Numerical Methods

ODEs. Dynamical Systems

- Numerical Solution of Ordinary Differential Equations

2023-2024

Numerical Solution of ODEs

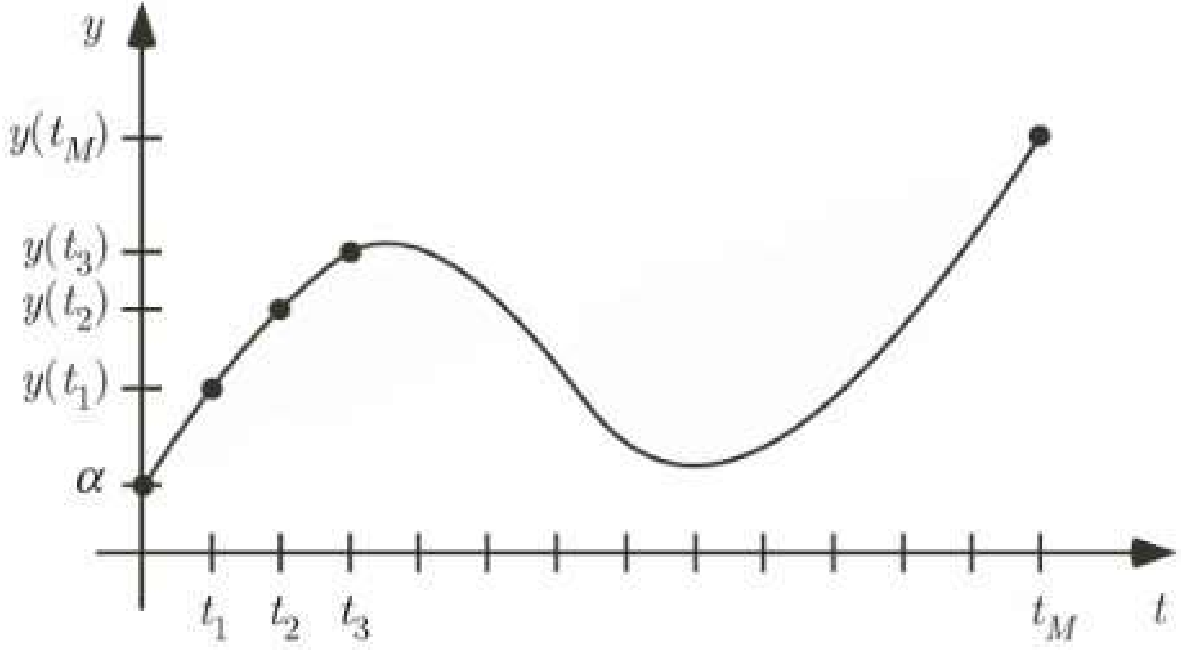
- The numerical solution of an IVP problems starts by discretizing the problem
- We are not interested in the analytical function $\boldsymbol{y}(t)$ which solves exactly the differential equation, we search instead for a tabulated solution

$$\{y_i = y(t_i)\}_{i\geq 0} \text{ on } t_{i+1} = t_i + h$$

- The value h is called the ***step***

Numerical Methods

Numerical Solution of ODEs



2023-2024

[DΣIM]

Classification of the Methods

- The methods of numerical solution can be classified in two groups:
  - ***One step methods***: The value y_{i+1} is obtained from the information in the former step (t_i, y_i)
  - ***Multistep methods***: They use information on several steps (t_{i-k}, y_{i-k}), where $k = 0, \dots, n$
- Another possible classification can be based in which methods use future information (***implicit*** methods) or not (***explicit*** methods)

Euler's Method

- This is the ***simplest*** one step method
- We start selecting N steps in a closed interval $[a, b]$

$$t_i = a + ih \quad \text{for each} \quad i = 0, 1, 2, \ldots, N$$

- With a step size of

$$h = \frac{b-a}{N}$$

Euler’s Method

- At step k, the differential equation may be written as:

$$\dot{y}(t_k) = f(t_k, y(t_k))$$

- The numerical derivative may be approximated by a forward derivation formula

$$\dot{y}(t_k) = \frac{y(t_{k+1}) - y(t_k)}{h} + \tau_k$$

- Where τ_k represents the error in this step

$$\tau_k = -\frac{h}{2}\ddot{y}(\eta_k)$$

Euler’s Method

- Then, we may write

$$y(t_{k+1}) - y(t_k) + h\tau_k = hf(t_k, y(t_k))$$

- The error term τ_j represents the error of our derivative approximation, proportional to h, $O(h)$. Thus, when the step $h \to 0$ we should be able to compute the new value with any desired precision. We say that ***the approximation is consistent***

Euler’s Method

- Numerically however, we are not able to deal with this truncation error, we just assume that it is small, and we write the recurrence

$$y_{k+1} = y_k + hf(t_k, y_k), \quad k = 0, 1, 2, \ldots, N-1$$

- Starting in the initial value $y_0 = \alpha$
- From now on we will use a simpler notation:

$$y(t_{k+1}) \approx y_{k+1}, \quad y(t_k) \approx y_k$$

Euler’s Method

- Consider for instance the following IVP.

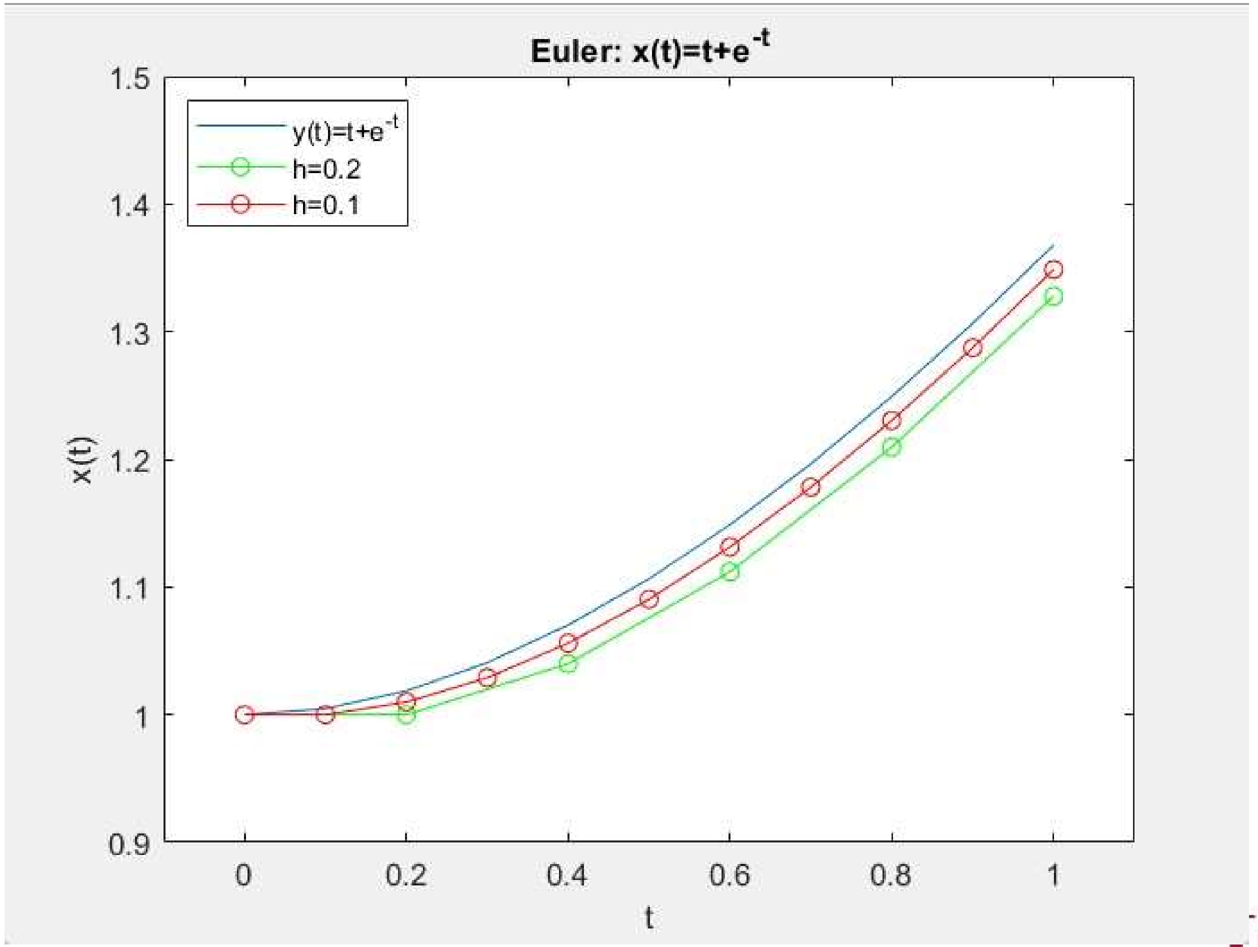
$$\dot{y} = -y + t + 1 \quad \text{where} \quad 0 \le t \le 1, \quad y(0) = 1$$

- With the known analytic solution

$$y(t) = t + e^{-t}$$

- We may compute the numerical solution using different steps.

Euler's Method



Euler's Method

- Consider now the following IVP

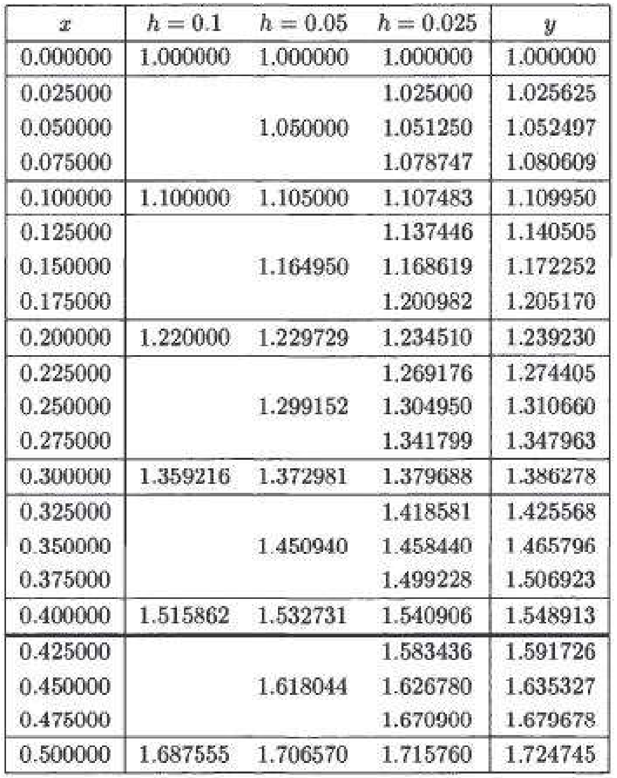
$$\dot{y}(t) = \frac{y+t}{y-t}, \quad y(0) = 1$$

- For which we know also the analytic exact solution.

$$y(t) = t + \sqrt{1+2t^2}$$

Euler’s Method

- We can check the effect of using different step values *h* in the following table:



x	$h = 0.1$	$h = 0.05$	$h = 0.025$	y
0.000000	1.000000	1.000000	1.000000	1.000000
0.025000			1.025000	1.025625
0.050000		1.050000	1.051250	1.052497
0.075000			1.078747	1.080609
0.100000	1.100000	1.105000	1.107483	1.109950
0.125000			1.137446	1.140505
0.150000		1.164950	1.168619	1.172252
0.175000			1.200982	1.205170
0.200000	1.220000	1.229729	1.234510	1.239230
0.225000			1.269176	1.274405
0.250000		1.299152	1.304950	1.310660
0.275000			1.341799	1.347963
0.300000	1.359216	1.372981	1.379688	1.386278
0.325000			1.418581	1.425568
0.350000		1.450940	1.458440	1.465796
0.375000			1.499228	1.506923
0.400000	1.515862	1.532731	1.540906	1.548913
0.425000			1.583436	1.591726
0.450000		1.618044	1.626780	1.635327
0.475000			1.670900	1.679678
0.500000	1.687555	1.706570	1.715760	1.724745

Errors

- When we want to compute numerically $y(t_k)$, we must distinguish the different values:
  - $y(t_k)$ is the exact value of the IVP problem when $t = t_k$
  - y_k is the exact solution of the recurrence relation when $t = t_k$
  - y_k^* is the numerical solution of the recurrence relation calculated by the computer when $t = t_k$.
- These values are associated to different computational errors.

Errors

- The ***global discretization error*** e_k is defined by

$$e_k = y(t_k) - y_k, \qquad k = 0,1,\ldots,M$$

- And is the difference between the unique solution and the solution obtained by the recurrence relation.
- The ***local discretization error*** ε_{k+1} is defined by

$$\varepsilon_{k+1} = y(t_{k+1}) - y^*_{k+1} = y(t_{k+1}) - y_k - h\Phi(t_k, y_k), \quad k = 0,1,\ldots,M-1$$

- Where $\Phi(t_k, y_k)$ represents the integration method used at step k. This is the error committed in a single step from t_k to t_{k+1}

Errors

- We need to have small local error terms:

$$\varepsilon_k = \left| y(t_k) - y_k^* \right|$$

- In a way that when we increase the number of steps N, or reduce the step h, the ***final global error***:

$$\varepsilon_f = \left| y(b) - y_N^* \right|$$

- Goes to zero or stays within the tolerance limits.

Euler's Method

- The final global error may be written as:

$$\varepsilon_f = \left| y(b) - y_N + y_N - y_N^* \right| \leq \left| y(b) - y_N \right| + \left| y_N - y_N^* \right|$$

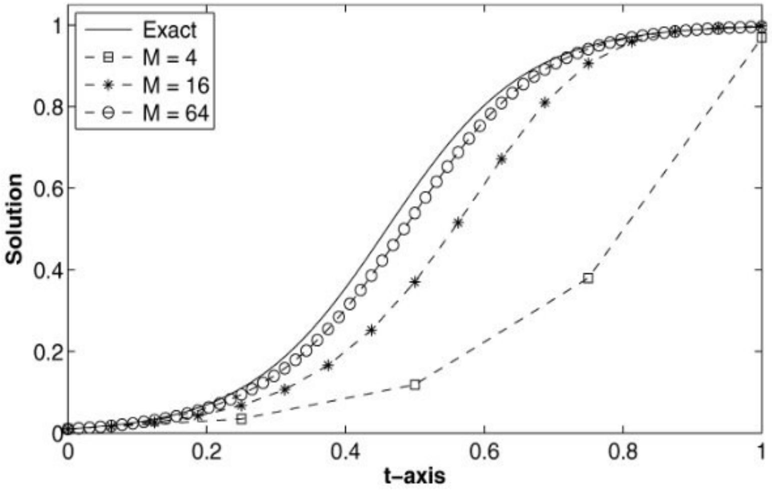
- We see that there are ***two sources for the error***. The first is due to the difference between the exact solution and the approximate numerical method. The second is due to the rounding errors of the floating-point operations.

Euler's Method

- When we apply this method to the logistic equation:

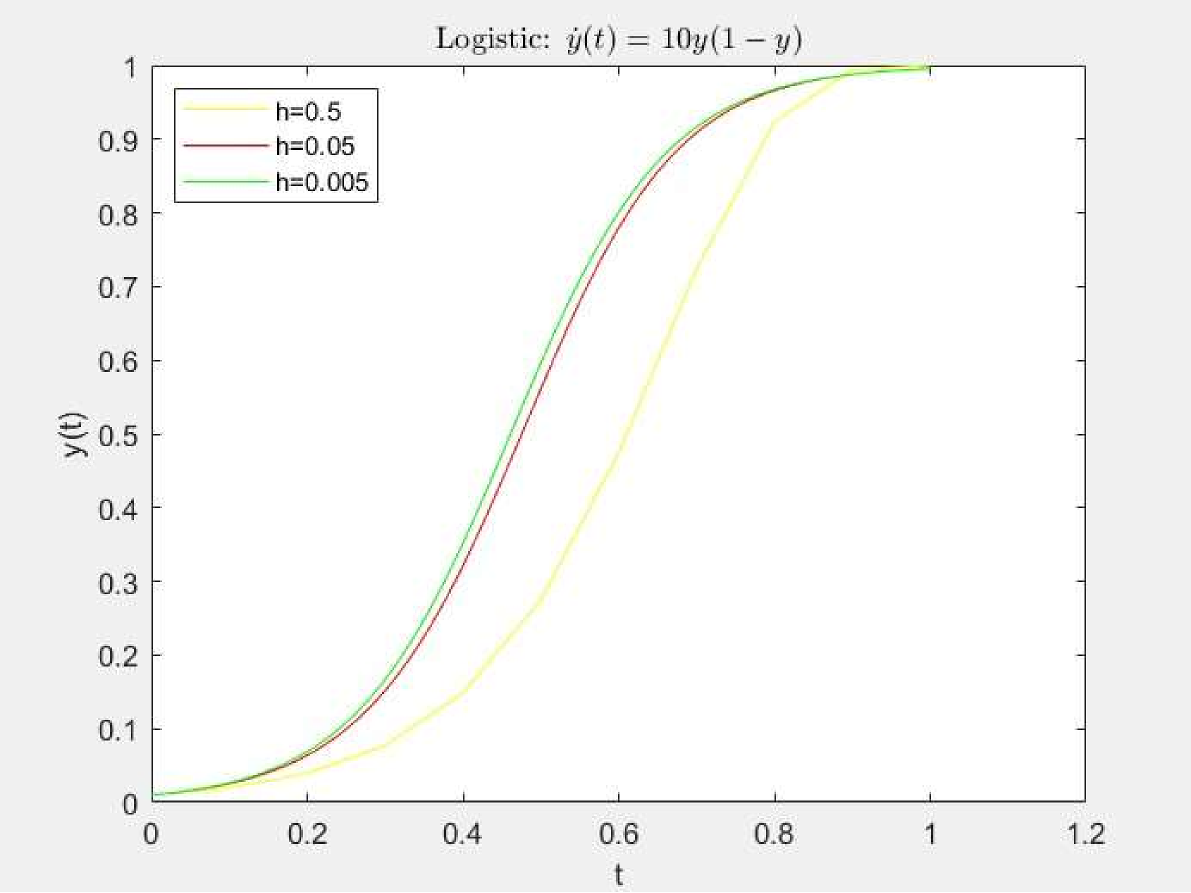
$$\dot{y}(t) = 10y(1-y), \quad 0 \leq t \leq 1$$
$$y(0) = 0.01$$

- Changing the step value *h=1/N* we can obtain the following trajectories.



Numerical Methods

Euler’s Method

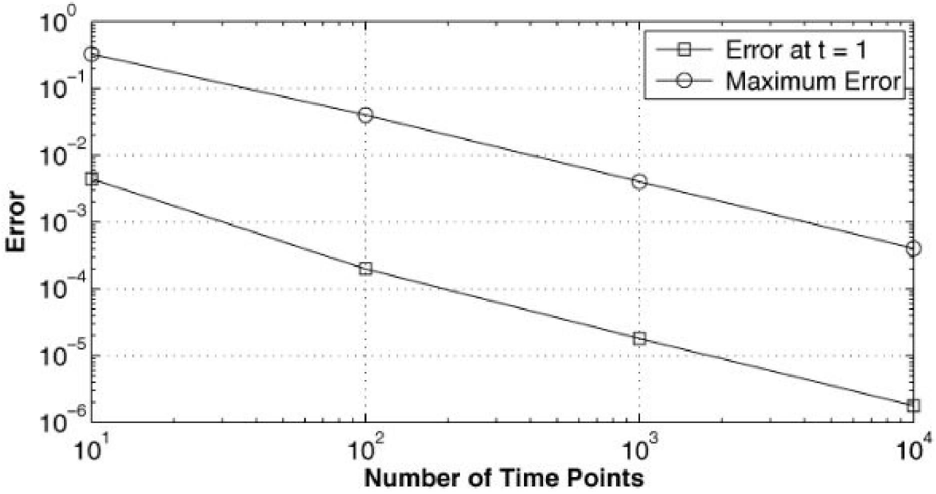


2023-2024

Euler’s Method

- For the example of the logistic equation, we can follow the evolution of the error when we increase the steps using Euler’s method:

$$\varepsilon_f = \left| y(b) - y_N^* \right|$$



Euler’s Method

- Note that the former graphic is in a logarithmic scale, The linear evolution tells us that the error goes as h^n with $n = 1$.
- This happens because the error in the derivative is of this size. In the recurrence relation the error term is of second order h^2
- But this will be the error of a single step and as there are a total of $N = 1/h$ steps, the final global error is $O(h)$

Euler's Method

- We may be interested also in the maximum error of the numerical approximation along the whole interval. This is:

$$\varepsilon_\infty = \max_{k=0,1,\ldots,M} \left| y(t_k) - y_k^* \right|$$

- In the example of the logistic equation, we can see that this error is bigger than the final error, but this error goes also as $O(h)$

Euler’s Method

- Euler’s method can also be obtained from the Taylor series of the solution $y(t)$. We have:

$$y(t_{k+1}) = y(t_k) + (t_{k+1} - t_k)\dot{y}(t_k) + \frac{(t_{k+1} - t_k)^2}{2}\ddot{y}(\xi) \quad \text{on} \quad \xi \in (t_k, t_{k+1})$$

$$= y(t_k) + h\dot{y}(t_k) + \frac{h^2}{2}\ddot{y}(\xi)$$

- Now, as $y(t)$ satisfies the differential equation,

$$\dot{y}(t) = f(t, y(t))$$

- Dropping second order terms we obtain:

$$\boxed{y_{k+1} = y_k + hf(t_k, y_k)}$$

Euler's Method

- The cumulative effect of numerical errors may be important. As the numerical solution goes on, we take values on the curves of nearby solutions. If we are in a region where nearby solutions diverge, we may obtain false solutions. We will need more accurate methods in these cases.

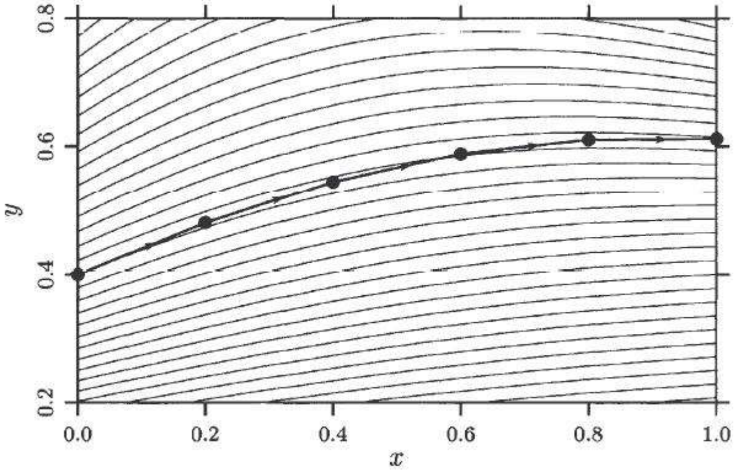


Figure 200(i) An example of the Euler method

Taylor's Methods

- Euler's method is just $O(h)$, as we are dropping second order terms in the Taylor expansion of the solution.

$$y(t_{k+1}) = y(t_k) + hf(t, y(t_k)) + \frac{h^2}{2}\ddot{y}(\xi)$$

- If we want to improve our method, we could consider the idea of adding more terms of the Taylor series

Taylor's Methods

- Thus, in the general expansion:

$$y(t_{k+1}) = y(t_k) + hf(t, y(t_k)) + \frac{h^2}{2}\ddot{y}(t_k) + \cdots$$

$$+\frac{h^n}{n!}y^{(n)}(t_k) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi_k)$$

- Considering that:

$$\begin{aligned}\dot{y}(t) &= f(t, y(t))\\ \ddot{y}(t) &= f\,'(t, y(t))\\ &\vdots\\ y^{(k)}(t) &= f^{(k-1)}(t, y(t))\end{aligned}$$

Taylor's Methods

- We can obtain the expression

$$y(t_{i+1}) = y(t_i) + hf(t, y(t_i)) + \frac{h^2}{2} f'(t_i, y(t_i)) + \cdots$$
$$+ \frac{h^n}{n!} f^{(n-1)}(t_i, y(t_i)) + \frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi_i)$$

- Where, dropping the last term we could obtain a method of order n. $O(h^n)$

Taylor's Methods

- Thus, in for the IVP

$$y' = -y + t + 1 \text{ on } 0 \le t \le 1, \; y(0) = 1$$

- We have

$$\begin{aligned} f'(t, y(t)) &= -y' + 1 = y - t \\ f''(t, y(t)) &= y' - 1 = -y + t \\ f'''(t, y(t)) &= -y' + 1 = y - t \end{aligned}$$

Taylor’s Methods

- For this example, we may write a Taylor method of second order:

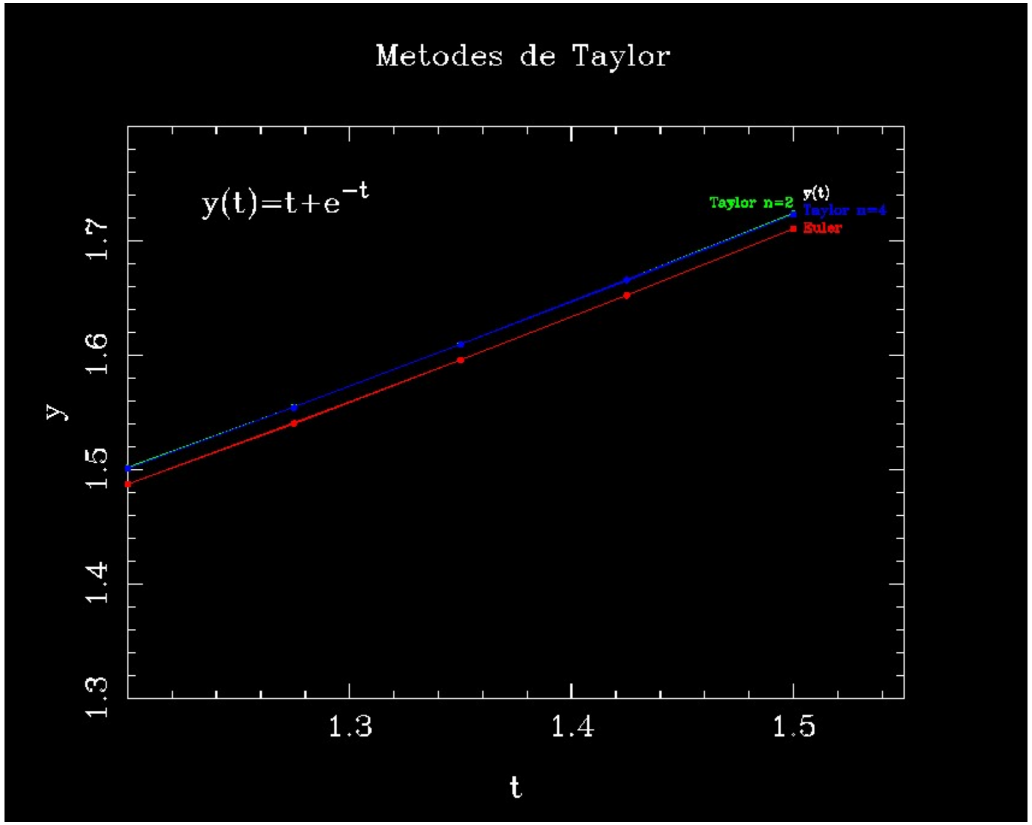
$$y_{i+1} = y_i + h\left[\left(1 - \frac{h}{2}\right)(t_i - y_i) + 1\right]$$

- Or a better method of order 4

$$y_{i+1} = y_i + h\left[\left(1 - \frac{h}{2} + \frac{h^2}{6} - \frac{h^3}{24}\right)(t_i - y_i) + 1\right]$$

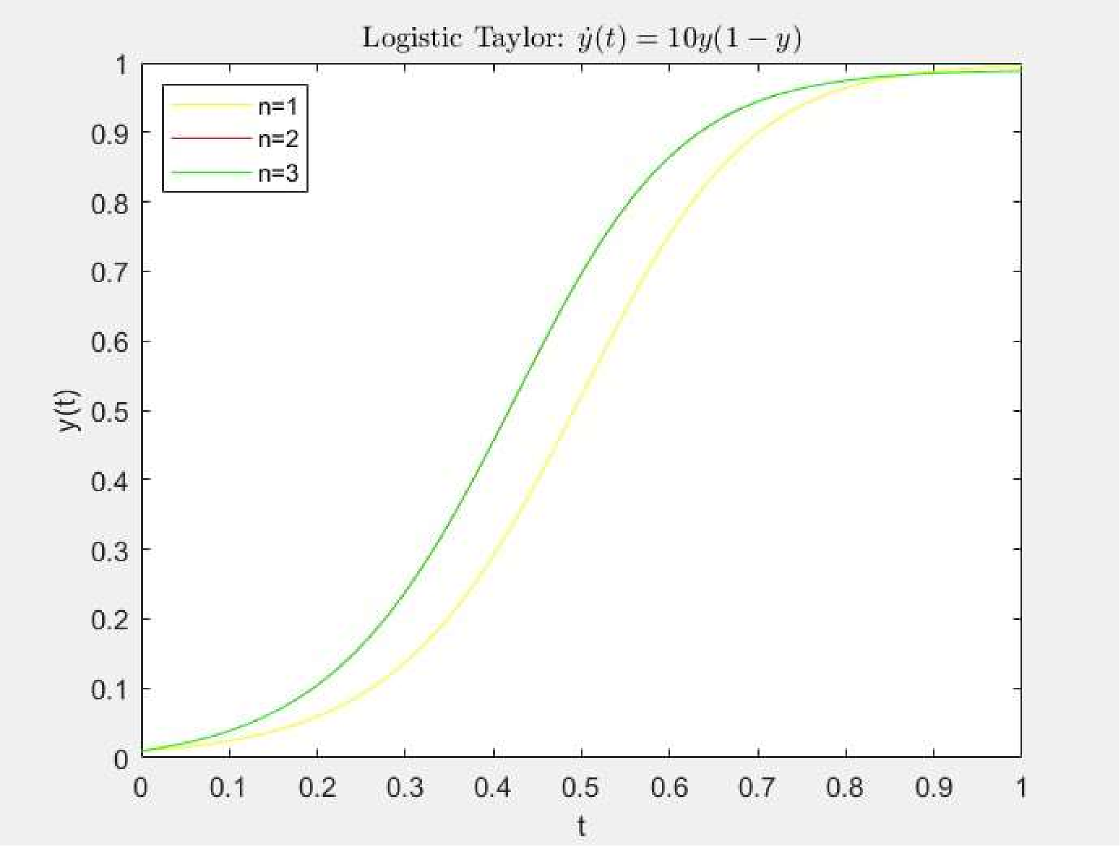
Numerical Methods

Taylor’s Methods



2023-2024

Taylor's Method



Stability

- If the theoretical error goes to zero, we can expect that the numerical solution will approximate the real solution better and better as we increase the number of steps. This property is called ***consistency***.
- This desirable property, however, is ***not enough***. Knowing the initial value y_0 precisely, we compute a following value y_1 which is not the real value $y(t_1)$. This difference will propagate to all the following computed values and in all timesteps

Stability

- These small differences should not be amplified as we go on with the numerical integration. This should be true, irrespective of the time at which some error is created.
- If our method can avoid the error amplification, we say that this is a ***stable method.***
- This stability can be measured with some degree. The strongest level corresponds to ***A-stability***. Other methods are just ***conditionally stable***.

Stability

- The stability of a given method is determined solving a linear equation like:

$$\frac{dy}{dt} = \dot{y} = -ry$$
$$y(0) = \alpha$$

- If the method gives bounded solutions irrespectively of the value of the parameter r and the step h the method is ***A-stable***. If this is true only for small values of h, the method ***conditionally stable.***

Stability

- When we apply Euler's method to the former linear method, we obtain the recurrence relation:

$$y_{i+1} = (1 + rh)y_i$$

- With the following general solution

$$y_i = \alpha(1 - rh)^j$$

- The real solution is however $y(t) = \alpha e^{-rt}$ which goes to zero with t. If this is to happen in Euler's method, we should have $|1 - rh| \leq 1$ and $h \leq 2/r$ so we can conclude that ***Euler's method is conditionally stable***.

Stability

- The importance of A-stable methods is that they guarantee the asymptotic behavior of the numerical solutions. Thus, numerical solutions behave as the analytical solutions for great values of t.
- Thus, consider a general equation of the form

$$\dot{y} = f(y)$$

- with an asymptotic solution $y = Y$

Stability

- Suppose that we start with the initial value:

$$y(0)=\alpha$$

- Which is very near the final asymptotic solution $y=Y$. The numerical solution of the IVP, $y(t)$, should also tend to $y=Y$ when $t\to\infty$. If we write $v(t)=y(t)-Y$. Using the initial ODE, we can write:

$$\dot{v}(t)=f(y)=f(Y+v)$$

Stability

- As the solutions $y(t)$ and Y should be very similar, using Taylor’s theorem to the former equation we can write

$$\dot{v}(t) = f(Y+v) = f(Y) + vf'(Y) + O(v^2)$$

- And dropping second order terms $O(v^2)$ we obtain the first order equation

$$\dot{v} = -rv \quad \text{on} \quad r = -f'(Y)$$

Stability

- The ODEs of our examples were single variable ODEs. All our former discussion is true also in the case of several variables with systems of differential equations:

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$$

- In this situation Euler’s method can be written also in vector form:

$$\mathbf{y}(t_{i+1}) = \mathbf{y}(t_i) + h\mathbf{f}(t, \mathbf{y}_\mathbf{i})$$

Stability

- But in this case, we may have a new problem. Not all the differential equations of the ODE system may behave in a similar way and may have different degrees of complexity.
- In some situations, it may be convenient to use different order approximations for the equations of the system to guarantee the stability at lower cost

Stability

- A particular case of ODE systems are the linear systems that can be written as:

$$\dot{\mathbf{y}} = -\mathbf{A}\mathbf{y}$$

- Where **A** is a constant coefficient matrix, and it is precisely from the generalization of these systems that we obtain the idea of A-stability.

Runge-Kutta Methods

- Using Taylor’s method, we can reach high precision if we use many terms in the expansion
- We need, however, the derivatives of the field $f(t, y(t))$ which will be different for each differential equation. Moreover, these derivatives may be difficult to compute.
- Runge-Kutta methods are high order methods that do not need to use the derivatives of the field.

Runge-Kutta Methods

- We will show the derivation of the second order Runge-Kutta method, but the standard methods are of higher order. They are derived in a similar way.
- The basic idea consists of finding constants a_1, α_1 and β_1 such as

$$a_1 f(t+\alpha_1, y+\beta_1)$$

- Approximates the second order Taylor term of $y(t)$

$$T^2(t,y) = f(t,y) + \frac{h}{2} f'(t,y)$$

- With an error of the order of $O(h^2)$

Runge-Kutta Methods

- We need to use Taylor's polynomial for a two-dimensional function. If $f(t,y)$ and all their derivatives of order lower or equal to *n+1* are continuous within

$$D = \{(t,y) \mid a \le t \le b, c \le y \le d\}$$

- Then

$$f(t,y) = P_n(t,y) + R_n(t,y)$$

Runge-Kutta Methods

- Where:

$$
\begin{aligned}
P_n(t,y) = f(t_0,y_0) &+ \left[(t-t_0)\frac{\partial f}{\partial t}(t_0,y_0) + (y-y_0)\frac{\partial f}{\partial y}(t_0,y_0) \right] \\
&+ \left[\begin{array}{l} \dfrac{(t-t_0)^2}{2}\dfrac{\partial^2 f}{\partial^2 t}(t_0,y_0) + (t-t_0)(y-y_0)\dfrac{\partial^2 f}{\partial t \partial y}(t_0,y_0) \\ + \dfrac{(y-y_0)^2}{2}\dfrac{\partial^2 f}{\partial^2 y}(t_0,y_0) \end{array} \right] \\
&+ \cdots + \left[\frac{1}{n!}\sum_{j=0}^{n} \binom{n}{j} (t-t_0)^{n-j}(y-y_0)^{j}\frac{\partial^n f}{\partial t^{n-j}\partial y^{j}}(t_0,y_0) \right]
\end{aligned}
$$

Runge-Kutta Methods

- The error term is:

$$R_n(t,y)=\left[\frac{1}{(n+1)!}\sum_{j=0}^{n+1}\binom{n+j}{j}(t-t_0)^{n+1-j}(y-y_0)^j\,\frac{\partial^{n+1}f}{\partial t^{n+1-j}\partial y^j}(t_0,y_0)\right]$$

- In our case

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

- And as

$$f'(t,y)=\frac{d}{dt}f(t,y(t))=\frac{\partial}{\partial t}f(t,y(t))+\frac{\partial}{\partial y}f(t,y(t))\cdot\frac{d}{dt}y(t)$$

Runge-Kutta Methods

- We have:

$$T^{(2)}(t,y) = f(t,y) + \frac{h}{2}\frac{\partial f(t,y)}{\partial t} + \frac{h}{2}\frac{\partial f(t,y)}{\partial y}\cdot f(t,y)$$

- We want to approximate this expression with a Taylor polynomial of second order

$$a_1 f(t+\alpha_1, y+\beta_1) = a_1 f(t,y) + a_1\alpha_1\frac{\partial f(t,y)}{\partial t} + a_1\beta_1\frac{\partial f(t,y)}{\partial y} + a_1 R_1(t+\alpha_1, y+\beta_1),$$

Runge-Kutta Methods

- Where the error term is:

$$R_1(t+\alpha_1, y+\beta_1)$$
$$=\left[\frac{\alpha_1{}^2}{2}\frac{\partial^2 f}{\partial^2 t}(\xi,\eta)+\alpha_1\beta_1\frac{\partial^2 f}{\partial t\partial y}(\xi,\eta)+\frac{\beta_1{}^2}{2}\frac{\partial^2 f}{\partial^2 y}(\xi,\eta)\right]$$

- Equating all the coefficients we have:

$$a_1 = 1$$
$$a_1\alpha_1 = \frac{h}{2}$$
$$a_1\beta_1 = \frac{h}{2}f(t,y).$$

Runge-Kutta Methods

- Solving these equations, we obtain:

$$a_1 = 1,\ \ \alpha_1 = \frac{h}{2},\ \text{i}\ \ \beta_1 = \frac{h}{2} f(t,y)$$

- And then:

$$T^{(2)}(t,y) = f\left(t + \frac{h}{2}, y + \frac{h}{2} f(t,y)\right) - R_1\left(t + \frac{h}{2}, y + \frac{h}{2} f(t,y)\right)$$

Runge-Kutta Methods

- With these constants the error term becomes:

$$R_1\left(t+\frac{h}{2}, y+\frac{h}{2}f(t,y)\right)=\frac{h^2}{8}\frac{\partial^2 f(\xi,\eta)}{\partial t^2}+\frac{h^2}{4}f(t,y)\frac{\partial^2 f(\xi,\eta)}{\partial t\partial y}$$
$$+\frac{h^2}{8}(f(t,y))^2\frac{\partial^2 f(\xi,\eta)}{\partial y^2}$$

- If the partial derivatives of f are bounded, the error of the method will be of $O(h^2)$

Runge-Kutta Methods

- The second order Runge-Kutta method has the specific name of ***midpoint method***

$$\boxed{y_{i+1} = y_i + hf\left(t_i + \frac{h}{2}, y_i + \frac{h}{2} f(t_i, y_i)\right)}$$

- As we evaluate the field f in the midpoint $y_i + h/2$ for the values $i = 0,1,\ldots,N-1$

Runge-Kutta Methods

- Since only three parameters are present in:

$$a_1 f(t+\alpha_1, y+\beta_1)$$

- And all are needed for the match of $T^{(2)}$, we need a more complicated form to satisfy the conditions required for any of the higher-order Taylor methods. The most appropriate form for approximating:

$$T^{(3)}(t,y) = f(t,y) + \frac{h}{2} f'(t,y) + \frac{h^2}{6} f''(t,y)$$

- is

$$a_1 f(t,y) + a_2 f(t+\alpha_2, y+\delta_2 f(t,y))$$

Runge-Kutta Methods

- However, this expression do not give sufficient flexibility to match the next term:

$$\frac{h^2}{6}\left[\frac{\partial f}{\partial y}(t,y)\right]^2 f(t,y)$$

- And we can not get better precision than $O(h^2)$ with these approximations. The new parameter, however, gives some flexibility in the choice of the constants and this gives to more methos of $O(h^2)$ precision.

Runge-Kutta Methods

- And we obtain just the modified Euler's method

$$a_1 = a_2 = \frac{1}{2} \text{ i } \alpha_2 = \delta_2 = h$$

$$y_0 = \alpha_0$$

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + h, y_i + hk)$$

$$y_{i+1} = y_i + \frac{h}{2}\left[k_1 + k_2\right]$$

Runge-Kutta Methods

- And the Heun method

$$a_1 = \frac{1}{4}, a_2 = \frac{3}{4} \quad \text{i} \quad \alpha_2 = \delta_2 = \frac{2}{3}h$$

$$y_0 = \alpha_0$$

$$k_1 = f(t_i, y_i)$$

$$k_2 = f\left(t_i + \frac{2}{3}h, y_i + \frac{2}{3}hk_1\right)$$

$$y_{i+1} = y_i + \frac{h}{4}\left[k_1 + 3k_2\right]$$

Runge-Kutta Methods

- ***Modified Euler method:***

$$\boxed{y_{i+1} = y_i + \frac{h}{2}\left[f(t_i, y_i) + f(t_{i+1}, y_i + hf(t_i, y_i))\right]}$$

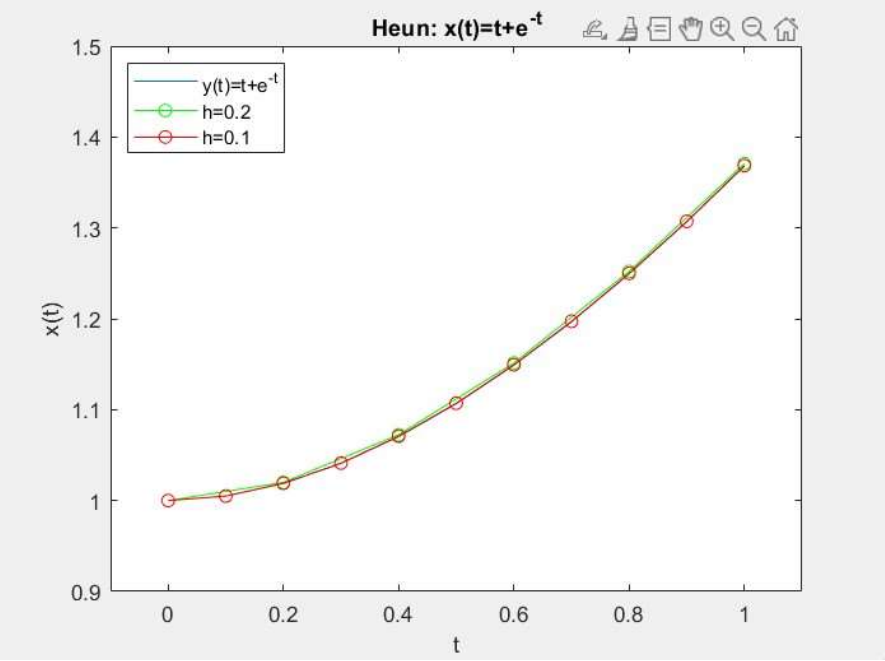
- And ***Heun's method:***

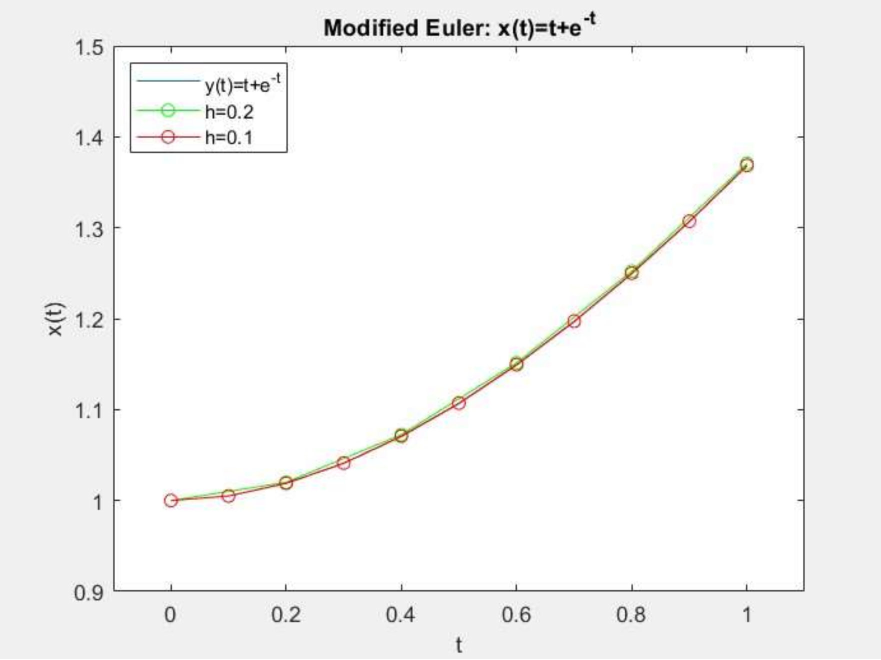
$$\boxed{y_{i+1} = y_i + \frac{h}{4}\left[f(t_i, y_i) + 3f\left(t_i + \frac{2}{3}h, y_i + \frac{2}{3}hf(t_i, y_i)\right)\right]}$$

- Both are considered Runge-Kutta methods of $O(h^2)$

Numerical Methods

Runge-Kutta Methods





2023-2024

RK4

- The method that is commonly used for simple differential systems is a method of fourth order which is described by the set of equations:

$$y_{i+1} = y_i + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

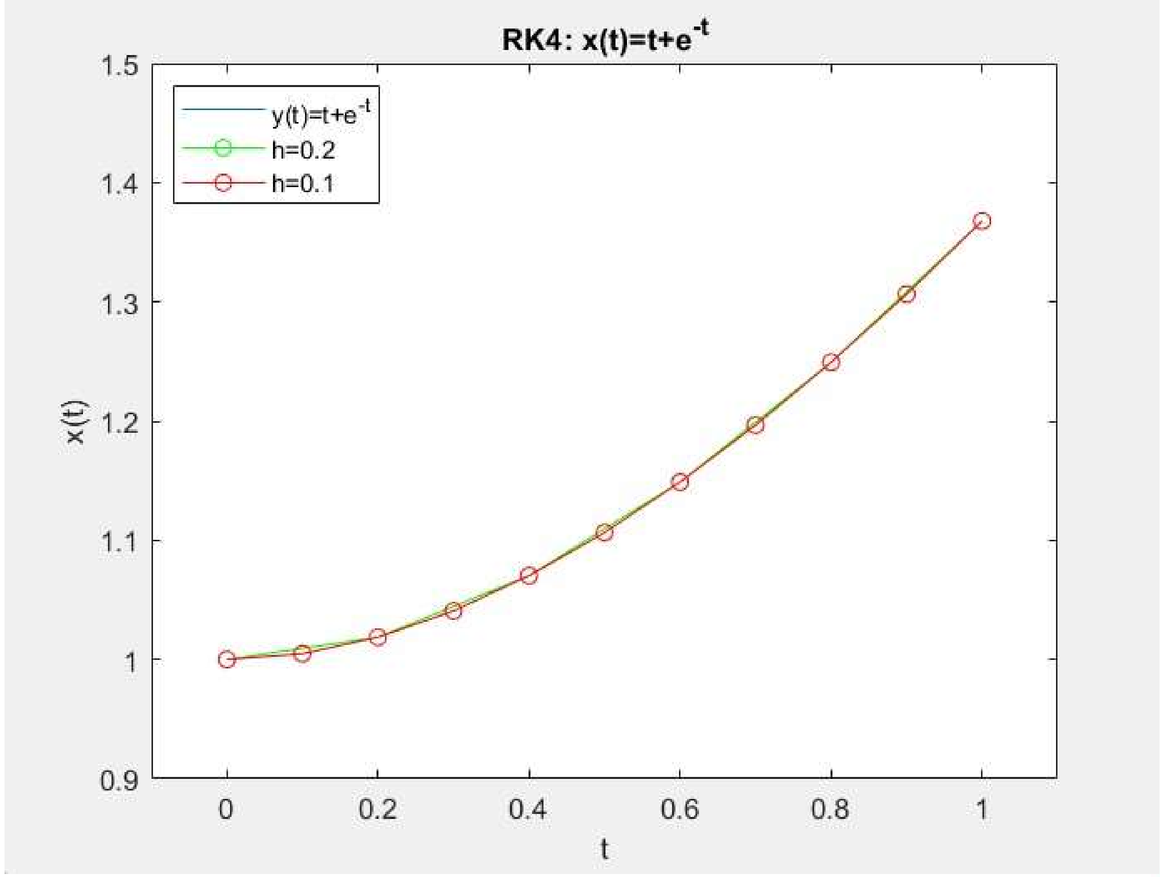
$$k_1 = f(t_i, y_i)$$

$$k_2 = f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_i + h, y_i + hk_3)$$

RK4



Runge-Kutta Methods

- The bigger computational effort is in the computation of $f(t,y)$. There is a relation between the number of evaluations and the order of the methods.

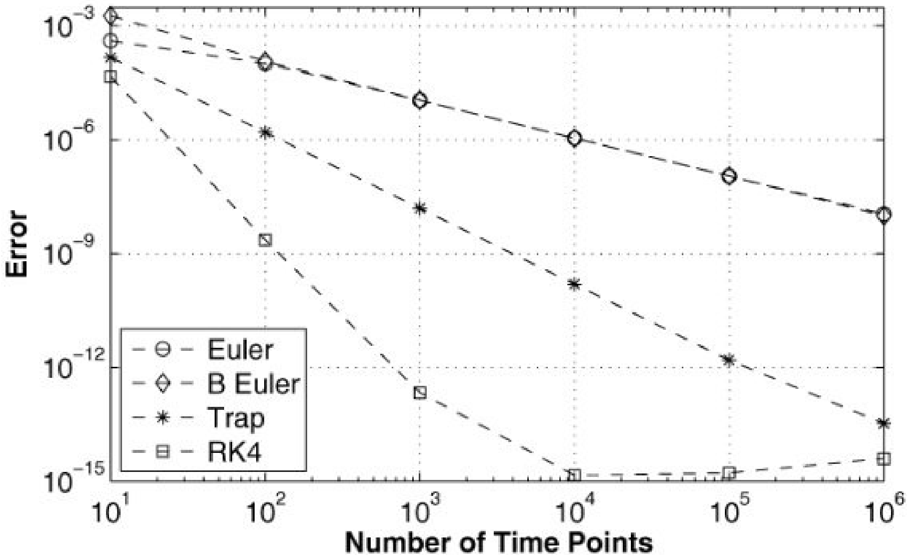
#N	2	3	4	5≤n≤7	8≤n≤9	10≤n
Order	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^{n-1})$	$O(h^{n-2})$	$O(h^{n-3})$

Runge-Kutta Methods

- We observe that the number of evaluations increases more rapidly than the order or the precision of the method. At some point, it may be better to use a lower order method with a smaller step than a higher method with a longer step.
- Using the RK4 we need 4 evaluations of the field per step, and we can obtain a better precision than with an Euler method using a step h smaller than $h/4$ in Euler's method.

Runge-Kutta Methods

- In the following figure we can compare the effect of increasing the number of steps or decreasing the value of *h* on the total error

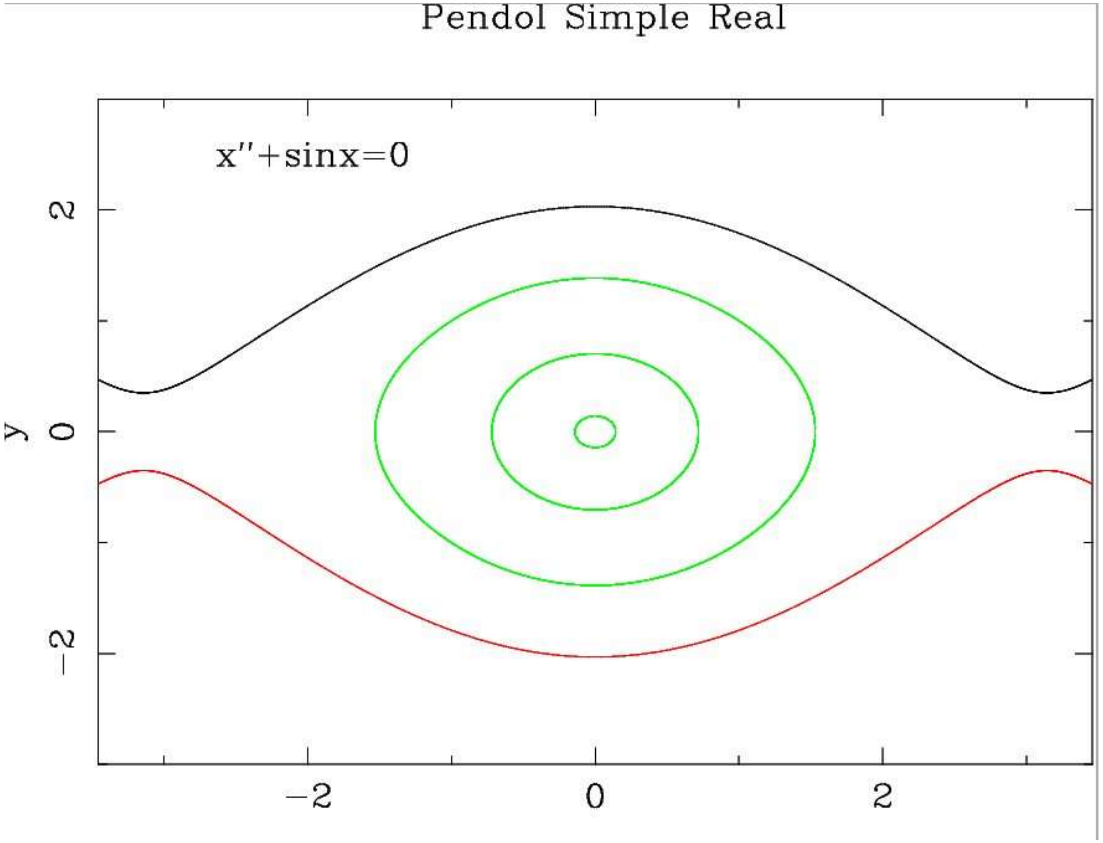


Classical Pendulum

- Consider the following differential system corresponding to the free classical pendulum:

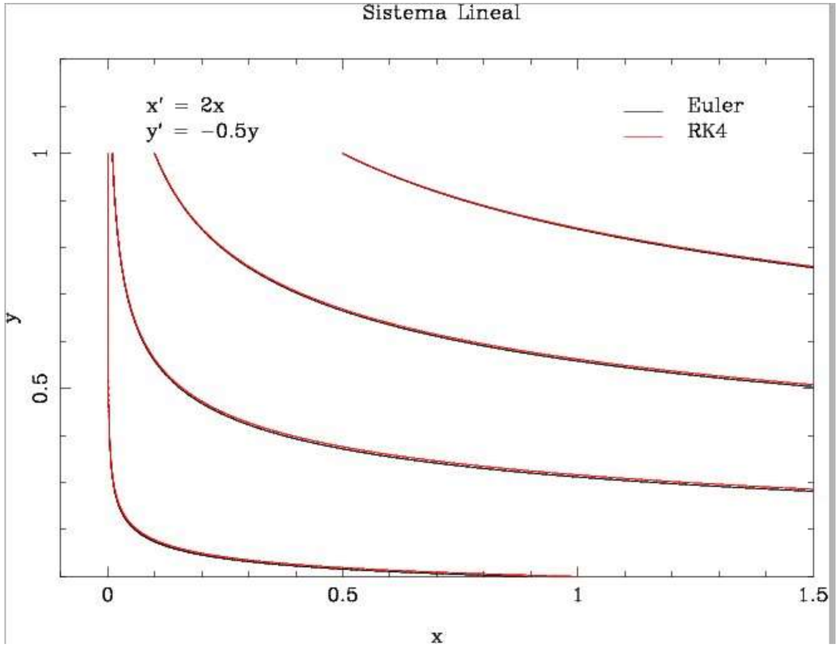
$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sin(x_1) \end{aligned}$$

Classical Pendulum



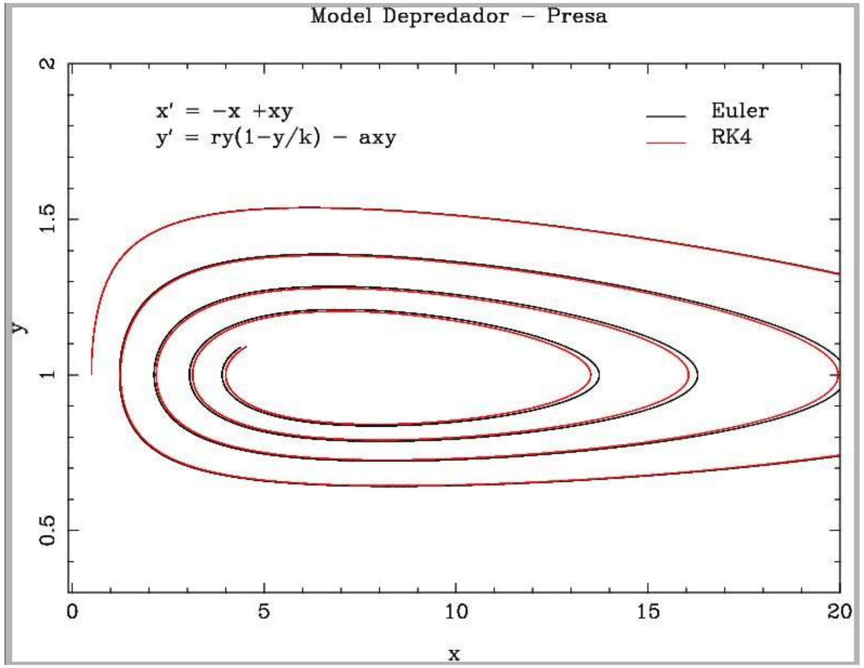
Comparing Euler – RK4

- Linear system



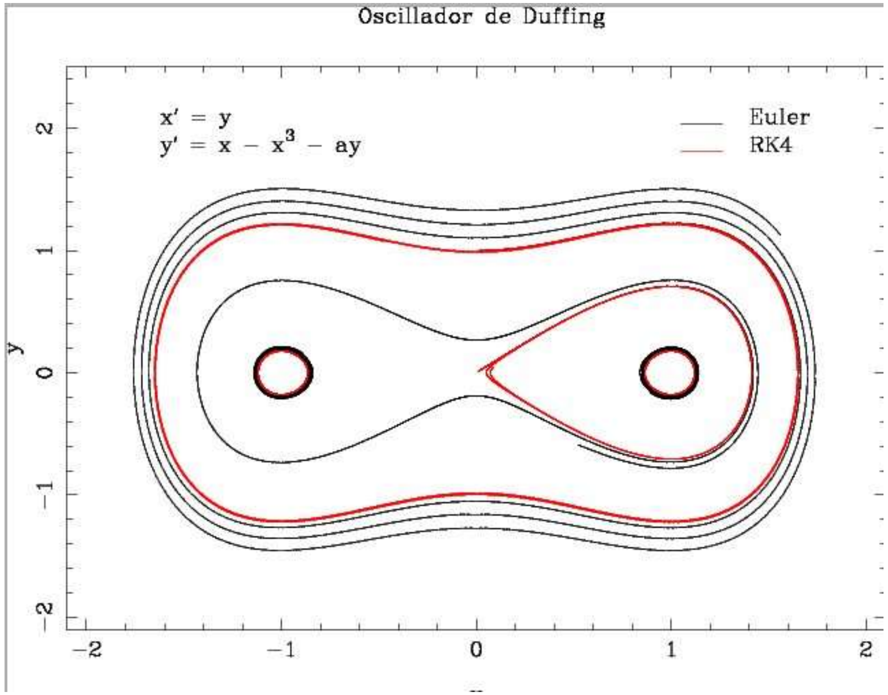
Comparing Euler – RK4

- System Predator-Prey



Comparing Euler – RK4

- Duffing oscillator


Oscillador de Duffing
$x' = y$
$y' = x - x^3 - ay$
Euler
RK4
y
x