

Universidade Federal de Santa Catarina

Departamento de Física

Física Geral 4

# **Método de Runge-Kutta aplicado a EDOs de circuitos**

Aluno: Muryel Guolo Pereira

Florianópolis

2016

# 1 O Método

## 1.1 Equação Diferencial Ordinária

São equações cujas incógnitas são funções de uma variável e suas derivadas. A ordem de uma EDO é determinada pelo grau da maior derivada da mesma, para a EDO de primeira ordem pode ser escrita como:

$$x'(t) = f(x, t)$$

onde  $x'(t)$  é a primeira derivada de  $x$  em relação a  $t$ ,  $\frac{dx}{dt}$ .

Já uma EDO de segunda ordem pode ser escrita como:

$$x''(t) + x'(t) = f(x, t)$$

onde  $x'(t)$  é a primeira derivada de  $x$  em relação a  $t$ ,  $\frac{dx}{dt}$ , e  $x''(t)$  é a segunda derivada de  $x$  em relação a  $t$ ,  $\frac{d^2x}{dt^2}$ .

A solução para uma equação diferencial é uma função que satisfaça a relação entre ela e suas derivadas dada pela EDO. Tome como exemplo a EDO de primeira ordem,  $x'(t) = -\sin(t)$ , a solução da EDO será uma função da variável  $t$ , que satisfaça a condição de ter sua primeira derivada  $-\sin(t)$ , portanto a função solução,  $x(t)$ , que satisfaz tal relação é  $x(t) = \cos(t)$ .

Existem problemas de equações diferenciais, chamados de PVI (problemas de valor inicial), onde a função solução além de satisfazer a equação diferencial ainda deve satisfazer um valor inicial tal como  $x(t_0) = x_0$ .

## 1.2 Soluções Numéricas para EDOs

Certas EDOs podem ser difíceis de serem resolvidas analiticamente, achando uma função que satisfaça-a, ou no melhor dos casos necessitam que seja dado um 'chute' de uma possível solução, a partir de conhecimentos prévios de equações similares. Para resolver esse problema desenvolveram-se métodos numéricos para resolução de EDOs onde a solução é aproximada numericamente, ponto a ponto, sem que seja necessário achar uma função analítica que satisfaça a equação.

Um dos primeiros e mais simples métodos de resolução de EDOs é o método de Euler. Seja um PVI de primeira ordem definido como:

$$y' = f(x, y)$$

$$y(x_0) = y_0$$

Para se aproximar o valor de  $y_j$  para as soluções exatas  $y(x_j)$  com  $j = 1, 2, 3, \dots, m$  procura-se inicialmente um  $y_1$ , traçando-se uma tangente  $T$  à curva  $y(x)$  no ponto

$(x_0, y(x_0))$ , como na Figura 1, cuja equação é:

$$y(x) = y(x_0) + (x - x_0)y'(x_0)$$

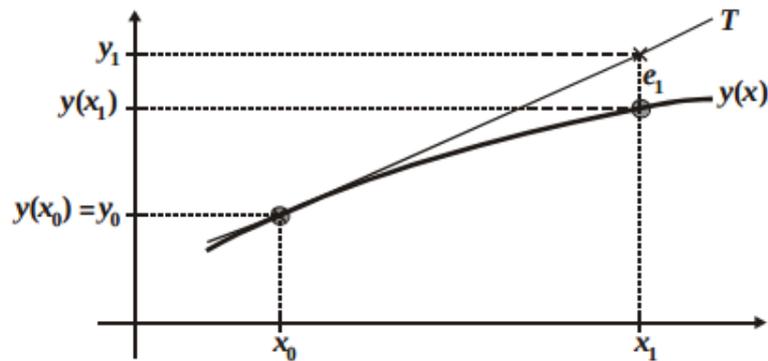


Figura 1: Aproximação do próximo valor de  $y$  tomando a tangente no ponto anterior (GALVÃO; NUNES, 2010).

Fazendo  $x = x_1$  e lembrando que  $y(x_0) = y_0$ ,  $x_1 - x_0 = h$ ,  $y'(x_0) = f(x_0, y(x_0))$  e  $y_1 \simeq y(x_1)$ , tem-se que:

$$y_1 = y_0 + h \cdot f(x_0, y(x_0)),$$

que pode ser generalizado para  $y_{j+1}$  como:

$$y_{j+1} = y_j + h \cdot y'(x_j, y_j)$$

Seguindo tal relação, sucessivamente, teremos valores aproximados para a função num dado intervalo escolhido. O que ocorre é que a cada iteração, o erro acumulado da iteração anterior é somado ao erro da iteração atual, fazendo que quanto maior o número de iterações maior é a diferença entre a solução numérica e a solução analítica, como pode ser visto da Figura 2.

Para resolver tal problema, diminuindo o erro acumulado, outros métodos mais precisos precisaram ser desenvolvidos, como é o caso dos métodos de Runge-Kutta.

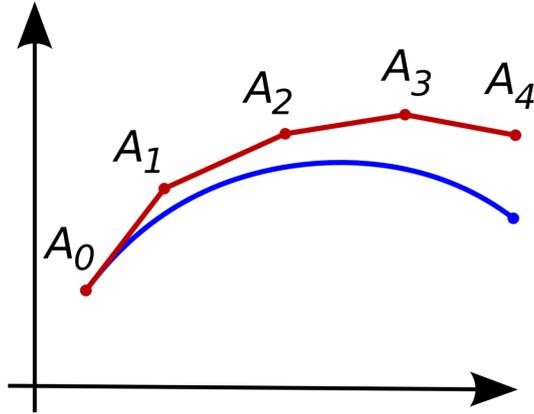


Figura 2: Erro da solução numérica em relação a analítica para o método de Euler(WIKIPEDIA, 2016).

### 1.3 Métodos de Runge-Kutta

Os métodos de Runge-kutta surgem da expansão de uma função, contínua e com  $k+1$  derivadas, em uma série pelo polinômio de Taylor com resto. Onde a ordem do método é numero de termos da serie que consideramos.

Podemos escrever uma função  $f(x)$ , que satisfaça ao condições citadas acima, como:

$$y(x) = y(a) + y'(a)\frac{x-a}{1!} + \dots + y^{(k)}(a)\frac{(x-a)^k}{k!} + y^{(k+1)}(a)\frac{(x-a)^{(k+1)}}{(k+1)!}$$

Substituindo  $a$  por  $x_n$  e  $x$  por  $x_{n+1} = x_n + h$ , a fórmula acima se torna:

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!}y''(x_n) + \dots + y^{(k)}(x_n)\frac{(h)^k}{k!} + y^{(k+1)}(x_n)\frac{(h)^{(k+1)}}{(k+1)!}$$

Se  $k=1$  e o resto  $\frac{h^2}{2!}y''(x_n)$  for pequeno, obtemos a formula do método de Euler:

$$y_{n+1} = y_n + hy'(n) = y_{n+1} = y_n + hf(x_n, y_n)$$

Portanto Runge-Kutta de primeira ordem é o método de Euler. Fazendo  $k=2$  temos:

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!}y''(x_n) + \frac{h^3}{3!}y'''(x_n)$$

Podemos escrever como:

$$y(x_{n+1}) = y(x_n)\phi(x_n, y_n; h)$$

onde  $\phi = ak_1 + bk_2$ . Pode-se mostrar, considerando  $\frac{h^3}{3!}y'''(x_n)$  pequeno e determinando  $aeb$ , que essa fórmula se resume em:

$$y_{n+1} = y_n + \frac{hf(x_n, y_n) + f(x_{n+1}, y_{n+1}^*)}{2}$$

Sendo esse o método de Runge-Kutta de segunda ordem, onde  $y_{n+1}^* = y_n + hf(x_n, y_n)$ .

O mesmo processo pode ser aplicado para uma ordem  $k$  para obter o método de Runge-Kutta de ordem  $k$ , mas normalmente o método mais utilizado é de quarta ordem, devido ao fato que para ordens superiores a diferença nos resultados é tão pequena que o custo computacional não a compensa.

#### 1.4 Uma Visão geométrica do método de Runge-Kutta de quarta ordem

Suponha o PVI  $x'(t) = f(t, x)$  onde  $x(t_0) = x_0$ . A partir disso primeiramente aplicamos a derivada ao ponto inicial  $(t_0, x_0)$ , tendo então  $x'(t_0) = f(x_0, t_0)$ , resultando na reta tangente a função  $x(t)$  no ponto  $t_0$ , como mostrado na Figura 3.

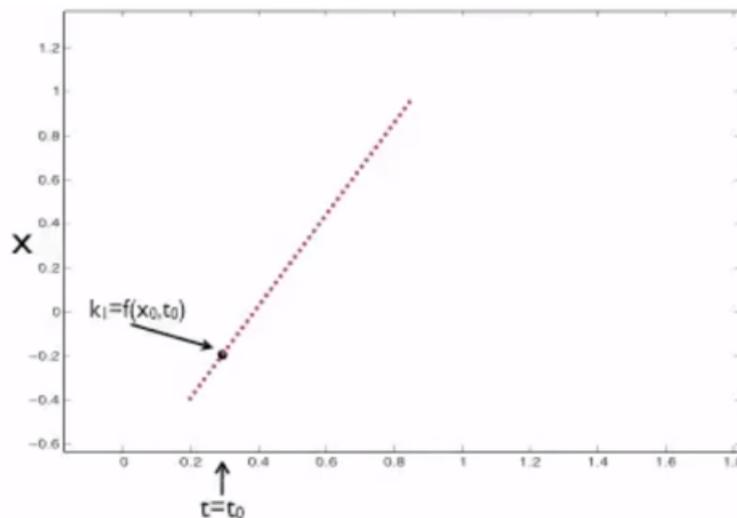


Figura 3: Equação  $x'(t)$  aplicada em  $(t_0, x_0)$ (SHIRTS, 2015).

A partir da derivada no ponto  $(t_0, x_0)$  utilizando-se do método de Euler, calcula-se um segundo ponto, dado por  $x(t_0 + \Delta t) = x_0 + \Delta t \cdot x'(x_0, t_0)$ .

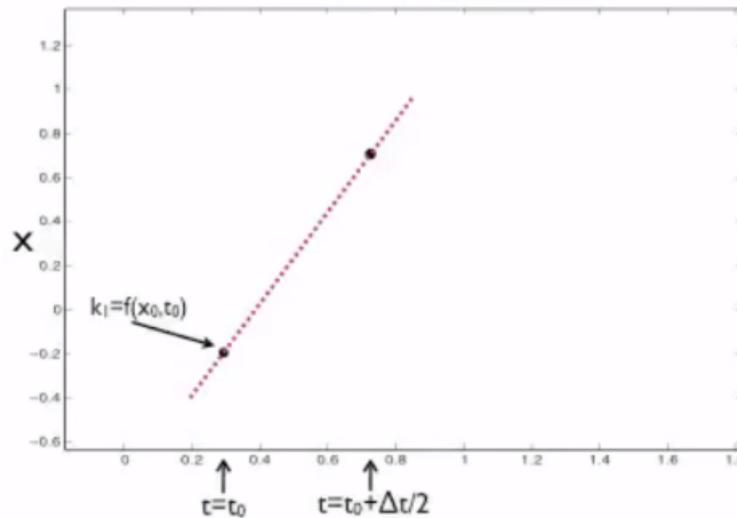


Figura 4: Segundo ponto calculado pelo método de Euler(SHIRTS, 2015)

Aplica-se novamente a equação  $x'(t)$  ao segundo ponto para encontrar uma segunda reta tangente.

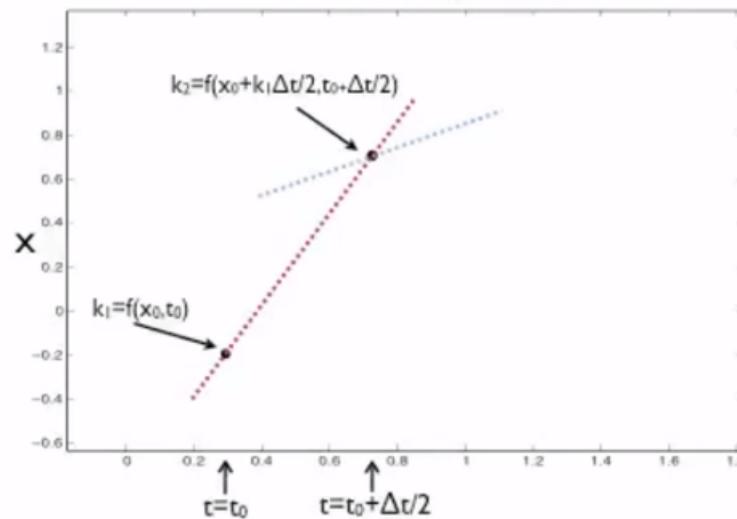


Figura 5:  $x'(t)$  aplicada ao segundo ponto(SHIRTS, 2015)

Aplicando novamente o método de Euler com a equação da segunda reta tangente ao ponto inicial  $(t_0, x_0)$ , achamos um terceiro ponto, como na Figura 6.

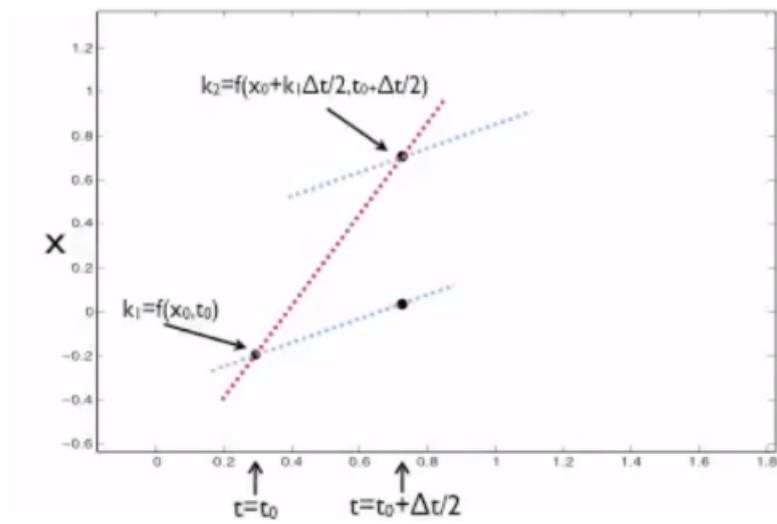


Figura 6: Terceiro ponto aplicando Euler com a equação da segunda reta tangente ao ponto inicial(SHIRTS, 2015).

Aplicando a equação  $x'(t)$  ao terceiro ponto, obtemos uma terceira reta tangente.

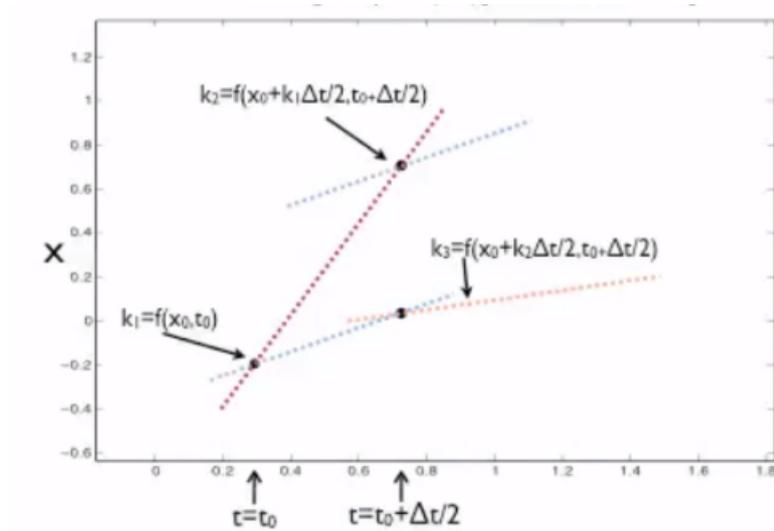


Figura 7: Terceira reta tangente(SHIRTS, 2015).

Novamente utilizando Euler no ponto inicial com a equação da terceira reta tangente, obtemos um quarto ponto.

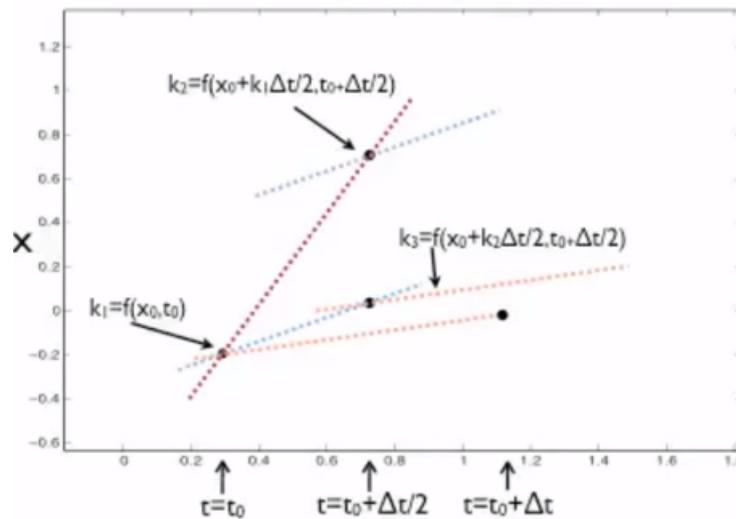


Figura 8: Quarto Ponto(SHIRTS, 2015).

Por último, com o mesmo processo, obtemos uma quarta reta tangente.

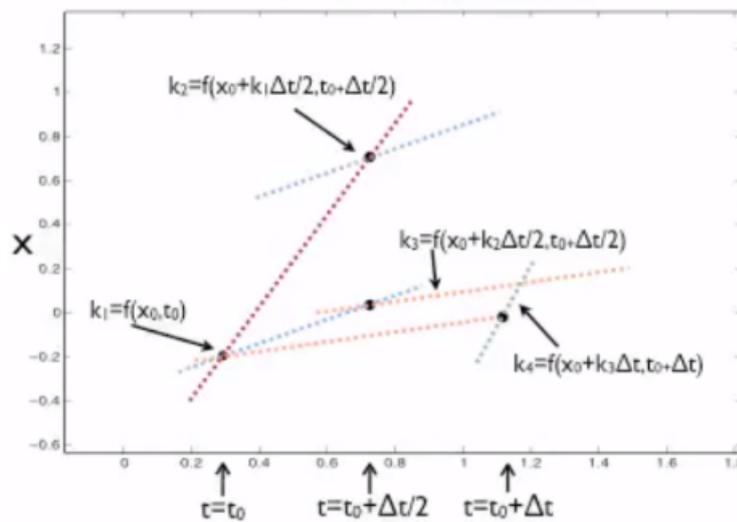


Figura 9: Quarta reta tangente(SHIRTS, 2015).

Ao final do processo, tira-se a média ponderada dos valor de inclinação das quatro reta resultantes, tendo assim uma reta tangente de maior qualidade, e aplicando o método

de Euler ao ponto inicial utilizando-se dessa média das inclinações, obtemos finalmente o valor da função solução no próximo ponto, e o processo se repete a cada iteração.

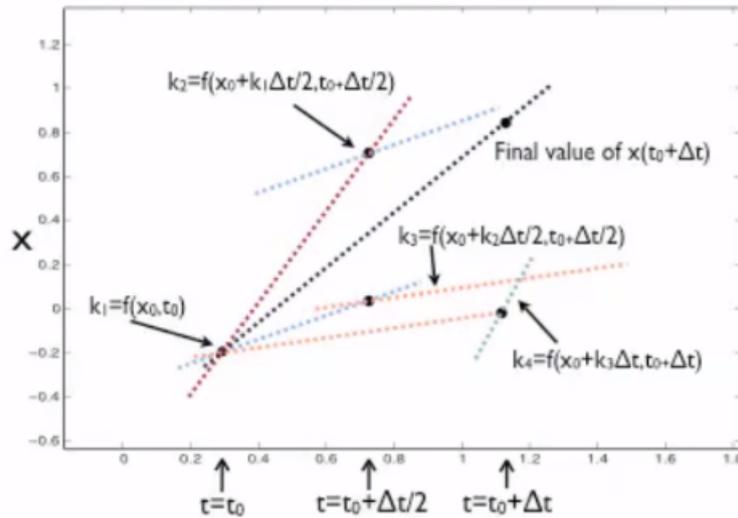


Figura 10: Valor de  $x(t)$  ao fim de uma iteração (SHIRTS, 2015).

O método de Runge-Kutta de quarta ordem pode ser escrito pelas seguintes equações:

$$x_{n+1} = x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$t_{n+1} = t_n + h$$

onde  $x_{n+1}$  é a aproximação por RK4 de  $x(t_{n+1})$  e

$$k_1 = f(t_n, x_n)$$

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

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

$$k_4 = f(t_n + h, x_n + hk_3)$$

## 2 Script generalizado em Python

```
1 import matplotlib.pyplot as plt
2 import math
3
4 def dxdt(T, X):
5     return math.cos(T)
6
7 def RK4(fx, x0, t0, h, n):
8     x= []
9     t= []
10    i=1
11    t.append(t0)
12    x.append(x0)
13    x[0]= x0
14    for i in range(1,n-1):
15        t.append((t[i-1]+h))
16        k1= fx(t[i-1], x[i-1])
17        k2= fx(t[i-1] + h/2, x[i-1] + h/2)
18        k3= fx(t[i-1] + h/2, x[i-1]+ (h/2)*k2)
19        k4= fx(t[i-1]+ h, x[i-1]+ h*k3)
20        x.append(x[i-1] + (h/6)*(k1 + 2*k2 + 2*k3 + k4))
21    return x,t
22
23
24
25 x0= 0
26 t0= 0
27 h=0.1
28 n= 100
29 x, t= RK4(dxdt, x0, t0, h, n)
30
31 plt.plot(t,x)
32 plt.xlabel("t")
33 plt.ylabel("x(t)")
34 plt.show()
```

Figura 11: Script para a equação  $y' = \cos(t)$ . Obs.: o arquivo .py segue em anexo

Na função `dxdt()` a equação diferencial é definida, no exemplo exemplo,  $\cos(T)$ .

A função `RK4()`, retorna duas listas com os valores da função  $X(T)$  e os valores de  $T$  associados e tem como parâmetros:

**fx:** a função da equação diferencial;

**x0:** o valor inicial de  $x$  ou  $x(t_0)$ ;

**t0:** o valor inicial de  $t$ ;

**h:** é o intervalo de repetição de  $t$ , ou seja  $t[n + 1] - t[n]$ ;

**n:** é o numero de passos a serem repetidos, e vai determinar o intervalo da função.

O resto da aplicação é responsável por determinar as constantes desejadas, chamar a função `RK4()` usando a função `dxdt()` como parâmetro, e plotar o resultado da função  $x(t)$  para o intervalo de  $(t_0, hn)$ .

O exemplo de script acima com  $y' = \cos(t)$  resulta na seguinte solução numérica:

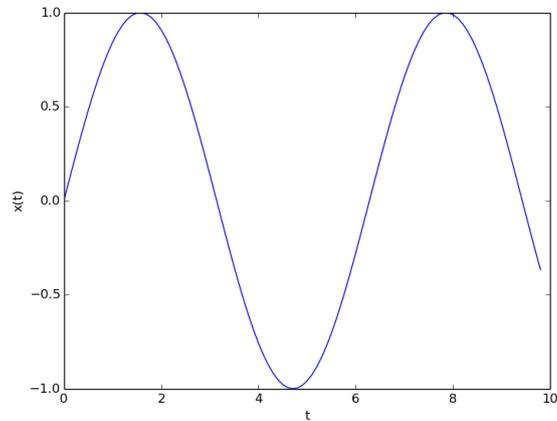


Figura 12: Resultado para  $y' = \cos(t)$

Para avaliarmos a eficiência do método, podemos observar a Figura 13, onde temos,  $\sin(t)_{RK} - \sin(t)$ .

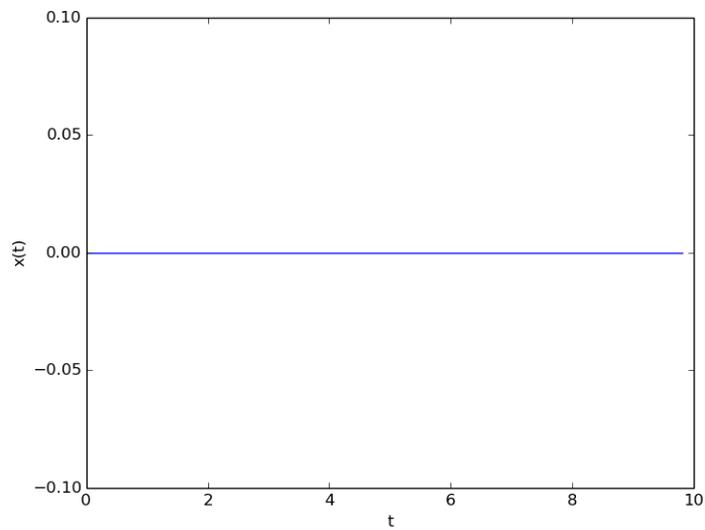
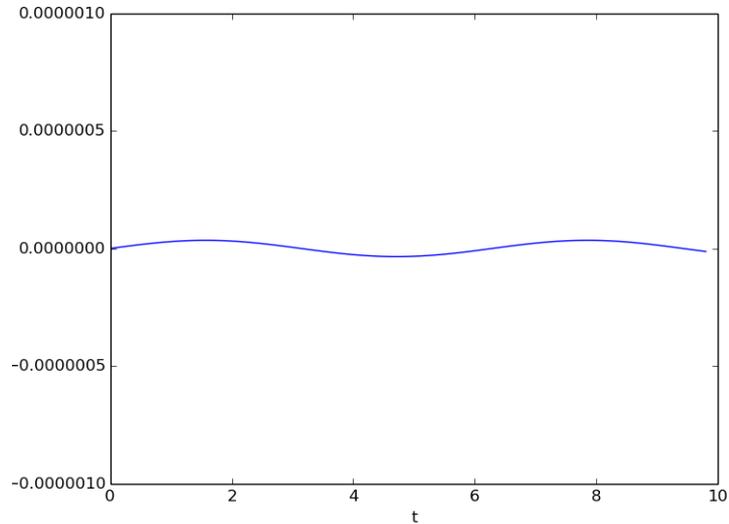


Figura 13: Erro na solução usando RK4 ( $\sin(t)_{RK} - \sin(t)$ )

Na escala da função, o erro na aproximação pelo método é tao pequeno que não pode

ser visto, aproximando para uma escala de intervalo de  $-10^{-5}$  a  $10^{-5}$  que o erro começa a ser observado, como pode ser visto na Figura 14.



### 3 Resultados para EDOs de circuito RC-RL

#### 3.1 Circuito RC

Seja um circuito RC1 em série com uma fonte de 12V, um resistor de  $5 \Omega$ , e um capacitor de 1F.

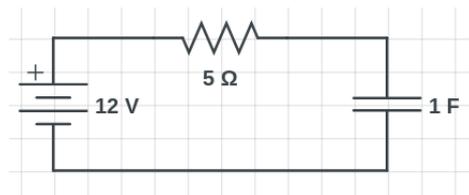


Figura 15: Circuito RC1.

Pela lei de Kirchhoff temos:

$$\varepsilon = \frac{dq}{dt}R + \frac{q}{C}$$

então:

$$\frac{dq}{dt} = \frac{\varepsilon - \frac{q}{C}}{R}$$

substituindo  $q$  por  $x$  tem-se:

$$\frac{dx}{dt} = \frac{\varepsilon - \frac{x}{C}}{R}$$

Aplicando o método de RK4, temos a seguinte equação  $x(t)$  como solução da equação diferencial:

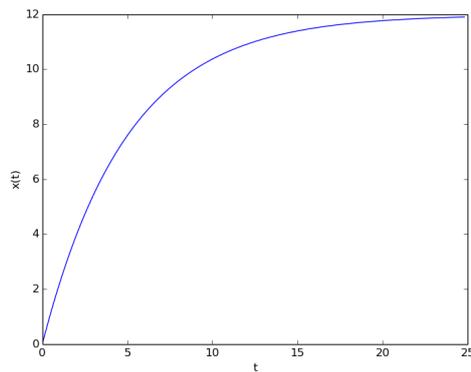


Figura 16:  $q(t)$  para o circuito RC1

O crescimento da carga no capacitor tem uma componente exponencial, de modo que, rigorosamente, ela só atingirá seu valor final,  $\varepsilon C = 12 \text{ C}$ , num tempo infinito, da mesma forma esperada pelo solução algébrica da equação diferencial.

Seja um segundo circuito RC2, dessa vez sem fonte, mas com o capacitor carregado de capacitância  $C= 1\text{F}$  com  $q_0= 10\text{C}$ , e uma resistência de  $R=5\Omega$ .

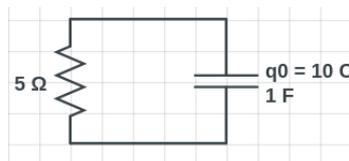


Figura 17: Circuito RC2

Pela lei de Kirchhoff temos:

$$\frac{q}{C} + \frac{dq}{dt} R$$

então:

$$\frac{dq}{dt} = \frac{-q}{RC}$$

substituindo  $q$  por  $x$  tem-se:

$$\frac{dx}{dt} = \frac{-x}{RC}$$

Aplicando o método de RK4, temos a seguinte equação  $x(t)$  como solução da equação diferencial:

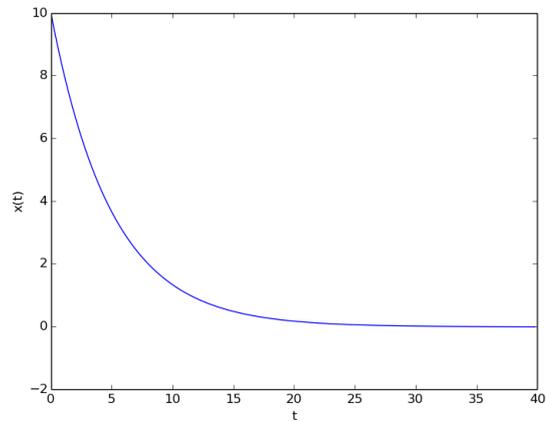


Figura 18:  $q(t)$  para o circuito RC2

A carga se inicia em  $q_0 = 10C$  e decai exponencialmente e tende a 0 quando  $t$  tende a infinito, da mesma forma que a solução algébrica da equação diferencial.

### 3.2 Circuito RL

Seja um circuito RL com fonte de  $\varepsilon = 12V$ , com um resistor de resistência  $R = 2\Omega$  e indutor de indutância  $L = 2H$ .

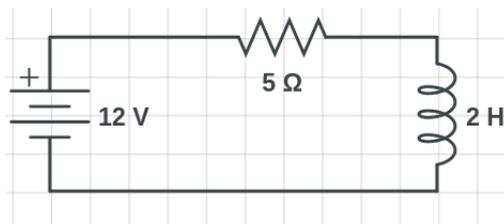


Figura 19: Circuito RL

Pela lei de Kirchhoff temos:

$$\varepsilon = Ri + \frac{di}{dt}L$$

então:

$$\frac{di}{dt} = \frac{\varepsilon - Ri}{L}$$

substituindo  $i$  por  $x$  tem-se:

$$\frac{dx}{dt} = \frac{\varepsilon - Rx}{L}$$

Aplicando o método de RK4, temos a seguinte equação  $x(t)$  como solução da equação diferencial:

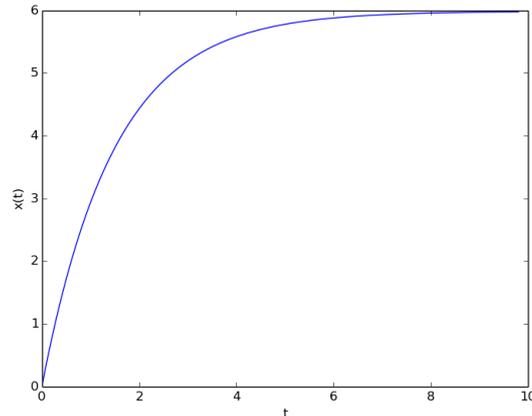


Figura 20: A corrente em função do tempo ( $i(t)$ )

O crescimento da corrente tem uma componente exponencial, e ela só atingirá seu valor final,  $\varepsilon/R = 6A$ , num tempo infinito, da mesma forma esperada pelo solução algébrica da equação diferencial.

## 4 Resultados para EDOs de circuito RLC

Seja um circuito RLC sem fonte, com um capacitor de capacitância  $C = 20$  carregado com carga  $q_0 = 10C$ , um resistor de resistência  $R = 10\Omega$  e um indutor de indutância  $L = 0.3H$ . Pela lei de Kirchhoff temos:

$$L \frac{d^2q}{dt^2} + R \frac{dq}{dt} + \frac{q}{C} = 0$$

isolando a derivada de maior ordem, temos a EDO:

$$\frac{d^2q}{dt^2} = -\frac{R}{L} \frac{dq}{dt} - \frac{q}{LC}$$

Nesse caso do circuito RLC a EDO é uma EDO de 2ª ordem, dessas formas o método deve ser aplicada duas vezes, a partir de uma substituição de variáveis da forma:

$$\frac{dq}{dt} = z$$

e a EDO original se torna:

$$\frac{dz}{dt} = -\frac{R}{L}z - \frac{q}{LC}$$

Dessa forma, aplicaremos o método de Runge-Kutta para as duas equações, que por sua vez uma dependente da outra. Os detalhes da aplicação do método nas duas EDOs pode ser melhor visualizado no script em anexo.

Substituindo  $q$  por  $x$  temos o conjunto de EDO's a ser resolvido:

$$\frac{dx}{dt} = z$$

$$\frac{dz}{dt} = -\frac{R}{L}z - \frac{x}{LC}$$

O Resultado de  $x(t)$  é mostrado na Figura 19.

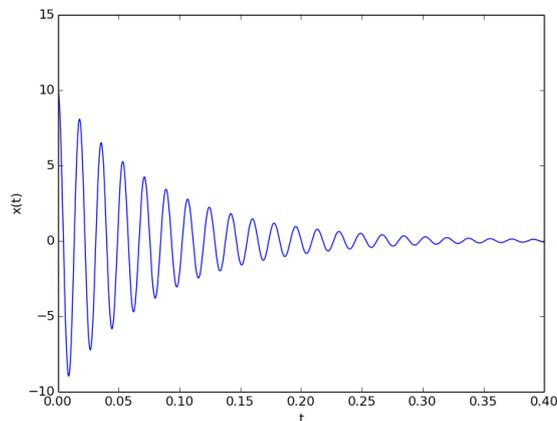


Figura 21:  $q(t)$  para o circuito RLC

O resultado condiz com o esperado, uma parcela oscilatória do tipo senóide, que inicia no valor inicial da quantidade de carga no capacitor, com sua amplitude decaindo exponencialmente, devido à dissipação por efeito Joule no resistor.

## 5 Bibliografia

ASANO, Claudio Hirofume. Cálculo Numérico — Fundamentos e Aplicações. 2009. Departamento de Matemática Aplicada – IME-USP. Disponível em: <https://www.ime.usp.br/asano/LivroNumerico.pdf>.

GALVÃO, Lauro Cesar; NUNES, Luiz Fernando. NOTAS DE AULA Cálculo Numérico. 2010. UTFPR. Disponível em: [http://paginapessoal.utfpr.edu.br/laurogalvao/disciplinas/calculo-numerico/materiais-didaticos/apostila/calculo\\_numerico.pdf](http://paginapessoal.utfpr.edu.br/laurogalvao/disciplinas/calculo-numerico/materiais-didaticos/apostila/calculo_numerico.pdf).

SHIRTS, Michel. 4th-Order Runge Kutta Method for ODEs. 2015. University of Colorado Boulder, Department of Chemical Biological Engineering.. Disponível em: <https://www.youtube.com/watch?v=1YZnic1Ug9g>.

WIKIPEDIA. Euler method. 2016. Disponível em: [https://en.wikipedia.org/wiki/Euler\\_method](https://en.wikipedia.org/wiki/Euler_method) >