PEA5009 **Aula 2**

Modelagem e simulação numérica de componentes de sistemas elétricos de potência

Introdução aos métodos de simulação numérica
Método Dommel

Prof. Dr. Eduardo Coelho Marques da Costa
Prof. Dr. Renato Machado Monaro

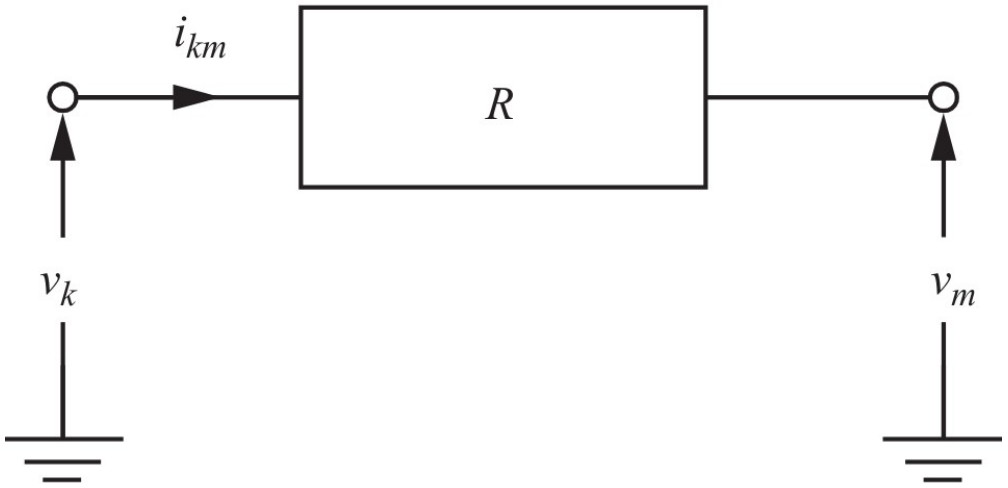
Simulação por Integração Numérica

- **Transformar equações diferenciais em lineares**
- **Proposto por Dommel**
  - Discretização das equações dos elementos
  - Integração trapezoidal
  - Solução circuito Nodal
  - Erros de truncamento (Taylor)
- **Usado no:**
  - ATP, PSCAD, EMTP, RTDS Etc

Fonte: [1]

Simulação por Integração Numérica

- **Resistor**

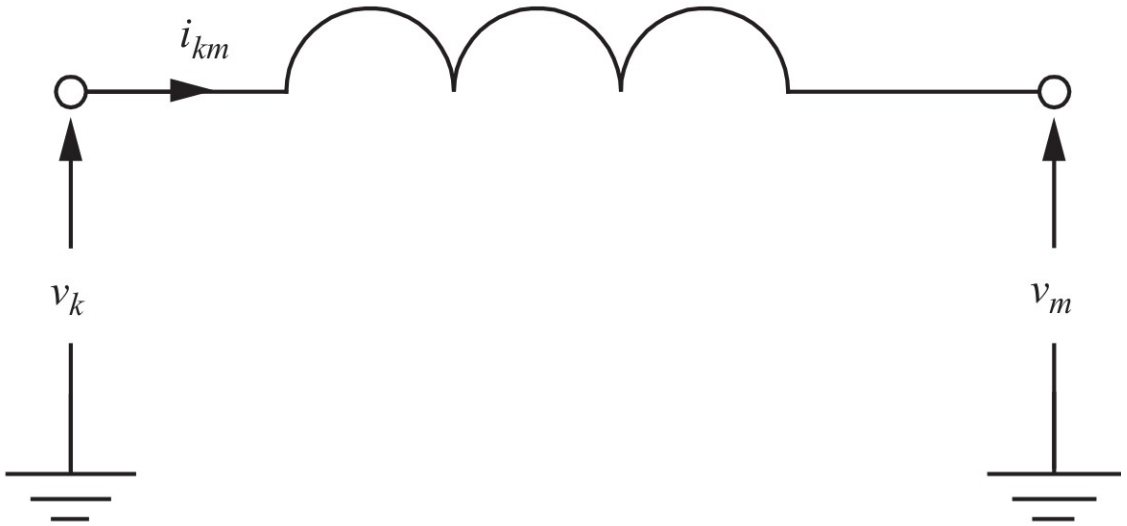


$$i_{km}(t) = \frac{1}{R}(v_k(t) - v_m(t))$$

Fonte: [1]

Simulação por Integração Numérica

- **Indutor**



$$v_L = v_k - v_m = L\frac{di_{km}}{dt}$$

Fonte: [1]

Simulação por Integração Numérica

- **Indutor**

$$v_L = v_k - v_m = L\frac{di_{km}}{dt}$$

$$i_{km(t)} = i_{km(t-\Delta t)} + \int_{t-\Delta t}^{t} \underset{1/L}{} (v_k - v_m)\,dt$$

$$i_{km(t)} = i_{km(t-\Delta t)} + \frac{\Delta t}{2L}((v_k - v_m)_{(t)} + (v_k - v_m)_{(t-\Delta t)})$$

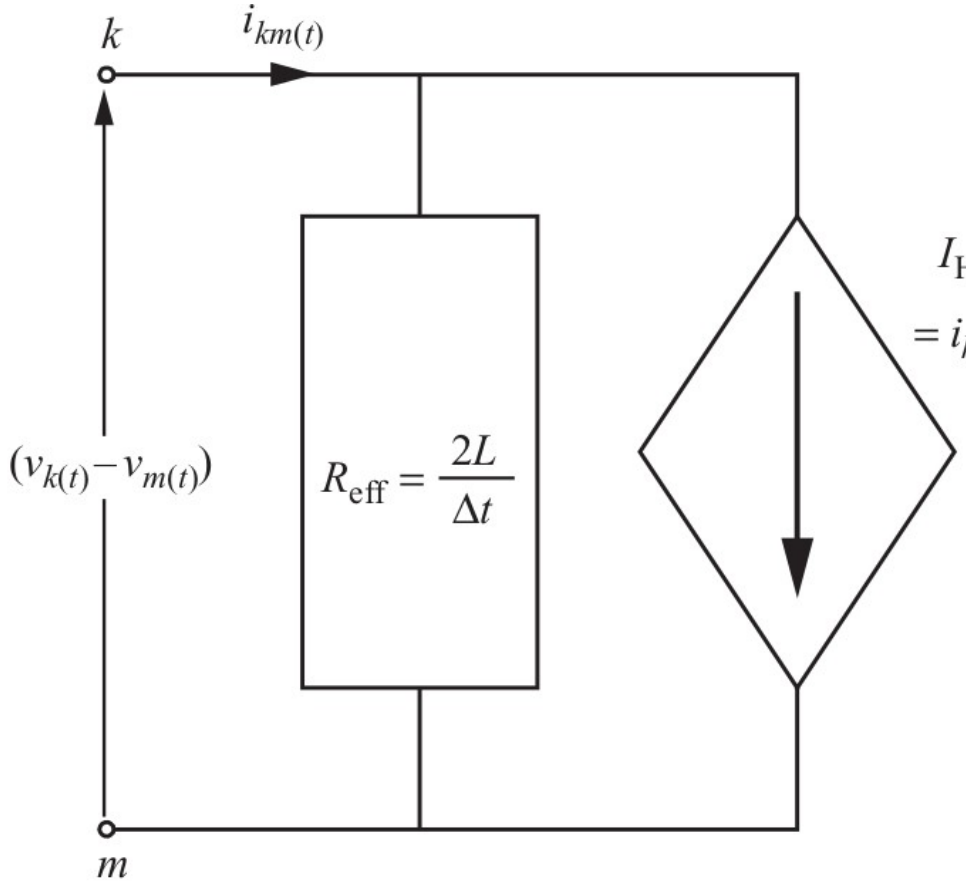
$$= i_{km(t-\Delta t)} + \frac{\Delta t}{2L}(v_{k(t-\Delta t)} - v_{m(t-\Delta t)}) + \frac{\Delta t}{2L}(v_{k(t)} - v_{m(t)})$$

$$i_{km}(t) = I_{\text{History}}(t - \Delta t) + \frac{1}{R_{\text{eff}}}(v_k(t) - v_m(t))$$

Fonte: [1]

Simulação por Integração Numérica

- **Indutor**

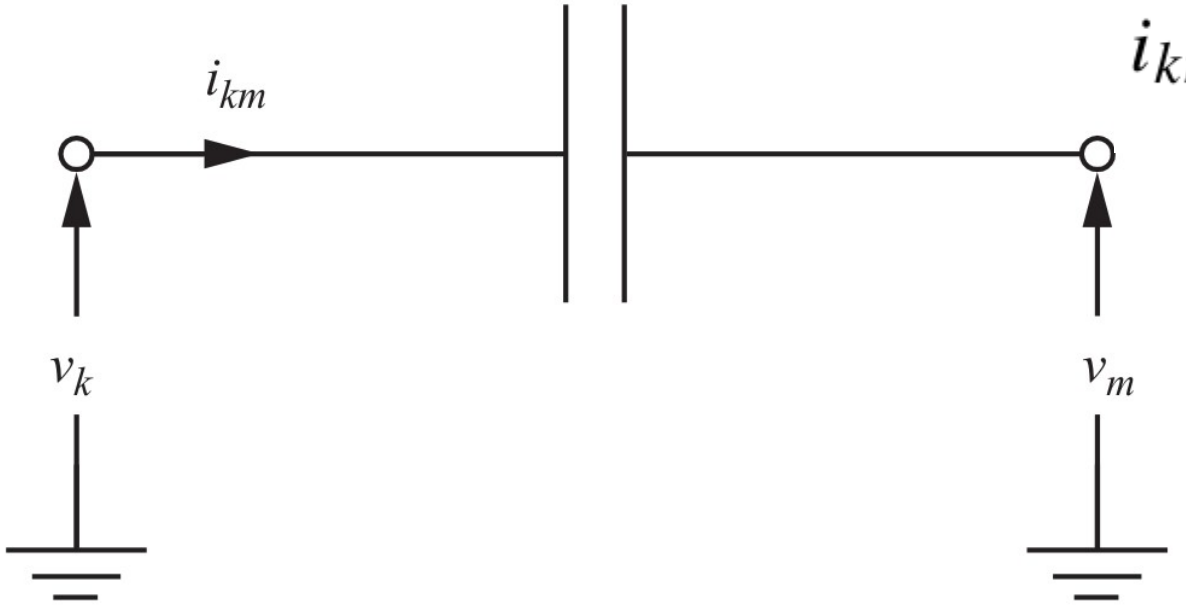


$$I_{\text{History}}(t-\Delta t) = i_{km}(t-\Delta t) + \frac{\Delta t}{2L}\left(v_k(t-\Delta t) - v_m(t-\Delta t)\right)$$

Fonte: [1]

Simulação por Integração Numérica

- **Capacitor**



$$i_{km}(t) = C\frac{d(v_k(t) - v_m(t))}{dt}$$

Fonte: [1]

Simulação por Integração Numérica

- **Capacitor**

$$i_{km}(t) = C\frac{d(v_k(t) - v_m(t))}{dt}$$

$$v_{km(t)} = (v_{k(t)} - v_{m(t)}) = (v_{k(t-\Delta t)} - v_{m(t-\Delta t)}) + \frac{1}{C}\int_{t-\Delta t}^{t} i_{km}\, dt$$

$$v_{km}(t) = (v_k(t) - v_m(t)) = (v_k(t-\Delta t) - v_m(t-\Delta t)) + \frac{\Delta t}{2C}(i_{km}(t) + i_{km}(t-\Delta t))$$

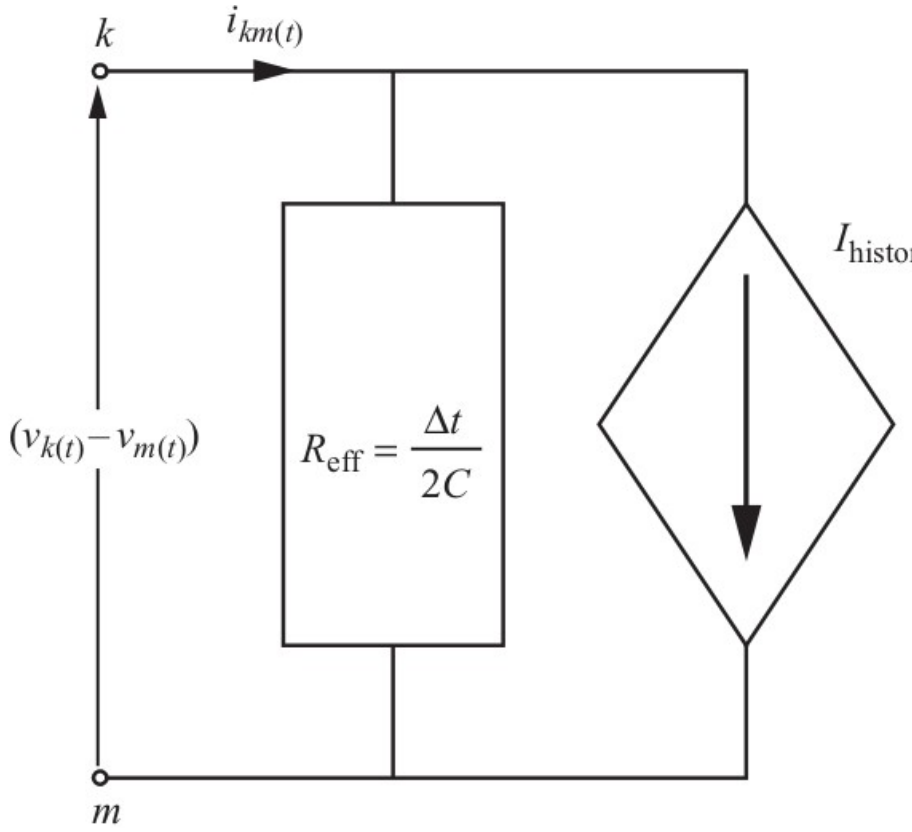
$$i_{km}(t) = \frac{2C}{\Delta t}(v_k(t) - v_m(t)) - i_{km}(t-\Delta t) - \frac{2C}{\Delta t}(v_k(t-\Delta t) - v_m(t-\Delta t))$$

$$= \frac{1}{R_{\text{eff}}}[v_k(t) - v_m(t)] + I_{\text{History}}(t-\Delta t)$$

Fonte: [1]

Simulação por Integração Numérica

- **Capacitor**



$$I_{\text{history}}(t-\Delta t) = -i_{km}(t-\Delta t) - \frac{2C}{\Delta t}\left(v_k(t-\Delta t) - v_m(t-\Delta t)\right)$$

Fonte: [1]

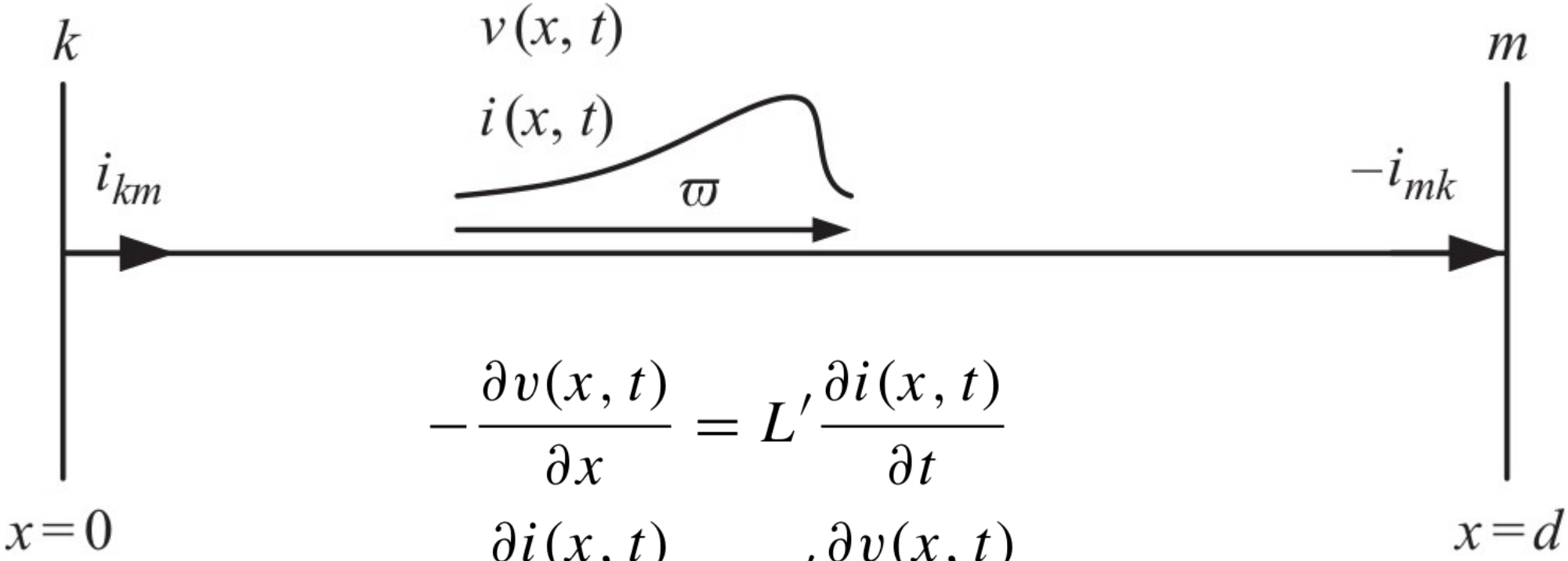
Simulação por Integração Numérica

Integration method	R_{eq}	$I_{History}$
Inductor		
Backward Euler	$\frac{L}{\Delta t}$	i_{n-1}
Trapezoidal	$\frac{2L}{\Delta t}$	$i_{n-1} + \frac{\Delta t}{2L} v_{n-1}$
Gear 2nd order	$\frac{3L}{2\Delta t}$	$\frac{4}{3} i_{n-1} - \frac{1}{3} i_{n-2}$
Capacitor		
Backward Euler	$\frac{\Delta t}{C}$	$-\frac{C}{\Delta t} v_{n-1}$
Trapezoidal	$\frac{\Delta t}{2C}$	$-\frac{2C}{\Delta t} v_{n-1} - i_{n-1}$
Gear 2nd order	$\frac{2\Delta t}{3C}$	$-\frac{2C}{\Delta t} v_{n-1} - \frac{C}{2\Delta t} v_{n-2}$

Fonte: [1]

Simulação por Integração Numérica

- **Linha de Transmissão**



$$-\frac{\partial v(x,t)}{\partial x} = L' \frac{\partial i(x,t)}{\partial t}$$

$$-\frac{\partial i(x,t)}{\partial x} = C' \frac{\partial v(x,t)}{\partial t}$$

Fonte: [1]

Simulação por Integração Numérica

$$-\frac{\partial v(x,t)}{\partial x} = L'\frac{\partial i(x,t)}{\partial t}$$

$$-\frac{\partial i(x,t)}{\partial x} = C'\frac{\partial v(x,t)}{\partial t}$$

$$i(x,t) = f_1(x - \varpi t) + f_2(x + \varpi t)$$

$$v(x,t) = Z \cdot f_1(x - \varpi t) - Z \cdot f_2(x + \varpi t)$$

$$\varpi = \frac{1}{\sqrt{L'C'}}$$

$$Z_C = \sqrt{\frac{L'}{C'}}$$

Fonte: [1]

Simulação por Integração Numérica

$$i(x,t) = f_1(x - \varpi t) + f_2(x + \varpi t)$$

$$v(x,t) = Z \cdot f_1(x - \varpi t) - Z \cdot f_2(x + \varpi t)$$

$$v(x,t) - Z_C \cdot i(x,t) = -2Z_C \cdot f_2(x + \varpi t)$$

$$i(x,t) = f_1(x - \varpi t) + f_2(x + \varpi t)$$

$$v(x,t) = Z \cdot f_1(x - \varpi t) - Z \cdot f_2(x + \varpi t)$$

$$v(x,t) + Z_C \cdot i(x,t) = 2Z_C \cdot f_1(x - \varpi t)$$

Fonte: [1]

Simulação por Integração Numérica

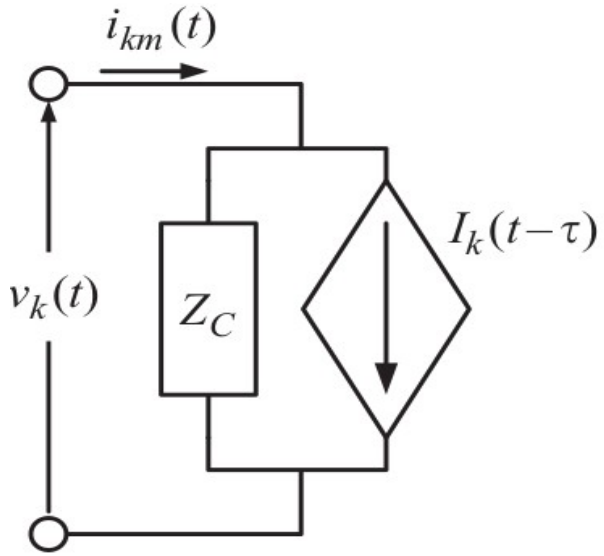
$$v(x,t) - Z_C \cdot i(x,t) = -2Z_C \cdot f_2(x + \varpi t)$$
$$v(x,t) + Z_C \cdot i(x,t) = 2Z_C \cdot f_1(x - \varpi t)$$
$$\tau = d/\varpi = d\sqrt{L'C'}$$

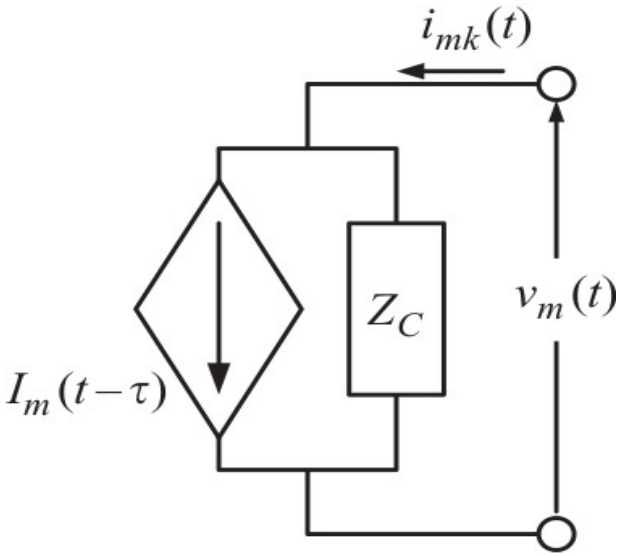
$$v_k(t-\tau) + Z_C \cdot i_{km}(t-\tau) = v_m(t) + Z_C \cdot (-i_{mk}(t))$$

$$i_{mk}(t) = \frac{1}{Z_C} v_m(t) + I_m(t-\tau)$$
$$I_m(t-\tau) = -\frac{1}{Z_C} v_k(t-\tau) - i_{km}(t-\tau)$$

Fonte: [1]

Simulação por Integração Numérica





$$i_{km}(t) = \frac{1}{Z_C} v_k(t) + I_k(t-\tau)$$

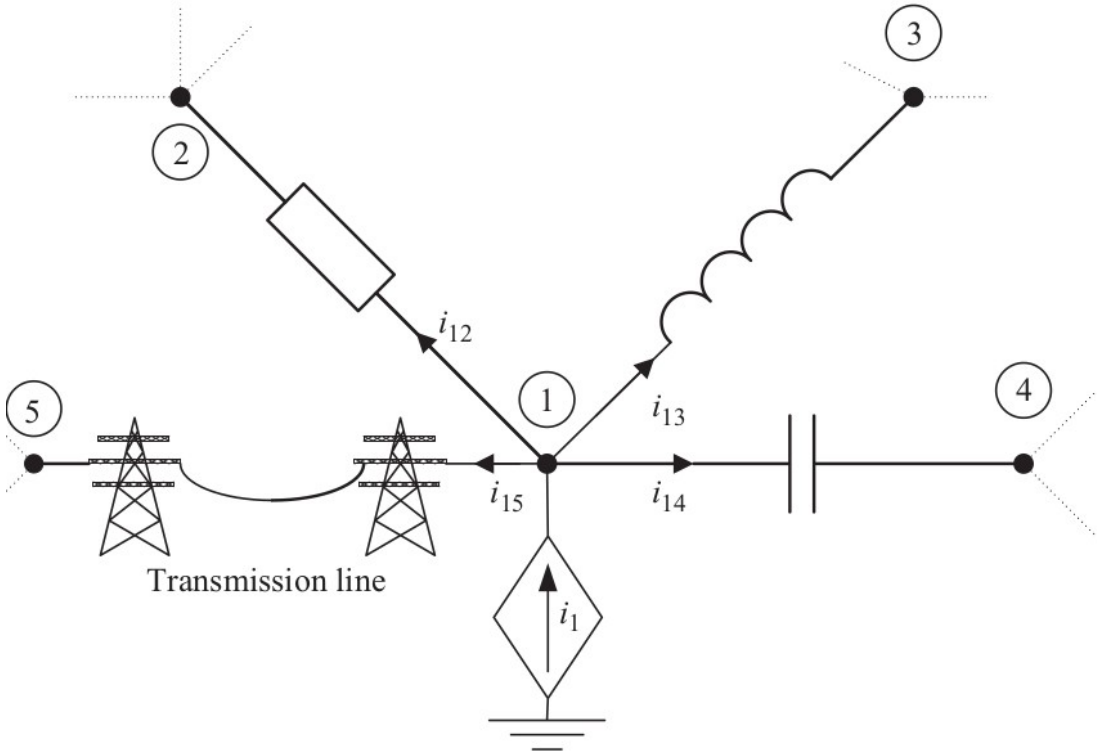
$$i_{mk}(t) = \frac{1}{Z_C} v_m(t) + I_m(t-\tau)$$

$$I_k(t-\tau) = -\frac{1}{Z_C} v_m(t-\tau) - i_{mk}(t-\tau)$$

$$I_m(t-\tau) = -\frac{1}{Z_C} v_k(t-\tau) - i_{km}(t-\tau)$$

Fonte: [1]

Simulação por Integração Numérica



$$i_{12}(t) + i_{13}(t) + i_{14}(t) + i_{15}(t) = i_1(t)$$

$$i_{12}(t) = \frac{1}{R}(v_1(t) - v_2(t))$$

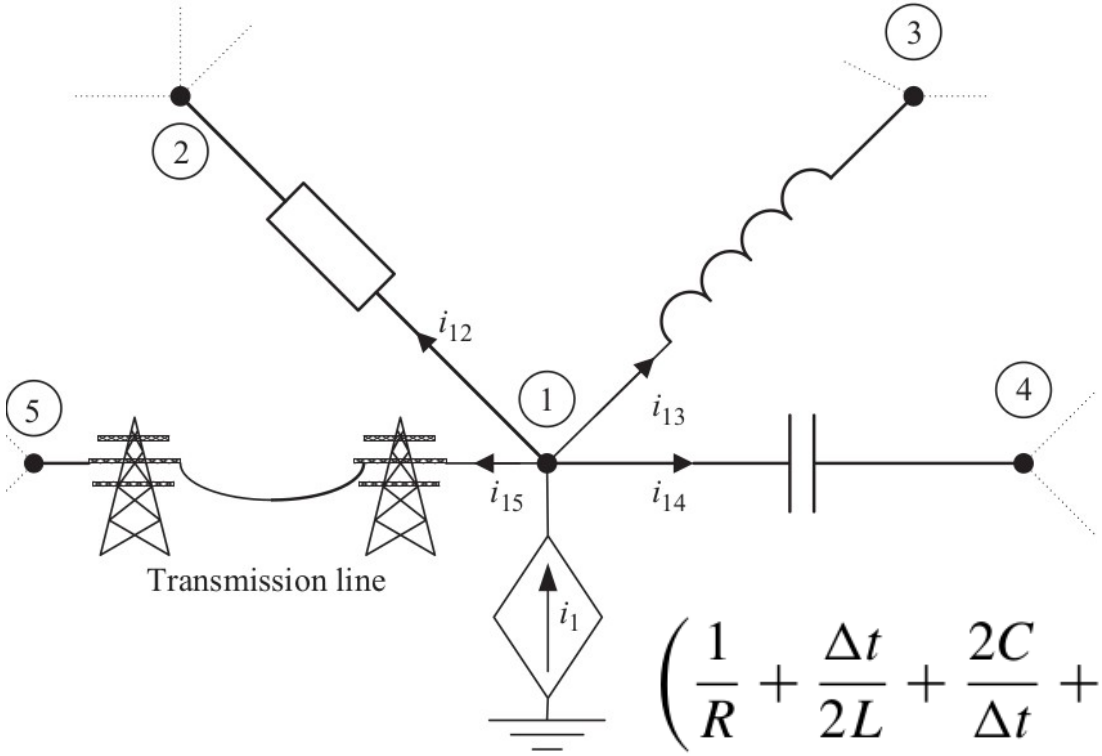
$$i_{13}(t) = \frac{\Delta t}{2L}(v_1(t) - v_3(t)) + I_{13}(t - \Delta t)$$

$$i_{14}(t) = \frac{2C}{\Delta t}(v_1(t) - v_4(t)) + I_{14}(t - \Delta t)$$

$$i_{15}(t) = \frac{1}{Z}v_1(t) + I_{15}(t - \tau)$$

Fonte: [1]

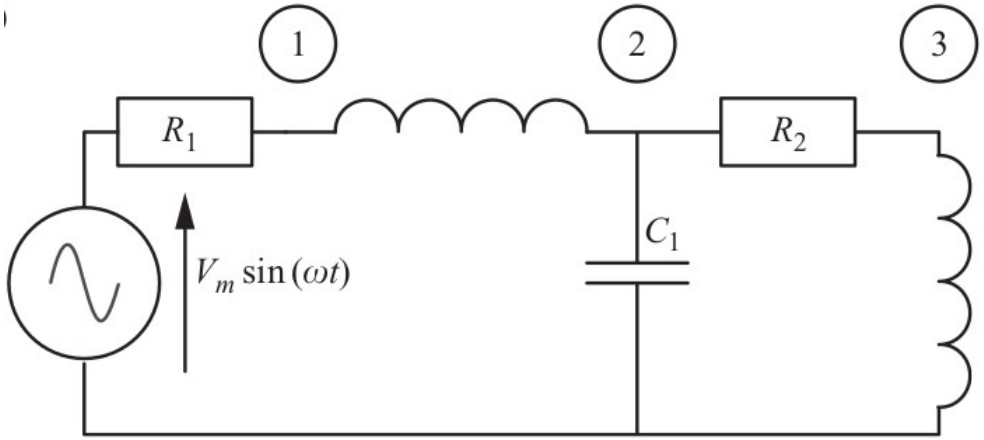
Simulação por Integração Numérica

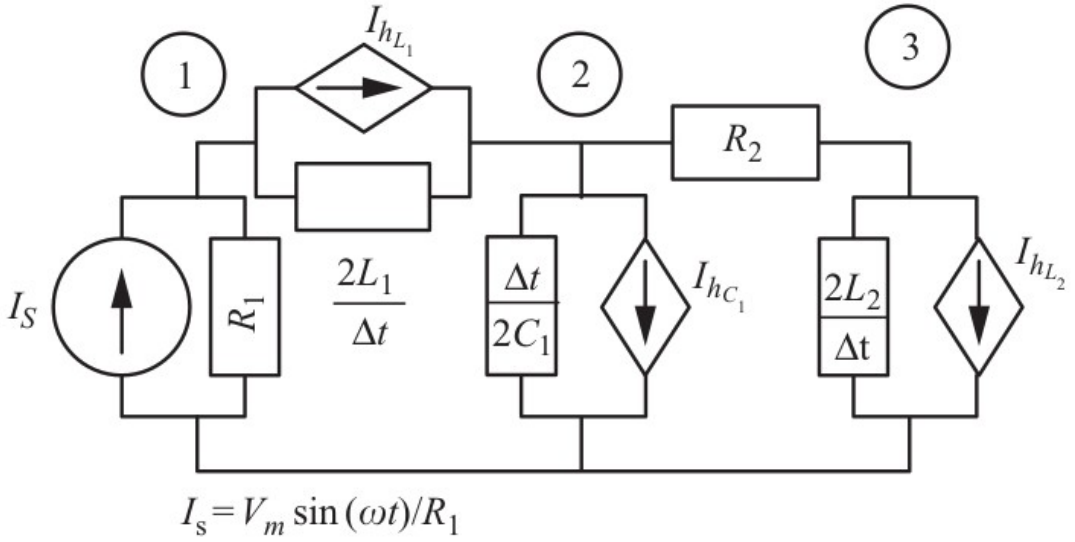


$$\left(\frac{1}{R}+\frac{\Delta t}{2L}+\frac{2C}{\Delta t}+\frac{1}{Z}\right)v_1(t)-\frac{1}{R}v_2(t)-\frac{\Delta t}{2L}v_3(t)-\frac{2C}{\Delta t}v_4(t)$$
$$= I_1(t-\Delta t)-I_{13}(t-\Delta t)-I_{14}(t-\Delta t)-I_{15}(t-\Delta t)$$

Fonte: [1]

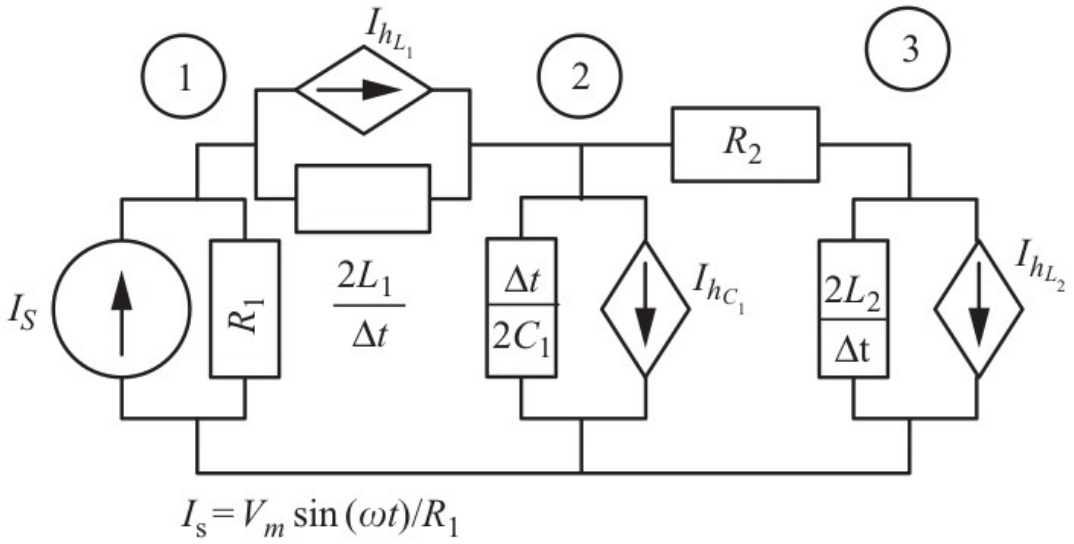
Simulação por Integração Numérica





Fonte: [1]

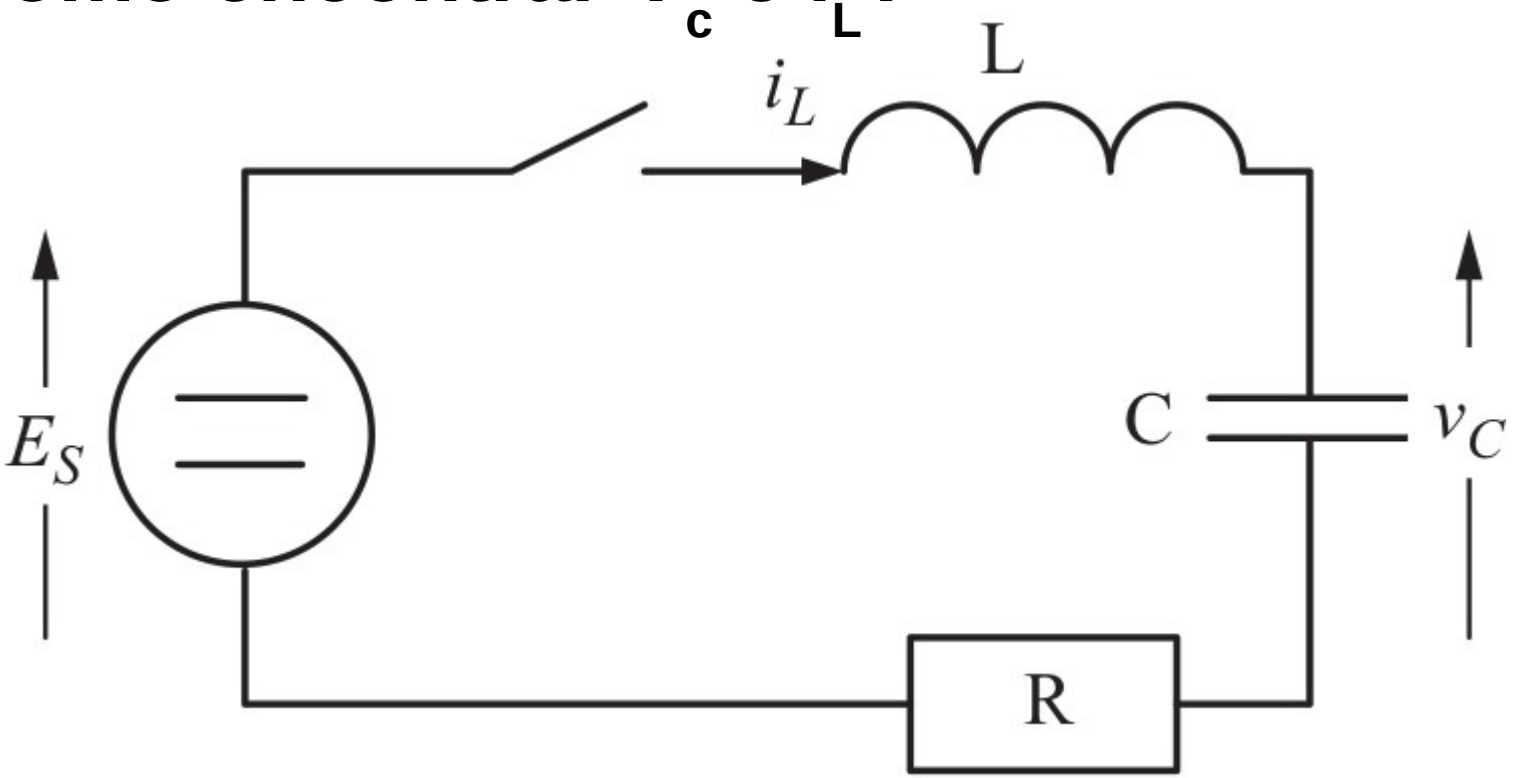
Simulação por Integração Numérica



$$[G]\mathbf{v}(t) = \mathbf{i}(t) + \mathbf{I}_{\text{History}}$$

$$\begin{bmatrix} \frac{1}{R_1} + \frac{\Delta t}{2L_1} & -\frac{\Delta t}{2L_1} & 0 \\ -\frac{\Delta t}{2L_1} & \frac{\Delta t}{2L_1} + \frac{1}{R_2} + \frac{2C_1}{\Delta t} & -\frac{1}{R_2} \\ 0 & -\frac{1}{R_2} & \frac{1}{R_2} + \frac{\Delta t}{2L_2} \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \frac{V_m \sin(\omega t)}{R_1} - I_{h_{L_1}} \\ I_{h_{L_1}} - I_{h_{C_1}} \\ -I_{h_{L_2}} \end{pmatrix}$$

Como encontrar V_c e i_L?



R=20Ω L=6,95mH C=1μF Es=1 V

Chave fecha em 0,1 ms

Fonte: [1]

Simulação por Integração Numérica

Simulação por Integração Numérica

Como Resolver?

Referências

[1] WATSON, Neville; ARRILLAGA, Jos. Power systems electromagnetic transients simulation. Iet, 2003.

[2] http://www.ece.ncsu.edu/power/factstna.html

[3] Monaro e Di Santo https://github.com/renato-monaro/OEMTP

[4] http://www.mathworks.com/products/matlab/

[5] https://www.gnu.org/software/octave/

http://www.mathworks.com/products/matlab/

Ferramentas Computacionais

Código aberto

Tipos de licenças: GPL, LGPL, BSD;

Pode ser modificado, vendido mas o direito autoral **deve** ser mantido;

Acesso livre aos códigos fontes;

Ferramentas Computacionais

Gnuplot

- Programa (modo terminal) para confecção de gráficos 2/3D
- Visualização em tela ou arquivo (pdf, eps, svg, png)
- Instalação: sudo apt-get install gnuplot-x11
- Uso: gnuplot

GCC

- Compilador multilinguagem padrão do linux
- Engloba a etapa de compilação e *linkagem* dos programas
- Instalação sudo apt-get install gcc
- Uso: gcc arquivo.c -o nomeprograma

Ferramentas Computacionais

GSL - GNU Scientific Library

Biblioteca que reúne um conjunto de rotinas computacionais

Foco em algoritmos para uso científico de código aberto

Complex Number, Roots of Polynomials Special Functions, **Vectors and Matrices,** Permutations, Sorting, **BLAS Support**, **Linear Algebra**, Eigensystems, Fast Fourier Transforms,Quadrature, Random Numbers, Quasi-Random Sequences,Random Distributions, Statistics, Histograms, N-Tuples, Monte Carlo Integration, Simulated Annealing, Differential Equations, Interpolation, Numerical Differentiation, Chebyshev Approximation, Series Acceleration, Discrete Hankel Transforms, Root-Finding, Minimization Least-Squares Fitting, Physical Constants, IEEE Floating-Point, Discrete Wavelet Transforms, Basis splines, Running Statistics, **Sparse Matrices and Linear Algebra**

Instalação sudo apt-get install libgsl0-**dev**

Uso: gcc arquivo.c -o nomeprograma -lm -lgsl -lgslcblas

Exemplo Espaço de Estados

```
#include <stdio.h>
#include <math.h>
#include <gsl/gsl_errno.h>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_odeiv2.h>

struct ODE_Data{
double R;
double L;
double C;
double Vp;
double Freq;
};

int func (double t, const double y[], double f[],void *parm){
 struct ODE_Data *Param;
 Param=parm;
 double E;
 E=Param->Vp*cos(2*M_PI*Param->Freq*t);
f[0]=-(Param->R/Param->L)*y[0]-(1.0/Param->L)*y[1]+(1.0/Param-
>L)*E;
f[1]=(1.0/Param->C)*y[0];
 return GSL_SUCCESS;
}
```

```
int main(void){
        struct ODE_Data Parameters;
        Parameters.R=1E0;
        Parameters.L=1.0E-3;
        Parameters.C=1.0E-5;
        Parameters.Vp=100.0;
        Parameters.Freq=60.0;

gsl_odeiv2_system sys = {func, NULL, 2, &Parameters};
gsl_odeiv2_driver*d=gsl_odeiv2_driver_alloc_y_new(&sys,gsl_odeiv2_step_r
kf45, 1e-8, 1e-6, 0.0);
  int i,status;
  double t = 0.0, dt=1E-4, ti=0;
  double y[2] = { 0.0, 0.0 };
  for (i=0; i<300; i++){
      ti+=dt;
      status=gsl_odeiv2_driver_apply (d, &t, ti, y);
      if (status != GSL_SUCCESS){
          printf ("error, return value=%d\n", status);
           break;
          }
     printf ("%.5e %.5e %.5e\n", t, y[0], y[1]);
    }
  gsl_odeiv2_driver_free (d);
  return 0;
}
```

Exemplo Espaço de Estados

Editando

> mkdir workshop

> cd workshop

>gedit exemplot_ee.c**&**

Compilando

>ls

> gcc exemplo_ee.c -o ex_ee **-lm -lgsl -lgslcblas**

>ls

Executando

>**./**ex_ee

>**./**ex_ee **>** ex_ee.txt

> gedit ex_ee.txt&

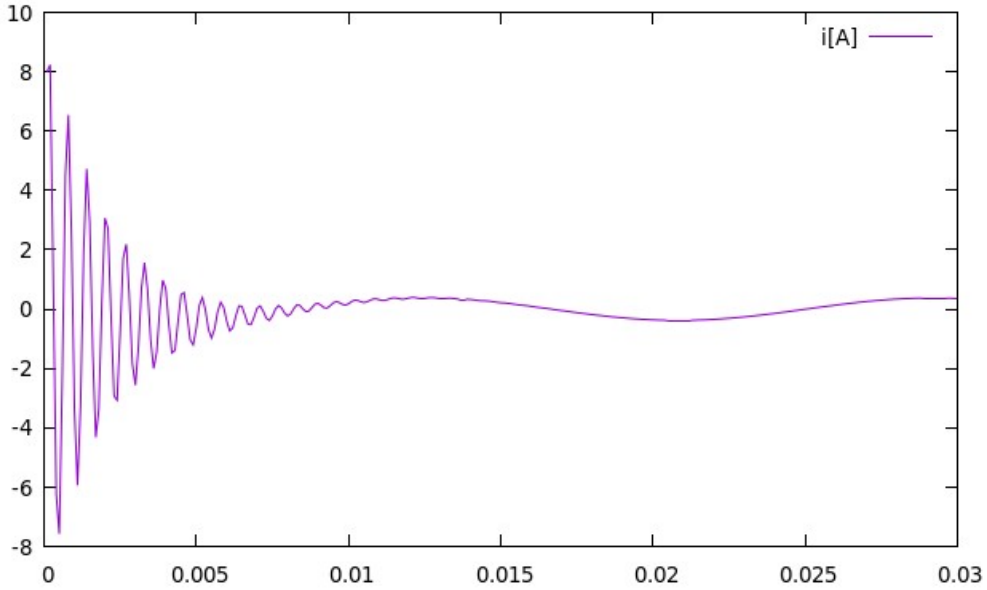
Exemplo Espaço de Estados

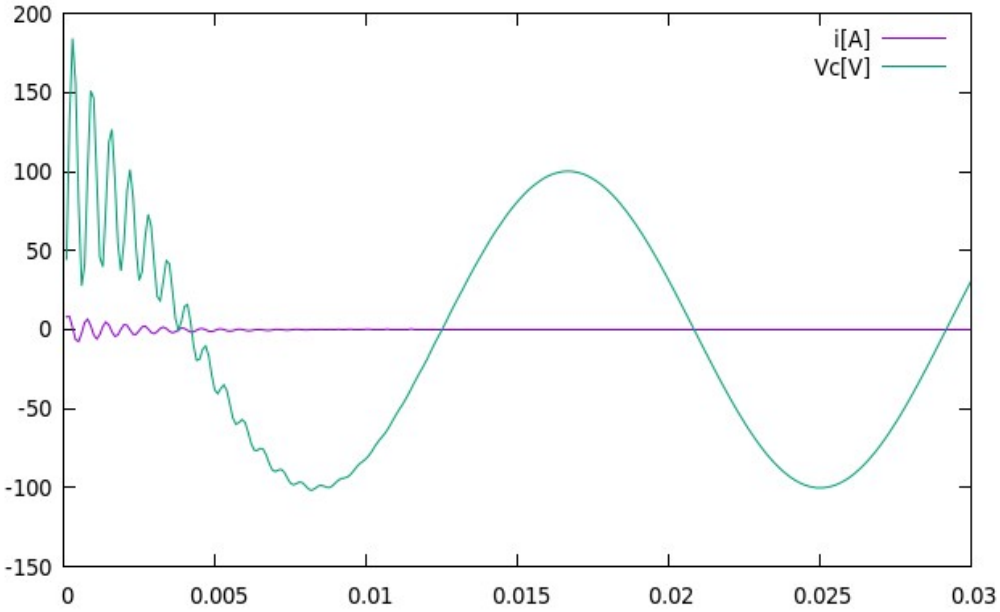
Visualizando no Gnuplot

> gnuplot

$ plot 'ex_ee.txt' using 1:2 with lines title 'i[A]'

$ replot '' using 1:3 with lines title 'Vc[V]'





Exemplo Substituição Numérica

```
#include <stdio.h>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_linalg.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>

struct ODE_Data{ double R; double L; double C;
 double Vp; double Freq; };

int main(void){
int i,k,j;
double t = 0.0, dt=1E-4;
struct ODE_Data Parameters;
Parameters.R=1E0; Parameters.L=1.0E-3; Parameters.C=1.0E-5;
Parameters.Vp=100.0; Parameters.Freq=60.0;
double Reff_L, Reff_C, R_F;
R_F=Parameters.R;
Reff_L=2*Parameters.L/dt;
Reff_C=dt/(2*Parameters.C);
double G_L, G_C, G_F;
G_L=1.0/(Reff_L);
G_C=1.0/(Reff_C);
G_F=1.0/(R_F);
gsl_matrix *G; //Ponteiro Matrix GSL
G=gsl_matrix_alloc(2, 2); //Aloca Memória 2x2
/*Montagem G*/
gsl_matrix_set(G,0,0,G_L+G_F);
gsl_matrix_set(G,0,1,-G_L);
gsl_matrix_set(G,1,0,-G_L);
gsl_matrix_set(G,1,1,G_L+G_C);
```

```
int s;
  gsl_permutation *p; //Vetor permutação
p=gsl_permutation_alloc(2); //Alocação vetor permutação
gsl_linalg_LU_decomp(G, p, &s); //Decoposição da matrix G em LU

gsl_vector *V; //Ponteiro Vetor Tensões
V=gsl_vector_alloc(2); //Aloca Memória 2

gsl_vector *I; //Ponteiro Vetor Correntes
I=gsl_vector_alloc(2); //Aloca Memória 2

double Ih_L, I_L=0, I_L_old, Ih_C, I_C_old, I_C=0, I_F, V1=0,V2=0;
for (i=0; i<300; i++){
 Ih_L=I_L+(dt/(2*Parameters.L))*(V1-V2);//Corrente Histórica Indutor
 Ih_C=-I_C-((2*Parameters.C)/dt)*(V2-0);//Corrente Histórica Capacitor
 I_F=Parameters.Vp*cos(2*M_PI*Parameters.Freq*t)/R_F; //Corrente da
Fonte
/*Popula vetor I*/
gsl_vector_set(I,0,I_F-Ih_L);
gsl_vector_set(I,1,Ih_L-Ih_C);
gsl_linalg_LU_solve(G,p,I,V);//Resolve o sistema Ax=B

V1=gsl_vector_get(V,0);
V2=gsl_vector_get(V,1);
/*Calcula a corrnte nos elementos*/
I_L=(V1-V2)/Reff_L+Ih_L;
I_C=(V2-0)/Reff_C+Ih_C;
printf ("%.5e %.5e %.5e\n", t, I_L, V2);
t+=dt;
}
  return 0;
}
```

Exemplo Substituição Numérica

Editando

>gedit exemplot_sn.c**&**

Compilando

>ls

> gcc exemplo_sn.c -o ex_sn **-lm -lgsl -lgslcblas**

>ls

Executando

>**./**ex_sn

>**./**ex_sn **>** ex_sn.txt

> gedit ex_sn.txt&

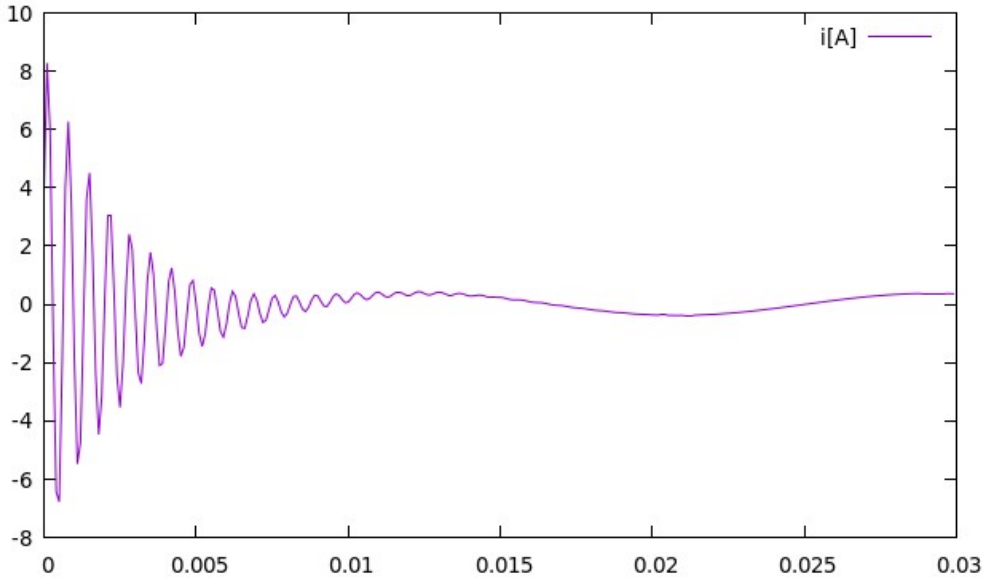
Exemplo Substituição Numérica

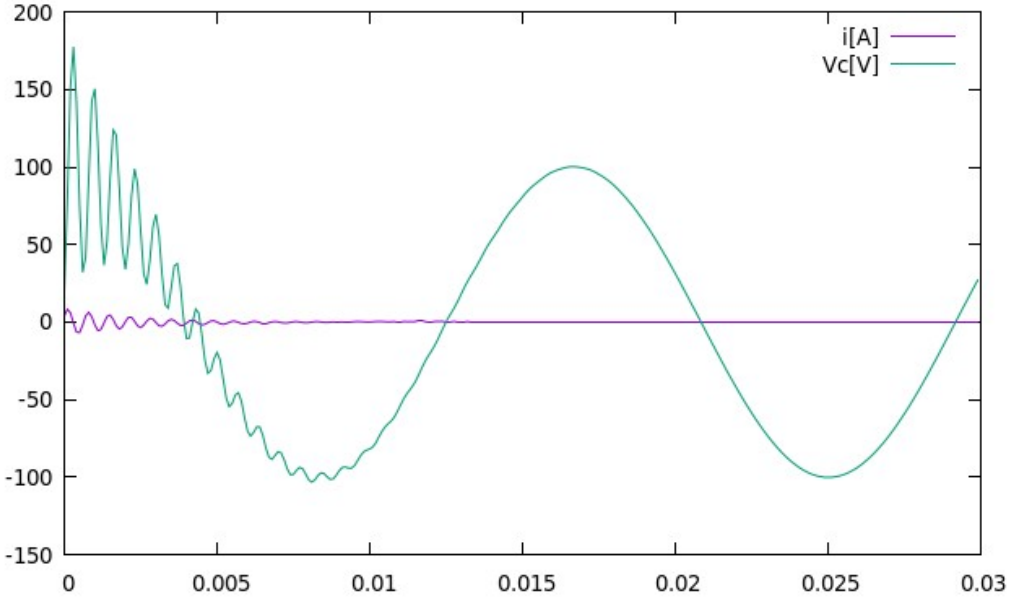
Visualizando no Gnuplot

> gnuplot

$ plot 'ex_sn.txt' using 1:2 with lines title 'i[A]'

$ replot '' using 1:3 with lines title 'Vc[V]'





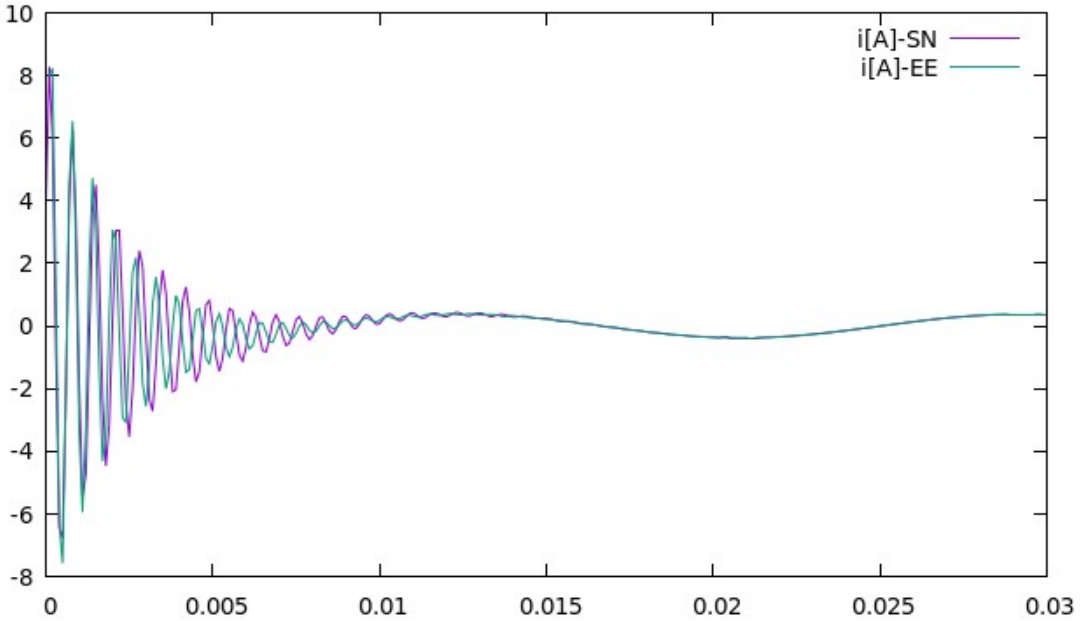
Comparação EE x SN

Visualizando no Gnuplot

> gnuplot

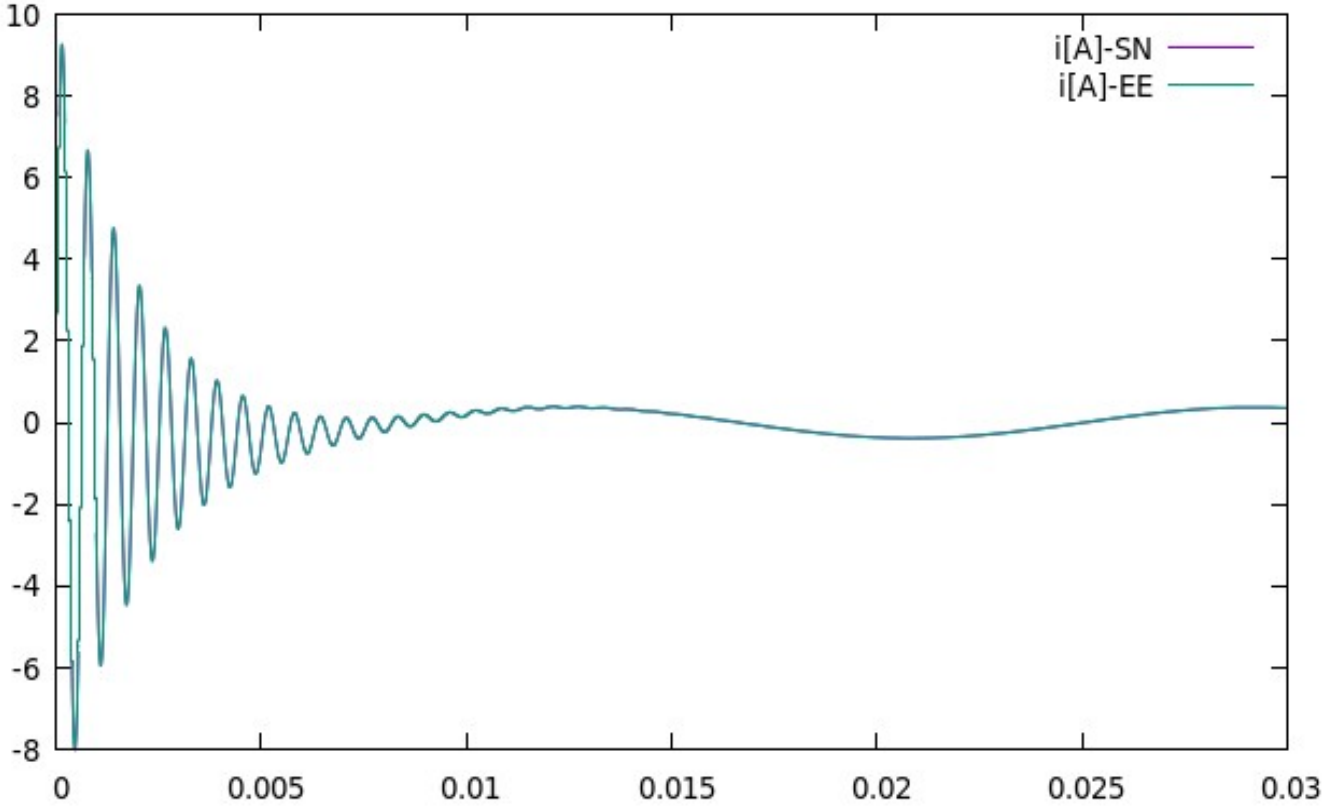
$ plot 'ex_sn.txt' using 1:2 with lines title 'i[A]-EE'

$ replot 'ex_ee.txt' using 1:3 with lines title 'i[A]-SN'

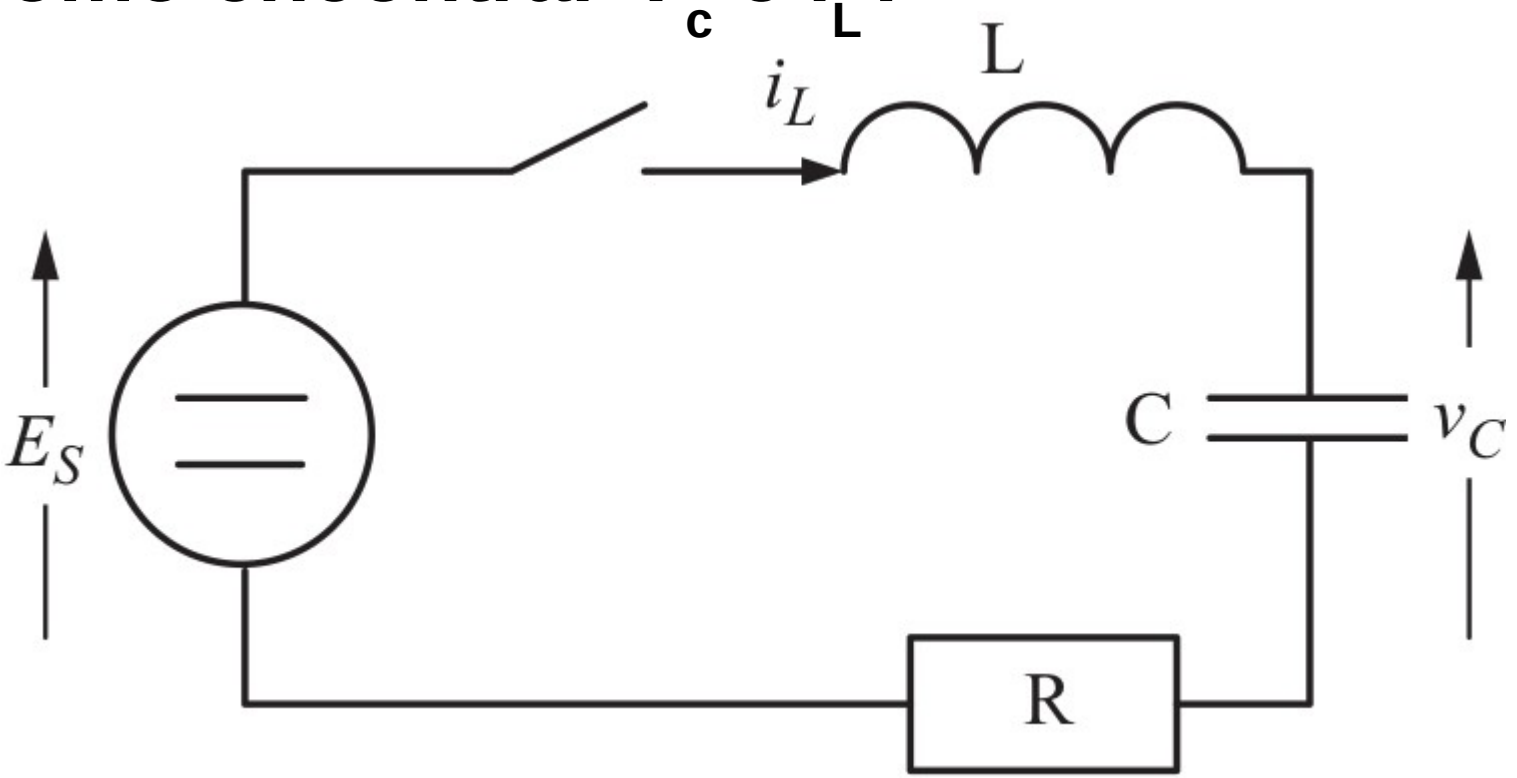


Comparação EE x SN

Alterando o passo para 1E-6



Como encontrar V_c e i_L?



R=20Ω L=6,95mH C=1μF Es=1 V

Chave fecha em 0,1 ms

Fonte: [1]

Espaço de Estado

Como Resolver o EE?

myode.m

```
##differential equation
function myode = f (x, t)
  xdot = zeros (3,1);
  xdot(1) = 77.27 * (x(2) + x(1) );
  xdot(2) = (x(3) - x(1)*x(2) - x(2))
  xdot(3) = 0.161*(x(1) – x(3));
endfunction
```

```
>>  x0 = [ 4; 1.1; 4 ];
>> t = linspace (0, 500, 1000);
>> y = ode45 (@f, t,x0);
```

```
>>  x0 = [ 4; 1.1; 4 ];
>> t = linspace (0, 500, 1000);
>> y = lsode ("f", x0, t);
```