February 28, 2018 Math 541 Name Key

Be sure to show all your work.

1. (20pts) Consider the function given by

$$f(x) = e^{-2x} + 2\sin(x) - 1.$$

a. Write a Taylor polynomial of degree 3, $P_3(x)$, and the remainder term $R_3(x)$ about $x_0 = 0$.

$f(x) = \left(1 - 2x + \frac{4x^2}{2} - \frac{8x^3}{6}\right) + 2\left(x - \frac{x^3}{6}\right) - 1 + R_3(x)$

$= 2x^2 - \frac{5}{3}x^3 + R_3(x)$, so $P_3(x) = 2x^2 - \frac{5}{3}x^3$ (4)

$R_3(x) = \frac{f^{(4)}(\xi(x))}{4!}x^4 = \frac{16e^{-2\xi(x)} + 2\sin(\xi(x))}{24}x^4$ (3)

$\xi(x) \in [a, b]$ with $0 \in [a, b]$, $x \in [a, b]$

b. Use the remainder term to get an upper bound on the error in the approximation, $P_3(x)$, for $x \in [0, 1]$. Briefly explain how you selected your upper bound. Find the absolute and relative error between $f(1)$ and $P_3(1)$.

Bounds of 1 for both e^{-2x} + $\sin(x)$.

$|R_3(x)| = \left|\frac{16e^{-2\xi(x)} + 2\sin(\xi(x))}{24}x^4\right| \le \frac{18}{24} = \frac{3}{4}$ for $x \in [0, 1]$ (4)

(optimal is $16/24 = 2/3$ as $f^{(4)}(x)$ monotonic.)

$f(1) = 0.81828$ $P_3(1) = 1/3 = 0.33333$

$|f(1) - P_3(1)| \approx 0.48494$ abs. error (2)

$\frac{|f(1) - P_3(1)|}{|f(1)|} = 0.59264$ rel. error (2)

c. One method to find the roots of $f(x)$ (solve $f(x) = 0$) is the application of Newton's method. Explicitly write the formula with the appropriate trigonometric and exponential functions, including x_n.

$x_{n+1} = x_n - \frac{e^{-2x_n} + 2\sin(x_n) - 1}{-2e^{-2x_n} + 2\cos(x_n)}$ (3)

1. d. Continuing from the first page with

$$f(x) = e^{-2x} + 2\sin(x) - 1,$$

we see that one root of $f(x)$ is $x_* = 0$. If we let $x_0 = 0.2$ and apply Newton's method, then the convergence to x_* is (circle one):

2 **Sublinear** **Linear** **Superlinear** **Quadratic** **Other**

Without actually performing any iterations we know that Newton's method will take approximately:

2 15 **iterations** to converge to x_* to a tolerance of 10^{-5}. Explain.

2 0 is a double root, so Newton convergence is linear, taking ~3.3 iterations/decimal pt.

e. Another root is near $x_0 = 2.5$, and it is a simple root. Applying Newton's method we obtain $x_* = 2.62104491$. In this case, the convergence to x_* is (circle one):

2 **Sublinear** **Linear** **Superlinear** **Quadratic** **Other**

Without actually performing any iterations we know that Newton's method will take approximately:

2 4 **iterations** to converge to x_* to a tolerance of 10^{-8}. Explain.

2 x_* is a single root, so Newton convergence is quadratic, doubling the number of accurate digits w/ each iteration,

2. (15pts) Consider the second order difference equation that satisfies :

$$y_n = y_{n-1} + 2\,y_{n-2}, \qquad n \geq 2.$$

a. Assume $y_0 = -1$, $y_1 = 2$, and find y_2, y_3, and y_4.

3

$y_2 = 0$ $\quad y_3 = 4$ $\quad y_4 = 4$

b. You want to design a MatLab function using a `while` statement that accepts two initial conditions, `s0` and `s1`, and finds the first iteration when `yn` exceeds `M`. Below is the outline of a MatLab code, and you fill in the few lines needed to find `n` and `yn`.

```
1  function [n,yn] = deqn(s0,s1,M)
2  % Outline for Problem 2b
3  y0 = s0; y1 = s1;
4  n = 2;
5  yn = y1 + 2*y0;
6  while % Fill in the details
7      % Several lines of code
8  end
9  fprintf('n = %d, yn = %d \n',n,yn)
10 end
```

```
while (yn <= M)
  y0 = y1;
  y1 = yn;
  fprintf('n = %d, yn = %d \n', n, yn)
yn = y1 + 2*y0;
 n = n + 1;
```

6

c. This MatLab function accepts an arbitrary number of iterations `N` and uses an `if` statement and the `mod` function to add together any values `yn` that are divisible by **4**. (Note that `mod(y,m)` returns the remainder after division of `y` by `m`.) Below is the outline of a MatLab code, and you fill in the few lines needed to find the sum, `sum`, of the values of `yn`,which are divisible by **4**.

```
1  function sum = deqn2(N)
2  % Outline for Problem 2c
3  y0 = -1; y1 = 2;
4  sum = 0;
5  for n = 2:N
6      yn = y1 + 2*y0;
7      % Insert a few lines of code
8      % Hint: Use an if statement with the mod function
9  end
10 end
```

```
if (mod(yn,4) == 0)
    sum = sum + yn;
end
y0 = y1;
y1 = yn;
```

6

3. (30pts) Consider the function:

$$f(x) = x^3 - 41.$$

a. Let $a = 3$ and $b = 4$. Perform **3** steps of the bisection method to approximate the positive root of $f(x)$. (Note: $m_0 = 3.5$, and you must iterate to m_3.) Clearly, show how you arrive at each of the steps using this technique, *i.e.*, provide any function evaluations needed in the procedure.

$f(3) = -14$ $\quad f(4) = 23$ $\quad m_0 = 3.5$ $\quad f(m_0) = 1.875$

1. $a = 3$ $\quad b = 3.5$ $\quad m_1 = 3.25$ $\quad f(m_1) = -6.672$

2. $a = 3.25$ $\quad b = 3.5$ $\quad m_2 = 3.375$ $\quad f(m_2) = -2.557$

3. $a = 3.375$ $\quad b = 3.5$ $\quad m_3 = 3.4375$ $\quad f(m_3) = -0.3811$

b. Write the secant method formula using this $f(x)$ for finding its root. (Do **NOT** just write $f(x_n)$.) Let $x_0 = 3$ and $x_1 = 4$. Iterate your formula twice with the secant method to approximate the positive root of $f(x)$. Clearly, show your calculations.

$$x_{n+1} = x_n - \frac{(x_n^3 - 41)(x_n - x_{n-1})}{(x_n^3 - 41) - (x_{n-1}^3 - 41)}$$

$$x_2 = 4 - \frac{(64 - 41)(1)}{(64 - 41) - (27 - 41)} = 3.378378$$

$$x_3 = 3.378378 - \frac{(x_2^3 - 41)(3.378378 - 4)}{(x_2^3 - 41) - (64 - 41)} = 3.438023$$

3. **Continued**

c. Write the Newton's method formula using this $f(x)$ for finding its root. (Do **NOT** just write $f(x_n)$.) Let $x_0 = 3$ and iterate your formula twice with the Newton's method to approximate the positive root of $f(x)$. Clearly, show your calculations.

$$x_{n+1} = x_n - \frac{x_n^3 - 41}{3x_n^2}$$

7

$$x_1 = 3 - \frac{27-41}{27} = 3.51852$$

$$x_2 = 3.51852 - \frac{x_1^3 - 41}{3x_1^2} = 3.44961$$

d. Briefly discuss the rate of convergence for each of the three methods using your knowledge about the theoretical convergence of these methods. Find the exact positive root of $f(x)$ and determine which iteration above comes closest to the root. Give the absolute error of the best approximation for each method.

$x_* = 41^{1/3} = 3.448217$

8

Bisection has linear convergence $|m_3 - x_*| \simeq 0.0107$

Secant has superlinear convergence $|x_3 - x_*| = 0.0102$

Newton's has quadratic convergene $|x_2 - x_*| = 0.0014$

Clearly Newton's method is closest with 2 iterations.