Prelim Exam

Start: Thur. Jan. 10, noon. End: Mon. Jan. 14, noon. Return all copies to Cecilia. Staple your exam, and put your name on the first page.

Question 1: Datastructures

I am interested in solving a 2-D differential equation on complex physical domain. To this end, I generated a multi-domain grid (i.e., the physical domain is covered by a collection of simpler curvilinear meshes such that each point in the physical domain is within one or more of these meshes). From a programmatic point of view, the grid is described by the following datastructure (I use a C++ structure, but this could be Java or Fortran 90):

```
struct Grid {
  int nx, ny;    // number of points in the i and j directions
  float **x;     // x(i,j), i = 0,...,nx-1
  float **y;     // y(i,j), j = 0,...,ny-1
  float area;    // area covered by the grid (precomputed once the grid is known)
  ...            // additional variables, not of relevance here
};

int nb_grids;
Grid* grid;
```

The main program will take care of actually constructing `nb_grids` `Grid` structures. Of course, they will be in random order. For reasons of efficiency, I am interested in sorting these grids in order of increasing area.

The numerical algorithm requires finding the `Grid` structure whose area is closest to a specified area A. This operation will be required millions, if not more, times.

(a) Assuming that the adaptive grid is static (does not change during the course of the algorithm), write a method (language must be one of Fortran 90, C++, Java) with signature

```
Grid* getGrid(float grid_area);
```

that returns the grid whose area is closest to `grid_area` in $O($`nb_grids`$)$ operations.

(b) Can this problem be solved with a linked lists? Why or why not?

(c) Can this problem be solved with a hash table? Why or why not?

There is no need to actually code this up, although you can if you have the time (no additional points will be given though).

Question 2: Interpolation
Consider the following function:

$$f(x) = x + H(x+1) - H(x-1)$$

where $H(x)$ is the Heaviside function. Consider the discrete version of $f(x)$ at the N points

$$x_i = -2 + ih$$

where

$$h = \frac{4}{N-1}.$$

We will call the discrete function

$$F_i = f(x_i).$$

The interpolating function is $G(x)$, with the understanding that

$$G(x_i) = F_i$$

Consider the following interpolation schemes:

1. Local Linear interpolation
2. Local Cubic interpolation
3. Local Hermite interpolation

Do the following

(a) Provide a bound on the error committed when evaluating $f(.99)$ for $N = 11$ and $N = 101$.

(b) Explain the advantages and disadvantages of each interpolation method applied to this case for both a low N and high N.

Question 3: Approximation
We wish to approximate the following functions by truncated series expansion:

1. $$f(x) = \sin(2x + 0.1x^3)$$
   on the interval $[0, 2\pi]$.

2. $$f(x) = \cos(2x + 0.01x^3)$$
   on the interval $[-2, 2]$

3. $$f(x) = H(x-1) - H(x+1)$$
   on the interval $[-2, 2]$, where $H(x)$ is the heaviside function

(a) Explain the advantages and disadvantages of expanding each of the above functions with each of the approximation methods listed below:

1. Truncated Fourier series
2. Truncated Chebyshev series
3. Truncated power series

(b) Compute the three-term Legendre expansion of $f(x) = H(x-1) - H(x+1)$ on the interval $[-1, 1]$ and on the interval $[-2, 2]$. Provide the analytical expression for the coefficients of the expansion.

Question 4: Monte-Carlo

Consider a square with sides of length two units, and hence area $A = 4$, and an inscribed circle of unit radius, and hence area π. Use a simple Monte Carlo program to estimate π.

If a single Monte Carlo run consists of N points, and supposing that the error E in your estimate of π decreases like

$$E \approx N^{-\alpha},$$

do a series of runs and estimate α. Explain your procedure carefully and in detail.

Question 5: Fourier Transform

(a) Show that, for a real-valued, one-dimensional function, the power spectral density (PSD) and the autocorrelation function form a Fourier-transform pair.

(b) Briefly explain the principles of the fast Fourier transform (FFT) algorithm. (You might want to demonstrate on a short data set of length 4.)

Question 6: Maximum Likelihood

Show how linear least-squares fitting to a data set with Gaussian noise can be justified as a maximum-likelihood estimator for the unknown parameters.

Question 7: Singular Value Decomposition

Discuss the advantages of using singular-value decomposition (SVD) instead of the normal equations for linear least-squares fitting to an ill-conditioned data set.

Question 8: PDEs
Given a constant $d > 0$ and functions $f(x,t)$, $u_0(x)$, $a(t)$, and $b(t)$, consider the problem:

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

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

b. Discuss the accuracy of method you defined in Part a.

Question 9: PDEs

Given the function $f(x, y)$, consider the problem:

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

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

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

Question 10: Parallel Solution of Recurrences

Suppose a large application program has the following code segment in it. Suppose further that this code segment is executed many times during the execution of the application code (with other unseen code executed in between). You may assume that the value of n is fixed for the entire execution of the application program.

```
do i = 2, n
  y(i) = y(i) + COEF(i)*y(i-1)
end do
```

where `COEF(1:n)` and `y(1:n)` have been set (or reset) before entering the code segment.

1. Write in matrix form the system of equations that relate the values of `y(1:n)` before entering the code segment to those upon exiting the code segment. (Be careful with indices.)
2. Suppose n is large and the code is executed on a parallel processor with p processors. Assuming p is moderate in size, i.e., $p > 4$ but $p \ll n$. Which of the parallel recurrence solvers discussed in class would you apply and why?
3. Suppose the values in the array `COEF(1:n)` do not change from their initial values over the entire execution of the application program. How could this be exploited by the recurrence solver you have chosen?
4. Suppose that all that is really required upon exiting each instance of executing code segment is the value `y(n)`. What recurrence solver would you recommend using and how would it exploit the supposition? Note it is still assumed that all values in `y(1:n)` when entering the code segment may have changed from the last execution of the segment.
5. Suppose the values in the array `COEF(1:n)` are all the same, i.e., `COEF(i)`$= \alpha$ for some scalar α. How would the recurrence solver you have chosen expoit this fact?

Part 1. Bases and Transformations

1. Suppose you have a set of vectors $x_i \in \mathbb{R}^n$ with $1 \le i \le k$ and $k \le n$ and a nonsingular matrix $M \in \mathbb{R}^{n \times n}$. Consider the set of vectors $z_i = Mx_i$ with $1 \le i \le k$.

   (a) Suppose the x_i, $1 \le i \le k$, are linearly independent. Are the z_i, $1 \le i \le k$, linearly independent?

   (b) Suppose the x_i, $1 \le i \le k$, are linearly dependent. Are the z_i, $1 \le i \le k$, linearly dependent?

2. Let $A \in \mathbb{R}^{m \times n}$ with $m \ge n$ have column rank $k \le n$. Recall that if $k = n$ then A can be transformed to upper triangular matrix via Householder reflectors, i.e.,

$$H_n \dots H_2 H_1 A = HA = \begin{pmatrix} R \\ 0 \end{pmatrix}$$

   where R is an $n \times n$ upper triangular matrix with nonzero entries on the diagonal and H is an orthogonal matrix.

   (a) Suppose $k < n$, i.e., A is not full column rank, and suppose the first k columns of A are **linearly independent**. Apply k steps (Householder reflectors) of the triangularization algorithm and consider

$$H_k \dots H_2 H_1 A = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$$

   where R_{11} is an $k \times k$ upper triangular matrix with nonzero entries on the diagonal. What can be said about the zero/nonzero element pattern of R_{12} and R_{22}. Justify your answer. (Hint: Consider the result from part (1).)

   (b) Under the same assumptions as part (2a) except that the first k columns of A are not necessarily linearly independent (but A still has column rank k), how would you modify the Householder-based triangularization algorithm to produce a matrix

$$\begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$$

   that has the same structure as you identified in part (2a)? How is this matrix related to A?

Part 2. Bases and Linear Programming

Consider the linear program defined by A, b, and c

$$\min_{x\in\mathbb{R}^n} c^T x$$

$$\text{subject to:}\begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

where A is an $m \times n$ matrix with $m < n$ and the rank of A is m, i.e., linearly independent rows. Also suppose that $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$.

Let $m = 2$, $n = 3$, and

$$b = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ and } c = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \text{ and } A == \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 0 \end{pmatrix}$$

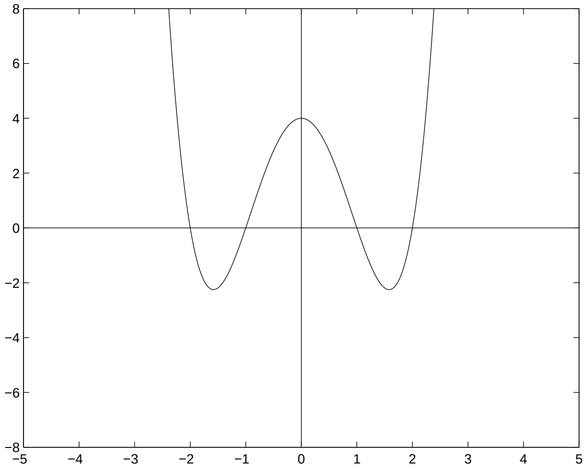
1. Find all of the basic solutions for the constraint equations.
2. What are the basic feasible solutions?
3. What are the minimizer and minimal value of $f(x) = c^T x$ subject to the constraints?

Question 12: Newton's Method for Unconstrained Optimization

Let $f(x) : \mathbb{R} \to \mathbb{R}$ be given by

$$f(x) = x^4 - 5x^2 + 4$$

and consider applying Newton's method for optimization. Here Newton's method refers to the basic form where the step size is 1 and nothing is done to alter the Hessian to guarantee positive definiteness. Note that $f(x)$ is a scalar function of a scalar argument and has the form



1. What are the values of x that are local minimizers or local maximizers of $f(x)$. Justify your answers.
2. Find the value $\beta > 0$ such that $f(x)$ has negative curvature for $-\beta < x < \beta$, and positive curvature outside the interval, i.e., for $x < -\beta$ or $x > \beta$.
3. What happens to the Newton step at $x = \beta$?
4. Determine $\mu(x) : \mathbb{R} \to \mathbb{R}$ such that the step of Newton's method applied to $f(x)$ can be written as $x_{k+1} = \mu(x_k)x_k$.
5. Find the value of $\alpha \in \mathbb{R}$ such that $\beta > \alpha > 0$ and Newton's method cycles and does not converge when $x_0 = \alpha$ or $x_0 = -\alpha$. That is, $-\alpha = \mu(\alpha)\alpha$ and $\alpha = -\mu(-\alpha)\alpha$.
6. Show that if $-\alpha < x < \alpha$ then
$$|\mu(x)| < 1$$
7. Show that if $-\alpha < x_0 < \alpha$ is the initial point for Newton's method then there is a constant $0 < \gamma < 1$ (possibly dependent on x_0 but independent of k) such that
$$|x_{k+1}| < \gamma |x_k|$$
and therefore $x_k \to 0$.

8. It can be shown that if $x_0 > \sigma > 0$ where σ is the rightmost local minimizer of $f(x)$ then $x_k \to \sigma$ for Newton's method. Using this fact and those above, show that it is possible to choose $-\beta < x_0 < -\alpha$ so that $x_k \to \sigma$ for Newton's method.

9. Implement Newton's method (in MATLAB preferably) and demonstrate the convergence behavior determined above. Attach the code and the results to your answers.

Question 13: Numerical Integration

Consider the definite integral

$$I = \int_1^3 (x-1)4^{-x}\,dx$$

The associated indefinite integral is

$$I(x) = \left[\frac{1-x}{\log 4} - (\log 4)^{-2}\right] 4^{-x}$$

(a) find the lowest order composite trapezoidal approximation for I that is accurate to eight (decimal) digits.

(b) find the lowest order composite Simpsons approximation for I that is accurate to eight (decimal) digits.

(c) find the lowest order GaussLegendre quadrature approximation for I that is accurate to eight (decimal) digits.

(d) explain the rapid drop in the order of the rules (just tell why it happens, no derivations are required).

(e) explain how you could use adaptive quadrature to compute this integral more efficiently. You need not actually do the computation.

(f) explain how you could use accelerated convergence techniques, e.g. Romberg, to compute this integral more efficiently. You need not actually do the computation.

Question 14: Numerical Differentiation

Consider the function

$$f(x) = e^x$$

(a) state whether you are doing the subsequent calculations in single or double precision.

(b) for an appropriate series of values for h estimate the derivative of at using the first order formula

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

and estimate the value of h which minimizes the error in the estimate of the derivative. Be sure your series of h values is chosen to clearly support your conclusion of which h value minimizes the error.

(c) repeat part (b) using the following formula to estimate the derivative:

$$f'(x) \approx \frac{-f(x+2h) + 8f(x+h) - 8f(x-h)f(x-2h)}{12h}$$

(d) for parts (b) and (c), explain why the errors in first decrease with decreasing h and then, at some value of h, begin to increase as h decreases further.

(e) explain why the optimum h estimated in part (b) is smaller than the optimum h estimated in part (c).