This assignment is to help you prepare for the first midterm. It is not to be turned in.

1. Derive the second-order Runge-Kutta method

\[ y(t + h) = y(t) + \frac{1}{2} (F_1 + F_2) \quad \text{for} \quad \begin{cases} F_1 = h f(t, y(t)) \\ F_2 = h f(t + h, y(t) + F_1) \end{cases} \]

**Solution:** With Taylor difference approximations, we expand the true solution at time step \( t_j + h \) using Taylor’s Theorem:

\[
y(t + h) = y(t) + h y'(t) + \frac{h^2}{2} y''(t) + \frac{h^3}{6} y'''(t) + \cdots
\]

In the Taylor difference scheme we could replace the derivatives of \( y \) with partial derivatives of a specific \( f \) w.r.t. to \( t \). Instead, we could use the chain rule to do this for general \( f \). This looks as follows

\[
y'(t) = f \\
y''(t) = f_t + f_y y' = f_t + f_y f \\
y'''(t) = f_{tt} + f_{ty} y' + f (f_{ty} + f_{yy} y') + f_y (f_t + f_y y') \\
= f_{tt} + f_{ty} f + f (f_{ty} + f_{yy} f) + f_y (f_t + f_y f)
\]

Substituting the first two terms into the Taylor series, and truncating after the quadratic term, we have

\[
y(t + h) = y(t) + h f + \frac{h^2}{2} (f_t + f_y y') + \mathcal{O}(h^3)
\]

\[
y(t + h) = y(t) + \frac{1}{2} h f + \frac{1}{2} h [f + h f_t + h f_y y] + \mathcal{O}(h^3)
\]

where here, \( f, f_t \) and \( f_y \) are all functions of \( (t, y) \). Remember that we don’t want to compute the partial derivatives of \( f \) explicitly. Instead we can approximate them using multi-dimensional taylor series.

\[
f(t + h, y + f h) = f(t, y) + h f_t(t, y) + h f_y(t, y) + \mathcal{O}(h^2)
\]

Notice that this is exactly the bracketted expression found above. Substituting in we have

\[
y(t + h) = y(t) + \frac{1}{2} h f + \frac{1}{2} h f(t + h, y + f h) + \mathcal{O}(h^3)
\]

Notice that the \( \mathcal{O}(h^3) \) truncation error includes the original truncation error plus the (also \( \mathcal{O}(h^2) \)) truncation error from approximating the partial derivatives.

Dropping the error term and approximating the exact values of the solution by their approximations at the grid points, we have

\[
y_{j+1} = y_j + \frac{1}{2} h f(t_j, y_j) + \frac{1}{2} h f(t_{j+1}, y_j) + h f(t_j, y_j)
\]

We can simplify the scheme by letting \( F_1 = h f(t_j, y_j) \) and \( F_2 = h f(t_{j+1}, y_j + F_1) \). We then have
\[ y_{j+1} = y_j + \frac{1}{2} (F_1 + F_2) \quad \text{for} \quad \begin{cases} F_1 = h f(t_j, y_j) \\ F_2 = h f(t_{j+1}, y_j + F_1) \end{cases} \]

Since the local truncation error of the scheme is \( O(h^3) \) the global error is \( O(h^2) \). We call this the second-order Runge-Kutta scheme.

2. (a) Discuss what it means for a method to be **convergent**.
   (b) Show that Euler’s method is convergent for the test problem (below) for particular restrictions on the step size \( h \), where here \( \lambda \) is positive and real.

\[
\begin{align*}
 y' &= -\lambda y \\
 y(0) &= 1
\end{align*}
\]

**Solutions:**

(a) A method is **convergent** if the global error goes to zero as \( h \to 0 \) (also called **consistency**) and the method is stable. For stability, it may be necessary to restrict the size of the time step.

(b) The global error in Euler’s method is \( O(h) \) which goes to zero as \( h \to 0 \). To show stability we plug a roundoff error \( \delta_0 \) into the initial condition and see what happens as we time step.

The Forward Euler scheme for this IVP looks like

\[
\begin{align*}
 y_{k+1} &= y_k - h \lambda y_k = (1 - h \lambda) y_k \\
 y_0 &= 1
\end{align*}
\]

Let’s assume that \( \lambda > 0 \).

Now consider the case where we have some small roundoff error, \( \delta_0 \), in the initial condition. So, we really have

\[ u_0 = y_0 + \delta_0 \]

Then after the first time-step we have

\[ u_1 = (y_0 + \delta_0) - h \lambda (y_0 + \delta_0) = (1 - h \lambda) (y_0 + \delta_0) = (1 - h \lambda) u_0 \]

In other words, the slightly perturbed approximation satisfies the same difference equation that the unperturbed approximation satisfies. So we have

\[ y_1 = (1 - h \lambda) y_0 \]

\[ u_1 = (1 - h \lambda) u_0 \]

And subtracting these we see that the roundoff error at the first time-step is given by

\[ \delta_1 = (u_1 - y_1) = (1 - h \lambda) (u_0 - y_0) = (1 - h \lambda) \delta_0 \]

So in fact, the roundoff error itself satisfies the difference equation. Now what happens as we march in time? We have

\[ \delta_2 = (1 - h \lambda) \delta_1 = (1 - h \lambda)^2 \delta_0 \]

and after \( k \) steps of this we have
\[ \delta_k = (1 - h\lambda)^k \delta_0 \]

From this we see that the roundoff error will be damped so long as we have \(|1 - h\lambda| < 1\). This means that the method will be stable so long as

\[
|1 - h\lambda| < 1 \quad \Rightarrow \quad -1 < 1 - h\lambda < 1 \quad \Rightarrow \quad 0 < h\lambda < 2 \quad \Rightarrow \quad h < \frac{2}{\lambda}
\]
3. Consider the following Adams-Bashforth method

\[ y_{k+2} = y_{k+1} + \frac{h}{2} [3f_{k+1} - f_k] \]

(a) Derive the method.
(b) Give the order and local truncation error of the method.
(c) Describe a suitable method for computing the starting values for the method.

Solution:

(a) In an Adams-Bashforth method we interpolate the previous data, and use the interpolant to approximate the integral in the general scheme

\[ y(t + h) = y(t) + \int_t^{t+h} f(s, y(s)) \, ds \]

Here we are using the data at the current time, and one previous time step, so the picture looks like

![Diagram](image)

The interpolating polynomial in this case is linear, and in particular we have

\[ P_1(t) = f_0 \frac{(t - t_1)}{(t_0 - t_1)} + f_1 \frac{(t - t_0)}{(t_1 - t_0)} \]

Integrating from \( t_1 \) to \( t_2 \) we have

\[ \int_{t_1}^{t_2} P_1(\tau) \, d\tau = \int_h^{2h} f_0 \frac{(\tau - h)}{(-h)} + f_1 \frac{(\tau)}{h} \, d\tau = -\frac{h}{2} f_0 + \frac{3h}{2} f_1 \]

Plugging this in to the integral form of the difference scheme we have (for general \( k \))

\[ y_{k+2} = y_{k+1} + \frac{h}{2} [3f_{k+1} - f_k] \]

as desired.

(b) The local truncation error is obtained by integrating the error in the interpolating polynomial. We have

\[ f(t, y(t)) = P_1(t) + \frac{f^{(2)}(\xi)}{2!} (t - t_0)(t - t_1) \]

Integrating again from \( t_1 \) to \( t_2 \) we have

\[ LTE = \int_{t_1}^{t_2} \frac{f^{(2)}(\xi)}{2!} (\tau - t_0)(\tau - t_1) = M \int_h^{2h} \tau (\tau - h) = M \frac{5h^3}{6} \]

so the LTE is \( O(h^3) \) and the global error is \( O(h^2) \).
(c) We need to compute the value of $y_1$ before we have enough previous data to use the two-step AB method. When computing $y_1$ we want to make sure that the method we use has at least as good of LTE as the two-step AB method. Since the LTE of the AB method is $O(h^3)$ we could use, for instance, RK2 to approximate $y_1$. 
4. Consider the following Adams-Moulton method

\[ y_{k+1} = y_k + \frac{h}{2} [f_{k+1} + f_k] \]

(a) Derive the method.
(b) Give the order and local truncation error of the method.
(c) Describe a suitable method for computing the starting values for the method.

**Solution:**

(a) In an Adams-Moulton method we interpolate some previous data and the data that we’re trying to compute on the next time step. The given scheme in this problem is a one-step AM method since it only uses one piece of previously computed data. We then form the (in this case) linear polynomial that interpolates the value of \( f_0 \) and \( f_1 \), and replace the integral below by the integral of the interpolant:

\[ y(t) = y(t_0) + \int_{t_0}^{t_1} f(s, y(s)) \, ds \]

Here we are using the data at the current time, and the future time step, so the picture looks like

The interpolating polynomial in this case is linear, and in particular we have

\[ P_1(t) = f_0 \frac{(t - t_1)}{(t_0 - t_1)} + f_1 \frac{(t - t_0)}{(t_1 - t_0)} \]

Integrating from \( t_0 \) to \( t_1 \) we have

\[ \int_{t_0}^{t_1} P_1(\tau) \, d\tau = \int_{t_0}^{t_1} f_0 \frac{(\tau - h)}{(-h)} + f_1 \frac{(\tau)}{h} \, d\tau = \frac{h}{2} f_0 + \frac{h}{2} f_1 \]

Plugging this in to the integral form of the difference scheme we have (for general \( k \))

\[ y_{k+2} = y_{k+1} + \frac{h}{2} [f_{k+1} + f_k] \]

as desired.

(b) The local truncation error is obtained by integrating the error in the interpolating polynomial. We have (again)

\[ f(t, y(t)) = P_1(t) + \frac{f^{(2)}(\xi)}{2!} \frac{(t - t_0)}{(t_0 - t_1)} \]

Integrating again from \( t_0 \) to \( t_1 \) we have

\[ LTE = \int_{t_0}^{t_1} \frac{f^{(2)}(\xi)}{2!} \frac{(t - t_0)}{(t_0 - t_1)} (t - t_1) = M \int_{t_0}^{t_1} \frac{(t - t_0)}{(t_0 - t_1)} (t - t_1) = -M \frac{h^3}{6} \]

so the LTE is \( O(h^3) \) and the global error is \( O(h^2) \).
(c) This was a (completely on purpose, but not really) trick question. The only piece of old information that the one-step AM method uses is the value at the current time step. Therefore it is not necessary to use another method to compute previous data at the start of the method. In particular, the one-step AM method is the same as the implicit Trapezoid rule.