$

The initial condition indicates that $h$ is zero everywhere at the beginning ($t = 0$), even at the Neumann boundaries (left and bottom boundaries). At the Dirichlet boundaries (right and top boundaries), $h = 1$ for all the time including the initial time, $t = 0$.

The exact solution is given by

$$h(x, y, t) = 1 - \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} C_{nm} \cos\left[\frac{1}{2}(2n-1)\pi x\right] \cos\left[\frac{1}{2}(2m-1)\pi y\right] \exp\left[-\frac{\pi^2 t}{4}\left((2m-1)(2n-1)\right)\right]$$

where

$$C_{nm} = \frac{16(-1)^{n+1}(-1)^{m+1}}{\pi^2(2m-1)(2n-1)}$$

and $m$ and $n$ values are always taken large enough to avoid oscillation.

This is a transient problem, and the solution, $h(x, y, t)$, changes with time. We wish to solve this problem by finite differences and compare our results with the known analytical solution of steady state. Let us choose a finite difference grid with spatial intervals $\Delta x = \Delta y = 0.1$.

(a) Write a code (MATLAB, FORTRAN, or any language of your preference) to evaluate the analytical solution. Explain how you select the maximum values of the coefficients $n$ and $m$. (5 points)

**The questions below are for the explicit finite difference method.**

(b) Write a code capable of solving this problem by the explicit finite difference method. (10 points)

(c) The stability condition for the this two-dimensional problem is,

$$\frac{\Delta t}{(\Delta x)^2} + \frac{\Delta t}{(\Delta y)^2} \leq \frac{1}{2}$$

Compute the stability limit of $\Delta t$ for this problem.

(d) Run your program up to $t = 1$ twice, once by using a equal to the stability limit, and once by using a $\Delta t$ which exceeds the former by 0.0002.

(e) Plot $h$ versus $t$ from both solutions at point (0.9, 0.1), and $h$ versus $x$ at $y = 0.1, t = 0.225$ (or close to this t value). What do you see, and why? (10 points)

(f) Evaluate the analytical solution at point (0.9, 0.1) as a function of time and compare with the numerical solutions. Plot the difference between the analytical and numerical solution in each case as a function of time. Explain your results. (10 points)

**The questions below are for the implicit finite difference method.**

(g) Write a code capable of solving the problem by the implicit finite difference method. (10 points)

(h) Run your program up to $t = 1$ twice, once by using a equal to the stability limit, and once by using a equal to 0.05.

(i) Plot $h$ versus $t$ from both solutions at point (0.9, 0.1), and $h$ versus $x$ at $y = 0.1, t = 0.225$ (or close to this t value). What do you see, and why? (10 points)

(j) Evaluate the analytical solution at point (0.9, 0.1) as a function of time and compare with the numerical solutions. Plot the difference between the analytical and numerical solution in each case as a function of time. Explain your results. (10 points)

(k) Compare the results of (f) and (j). If there are differences, explain them. (5 points)

---

**Q5**. **Linear Algebra** (Dr. Peterson)

Suppose that $A$ is an $n \times n$ symmetric matrix and we want to perform the decomposition

$$A = LDL^T$$

where $L$ is an $n \times n$ *unit* lower triangular and $D$ is an $n \times n$ diagonal matrix.

a) Derive the equations for the entries of $L$ and $D$ in terms of the entries $A_{ij}$ of $A$ and any previously computed entries of $L$ or $D$. (40%)

b) Write an algorithm using pseudo-code for obtaining this decomposition. (20%)

c) Determine the number of additions/subtractions and multiplications/divisions required for this decomposition. (25%)

d) Are there any advantages or disadvantages to this decomposition over the standard Cholesky decomposition? (15%)

---

**Q6**. **Linear Algebra** (Dr. Burkardt)

All questions in this section concern *strictly* upper triangular matrices, or else upper triangular matrices.

a) Suppose that the real $n$ by $n$ matrix $A$ is *strictly* upper triangular, that is, the lower triangle and the diagonal are zero. Show that it is impossible for $A$ to have a nonzero eigenvalue $\lambda$.

b) Let $B$ be a *strictly* upper triangular matrix. From part a), we know that $B$ can't have any nonzero eigenvalues. We know that $B$ must have at least one eigenvalue. Therefore...it is zero. We know that every matrix must have at least one (nonzero!) eigenvector corresponding to each eigenvalue. What is one eigenvector of $B$?

c) Suppose that $C$ is the following 4x4 upper triangular matrix:

$$C = \begin{pmatrix} 4 & 1 & 1 & 1 \\ 0 & 3 & 1 & 1 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The eigenvalues of $C$ are stored in a diagonal matrix $\Lambda$, and the corresponding eigenvectors are the columns of the matrix $U$:

$$\Lambda = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 1 & -t & 0 & 0 \\ 0 & t & -t & 0 \\ 0 & 0 & t & -t \\ 0 & 0 & 0 & t \end{pmatrix}$$

where $t = \sqrt{2}/2$.

What equation relates $C$, $\Lambda$ and $U$? Suppose we wish to compute the value of $C^{100}$. Instead of carrying out 99 matrix multiplications, how could you could use the factors $\Lambda$ and $U$ to compute the result at a much lower cost?

d) Let $D$ be the 4x4 upper triangular matrix of 1's, and suppose we want to compute $D^{100}$. We cannot do this using the eigenvalue idea, because $D$ is a defective matrix. However, there is another possibility. Compute the powers $D^1, D^2, D^3, D^4, D^5$. What is the relationship between each of these matrices and Pascal's triangle? There is an exact formula for any element of Pascal's triangle, given its row and column. In what row and column of Pascal's triangle will you find the value of the element in the first row, last column of the matrix $D^{100}$?

e) Suppose that $E$ is an $n$ by $n$ upper triangular matrix of 1's. Describe an efficient procedure for solving a linear system $E * x = b$ for the unknown vector $x$, given $b$.

**Q7**. **Parallel Programming - MPI** (Dr. Burkardt)

A hole has been drilled into the earth to an underground reservoir of oil. A pipeline has been created, consisting of 100 consecutive stations, with station 1 down there in the reservoir, and station 100 up at the surface. There are 25 million barrels of oil in the reservoir, and it is desired to bring at least 24 million barrels of oil to the surface.

The oil is moved by pumps from one station to the next higher one. Each pump has the property that, over the course of a day, it can move 15% of the oil at that station to the next higher station. There is a pump at every station except the last, where the oil is collected in a gigantic tub.

We measure oil in millions of barrels, so $U(1)$ is initially 25, and we want to pump until $U(100)$ is at least 24. A simplified model allows us to start from a list $U()$ of the amount of oil at each station at the beginning of a day, and determine $V()$, the amount at the end of the day:

$$\begin{aligned} V(1) = & \quad 0.85 * U(1) \\ V(2) = & \quad 0.15 * U(1) + 0.85 * U(2) \\ & \ldots \\ V(99) = & \quad 0.15 * U(98) + 0.85 * U(99) \\ V(100) = & \quad 0.15 * U(99) + 1.00 * U(100) \end{aligned}$$

(Note that U(100) has a coefficient of 1 in the last equation!)

a) Write a program that will:

(i) initialize the first entry of U to be 25, the remaining entries zero;

(ii) compute V, the updated value of U for the end of the day;

(iii) if V(100) is at least 24, stop, printing the number of days;

(iv) Otherwise, set U = V and return to step 2.

Your code may be in pseudocode, or a standard computer language, and you do not need to run it.

b) Rewrite your program from part a) to use MPI. Assume that you only have two MPI processes available. Write your program in such a way that you can hope it will run about twice as fast as your original program. Your program should also measure and report the program execution time. Again, you do not need to run the program.

You are free to present your program as a complete or partial program in a standard language, or in pseudocode. However, your answer must specify all important MPI issues, including initialization and shutdown. For instance, instead of a full call to **MPI_Send()**, it would be clear enough to say *"Use MPI_Send() to send the value of* **i** *to process* **0***"*.

---

**Q8**. **Statistics**: (Dr. Slice)

Consider the standard normal, Student's $t$, $\chi^2$, and the Fisher-Snedecor distributions.

a) Describe these distributions.

b) Discuss their use.

c) Discuss the relationship amongst them.

d) Where does the binomial distribution fit in?

e) Why are we so obsessed with the standard normal?

f) Outline a program to statistically test whether or not two samples are drawn from populations having the same mean *without* reference to any of the above distributions.

---

---

**Q9**. **Approximation**: (Dr. Wang)

Suppose we have 4 discrete points on the $x-y$ plane. $(0.1, 1.1)$, $(0.9, 2.0)$, $(3.1, 3.8)$, $(5.3, 6.5)$.

a) Use least square (error of $y$) approximation to find a straight line $y = ax + b$ best approximate those points.

b) Find another straight line $Ax + By = C$ best approximate those points in the sense of least square distance from those points to the line.

---

---

**Q10**. **Numerical Integration**: (Dr. Beerli)

Integrate the function

$$f(x,y) = \exp\left(\frac{-x^2}{2}\right)\exp\left(\frac{-y^2}{2}\right) + \exp\left(\frac{-(x-3)^2}{10}\right)\exp\left(\frac{-(y-3)^2}{10}\right)$$

over $[-5,5]$ for $x$ and $y$.

1. Give results for the Simpson's rule for $2^n$ points for $n$ in the range of 4 to 10.
2. Report the error at each $n$. You may assume that the most accurate result above is the exact value of the integral.
3. Write a Monte Carlo program that computes the result. Compare the accuracy of this result with the earlier results.

---

**Q11**. **Molecular Dynamics** (Dr. Shanbhag)

Consider a 1D particle initially at $r = 0$ moving with a constant velocity of $v = +1$ units in a periodic lattice of "gates" separated by a distance of $L = 1$ unit, as shown in figure below. When the particle "arrives" at a gate, it is reflected back instantaneously (the direction of $v$ is reversed) with a probability of 0.5. If it is not reflected, it continues through the gate until it encounters another gate, where the same scenario unfolds again. The characteristic time between arrivals at gates is therefore $\tau = L/v = 1$ unit.


At timescales much smaller than $\tau$, the particle moves with uniform velocity. At much larger timescales, the particle hops between gates, and its motion resembles simple 1D diffusion. Let us assume that we take snapshots of the particle at fixed intervals $\Delta t = 0.1\tau$ for $\Delta t \leq t \leq N\Delta t$, with $N = 100,000$.

We wish to compute the mean-squared displacement of the particle $\rho(t)$ according to equation:

$$\rho(t = n\Delta t) = \frac{1}{N-n} \sum_{i=1}^{N-n} \left(r((i+n)\Delta t) - r(i\Delta t)\right)^2, \quad 1 \leq n \leq N-1.$$

Write a program to compute $\rho(t)$ and plot it.

---

**Q12**. **PDEs**: (Dr. Plewa)

Consider an explicit numerical solution for a one-dimensional heat equation:

$$\frac{du}{dt} - k\frac{d^2u}{dx^2} = f(x,t)$$

with boundary conditions $u(a,t) = u_a(t)$ and $u(b,t) = u_b(t)$, and initial condition $u(t_0) = u_0(x)$.

A Fortran implementation of finite difference solver for this equation is available at John Burkardt's website[1]. Links to C, C++ and Matlab implementations are also provided there. Use the initial and boundary conditions for the sample test problem provided[2].

You can either use the above code or implement the algorithm using programming language of your choice. Compile the code and build an executable. Obtain, analyze, and discuss the results for the following:

a) Code order verification: Using a self-convergence method estimate order of convergence of the above code in $L_1$, $L_2$, and $L_\infty$ norms. Present the results in a form of a table showing mesh resolution, error norms, and the corresponding code order of convergence for all norms.

   Tomek: should we say what the self-convergence method is?

b) Use $L_1$ norm, and demonstrate the effect of accumulation of round-off errors by conducting a series of test runs with progressively smaller time step for a fixed mesh resolution.

c) How does the maximum stable time step depend on the mesh resolution for an explicit time integration scheme for the diffusion equation? Verify that theoretical result for the current algorithm implementation through a series of numerical experiments.

---

[1] `http://people.sc.fsu.edu/~jburkardt/f_src/fd1d_heat_explicit/fd1d_heat_explicit.f90`
[2] `http://people.sc.fsu.edu/~jburkardt/f_src/fd1d_heat_explicit/fd1d_heat_explicit_prb.f90`