
Department of Scientific Computing

Written Preliminary Examination

Summer 2011

July 6, 2011

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 Dr. Sachin Shanbhag no later than 1:00 pm on Saturday, July 9, 2011; no exceptions allowed. If you finish ahead to time, slip it under the door to his office.
- 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.

---

1. *Linear Algebra* (Dr. Peterson)

One way to generate a stationary iterative method for the solution of $A\vec{x} = \vec{b}$ is to split A as $A = A_1 + A_2$ to obtain the following iteration. Given $\vec{x}^0$; for $k = 0, 1, 2, \ldots$ find $\vec{x}^{k+1}$ by solving the linear system

$$A_1\vec{x}^{k+1} = -A_2\vec{x}^k + \vec{b}\,.$$

Here $P = -A_1^{-1}A_2$ is called the *iteration matrix* for the method where of course A_1 is invertible.

a. The component form of the standard Gauss-Seidel method is

$$x_i^{k+1} = \frac{1}{a_{ii}}\big(b_i - \sum_{j<i} a_{ij}x_j^{k+1} - \sum_{j>i} a_{ij}x_j^k\big) \quad i = 1, 2, \ldots, n$$

where a_{ij} denotes the (i, j) entry of the $n \times n$ matrix A. Derive A_1 and A_2 for this standard Gauss-Seidel method.

b. Unlike the Jacobi method, in the standard Gauss-Seidel method the ordering of the unknowns is important. In the method of (a) we begin by updating the first coordinate of our iterate. We can also begin with the *last* component of the iterate (i.e., update x_n^{k+1} first). Derive the matrix and component form of this modified Gauss-Seidel method.

c. An obvious extension of the standard Gauss-Seidel method is at each iteration to perform two updates; in the first we start with the first component of $\vec{x}^{k+1}$ (as in the standard Gauss-Seidel given in (a)) and create say $\vec{y}^{k+1}$ and then do a second update where we start with the last component, i.e., using your method of (b) starting with $\vec{y}^{k+1}$ to obtain $\vec{x}^{k+1}$. Obtain the iteration matrix P for this method.

d. Use your algorithm in (c) to obtain $\vec{x}^1, \vec{x}^2$ for the linear system $A\vec{x} = \vec{b}$ where

$$A = \begin{pmatrix} 10 & -1 & 0 \\ -1 & 10 & -2 \\ 0 & -2 & 10 \end{pmatrix} \qquad \vec{b} = \begin{pmatrix} 9 \\ 7 \\ 6 \end{pmatrix} \qquad \vec{x}^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

e. If your original matrix A is symmetric and positive definite, then prove that your iteration matrix P in (c) is also symmetric and positive definite.

---

2. *Partial Differential Equations* (Dr. Peterson)

a. Let Ω be a bounded domain in R^2 with boundary $\Gamma = \Gamma_1 \cup \Gamma_2$ where $\Gamma_1 \cap \Gamma_2 = \emptyset$. Consider the following PDE and boundary conditions for $u(x, y)$

$$-\Delta u + uu_x = f(x, y) \quad (x, y) \in \Omega$$

$$u = 0 \quad \text{on } \Gamma_1 \qquad \frac{\partial u}{\partial n} = 4 \quad \text{on } \Gamma_2$$

and the weak formulation

Seek $u \in \hat{H}^1$ satisfying

$$\int_\Omega \nabla u \cdot \nabla v + \int_\Omega uu_x v = \int_\Omega fv + 4 \int_{\Gamma_2} v \quad \forall v \in \hat{H}^1$$

where $\hat{H}^1$ is all functions that are zero on Γ_1 and which possess one weak derivative. Here $\Delta u = u_{xx} + u_{yy}$ and $\partial u/\partial n$ denotes the derivative of u in the direction of the unit outer normal, i.e., $\nabla u \cdot \vec{n}$. Show that if u satisfies the classical boundary value problem then it satisfies the weak problem. Then show that if u is a sufficiently smooth solution to the weak problem, then it satisfies the PDE and the boundary conditions.

b. Now let $w = w(x, t)$ and consider the initial boundary value problem

$$w_t - w_{xx} = f(x, t) \quad 0 \le x \le 2, \quad t > 0$$

$$w(0, t) = w(2, t) = 0 \quad t > 0$$

$$w(x, 0) = w_0 \quad 0 \le x \le 2$$

Write down an *implicit* finite difference scheme for this problem which is *second order in space and time*. Then show that at a fixed time, we are required to solve a linear system $A\vec{w} = \vec{f}$ and explicitly give A and $\vec{f}$.

c. Let $w(x, t)$. Derive a finite difference approximation to $w_{xxx}(x, t)$ using the values $w(x, t)$, $w(x + h, t)$, $w(x - h, t)$ and $w(x + 2h, t)$. Determine the truncation error for your approximation.

---

3. *Constrained Optimization* (Dr. Navon)

Consider the L_1 exact penalty method for the constrained optimization problem

$$\min f(x),$$

$$s.t. \quad \begin{array}{l} g_i(x) = 0,\ i \in \mathcal{E} \\ g_i(x) \geq 0,\ i \in I. \end{array}$$

We are assuming that f and constraint functions g_i are continuous.
Consider the penalty function:

$$\pi_\rho(x) = f(x) + \rho \sum_{i \in \mathcal{E}} |g_i(x)| - \rho \sum_{i \in I} \min(0, g_i(x)).$$

Apply it to the problem

$$\min f(x) = \frac{1}{2}(x_1 - x_2)^2 + \frac{1}{2}x_2^2,$$
$$s.t. \ \ x_1 \geq 1.$$

Please derive the function $\pi_\rho(x)$, then show that for any $\rho > \frac{1}{2}$, $x_1 = 1$ is the minimizer of the function f - and that this yields $x_2 = \frac{1}{2}$.
Hence for any $\rho > \frac{1}{2}$, solution x^* is also a minimizer of the penalty function $\pi_\rho(x)$.

Hint: The function $\pi_\rho(x)$ is not differentiable when $x_1 = 1$.

---

4. *Unconstrained Optimization* (Dr. Navon)

Consider the Davidon-Fletcher-Powell (DFP) Q-N method for

$$f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

with starting point $(0, 0)^T$. The initial approximation for the Hessian is the unit matrix

$$H_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

a. Show that the first search direction d_1 is the same as the steepest-descent direction $-\nabla f(x_1)$.

b. Obtain minimizing step-length λ_1^* along d_1, update matrix H_1 and compute one full DFP iteration. Compute H_2 and the new search direction $d_2 = -H_2\nabla f(x_2)$

Find minimizing step-length λ_2^* along d_2 by minimizing

$$f(x_2 + \lambda_2 d_2)$$

and obtain x_3.

Show that this point x_3 is the optimum by showing that $\nabla f(x_3) = (0, 0)^T$.

Hint: the DFP Q-N update is

$$H_{k+1} = H_k + \frac{p_k p_k^T}{p_k^T y_k} - \frac{(H_k y_k)(H_k y_k)^T}{y_k^T H_k y_k}.$$

---

5. *Parallel Programming* AMB

Consider the following C++ program:

```
float prefixSum(float* x, int size) {
     float sum = 0;
     for( int i=0; i< size; i++){
          sum = sum + x[i];
     }
     return sum;
}
```

a. Explain why the loop inside the `prefixSum` function cannot be parallelized as it is. Modify this loop and make a parallel version of it using pseudo code. In your algorithm show what loops are executed in parallel, and which variables are private or shared. You can assume the size of the vector `x` is a power of two.

b. Describe the theoretical speedup of your algorithm.

---

6. *Datastructures* (Dr. Wang)

Suppose a simple closed surface divides the 3D space into two parts: inside and outside. Consider an unstructured triangular mesh of this closed surface: one array of integers `tri(nTri, 3)` that contains the indexes of the three vertices belonging to each of the `nTri` triangles, and one array of real numbers `node(nVert, 3)` that contains the positions of the `nVert` vertices.

We wish to compute the unit normal vectors uniformly towards inside or outside for each triangle with the least possible cost.

a. In order to complete this task, you may need to reorganize the data with a datastructure. If you do, please make a detailed description of your datastructure, and write the pseudocode used to generate the new data.

b. Write a pseudocode that implements your algorithm to compute the required unit normal vectors based on your datastructure.

c. In the most common case, what is the total asymptotic cost of your algorithm?

---

7. *Linear Algebra* (Dr. Wang)

Suppose A is a nonsingular $n \times n$ matrix. We would like to solve the system:

$$\begin{bmatrix} A & g \\ h^T & \alpha \end{bmatrix} \begin{bmatrix} x \\ \mu \end{bmatrix} = \begin{bmatrix} b \\ \beta \end{bmatrix}.$$

where α, β are real numbers and h, g and b are n-dimensional vectors. The unknowns are vector x and real number μ.

In general, we cannot solve the system as fast as $O(n)$ flops. But let us say that we already know the solutions to the linear systems $Az = b$ and $Ay = g$.

a. With the help of z, y, please show how to solve the system in $O(n)$ flops.

b. Write down your algorithm in pseudo code.

---

8. *Approximation* (Dr. Shanbhag)

We want to approximate the function

$$u(r) = \frac{1}{r^{12}} - \frac{1}{r^6},$$

over the range $0.9 \leq r \leq 2.0$.

a. Using Chebyshev polynomials as a basis, find a tenth order polynomial approximation of the form:

$$p_{10}(r) = \sum_{i=0}^{10} a_i T_i(\alpha r + \beta),$$

where α and β are constants needed to transform the domain from $[0.9, 2.0]$ to $[-1, 1]$, and a_i are the weights associated with T_i. You may use either "interpolated" or "truncated" Chebyshev series. Report the approximate $p_{10}(r)$, and a present a plot of the approximate and actual function over the domain $0.9 \leq r \leq 2.0$. (85 points)

b. Estimate the L_2 norm for the approximate function. (15 points)

---

9. *Gaussian Quadrature* (Dr. Shanbhag)

We are interested in developing quadrature rules for integrals of the form:

$$I = \int_0^{10} e^{-x} f(x) dx.$$

a. Find the first three (i=0, 1, and 2) *orthonormal* polynomials $P_i(x)$, with weight function $w(x) = e^{-x}$, and range [0, 10]. (60 points)

b. Find the roots x_1 and x_2 of the polynomial $P_2(x)$. (10 points)

c. Develop a 2-point quadrature rule $I \approx w_1 f(x_1) + w_2 f(x_2)$. Compare with the corresponding Gauss-Laguerre quadrature rule. (30 points)

---

10. *Ordinary Differential Equations* (Dr. Erlebacher)

Consider the system of differential equations for $Q = (x(t), y(t))$:

$$\begin{aligned} \frac{dx}{dt} &= ax + by \\ \frac{dy}{dt} &= cx + dy \end{aligned}$$

Initial conditions are $x_0 = x(0) = 1$ and $y_0 = y(0) = -2$. Let $h = \Delta t$ be the numerical time step.

a. Choose non-zero values of the parameters a, b, c, d such that the analytical solution to the differential equation has decaying solutions. (10 points)

b. Derive a first order numerical algorithm using Taylor series. Show all steps of your derivation. (15 points)

c. Derive a second order numerical algorithm using Taylor series. Show all steps of your derivation. (15 points)

d. Derive a third order numerical algorithm using numerical integration formulas. Show all steps of your derivation. (15 points)

e. Implement these three approaches numerically and solve the system of equations for $t = 0$ to $t = 1$ for a range of time steps ranging from $h = 10^{-4}$ to $h = 10^{-1}$ by increments of 10. Include the source code. Choose any language for coding. (20 points)

f. Compute numerically the accuracy of the solution by comparing the numerical solution to the exact solution of the system of equations. Verify that the accuracy of the three algorithms are first, second and third order accurate. Present plots of the logarithm of the L2 norm of the error vector as a function of time for the different time steps. (15 points)

g. What are the conditions on a, b, c, d that lead to instability in the exact solution? (10 points)

---

11. *Statistics and Random Processes* (Dr. Beerli)

A dataset contains heights of fictitious eagles that have a sexual dimorphism: females are considerably larger than males. The data can be downloaded from `http://people.sc.fsu.edu/~beerli/eagle.txt` (also see below). The data does not contain any information about the sex of the individuals. We also assume that the standard deviations of the height in males and females is the same and known from some breeding study ($\sigma = 100$).

Task:

- **Assign (and show) a probability of being female to every bird in the sample [the data is ordered from smallest to tallest]**.
- **Give the parameter values for the mixture model:** p, μ_1, μ_2
- **Show your program and thoughts**.

Use this mixture model with parameters p, μ_1, and μ_2, the standard deviations are fixed at $\sigma_1 = \sigma_2 = 100$:

$$\text{Prob}(x_i|p, \mu_1, \mu_2, \sigma_1 = 100, \sigma_2 = 100) = (1-p)\frac{1}{\sqrt{2\pi}\sigma_1}e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}} + p\frac{1}{\sqrt{2\pi}\sigma_2}e^{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}}$$

and estimate the distribution parameters. This could be done either with a least-square method, a Bayesian approach (using MC or MCMC), or an expectation/maximization approach. Pick the approach that suits you best. You need to show your computer program. I expect that you code all functions, therefore no high-end matlab/python etc function, such as *lsqlin* etc. Once you have inferred the model parameters of the mixture model, calculate the probability of being female.

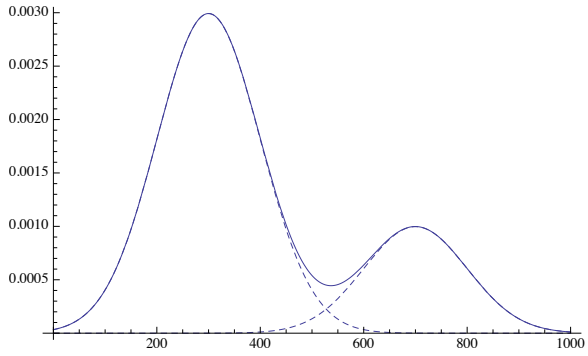

Figure 1: Example distribution, this figure does not necessarily match the data; the probability being female is essentially the cumulative distribution function of the female distribution.
