CS3911 Introduction to Numerical Methods with Fortran Exam 1 Fall 2006

150 points – 8 pages

Name: _____

- Most of the following questions only require very short answers. Usually a few concise and precise sentences would be sufficient. Please write to the point. If I don't understand what you are saying, I believe, in most cases, you don't understand the subject.
- *Justify your answer with a convincing argument*. If there is no justification when it is needed, you will receive ZERO point for that question even though you have provided a correct answer. *I consider a good and correct justification more important than a right answer. Thus, if you provide a very vague answer without a convincing argument to show your answer being correct, you will likely receive a very low grade.*
- Do those problems you know how to do first. Otherwise, you may not be able to complete this exam on time. If you follow our classroom discussions and understand the most basic components, you should be able to quickly complete about 50% of this exam. The remaining problems, however, test if you are able to apply and use the basics properly.

1. **Fortran 90:**
   What is the output of each of the following Fortran 90 programs? Note that some programs may contain errors. If a program has errors, you should clearly identify the errors, and explicitly state what the error is and why it is an error. **Missing any one of these three elements, you will receive zero point for that problem. Do not change any program to make it working!**

   (a) **[5 points]**

```
PROGRAM  Test
   IMPLICIT  NONE
   INTEGER :: a, b, c, d
   READ(*,*)  a, b, c
   d = a**b**c
   WRITE(*,*) d
END PROGRAM Test
```

   Input: `2 3 2`

   (b) **[5 points]**

```
PROGRAM  Sum
   IMPLICIT  NONE
   REAL :: s, x, y, k = 1.0
   READ(*,*)  x
   s = 0.0
   DO
      y = x**k
      IF (ABS(y) < 0.000005) EXIT
      k = k + 1.0
      s = s + y
   END DO
   WRITE(*,*)  x, s
END PROGRAM Sum
```

   Input: `-1.0`

(c) [**5 points**]

```
PROGRAM  Sum
   IMPLICIT  NONE
   INTEGER :: a, b, c, IO
   DO
      READ(*,*,IOSTAT=IO) a, b
      IF (IO /= 0) EXIT
      c = a + b
      WRITE(*,*) a, b, c
   END DO
END PROGRAM Sum
```

Input:

```
1    4    7
2    5    8
1.0  3.0  5.0
```

(d) [**5 points**]

```
PROGRAM  XYZ
   IMPLICIT  NONE
   INTEGER :: a, b, c = 1
   READ(*,*) a, b
   WRITE(*,*) foo(a,b)
   READ(*,*) a, b
   WRITE(*,*) foo(a,b)

CONTAINS
   INTEGER FUNCTION  foo(x, y)
      IMPLICIT NONE
      INTEGER, INTENT(IN) :: x, y
      c = x + y + c
      foo = c
   END FUNCTION  foo
END PROGRAM  XYZ
```

Input:

```
1  3
5  7
```

2. **Finite Precision Arithmetic and Accuracy:**

   (a) A program must compute the following expression for some large value of x that is sure to have no overflow when evaluating x^2:

$$\ln\left[\frac{\sqrt{1+x^2}-x}{\sqrt{1+x^2}+x}\right]$$

   Do the following questions:

   i. [**10 points**] What would be the anticipated problem or problems when evaluating the above expression if x is large? **Answer the question and provide an explanation.**

   ii. [**20 points**] How would you overcome the problem? Discuss your approach and provide a Fortran 90 expression. **To receive full credits, you have to discuss your approach, prove its correctness, and provide a solution. Just saying your approach being correct does not count.**

   iii. [**10 points**] What would be the expected result if x is large? **Don't forget to provide your detailed calculation. Only showing an answer receives zero point.**

(b) A degree n polynomial $p(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n$ has an equivalent form:

$$a_0 + (a_1 + (a_2 + \cdots (a_{n-1} + a_n x)x) \cdots)x$$

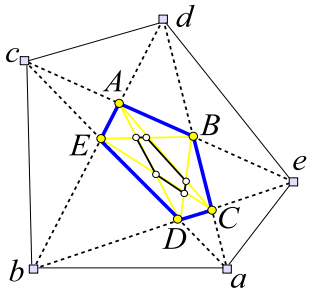
For example, $a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 = a_0 + (a_1 + (a_2 + (a_3 + (a_4 + a_5 * x) * x) * x) * x) * x$. Do the following two problems:

i. **[15 points]** Analyze the efficiency issue of both methods (*i.e.*, which version is faster). **Only an answer receives zero point.**

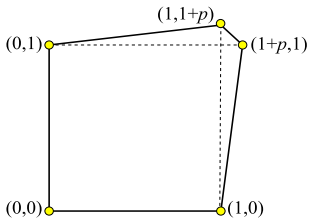
ii. **[10 points]** What if the i in the exponent of $a_i x^i$ is declared to be of type `REAL` in Fortran 90 and used as $a_2 x^{2.0}$ and $a_5 x^{5.0}$? **Provide your answer first and then justify it with a convincing argument. A vague answer receives a zero.**

(c) **[20 points]** Given a convex pentagon $ABCDE$, the intersection points of its "opposite" sides yields a larger or "outer" pentagon, the intersection points of its "diagonals" yields a second smaller or "inner" pentagon. See Figure (a) below. Given a pentagon P, the operation from P to its larger or "outer" one is denoted by $\texttt{out}(P)$, and the operation from P to its smaller or "inner" one is denoted by $\texttt{in}(P)$. In Figure (a), $abcde = \texttt{out}(ABCDE)$ and $ABCDE = \texttt{in}(abcde)$, and, hence, $ABCDE = \texttt{in}(\texttt{out}(ABCDE))$ and $abcde = \texttt{out}(\texttt{in}(abcde))$. However, this relation only holds in theory, and can fail miserably with finite precision arithmetic.

Consider a pentagon with vertices $(0, 0)$, $(1, 0)$, $(1+p, 1)$, $(1, 1+p)$ and $(0, 1)$ as shown in Figure (b) below, where p is a very small number, say $p = 0.0005$ or even $p = 0.000001$. What are the reasons that can cause $P = \texttt{in}(\texttt{out}(P))$ and $P = \texttt{out}(\texttt{in}(P))$ to fail, where P is a convex pentagon? **State your findings clearly, and provide a convincing argument for each reason. A vague and off-point reason receives no credits.**



(a)



(b)

3. **Nonlinear Equations:**

   (a) **[15 points]** Use Newton's method to solve $f(x) = x^2 - x - 1$ with initial guess $x_0 = 3$; however, you only need to go for three iterations (*i.e.*, computing only x_1, x_2 and x_3). **You should show all computation steps and intermediate results, and clearly state which value is for which item. Only providing three values for x_1, x_2 and x_3 receive zero point.**

(b) **[15 points]** Use the simple fixed-point method to solve $f(x) = e^{-x} - x$ with initial value $x_0 = 1$; however, you only need to go for three iterations (*i.e.*, computing only x_1, x_2 and x_3). **You should show all computation steps and intermediate results, and clearly state which value is for which item. Only providing three values for x_1, x_2 and x_3 receive zero point.**

(c) **[15 points]** The secant method requires two initial points a and b and their function values $f(a)$ and $f(b)$. Then, it uses $(a, f(a))$ and $(b, f(b))$ to compute a new point c and its function value $f(c)$. The next iteration will use $(b, f(b))$ and $(c, f(c))$. The following shows the output of using the secant method to solve $f(x) = x^{10} - 1$ with initial points $a = 0$ and $b = 1.5$. As you can see from the table, the third iteration failed to compute c and $f(c)$ with a floating exception. Answer the following two questions with full details and convincing arguments. Otherwise, you will receive zero point. **Question 1:** What is the cause of this "floating exception" error? **Question 2:** What is the reason(s) the may lead to this problem?

Iteration	a	$f(a)$	b	$f(b)$	c	$f(c)$
1	0.0	-1.0	1.5	56.66504	0.026012301	-1.0
2	1.5	56.66504	0.026012301	-1.0	0.051573503	-1.0
3	0.026012301	-1.0	0.051573503	-1.0	failed	failed

Grade Report

Problem		*Possible*	*You Received*
1	a	5	
	b	5	
	c	5	
	d	5	
2	a(i)	10	
	a(ii)	20	
	a(iii)	10	
	b(i)	15	
	b(ii)	10	
	c	20	
3	a	15	
	b	15	
	c	15	
Total		150	