Summer 2015

Department of Scientific Computing

Written Preliminary Examination

July 31-August 3, 2015

The exam solutions are due back to Mark Howard no later than

12 noon on Monday, August 3, 2015.

Instructions:

- Solve as many problems as completely as you can. Submit no more than 10 problem solutions.
- All problems are weighted equally.
- All parts of a problem are weighted equally unless otherwise noted at the end of the problem.
- Do not write your name on your solution sheets. Rather write your Student ID on each page.
- If you use additional resources (i.e. books, web), please reference them clearly.
- You may not discuss the exam with any individual other than the problem authors before 12 noon on Monday.
- If you have a question related to specific exam problem, please contact the corresponding problem's author by email with a copy to Dr. Tomasz Plewa (tplewa@fsu.edu). The exam problem authors are listed near the beginning of each problem.

1. Fast Fourier Transforms (Meyer-Baese)

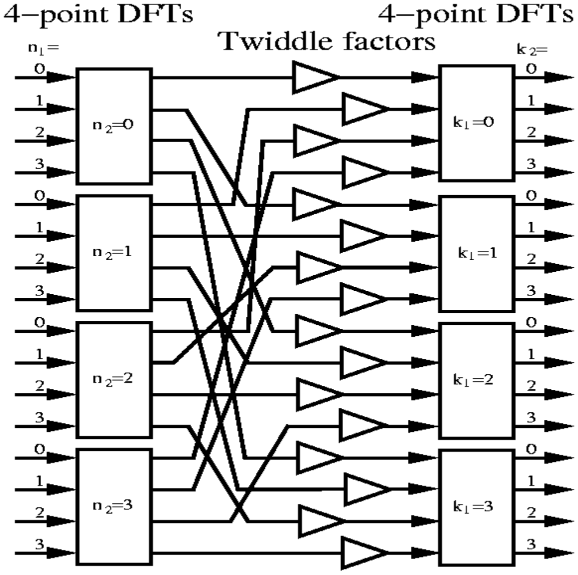
For a common factor FFT the following 2D DFT is used:

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

a) Complete the following table for the index map for a N = 16 radix-4 FFT with: $n = 4n_1+n_2$
$k = k_1+4k_2$

n_2/k_1	**n_1/k_2**			
	0	**1**	**2**	**3**
0				
1				
2				
3				

b) Complete the SFG (x, X and twiddle factors) for the N=16 radix 4 fft:



c) Compute the 16 point fft for x=[0 1 0 0 0 2 0 0 0 3 0 0 0 4 0 0] using the following steps:

i) Map the input data and compute the DFTs of the first stage.

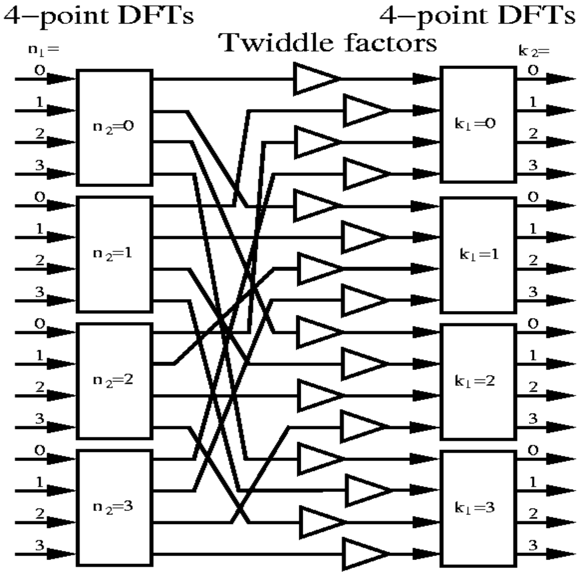
ii) Multiply the (none-zero DFTs) with the twiddle factors (hint: w=exp(-j2π/16).

iii) Compute the second level DFT.

iv) Sort the output sequence in the right order (use two fractional digits): X=

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Re																
Im																

Use the SFG to fill in the data (use two fractional digits):


4−point DFTs
Twiddle factors
4−point DFTs
n_1=
k_2=
0
1
2
3
n_2=0
n_2=1
n_2=2
n_2=3
k_1=0
k_1=1
k_1=2
k_1=3

2. Linear Algebra (Wang)

Consider the following matrix:

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

a) Prove that when $n \geq 3, A^n = A^{n-2} + A^2 - I$, where I is the 3×3 identity matrix.

b) Please calculate A^{100} by hand.

3. Finite Elements (Burkardt)

Consider the following partial differential equation:

$$-u'' + u = -e^{x} \text{ for } x \in [10, 22]$$

$$u(10.0) = 70.0; u(22.0) = 145$$

Suppose the finite element method is applied, so that the interval [10,22] is broken up into three elements of equal size, each of which uses piecewise **quadratic** basis functions. Thus, there will be a total of 7 equally spaced nodes x_i. For each node x_i, we may define a corresponding basis function $\psi_i(x)$ which is continuous everywhere, is a quadratic polynomial within each element, attains the value 1 at the node x_i and is 0 at the other nodes.

We may assume that the finite element approximate solution is represented by

$$u^h(x) = \sum_{i=1}^{7} c_i \psi_i(x)$$

Because of the boundary conditions, c_1 and c_7 are known immediately. To find the remaining coefficients, we need to set up a 5x5 linear system $Ac = f$.

a) Forming the matrix A requires evaluating the stiffness matrix

$$K(i, j) = \int_{10}^{22} \psi_i'(x)\, \psi_j^{i}(x) dx,$$

for i from 2 to 6. Simply because of the choice of piecewise quadratic basis functions, many entries of K are guaranteed to be zero or positive.

Fill the matrix below with ’0’, ’+’ or ’?’ according to whether the corresponding entry is guaranteed to be zero, positive, or is not automatically known. *This answer does not require any computation.*

	2	3	4	5	6
2					
3					
4					
5					
6					

b) Give a formula for the basis function associated with the node $x = 14.0$. Your formula must be correct for any possible value of x in [10,22]. Your formula will depend on the locations of the nodes. Use explicit values for these quantities; that is, instead of a factor like $(x - x_3)$ say $(x - 14)$.

c) Suppose that the table below contains the finite element coefficients c_i for this problem. Use this information to determine the value of $u^h(15.0)$.

i	1	2	3	4	5	6	7
x_i	10.0	12.0	14.0	16.0	18.0	20.0	22.0
c_i	70.0	85.0	100.0	50.0	75.0	125.0	145.0

d) The computation described above used a number of elements $ne = 3$ and a typical element size $h = 12/ne = 4$. Suppose that the problem is solved repeatedly, reducing the value of h each time, and that the exact solution $u(x)$ is known, and that we have recorded the L^2 norm of the error $e(h) = ||u - u^h||$ for each value of h in the table below.

For finite element applications, the L^2 norm of the error typically has an asymptotic convergence rate, related to the element size h, such as $O(1)$, $O(h^1)$, $O(h^2)$, and so on.

Using the h and $e(h)$ data in the table, fill in the row labeled “norm rate” that estimates the convergence rate by comparing successive pairs of element sizes and errors. You will need to transform the data in the usual way in order to produce estimates of the convergence rate exponent (which is typically a small integer).

ne	3	8	10	20
h	4	1.5	1.2	0.6
$e(h)$	11.8496	0.126155	0.0646452	0.00808974
norm rate	XXX			

Does your observed norm rate agree with the expected value from the theory?

4. Numerical Linear Algebra/Eigenproblems (Peterson)

Let A be an $n \times n$ full rank matrix. We are interested in solving the linear system $A\mathbf{x} = \mathbf{b}$ using an iterative method.

a) Give reasons why you would choose to use an iterative method instead of a direct solver.

b) Let $\mathbf{x}^0$ be given and consider the preconditioner form of an iterative method given by

$$(\dagger) \qquad \mathbf{x}^{k+1} = \mathbf{x}^k + Q^{-1}\mathbf{r}_k, \quad k = 0, 1, 2, \ldots$$

where Q is an $n \times n$ invertible matrix and $\boldsymbol{r}_k$ is the residual

$$\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}^k .$$

i. If $\boldsymbol{x}^k$ converges to a vector $\boldsymbol{y}$, prove that this vector is the unique solution to $A\boldsymbol{x} = \boldsymbol{b}$.

ii. The motivation for this form of the iterative method can be seen by taking $Q = A$. In this case, what is $\mathbf{x}^1$? What does this suggest we could choose for Q?

c) Let $\mathbf{e}_k = \mathbf{x}^k - \mathbf{x}$ be the error vector.

i. Show that $\mathbf{e}_k$ satisfies

$$\mathbf{e}_k = B^k\mathbf{e}_0$$

for an appropriately chosen matrix B; explicitly give B.

ii. Use your result in (i) to given a sufficient condition for convergence of the iterative method $(\dagger)$ in terms of the norm of B. Justify your answer. Is this true for any matrix norm? Relate your answer in terms of the norm of B to the eigenvalues of B.

iii. Show that the residual $\mathbf{r}_k$ satisfies

$$\boldsymbol{r}_k = C^k\boldsymbol{r}_0$$

for an appropriately chosen matrix C; explicitly give C.

d) Write pseudo-code to efficiently implement $(\dagger)$ assuming you are given the matrix Q. Then

i. give the operation count for computing p iterations if Q is a full matrix;

ii. give the operation count for computing p iterations if Q is a tridiagonal matrix.

e) If B and C are similar matrices then explain how this effects the behavior of $\mathbf{e}_k$ and $\mathbf{r}_k$.

Point distribution: a - 10%, b - 25%, c - 25%, d - 25 %, e - 15%

5. Interpolation/Numerical Quadratures (Peterson)

If we have $n+1$ distinct points (xi,fi), $i=0,1,\ldots,n$, then we know that we can find a unique polynomial, $pn(x)$, which satisfies the *interpolation conditions* $p_n(x_i) = f_i, i = 0, 1, \ldots, n$. However, we can use functions other than polynomials for our interpolation basis. Here we compare polynomial interpolation with interpolation using basis functions of the form
$e^{-(x-x_i)^2}, i = 0,1,\cdots,n$, instead of the standard polynomial basis $1, x, x^2, \ldots, x^n$.

For this problem, we will use the concrete example where we have three evenly spaced points on the interval $[-1, 1]$ and we want to interpolate the function $f\ (x) = 1/(1 + x^2)$ on this interval.

a) Recall that the Lagrange form of the interpolating polynomial is

$$p_n(x) = \sum_{i=0}^{n} a_i l_i(x)$$

where $l_i(x)$ are polynomials of degree n which satisfy $l_i(x_j) = \delta_{ij}$ (the Kronecker delta).

i. Explicitly give the Lagrange form of the interpolating polynomial for the given $f(x)$ and the three interpolation points; i.e., don't just use x_i, f_i but give the exact values. Plot $f(x)$ and this polynomial on $[-1, 1]$ using 100 evenly spaced points.

ii. Compute the l_2 norm of the error using these 100 points and then get the best theoretical bound on the error and compare the two results.

Use your interpolation formula to obtain an approximation to

$$\int_{-1}^{1} \frac{1}{1+x^2}\, dx$$

This is equivalent to using what quadrature rule? Calculate your exact error.

b) Now suppose we seek an interpolating function of the form

$$s(x) = \sum_{k=0}^{n} c_k \phi_i(x)$$

where

$$\phi_i(x) = e^{-(x-x_i)^2}$$

When using basis functions like these it is advantageous to also impose the condition that s(x) interpolates constant functions exactly which means we seek s(x) of the modified form

$$\mathrm{s}(x) = \sum_{k=0}^{n} c_k \phi_i(x) + \beta$$

where β is an unknown scalar. Now if the function we are interpolating is a constant, i.e., $f(x) = C$, then

$$\mathrm{C} = \mathrm{s}(x_i) = \sum_{k=0}^{n} c_k \phi_i(x_i) + \beta = \sum_{k=0}^{n} c_k + \beta$$

for our choice of $\varphi i(x)$. The only way that this holds for each constant C is if $\sum c_k = 0$, so we add this equation to our interpolation conditions.

i. Set up a matrix equation which must be solved in order to determine the coefficients ck and β when you interpolate the given function $f(x)$ over the three equally spaced points. Explicitly compute the entries in the matrix.

ii. Solve the system. Explicitly give the coefficients and plot $f(x)$ and $s(x)$.

iii. Is it easy to use this interpolation formula to obtain an approximation to

$$\int_{-1}^{1} \frac{1}{1+x^2}\, dx;$$

why or why not?

c) Compare your results in (a) and (b). Compare the cost of obtaining each interpolation function.

Point distribution: a - 40%, b - 40%, c - 20%

6. Numerical Differentiation (Shanbhag)

a) Consider the following two formulae for numerically estimating the first derivative:

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} + \mathcal{O}(h^m) \qquad (1)$$

$$f'(x) = \frac{f(x-2h)-8f(x-h)+8f(x-h)+f(x+2h)}{12h} + \mathcal{O}(h^n) \qquad (2)$$

Using Taylor series expansion, show that m = 2 and n = 4. [25 points]

b) Suppose we are interested in the first derivative of the function

$$g(x) = \frac{e^x}{\sin^3 x + \cos^3 x}$$

at x=1. The true derivative is $g'(1) = 1.64087713599607$.
Estimate $g'(1)$ with $h = 0.01$ using eqns (1) and (2). Report the absolute error. [10 points]

c) Using eqn (1) and half the step-size $h = 0.01/2$, find a new estimate of the first derivative and the resulting absolute error.
Now, assuming

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} + constant \times h^2 + \mathcal{O}(h^4)$$

use Richardson extrapolation to combine the two estimates with step-size h and $h/2$. Write down the formula you use, and the absolute error. [30 points]

d) Numerical approximations to derivatives can be found by "integration" using the Lanczos formula:

$$g'(x) = \lim_{h\to 0} \frac{3}{2h^3} \int_{-h}^{h} t g(x+t) dt \qquad (3)$$

Use the following Newton-Cotes integration rule,

$$\int_{-h}^{h} f(x) dx = \frac{h}{4}\left(f(x-h) + 3f\left(x - \frac{h}{3}\right) + 3f\left(x + \frac{h}{3}\right) + f(x+h)\right) + \mathcal{O}(h^5),$$

to show that eqn (3) can be written as:

$$g'(x) = \frac{A}{4}\left(-g(x-h) - g\left(x - \frac{h}{3}\right) + g\left(x + \frac{h}{3}\right) + g(x+h)\right) + \mathcal{O}(h^p).$$

Determine A and p.

Point distribution: a – 25%, b – 10%, c – 30%, d – 35%

7. Statistical Analysis (Ye)

For a variable *X*, both **confidence** and **credible** intervals can be defined symbolically as $\text{Prob}(l \le X \le u) = 1 - \alpha$, where l and u are the lower and upper interval limits and α is significance level. However, the definition is interpreted in different ways when estimating regression confidence intervals and Bayesian credible intervals. Answer the following questions:

a) What are the conceptual differences between the two kinds of intervals?

b) Describe how to calculate the two kinds of intervals.

c) Under what circumstances are the two kinds of intervals the same?

d) The two intervals are always different. What are the possible reasons that render the two kinds of intervals different?

For your convenience, you may limit your discussion in the context of parameter estimation and model predictions.

8. Numerical Optimization (Navon)

Compare the Newton method and the secant method to find the sequence of errors: $\{x_k^2 - 4\}$

a) For Newton method start at the point $x_0 = 2.5$ for iterations $k = 1,2,\cdots,7$.
   - Write a small code to implement Newton method using double precision.
   - Plot the sequence of errors versus number of iterations on a semi log graph.
   - What is the convergence rate from reading the slope of the graph?

b) For the secant method use the initial starting points $x_0 = 1$ and $x_1 = 2.5$ for iterations $k = 1,2,\cdots,7$.
   - And again write a small code to implement the secant method and plot the sequence of errors versus number of iterations on a semi log graph.
   - What is the convergence rate of the secant method from reading the slope of the graph?
   - Comment after reading theoretical derivation of corresponding rates of convergence of Newton and secant method.

9. Parallel Programming (Huang)

Data exchange between processors is essential in MPI programming.

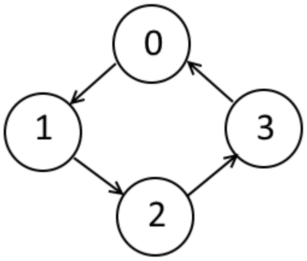
a) Briefly discuss the key difference between the following MPI functions: MPI_SSEND(), MPI_SEND(), and MPI_ISEND().

b) We try to exchange data between two processors (p1 and p2) using the following code:

```
If (myrank==p1) then
  other_processor = p2
endif
if (myrank==p2) then
  other_processor = p1
endi
CALL MPI_SSend(myValue, 1, MPI_INT, other_processor, 0, MPI_COMM_WORLD, mpi_ierr);
CALL MPI_Recv(otherValue, 1, MPI_INT, other_processor, 0, MPI_COMM_WORLD, mpi_status, mpi_ierr);
```

Explain why deadlock occurs and revise it by using the non-blocking function MPI_ISEND() to avoid the deadlock.

c) In the figure, a ring topology is created by using MPI. The idea is that the data is passed around each processor and then returns to processor 0.

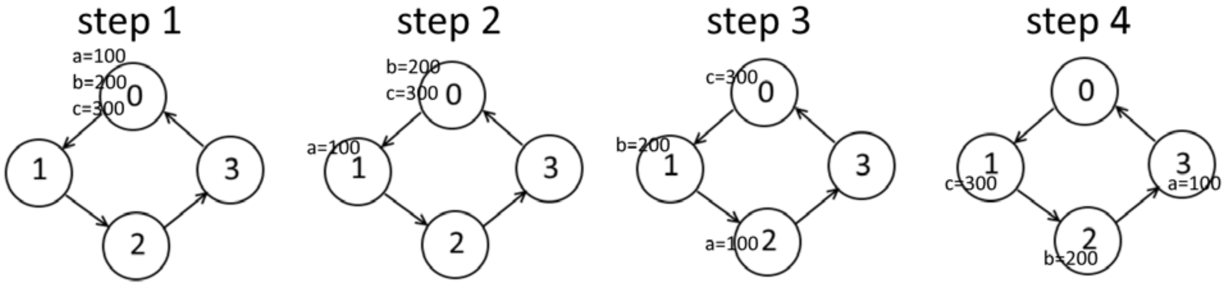


A code for realizing this “ring” is given, in which a token is passed through the ring. Discuss how it works.

```
INTEGER token;
IF (myrank /= 0) THEN
    CALL MPI_Recv(token, 1, MPI_INT, myrank-1, 0, MPI_COMM_WORLD, mpi_status, &
                  mpi_ierr);
    PRINT *,"Process ",myrank," received token ",token," from process ", myrank-1
ELSE
    ! Set the token's value if you are process 0
    token = 100;
ENDIF
CALL MPI_SSend(token, 1, MPI_INT, myrank+1, 0, MPI_COMM_WORLD);

! Now process 0 can receive from the last process.
IF (myrank == 0) THEN
    CALL MPI_Recv(token, 1, MPI_INT, myrank-1, 0, MPI_COMM_WORLD, mpi_status, &
                  mpi_ierr);
    PRINT *,"Process ",myrank," received token",token," from process ",myrank-1
ENDIF
```

d) In the above example, a token is passed around every processor with N steps, where N is the total number of processors. Here we want to design a different “ring”, in which tokens are continuously inserted into the “ring” from processor 0, as illustrated in the following figure. At the first step, there are three tokens, a=100, b=200, and c=300. In the next step, token “a” is inserted into the “ring”. At step 3, token “a” moves to processor 2, and token “b” is inserted into the “ring”. At step 4, the last token “c” is inserted into the ring.



Give statements in C, C++, or FORTRAN to realize this “ring”. No debugging is required. Discuss your code in detail and explain the underlying philosophy.

Point distribution: a – 15%, b – 25%, c – 15%, d – 45%

10. Monte Carlo Integration (Shanbhag)

For $\gamma > -1$, the integral

$$I(\gamma) = \int_0^1 x^\gamma dx = \frac{1}{\gamma + 1}.$$

We want to use Monte Carlo to evaluate this integral. Note that for $\gamma < 0$, the integrand has a discontinuity at x = 0.

We expect the (error) variance in the Monte Carlo estimate to be:

$$\sigma_I^2 = \frac{\langle f^2 \rangle - \langle f \rangle^2}{n},$$

where the numerator is the variance of the integrand $f(x) = x^\gamma$, and n is the number of independent samples. Thus, $\langle f \rangle = I(\gamma)$ and $\langle f^2 \rangle = \int_0^1 x^{2\gamma}\, dx$.

a) For (i) γ = –0.2 and (ii) γ = –0.9, estimate the integral I(γ) and the numerically estimated error, using n = $10^3 - 10^6$ independent samples. Note that you will have to concurrently estimate $\int_0^1 x^{2\gamma}\, dx$ to find the error.

b) Analytically determine the variance $\langle f^2 \rangle - \langle f \rangle^2$, and plot the expected σ_I and numerically computed error for both γ = –0.2 and γ = –0.9. Explain your observations.

11. Numerical Partial Differential Equations (Erlebacher)

a) Consider the following PDE on a periodic domain $x \in [0, 2\pi]$:

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = \sin(2x) + 0.3\sin(3x)$$

with initial condition $u(x,t) = \sin(3x)$.
Implement a 3rd order numerical scheme (in space and time), and perform a series of numerical experiments to study the spatial accuracy of the discrete solution. Compute the $L2$ error at $t = 1$ and plot it as a function of the spatial interval. Perform a linear regression to determine the rate of convergence. Give the rate of convergence. Provide all steps of your derivation.

b) Repeat the exercise in part 1, but with the following equation:

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = |\sin(2x)| + 0.3|\sin(3x)|$$

I. Compute the spatial convergence rate.
II. Is the rate of converge the same as in Part 1? State why or why not.

c) Consider the equation:

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = e^{-3x}$$

on the infinite domain $[-\infty, \infty]$ and solve the equation by first taking its Fourier transform (the transform of $u(x)$ is $U(\omega)$ where ω is proportional to spatial frequency), solving for $u(\omega)$ and transforming back to physical space. The solution is expressed using integrals from -∞ to ∞. If the integral cannot be solved analytically (you can use any method available to try and get an analytical solution), state how you would generate an accurate approximate solution (no need to actually do it.)

Point distribution: a – 50%, bI – 20%, bII– 10%, d – 20%