Department of Scientific Computing

Written Preliminary Examination

Summer 2020

May 26-29, 2019

Instructions:

- Solve all 10 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.
- You must score 75% or more on 7 or more questions to pass the written portion of the preliminary exam.
- If you have any questions related to this exam as you work on it, please send an e-mail to the person responsible, with a copy to Dr. Tomasz Plewa (tplewa@fsu.edu). The person responsible (and their corresponding `fsuid`) is listed at the beginning of each question. Faculty email addresses have the format `[fsuid]@fsu.edu`.
- The exam is due back to Dr. Tomasz Plewa (tplewa@fsu.edu) no later than 12 noon on Friday, May 29, 2020; no exceptions allowed.
- Submission is done electronically in email with scans of your solutions sheets attached. It is suggested you submit one solution per email message (rather than one email with several attachments). The email subject line should read “solution to Problem #”.
- When submitting your solutions, include a cover page with your name and ID. Write your FSU employee ID (provided to you by Karey Fowler) on each of your answer sheets. This serves as your unique identifier and signature. Scans of your solution sheets must clearly show your ID.

1 Statistics

Dr. Beerli, `pbeerli`

Describe the Monty-Hall problem in terms of probabilities with 3 doors and several extensions of the original problem. The game show went like this:

i. There are 3 doors, behind **one** door is a prize, behind each of the other 2 doors is a goat.

ii. You (the contestant) picks a door.

iii. The show host opens one of the doors you did not choose and reveals a goat.

iv. You have now the opportunity to switch your pick or stick with your choice from before

v. The host reveals the prize.

1. Questions:

   a) What is the exact probability of winning a prize with not switching? **[2 points]**

   b) What is the exact probability of winning a prize with switching? **[4 points]**

2. We now extend the Monty-Hall problem to n doors, without changing the procedure above. Questions:

   a) What is the exact probability of winning a prize with not switching? **[4 points]**

   b) What is the exact probability of winning a prize with always switching? **[8 points]**

3. We have a Monty-Hall problem with n doors, we change the procedure to:

   i. There are n doors, and behind **one** door is a prize, and behind each of the other $(n-1)$ doors is a goat

   ii. You (the contestant) pick a door

   iii. The show host opens one of the doors you did not pick and reveals a goat.

   iv. You now have the opportunity to switch your pick or stick with your choice from before

   v. The show host opens another door with a goat, and you can again switch (or not)

   vi. The host reveals the prize once all except 2 doors are opened.

Questions:

   a) What is the probability of winning a prize with never switching? **[6 points]**

b) What is the probability of winning a prize with always switching? **[12 points]**

c) Show a simulation where the contestant is switching only with probability p (the graph should show the probability of switching p versus probability of winning). **[24 points]**

4. Extend the problem of the question (3) using simulation. The host can reveal up to m doors at one time, m is drawn uniformly from the remaining closed doors with no prize behind them. Show two graphs, that plots a contour plot with n vs. m and contour color is the probability of winning for the probabilities of switching of $p = 0.5$ and $p = 1$ **[40 points]**

---

2 Fourier Transform

Dr. Meyer-Baese, `ameyerbaese`

For a common factor FFT the following 2D DFT is used:

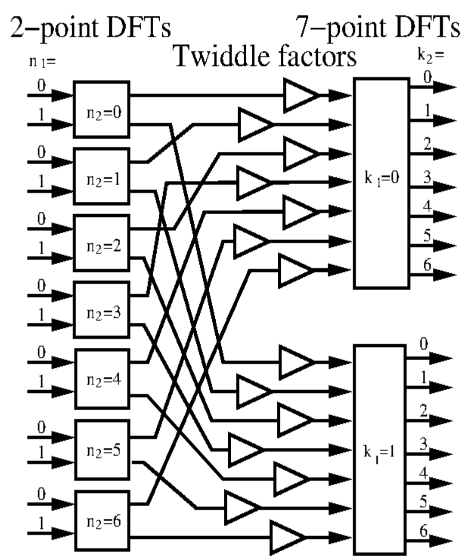
$$X[k_1,k_2] \;=\; \sum_{n_2=0}^{N_2-1} W_{N_2}^{\,n_2k_2}\left(W_N^{\,n_2k_1} \sum_{n_1=0}^{N_1-1} x[n_1,n_2] W_{N_1}^{\,n_1k_1}\right)$$

1. **[40 points]** Complete the following table for the index map for a $N = 14$ with N_1=2 and N_2=7 FFT with:

   $n = 7n_1+n_2$ and $k = k_1+2k_2$

n_1	**n_2**							**k_1**	**k_2**						
	0	**1**	**2**	**3**	**4**	**5**	**6**		**0**	**1**	**2**	**3**	**4**	**5**	**6**
0								**0**							
1								**1**							

2. **[60 points]** Complete the SFG (for x[n], X[k], and twiddle factors) for the FFT:



3 Integration and Function Approximation

Dr. Huang, chuang3

1. Calculate the following integral using Gauss-Legendre quadrature with five nodes. Note that the integral needs to be transformed to [-1,1] before using the Gauss-Legendre formula. **[30 points]**

$$Q = \int_0^5 \frac{1}{\sqrt{1+x^2}} dx$$

The nodes and weights for the Gaussian-Legendre quadrature are

i	x_i	w_i
1	-0.90617984593866399279 76	0.23692688505618908751 43
2	-0.53846931010568309103 63	0.47862867049936646804 13
3	0	0.56888888888888888888 89
4	0.53846931010568309103 63	0.47862867049936646804 13
5	0.90617984593866399279 76	0.23692688505618908751 43

2. Write a program to calculate the following integral

$$Q = \int_{-1}^{10} \frac{1}{\sqrt{1+x^2}} dx$$

using the Monte Carlo method by generating uniformly distributed random numbers in [-1,10]. Report Q for a series of N values: N =102, 103, 104, 105, 106, and 107, where N is the number of random numbers. Show that the error, ε, decays as ~ $1/\sqrt{N}$ by plotting log N versus log |ε|. The exact value for Q is 3.8795965. **[40 points]**

3. Approximate

$$f(x) = x^2\sqrt{1-x^2}$$

On the interval [-1,1] using the fourth order Chebyshev polynomial,

$$f(x) \approx a_0T_0(x) + a_1T_1(x) + a_2T_2(x) + a_3T_3(x) + a_4T_4(x)$$

where $T_0(x) = 1, T_1(x) = x, T_2(x) = 2x^2 - 1, T_3(x) = 4x^3 - 3x$, and $T_4(x) = 8x^4 - 8x^2 + 1$. Report the coefficients $\{a_i\}$. **[30 points]**

You may use any programming language to solve the above problems. Along with solutions, submit your computer code.

4 Ordinary Differential Equations

Dr. Huang, `chuang3`

In this problem, we use the finite difference method to solve the one-dimensional Schrödinger equation,

$$\left[-\frac{d^2}{dx^2} + V(x)\right]\phi(x) = E\phi(x), \tag{1}$$

where $V(x) = 0$ in [0, 10] and $V(x) = \infty$ for $x < 0$ and $x > 10$. $\varphi(x)$ is the electron's orbital and E is the energy. We first discretize the domain [0,10] into $N + 1$ points: x_0, $x_1, \ldots, x_N$, with $x_0 = 0$ and $x_N = 10$. For the boundary conditions, since $V(x) = \infty$ outside [0, 10], we have $\varphi(x_0) = 0$ and $\varphi(x_N) = 0$. Therefore we only need to solve $\varphi(x)$ at the interior points $(x_1, \ldots, x_{N-1})$. To discretize the equation, let us replace d^2/dx^2 with the central finite difference,

$$f''(x) = [f(x+h) + f(x-h) - 2f(x)]/h^2$$

with $h = 10/N$. Again, this only needs to be done for the interior points, since both $\varphi(x_0)$ and $\varphi(x_N)$ are zero. Eq. 1 then becomes an eigenvalue problem $A\overline{\varphi} = E\overline{\varphi}$ where E is the eigenvalue and $\overline{\varphi}$ is the discretized orbital ($\overline{\varphi} = [\varphi_1, \varphi_2, \ldots, \varphi_{N-1}]^T$).

1. What is the matrix A for $N = 6$? **[50 points]**

2. Write a code to solve
$$A\vec{\phi} = E\vec{\phi}$$
for N = 200. Report the lowest eigenvalue and plot the associated φ(x). If the code is written with MATLAB, you may use the eig() function to calculate the eigenvalues. For Python, you may use `numpy.linalg.eig()`. For C/C++/Fortran, you may use the LAPACK library. **[50 points]**

Submit the code you wrote to produce the reported results.

5 Function Integration

Dr. Quaife, `bquaife`

Throughout this problem, let $z_0 \in \mathbb{C}$ with $|z_0| < 1$. If γ is the unit circle in the complex plane, and $f: \mathbb{C} \to \mathbb{C}$ is holomorphic, Cauchy's Integral Theorem guarantees that,

$$\frac{1}{2\pi i}\oint_\gamma \frac{f(z)}{z - z_0}\,dz = f(z_0).$$

Recall that if γ has the parameterization,

$$\gamma = \{z(\theta) \mid \theta \in [0, 2\pi)\},$$

Then,

$$\oint_\gamma g(z)\,dz = \int_0^{2\pi} g(z(\theta))z'(\theta)\,d\theta.$$

Since the unit circle in $\mathbb{C}$ can be parameterized as,

$$\gamma = \{e^{i\theta} \mid \theta \in [0, 2\pi)\},$$

Cauchy's Integral Theorem guarantees that

$$f(z_0) = \frac{1}{2\pi i}\int_0^{2\pi} \frac{f(e^{i\theta})}{e^{i\theta} - z_0} ie^{i\theta}\,d\theta. \tag{1}$$

1. Letting $f(z) = z^2 + 1$ and $z_0 = 1$, verify by hand that equation (1) is satisfied. **[10 points]**
2. Write code that applies the trapezoid rule to equation (1). You may use any language you want. Your code's inputs should be the number of quadrature points N, a function handle for f, and a point $z_0 \in \mathbb{C}$ with $|z_0| < 1$. Your code's output should be an approximation of $f(z_0)$. Attach a copy of your code with your submission. **[30 points]**
3. Let $f(z) = \sin z$. For $z_0 = 0.1$, use your code to perform a convergence study using $N = 8, 16, 32, 64, 128$ points. Report the relative error as either a table or a loglog plot. Repeat the convergence study for $z_0 = 0.5$ and $z_0 = 0.99$. You should observe fast convergence for $z_0 = 0.1$ and much slower convergence for $z_0 = 0.99$. **[30 points]**
4. Since the function $f(z) = 1$ is holomorphic, we have

$$1 = \frac{1}{2\pi i}\oint \frac{1}{z - z_0}\,dz.$$

Therefore, Cauchy's Integral Theorem can also be written as

$$f(z_0)\frac{1}{2\pi i}\oint \frac{1}{z - z_0}\,dz = \frac{1}{2\pi i}\oint_\gamma \frac{f(z)}{z - z_0}\,dz.$$

Solving for $f(z_0)$, we have

$$f(z_0) = \frac{\frac{1}{2\pi i}\oint \frac{f(z)}{z - z_0}\,dz}{\frac{1}{2\pi i}\oint \frac{1}{z - z_0}\,dz}. \tag{2}$$

Repeat the convergence studies from part (3), but use your code to approximate both integrals in equation (2) and then divide the result to estimate $f(z_0)$. Describe what you observe. **[30 points]**

6 Numerical Quadrature

Dr. Shanbhag, sshanbhag

Consider the integral,

1. Let e_i be a $n \times 1$ unit vector with 1 in the row i and 0 elsewhere. What is the outer product $e_j e_k^T$?

[10 points]

2. Consider a $n \times n$ nonsingular matrix A. If u and v are $n \times 1$ vectors, and $A' = A + uv^T$ is a invertible matrix, verify that

$$(A')^{-1} = A^{-1} - \frac{A^{-1}uv^{\mathsf{T}}A^{-1}}{1 + v^{\mathsf{T}}A^{-1}u}, \tag{1}$$

30 points]

3. Now suppose the element at the (j, k) location in A is changed from a_{jk} to a'_{jk}, to produce matrix A'. Using your solution to the first question, find a u and v such that,

$$A' = A + uv^T.$$

[10 points]

4. Furthermore, suppose an LU decomposition is available for the matrix A, and you want to solve the problem $A'x = b$ without refactorizing the matrix A' (which would be an $\mathcal{O}(n^3)$ operation),

$$x = A^{-1}b - \frac{A^{-1}uv^T A^{-1}b}{1 + v^T A^{-1}u}. \tag{2}$$

Show step-by-step how you could compute x using the formula above in $\mathcal{O}(n^2)$ steps.

[50 points]

---

7 Data Structures

Dr. Wang, `wwang3`

In this coding project, you are required to implement a binary tree to sort a series of integer numbers. The programming language is C++. Please implement the tree from scratch, instead of using any existing library or package.

Description of the tree structure:

- Each node stores a number.
- Each node has no more than two children nodes.
- All the numbers stored in each node's left branch are less than (or equal) the number stored in the node; all the numbers stored in each node's right branch are greater than the number stored in the node.

Please implement the tree as a class as follows:

1. The class has a public member function `insert', with which we can insert a integer number to the tree. **[20 points]**
2. The class has a public member function `search', with which we can check whether a integer number is stored in the tree or not. **[20 points]**
3. The class has a public member function `output', with which we can output all the numbers stored in the tree in a sorted way. **[20 points]**
4. Write a `main' function to test your code with adequate input; present the results; supply the source code. **[40 points]**

---

8 Optimization

Dr. Navon, `inavon`

Please solve one out of the following two numerical problems:

1. Use the Quasi-Newton method of Davidon-Powell Fletcher to minimize the function,

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

from the starting point,

$$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and the initial Hessian,

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

Use minimizing step length, α, by minimizing

$$f(x + \alpha d)$$

with respect to α. Note that due to the function being quadratic, you should obtain solution in two complete iterations.

2. Consider Newton's method,

$$\nabla^2 f(x_k) p = -\nabla f(x_k)$$

for the quadratic function,

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

and with starting point,

$$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

calculate one iteration using

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

Show that the solution indeed converged in one iteration, i.e., that

$$g_2 = \nabla f(x_2) = \begin{bmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \end{bmatrix}_{x_2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

9 Optimization

Dr. Quaife, bquaife

Consider the one-dimensional PDE,

$$\frac{\partial u}{\partial t} + c(x)\frac{\partial u}{\partial x} = 0, \qquad x \in \mathbb{R},\ t > 0,$$
$$u(x,0) = u_0(x), \quad x \in \mathbb{R}.$$

(a) The characteristics of this PDE satisfy $x'(t) = c\big(x(t)\big)$. Demonstrate that u is constant along the characteristics by showing that

$$\frac{d}{dt}u(x(t),t) = 0.$$

[20 points]

(b) For the remainder of this problem, consider the PDE

$$\frac{\partial u}{\partial t} + x\frac{\partial u}{\partial x} = 0, \qquad x \in \mathbb{R},\ t > 0,$$
$$u(x,0) = u_0(x), \quad x \in \mathbb{R}.$$

Find the characteristics of this PDE.

[20 points]

(c) Draw space-time diagram with the time variable on the positive y-aixs and the space variable on the x-axis. Draw a collection of five or more characteristics that originate at $t = 0$ with $x < 0$, $x = 0$, and $x > 0$.

[20 points]

(d) Given an arbitrary point in space-time (x,t), find the corresponding point $(\tilde{x},0)$ that is on the same characteristic as (x,t). Use this point to form the solution of the PDE in terms of the initial condition. Label these points in your plot from part (c).

[30 points]

(e) Verify that your solution from part (d) satisfies the PDE and the initial condition.

[10 points]

---

10 Parallel Programming

Dr. Wang, `wwang3`

A prime twin is a pair of prime numbers that is either 2 less or 2 more than another prime number. For example, (41; 43).

In this coding project, you are required to implement a parallel code using MPI to implement a parallel algorithm using MPI to find the total number of prime twins smaller than a given number; for example, 1×10^8.

[40 points]

Please carefully design your algorithm and distribute the work as evenly as possible across the processes. Please output the wall time each process used in solving the problem (exclude waiting time). We would like to obtain good load balance between all processors. There are different ways to achieve that. Please describe the way you load-balanced your code.

[**40 points]**

Test your code with different numbers of processes (one process per processor core) and illustrated code performance by graphing wall clock time as a function of number of processes used.

[20 points]

Provide the source code along with the required results.
