Department of Scientific Computing
Written Preliminary Examination
Spring 2009

Instructions:

Do 10 of the 11 questions as completely as you can. If you do all 11, then we will only count the first ten.

All questions are weighted equally.

All parts to each question are weighted equally unless indicated in which case the percentage is listed in parentheses.

If you use any sources on the web, please list them.

The exam is due back to Cecilia Farmer or Professor Peterson **no later than 1 pm on Monday, January 12, 2009**; no exceptions allowed.

Put your Student ID on each page of your exam. When you turn in your exam, put a cover page with your name and Student ID; staple your exam to this cover page and do not put your name on each page. Doing this will allow the initial grading to be blind.

---

1. *Datastructures*

Consider the construction of an adaptive triangular grid in $\mathbb{R}^2$. First start with a standard datastructure for an unstructured triangular mesh: one array `tri(nTri,3)` that contains the vertex indexes for each of the `nTri` triangles, and one array `node(nVert,2)` that contains the list of `nVert` vertices.

We wish to refine some of these triangles, and then again, some of the refined triangles. Each triangle, if refined, is subdivided as follows: add a point at the midpoint of each edge, and join them. Thus, each triangle is subdivided into four triangles. Imagine the following algorithm where `mx` is given:

1. Original `nTri` triangles correspond to level 0
2. set `lev=0`
3. set `maxNbLevels= mx`
4. Loop through original `nTri` triangles, and run a procedure that determines whether each triangle needs to be refined. Set a flag (true/false) for each triangle.
5. For each triangle of level `lev` with flag set to `true`, refine the triangle. (Each refined triangle is replaced by 4 triangles)
6. set `lev = lev + 1`
7. exit the algorithm if `lev == maxNbLevels`
8. return to step 4

Use the given information to answer the following questions.

(a.) Come up with a datastructure that implements this algorithm and explain how you would use it. This datastructure should be constructed out of (arrays, sets, linked lists, hash tables, quadtrees).

(b.) Given some arbitrary point $P = (x, y)$, how many work units are required to find the smallest triangle that contains P, along with the level at which this triangle is located? Express your answer in $O()$ notation as a function of `nTri` and `maxNbLevels`.

(c.) Write pseudocode that implements your algorithm. Pseudocode need not be implemented. If you do not know what pseudocode is, please look it up online!

(d.) With your datastructure, assume I wish to remove a particular triangle at the 2nd level `lev=2` (assume that `maxNbLevels > 2`) and all subsequent triangles it contains at refinements. How is this done? What is the cost?

2. *Integration*

Suppose we want to calculate an approximation to the integral

$$I = \int_0^3 \frac{e^{-x}}{2 + \sin x}\, dx$$

which does not have an analytical solution.

(a.) What is the smallest number of intervals that is needed to implement the composite trapezoidal approximation for I so that the result is accurate to 10 decimal places?

(b.) What is the lowest order Gauss-Legendre quadrature approximation for I that is accurate to 10 decimal digits?

(c.) Replace the upper integration limit by ∞. How would you solve this integral to get 8 places of accuracy? Explain the procedure, perhaps use formulas, but there is no need to write a program.

3. *Monte-Carlo*

We wish to write a function `float density()` (Java notation) that returns a random number that follows the (probability density function) distribution

$$p(x) = \alpha e^{-x}$$

for $x \in [0, 4]$.

(a.) (20%) Compute α. You may use a symbolic calculator or a pencil. The result must be exact. (hint: The probability that x is in the range $[0, 4]$ is unity.)

(b.) (30%) Write a function that returns a random variable x with the distribution $p(x)$. Demonstrate this function by executing it N times. Store the results in an array. Write a function that subdivides the interval [0,4] into 20 bins, and counts how many of 10,000 values falls into each bin. Print out a table with these values:

```
  1     value_1
  2     value_2
...
  20     value_n
```

Run the experiment for $N = 100$ and $N = 10,000$.

(c.) (30%) How would you use this function to compute the integral

$$I = \int_0^4 \frac{e^{-x}}{1+x}\, dx$$

Explain the procedure with words and pseudocode.

(d.) (20 %) How would you generalize the above functions to generate a pair of floats that satisfy the distribution $p(x, y) = e^{-x-y^2}$? Pseudocode is sufficient.

4. *Numerical Linear Algebra*

Let A be an $n \times n$ real nonsingular matrix and let $\vec{x}$, $\vec{b}$ be real n-vectors; consider the linear system $A\vec{x} = \vec{b}$. Suppose that we have an approximation $\vec{x}^0$ to the exact solution $\vec{x}$ and an approximation Q to A^{-1}. We want to consider an iterative method for improving our approximate solution $\vec{x}^0$. Define the error at the kth step to be $\vec{e}^k = \vec{x} - \vec{x}^k$. Consider the following algorithm for solving $A\vec{x} = \vec{b}$ where we are given $\vec{x}^0$ and Q.

for $k = 0, 1, 2, \ldots$

calculate the residual $\vec{r}^k = \vec{b} - A\vec{x}^k$

determine the new approximation from the expression $\vec{x}^{k+1} - \vec{x}^k = Q\vec{r}^k$

(a.) Show that the residual $\vec{r}^k$ satisfies

$$\|\vec{r}^k\| \leq \|A\|\|\vec{e}^k\| \,.$$

In this problem $\|A\|$ represents some matrix norm with an associated vector norm.

(b.) Show that

$$(\dagger)\,. \qquad \|\vec{e}^{k+1}\| \leq \|I - QA\|\|\vec{e}^k\|$$

(c.) Starting with ($\dagger$), determine a condition on $I - QA$ which guarantees that the method converges. Is there a restriction on the initial guess for convergence? Why or why not? If the method converges what is the rate of convergence? Why?

(d.) Discuss the storage required to efficiently implement the method. Discuss the work required for each iteration of the scheme. Assume A and Q are full $n \times n$ matrices. Be specific and give operation counts.

(e.) Repeat (d) if A is a symmetric tridiagonal matrix.

5. *Numerical Linear Algebra*

Let A be a real, $m \times n$ matrix with $m > n$ and let $\vec{b}$ be a real m-vector. Then the linear least squares problem is to determine $\vec{x} \in \mathbb{R}^n$ that minimizes the function

$$\rho^2(\vec{x}) = \|\vec{b} - A\vec{x}\|_2^2 \,,$$

where $\|\cdot\|_2$ denotes the standard Euclidean norm. Since A is not square, we can't define its inverse but we can define a *pseudo-inverse* $A^\dagger$ when the linear least squares problem has a unique solution; this is given by

$$A^\dagger = (A^T A)^{-1} A^T \,,$$

where A^T denotes the transpose of A. Similar to the case when A is square and invertible, we can define a condition number $\kappa(A) = \|A\|_2 \|A^\dagger\|_2$ where $\|\cdot\|_2$ denotes the matrix norm induced by the standard Euclidean vector norm.

(a.) Does the linear least squares problem always have a unique solution? If not, give a condition that guarantees uniqueness.

(b.) When solving a square linear system one can derive perturbation results; e.g., if the data is changed by a small amount we can bound the change in the solution. We can obtain similar results for the linear least squares problem when it has a unique solution. In particular consider a perturbation $\vec{c}$ of the right-hand side $\vec{b}$ and write each vector as the sum of its projection onto $\mathcal{R}(A)$ and the orthogonal complement of $\mathcal{R}(A)$ where $\mathcal{R}(A)$ denotes the range of A. Denote $\vec{b}_{\mathcal{R}(A)}$ as the projection of $\vec{b}$ onto the range of A; use similar notation for $\vec{c}$. Then we have

$$\frac{\|A^\dagger \vec{b} - A^\dagger \vec{c}\|_2}{\|A^\dagger \vec{b}\|_2} \le \kappa(A) \frac{\|\vec{b}_{\mathcal{R}(A)} - \vec{c}_{\mathcal{R}(A)}\|_2}{\|\vec{b}_{\mathcal{R}(A)}\|_2}$$

You do NOT have to prove this result. Describe in words what this result says. Give the corresponding result for solving a square invertible linear system. Compare and contrast the results.

(c.) Consider the linear least squares problem where

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \qquad \vec{b} = \begin{pmatrix} 0.01 \\ 0.00 \\ 1.00 \end{pmatrix}$$

and we have a perturbed right-hand side

$$\hat{b} = \begin{pmatrix} 0.0101 \\ 0.0000 \\ 1.0000 \end{pmatrix} .$$

Compute the quantities in the perturbation estimate in (b) for these choices of A, $\vec{b}$ and $\hat{b}$ and discuss the results.

(d.) Compute an upper bound for $\kappa(A)$ as a function of ϵ if

$$A = \begin{pmatrix} 1 & 1+\epsilon \\ 1 & 1-\epsilon \\ 1 & 1 \end{pmatrix} .$$

6. *Parallel computations*

Consider the following FORTRAN90 program:

```
program main
  integer i
  real dx, sum
  sum = 0.0
  x = 0.0
  dx = 3.14159265 / 10000.0
  do i = 1, 10001
    sum = sum + sin ( x )
    x = x + dx
  end do
  print *, 'The integral is ', sum
  stop
end
```

(a.) In many cases, a sequential program can be converted to a parallel OpenMP program by inserting some directives, without having to change any lines of the code itself. This program cannot be transformed in that way. Explain why the `DO` loop, as it is written, is not parallelizable.

(b.) Modify the code, adding variables if necessary, so that the computation is correct and the `DO` loop can be easily parallelized. You **do not** need to show any OpenMP directives; what's important is to modify the FORTRAN code itself. If you prefer, you may write your answer in C, C++ or pseudocode.

7. *Numerical PDEs*

Given a constant $c > 0$ and functions $f(x,t)$, $u_0(x)$, $u_1(x)$, $a(t)$, and $b(t)$, consider the problem:

$$
\begin{aligned}
\frac{\partial^2 u}{\partial t^2} - c\frac{\partial^2 u}{\partial x^2} &= f(x,t) && \text{for } 0 < x < 1 \text{ and } 0 < t \leq 1 \\
u(x,0) &= u_0(x) && \text{for } 0 < x < 1 \\
\frac{\partial u}{\partial t}(x,0) &= u_1(x) && \text{for } 0 < x < 1 \\
u(0,t) &= a(t) && \text{for } 0 < t \leq 1 \\
u(1,t) &= b(t) && \text{for } 0 < t \leq 1.
\end{aligned}
$$

(a.) Give a complete description of a centered finite difference method in space and time using a uniform grid size h in x and a uniform time step δ.

(b.) Discuss the accuracy of the method you defined in Part a; in particular, derive an expression for the truncation error of the method.

8. *Numerical PDEs*

Given the functions $f(x,y)$ and $g(x,y)$, consider the problem:

$$\begin{aligned} -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + g(x,y)u = f(x,y) \qquad & \text{for } 0 < x < 1 \text{ and } 0 < y < 1 \\ u(x,0) = u(x,1) = 0 \qquad & \text{for } 0 \le x \le 1 \\ u(0,y) = u(1,y) = 0 \qquad & \text{for } 0 \le y \le 1. \end{aligned}$$

(a.) Discuss how you would determine an approximate solution of this problem using a piecewise linear finite element method.

(b.) Discuss the factors that affect the accuracy of finite element methods for the approximation solution of this problem.

9. *Fourier Transform/Optimization*

Consider the optimization problem of finding a function $p(t)$ which minimizes

$$E(p) = \int_{-\infty}^{\infty} \frac{1}{2}(p'' - (3q^2 - 1)p)^2 - q(q^2 - 1)p \, dt \,,$$

where $q(t) = \tanh(t/\sqrt{2})$. The boundary/constraint condition for $p(t)$ is

$$p(\pm\infty) = p'(\pm\infty) = p''(\pm\infty) = 0 \,.$$

(a.) (12.5%) Given a function $v(t)$, if the function $u(t)$ satisfies

$$\lim_{\epsilon \to 0} \frac{1}{\epsilon}(E(p + \epsilon v) - E(p)) = u \cdot v$$

where the inner product $u \cdot v \doteq \int_{-\infty}^{\infty} u(t)v(t) \, dt$, we denote $u(t)$ by $\frac{\partial E(p)}{\partial p}$. Express $u(t)$ in terms of the functions p and q.

(b.) (12.5%) What is the relationship between $u(t)$ and the steepest descent direction of this optimization problem? Why?

(c.) (12.5%) If we use the steepest descent method to solve this optimization problem, what is the Wolfe condition?

(d.) (37.5%) Suppose a smooth function f is periodic with a period [a, b]. Write a routine in Matlab to numerically calculate the second order derivative f'' by a fast fourier transform, given a mesh size $h = (b - a)/N$.

(e.) (12.5%) If we assume that $p \equiv 0$ for $|t| > 10$, how can you use the routine in part (d) to calculate $u(t) = \frac{\partial E(p)}{\partial p}$?

(f.) (12.5%) With the same assumption as in part (e), we would like to use the nonlinear CG method to solve this optimization problem by finding $p(t)$ for $t \in [-10, 10]$. How should we update the direction by Fletcher-Reeves method? Write down a paragraph of Matlab code for this direction updating.

10. *Linear Programming*

Consider the following linear program:

Max $x_1 + 2x_2$, such that $x_1 + x_2 \leq 40$, $2x_1 + x_2 \leq 60$, $0 \leq x_1 \leq 30$.

(a.) Write this linear program in standard form.

(b.) Solve this problem by the simplex method.

11. *Statistical Methods*

Suppose we have data from an obese patient in a weight loss program. It is stored in the file `wt.dat` and it can be downloaded from `http://people.scs.fsu.edu/ xwang/wt.dat`. There are three numbers on each line of the file which represent, in order, the patient id, the number of days since the beginning of the program, and the weight (in kg). You can use the following `R` commands to take a look at the data:

```
wt <- read.table("http://people.scs.fsu.edu/~xwang/wt.dat")
wt
```

In general, the weight loss is at a diminishing rate as the program progresses. We can consider the model

$$Y_i = \beta_0 + \beta_1 e^{-(X_i/\beta_2)^{\beta_3}} + \epsilon_i$$

where $\beta_1 > 0$, $\beta_2 > 0$ and $\beta_3 > 0$. Also, Y_i denotes the observed weight in the i-th row, X_i is the days since the beginning of the program, and ϵ_1, ϵ_2, ..., ϵ_n are independent random errors.

Please try the following commands in `R`.

```
wt.nls <- nls(formula = Weight ~ b0+b1*exp(-(Days/b2)^b3), data = wt, start = c(b0=90,
b1=95, b2=190, b3=1), trace = T)
summary(wt.nls)
vcov(wt.nls)
```

From the output of the above three commands, you can find the estimated value for β_i, $i = 0, \ldots, 3$ and the corresponding variance matrix.

To answer the following questions you can only use the output from the above commands and a calculator. You can use $t_{(df),\alpha/2}$ to denote the student t-test output, but please specify the degree of freedom df, and α.

(a.) (12.5%) Give an interpretation of β_0 and β_1 in this model.

(b.) (12.5%) Test the hypothesis $H_0 : \beta_3 = 1$, with $H_a : \beta_3 \neq 1$.

(c.) (12.5%) Estimate the mean weight in the 100th day from the beginning of the program.

(d.) (25%) Get an approximate 90 percent confidence interval for the mean weight for the 100th day from the beginning of the program.

(e.) (12.5%) Estimate the mean time (in days) when the patient weighs 100kg.

(f.) (25%) Get an approximate 90 percent confidence interval for the days that the patient weighs 100kg.