MET-576-4

Modelagem Numérica da Atmosfera

Dr. Silvio Nilo Figueroa Rivero & Dr. Paulo Yoshio Kubota

Os métodos numéricos, formulação e parametrizações utilizados nos modelos atmosféricos serão descritos em detalhe.

3 Meses
24 Aulas (2 horas cada)

Dinâmica:

Métodos numéricos amplamente utilizados na solução numérica das equações diferencias parciais que governam os movimentos na atmosfera serão o foco, mas também serão analisados os novos conceitos e novos métodos.

Dinâmica 29/09/2020 a 29/10/2020

- ✓ **Métodos de diferenças finitas.**
- ✓ **Acurácia.**
- ✓ **Consistência.**
- ✓ **Estabilidade.**
- ✓ **Convergência.**
- ✓ **Grades de Arakawa A, B, C e E.**
- ✓ **Domínio de influência e domínio de dependência.**
- ✓ **Dispersão numérica e dissipação.**
- ✓ **Definição de filtros monótono e positivo.**
- ✓ **Métodos espectrais.**
- ✓ **Métodos de volume finito.**
- ✓ **Métodos Semi-Lagrangeanos.**
- ✓ **Conservação de massa local.**
- ✓ **Esquemas explícitos versus semi-implícitos.**
- ✓ **Métodos semi-implícitos.**

Dinâmica 29/09/2020 a 29/10/2020
Métodos de diferenças finitas.

Notas: esquema upstream (upwind- FTBS)

- *Centered explicit scheme*

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0.$$

- *Centered implicit scheme*

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} = 0.$$

- *Upwind scheme*

$$\begin{cases} \dfrac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\dfrac{u_j^n - u_{j-1}^n}{\Delta x} = 0 \text{ , if } a > 0, \\ \dfrac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\dfrac{u_{j+1}^n - u_j^n}{\Delta x} = 0 \text{ , if } a < 0. \end{cases}$$

- *Lax-Friedrichs*

$$\frac{2u_j^{n+1} - u_{j+1}^n - u_{j-1}^n}{2\Delta t} + a\,\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0.$$

- *Beam-Warming* if $a > 0$,

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\frac{3u_j^n - 4u_{j-1}^n + u_{j-2}^n}{2\Delta x} - \frac{a^2\Delta t}{2}\,\frac{u_j^n - 2u_{j-1}^n + u_{j-2}^n}{\Delta x^2} = 0.$$

Lax–Friedrich method

Lax-Friedrichs scheme is an explicit, first order scheme, using forward difference in time and central difference in space. However, the scheme is stabilized by averaging u_i^n over the neighbour cells in the in the temporal approximation:

$$\frac{u_i^{n+1} - \frac{1}{2}(u_{i+1}^n + u_{i-1}^n)}{\Delta t} = -\frac{F_{i+1}^n - F_{i-1}^n}{2\Delta x}$$

The Lax-Friedrich scheme is the obtained by isolation u_i^{n+1} at the right hand side:

$$u_i^{n+1} = \frac{1}{2}(u_{i+1}^n + u_{i-1}^n) - \frac{\Delta t}{2\Delta x}(F_{i+1}^n - F_{i-1}^n)$$

By assuming a linear flux $F = a_0 u$ it may be shown that the Lax-Friedrich scheme takes the form:

$$u_i^{n+1} = \frac{1}{2}(u_{i+1}^n + u_{i-1}^n) - \frac{C}{2}(u_{i+1}^n - u_{i-1}^n)$$

Lax–Wendroff method

O método Lax–Wendroff, nomeado após Peter Lax e Burton Wendroff, é um método numérico para a solução de equações diferenciais parciais hiperbólicas, com base em diferenças finitas.

É de precisão de segunda ordem no espaço e no tempo.

Este método é um exemplo de integração de tempo explícita em que a função que define a equação governante é avaliada no momento atual.

Lax–Wendroff method

$$u_j^{n+1} = u_j^n + \Delta t \frac{\partial u}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2} + \cdots .$$

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial f\big(u(x,t)\big)}{\partial x} = 0$$

$$\frac{\partial u(x,t)}{\partial t} = -\frac{\partial f\big(u(x,t)\big)}{\partial x}$$

$$\frac{\partial^2 u(x,t)}{\partial t^2} = -\frac{\partial}{\partial t} \frac{\partial f\big(u(x,t)\big)}{\partial x}$$

$$\frac{\partial^2 u(x,t)}{\partial t^2} = -\frac{\partial^2 f\big(u(x,t)\big)}{\partial x \partial t}$$

$$u_j^{n+1} = u_j^n - \Delta t \frac{\partial f\big(u(x,t)\big)}{\partial x} - \frac{\Delta t^2}{2} \frac{\partial^2 f\big(u(x,t)\big)}{\partial x \partial t} + \cdots .$$

Lax–Wendroff method

$$u_j^{n+1} = u_j^n - \Delta t \frac{\partial f\big(u(x,t)\big)}{\partial x} - \frac{\Delta t^2}{2} \frac{\partial^2 f\big(u(x,t)\big)}{\partial x \partial t} + \cdots .$$

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial f\big(u(x,t)\big)}{\partial x} = 0$$

$$\frac{\partial u(x,t)}{\partial t} = - \frac{\partial f\big(u(x,t)\big)}{\partial x}$$

$$F = a_0 u(\text{x,t})$$

$$\frac{\partial^2 u(x,t)}{\partial t^2} = - \frac{\partial}{\partial t} \frac{\partial f\big(u(x,t)\big)}{\partial x}$$

$$\frac{\partial^2 u(x,t)}{\partial t^2} = - \frac{\partial^2 f\big(u(x,t)\big)}{\partial x \partial t}$$

$$u_j^{n+1} = u_j^n - a_0 \Delta t \frac{\partial u(x,t)}{\partial x} - a_0 \frac{\Delta t^2}{2} \frac{\partial}{\partial x} \frac{\partial u(x,t)}{\partial t} + \cdots .$$

Lax–Wendroff method

$$u_j^{n+1} = u_j^n - a_0 \Delta t \frac{\partial u(x,t)}{\partial x} - a_0 \frac{\Delta t^2}{2} \frac{\partial}{\partial x} \frac{\partial u(x,t)}{\partial t} + \cdots .$$

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial f\big(u(x,t)\big)}{\partial x} = 0$$

$$F = a_0 u(\text{x,t})$$

$$\frac{\partial u(x,t)}{\partial t} = -\frac{\partial f\big(u(x,t)\big)}{\partial x}$$

$$\frac{\partial^2 u(x,t)}{\partial t^2} = -\frac{\partial}{\partial t} \frac{\partial f\big(u(x,t)\big)}{\partial x}$$

$$\frac{\partial^2 u(x,t)}{\partial t^2} = -\frac{\partial^2 f\big(u(x,t)\big)}{\partial x \partial t}$$

$$u_j^{n+1} = u_j^n - a_0 \Delta t \frac{\partial u(x,t)}{\partial x} + a_0 \frac{\Delta t^2}{2} \frac{\partial^2 f(x,t)}{\partial x \partial x} + \cdots .$$

Lax–Wendroff method

$$u_j^{n+1} = u_j^n - a_0 \Delta t \frac{\partial u(x,t)}{\partial x} + a_0 \frac{\Delta t^2}{2} \frac{\partial^2 f(x,t)}{\partial x \partial x} + \cdots.$$

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial f\big(u(x,t)\big)}{\partial x} = 0$$

$$\frac{\partial u(x,t)}{\partial t} = -\frac{\partial f\big(u(x,t)\big)}{\partial x}$$

$$F = a_0 u(\mathrm{x,t})$$

$$\frac{\partial^2 u(x,t)}{\partial t^2} = -\frac{\partial}{\partial t} \frac{\partial f\big(u(x,t)\big)}{\partial x}$$

$$\frac{\partial^2 u(x,t)}{\partial t^2} = -\frac{\partial^2 f\big(u(x,t)\big)}{\partial x \partial t}$$

$$u_j^{n+1} = u_j^n - a_0 \Delta t \frac{\partial u(x,t)}{\partial x} + a_0^2 \frac{\Delta t^2}{2} \frac{\partial^2 u(x,t)}{\partial x \partial x} + \cdots.$$

Lax–Wendroff method

Suponha que se tenha uma equação da seguinte forma:

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial f\big(u(x,t)\big)}{\partial x} = 0$$

Onde x**e** t **são variáveis independentes, e o estado inicial** $u(x, t = 0)$ **é especificado.**

$$u_j^{n+1} = u_j^n - a_0 \Delta t \frac{\partial u(x,t)}{\partial x} + a_0^2 \frac{\Delta t^2}{2} \frac{\partial^2 u(x,t)}{\partial x \partial x} + \cdots .$$

Lax–Wendroff method

Caso Linear:

Onde $f(u(x,t)) = Au(x,t)$ **e** $A = Cte$

The Lax–Wendroff method belongs to the class of conservative schemes and can be derived in various ways. For simplicity, we will derive the method by using a simple model equation adv., namely the linear advection equation with $F = a_0 u$ **, where *a* is a constant propagation velocity. The Lax-Wendroff outset is a Taylor approximation of** u_j^{n+1}**:**

$$u_j^{n+1} = u_j^n + \Delta t \frac{\partial u}{\partial t}\bigg|_j^n + \frac{(\Delta t)}{2} \frac{\partial^2 u}{\partial t^2}\bigg|_j^n + \cdots$$

From the differential equation of Adv. we get by differentiation

$$\frac{\partial u}{\partial t}\bigg|_j^n = -a_0 \frac{\partial u}{\partial x}\bigg|_j^n \quad \text{and} \quad \frac{\partial^2 u}{\partial t^2}\bigg|_j^n = a_0^2 \frac{\partial^2 u}{\partial x^2}\bigg|_j^n$$

Lax–Wendroff method

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial f\big(u(x,t)\big)}{\partial x} = 0$$

$$f = a_0 u$$

$$u_j^{n+1} = u_j^n - a_0 \Delta t \frac{\partial u(x,t)}{\partial x} + a_0^2 \frac{\Delta t^2}{2} \frac{\partial^2 u(x,t)}{\partial x \partial x} + \cdots .$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2\Delta x} + O(\Delta x^2)$$

$$C = a_0 \frac{\Delta t}{\Delta x}$$

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{\Delta x^2} + O(\Delta x^2)$$

$$u_j^{n+1} = u_j^n - \frac{C}{2}\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{C^2}{2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)$$

Lax–Wendroff method

Before substitution in the Taylor expansion we approximate the spatial derivatives by central differences:

$$\left.\frac{\partial u}{\partial x}\right|_j^n \approx \frac{u_{j+1}^n - u_{j-1}^n}{(2\Delta x)} \qquad \text{and} \qquad \left.\frac{\partial^2 u}{\partial x^2}\right|_j^n \approx \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}$$

and then the Lax-Wendroff scheme follows by substitution:

$$u_j^{n+1} = u_j^n - \frac{C}{2}\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{C^2}{2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)$$

with the local truncation error*j*

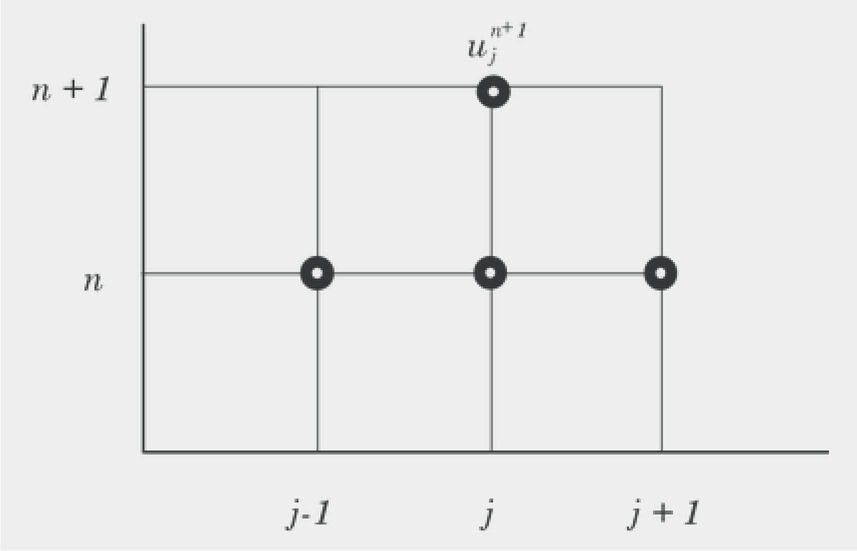
$$T_j^n = \frac{1}{6}\cdot\left[(\Delta t)^2\frac{\partial^3 u}{\partial t^3} + a_0(\Delta x)^2\frac{\partial^3 u}{\partial x^3}\right]_j^n = O[(\Delta t)^2, (\Delta x)^2]$$

Lax–Wendroff method

The resulting difference equation in may also be formulated as:

$$u_j^{n+1} = \frac{C}{2}(1+C)u_{j-1}^n + (1-C^2)u_j^n - \frac{C}{2}(1-C)u_{j+1}^n$$

The explicit Lax Wendroff stenticl is illustrated in Fig.



Exercicios Lax–Wendroff method

- *Centered explicit scheme*

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \;=\; 0.$$

- *Centered implicit scheme*

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} \;=\; 0.$$

- *Upwind scheme*

$$\begin{cases} \dfrac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\dfrac{u_j^n - u_{j-1}^n}{\Delta x} \;=\; 0\ \text{, if } a > 0, \\ \dfrac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\dfrac{u_{j+1}^n - u_j^n}{\Delta x} \;=\; 0\ \text{, if } a < 0. \end{cases}$$

- *Lax-Friedrichs*

$$\frac{2u_j^{n+1} - u_{j+1}^n - u_{j-1}^n}{2\Delta t} + a\,\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \;=\; 0.$$

- Lax-Wendroff

$$u_j^{n+1} = u_j^n - \frac{C}{2}\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{C^2}{2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)$$

Runge-Kutta method

First Order Runge-Kutta method

Consider the situation in which the solution, *y(t)*, to a differential equation

$$\frac{dy(t)}{dt} = y'(t) = f(y(t), t)$$

is to be approximated by computer starting from some known initial condition, $y(t_0) = y_0$ (note that the (') tick mark denotes differentiation). The following text develops an intuitive technique for doing so, and then presents several examples. This technique is known as "Euler's Method" or "First Order Runge-Kutta".

First Order Runge-Kutta method

A First Order Linear Differential Equation with No Input

$$\frac{dy(t)}{dt} + 2y(t) = 0$$

or

$$\frac{dy(t)}{dt} = -2y(t)$$

$$\frac{1}{y(t)}\frac{dy(t)}{dt} = -2$$

$$\int_{y(t=0)}^{y(t=h)} \frac{1}{y(t)}dy(t) = -2\int_0^h dt$$

$$\ln\big(y(t=h)\big) - \ln\big(y(t=0)\big) = -2h + 0$$

$$\ln\left(\frac{y(t=h)}{y(t=0)}\right) = -2h + 0$$

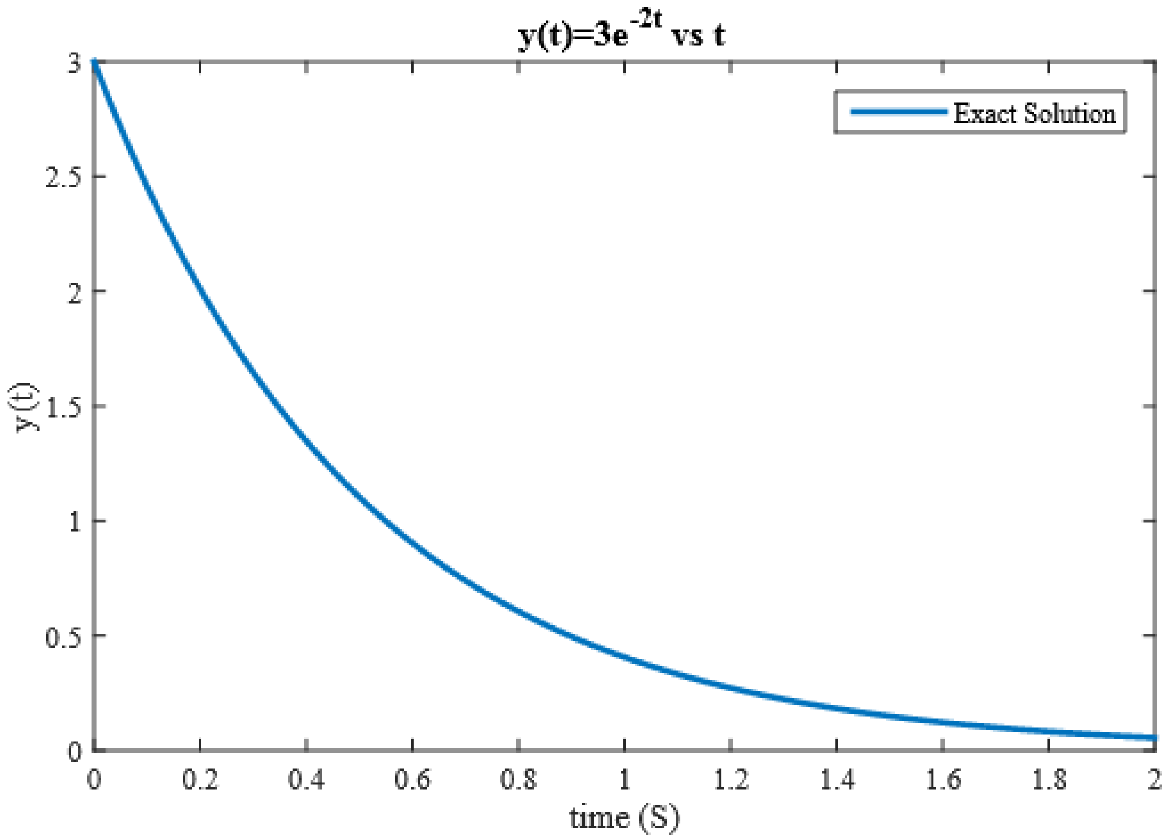
$$\frac{y(t=h)}{y(t=0)} = e^{-2h}$$

$$y(t=h) = y(t=0)e^{-2h}$$

$$y(t) = 3e^{-2t}, t \geq 0$$

with the initial condition set as $y(0) = 3$. For this case the exact solution can be determined to be ($y(t) = 3e^{-2t}, t \geq 0$) and is shown below. Since we know the exact solution in this case we will be able to use it to check the accuracy of our approximate solution.

First Order Runge-Kutta method



First Order Runge-Kutta method

There are several ways to develop an approximate solution, we will do so using the Taylor Series for $y(t)$ expanded about $t = 0$ (in general we expand around $t = t_0$).

$$y(t) = y(0) + y'(0)t + y''(0)\frac{t^2}{2} + \cdots$$

We now restrict our solution to a short time step, h, after $t = 0$ and truncate the Taylor series after the first derivative

$$y(h) = y(0) + y'(0)h + y''(0)\frac{h^2}{2} + \cdots$$

$$y(t) = y(0) + \frac{1}{1!}\frac{\partial y}{\partial t}\Delta t^1 + \frac{1}{2!}\frac{\partial^2 y}{\partial t^2}\Delta t^2 \ldots.$$

$$y(h) \approx y(0) + y'(0)h$$

$$y(t) = y(0) + \frac{\partial y(0)}{\partial t}\Delta t + \cdots.$$

First Order Runge-Kutta method

We now restrict our solution to a short time step, h, after $t = 0$ and truncate the Taylor series after the first derivative

$$y_1(h) = y(0) + \frac{\partial y(0)}{\partial t} h + \cdots .$$

$$\boxed{\frac{\partial y}{\partial t} = C * \frac{\partial y}{\partial x}}$$

$$y_2\left(\frac{h}{2}\right) = y_1(h) + \frac{\partial y_1(h)}{\partial t}\frac{h}{2} + \cdots .$$

$$y_3\left(\frac{h}{2}\right) = y_2\left(\frac{h}{2}\right) + \frac{\partial y_2(\frac{h}{2})}{\partial t}\frac{h}{2} + \cdots .$$

$$y_4(h) = y_3\left(\frac{h}{2}\right) + \frac{\partial y_3(\frac{h}{2})}{\partial t} h + \cdots .$$

First Order Runge-Kutta method

We call the value of the approximation $y^*(h)$, and we call the derivative $y'(0) = k\_1$.

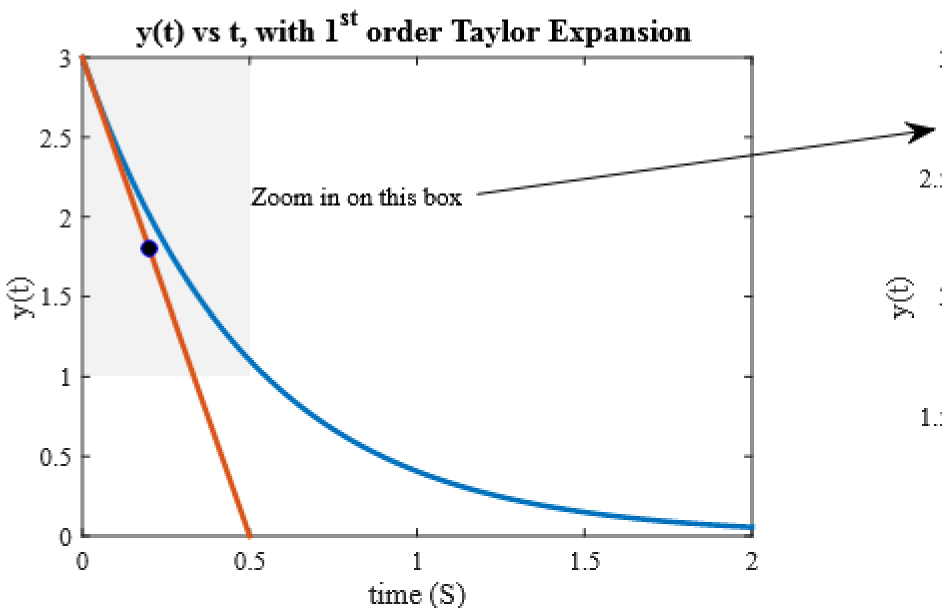
$$y(t) = 3e^{-2t}, t \geq 0$$

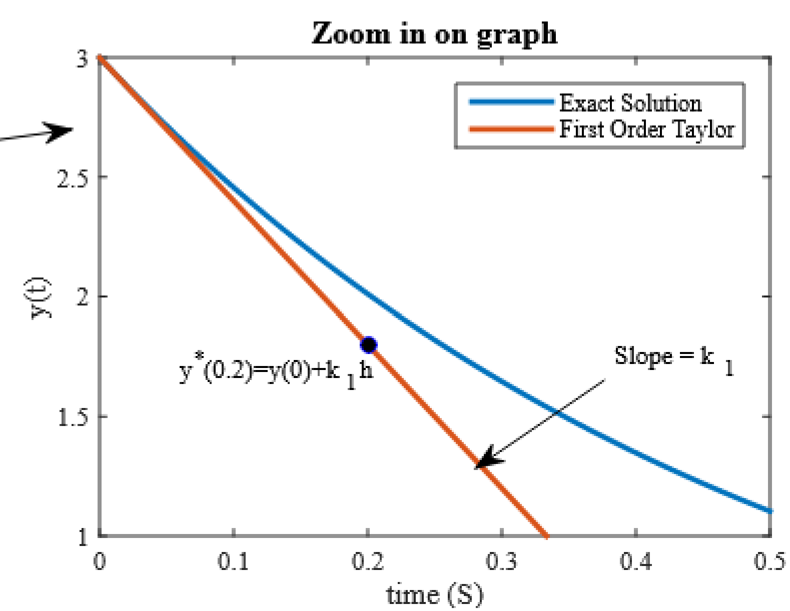
$$y(h) \approx y(0) + y'(0)h$$

$$\frac{dy(t)}{dt} = -2y(t)$$

$$y^*(h) = y(0) + k_1 h$$

This is shown on the graph below for *h=0.2*





First Order Runge-Kutta method

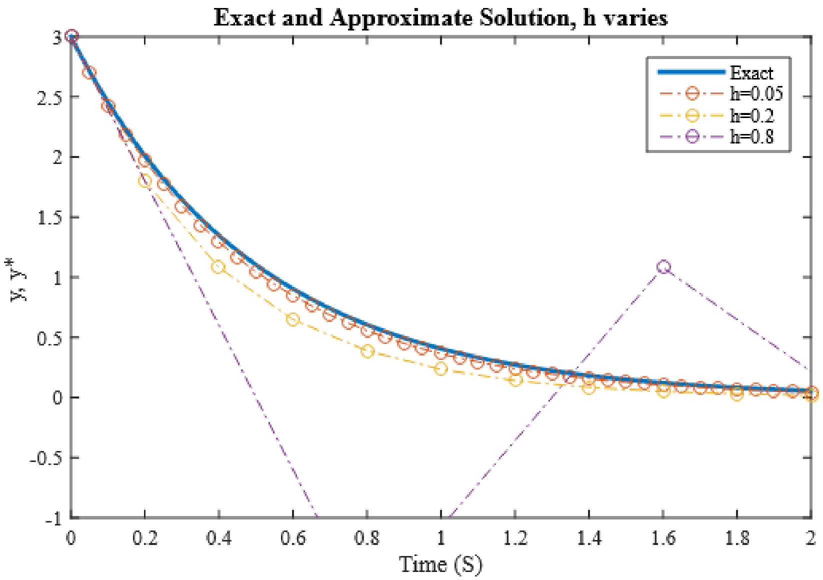
To find the value of the approximation after the next time step, $y^*(2h)$, we simply repeat the process using our approximation, $y^*(h)$ to estimate the derivative at time h (we don't know y(h) exactly, so we can only estimate the derivative - we call this estimate k_1).

$y'(t) = -2y(t)$	exact expression for derivative
$k_1 = -2y^*(h)$	approximation for derivative
$y(2h) = y(h) + y'(h)h + y''(h)\frac{h^2}{2} + \cdots$	4aylor Series around t = h
$y(2h) \approx y(h) + y'(0)h$	Truncated 4aylor Series
$y^*(2h) = y^*(h) + k_1 h$	ApproximateSolution

First Order Runge-Kutta method

In general we move forward one step in time from t_0 to $t_0 + h$

$y'(t_0) = -2y(t_0)$	exact expression for derivative at $t = t_0$
$k_1 = -2y^*(t_0)$	Previous approx for y(t) gives approx for derivative
$y(t_0 + h) = y(t_0) + y'(t_0)h + y''(t_0)\frac{h^2}{2} + \cdots$	4aylor Series around t =t_0
$y(t_0 + h) \approx y(t_0) + y'(t_0)h$	Truncated 4aylor Series
$y^*(t_0 + h) = y^*(t_0) + k_1 h$	Approximate Solution at next value of y



First Order Runge-Kutta method

Key Concept: First Order Runge-Kutta Algorithm

For a first order ordinary differential equation defined by

$$\frac{dy(t)}{dt} = f(y(t), t)$$

to progress from a point at $t = t_0$, $y^*(t_0)$, by one time step, h, follow these steps (repetitively).

$$k_1 = f(y^*(t_0), t_0) \quad \text{approximation for derivative}$$
$$y^*(t_0 + h) = y^*(t_0) + k_1 h \quad \text{approximate solution at next time step}$$

Notes:
an initial value of the function must be given to start the algorithm.

Second Order Runge-Kutta method

Consider the situation in which the solution, *y(t)*, to a differential equation

$$\frac{dy(t)}{dt} = y'(t) = f(y(t), t), \qquad \text{with } y(t_0) = y_0$$

is to be approximated by computer (starting from some known initial condition, $y(t_0) = y_0$; also, note that the (´) tick mark denotes differentiation).

This technique is known as "Second Order Runge-Kutta".

Second Order Runge-Kutta method

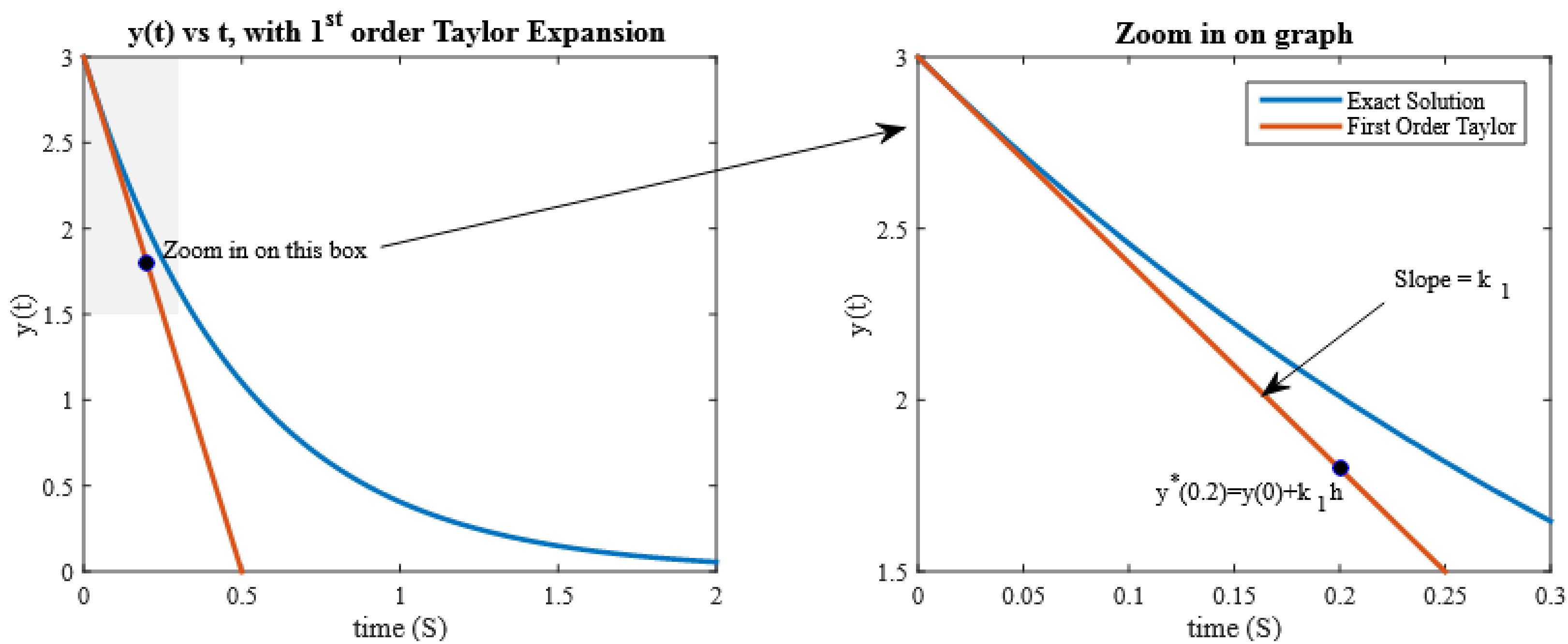
The first order Runge-Kutta method used the derivative at time t_0 ($t_0 = 0$ in the graph below) to estimate the value of the function at one time step in the future. t. We repeat the central concept of generating a step forward in time.

$$\frac{dy(t)}{dt} + 2y(t) = 0 \quad \text{or} \quad \frac{dy(t)}{dt} = -2y(t)$$

with the initial condition set as $y(0) = 3$. The exact solution in this case is $y(t) = 3e^{-2t}$, $t \geq 0$, though in general we won't know this and will need numerical integration methods to generate an approximation.

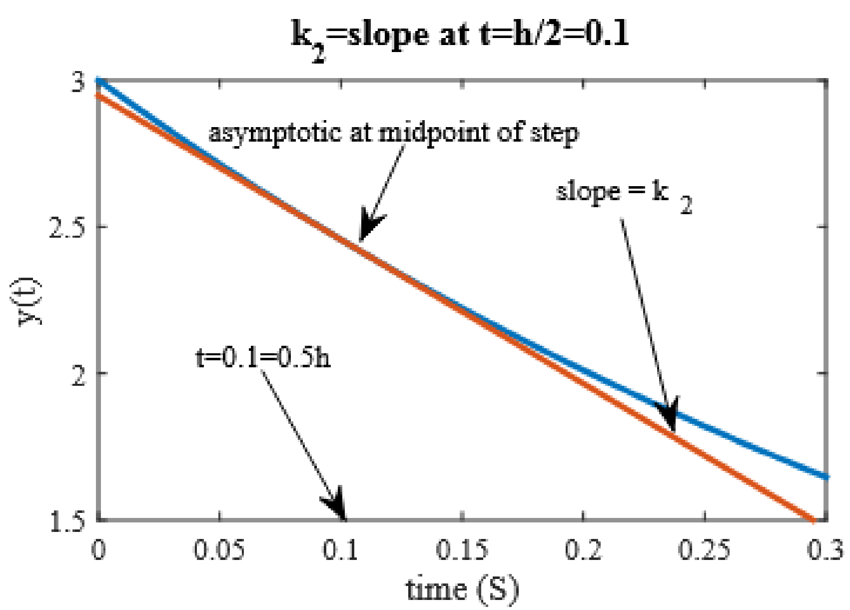
Second Order Runge-Kutta method

In the graph below, the slope at $t = 0$ is called k_1, and the estimate is called $y^*(h)$; in this example $h = 0.2$.



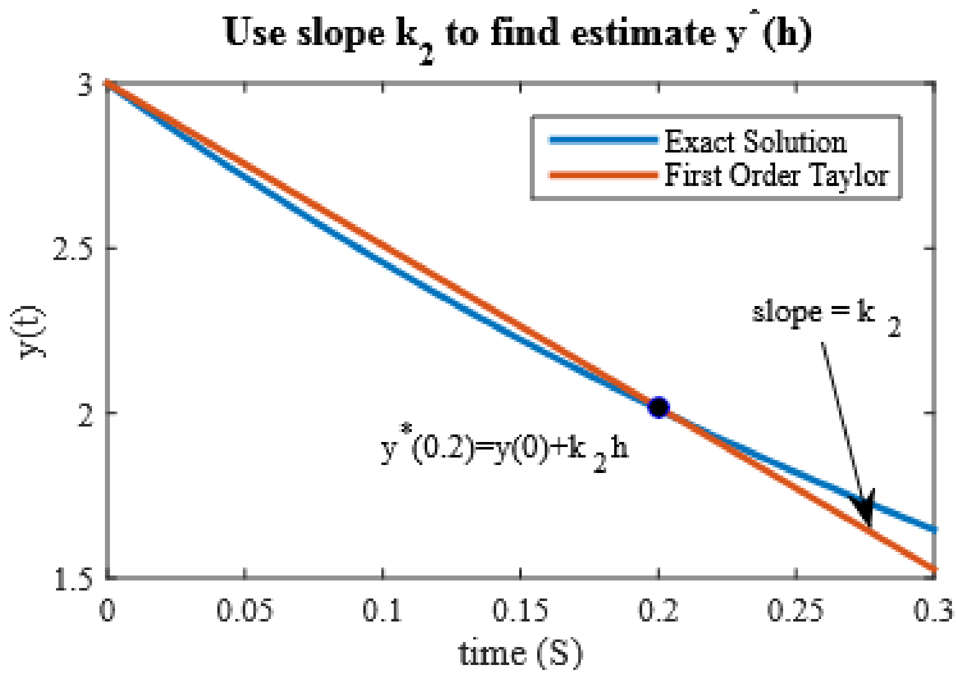
Second Order Runge-Kutta method

This obviously leads to some error in the estimate, and we would like to reduce this error. One way we could do this, conceptually, is to use the derivative at the halfway point between *$t=0$* and *$t=h=0.2$*. The slope at this point (*$t=½h=0.1$*) is shown below (and is labeled *k_2*). Note the line (orange) is tangent to the curve (blue) at *$t=½h$*.



Second Order Runge-Kutta method

Now if we use this intermediate slope, k_2, as we step ahead in time then we get better estimate, $y^*(h)$, than we did before. On the diagram below the exact value of the solution is $y(0.1) = 2.0110$ and the approximation is $y^*(0.1) = 2.0175$ for an error of about 0.3% (compared with about 10% error for the first order Runge-Kutta).



Second Order Runge-Kutta method

This seems like a very nice solution, and obviously generates a significantly more accurate approximation than the first order technique that uses a line with slope, k_1, calculated at $t = 0$. The problem is we don't know the exact value of $y(½h)$ so we can't find the exact value of k_2 the slope at $t = ½h$ (Recall that the calculation of the derivative requires knowledge of the value of the function, $y'(t) = -2y(t)$).

What we do instead is use the First Order Runge-Kutta to generate an approximate value for $y(t)$ at $t = ½h = 0.1$, call it $y_1(½h)$. We then use this estimate to generate k_2 (which will be an approximation to the slope at the midpoint), and then use k_2 to find $y^*(h)$. To step from the starting point at $t = 0$ to an estimate at $t = h$, follow the procedure below.

Second Order Runge-Kutta method

$y'(0) = -2y(0)$	expression for derivative at $t = 0$
$k_1 = -2y(0)$	derivative at $t = 0$
$y_1\left(\frac{h}{2}\right) = y(0) + k_1\frac{h}{2}$	intermediate estimate of function at $t = h/2$
$k_2 = -2y_1\left(\frac{h}{2}\right)$	estimate of slope at $t = h/2$
$y(h) = y(0) + y'(0)h + y''(0)\frac{h^2}{2} + \cdots$	Taylor Series around $t = 0$
$y(t) \approx y(0) + y'(0)h$	Truncate Taylor Series
$y^*(h) = y(0) + k_2 h$	estimate of $y(h)$

$$k_2 = -2\left(y(0) + k_1\frac{h}{2}\right)$$

In general, to go from the estimate $t = t_0$ **to an estimate at** $t = t_0 + h$

Second Order Runge-Kutta method

$y'(t_0) = -2y(t_0)$ expression for derivative at $t = t_0$

$k_1 = -2y^*(t_0)$ approximate derivative at $t = t_0$

$y_1\left(t_0 + \frac{h}{2}\right) = y^*(t_0) + k_1\frac{h}{2}$ intermediate estimate of function at $t = t_0 + h/2$

$k_2 = -2y_1\left(t_0 + \frac{h}{2}\right)$ estimate of slope at $t = t_0 + h/2$

$y(t_0 + h) = y(t_0) + y'(t_0)h + y''(0)\frac{h^2}{2} + \cdots$ Taylor Series around $t = t_0$

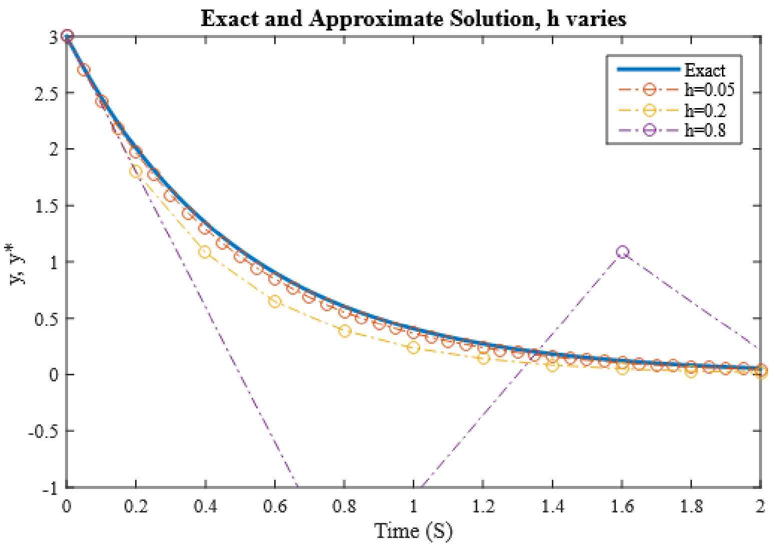
$y(t_0 + h) \approx y(t_0) + y'(t_0)h$ Truncated Taylor Series

$y^*(t_0 + h) = y(t_0) + k_2 h$ estimate of $y(t_0 + h)$

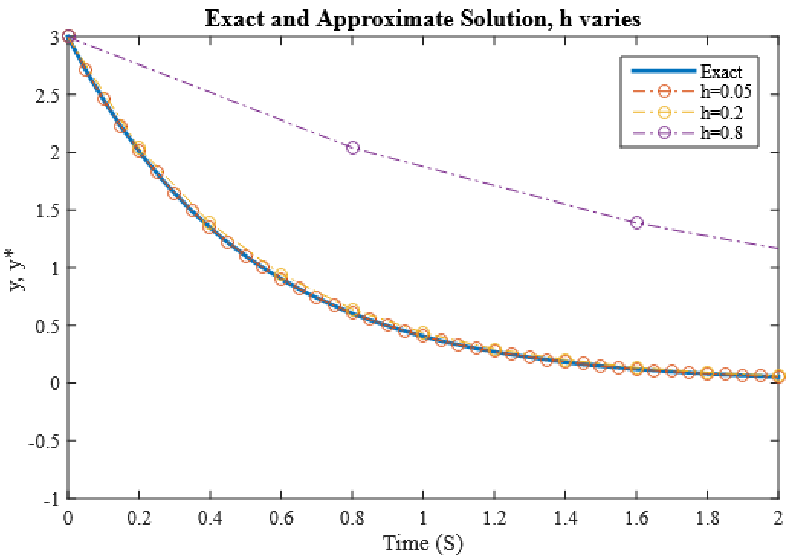
Second Order Runge-Kutta method

Note that larger values of *h* result in poorer approximations, but that the solutions are much better than those obtained with the First Order Runge-Kutta for the same value of *h*.

RK1



RK2



Second Order Runge-Kutta method

Key Concept: Second Order Runge-Kutta Algorithm (midpoint)

to progress from a point at $t = t_0$, $y^*(t_0)$, by one time step, h, follow these steps (repetitively).

$k_1 = f(y^*(t_0),\ t_0)$	estimate of derivative at $t = t_0$
$y_1\left(t_0 + \frac{h}{2}\right) = y^*(t_0) + k_1 \frac{h}{2}$	intermediate estimate of function at $t = t_0 + \frac{h}{2}$
$k_2 = f\left(y_1\left(t_0 + \frac{h}{2}\right),\ t_0 + \frac{h}{2}\right)$	estimate of slope at $t = t_0 + \frac{h}{2}$
$y^*\left(t_0 + h\right) = y^*(t_0) + k_2 h$	estimate of $y\left(t_0 + h\right)$

Notes:
an initial value of the function must be given to start the algorithm.
this is often refered to as the "midpoint" algorithm for Second Order Runge-Kutta because it uses the slope at the midpoint, k_2.

Fourth Order Runge-Kutta method

we wisth to approximate the solution to a first order differential equation given by

$$\frac{dy(t)}{dt} = y'(t) = f(y(t), t), \qquad \text{with } y(t_0) = y_0$$

The development of the Fourth Order Runge-Kutta method closely follows those for the Second Order,

As with the second order technique there are many variations of the fourth order method, and they all use four approximations to the slope

Fourth Order Runge-Kutta method

We will use the following slope approximations to estimate the slope at some time t_0 (assuming we only have an approximation to $y(t_0)$ (which we call $y^*(t_0)$).

$$k_1 = f(y^*(t_0), t_0)$$

$$k_2 = f\left(y^*(t_0) + k_1\frac{h}{2}, t_0 + \frac{h}{2}\right)$$

$$k_3 = f\left(y^*(t_0) + k_2\frac{h}{2}, t_0 + \frac{h}{2}\right)$$

$$k_4 = f\left(y^*(t_0) + k_3 h, t_0 + h\right)$$

Fourth Order Runge-Kutta method

Each of these slope estimates can be described verbally.

k_1 **is the slope at the beginning of the time step (this is the same as** k_1 **in the first and second order methods).**

If we use the slope k_1 **to step halfway through the time step, then** k_2 **is an estimate of the slope at the midpoint. This is the same as the slope,** k_2**, from the second order midpoint method. This slope proved to be more accurate than** $\boldsymbol{k_1}$ **for making new approximations for** $\boldsymbol{y(t)}$**.**

If we use the slope k_2 **to step halfway through the time step, then** k_3 **is another estimate of the slope at the midpoint.**

Finally, we use the slope, $\boldsymbol{k_3}$**, to step all the way across the time step (to** $t_0 + h$**), and** k_4 **is an estimate of the slope at the endpoint.**

Fourth Order Runge-Kutta method

We then use a weighted sum of these slopes to get our final estimate of *$y^*(t_0+h)$*

$$y^*(t_0+h) = y^*(t_0) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} h = y^*(t_0) + \left(\frac{1}{6} k_1 + \frac{1}{3} k_2 + \frac{1}{3} k_3 + \frac{1}{6} k_4 \right) h$$

$$= y^*(t_0) + mh \qquad \text{where } m \text{ is a weighted average slope approximation}$$

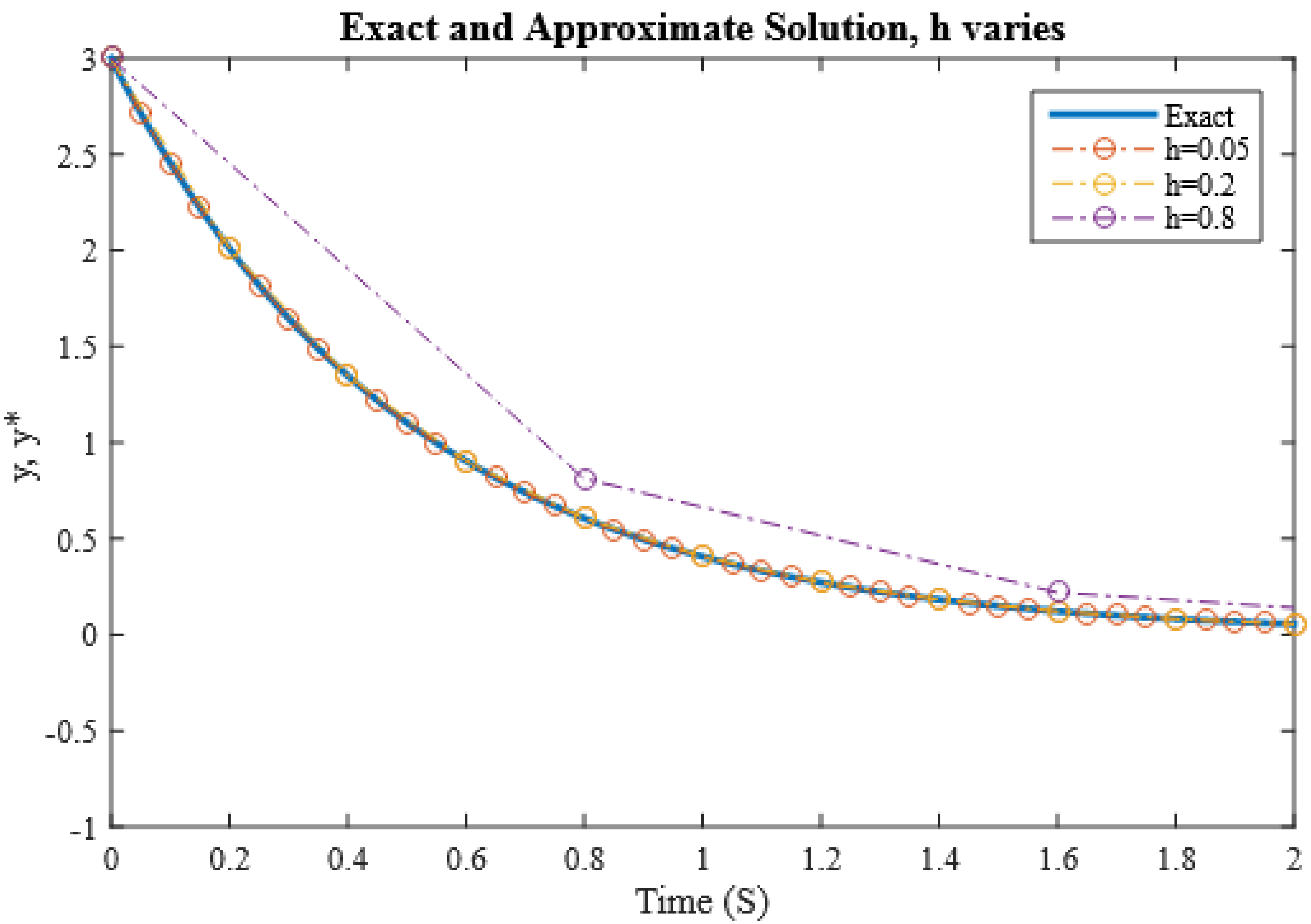
Fourth Order Runge-Kutta method

initial condition set as *y(0)=3*. To get from the initial value at *t=0* to an estimate at *t=h*, follow the procedure outlined below

$y'(0) = -2y(0)$	expression for derivative at $t = 0$
$k_1 = -2y(0)$	derivative at $t = 0$
$y_1\left(\frac{h}{2}\right) = y^*(0) + k_1\frac{h}{2}$	intermediate estimate of function at $t = h/2$ (using k_1)
$k_2 = -2y_1\left(\frac{h}{2}\right)$	estimate of slope at $t = h/2$
$y_2\left(\frac{h}{2}\right) = y^*(0) + k_2\frac{h}{2}$	anotherintermediate estimate of function at $t = h/2$ (using k_2)
$k_3 = -2y_2\left(\frac{h}{2}\right)$	another estimate of slope at $t = h/2$
$y_3(h) = y^*(0) + k_3 h$	an estimate of function at $t = h$ (using k_3)
$k_4 = -2y_3(h)$	estimate of slope at $t = h$
$y^*(h) = y(0) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}h$	estimate of $y(h)$

Fourth Order Runge-Kutta method

initial condition set as *y(0)=3*. To get from the initial value at *t=0* to an estimate at *t=h*, follow the procedure outlined below



Exercicos:

Implemente o esquema

Fourth Order Runge-Kutta method