1. (20 points)
   (a) Construct a Newton-Cotes formula of the form
   \[ \int_0^1 f(x) \ln(1/x) \, dx = \alpha_1 f(0) + \alpha_2 f'(1) + E(f) \]
   which integrates \( f(x) = 1 \) and \( f(x) = x \) exactly. Find an expression for the error \( E(f) \) in terms of a derivative of \( f \).

**Solution:** I will use the method of undetermined coefficients. Plug in \( f(x) = 1 \) and \( f(x) = x \) to get
\[ \int_0^1 \ln(1/x) \, dx = 1 = \alpha_1, \]
and
\[ \int_0^1 x \ln(1/x) \, dx = 1/4 = \alpha_2 f'(1) = \alpha_2. \]
So \( \alpha_1 = 1 \) and \( \alpha_2 = 1/4 \). The error is based on interpolation theory. Given \( f \), let \( p \) be the linear interpolation polynomial with \( p(0) = f(0) \) and \( p'(1) = f'(1) \). Then
\( f(x) = p(x) + E(x) \) where
\[ E(x) = \frac{f''(\xi)x(x-1)^2}{2} \]
The maximum of \( x(x-1) \) is attained at \( x = 1/3 \) and the value is \( 4/27 \). Hence the bound for the error is
\[ \frac{2\|f''\|_{\infty}}{27}. \]
The other thing you have to do is verify the formula for \( E(x) \). This is a Hermite deal. Just apply the usual Rolle’s theorem argument to
\[ f(t) - p(t) = \frac{(f(x) - p(x))t(1-t)}{x(1-x)^2} \]
for \( 0 < x < 1 \).
There are other reasonable ways to estimate the error, and some of you did it a different way.
(b) Find the weight and node for one point Gaussian quadratures on \([0, 1]\) with the weight function \(w(x) = \ln(1/x)\).

**Solution:** The first thing to do is find the orthogonal polynomial of degree 1. Let \(p(x) = x + a\). We want
\[
0 = \int_0^1 p(x) \ln(1/x) \, dx = 1/4 + a.
\]
So \(a = -1/4\). The node is the root of \(p\): \(x_1 = 1/4\). Getting the weights is very simple. I have to integrate constant functions exactly, so
\[
\int_0^1 \ln(1/x) \, dx = 1 = w_1.
\]

2. (20 points) Do problem 56 on page 204.

**Solution:** I will use the Lagrange form. It’s gonna be

(a) \(p(x, y) = g(0, 0)(1 - x)(1 - y) + g(0, 1)(1 - x)y + g(1, 0)x(1 - y) + g(1, 1)xy\),

which is really just the two-variable Lagrange form.

(b) \[I = \frac{(g(0, 0) + g(1, 0) + g(0, 1) + g(1, 1))}{4}\]

It’s the trapezoid rule!

(c) \(w_{ij} = w_i w_j\)

where \(w_i\) is the \(i\)th weight of the composite trapezoid rule.

3. (20 points) Compute the local truncation error, principal error function, absolute stability region, and order for this method:

\[u_{n+1} = u_n + \frac{h}{2} \left( f(t_n, u_n) + f \left( t_n + \frac{h}{2}, u_n + \frac{h}{2} f(t_n, u_n) \right) \right).\]
Solution: This is just two steps of forward Euler with step size \( h/2 \). So the order is 1, the stability region is
\[
\{ z \mid |z + 2| < 2 \},
\]
and the local truncation error is just Euler with \( h/2 \) substituted for \( h \). I use equation (2.3) on page 274 to get the principal error function and truncation error:
\[
\tau(t, y) = -\left( f_t + f f_y \right)/4 = -\frac{u''(\xi)}{4}.
\]
You could also derive all this from scratch.

4. (20 points) Do problem 5 on page 320.

**Solution:** The order of the first equation is three and the second is two. So I will have a first order system of five equations. I will set
\[
y_1 = u, \quad y_2 = u', \quad y_3 = u'', \quad y_4 = v, \quad y_5 = v'.
\]
(a) Then the equations are
\[
\begin{align*}
y_1' &= y_2 \\
y_2' &= y_3 \\
y_3' &= t^2 y_1 y_3 - y_1 y_5 \\
y_4' &= y_5 \\
y_5' &= t y_1 y_5 + 4 y_2
\end{align*}
\]
(b) Since
\[
f(t, y) = \begin{pmatrix}
y_2 \\
y_3 \\
t^2 y_1 y_3 - y_1 y_5 \\
y_5 \\
y_4 y_5 + 4 y_2
\end{pmatrix},
\]
it’s a simple computation to get
\[
f_y = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
t^2 y_3 - y_5 & 0 & t^2 y_1 & 0 & -y_1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 4 & 0 & t y_4 & t y_5
\end{pmatrix}
\]
**Solution:** It’s easy to show, and we did it, that $\|f_y\|$ is a good upper estimate for the Lipschitz constant for $f$. So I will estimate $f_y$ in a reasonable way and compute the norm of the upper bound. If $\|y\| \leq 1$, then $-1 \leq y_i \leq 1$ and so $f_y$ is component-wise less than

$$A = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
2 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 4 & 0 & 1 & 1 \\
\end{pmatrix}$$

$\|A\|_\infty$ is the maximum row sum, so $\|A\|_\infty = 6$. $\|A\|_1$ is the maximum column sum, so $\|A\|_1 = 5$. The $l^2$ norm is the maximum singular value. I will use MATLAB for that one, and I got $\text{norm}(A) = 4.5547$.

5. (20 points) **Programming. Your matlab code, plots, and discussion must be turned in electronically.** Write an Romberg integration code. Your code must accept as its input the interval of integration $[a, b]$, the function to be integrated $f$, the maximum number of subintervals $n$ (so the smallest mesh size will be $h = (b - a)/n$), and the desired order of accuracy. The output will be the approximation $\int_a^b f(x) \, dx$ to the desired order. You may want to limit the maximum order to something reasonable (like 10). Validate your code with a well designed (and well explained) grid refinement study. The design of that study is part of the problem. The code must be well documented and easy for me to understand.
**Solution:** I wrote a MATLAB function `romberg.m` which does the job. I tested it with `romtest.m` using

\[ f(x) = 5e^{\sin(5x)} \cos(5x) \]

on the interval \([0, 10]\). Using a much shorter interval didn’t let me see orders higher than 4. As it is, anything with more than 5 levels was too difficult for me to measure. My test code has a print function that will do a LaTeX table, which is what I used for the tables below. The first table is the errors. The second is the ratios of errors, which shows that the orders are correct until the grid is refined to the point that the integral is so accurate that I can’t measure the order of accuracy.

<table>
<thead>
<tr>
<th>h</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/64</td>
<td>5.14e-03</td>
</tr>
<tr>
<td>1/128</td>
<td>1.10e-03 2.41e-04</td>
</tr>
<tr>
<td>1/256</td>
<td>2.64e-04 1.55e-05 4.88e-07</td>
</tr>
<tr>
<td>1/512</td>
<td>6.53e-05 9.58e-07 1.36e-08 2.16e-08</td>
</tr>
<tr>
<td>1/1024</td>
<td>1.63e-05 5.96e-08 2.23e-10 1.08e-11 7.38e-11</td>
</tr>
<tr>
<td>1/2048</td>
<td>4.07e-06 3.72e-09 3.52e-12 3.59e-14 6.19e-15 6.59e-14</td>
</tr>
</tbody>
</table>

The table of ratios is pretty good. The ratios are supposed to be 4, 16, 64, 256, and 1024. I’m getting close in the last row until I run into a wall at the final column.

<table>
<thead>
<tr>
<th>h</th>
<th>Error Ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/128</td>
<td>4.66e+00</td>
</tr>
<tr>
<td>1/256</td>
<td>4.18e+00 1.55e+01</td>
</tr>
<tr>
<td>1/512</td>
<td>4.04e+00 1.62e+01 3.58e+01</td>
</tr>
<tr>
<td>1/1024</td>
<td>4.01e+00 1.61e+01 6.10e+01 2.00e+03</td>
</tr>
<tr>
<td>1/2048</td>
<td>4.00e+00 1.60e+01 6.34e+01 3.00e+02 1.19e+04</td>
</tr>
</tbody>
</table>