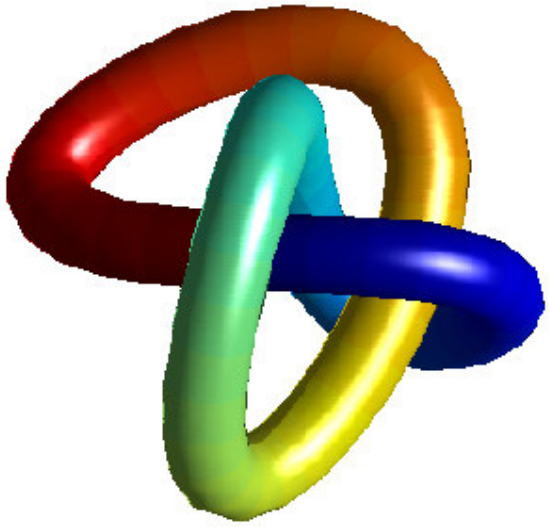
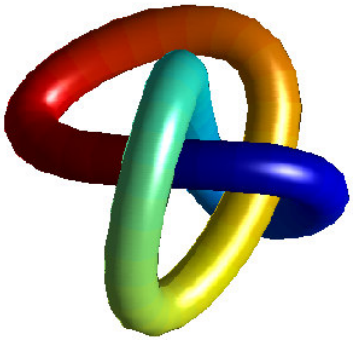
TRABALHO
DE
ANÁLISE NUMÉRICA



UNIVERSIDADE ESTADUAL PAULISTA – UNESP.
INSTITUTO DE GEOCIÊNCIAS E CIÊNCIAS EXATAS "CAMPUS" DE RIO CLARO.
ABRIL – 2000.

TRABALHO
DE
ANÁLISE NUMÉRICA



ERICH GEBRIN BACHION
FÁBIO CÔA
5.º SEMESTRE – CIÊNCIA DA COMPUTAÇÃO
PROFº.: JOSÉ MANOEL BALTHAZAR.

UNIVERSIDADE ESTADUAL PAULISTA – UNESP.
INSTITUTO DE GEOCIÊNCIAS E CIÊNCIAS EXATAS "CAMPUS" DE RIO CLARO.
ABRIL – 2000.

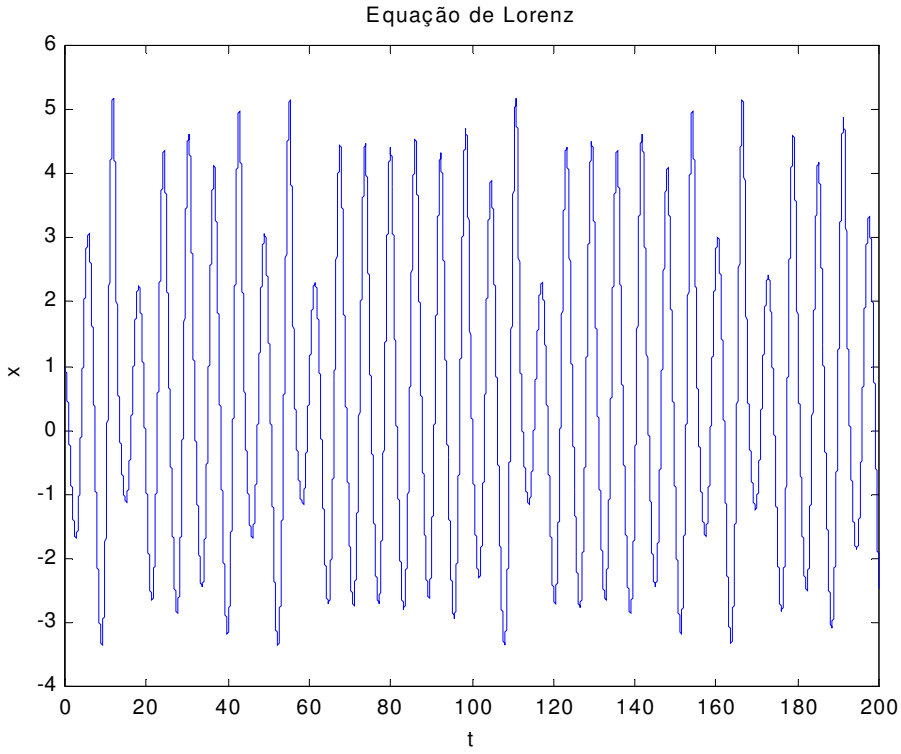
Equação de Lorenz

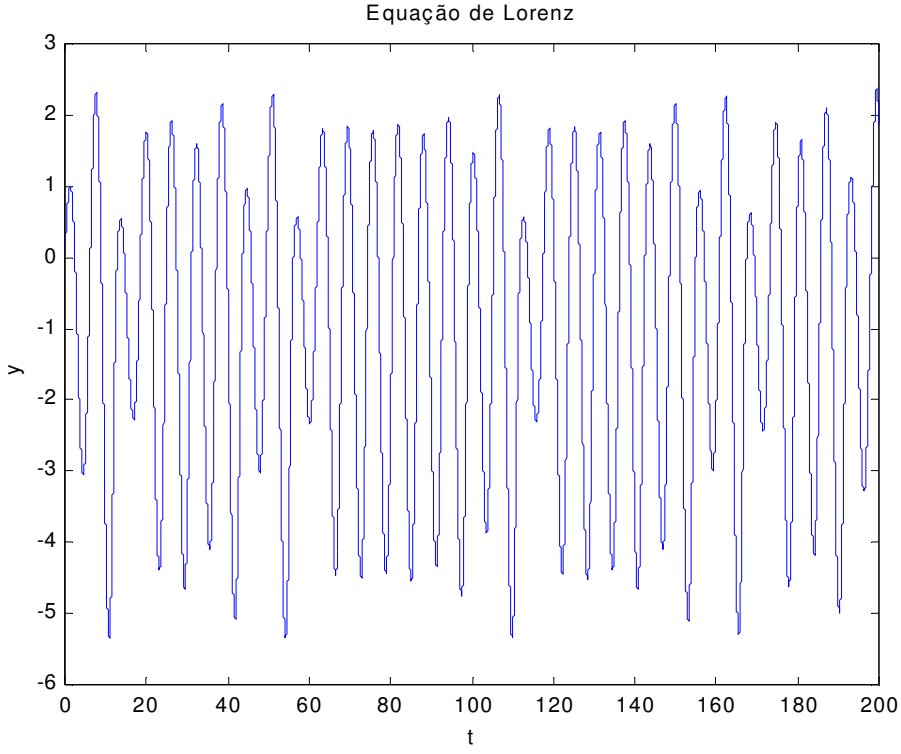
$$\dot{x} = -(y + z)$$
$$\dot{y} = x + .398y$$
$$\dot{z} = 2 + z(x - 4)$$

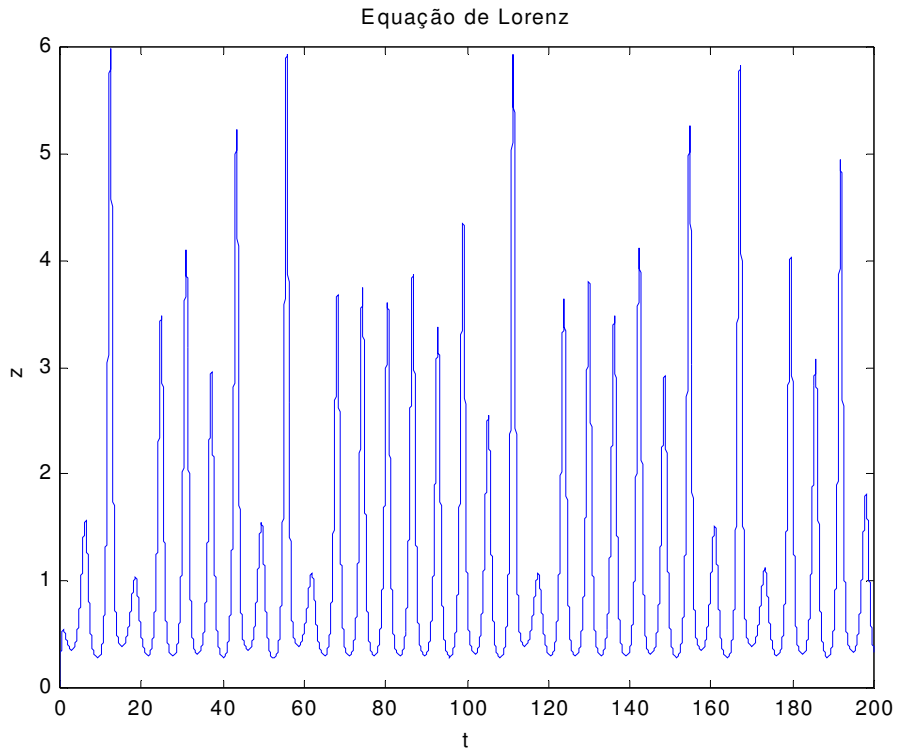
Range Kutta Ordem 4 em C++ 3.1

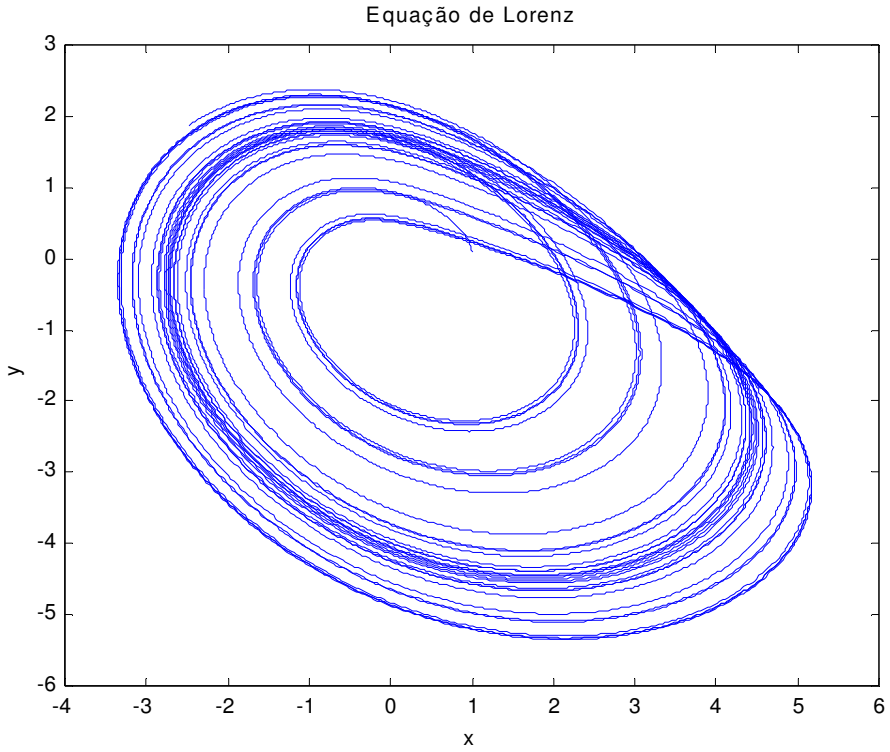
```
// Inicio --- include
#include <stdio.h>
#include <conio.h>
#include <float.h>
// Fim ------ include

// Inicio --- Constantes
#define NOME_DO_ARQUIVO "dados.dat"
#define MODO_DE_ABERTURA "wt"
#define fx(t,x,y,z) (-(y + z))
#define fy(t,y,x) (x + 0.398*(y))
#define fz(t,z,x) (2 + (z)*(x-4))
// Fim ------ Constantes

void main(){
        float   h = 0.01,       // Incremento
                a = 0.0,        // Inicio
                b = 200.0,      // Fim
                t = a,          // Tempo
                x = 1.0, y = 0.1, z = 0.01, x1 = 0.0, y1= 0.0, z1 = 0.0,
                k1, k2, k3, k4;    // variaveis auxiliares
        int     N = (int)((b-a)/h), // Numero de iterações
                i;
        FILE *F0;
        clrscr(); F0=fopen(NOME_DO_ARQUIVO, MODO_DE_ABERTURA);
        fprintf(F0 ,"%f %f %f %f\n", t, x, y, z);
        for (i=1; i <= N; i++) {
           // para x
           k1 = h*fx(t, x, y, z);
           k2 = h*fx((t+h/2.0), (x+k1/2.0), y, z);
           k3 = h*fx((t+h/2.0), (x+k2/2.0), y, z);
           k4 = h*fx((t+h), (x+k3), y, z);
           x1 = x + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;
           // para y
           k1 = h*fy(t, y, x);
           k2 = h*fy((t+h/2.0), (y+k1/2.0), x);
           k3 = h*fy((t+h/2.0), (y+k2/2.0), x);
           k4 = h*fy((t+h), (y+k3), x);
           y1 = y + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;
           // para z
           k1 = h*fz(t, z, x);
           k2 = h*fz((t+h/2.0), (z+k1/2.0), x);
           k3 = h*fz((t+h/2.0), (z+k2/2.0), x);
           k4 = h*fz((t+h), (z+k3), x);
           z1 = z + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;
           t = a + (float)(i)*h;
           x=x1; y=y1; z=z1;
           fprintf(F0 ,"%f %f %f %f\n", t, x, y, z);
        }
        fclose(F0);
}
```

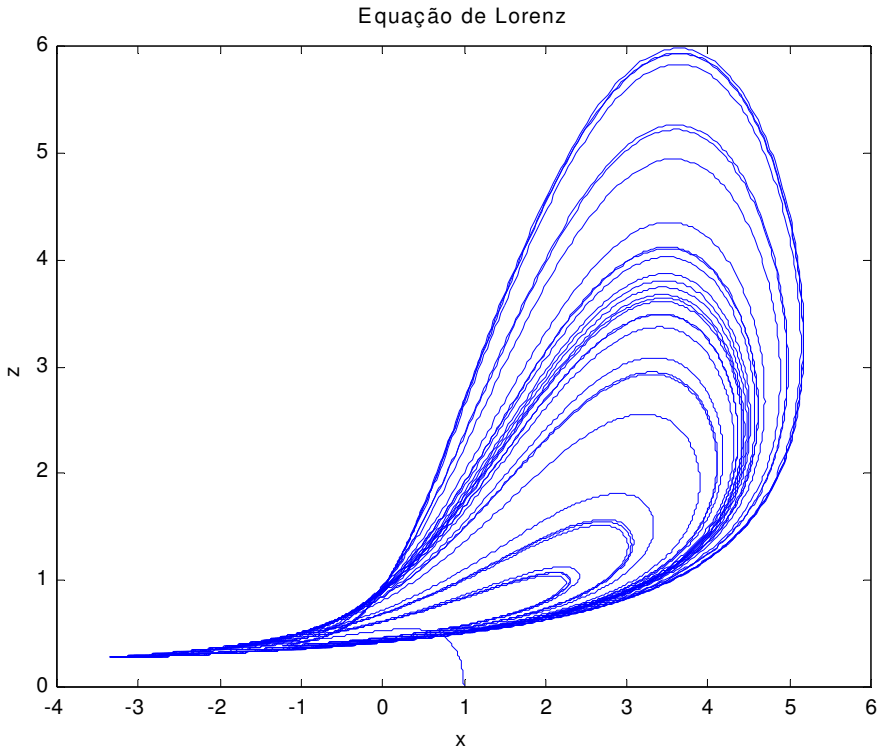
Gráficos

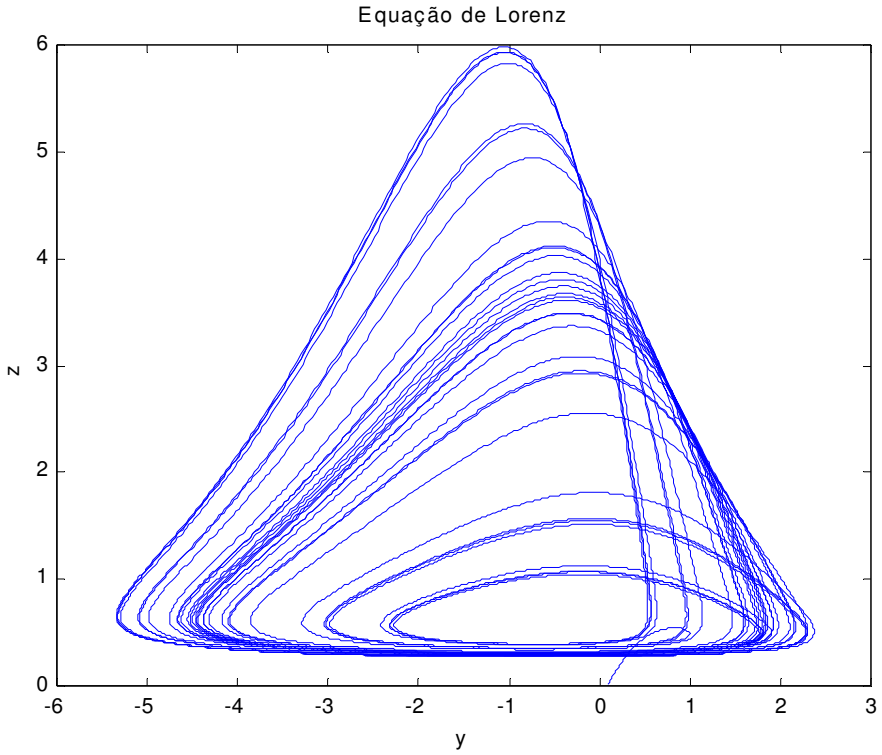

Equação de Lorenz
x
t
6
5
4
3
2
1
0
-1
-2
-3
-4
0
20
40
60
80
100
120
140
160
180
200

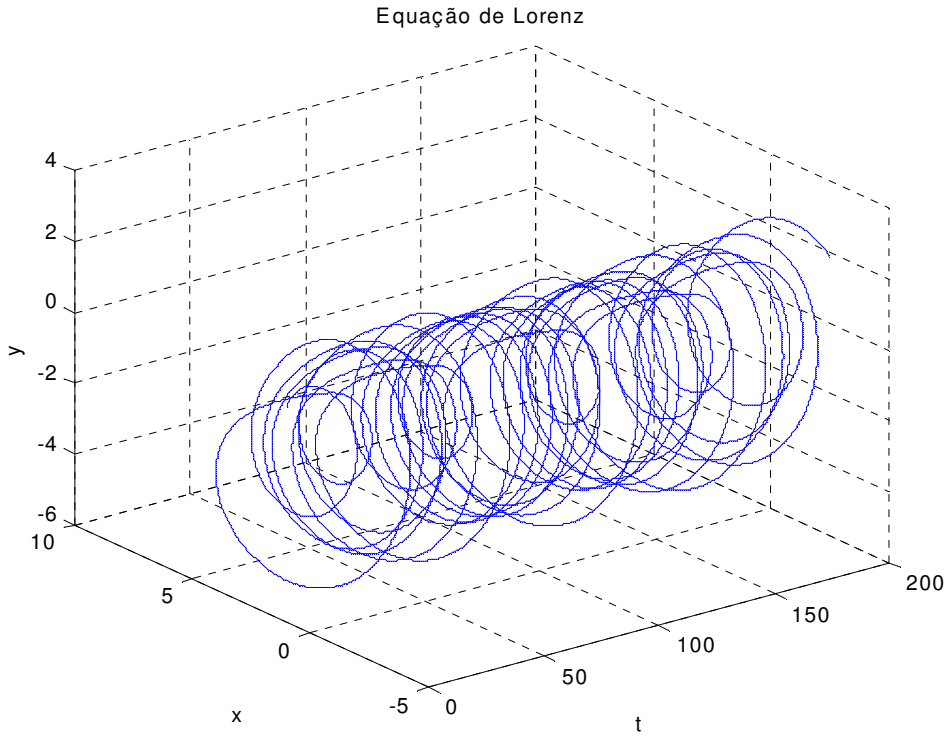

Equação de Lorenz
y
t
3
2
1
0
-1
-2
-3
-4
-5
-6
0
20
40
60
80
100
120
140
160
180
200

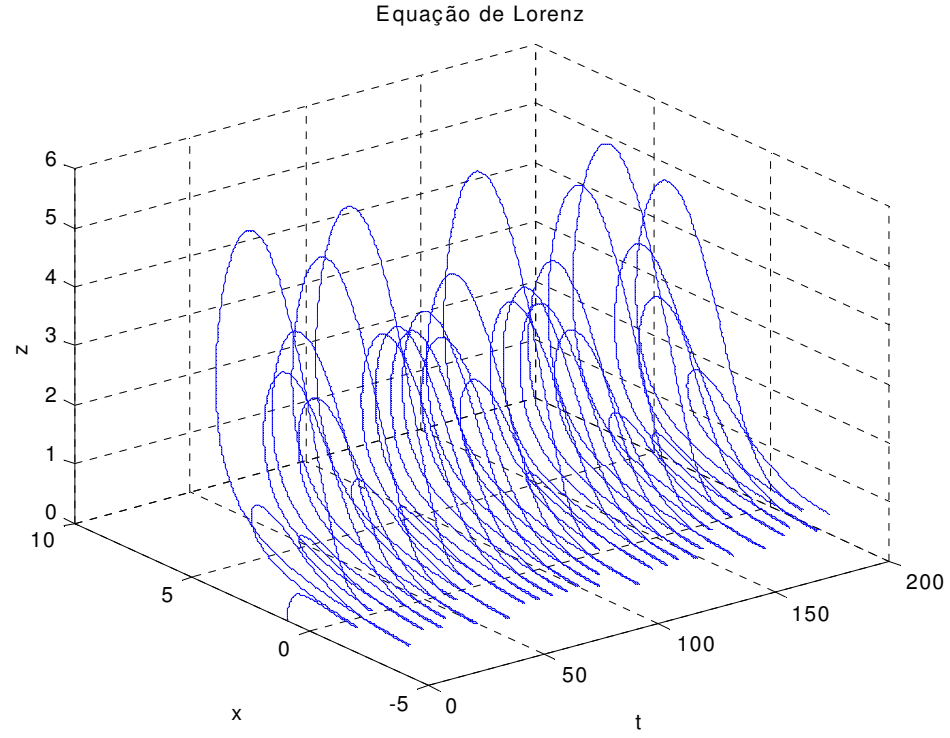

Equação de Lorenz
z
t
0
1
2
3
4
5
6
0
20
40
60
80
100
120
140
160
180
200

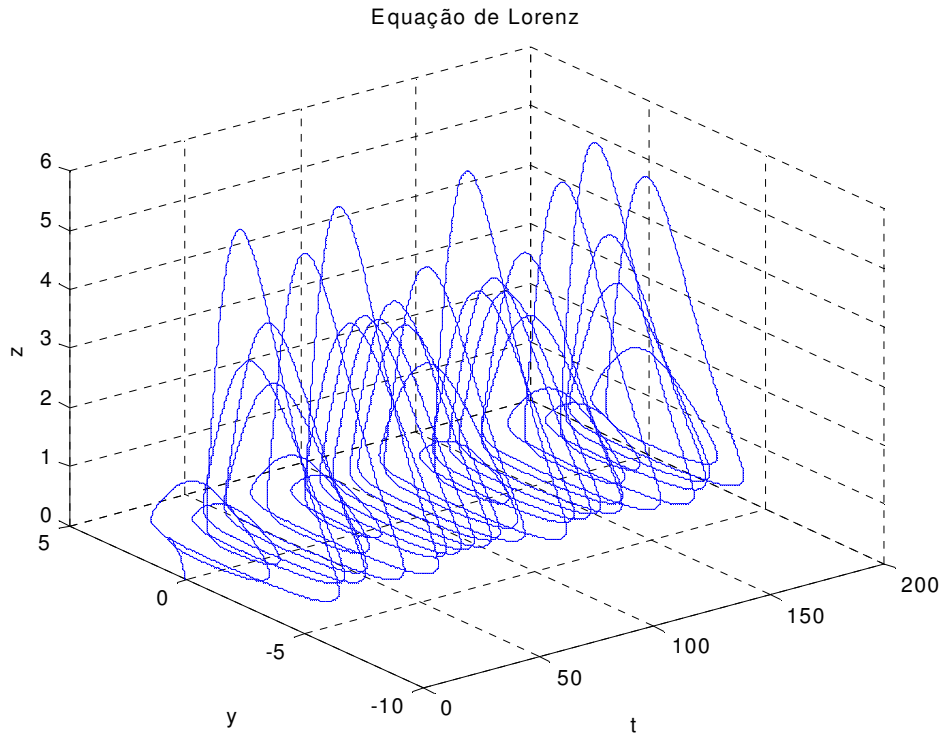

Equação de Lorenz
y
x
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-3
-2
-1
0
1
2
3
4
5
6

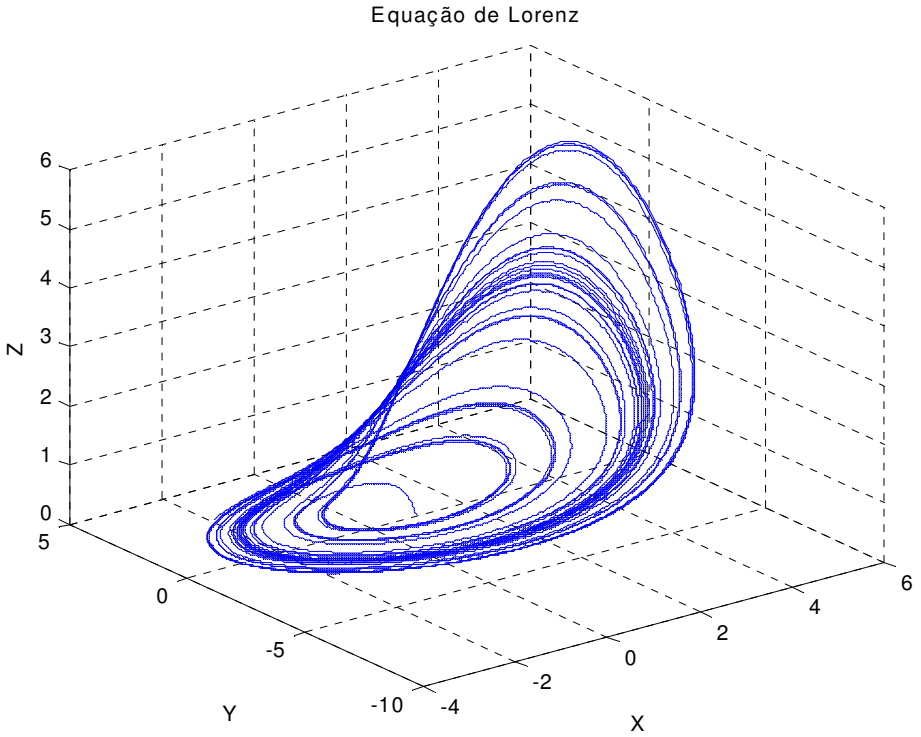

Equação de Lorenz
z
x
6
5
4
3
2
1
0
-4
-3
-2
-1
0
1
2
3
4
5
6


Equação de Lorenz
z
y
6
5
4
3
2
1
0
-6
-5
-4
-3
-2
-1
0
1
2
3


Equação de Lorenz
y
4
2
0
-2
-4
-6
10
5
0
-5
x
0
50
100
150
200
t


Equação de Lorenz
z
6
5
4
3
2
1
0
10
5
0
-5
x
0
50
100
150
200
t


Equação de Lorenz
z
y
t
6
5
4
3
2
1
0
5
0
-5
-10
0
50
100
150
200


Equação de Lorenz
Z
Y
X
6
5
4
3
2
1
0
5
0
-5
-10
-4
-2
0
2
4
6

Barra Flambada

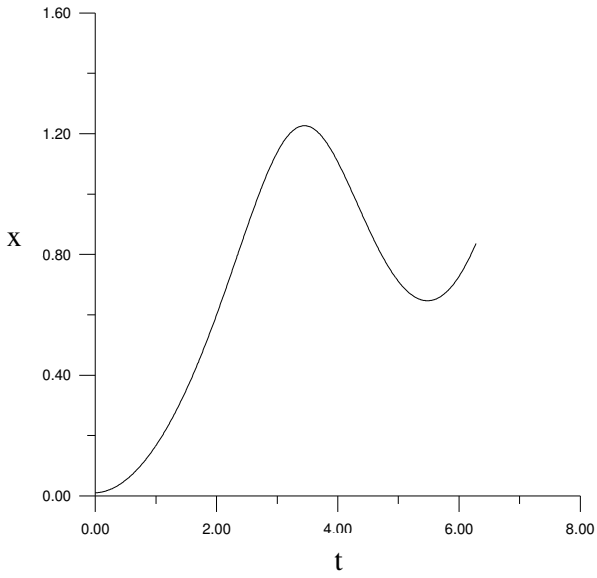
$$\ddot{x} + .15\dot{x} - 1.0x + 1.0x^3 = .3\cos(1.t)$$

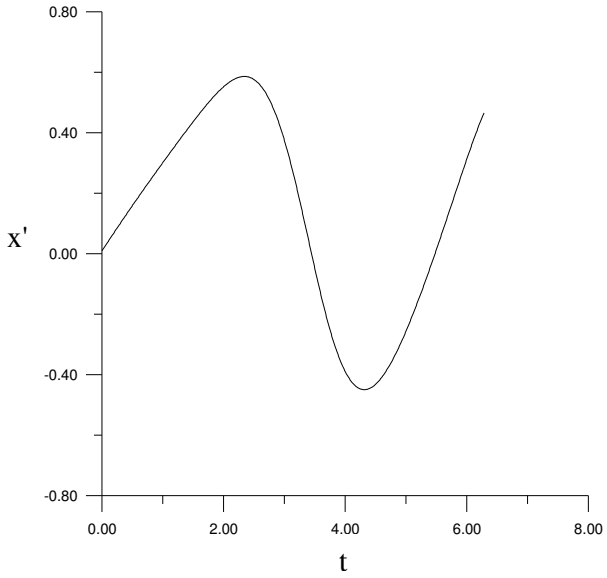
Range Kutta Ordem 4 em C++ 3.1

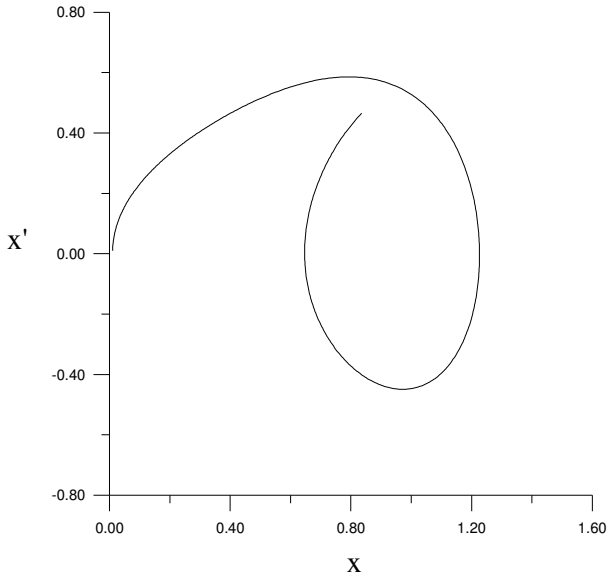
```
// Inicio --- include
#include <stdio.h>
#include <conio.h>
#include <float.h>
#include <math.h>
// Fim ------ include

// Inicio --- Constantes
#define NOME_DO_ARQUIVO "dados.dat"
#define MODO_DE_ABERTURA "wt"
#define fx1(t,x1,x2) (x2)
#define fx2(t,x2,x1) (-0.15*(x2) + (x1) - ((x1)*(x1)*(x1)) + 0.3*(cos(t)))
#define pi 3.141592
// Fim ------ Constantes

void main(){

        float
                h = 0.01,           // Incremento
                a = 0.0,            // Inicio
                b = 100.0*pi,           // Fim
                t = a,              // Tempo
                x1 = 0.01, x2 = 0.01,
                z = 0.0, z2= 0.0,
                k1, k2, k3, k4;     // variaveis auxiliares
                long int  N = (long int)((b-a)/h), // Numero de iteracoes
                i;

        FILE *F0;

        clrscr();

        F0=fopen(NOME_DO_ARQUIVO, MODO_DE_ABERTURA);
        fprintf(F0 ,"%f %f %f \n", t, x1, x2);
        for (i=1; i <= N; i++) {
                // para x1
                k1 = h*fx1(t, x1, x2);
                k2 = h*fx1((t+h/2.0), (x1+k1/2.0), x2);
                k3 = h*fx1((t+h/2.0), (x1+k2/2.0), x2);
                k4 = h*fx1((t+h), (x1+k3), x2);
                z = x1 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;
                // para x2
                k1 = h*fx2(t, x2, x1);
                k2 = h*fx2((t+h/2.0), (x2+k1/2.0), x1);
                k3 = h*fx2((t+h/2.0), (x2+k2/2.0), x1);
                k4 = h*fx2((t+h), (x2+k3), x1);
                z2 = x2 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

                t = a + (float)(i)*h;
                x1=z; x2=z2;

                fprintf(F0 ,"%f %f %f\n", t, x1, x2);
        }
        fclose(F0);
}
```

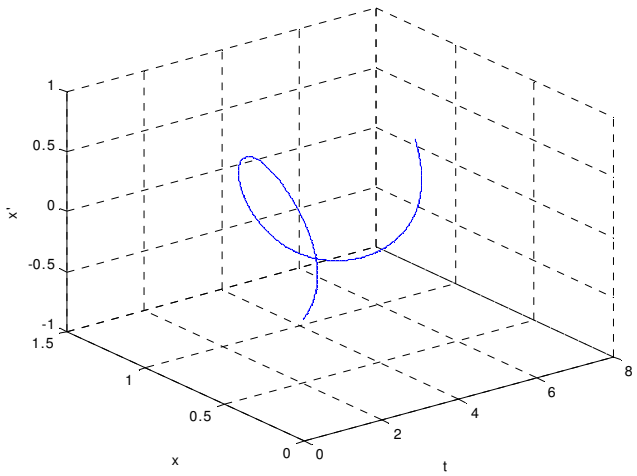
Gráficos

T=1

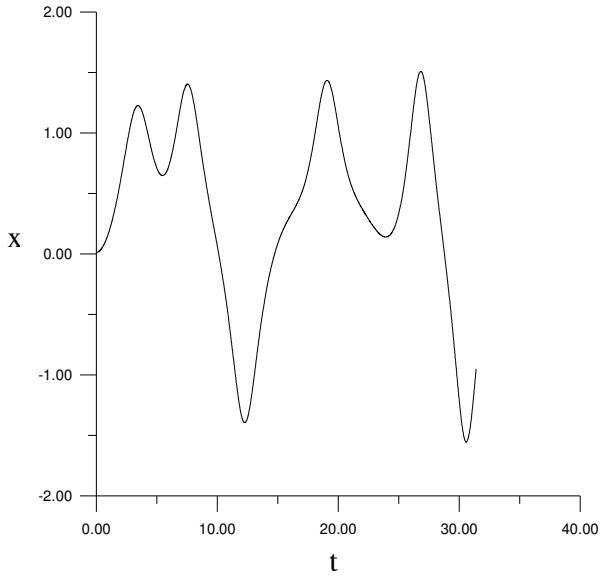

1.60
1.20
0.80
0.40
0.00
x
0.00
2.00
4.00
6.00
8.00
t

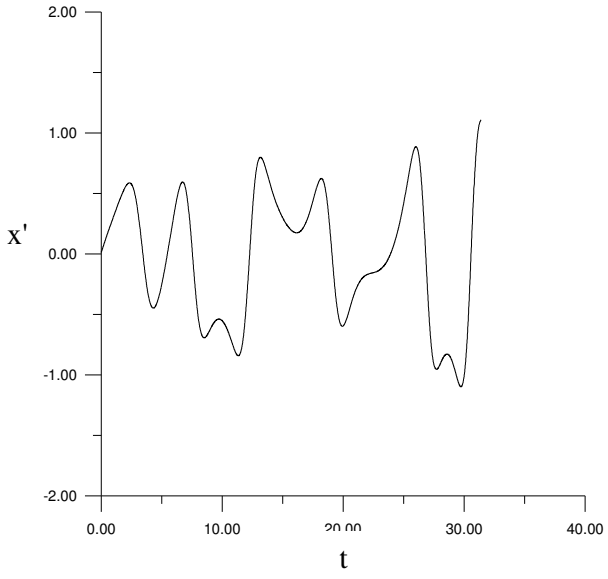

0.80
0.40
0.00
-0.40
-0.80
x'
0.00
2.00
4.00
6.00
8.00
t

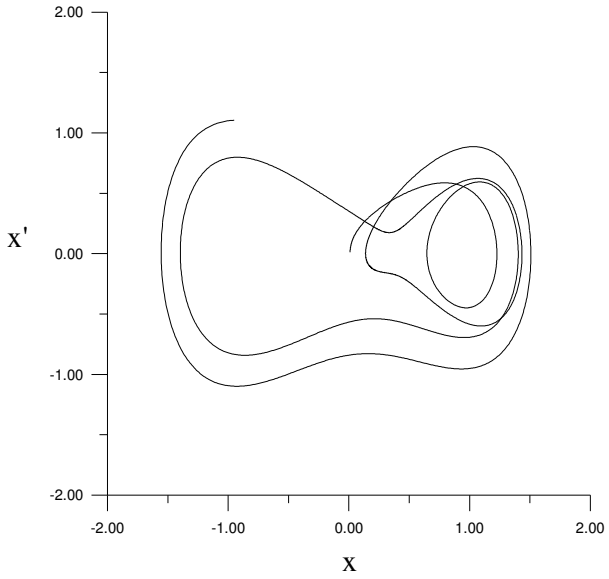

0.80
0.40
0.00
-0.40
-0.80
x'
0.00
0.40
0.80
1.20
1.60
x

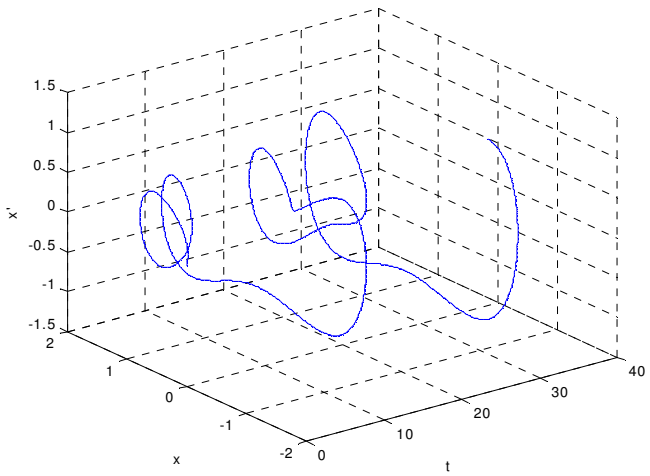

1
0.5
0
-0.5
-1
x'
1.5
1
0.5
0
x
0
2
4
6
8
t

T=5

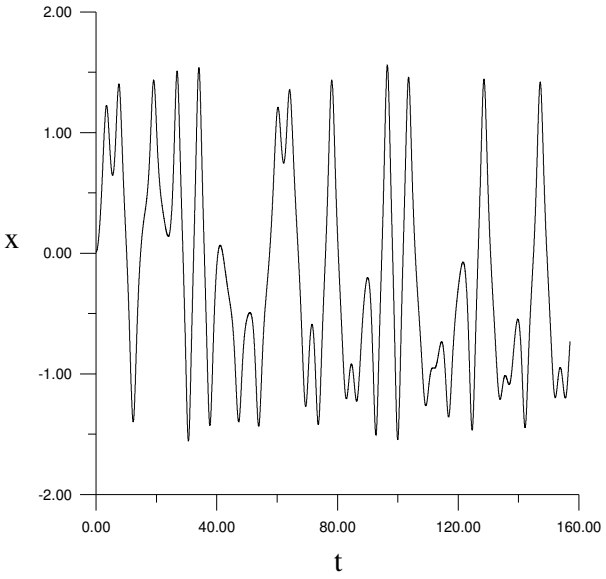

2.00
1.00
0.00
-1.00
-2.00
x
0.00
10.00
20.00
30.00
40.00
t

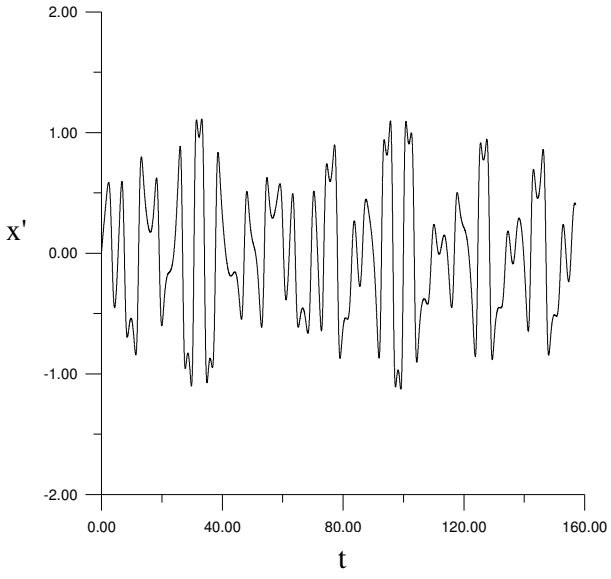

2.00
1.00
0.00
-1.00
-2.00
x'
0.00
10.00
30.00
40.00
t

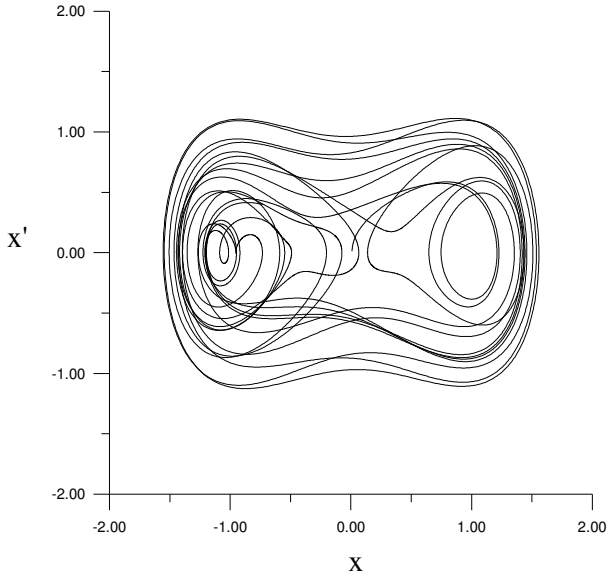

2.00
1.00
0.00
-1.00
-2.00
x'
-2.00
-1.00
0.00
1.00
2.00
x

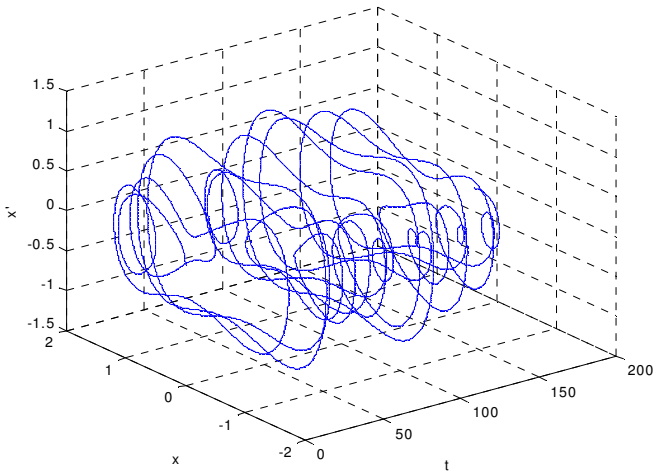

1.5
1
0.5
0
-0.5
-1
-1.5
x'
2
1
0
-1
-2
x
0
10
20
30
40
t

T=25

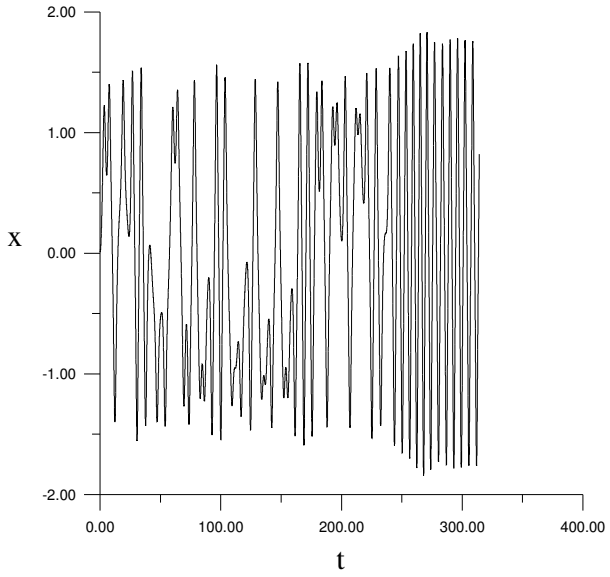

2.00
1.00
0.00
-1.00
-2.00
x
0.00
40.00
80.00
120.00
160.00
t

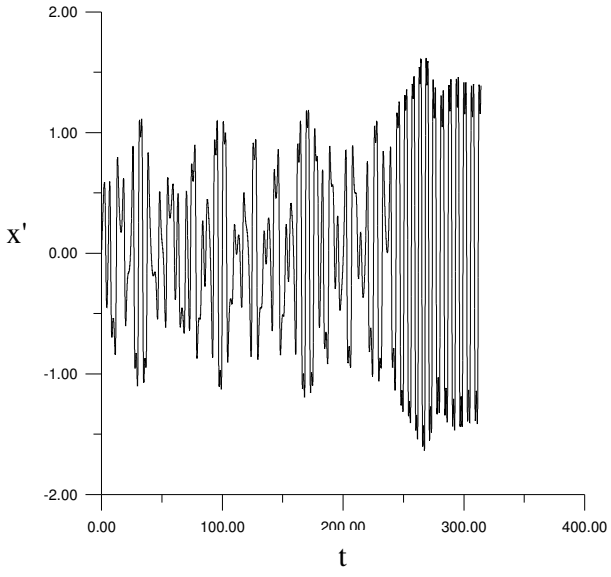

2.00
1.00
0.00
-1.00
-2.00
x'
0.00
40.00
80.00
120.00
160.00
t

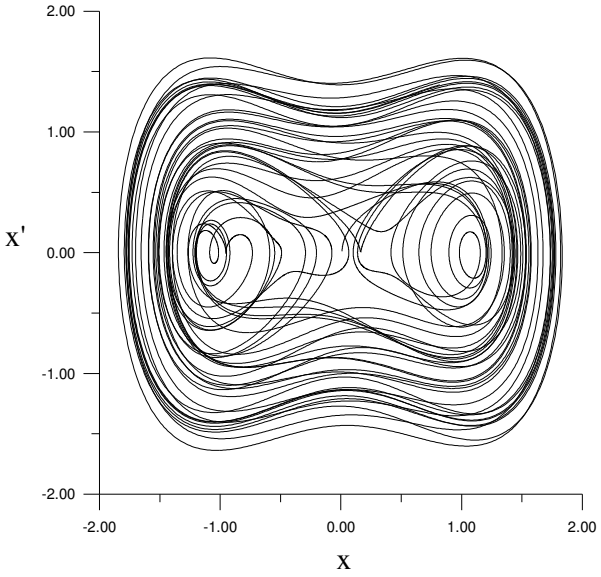

2.00
1.00
0.00
-1.00
-2.00
x'
-2.00
-1.00
0.00
1.00
2.00
x

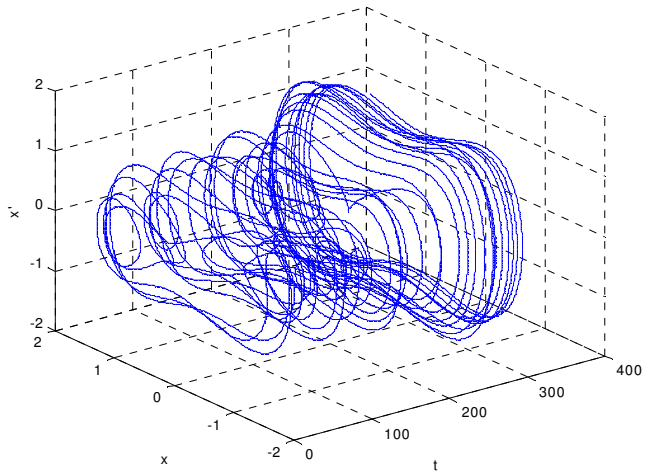

1.5
1
0.5
0
-0.5
-1
-1.5
x'
2
1
0
-1
-2
x
0
50
100
150
200
t

T=50


2.00
1.00
0.00
-1.00
-2.00
x
0.00
100.00
200.00
300.00
400.00
t


2.00
1.00
0.00
-1.00
-2.00
x'
0.00
100.00
300.00
400.00
t


2.00
1.00
0.00
-1.00
-2.00
x'
-2.00
-1.00
0.00
1.00
2.00
x


2
1
0
-1
-2
x'
2
1
0
-1
-2
x
0
100
200
300
400
t

Mola Rígida

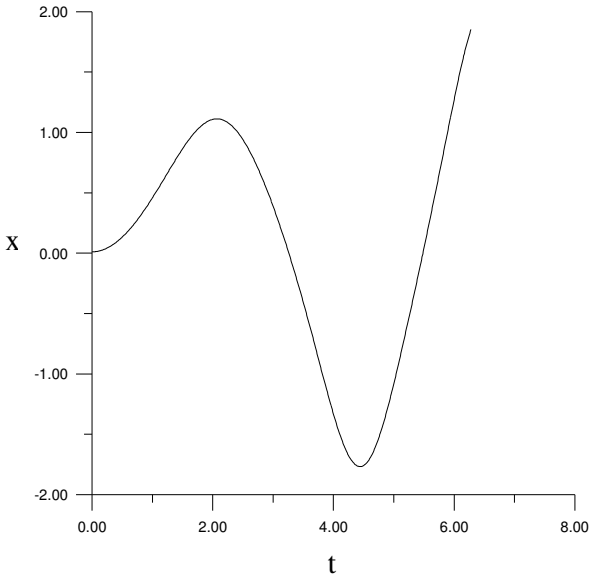
$$\ddot{x} + .1\dot{x} + 0.0x + 1.0x^3 = 1.0\cos(1.0t)$$

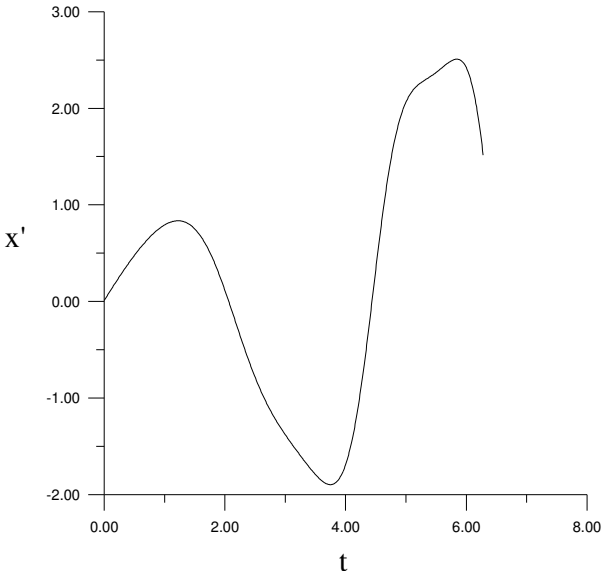
Range Kutta Ordem 4 em C++ 3.1

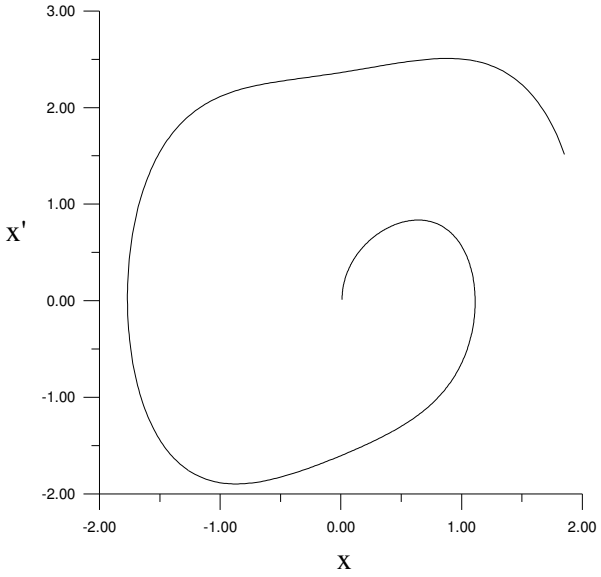
```
// Inicio --- include
#include <stdio.h>
#include <conio.h>
#include <float.h>
#include <math.h>
// Fim ------ include

// Inicio --- Constantes
#define NOME_DO_ARQUIVO "dados.dat"
#define MODO_DE_ABERTURA "wt"
#define fx1(t,x1,x2) (x2)
#define fx2(t,x2,x1) (-0.1*(x2) -(x1)*(x1)*(x1) + cos(t))
#define pi 3.141592
// Fim ------ Constantes

void main(){

        float
                h = 0.01,           // Incremento
                a = 0.0,            // Inicio
                b = 100.0*pi,           // Fim
                t = a,              // Tempo
                x1 = 0.01, x2 = 0.01,
                z = 0.0, z2= 0.0,
                k1, k2, k3, k4;     // variaveis auxiliares
                long int  N = (long int)((b-a)/h), // Numero de iteracoes
                i;

        FILE *F0;

        clrscr();

        F0=fopen(NOME_DO_ARQUIVO, MODO_DE_ABERTURA);
        fprintf(F0 ,"%f %f %f \n", t, x1, x2);
        for (i=1; i <= N; i++) {
                // para x1
                k1 = h*fx1(t, x1, x2);
                k2 = h*fx1((t+h/2.0), (x1+k1/2.0), x2);
                k3 = h*fx1((t+h/2.0), (x1+k2/2.0), x2);
                k4 = h*fx1((t+h), (x1+k3), x2);
                z = x1 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;
                // para x2
                k1 = h*fx2(t, x2, x1);
                k2 = h*fx2((t+h/2.0), (x2+k1/2.0), x1);
                k3 = h*fx2((t+h/2.0), (x2+k2/2.0), x1);
                k4 = h*fx2((t+h), (x2+k3), x1);
                z2 = x2 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

                t = a + (float)(i)*h;
                x1=z; x2=z2;

                fprintf(F0 ,"%f %f %f\n", t, x1, x2);
        }
        fclose(F0);
}
```

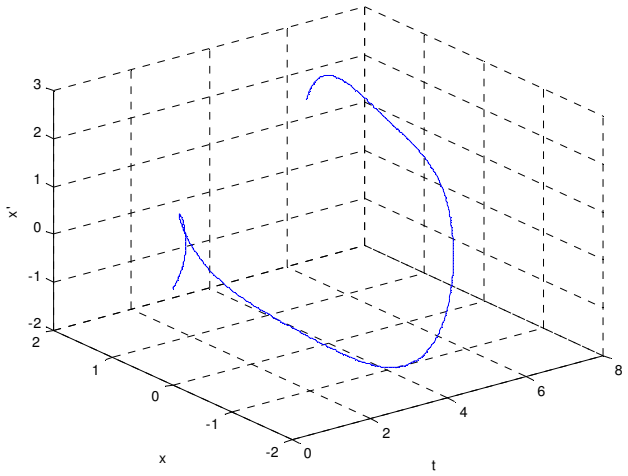
Gráficos

T=1

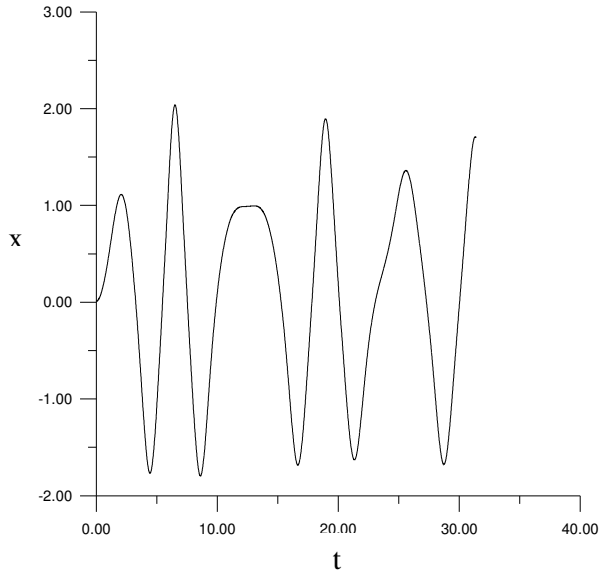

x
2.00
1.00
0.00
-1.00
-2.00
0.00
2.00
4.00
6.00
8.00
t

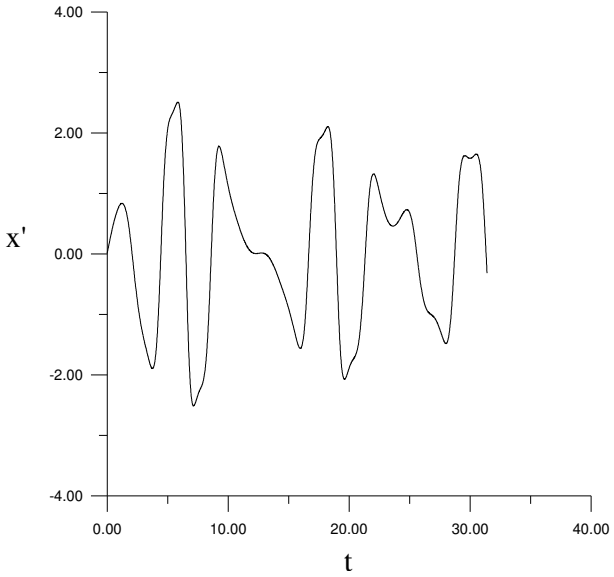

x'
3.00
2.00
1.00
0.00
-1.00
-2.00
0.00
2.00
4.00
6.00
8.00
t

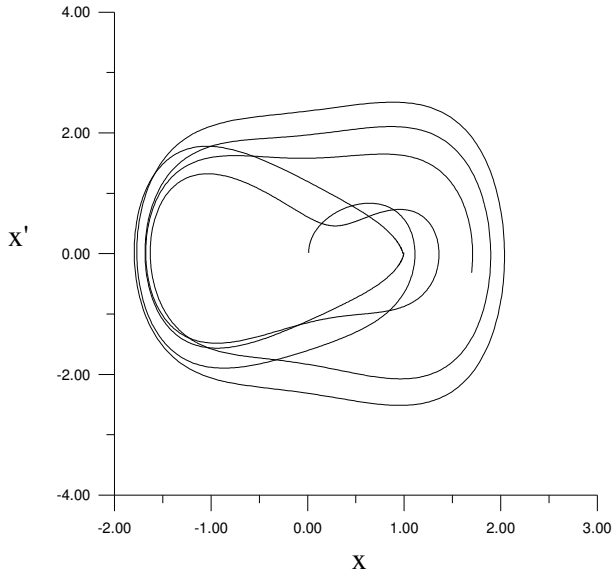

x'
3.00
2.00
1.00
0.00
-1.00
-2.00
-2.00
-1.00
0.00
1.00
2.00
x

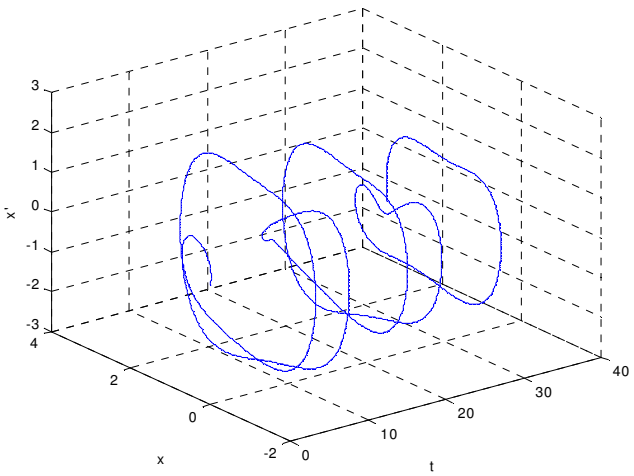

x'
3
2
1
0
-1
-2
x
2
1
0
-1
-2
t
0
2
4
6
8

T=5

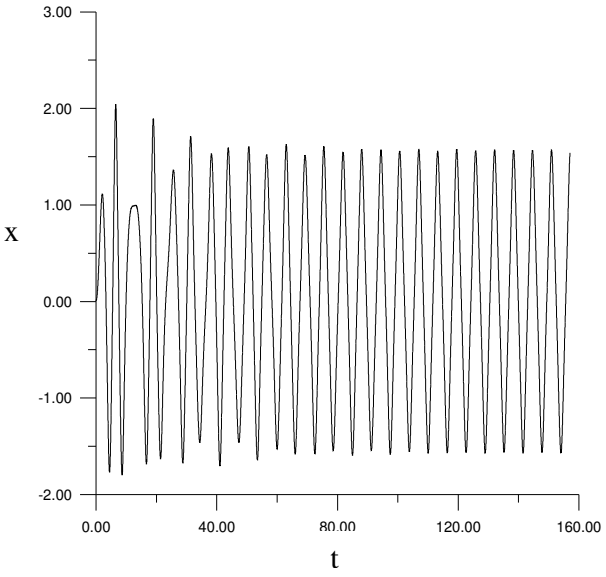

3.00
2.00
1.00
0.00
-1.00
-2.00
x
0.00
10.00
20.00
30.00
40.00
t

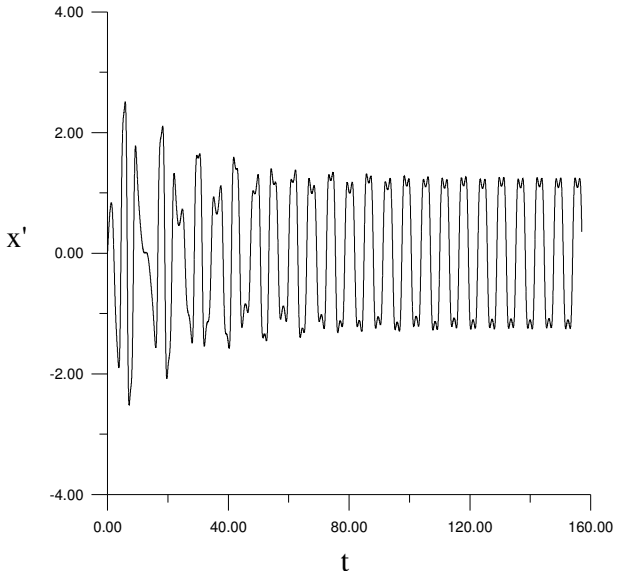

4.00
2.00
0.00
-2.00
-4.00
x'
0.00
10.00
20.00
30.00
40.00
t

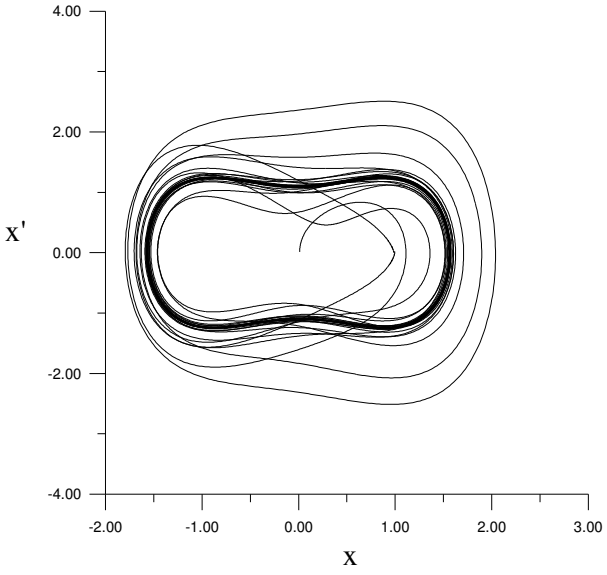

4.00
2.00
0.00
-2.00
-4.00
x'
-2.00
-1.00
0.00
1.00
2.00
3.00
x

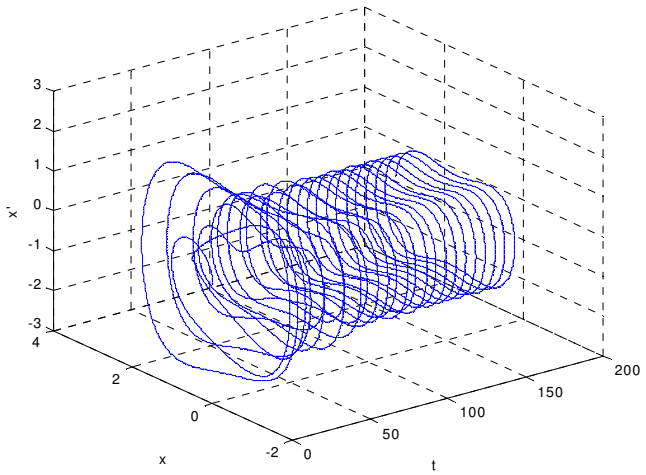

3
2
1
0
-1
-2
-3
x'
4
2
0
-2
x
0
10
20
30
40
t

T=25

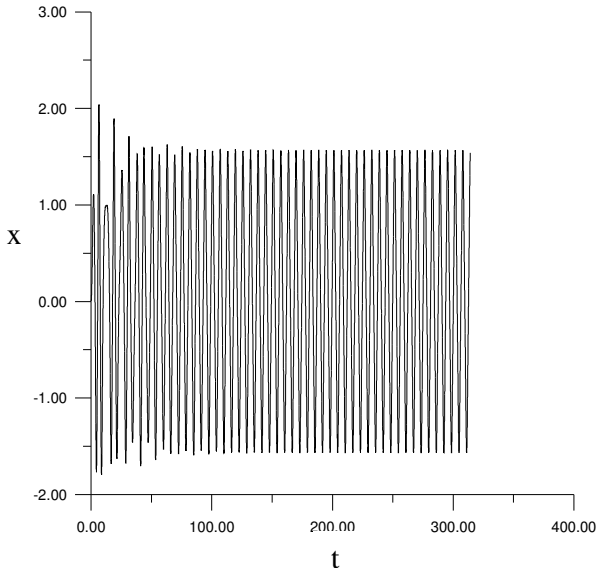

3.00
2.00
1.00
0.00
-1.00
-2.00
x
0.00
40.00
80.00
120.00
160.00
t

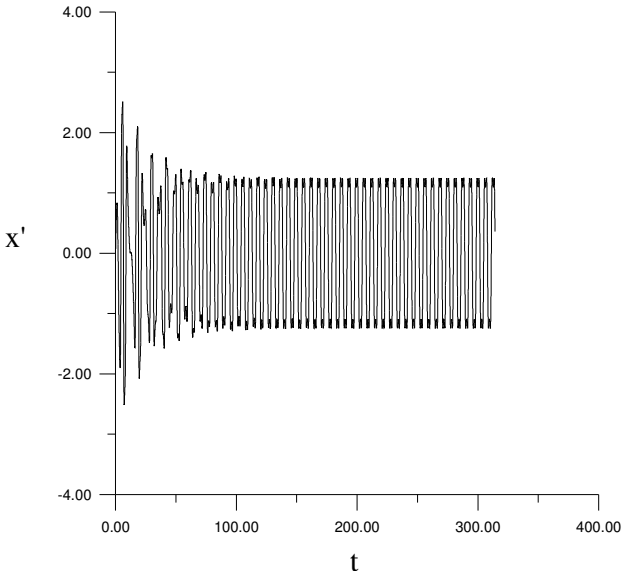

4.00
2.00
0.00
-2.00
-4.00
x'
0.00
40.00
80.00
120.00
160.00
t

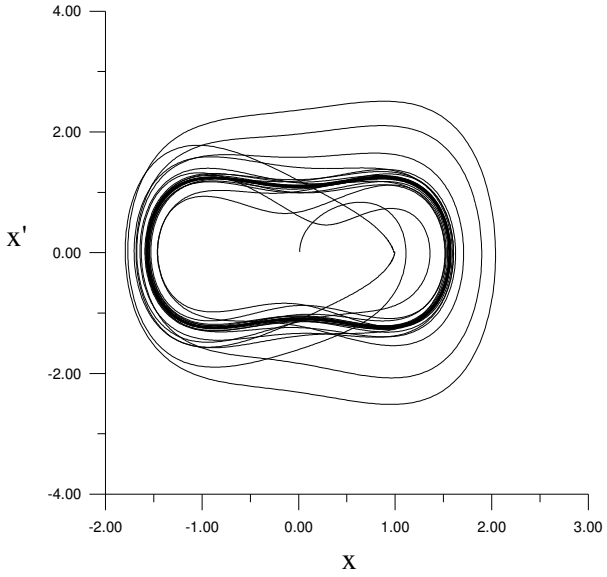

4.00
2.00
0.00
-2.00
-4.00
x'
-2.00
-1.00
0.00
1.00
2.00
3.00
x

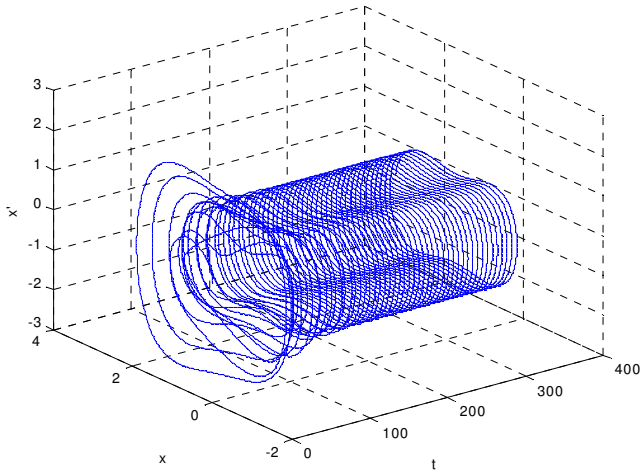

3
2
1
0
-1
-2
-3
x'
4
2
0
-2
x
0
50
100
150
200
t

T=50


3.00
2.00
1.00
0.00
-1.00
-2.00
x
0.00
100.00
200.00
300.00
400.00
t


4.00
2.00
0.00
-2.00
-4.00
x'
0.00
100.00
200.00
300.00
400.00
t


4.00
2.00
0.00
-2.00
-4.00
x'
-2.00
-1.00
0.00
1.00
2.00
3.00
x


3
2
1
0
-1
-2
-3
x'
4
2
0
-2
x
0
100
200
300
400
t

Equação de Vanderpol

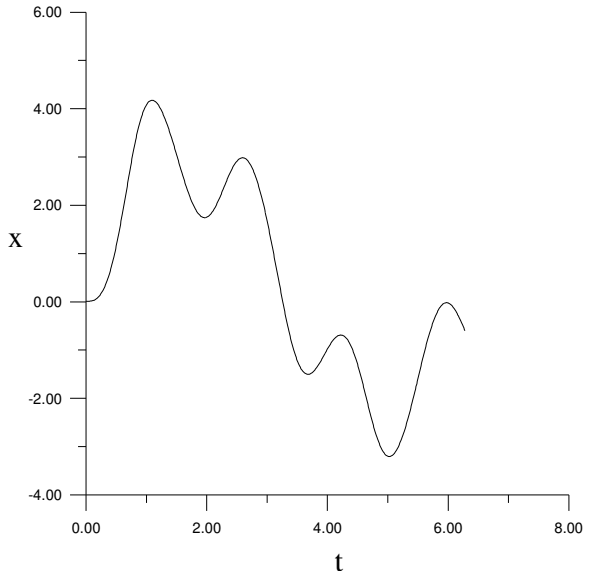
$$\ddot{x} + .2(x^2 - 1)\dot{x} + x = 17.0sin(4t)$$

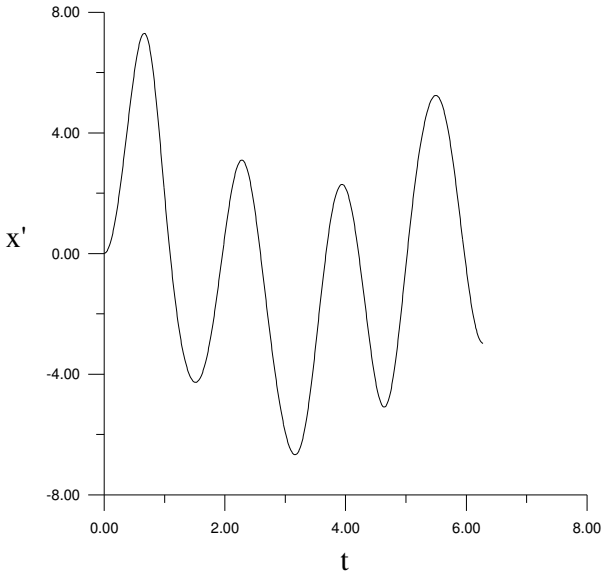
Range Kutta Ordem 4 em C++ 3.1

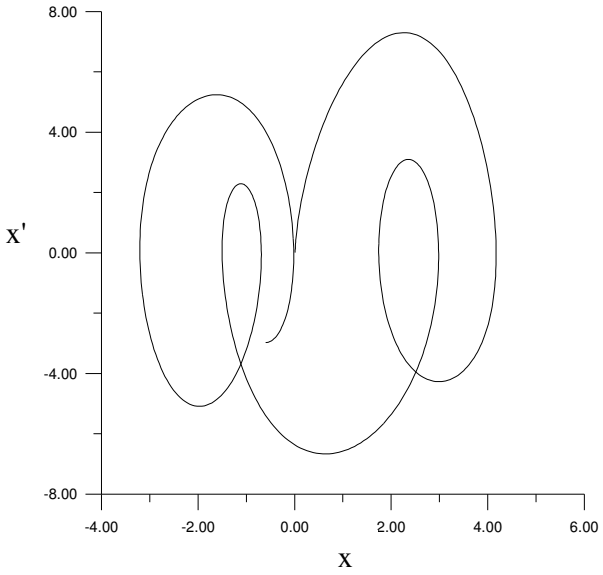
```
// Inicio --- include
#include <stdio.h>
#include <conio.h>
#include <float.h>
#include <math.h>
// Fim ------ include

// Inicio --- Constantes
#define NOME_DO_ARQUIVO "dados.dat"
#define MODO_DE_ABERTURA "wt"
#define fx1(t,x1,x2) (x2)
#define fx2(t,x2,x1) (-0.2*(x2)*((x1)*(x1)-1.0) - (x1) + 17.0*(sin(4.0*(t))))
#define pi 3.141592
// Fim ------ Constantes

void main(){

	float
		h = 0.01,		// Incremento
		a = 0.0,		// Inicio
		b = 100*pi,		// Fim
		t = a,		// Tempo
		x1 = 0.01, x2 = 0.01,
		z = 0.0, z2= 0.0,
		k1, k2, k3, k4;	// variaveis auxiliares
		long int N = (long int)((b-a)/h), // Numero de iteracoes
		i;

	FILE *F0;

	clrscr();

	F0=fopen(NOME_DO_ARQUIVO, MODO_DE_ABERTURA);
	fprintf(F0 ,"%f %f %f \n", t, x1, x2);
	for (i=1; i <= N; i++) {
		// para x1
		k1 = h*fx1(t, x1, x2);
		k2 = h*fx1((t+h/2.0), (x1+k1/2.0), x2);
		k3 = h*fx1((t+h/2.0), (x1+k2/2.0), x2);
		k4 = h*fx1((t+h), (x1+k3), x2);
		z = x1 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;
		// para x2
		k1 = h*fx2(t, x2, x1);
		k2 = h*fx2((t+h/2.0), (x2+k1/2.0), x1);
		k3 = h*fx2((t+h/2.0), (x2+k2/2.0), x1);
		k4 = h*fx2((t+h), (x2+k3), x1);
		z2 = x2 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

		t = a + (float)(i)*h;
		x1=z; x2=z2;

		fprintf(F0 ,"%f %f %f\n", t, x1, x2);
	}
	fclose(F0);
}
```

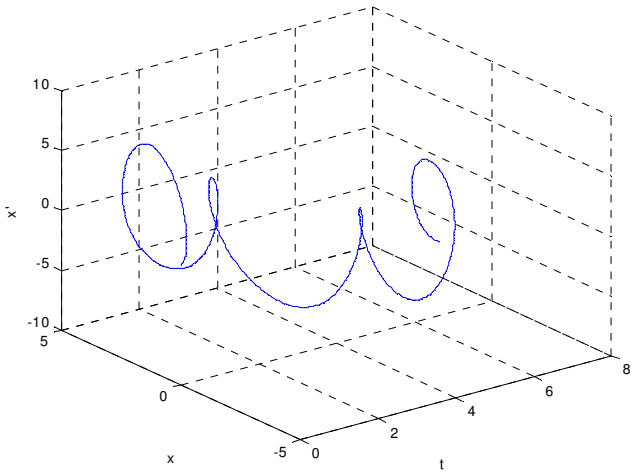
Gráficos

T=1

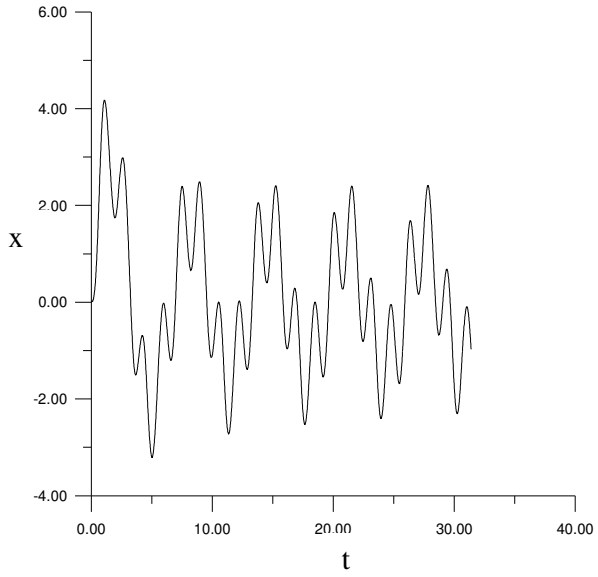

6.00
4.00
2.00
0.00
-2.00
-4.00
x
0.00
2.00
4.00
6.00
8.00
t

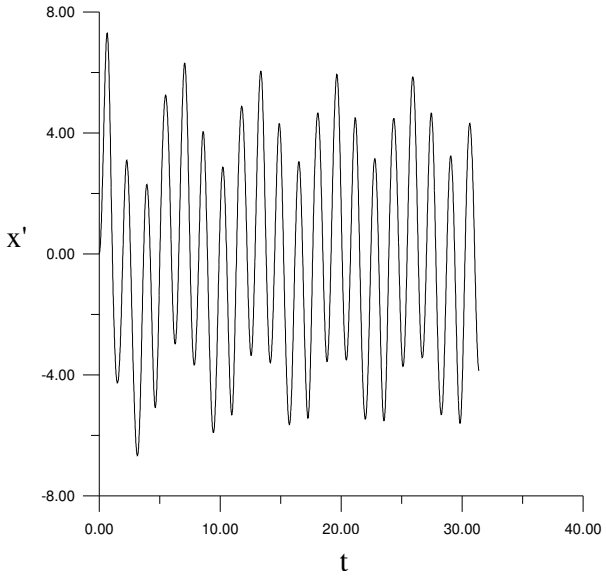

8.00
4.00
0.00
-4.00
-8.00
x'
0.00
2.00
4.00
6.00
8.00
t

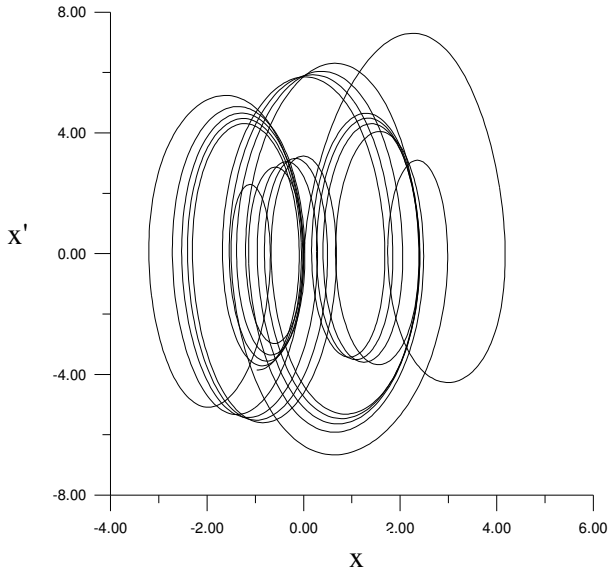

8.00
4.00
0.00
-4.00
-8.00
x'
-4.00
-2.00
0.00
2.00
4.00
6.00
x

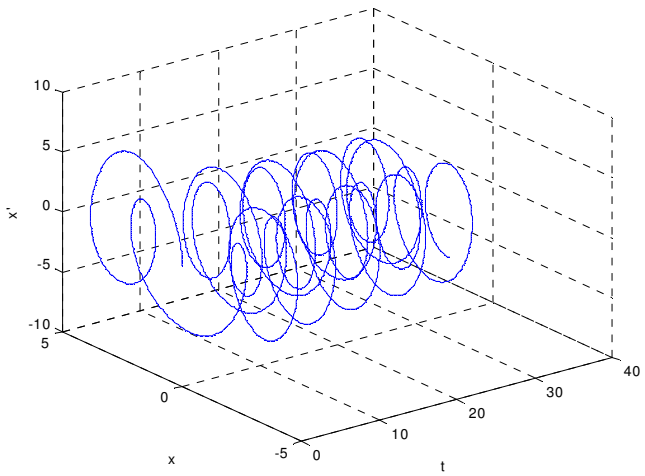

10
5
0
-5
-10
x'
5
0
-5
x
0
2
4
6
8
t

T=5

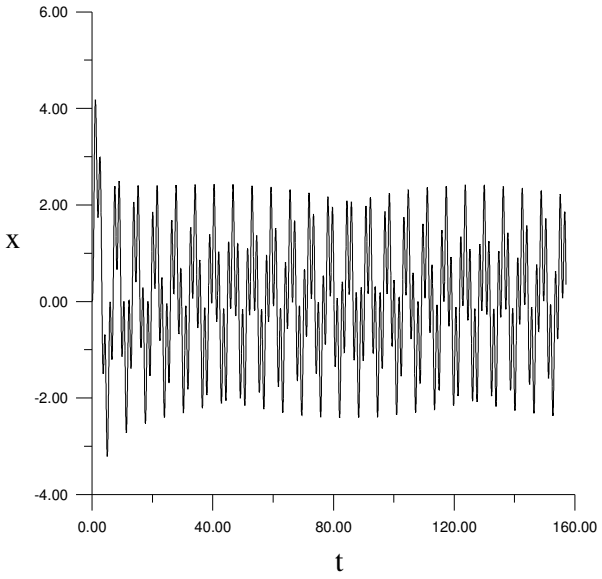

6.00
4.00
2.00
0.00
-2.00
-4.00
x
0.00
10.00
20.00
30.00
40.00
t

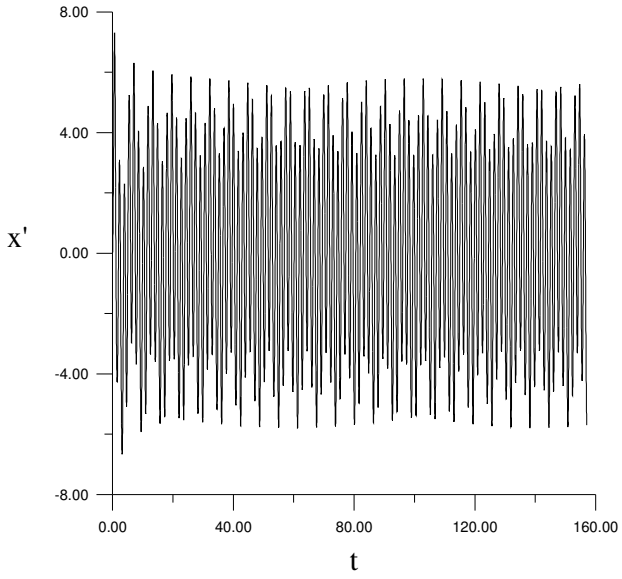

8.00
4.00
0.00
-4.00
-8.00
x'
0.00
10.00
20.00
30.00
40.00
t

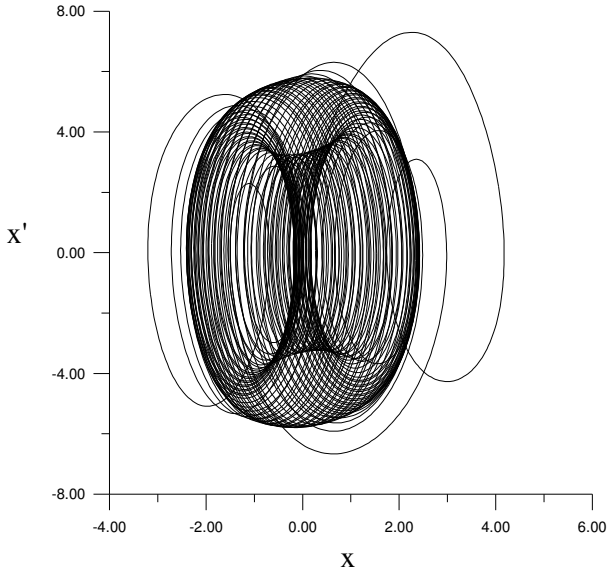

8.00
4.00
0.00
-4.00
-8.00
x'
-4.00
-2.00
0.00
2.00
4.00
6.00
x

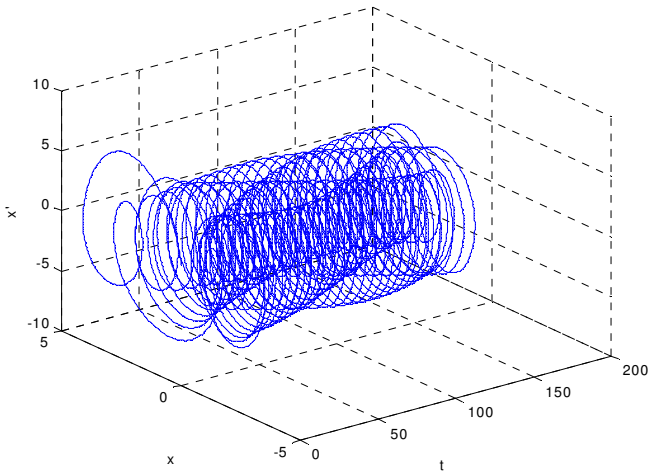

10
5
0
-5
-10
x'
5
0
-5
x
0
10
20
30
40
t

T=25

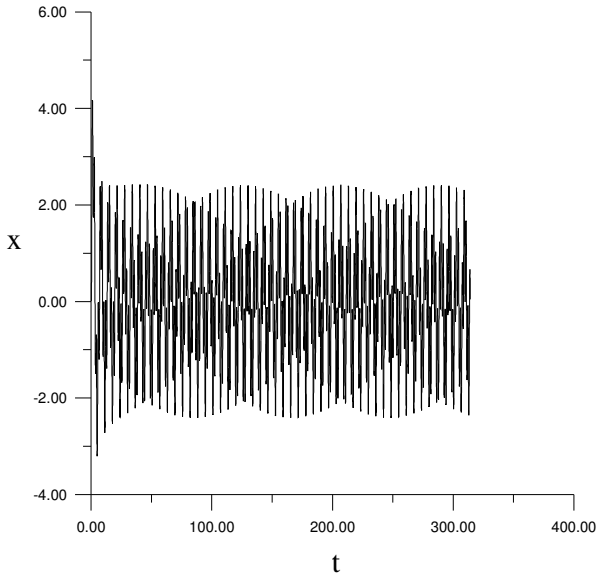

6.00
4.00
2.00
0.00
-2.00
-4.00
x
0.00
40.00
80.00
120.00
160.00
t

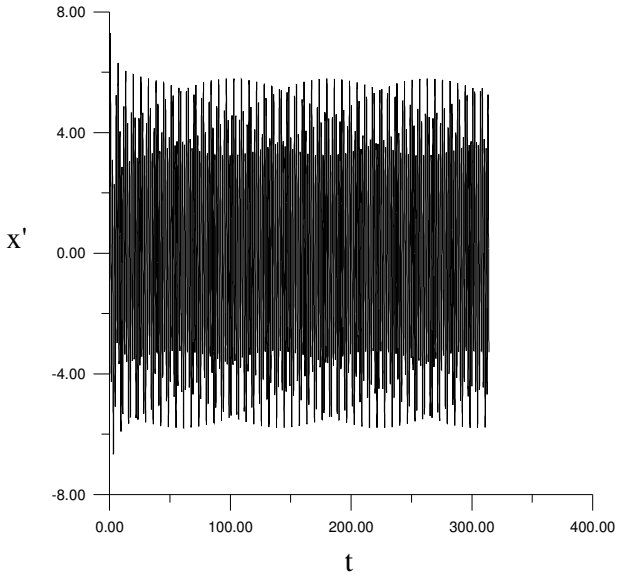

8.00
4.00
0.00
-4.00
-8.00
x'
0.00
40.00
80.00
120.00
160.00
t

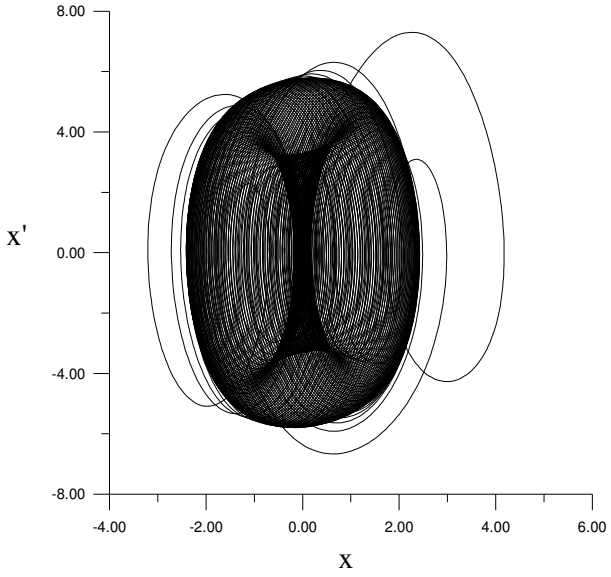

8.00
4.00
0.00
-4.00
-8.00
x'
-4.00
-2.00
0.00
2.00
4.00
6.00
x

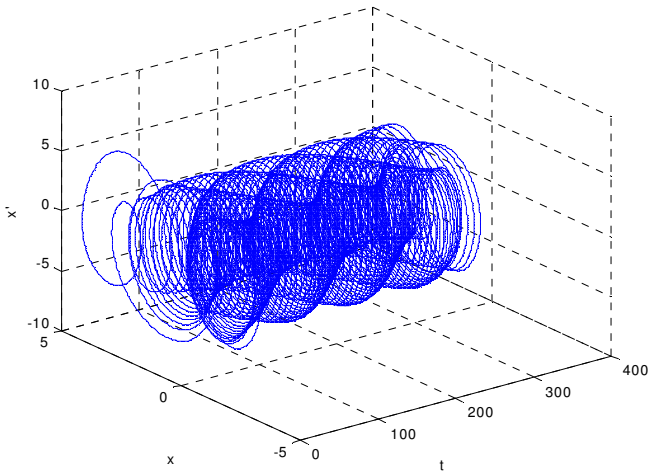

10
5
0
-5
-10
x'
5
0
-5
x
0
50
100
150
200
t

T=50


6.00
4.00
2.00
0.00
-2.00
-4.00
x
0.00
100.00
200.00
300.00
400.00
t


8.00
4.00
0.00
-4.00
-8.00
x'
0.00
100.00
200.00
300.00
400.00
t


8.00
4.00
0.00
-4.00
-8.00
x'
-4.00
-2.00
0.00
2.00
4.00
6.00
x


10
5
0
-5
-10
x'
5
0
-5
x
0
100
200
300
400
t

Equação

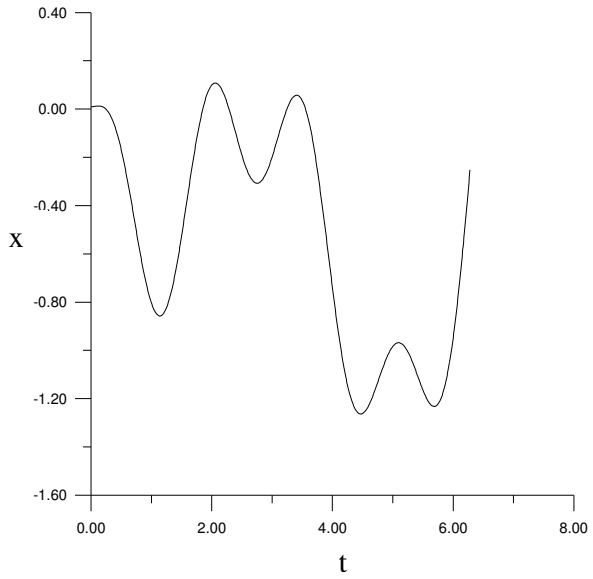
$$\ddot{x} + .1\dot{x} + \text{sen}(x) + 4.6sin(4t) = 1.0\cos(1.0t)$$

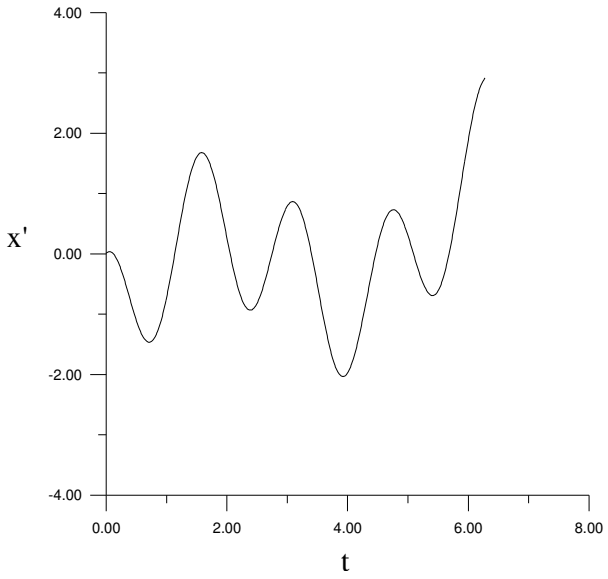
Range Kutta Ordem 4 em C++ 3.1

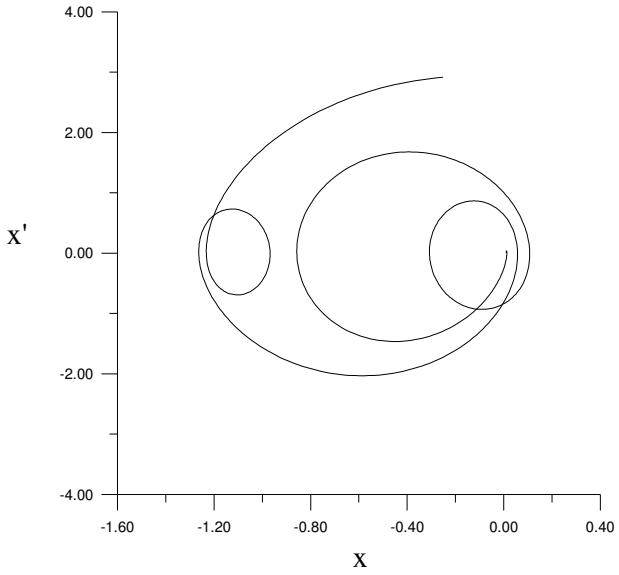
```
// Inicio --- include
#include <stdio.h>
#include <conio.h>
#include <float.h>
#include <math.h>
// Fim ------ include

// Inicio --- Constantes
#define NOME_DO_ARQUIVO "dados.dat"
#define MODO_DE_ABERTURA "wt"
#define fx1(t,x1,x2) (x2)
#define fx2(t,x2,x1) (-0.1*(x2)-sin(x1)-4.6*sin(4.0*t)+cos(t))
#define pi 3.141592
// Fim ------ Constantes

void main(){

        float
                h = 0.01,           // Incremento
                a = 0.0,            // Inicio
                b = 100.0*pi,           // Fim
                t = a,              // Tempo
                x1 = 0.01, x2 = 0.01,
                z = 0.0, z2= 0.0,
                k1, k2, k3, k4;     // variaveis auxiliares
                long int  N = (long int)((b-a)/h), // Numero de iteracoes
                i;

        FILE *F0;

        clrscr();

        F0=fopen(NOME_DO_ARQUIVO, MODO_DE_ABERTURA);
        fprintf(F0 ,"%f %f %f \n", t, x1, x2);
        for (i=1; i <= N; i++) {
                // para x1
                k1 = h*fx1(t, x1, x2);
                k2 = h*fx1((t+h/2.0), (x1+k1/2.0), x2);
                k3 = h*fx1((t+h/2.0), (x1+k2/2.0), x2);
                k4 = h*fx1((t+h), (x1+k3), x2);
                z = x1 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;
                // para x2
                k1 = h*fx2(t, x2, x1);
                k2 = h*fx2((t+h/2.0), (x2+k1/2.0), x1);
                k3 = h*fx2((t+h/2.0), (x2+k2/2.0), x1);
                k4 = h*fx2((t+h), (x2+k3), x1);
                z2 = x2 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

                t = a + (float)(i)*h;
                x1=z; x2=z2;

                fprintf(F0 ,"%f %f %f\n", t, x1, x2);
        }
        fclose(F0);
}
```

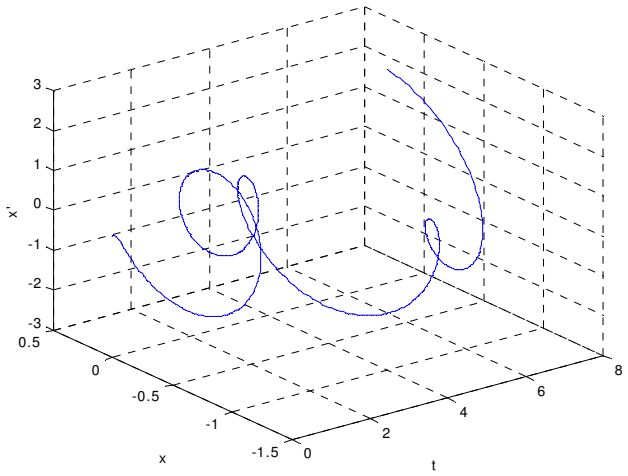
Gráficos

T=1

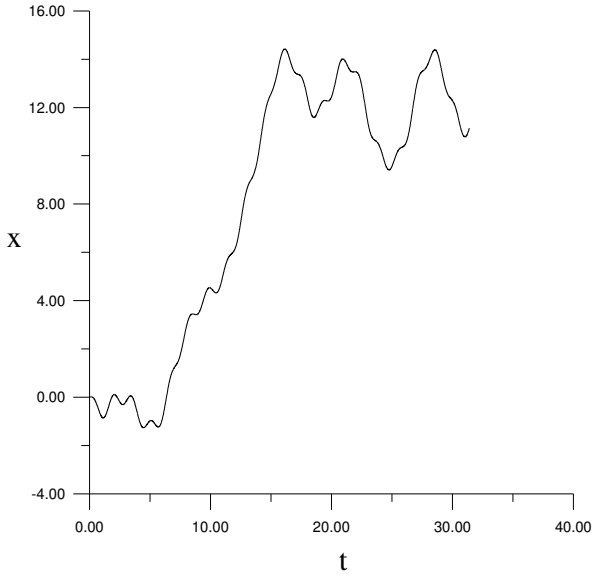

0.40
0.00
-0.40
-0.80
-1.20
-1.60
x
0.00
2.00
4.00
6.00
8.00
t

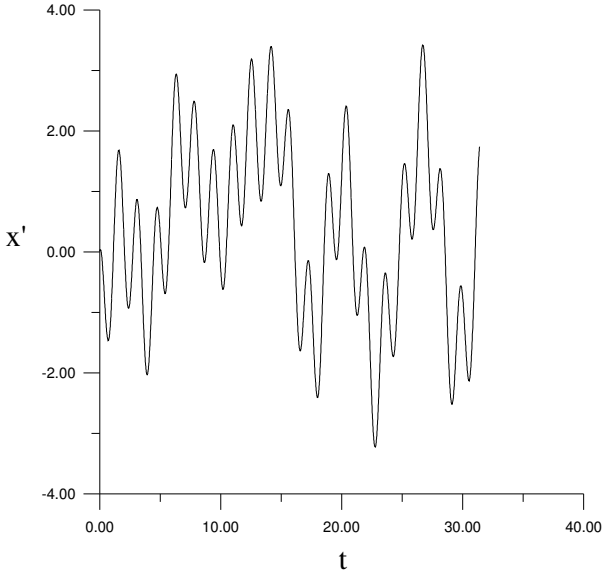

4.00
2.00
0.00
-2.00
-4.00
x'
0.00
2.00
4.00
6.00
8.00
t

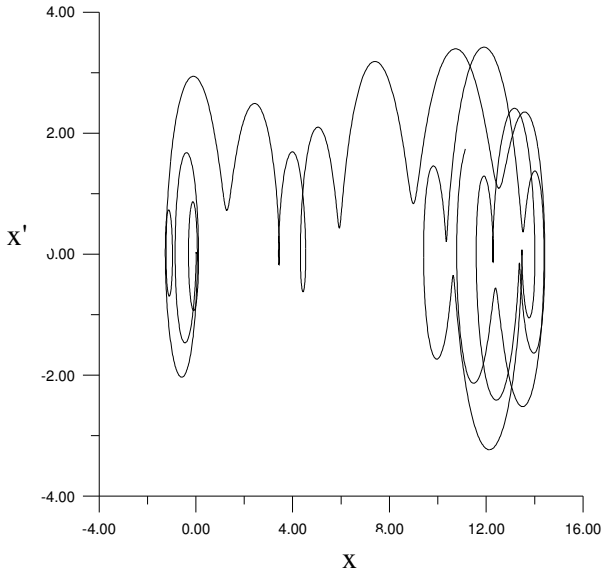

4.00
2.00
0.00
-2.00
-4.00
x'
-1.60
-1.20
-0.80
-0.40
0.00
0.40
x

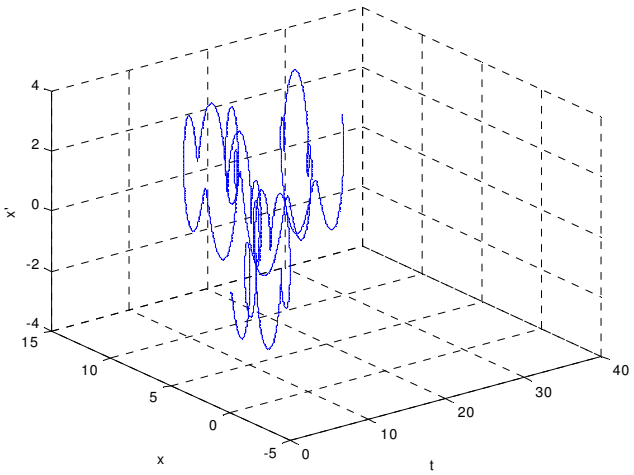

3
2
1
0
-1
-2
-3
x'
0.5
0
-0.5
-1
-1.5
x
0
2
4
6
8
t

T=5

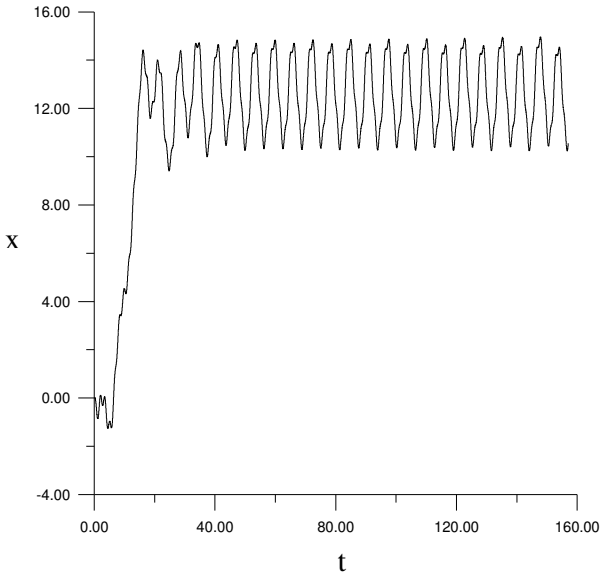

16.00
12.00
8.00
4.00
0.00
-4.00
x
0.00
10.00
20.00
30.00
40.00
t

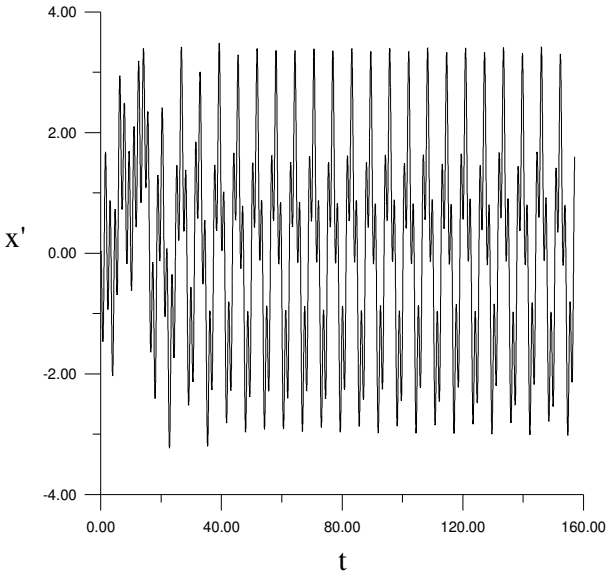

4.00
2.00
0.00
-2.00
-4.00
x'
0.00
10.00
20.00
30.00
40.00
t

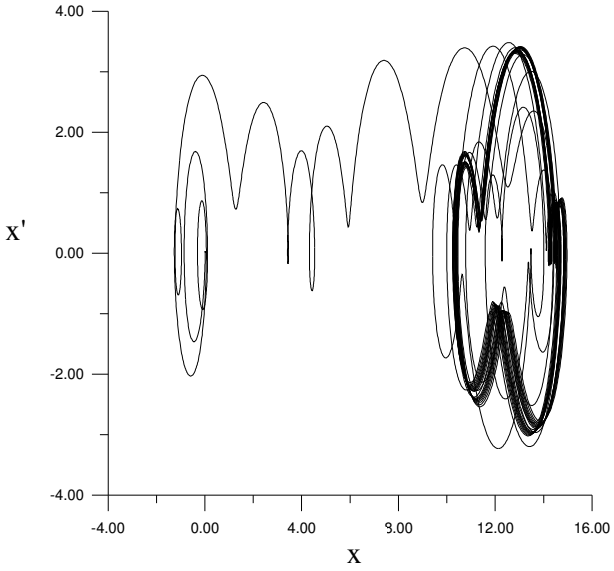

4.00
2.00
0.00
-2.00
-4.00
x'
-4.00
0.00
4.00
8.00
12.00
16.00
x

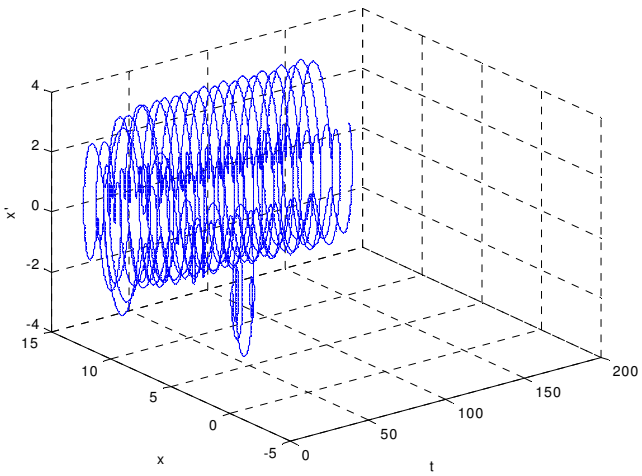

4
2
0
-2
-4
x'
15
10
5
0
-5
x
0
10
20
30
40
t

T=25

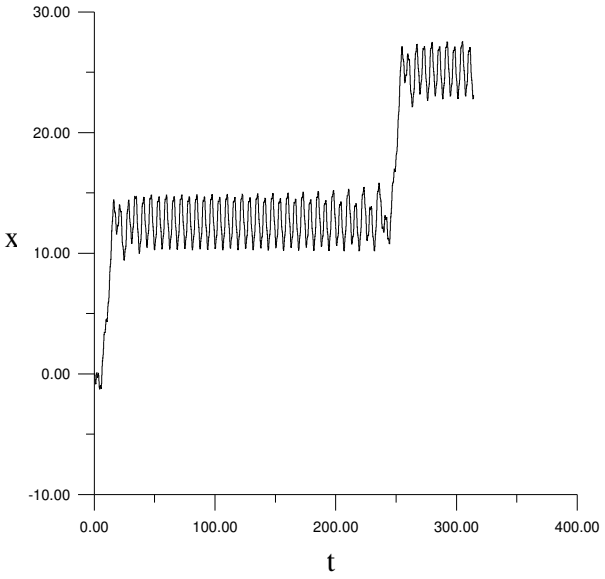

16.00
12.00
8.00
4.00
0.00
-4.00
x
0.00
40.00
80.00
120.00
160.00
t

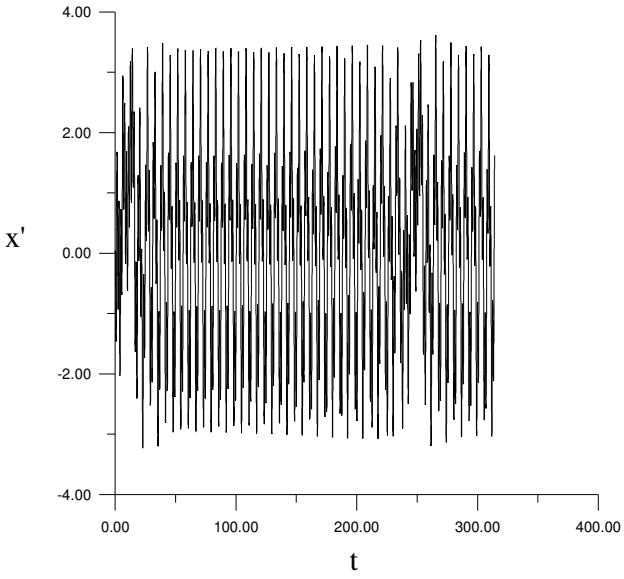

4.00
2.00
0.00
-2.00
-4.00
x'
0.00
40.00
80.00
120.00
160.00
t

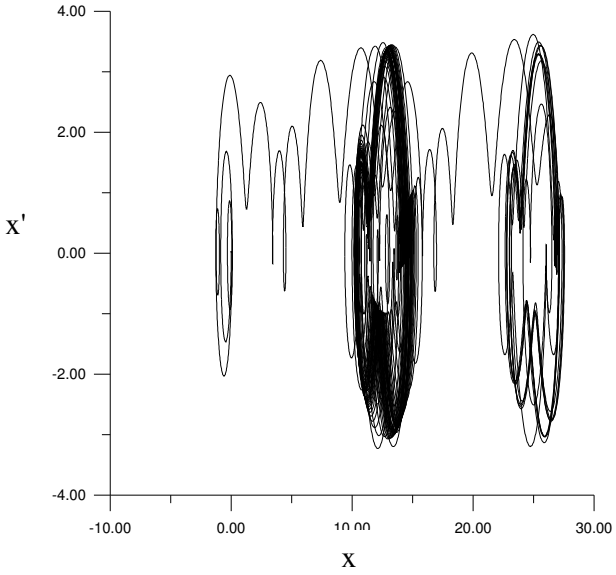

4.00
2.00
0.00
-2.00
-4.00
x'
-4.00
0.00
4.00
8.00
12.00
16.00
x

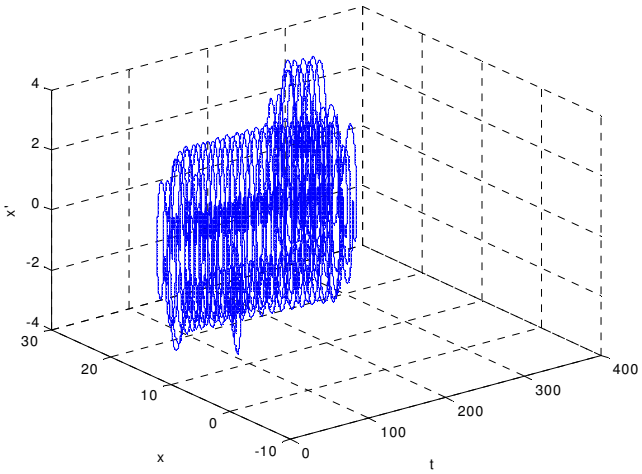

4
2
0
-2
-4
x'
15
10
5
0
-5
x
0
50
100
150
200
t

T=50


30.00
20.00
10.00
0.00
-10.00
x
0.00
100.00
200.00
300.00
400.00
t


4.00
2.00
0.00
-2.00
-4.00
x'
0.00
100.00
200.00
300.00
400.00
t


4.00
2.00
0.00
-2.00
-4.00
x'
-10.00
0.00
10.00
20.00
30.00
x


4
2
0
-2
-4
x'
30
20
10
0
-10
x
0
100
200
300
400
t

Exercício página 279

Adams Fourth-Order Predictor-Corrector

```
// Inicio --- include
#include <stdio.h>
#include <conio.h>
#include <float.h>
// Fim ------ include

// Inicio --- Constantes
#define NOME_DO_ARQUIVO "dados.dat"
#define MODO_DE_ABERTURA "wt"
#define fx(t,y) (y - (t)*(t)+ 1.0)
// Fim ------ Constantes

void main(){
        float   h = 0.2,        // Incremento
                a = 0.0,        // Inicio
                b = 2.0,        // Fim
                k1, k2, k3, k4; // variaveis auxiliares
        int     N = 10,         // Numero de iteracoes
                i=0,j=0;
        float y[10+1], t[10+1], Y, T, y1;
        FILE *F0;

        clrscr();
        t[i]=0;
        y[i]=0.5;
        F0=fopen(NOME_DO_ARQUIVO, MODO_DE_ABERTURA);
        fprintf(F0 ,"%f %f %f %f\n", t[i], y[i]);
        printf("t[i]    y[i]\n");
        printf("%f %f\n", t[i], y[i]);
        for (i=1; i <= 3; i++) {
                // para x
                k1 = h*fx(t[i-1], y[i-1]);
                k2 = h*fx((t[i-1]+h/2.0), (y[i-1]+k1/2.0));
                k3 = h*fx((t[i-1]+h/2.0), (y[i-1]+k2/2.0));
                k4 = h*fx((t[i-1]+h), (y[i-1]+k3));
                y[i] = y[i-1] + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;
                t[i] = a + (float)(i)*h;
                fprintf(F0 ,"%f %f %f %f\n", t[i], y[i]);
                printf("%f %f\n", t[i], y[i]);
        };
                for (i=4; i<=N; i++) {
                        t[i]=a+(float)(i)*h;
                        T=t[i];
                        Y=y[3] + h*( 55.0*fx(t[3],y[3]) - 59.0*fx(t[2],y[2]) +37.0*fx(t[1],y[1]) - 9.0*fx(t[0],y[0]))/24.0;
                        Y=y[3] + h*(9.0*fx(T,Y) + 19.0*fx(t[3],y[3]) - 5.0*fx(t[2],y[2]) + 1.0*fx(t[1],y[1]))/24.0;
                        fprintf(F0 ,"%f %f %f %f\n", t[i], Y);
                        printf("%f %f\n", t[i], Y);
                        for (j=0;j<=2; j++) {
                                        t[j]=t[j+1];
                                        y[j]=y[j+1];
                        };
                        t[3]=T;
                        y[3]=Y;
                };
                fclose(F0);
        getch();
}
```

Execução

t[i]	y[i]
0.000000	0.500000
0.200000	0.829293
0.400000	1.214076
0.600000	1.648922
0.800000	2.127206
1.000000	2.640829
1.200000	3.179903
1.400000	3.732350
1.600000	4.283421
1.800000	4.815096
2.000000	5.305370

Exercício 1 Lista 2

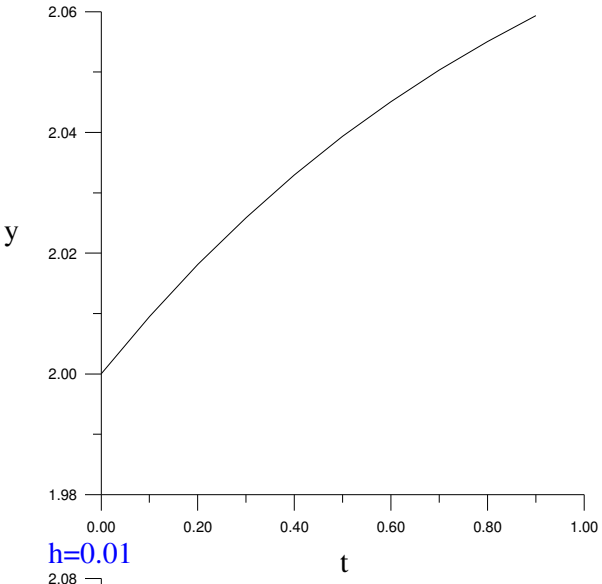
$y' = y - t^2 + 1$

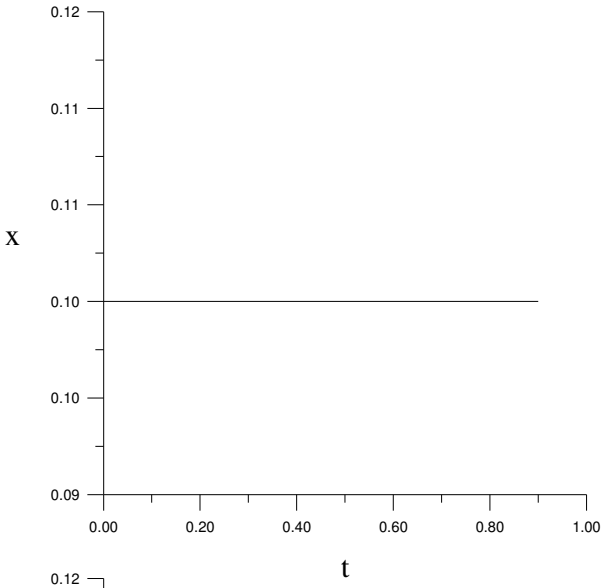
Range Kutta Ordem 4 em C++ 3.1

```
// Inicio --- include
#include <stdio.h>
#include <conio.h>
#include <float.h>
#include <math.h>
// Fim ------ include

// Inicio --- Constantes
#define NOME_DO_ARQUIVO "l2s1_2.dat"
#define MODO_DE_ABERTURA "wt"
#define fx1(t,y,x) (x - (y) + 2.0)
// Fim ------ Constantes

void main(){

        float
                h = 0.001,       // Incremento, h=0.1, h=0.01, h=0.001
                a = 0.0,         // Inicio
                b = 1.0,         // Fim
                t = a,           // Tempo
                x = 0.1,
                y = 2.0,
                k1, k2, k3, k4; // variaveis auxiliares
                 long int
                N = (long int)((b-a)/h), // Numero de iteracoes
                i;

        FILE *F0;

        clrscr();

        F0=fopen(NOME_DO_ARQUIVO, MODO_DE_ABERTURA);
        fprintf(F0 ,"%f %f %f \n", t, y, x);
        for (i=1; i <= N; i++) {

                // para x1
                k1 = h*fx1(t, y, x);
                k2 = h*fx1((t+h/2.0), (y+k1/2.0), x);
                k3 = h*fx1((t+h/2.0), (y+k2/2.0), x);
                k4 = h*fx1((t+h), (y+k3), x);
                y = y + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

                t = a + (float)(i)*h;

                fprintf(F0 ,"%f %f %f\n", t, y, x);
        }

        fclose(F0);
}
```

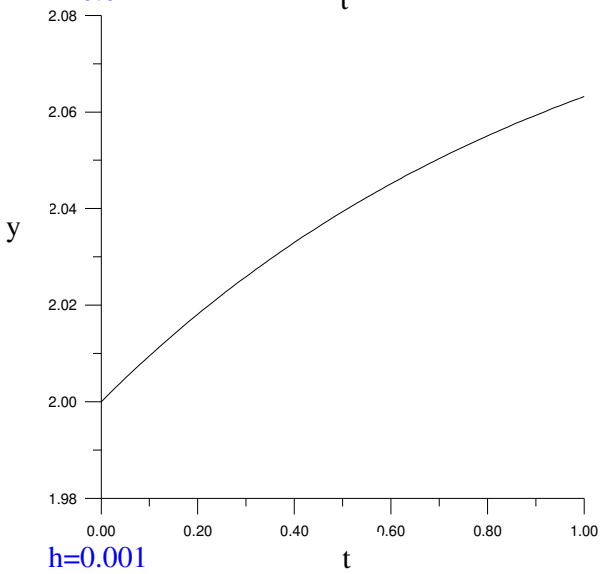
Gráficos

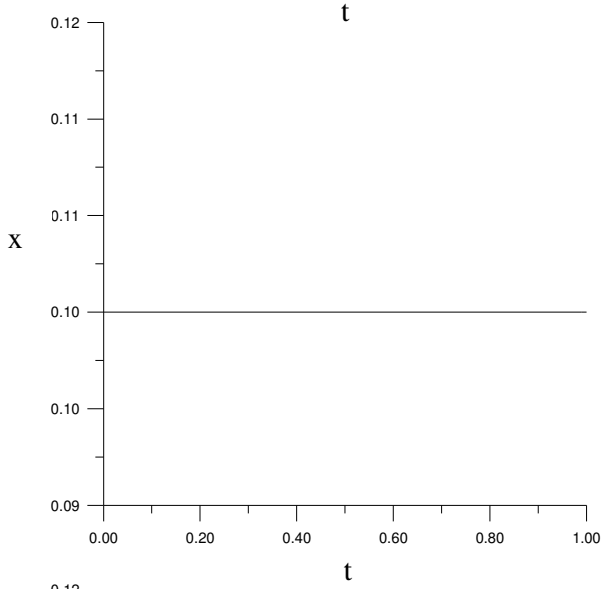
h=0.1



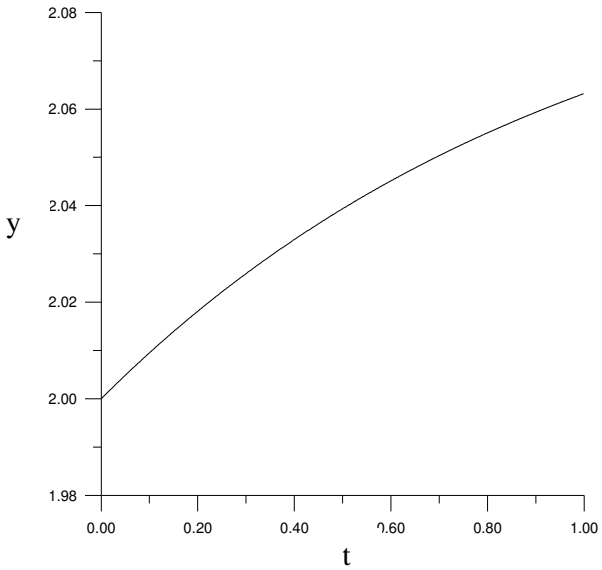


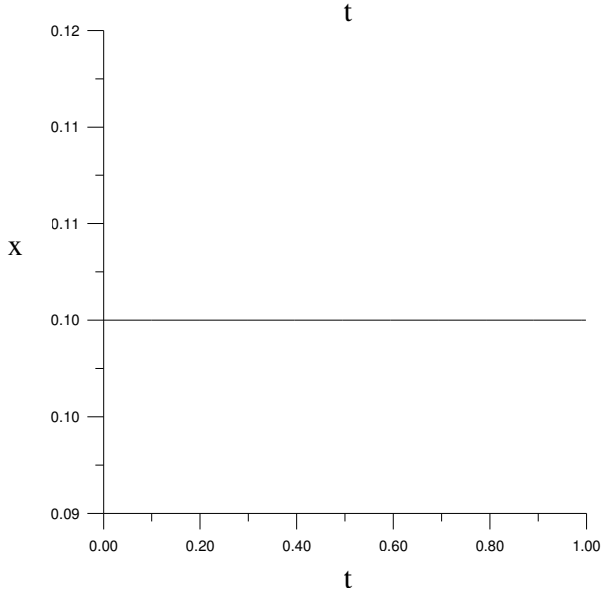
h=0.01





h=0.001





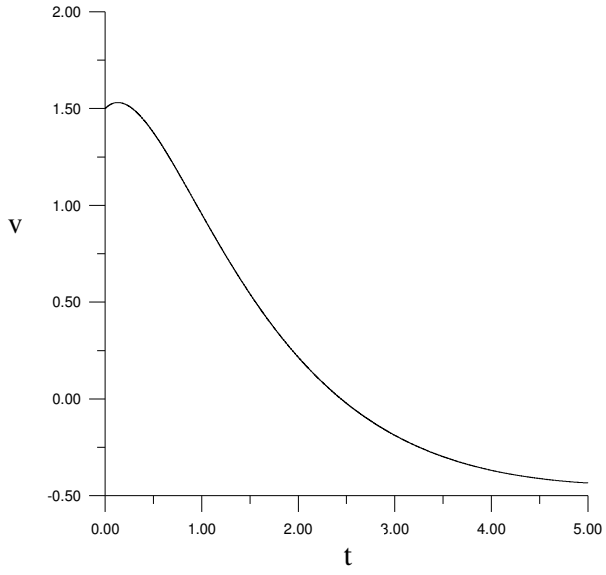
Exercício 2 Lista 2

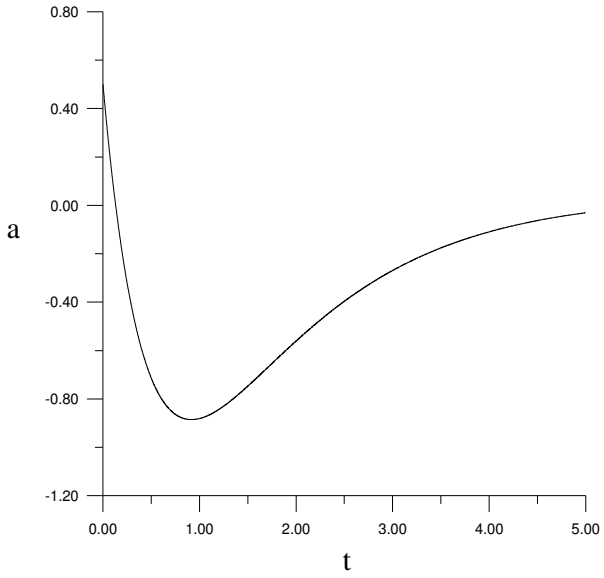
$$5\frac{d^2v(t)}{dt^2} + 4\frac{dv(t)}{dt} + \frac{1}{3}v(t) = -20e^{-2t}$$

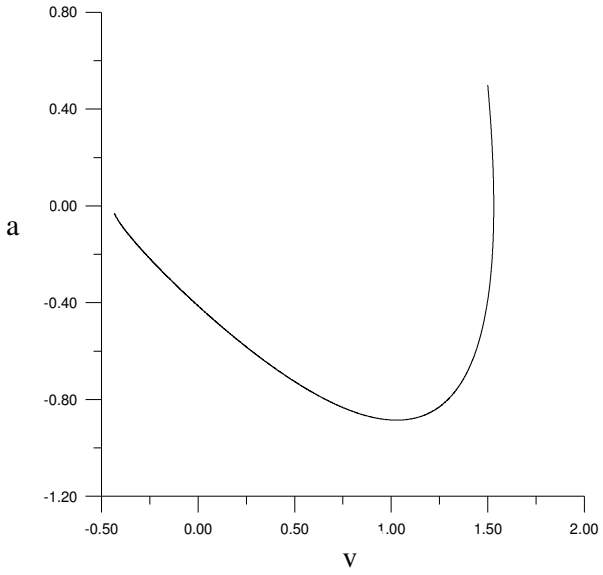
Range Kutta Ordem 4 em C++ 3.1

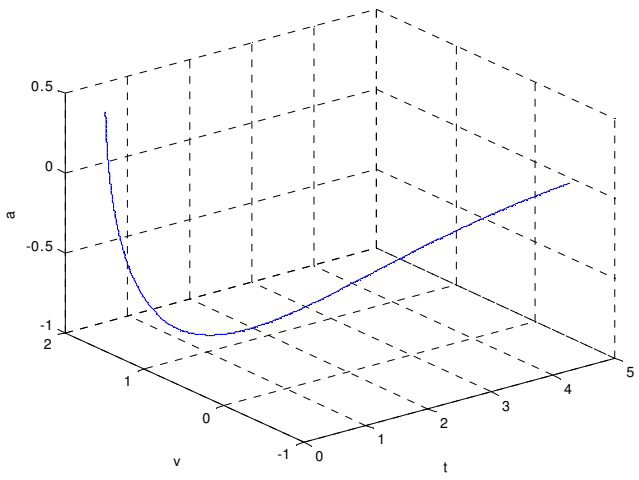
```
// Inicio --- include
#include <stdio.h>
#include <conio.h>
#include <float.h>
#include <math.h>
// Fim ------ include

// Inicio --- Constantes
#define NOME_DO_ARQUIVO "l2s2.dat"
#define MODO_DE_ABERTURA "wt"
#define fy1(t,v,a) (a)
#define fy2(t,a,v) (-(4.0/5.0)*(a) - (1.0/15.0)*(v) - 4.0*exp(-2.0*(t)))
// Fim ------ Constantes

void main(){

        float
                h = 0.001,      // Incremento
                in = 0.0,       // Inicio
                b = 5.0,        // Fim
                t = in,         // Tempo

                x = 0.0,   v = 1.5,  a = 0.5,  W=0, W1=0,
                k1, k2, k3, k4;                         // variaveis auxiliares
                long int  N = (long int)((b-in)/h),     // Numero de iteracoes
                i;

        FILE *F0;

        clrscr();

        F0=fopen(NOME_DO_ARQUIVO, MODO_DE_ABERTURA);
        fprintf(F0 ,"%f %f %f\n", t, v, a);
        for (i=1; i <= N; i++) {
                // para x1
                k1 = h*fy1(t, v, a);
                k2 = h*fy1((t+h/2.0), (v+k1/2.0), a);
                k3 = h*fy1((t+h/2.0), (v+k2/2.0), a);
                k4 = h*fy1((t+h), (v+k3), a);
                W = v + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

                k1 = h*fy2(t, a, v);
                k2 = h*fy2((t+h/2.0), (a+k1/2.0), v);
                k3 = h*fy2((t+h/2.0), (a+k2/2.0), v);
                k4 = h*fy2((t+h), (a+k3), v);
                W1 = a + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

                v=W; a=W1;
                t = in + (float)(i)*h;
                fprintf(F0 ,"%f %f %f\n", t, v, a);
        }
        fclose(F0);
}
```

Gráficos


2.00
1.50
1.00
0.50
0.00
-0.50
v
0.00
1.00
2.00
3.00
4.00
5.00
t


0.80
0.40
0.00
-0.40
-0.80
-1.20
a
0.00
1.00
2.00
3.00
4.00
5.00
t


0.80
0.40
0.00
-0.40
-0.80
-1.20
a
-0.50
0.00
0.50
1.00
1.50
2.00
v


0.5
0
-0.5
-1
a
2
1
0
-1
v
0
1
2
3
4
5
t

Exercício 3 Lista 2

$$y_1' = y_2$$

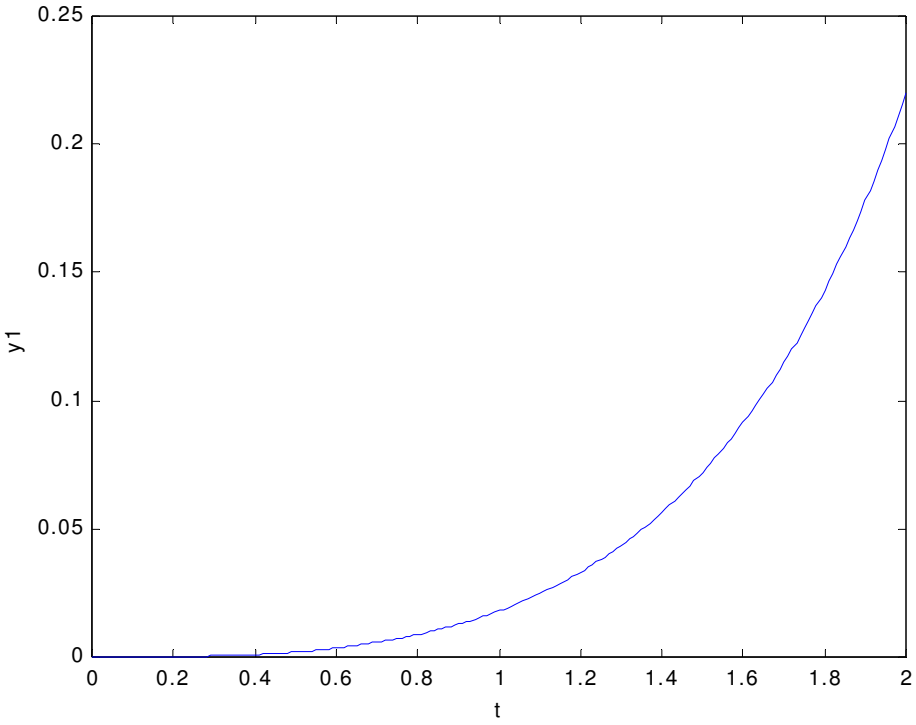
$$y_2' = y_3$$

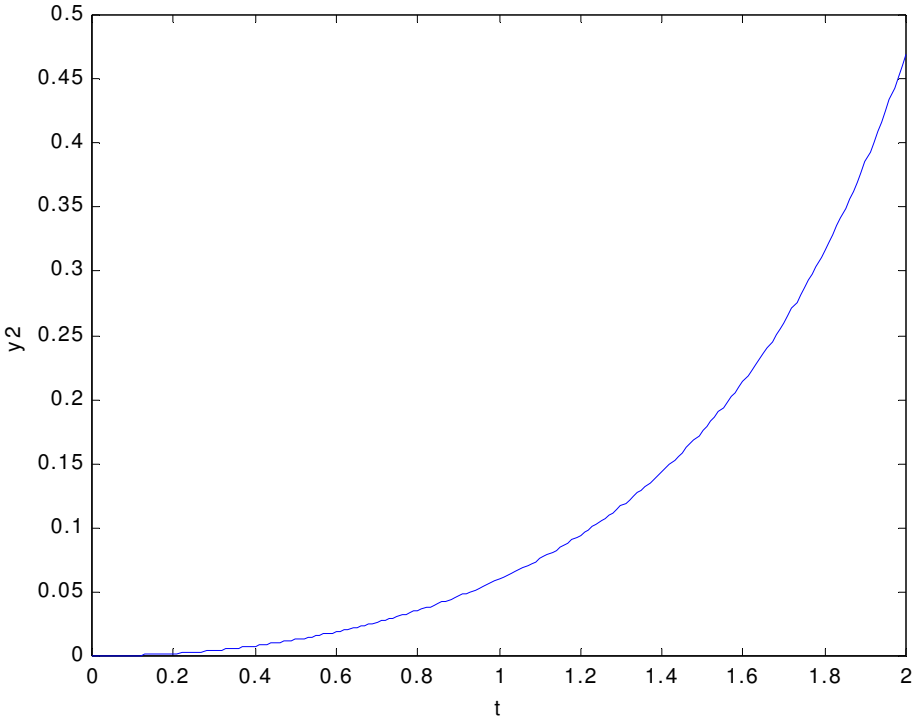
$$y_3' = x + y_1 + 2y_2 + y_3^2$$

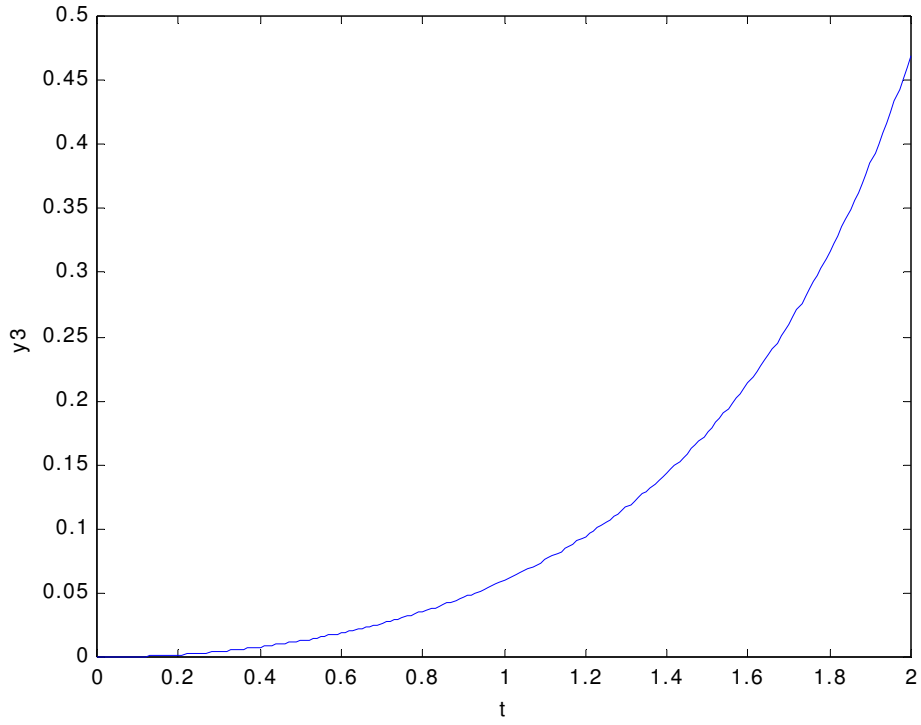
Range Kutta Ordem 4 em C++ 3.1

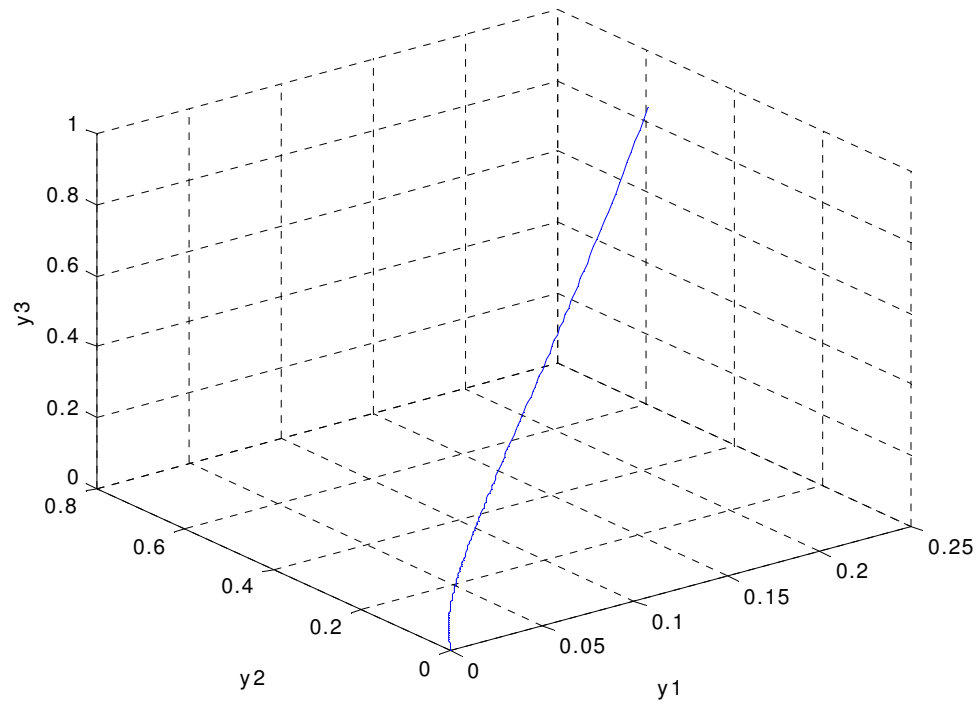
```
// Inicio --- include
#include <stdio.h>
#include <conio.h>
#include <float.h>
#include <math.h>
// Fim ------ include

// Inicio --- Constantes
#define NOME_DO_ARQUIVO "dados.dat"
#define MODO_DE_ABERTURA "wt"
#define fy1(t,y1,y2) (y2)
#define fy2(t,y2,y3) (y3)
#define fy3(t,y3,x,y1,y2) (x + y1 + 2.0*(y2) + (y3)*(y3))
// Fim ------ Constantes

void main(){
        float   h = 0.01,       // Incremento
                a = 0.0,        // Inicio
                b = 2.0,      // Fim
                t = a,          // Tempo
                x = 0.1,  y1 = 0.0, y2 = 0.0, y3 = 0.0, W=0,W1=0,W2=0,
                k1, k2, k3, k4; // variaveis auxiliares
                long int N = (long int)((b-a)/h), // Numero de iteracoes
                i;
        FILE *F0;
        clrscr();
        F0=fopen(NOME_DO_ARQUIVO, MODO_DE_ABERTURA);
        fprintf(F0 ,"%f %f %f %f %f\n", t, x, y1, y2, y3);
        for (i=1; i <= N; i++) {
                // para x1
                k1 = h*fy1(t, y1, y2);
                k2 = h*fy1((t+h/2.0), (y1+k1/2.0), y2);
                k3 = h*fy1((t+h/2.0), (y1+k2/2.0), y2);
                k4 = h*fy1((t+h), (y1+k3), y2);
                W = y1 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

                k1 = h*fy2(t, y2, y3);
                k2 = h*fy2((t+h/2.0), (y2+k1/2.0), y3);
                k3 = h*fy2((t+h/2.0), (y2+k2/2.0), y3);
                k4 = h*fy2((t+h), (y2+k3), y3);
                W1 = y2 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

                k1 = h*fy3(t, y3, x, y1, y2);
                k2 = h*fy3((t+h/2.0), (y3+k1/2.0), x, y1, y2);
                k3 = h*fy3((t+h/2.0), (y3+k2/2.0), x, y1, y2);
                k4 = h*fy3((t+h), (y3+k3), x, y1, y2);
                W2 = y3 + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0;

                y1=W; y2=W1; y3=W2;
                t = a + (float)(i)*h;
                fprintf(F0 ,"%f %f %f %f %f\n", t, x, y1, y2, y3);
        }
        fclose(F0);
}
```

Gráficos


0.25
0.2
0.15
0.1
0.05
0
y1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
t


0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
y2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
t


0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
y3
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
t


1
0.8
0.6
0.4
0.2
0
y3
0.8
0.6
0.4
0.2
0
y2
0
0.05
0.1
0.15
0.2
0.25
y1