1. (10 points) How are the Adams and BDF methods constructed? I’m not asking for the formulae, but only for the general idea of how they’re derived.

**Solution:** Adams methods interpolate \( f \) at the points \((t_j, u_j)\) and then use the interpolating polynomial for \( y'(t) \), integrate that to obtain the method.  
BDF methods interpolate \( y \) at the points \((t_j, u_j)\), differentiate the interpolating polynomial at \( t_{n+1} \) and equate that to \( f(t_{n+1}, u_{n+1}) \) to obtain the method.

2. (30 points)
   (a) What is a PECE method?
   (b) Take one step of the AB1-AM1 (forward Euler - backward Euler) predictor-corrector pair for the equation 
   \[
   y' = \lambda y.
   \]
   (c) What is the stability region for the AB1-AM1 PECE method?

**Solution:**
   (a) A PECE method takes one step of an explicit method (predictor) and the one step of successive substitution for an implicit method (corrector).
   (b) The predictor is
   \[
   \hat{u}_{n+1} = u_n + hf_n = u_n + \lambda hu_n = (1 + \lambda h)u_n.
   \]
   So the corrector is
   \[
   u_{n+1} = u_n + hf(\hat{u}_{n+1}) = u_n + h\lambda((1 + \lambda h)u_n = (1 + h\lambda + (h\lambda)^2)u_n.
   \]
   (c) So the stability region is the set
   \[
   \{h\lambda \mid |1 + h\lambda + (h\lambda)^2| \leq 1 \}.
   \]

3. (20 points) Consider the the multistep method
   \[
   u_n = -4u_{n-1} + 5u_{n-2} + 4hf(u_{n-1}) + 2hf(u_{n-2}).
   \]
   (a) Is this method consistent? How do you know?
   (b) Is this method stable? How do you know?

**Solution:** Here we have \( a_0 = -4, a_1 = 5, b_0 = 4 \) and \( b_1 = 2. \)
   (a) The method is consistent, because \( a_0 + a_1 = -4 + 5 = 1 \) and \( -a_1 + b_0 + b_1 = -5 + 4 + 2 = 1 \).
   (b) For this method
   \[
   \rho(r) = r^2 + 4r - 5 = (r - 1)(r + 5)
   \]
   The roots are \( r = 1 \) and \( r = -5 \), so the method is not stable.
Take-home part:
1. Write a PECE code using these two methods. Generate the starting points with RK4.

**Solution:** My PECE code is `adams.m`. It does the job and is very careful to only evaluate \( f \) the minimal number of times.

2. Demonstrate that your code works with a grid-refinement study and explain how you designed that study.

**Solution:** My grid refinement study is `atest.m`. My test was an easy one:

\[
y' = -y, \quad y(0) = 1.
\]

I did nine levels, reducing \( h \) by a factor of two each time. This was enough to give me the results in the table below, which shows the reduction by a factor of 16 that I’d expect from a 4th order method. I’m tabulating rations of maximum errors.

<table>
<thead>
<tr>
<th>( h )</th>
<th>( | e_h | / | e_{h/2} | )</th>
<th>( h )</th>
<th>( | e_h | / | e_{h/2} | )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1/4 )</td>
<td>8.52e+00</td>
<td>( 1/1024)</td>
<td>1.62e+01</td>
</tr>
<tr>
<td>( 1/8 )</td>
<td>1.59e+01</td>
<td>( 1/2048)</td>
<td>1.61e+01</td>
</tr>
<tr>
<td>( 1/16 )</td>
<td>1.66e+01</td>
<td>( 1/4096)</td>
<td>1.61e+01</td>
</tr>
<tr>
<td>( 1/32 )</td>
<td>1.64e+01</td>
<td>( 1/8192)</td>
<td>1.56e+01</td>
</tr>
</tbody>
</table>

3. Use your code to solve

\[
y'''(t) = 120 \sqrt{y - 1}; \quad y(0) = 1, y'(0) = 0, y''(0) = 0
\]

on the interval \([0, 2]\). What \( h \) do you need to approximate the exact solution to within an absolute error of \( 10^{-8} \)?

**Solution:** You have to convert this to a first-order system for

\[
\begin{align*}
u &= (u_1, u_2, u_3)^T 
\equiv (y, y', y'')^T
\end{align*}
\]

which is

\[
u' = \begin{pmatrix} u_2 \\ u_3 \\ 120\sqrt{u_1 - 1} \end{pmatrix}
\]

The solution I had in mind was \( y = e^t + 1 \). However the solution \( y \equiv 1 \) is the solution the code finds, and finds it to infinite precision. Most of you called me on this.

What happened? The right side is not Lipschitz continuous at \( y = 1 \), so the theory for existance and uniqueness fails.