Approximation Methods
Numerical Analysis

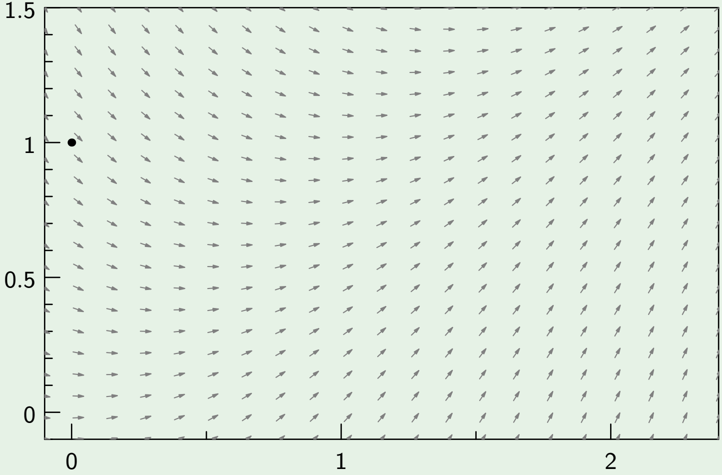
Department of Mathematics

Salt Lake Community College

Example 1

Consider the IVP

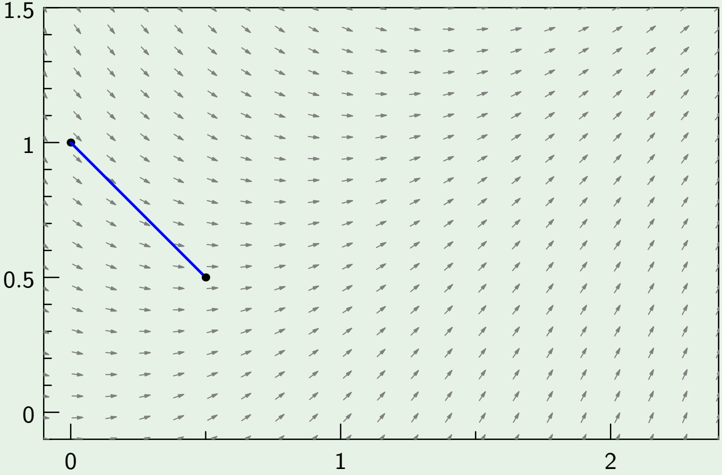
$$y' = t - y, \ \ y(0) = 1$$



Example 1

Consider the IVP

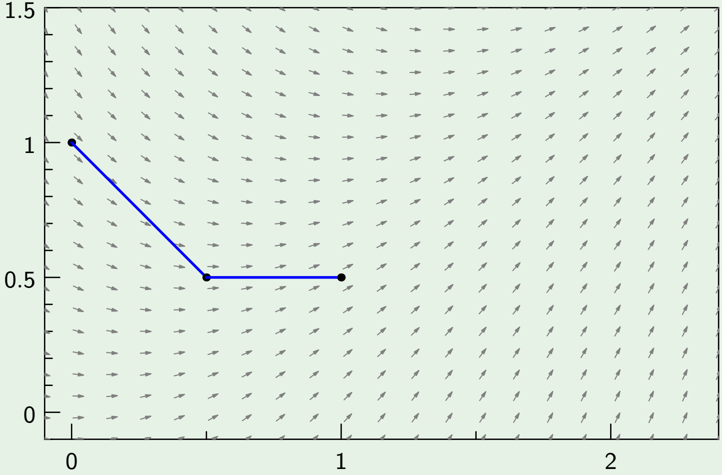
$$y' = t - y, \; y(0) = 1$$



Example 1

Consider the IVP

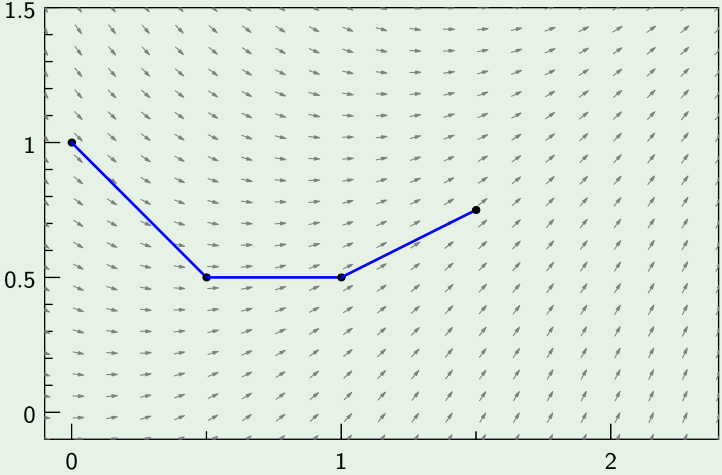
$$y' = t - y, \; y(0) = 1$$



Example 1

Consider the IVP

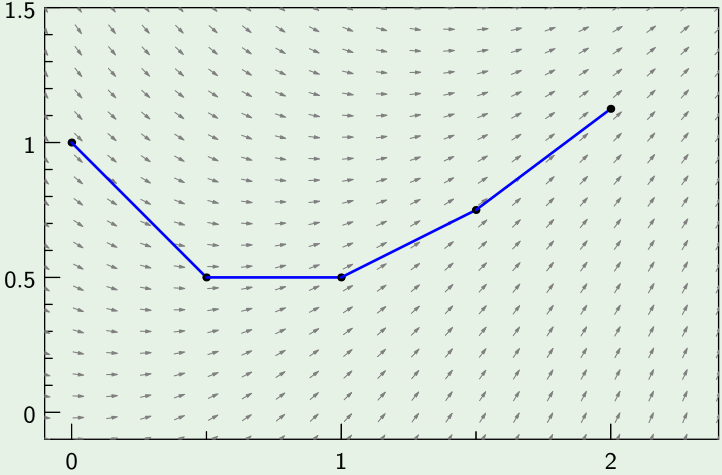
$$y' = t - y, \ y(0) = 1$$



Example 1

Consider the IVP

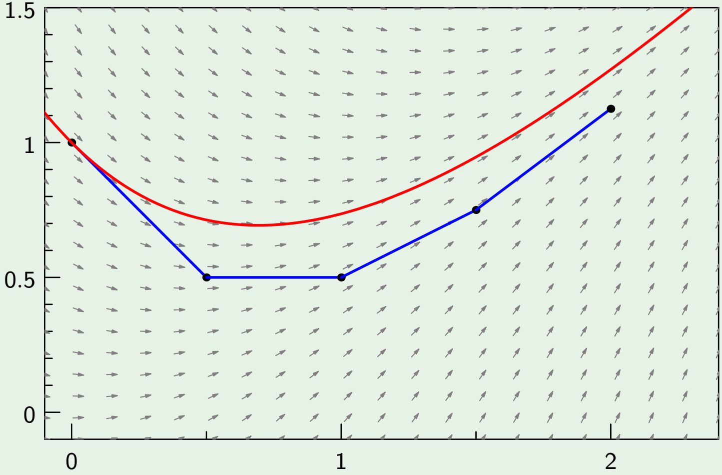
$$y' = t - y, \ \ y(0) = 1$$



Example 1

Consider the IVP

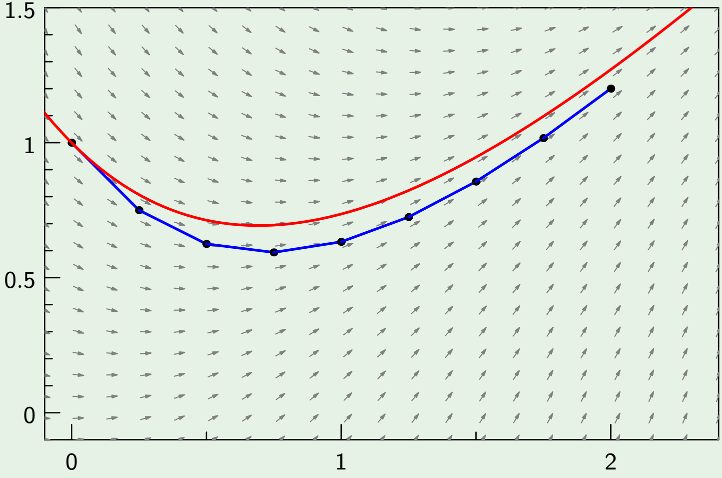
$$y' = t - y,\ \ y(0) = 1$$



Example 1

Consider the IVP

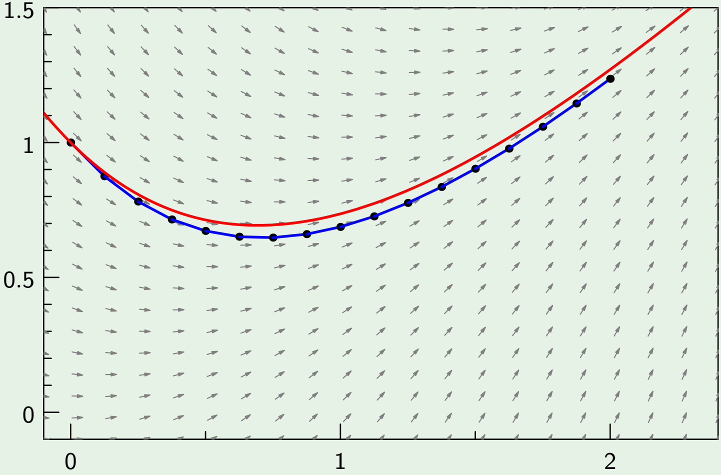
$$y' = t - y, \ \ y(0) = 1$$



Example 1

Consider the IVP

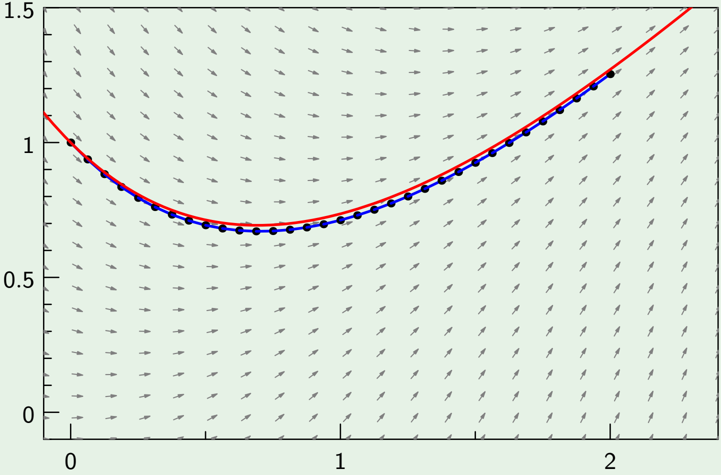
$$y' = t - y, \;\; y(0) = 1$$



Example 1

Consider the IVP

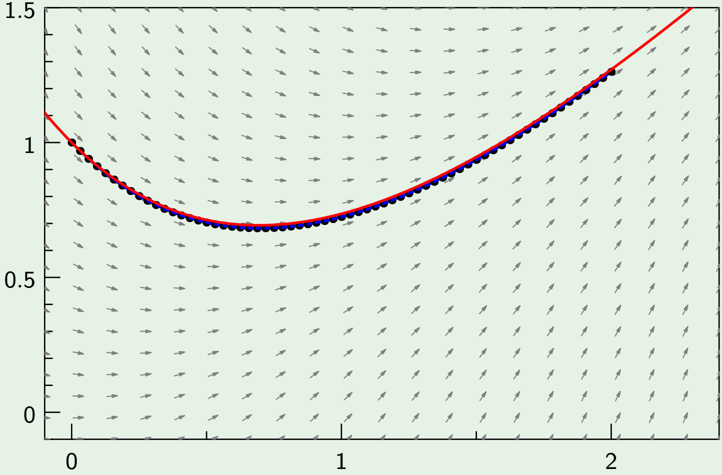
$$y' = t - y, \ \ y(0) = 1$$



Example 1

Consider the IVP

$$y' = t - y, \; y(0) = 1$$



Consider

Given the IVP

$$y' = f(t, y),\ \ y(t_0) = y_0$$

We want to compute approximate values for $y(t_n)$ at the (finite) set of points $t_1, t_2, t_3, \ldots, t_k$.

Consider

Given the IVP

$$y' = f(t, y), \; y(t_0) = y_0$$

We want to compute approximate values for $y(t_n)$ at the (finite) set of points $t_1, t_2, t_3, \ldots, t_k$.

We can calculate the t-values, for $1, 2, 3, \ldots, k$, with $t_n = t_0 + n \cdot h$.

Where h, called the **step size**, is the common difference between successive points.

Consider

Given the IVP

$$y' = f(t, y),\ \ y(t_0) = y_0$$

We want to compute approximate values for $y(t_n)$ at the (finite) set of points $t_1, t_2, t_3, \ldots, t_k$.

We can calculate the t-values, for $1, 2, 3, \ldots, k$, with $t_n = t_0 + n \cdot h$.

Where h, called the **step size**, is the common difference between successive points.

So, starting at (t_0, y_0) we want to follow the tangent line determined by

$$y - y_0 = (t - t_0)f(t_0, y_0)$$

Consider

Given the IVP

$$y' = f(t, y),\ y(t_0) = y_0$$

We want to compute approximate values for $y(t_n)$ at the (finite) set of points $t_1, t_2, t_3, \ldots, t_k$.

We can calculate the t-values, for $1, 2, 3, \ldots, k$, with $t_n = t_0 + n \cdot h$.

Where h, called the **step size**, is the common difference between successive points.

So, starting at (t_0, y_0) we want to follow the tangent line determined by

$$y - y_0 = (t - t_0) f(t_0, y_0)$$

to find the approximate solution $(t_1, y(t_1))$:

$$y_1 = y_0 + h \cdot f(t_0, y_0)$$

Consider

Given the IVP

$$y' = f(t, y),\ \ y(t_0) = y_0$$

We want to compute approximate values for $y(t_n)$ at the (finite) set of points $t_1, t_2, t_3, \ldots, t_k$.

We can extend this process to find all k points.

$$y_1 = y_0 + h \cdot f(t_0, y_0)$$

Consider

Given the IVP

$$y' = f(t, y),\ y(t_0) = y_0$$

We want to compute approximate values for $y(t_n)$ at the (finite) set of points $t_1, t_2, t_3, \ldots, t_k$.

We can extend this process to find all k points.

$$\begin{aligned} y_1 &= y_0 + h \cdot f(t_0, y_0) \\ y_2 &= y_1 + h \cdot f(t_1, y_1) \end{aligned}$$

Consider

Given the IVP

$$y' = f(t, y),\ \ y(t_0) = y_0$$

We want to compute approximate values for $y(t_n)$ at the (finite) set of points $t_1, t_2, t_3, \ldots, t_k$.

We can extend this process to find all k points.

$$\begin{aligned} y_1 &= y_0 + h \cdot f(t_0, y_0) \\ y_2 &= y_1 + h \cdot f(t_1, y_1) \\ y_3 &= y_2 + h \cdot f(t_2, y_2) \end{aligned}$$

Consider

Given the IVP

$$y' = f(t, y),\ \ y(t_0) = y_0$$

We want to compute approximate values for $y(t_n)$ at the (finite) set of points $t_1, t_2, t_3, \ldots, t_k$.

We can extend this process to find all k points.

$$\begin{aligned}
y_1 &= y_0 + h \cdot f(t_0, y_0) \\
y_2 &= y_1 + h \cdot f(t_1, y_1) \\
y_3 &= y_2 + h \cdot f(t_2, y_2) \\
&\vdots \\
y_k &= y_{k-1} + h \cdot f(t_{k-1}, y_{k-1})
\end{aligned}$$

Consider

Given the IVP

$$y' = f(t, y), \ y(t_0) = y_0$$

We want to compute approximate values for $y(t_n)$ at the (finite) set of points $t_1, t_2, t_3, \ldots, t_k$.

We can extend this process to find all k points.

$$\begin{aligned}
y_1 &= y_0 + h \cdot f(t_0, y_0) \\
y_2 &= y_1 + h \cdot f(t_1, y_1) \\
y_3 &= y_2 + h \cdot f(t_2, y_2) \\
&\vdots \\
y_k &= y_{k-1} + h \cdot f(t_{k-1}, y_{k-1})
\end{aligned}$$

The resulting piecewise-linear function (i.e. play connect-the-dots) is called the **Euler-approximate** solution.

Euler's Method

For the Initial-value problem

$$y' = f(t, y),\ \ y(t_0) = y_0$$

use the formulas

$$\begin{aligned} t_{n+1} &= t_n + h \\ y_{n+1} &= y_n + h \cdot f(t_n, y_n) \end{aligned}$$

to iteratively compute the points, using step size h,

$$(t_1, y_1), (t_2, y_2), \ldots, (t_k, y_k).$$

The piecewise-linear function connecting these points is the Euler approximtion to the solution $y(t)$ of the IVP for $t_0 \leq t \leq t_k$.

Example 2

Let us obtain the Euler-approximate solution of the IVP

$$y' = -2ty + t,\ y(0) = -1$$

with step size 0.1 on $[0, 0.4]$.

Example 2

Let us obtain the Euler-approximate solution of the IVP

$$y' = -2ty + t, \;\; y(0) = -1$$

with step size 0.1 on $[0, 0.4]$.

In other words:

$$\begin{aligned} f(t, y) &= -2ty + t = t(1 - 2y) \\ t_0 &= 0 \\ y_0 &= -1 \\ h &= 0.1 \\ k &= 1, 2, 3, 4 \end{aligned}$$

Example 2

$$t_1 = t_0 + h = 0 + 0.1 = 0.1$$
$$y_1 = y_0 + h \cdot f(t_0, y_0) = -1 + (0.1)(0)(1 - 2(-1)) = -1$$

Example 2

$$t_1 = t_0 + h = 0 + 0.1 = 0.1$$
$$y_1 = y_0 + h \cdot f(t_0, y_0) = -1 + (0.1)(0)(1 - 2(-1)) = -1$$

$$t_2 = t_1 + h = 0.1 + 0.1 = 0.2$$
$$\begin{aligned} y_2 &= y_1 + h \cdot f(t_1, y_1) \\ &= -1 + (0.1)(0.1)(1 - 2(-1)) = -0.97 \end{aligned}$$

Example 2

$$t_1 = t_0 + h = 0 + 0.1 = 0.1$$
$$y_1 = y_0 + h \cdot f(t_0, y_0) = -1 + (0.1)(0)(1 - 2(-1)) = -1$$

$$t_2 = t_1 + h = 0.1 + 0.1 = 0.2$$
$$\begin{aligned} y_2 &= y_1 + h \cdot f(t_1, y_1) \\ &= -1 + (0.1)(0.1)(1 - 2(-1)) = -0.97 \end{aligned}$$

$$t_3 = t_2 + h = 0.2 + 0.1 = 0.3$$
$$\begin{aligned} y_3 &= y_0 + h \cdot f(t_2, y_2) \\ &= -0.97 + (0.1)(0.2)(1 - 2(-0.97)) = -0.9112 \end{aligned}$$

Example 2

$$t_1 = t_0 + h = 0 + 0.1 = 0.1$$
$$y_1 = y_0 + h \cdot f(t_0, y_0) = -1 + (0.1)(0)(1 - 2(-1)) = -1$$

$$t_2 = t_1 + h = 0.1 + 0.1 = 0.2$$
$$\begin{aligned} y_2 &= y_1 + h \cdot f(t_1, y_1) \\ &= -1 + (0.1)(0.1)(1 - 2(-1)) = -0.97 \end{aligned}$$

$$t_3 = t_2 + h = 0.2 + 0.1 = 0.3$$
$$\begin{aligned} y_3 &= y_0 + h \cdot f(t_2, y_2) \\ &= -0.97 + (0.1)(0.2)(1 - 2(-0.97)) = -0.9112 \end{aligned}$$

$$t_4 = t_3 + h = 0.3 + 0.1 = 0.4$$
$$\begin{aligned} y_4 &= y_3 + h \cdot f(t_3, y_3) \\ &= -0.9112 + (0.1)(0.3)(1 - 2(-0.9112)) = -0.82652 \end{aligned}$$

Example 2

How does this compare to the exact solution $y(t) = 0.5 - 1.5e^{-t^2}$?

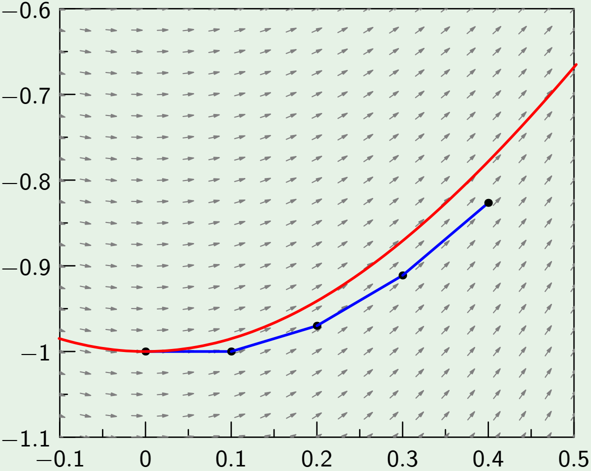
Example 2

How does this compare to the exact solution $y(t) = 0.5 - 1.5e^{-t^2}$?

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	-1.000000	-1.000000	**0.000000**
1	0.1	-1.000000	-0.985075	**-0.014925**
2	0.2	-0.970000	-0.941184	**-0.028815**
3	0.3	-0.911200	-0.870897	**-0.040303**
4	0.4	-0.826528	-0.778216	**-0.048312**

Notice how the error grows rapidly.

Example 2



Example 3

Find the Euler-approximation of

$$y' = -2ty, \ y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**

Example 3

Find the Euler-approximation of

$$y' = -2ty, \ y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**

Example 3

Find the Euler-approximation of

$$y' = -2ty, \ y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**

Example 3

Find the Euler-approximation of

$$y' = -2ty, \; y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**
2	0.4	0.9200000	0.8521437	**-0.067856**

Example 3

Find the Euler-approximation of

$$y' = -2ty,\ y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**
2	0.4	0.9200000	0.8521437	**-0.067856**
3	0.6	0.7728000	0.6976763	**-0.075124**

Example 3

Find the Euler-approximation of

$$y' = -2ty, \ y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**
2	0.4	0.9200000	0.8521437	**-0.067856**
3	0.6	0.7728000	0.6976763	**-0.075124**
4	0.8	0.5873280	0.5272925	**-0.060036**

Example 3

Find the Euler-approximation of

$$y' = -2ty, \ y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**
2	0.4	0.9200000	0.8521437	**-0.067856**
3	0.6	0.7728000	0.6976763	**-0.075124**
4	0.8	0.5873280	0.5272925	**-0.060036**
5	1.0	0.3993830	0.3678794	**-0.031504**

Example 3

Find the Euler-approximation of

$$y' = -2ty,\ y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**
2	0.4	0.9200000	0.8521437	**-0.067856**
3	0.6	0.7728000	0.6976763	**-0.075124**
4	0.8	0.5873280	0.5272925	**-0.060036**
5	1.0	0.3993830	0.3678794	**-0.031504**
6	1.2	0.2396298	0.2369277	**-0.002702**

Example 3

Find the Euler-approximation of

$$y' = -2ty, \; y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**
2	0.4	0.9200000	0.8521437	**-0.067856**
3	0.6	0.7728000	0.6976763	**-0.075124**
4	0.8	0.5873280	0.5272925	**-0.060036**
5	1.0	0.3993830	0.3678794	**-0.031504**
6	1.2	0.2396298	0.2369277	**-0.002702**
7	1.4	0.1246075	0.1408584	**0.016251**

Example 3

Find the Euler-approximation of

$$y' = -2ty, \; y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**
2	0.4	0.9200000	0.8521437	**-0.067856**
3	0.6	0.7728000	0.6976763	**-0.075124**
4	0.8	0.5873280	0.5272925	**-0.060036**
5	1.0	0.3993830	0.3678794	**-0.031504**
6	1.2	0.2396298	0.2369277	**-0.002702**
7	1.4	0.1246075	0.1408584	**0.016251**
8	1.6	0.0548273	0.0773047	**0.022477**

Example 3

Find the Euler-approximation of

$$y' = -2ty, \; y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**
2	0.4	0.9200000	0.8521437	**-0.067856**
3	0.6	0.7728000	0.6976763	**-0.075124**
4	0.8	0.5873280	0.5272925	**-0.060036**
5	1.0	0.3993830	0.3678794	**-0.031504**
6	1.2	0.2396298	0.2369277	**-0.002702**
7	1.4	0.1246075	0.1408584	**0.016251**
8	1.6	0.0548273	0.0773047	**0.022477**
9	1.8	0.0197378	0.0391639	**0.019426**

Example 3

Find the Euler-approximation of

$$y' = -2ty,\ y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.

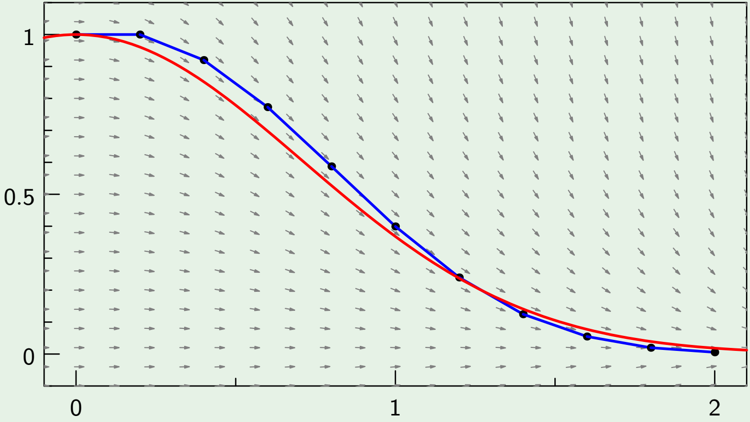
n	t_n	y_n	$y(t_n)$	**Error**
0	0.0	1.0000000	1.0000000	**0.000000**
1	0.2	1.0000000	0.9607894	**-0.039211**
2	0.4	0.9200000	0.8521437	**-0.067856**
3	0.6	0.7728000	0.6976763	**-0.075124**
4	0.8	0.5873280	0.5272925	**-0.060036**
5	1.0	0.3993830	0.3678794	**-0.031504**
6	1.2	0.2396298	0.2369277	**-0.002702**
7	1.4	0.1246075	0.1408584	**0.016251**
8	1.6	0.0548273	0.0773047	**0.022477**
9	1.8	0.0197378	0.0391639	**0.019426**
10	2.0	0.0055265	0.0183156	**0.012789**

Example 3

Find the Euler-approximation of

$$y' = -2ty, \ \ y(0) = 1$$

using a step size of 0.2 over the range of $[0, 2]$.



There are two types of error:

There are two types of error:

- **Roundoff error** is the discrepancy arising from rounding numbers. This tends to snowball pretty fast when you have a great many calculations.

There are two types of error:

- **Roundoff error** is the discrepancy arising from rounding numbers. This tends to snowball pretty fast when you have a great many calculations.
- **Discretization error** is the error that results from the approximation method itself. For Euler's method this is cause by using the linear tangent lines to approximate a nonlinear curve.

There are two types of error:

- **Roundoff error** is the discrepancy arising from rounding numbers. This tends to snowball pretty fast when you have a great many calculations.
- **Discretization error** is the error that results from the approximation method itself. For Euler's method this is cause by using the linear tangent lines to approximate a nonlinear curve.

It can be shown, using Taylor series expansions, that the error is proportional to the square of the step size.

$$|y_i - y(t_i)| \leq C \cdot h^2$$

Where the constant C depends of the size of the second derivative of the exact solution.

There are two types of error:

- **Roundoff error** is the discrepancy arising from rounding numbers. This tends to snowball pretty fast when you have a great many calculations.
- **Discretization error** is the error that results from the approximation method itself. For Euler's method this is cause by using the linear tangent lines to approximate a nonlinear curve.

It can be shown, using Taylor series expansions, that the error is proportional to the square of the step size.

$$|y_i - y(t_i)| \leq C \cdot h^2$$

Where the constant C depends of the size of the second derivative of the exact solution.

We call this error the **local discretization error** because it estimates the error for a single step only. After n steps, we have n times the error. Which we call the **global discretization error**.

Global Discretization Error in Euler's Method

If the solution of the IVP $y' = f(t, y)$, $y(t_0) = y_0$ has a continuous second derivative on the interval $[t_0, t_k]$, and y_n is the value of the Euler approximation at t_n, $t_0 < t_1 < \cdots < t_n \cdots < t_k$, then there exists a constant C such that

$$|y_n - y(t_n)| \leq C \cdot h, \qquad n = 1, 2, \ldots, k.$$

where step size $h = t_n - t_{n-1}$.

Second-Order Runge-Kutta Method

For the IVP $y' = f(t, y)$, $y(t_0) = y_0$, use the following formulas to compute the points $(t_1, y_1), (t_2, y_2), \ldots$ of the approximate solution, using step size h:

$$\begin{aligned} t_{n+1} &= t_n + h \\ y_{n+1} &= y_n + h \cdot k_{n_1} \end{aligned}$$

where

$$\begin{aligned} k_{n_1} &= f(t_n, y_n) \\ k_{n_2} &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_{n_1}\right) \end{aligned}$$

Fourth-Order Runge-Kutta Method

For the IVP $y' = f(t, y)$, $y(t_0) = y_0$, use the following formulas to compute the points $(t_1, y_1), (t_2, y_2), \ldots$ of the approximate solution, using step size h:

$$t_{n+1} = t_n + h$$
$$y_{n+1} = y_n + \frac{h}{6}(k_{n_1} + 2k_{n_2} + 2k_{n_3} + k_{n_4})$$

where

$$k_{n_1} = f(t_n, y_n)$$
$$k_{n_2} = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_{n_1}\right)$$
$$k_{n_3} = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_{n_2}\right)$$
$$k_{n_2} = f\left(t_n + h, y_n + h \cdot k_{n_3}\right)$$