Section 2.3 (Homework)

#17. \( \cos t \frac{dy}{dt} + (\sin t) y = 1 \)

\[
\frac{1}{\cos t} \frac{dy}{dt} + \frac{\tan t}{\cos t} y dt = \sec t dt
\]

\[
[\text{Solve}] = e^{\tan t dt} = e^{\ln \sec t} = \sec t
\]

\( \sec t \frac{dy}{dt} + \sec t \tan t y dt = \sec^2 t dt \)

\( \frac{d[\sec t \cdot y]}{dt} = \sec^2 t dt \)

\[
[\sec t \cdot y] = \sec^2 t dt
\]

\[
\sec t \cdot y = \int \sec^2 t dt + C,
\]

\[
\sec t \cdot y = \tan t + C,
\]

\[
\frac{1}{\cos t} y = \frac{\sin t}{\cos t} + C,
\]

\[
y = \sin t + C, \cos t
\]
Section 2.4  (Homework)

#35. \( v(0) = v_0 \)

\[
\frac{mdv}{dt} = mg - kv^2 \quad (\div m)
\]

\[
\frac{dv}{dt} = g - \frac{k}{m}v^2 \quad \left( \div \frac{1}{g - \frac{k}{m}v^2} \cdot dt \right)
\]

\[
\frac{dv}{g - \frac{k}{m}v^2} = dt \quad \text{(integrate)}
\]

\[
= t + C_1
\]

\[
v = \sqrt{\frac{mg}{k}} \left( e^\frac{A}{k} - e^{-\frac{A}{k}} \right) \]
\[ t + c, \quad \left( \text{mult. by } \frac{\arg K}{\frac{1}{m}} \right) \]

\[
\ln \left[ \frac{1 - V}{1 + V} \right] = -2 \frac{\frac{1}{m}}{B} t + c_2
\]

(Take exp of both sides)

\[
\frac{1 - V}{1 + V} = e^{-2 \frac{\frac{1}{m}}{B} t}
\]

\[
\begin{align*}
1 - V + V e^{A} & = \sqrt{k} \frac{e^{A}}{1 + \sqrt{g} e^{A}} \\
1 - V & = \sqrt{k} \frac{e^{A}}{1 + \sqrt{g} e^{A}} - V e^{A}
\end{align*}
\]

\[
\frac{1}{1 - e^{A}} = \sqrt{g} \frac{\left( 1 + e^{A} \right)}{1 + 2 \sqrt{g} e^{A} \frac{1}{m} t}
\]

\[
V = \frac{\sqrt{g} \sqrt{m} \left( 1 + e^{A} \right)}{\sqrt{k} \left( 1 - e^{2 \sqrt{g} \frac{1}{m} t} \right)}
\]

Take \[ \lim_{t \to \infty} \quad \lim_{t \to \infty} V = \frac{\sqrt{g} \sqrt{m}}{\sqrt{k}} (1) = \frac{\sqrt{mg}}{k}
\]
2.5 Numerical Methods

Euler's Method: (Method of Tangent Lines)

\[ y' = f(x, y) \]
\[ y(x_0) = y_0 \]

Linearization of \( y(x) \) at \( x = x_0 \):

\[ L(x) = y'(x_0)(x - x_0) + y_0 \]

\[ L(x_1) = y_1 \approx y'(x_0)(x_1 - x_0) + y_0 \]
\[ = y'(x_0)(x_0 + h - x_0) + y_0 \]
\[ = y_0 + y'(x_0)h \]
\[ y_1 \approx y_0 + y'(x_0)h \]
\[ y_2 = y(x_0 + 2h) = y(x_1 + h) \approx y_1 + hf(x_1, y_1) \]
\[ y_{n+1} = y_n + hf(x_n, y_n) \]

where \( x_n = x_0 + nh \)
Example: Apply Euler's method to \( h = 0.01 \)

\[ f(x, y) = 0.2xy \]

\[ y' = 2xy, y(1) = 1 \]

\[ \star y_n = e^{1(x_n^2 - 1)} \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>( x_n )</th>
<th>( y_n )</th>
<th>( y(x) = f(x, y) )</th>
<th>( y_n + hf(x, y) )</th>
<th>True ( y_n ) *</th>
<th>error</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>1.00</td>
<td>( 0.2(1)(1) = 0.2 )</td>
<td>1 + 0.01(2) = 1.002</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1.01</td>
<td>1.002</td>
<td>( 0.2(1.01)(1.002) = 0.20240 )</td>
<td>1.002 + 0.01(0.20482) = 1.00402</td>
<td>1.00201</td>
<td>.00001</td>
<td>.001%</td>
</tr>
<tr>
<td>2</td>
<td>1.02</td>
<td>1.00402</td>
<td>( 0.2(1.02)(1.00402) = 0.20482 )</td>
<td>1.00607</td>
<td>1.00405</td>
<td>.00003</td>
<td></td>
</tr>
</tbody>
</table>

\[ \star y_n = e^{1(x_n^2 - 1)} \]
The Euler Method can be improved by

1. Using a smaller step size $h$.

2. Using the improved Euler method:

   **Euler** 
   \[ y_{n+1} = y_n + hf(x_n, y_n) \]

   **Improved Euler (average slope method)**
   \[ y_{n+1} = y_n + \frac{h}{2} \left( f(x_n, y_n) + f(x_{n+1}, y_{n+1}) \right) \]
Error

Using Taylor’s formula with remainder we can write

\[
y(x) = y(a) + y'(a)(x - a) + \ldots + y^{(k)}(a) \frac{(x - a)^k}{k!} + y^{(k+1)}(c) \frac{(x - a)^k}{(k+1)!}
\]

where c is between x and a. If \( k = 1 \),

\[
y(x_{n+1}) = y(x_n) + y'(x_n) \frac{(x_{n+1} - x_n)}{1} + y''(c) \frac{(x_{n+1} - x_n)^2}{2!}
\]

\[
= y(x_n) + y'(x_n) \cdot h + y''(c) \frac{h^2}{2}
\]

\( y_{n+1} \approx \) Euler’s method

\( x_n < c < x_{n+1} \)

(Euler’s formula + error)
If \( M = \max |y^n(c)|, x_n < c < x_{n+1} \)

Max error = \( M \cdot \frac{h^2}{2!} \) (Local truncation error of \( O(h^2) \))

In general, \( e(h) = \) error in calculation depending on \( h \)

\( e(h) \) is of \( O(h^n) \) if \(|e(h)| \leq Ch^3 \) for some constant \( C \).

The global truncation error will be \( O(h^{n+1}) \).

For the Euler method, the local truncation error is \( O(h^2) \) and the global truncation error is \( O(h) \).

For the improved Euler method, the local truncation error is \( O(h^3) \) and the global truncation error is \( O(h^2) \).
Runge-Kutta Methods:

There are many Runge-Kutta methods which use formulas that must agree with the Taylor polynomial approximation to a function up to a specified degree.

Using Taylor’s formula with remainder

\[ y(x) = y(a) + y'(a)(x-a) + \ldots + y^{(k)}(a) \frac{(x-a)^k}{k!} + y^{(k+1)}(c) \frac{(x-a)^k}{(k+1)!} \]

If \( k = 1 \), the first-order Runge-Kutta is just Euler’s method.
The most commonly used is fourth-order Runge-Kutta. This should agree with a Taylor polynomial of degree 4 and requires the solution of 11 equations containing 13 unknowns. The most commonly used result is listed on page 84.

For fourth-order Runga-Kutta, the local truncation error is $O(h^5)$ and the global truncation error is $O(h^4)$. 
Both methods are available on the Voyage 200 calculator.

\[ \frac{dy}{dt} = t + y \]
\[ \frac{dy}{dt} - y = t \]

\[
\int (e^{-t} \, dt) \quad e^{-t} \, dy - e^{-t} \, dt = e^{-t} \, dt \\
\] (integrate)
\[
\int (e^{-t} \, dt) = e^{-t} \, dt \\
\]

\[
e^{-t} \, y = (-1)e^{-t} + c \\
\]

\[
\Rightarrow \quad y = -t - 1 + ce^t \\
y(0) = 1 \\
1 = 0 - 1 + c(1) \quad \Rightarrow \quad c = 2 \\
\]

\[
y = -t - 1 + 2e^t \\
y(3) = 1.39972
\]
Besides these one-step methods (which use only information about \( y_n \) to compute \( y_{n+1} \), there are multistep methods which use information from several previously computed values.

A numerical method is stable if small changes in the initial condition results in only small changes in the computed solution. Otherwise, the method is unstable.
> restart;

> DE := diff(y(t), t) = 2 \cdot t \cdot y(t);

\[ DE = \frac{d}{dt} y(t) = 0.2 t y(t) \]  \hspace{1cm} (1)

> dsolve[interactive]([DE]);