Name:__________________________________________

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Problem 1:

Approximate the derivative of \( f(x) = \frac{x}{\ln x} \), at \( x = 2 \) using a backward finite-divided-difference approximation and let \( \Delta x = h = 0.1 \). What is the true relative error? (10pts)

\[
f'(2) \approx \frac{f(2) - f(2 - 0.1)}{(0.1)} = \frac{2 - 1.9}{\ln(2) - \ln(1.9)} = -0.748
\]

The true derivative is \( f'(x) = \frac{\ln x - 1}{(\ln x)^2} \), which at \( x=0 \) is equal to -0.639.

Therefore, the true relative error is:

\[
\varepsilon_t = \frac{-0.639 - (-0.748)}{-0.639} = -17.1\%
\]

Problem 2

Solve the following ODE using the classical fourth order RK method. Solve at \( x=0.5 \) with a step size of 0.5. The initial condition is \( y(0)=1 \) (15pts)

\[
\frac{dy}{dx} = xy^2 - 1.2y
\]

\[
k_1 = f(x, y) = f(0.1) = 0.5 - 1.2 = -1.2
\]

\[
k_2 = f\left(x + \frac{h}{2}, y + \frac{k_1}{2}, y + \frac{k_2}{2}\right) = f\left(0.5, 1 - 1.2 \frac{0.5}{2}\right) = 0.7(0.25^2 - 1.2) = -0.79625
\]

\[
k_3 = f\left(x + \frac{h}{2}, y + \frac{k_3}{2}, y + \frac{k_2}{2}\right) = f\left(0.5, 1 - 0.79625 \frac{0.5}{2}\right) = 0.8009(0.25^2 - 1.2) = -0.911066
\]

\[
k_4 = f\left(x + h, y + k_3, y + k_2\right) = f(0.5, 1 - 0.911066(0.5)) = 0.544467(0.5^2 - 1.2) = -0.517243
\]

\[
f(0.5) = f(0) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}h
\]

\[
= 1 + \frac{-1.2 + 2(-0.79625) + 2(-0.911066) + (-0.517243)}{6}(0.5)
\]

\[
= 0.572344
\]
Problem 3:
Set up a Crank-Nicholson matrix to solve the following PDE for the first time step (20 pts)

\[
\begin{align*}
\frac{\partial^2 u}{\partial x^2} + 4 \frac{\partial u}{\partial x} &= \frac{\partial u}{\partial t} \\
\text{Boundary conditions} \quad &u(0,t) = 0 \\
\text{Initial conditions} \quad &u(1,t) = 1 \\
&u(x,0) = 0 \\
&0 \leq x \leq 1
\end{align*}
\]

Use \( \Delta x = 0.25, \Delta t = 0.125 \)

\[
\frac{\partial^2 u}{\partial x^2} = \frac{1}{2} \left[ \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{\Delta x^2} + \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \right]
\]

\[
\frac{\partial u}{\partial x} = \frac{1}{2} \left[ \frac{u_{i+1,j+1} - u_{i-1,j+1}}{2\Delta x} + \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \right]
\]

\[
\frac{\partial u}{\partial t} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t}
\]

Substituting into differential equation, we get:

\[
\frac{1}{2} \left[ \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{\Delta x^2} + \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \right] + 4 \frac{1}{2} \left[ \frac{u_{i+1,j+1} - u_{i-1,j+1}}{2\Delta x} + \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \right] = \frac{u_{i,j+1} - u_{i,j}}{\Delta t}
\]

Rearrange terms:

\[
(1 - 2\Delta x)u_{i,j+1} + \left( -2 + \frac{2\Delta x^2}{\Delta t} \right)u_{i,j+1} + (1 + 2\Delta x)u_{i,j+1} = (-1 + 2\Delta x)u_{i,j+1} + \left( 2 - \frac{2\Delta x^2}{\Delta t} \right)u_{i,j+1} + (-1 - 2\Delta x)u_{i,j+1}
\]

Substitute \( \Delta x = 0.25, \Delta t = 0.125 \)

\[
\frac{u_{i,j+1}}{2} - 3u_{i,j+1} + 1.5u_{i+1,j+1} = -\frac{u_{i-1,j+1}}{2} + u_{i,j} - 1.5u_{i+1,j}
\]

At boundary condition \( x = 0, u = 0 \)

\[-3u_{i,j+1} + 1.5u_{2,j+1} = +u_{i,j} - 1.5u_{i,j} \]

At boundary condition \( x = 1, u = 1 \)

\[-3u_{i,j+1} + 1.5u_{3,j+1} = -\frac{u_{2,j+1}}{2} + u_{3,j} - 3 \]

Putting everything together, we get the following matrix for the first time step

\[
\begin{bmatrix}
-3 & 1.5 \\
0.5 & -3 & 1.5 \\
0.5 & -3 & 1.5
\end{bmatrix}
\begin{bmatrix}
\text{\( u_{i,j+1} \)} \\
\text{\( u_{2,j+1} \)} \\
\text{\( u_{3,j+1} \)}
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
-3
\end{bmatrix}
\]
Problem 4:

Given the following data, use least-squares regression to fit a straight line.

(15 pts)

<table>
<thead>
<tr>
<th>X</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>16</td>
<td>25</td>
<td>32</td>
<td>33</td>
<td>38</td>
<td>36</td>
<td>39</td>
<td>40</td>
<td>42</td>
<td>42</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\sum x &= 275 \\
\sum y &= 343 \\
\sum xy &= 10455 \\
\sum x^2 &= 9625 \\
a_1 &= \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} = \frac{10(10455) - 275(343)}{10(9625) - (275)^2} = 0.495758 \\
a_0 &= \frac{\sum y}{n} - a_1 \frac{\sum x}{n} = \frac{343}{10} - 0.495758 \frac{275}{10} = 20.66667
\end{align*}
\]

Best fit straight line is 20.666667 + 0.495758x
Problem 5

Using the data from the first three points in Problem 4, set up the system of equations to determine the coefficient of functions for quadratic splines between those three points? (15 pts)

Condition 1: Match interior points
\[100a_1 + 10b_1 + c_1 = 25\]
\[100a_2 + 10b_2 + c_2 = 25\]

Condition 2: Match end points
\[25a_1 + 5b_1 + c_1 = 16\]
\[225a_2 + 15b_2 + c_2 = 32\]

Condition 3: Match slopes
\[20a_1 + b_1 - 20a_2 - b_2 = 0\]

Condition 4: Set \(a_1=0\)

Put these together and get the following linear system.

\[
\begin{bmatrix}
10 & 1 & 0 & 0 & 0 \\
0 & 0 & 100 & 10 & 1 \\
5 & 1 & 0 & 0 & 0 \\
0 & 0 & 225 & 15 & 1 \\
1 & 0 & -20 & -1 & 0
\end{bmatrix}
\begin{bmatrix}
b_1 \\
c_1 \\
a_2 \\
b_2 \\
c_2
\end{bmatrix}
= \begin{bmatrix}
25 \\
25 \\
16 \\
32 \\
0
\end{bmatrix}
\]
Problem 6

The following is a predictor corrector form of Euler’s method.

\[ y_{n+1}^0 = y_n + \Delta t f\left(y_n, t_n\right) \]
\[ y_{n+1}^1 = y_n + \Delta t f\left(y_{n+1}^0, t_{n+1}\right) \]

where \( \frac{dy}{dt} = f\left(y, t\right) \)

a) Using this method with \( \frac{dy}{dt} = -y \), solve for \( y(1.5) \) where \( y(0)=1 \) and \( \Delta t = 0.5 \). Show the true error as well. (7 pts)

\[ y_1^0 = y_0 + \Delta t f\left(y_0, t_0\right) = 1 + 0.5 f\left(1,0\right) = 1 + 0.5(-1) = 0.5 \]
\[ y_1 = y_0 + \Delta t f\left(y_{1}^0, t_{1}\right) = 1 + 0.5 f\left(0,0.5\right) = 1 + 0.5(-0.5) = 0.75 \]
\[ y_2^0 = y_1 + \Delta t f\left(y_1, t_1\right) = 0.75 + 0.5 f\left(0.75,0.5\right) = 0.75 + 0.5(-0.75) = 0.375 \]
\[ y_2 = y_1 + \Delta t f\left(y_{2}^0, t_{2}\right) = 0.75 + 0.5 f\left(0.375,1.0\right) = 0.75 + 0.5(-0.375) = 0.5625 \]
\[ y_3^0 = y_2 + \Delta t f\left(y_2, t_2\right) = 0.5625 + 0.5 f\left(0.5625,1.0\right) = 0.5625 + 0.5(-0.5625) = 0.28125 \]
\[ y_3 = y_2 + \Delta t f\left(y_{3}^0, t_{3}\right) = 0.5625 + 0.5 f\left(0.28125,1.5\right) = 0.5625 + 0.5(-0.28125) = 0.421875 \]

The analytical solution is \( y(t) = e^{-t} \). Therefore, \( y(1.5) = 0.223130 \). The true error is
\[ \varepsilon_t = \frac{0.223130 - 0.421875}{0.223130} = 89.1\% \]

b) What are the conditions on stability for this method? (8 pts)

\[ y_{n+1}^0 = y_n + \Delta t f\left(y_n, t_n\right) \]
\[ = y_n + \Delta t(-y_n) \]
\[ y_{n+1}^1 = y_n + \Delta t f\left(y_{n+1}^0, t_{n+1}\right) \]
\[ = y_n + \Delta t f\left(y_n + \Delta t(-y_n)\right) \]
\[ = y_n + \Delta t(-y_n + \Delta t(y_n)) \]
\[ = y_n\left(1 - \Delta t + \Delta t^2\right) \]

The method is stable as long as \( |1 - \Delta t + \Delta t^2| < 1 \) which is true for \( 0 < |\Delta t| < 1 \).
Problem 7
The following problems are True/False questions. (10 pts)

T/F

Gauss-Seidel is always guaranteed to converge _F_
If $\lambda$ is not selected carefully, Gauss-Seidel may not converge

Cubic splines can be determined by solving an $n-1$x$n-1$ matrix _T_
where $n$ is the number of intervals
Using the simplification shown in Box 18.3 (pp. 502 of book), you can
solve for cubic splines with just a $n-1$x$n-1$ matrix instead of a $4n$x$4n$
matrix as you would expect.

More corrector iterations of Heun’s method will increase the _F_
accuracy of the solution
More corrector iterations will only converge on an error for that step,
not necessarily improve the solution or reduce the error for that step

Euler’s method is a form of the Runge-Kutta method _T_
Euler’s method is a first-order RK method

Romberg integration relies on three previous estimates to _F_
arrive at a new more accurate estimate
Romberg integration uses two previous estimates to
arrive at a new more accurate estimate