Department of Scientific Computing

Written Preliminary Examination

Spring 2014

January 10–13, 2014

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 you use web sources, please list them clearly.
- The exam is due back to Cecelia Farmer no later than 1:00 pm on Monday, January 13, 2014; 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 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)

Consider the Quasi-Newton method BFGS. Let

$$s_k = x_{k+1} - x_k$$

and

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$B_{k+1} = B_k - \frac{(B_k s_k)(B_k s_k)^T}{{s_k}^T B_k s_k} + \frac{y_k {y_k}^T}{{y_k}^T s_k}$$

minimize

$$f(x_1, x_2) = x_1 - x_2 + 2{x_1}^2 + 2x_1 x_2 + {x_2}^2$$

from the starting point $\underline{x_0} = (0, 0)^T$.
Use

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Carry out 3 iterations. Use exact line search formula.

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

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

$$g(t) = 14\sin 8t + 6\sin(7t - \pi/2) + 5\cos(10t - \pi/10) \tag{1}$$

a.) Sketch the exponential Fourier series spectra.

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

II. From the definition $g(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} G(\omega)e^{j\omega t}d\omega$ show that the inverse Fourier transform of rect$[\frac{(\omega-10)}{2\pi}]$ is sinc$(\pi t)e^{j10t}$.

Hint: rect is given as

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

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

Q3. **Integration and Approximation**: (Dr. Shanbhag)

The following integral arises in calculating the mean size of certain polymers:

$$I(f) = f \int_0^\infty \Big[\exp(-t)\ \mathrm{erf}(\sqrt{t})\Big]^{f-1} dt,$$

where $f \geq 1$ is an integer.

1. List two numerical quadrature methods which may be used to evaluate this integral. Explain their advantages and disadvantages. [10 points]

2. Choose *any one* of the methods above, and evaluate the integral between $10^0 \leq f \leq 10^4$ at sufficiently many points. Plot $I(f)$ versus $\log(f)$ in this interval. [55 points]

3. We wish to summarize the $I(f)$ data obtained above using a suitable functional form. Use any reasonable (global) function to approximate the data. Plot the approximation and the data together. [35 points]

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

The higher-order ODE below,

$$\begin{aligned} \frac{d^2h}{dx^2} &= \frac{C}{T}(h - h_{wt}) \\ h_{wt} &= 100 + 0.06x - 0.00003x^2 \end{aligned}$$

is defined for a one-dimensional domain from $x = 0$m to $x = 1000$m. The coefficients $T = 2.5 \times 10^{-5}\text{m}^2\text{s}^{-1}$ and $C = 10^{-9}\text{s}^{-1}$ are constant over the domain. The boundary condition at the left boundary is $h(x = 0) = 100$m.

Answer the questions below for different boundary conditions:

Initial-value problem (25%)
Given the boundary condition at the left boundary, $q_w = -T\frac{dh}{dx}|_{x=0} = 1.0 \times 10^{-7}\text{m}^2\text{s}^{-1}$, transform the initial-value problem of second-order ODE into a set of first-order ODEs. Solve the first-order ODEs numerically (e.g., using MATLAB) and plot h with x.

Boundary-value problem (25%)
Ignore the boundary condition of q_w above, and use the boundary condition, $h(x = 1000\text{m}) = 93$m, at the right end of the domain. Solve the second-order ODE of this boundary condition using the shooting method. Plot h with x for the first and second guess and the final solution.

Finite difference (50%)
The boundary condition at the right end of the aquifer ($x = 1000$m) now becomes $\frac{dh}{dx}|_{x=1000} = 0$. Solve the second-order ODE using the finite difference method and plot h with x.

Q5. **Statistics**: (Dr. Ye)

Consider a nonlinear synthetic model $y = x/a + \sin(amx) + \epsilon$, where its parameters are $a = 2$ and $m = 0.1$. Use this model, we generate twenty samples of y for $x = \{1, 2, , 20\}$ and subsequently corrupt them using one realization of white noise ϵ with mean zero and constant variance $\sigma^2 = 1$. The data of x and y are given below:

$x = \{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0,$
$19.0, 20.0\}$;

$y = \{-0.3921, 2.6129, 2.4246, 2.4561, 3.7999, 3.2913, 6.2434, 5.2106, 5.6054, 3.8721, 6.5784,$
$6.8246, 8.1501, 8.5362, 7.7012, 8.6872, 8.5424, 10.0760, 8.7685, 9.1889\}$;

Answer the following questions:

a Use the data above to define the likelihood function of the two model parameters. (10%)

b Maximize the likelihood function to obtain the maximum likelihood (ML) parameter estimates. (20%)

c Use the following prior distributions: $a \sim N(2, 1)$ and $m \sim N(0.1, 1)$ to estimate the maximum a posterior (MAP) parameter estimates. (30%)

d Use the Bayesian theorem to estimate the distributions of a and m. You may need numerical integration to answer this question. (40%)

Q6. **Parallel Programming - MPI** (Dr. Burkardt)

A character array **s** contains the text of a book, and consists of exactly 10,000,000 characters. The only chararacters are lowercase alphabetic characters ('a' through 'z') and blanks. We may regard the text as a sequence of words of various lengths, where words are strings of nonblanks separated by one or more blanks.

a Describe an algorithm to count the number of words in **s**;

b We wish to use parallel processing to count the number of words in s. We are going to try to do parallel processing without using MPI. Suppose we have 10 separate computers, we cut the text into 10 parts, we run the same program on each computer, and we take the partial sums from each computer and add them up by hand. Even though the computation might be faster, the result will probably be slightly incorrect. Why is the sum of the partial word counts probably not the exact number of words in the text?

c Now we want to use MPI to carry out the task. Again, we plan to use 10 processes, and each process will only have access to 1/10 of the data. The fact that the MPI processes can communicate is very important, and means that we can avoid the error noticed in the previous section. Focus on a single MPI process with identifier ID, and explain what communication is necessary in order for this process to verify or adjust the partial word count it performs.

d Once each process has done its work, the partial counts need to be combined into a single result. Give a statement in C, C++, or Fortran, including all the arguments, that implements this function call. Indicate which of the arguments to the function represent the actual variables containing the partial and final count information.

Q7. **Finite Difference Methods** (Dr. Peterson)

Let $u = u(x, y, t)$ and consider the Dirichlet initial boundary value problem (IBVP)

$$u_t - \Delta u + g(u) = f(x, y, t) \quad (x, y) \in \Omega, \quad 0 < t \leq T$$

where $\Omega = (0, 1) \times (0, 1)$ and we have the initial condition $u(x, y, 0) = u_0(x, y)$. Let $u = 0$ on three sides of Ω and let $u = \sin \pi y$ on the left boundary where $x = 0$. Here Δu denotes the Laplacian of u, i.e., $u_{xx} + u_{yy}$, and $g(u)$ is a given function of the unknown u. To discretize this problem with finite differences, first overlay the domain Ω with a uniform Cartesian grid with $\Delta x = \Delta y = h$ where we have $N + 2$ points on a side of Ω, i.e., use $x_0, x_1, \ldots, x_{N+1}$ and $y_0, y_1, \ldots, y_{N+1}$. Let $U_{ij}^n \approx u(x_i, y_j, t^n)$.

a. Let $g(u) = u$. Write down an *implicit* finite difference approximation at the point (x_i, y_j, t^n) which is first order in time and second order in space and only involves two time levels, t^n and t^{n+1}. You do not have to derive the truncation error for your difference quotients. Give the resulting matrix equation where you specifically give the entries of the coefficient matrix and the right hand side vector.

b. Repeat (a) where your implicit method is second order in time and space and only involves two time levels, t^n and t^{n+1}.

c. Assuming a uniform time step, write down pseudo-code for an efficient algorithm to solve our IBVP with $g(u) = u$ using the method in (a) and a direct solver for the resulting linear system. Indicate which algorithm for the solver you recommend using and be specific about where the matrix is set up and solved in your pseudo-code.

d. Now let $g(u) = e^{2u}$ so that the resulting difference equations are now nonlinear. Use your method from (a) to write down the difference equation at (x_i, y_j, t^n). Now discuss how you would solve these equations using Newton's Method. Specifically give the linear system that must be solved at each iteration of Newton's Method. Also indicate what stopping criteria you would use for convergence of Newton's Method. Then indicate how your algorithm in (c) would have to be modified to solve this nonlinear problem.

Q8. **Approximation**: (Dr. Wang)

Suppose we have a set of discrete points in an $x-y-z$ space: $(1.1, -0.8, 1.1)$, $(1.4, -0.7, 1.6)$, $(2.2, 0.0, 1.6)$, $(2.5, 0.3, 2.0)$.

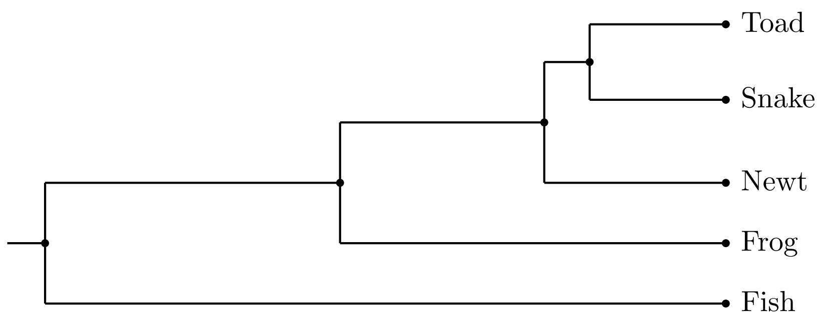
a) Find a straight line that best approximates these points and passes through the mean of those points. (60%)

b) Find the plan $Ax + By + Cz = 1$ that perpendicular to the straight line and passes through the mean of those points. (20%)

c) Note that this plan divides those points into two groups. Please list the points for each group. And now we have a test point $(1.7, 0.5, 1.4)$, please tell which group it belongs to. (20%)

Q9. **Data structures**: (Dr. Beerli)

In phylogenetic research a key interest is finding the best relationship among species (finding the best tree). This question does not have a single unique answer; describe all parts in great detail.
For example, consider this tree of the example taxa (it is not the best tree). The edge lengths are often measured in expected mutations, but sometimes they are arbitrary and only the topology is important. For this question focus on topology, dependent on the optimality criteria you invent you may consider branch lengths or not.



- Define a data structure using linked lists that can easily be oriented so that we can say that the leaves are today and the last bifurcation is in the past. When you research this topic, you will find discussions about rooted and unrooted trees. I am looking for data structures for rooted trees, like the one shown, the root is attached at the leftmost bifurcation – think of the root as the origin! (detailed [!] pseudo code is sufficient).

- Define a data structure that expresses the tree in a matrix. (detailed [!] pseudo code is sufficient).

- Give at least one algorithm to change from one tree to another for the linked list data structure AND for the matrix method. (detailed [!] pseudo code is sufficient.)

- Describe a potential optimality criterion you could use to get the tree that is most probable with the data and sketch an algorithm using either of the two data structure that can deliver this most probable (best) tree. (detailed description or pseudo code)

Example dataset

(the 0 and 1 represent presence/absence data, total 6 characters where measured)

Fish	110110
Snake	110000
Newt	100110
Frog	001001
Toad	001110

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

Calculate this integral

$$A = \int_{x=-1}^{x=1} \int_{y=-1}^{y=1} f(x) f(y) dx dy \tag{3}$$

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-(z^2)} \tag{4}$$

Evaluate this integral numerically using

1. Gaussian quadrature
2. Clenshaw-Curtis quadrature,
3. Romberg's method,
4. Monte Carlo method.

Give detail how you implement each method, including intermediate results, and compare the error behavior in relationship to approximate number of operations.

Q11. **PDEs**: (Dr. Plewa)

Consider an explicit numerical solution of a one-dimensional advection problem,
http://people.sc.fsu.edu/~jburkardt/f_src/fd1d_advection_lax/fd1d_advection_lax.html
The above website provides implementations of the Lax method in C, C++., and Fortran 77/90, including the initial and boundary conditions for a smooth test problem.
You can either use one of the above codes or implement the Lax algorithm yourself using one of the above languages.
Compile the code and build an executable. Obtain, analyze, and discuss the results for the following:

1. Code order verification: Estimate the rate of convergence of the code using a self-convergence method, see, e.g., Equation (11) and its discussion in chapter "Extraction of Convergence Rate from Grid-Convergence Tests" starting on page 140 in the review article by Roache,

   http://www.stanford.edu/group/uq/docs/roache.pdf

   and slides 24 and 25 in

   http://www.icis.anl.gov/programs/file.php?id=399&obj=MultiFile&field=filename &attachment=yes

   To this end, you need to perform a series of experiments progressively increasing mesh resolution. Start with a coarse mesh containing only 8 cells, then double the mesh resolution until there are 256 mesh cells. For each resolution, compute L_1, L_2, and L_∞ norms. Based on those results, compute the rate of convergence of your code. Present the results in a form of a table showing mesh resolution, error norms, and the corresponding code order of convergence for each norm, i.e.

nx	L_1	order	L_2	order	L_∞	order
8						
16						
32						
...						
256						

   This part is worth 50%.

2. How does the maximum stable time step depend on the mesh resolution for an explicit time integration scheme for the advection equation? Show how the solution to the above problem changes when this limit is violated using a mesh with 128 cells. (20%)

3. 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 steps on a mesh with 128 cells. (30%)