

Breve relatório com os resultados obtidos durante o modulo-3

Aluno: Ruben Esteche Araújo

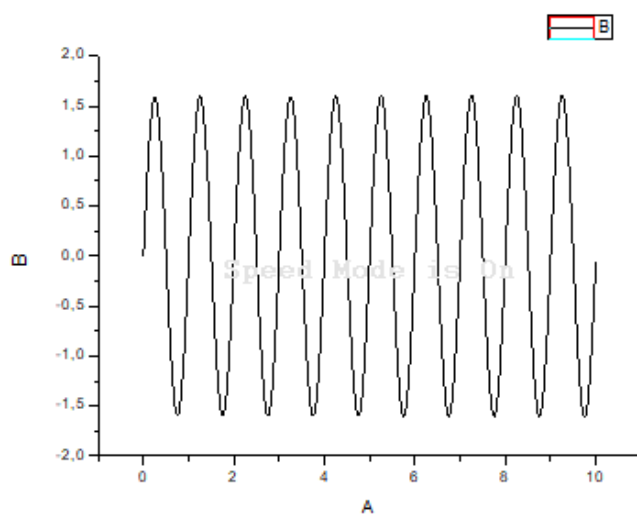
CPF: 109.429.904-98

Aqui, estudamos fenômenos de oscilação harmônica simples, amortecida e sobre influência de forças externas nos sistemas massa mola e pêndulo simples. Os algoritmos integracionais utilizados foram os de Euler, Euler-Cromer e Midpoint; Ambos são algoritmos ditos simpléticos (de no máximo ordem 2) , isso quer dizer que além da energia ser conservada, o volume ocupado por um conjunto de estados no espaço de fase permanece constante.

Os programas serão disponibilizados por extenso no final do relatório para consulta.

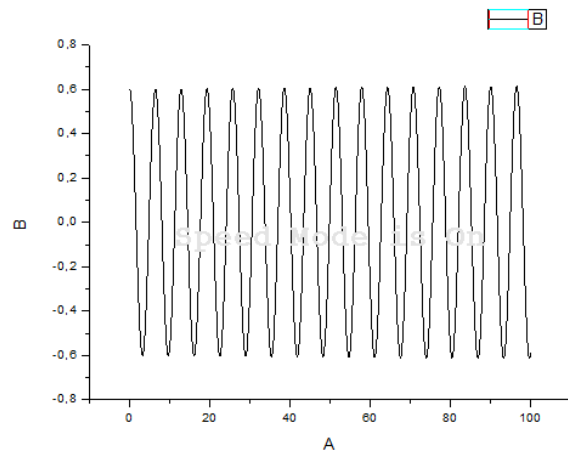
MIDPOINT

OHS:



Pendulo:

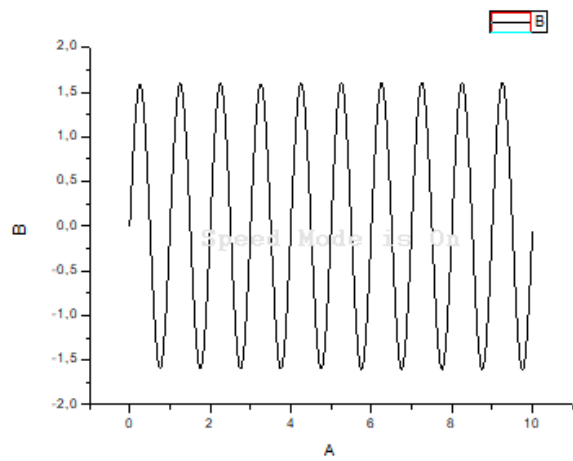
1

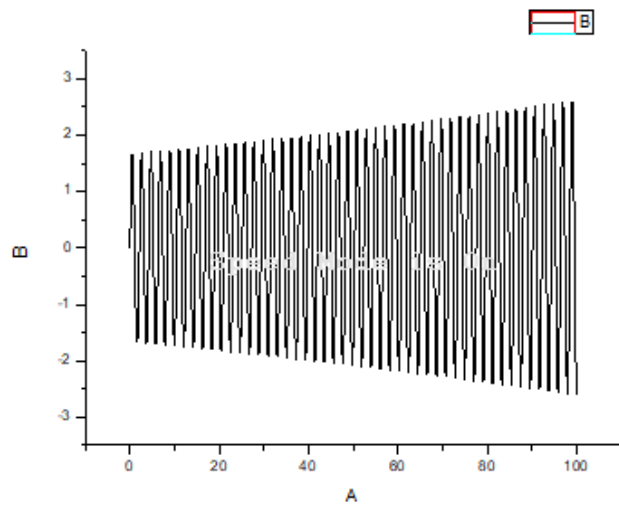


EULER

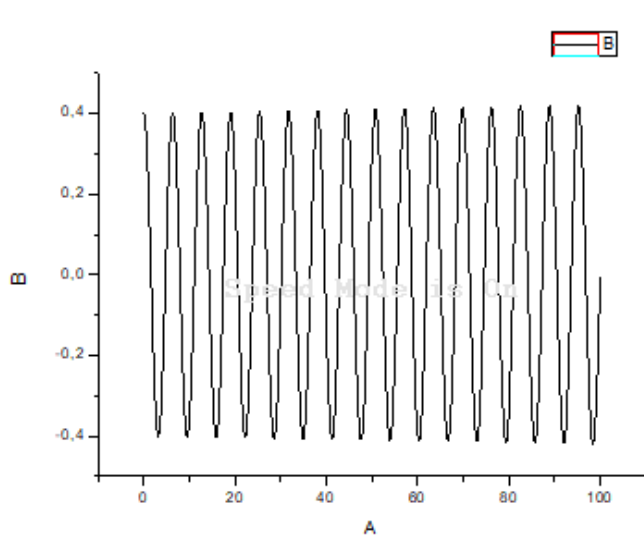
OHS:

1



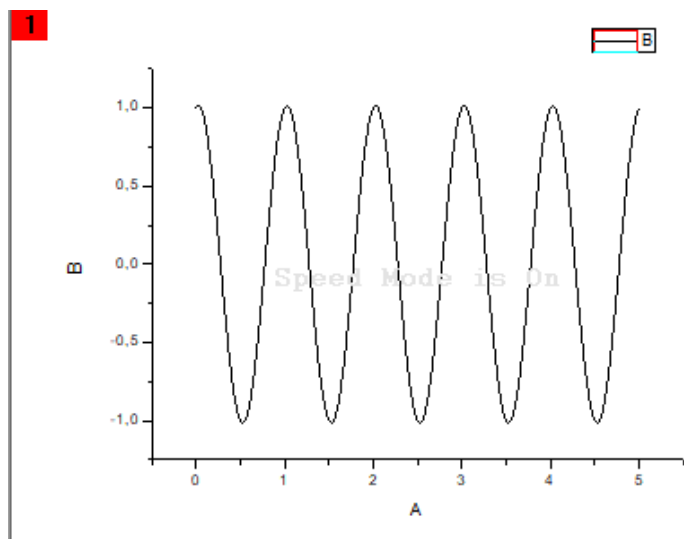


Pendolo:

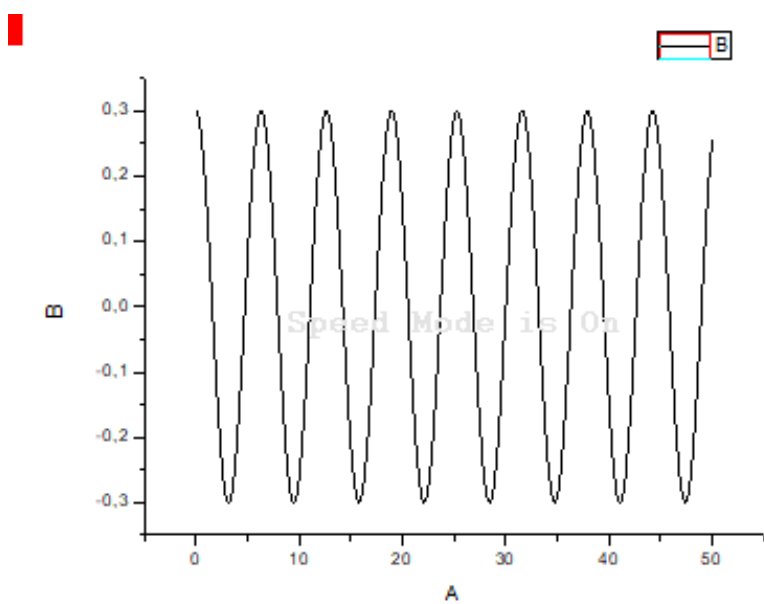


EULER CROMER

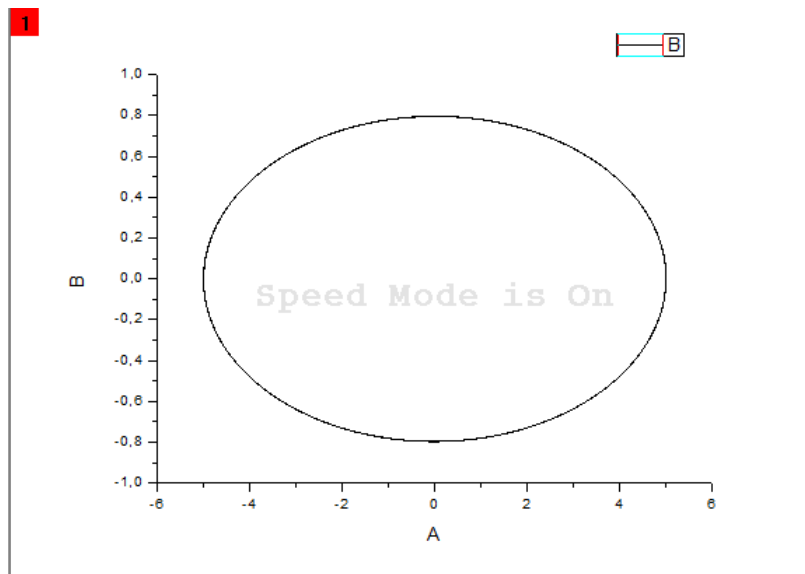
OHS:



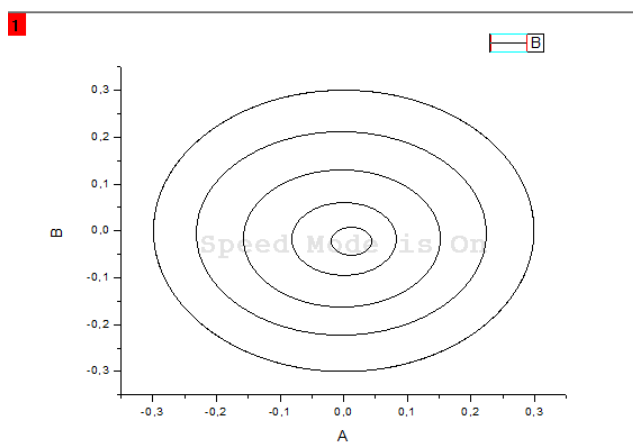
Pendulo:



Espaço de fase do oscilador harmônico no método Euler Cromer:



Espaço de fase do pêndulo simples no método Euler Cromer:
(para várias energias pequenas diferentes)



Espaço de fase do pêndulo simples AMORTECIDO no método Euler Cromer (força externa fraca):

1

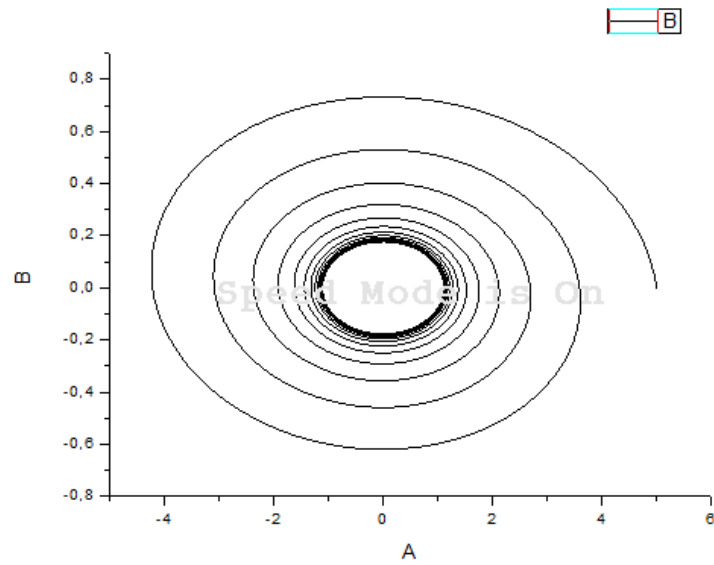
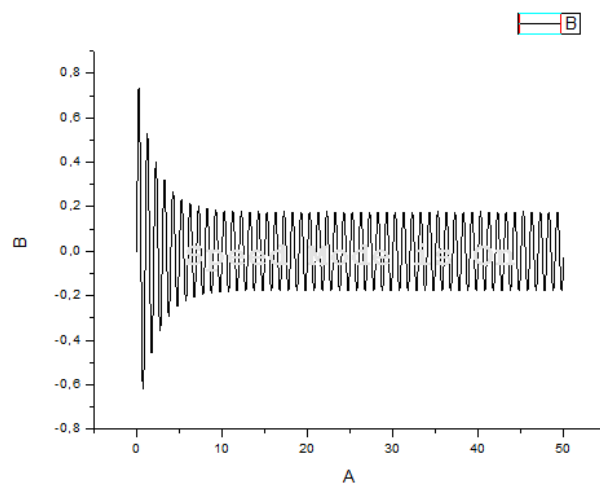


Gráfico de posição por tempo da simulação acima:

1



Espaço de fase do pêndulo simples AMORTECIDO no método Euler Cromer (força externa forte):

1

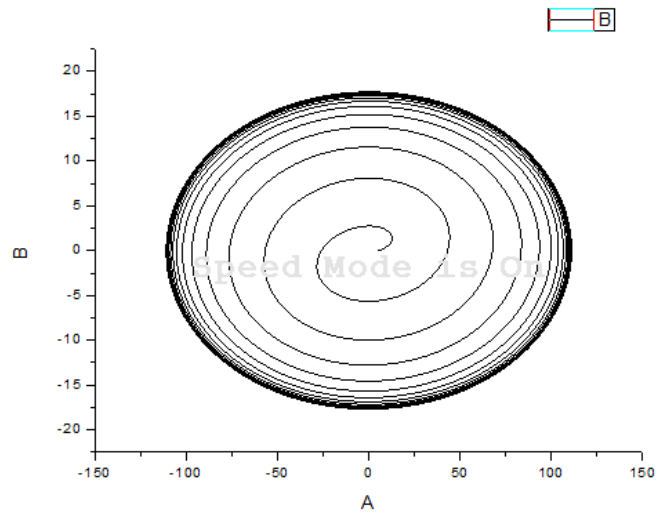
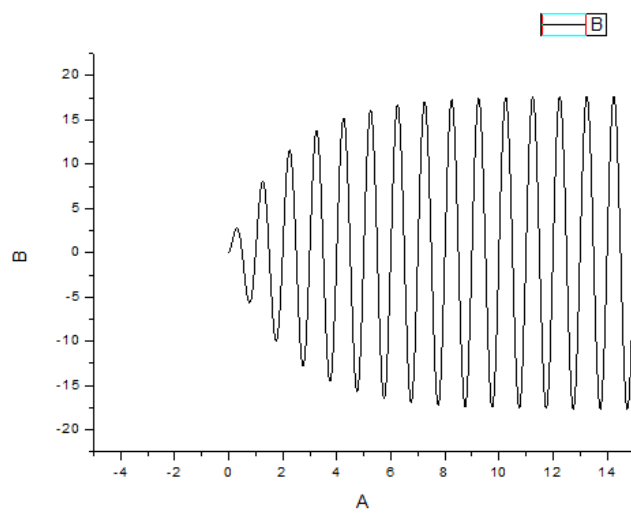


Gráfico de posição por tempo da simulação acima:

1



Espaço de fase do pêndulo simples AMORTECIDO no método Euler Cromer (força externa fraca):

1

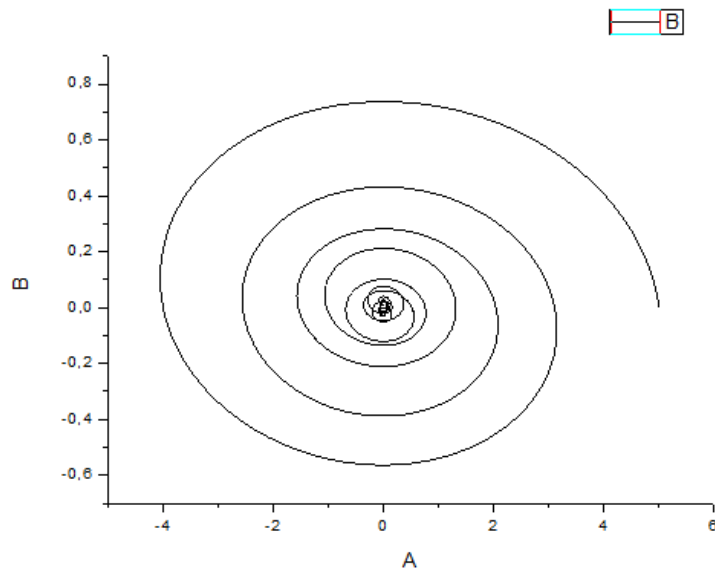
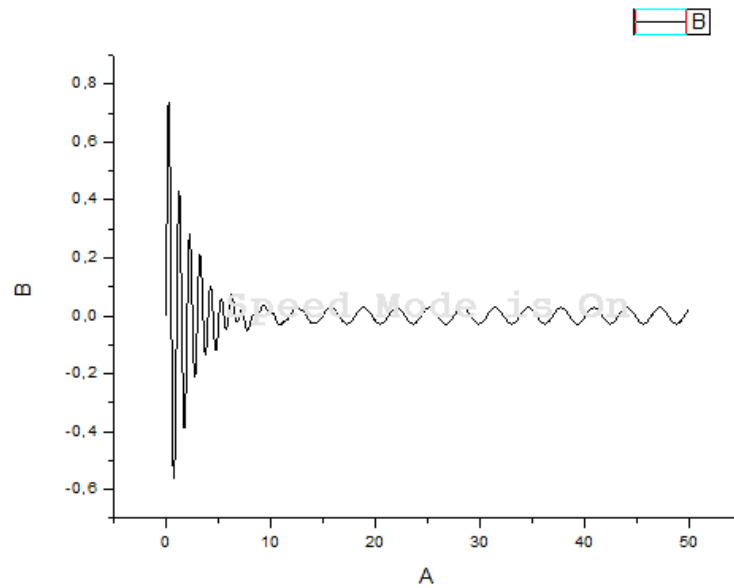


Gráfico de posição por tempo da simulação acima:

1



OBS.: Os gráficos do espaço de fase estão com os eixos invertidos, em vez de estarem plotados nos eixos “v VS x”, estão no formato “x VS v”. Por isso que estão girando no sentido anti-horário em vez do sentido horário, que seria o esperado.

PROGRAMAS

EULER

```
#include<stdlib.h>

#include<math.h>

#include<stdio.h>


int main(){

    //Definindo variáveis de movimento

    int n,i,p, g=10;

    double x, vx,v, w0,w, l=10,teta,t, dt=0.01;


    //obtendo valores do usuário


    //quantidade de interações

    n=100000;


    //Seleção de qual programa executar

    printf("Selecione qual problema voce deseja resolver:\n01.Pendulo simples(1)\n02.Oscilador(2)\n");
    scanf("%d",&p);

    if(p==1){

        //arquivo

        FILE *Rx_pendulo;

        Rx_pendulo = fopen("posição (pendulo).txt", "w+");


        //Variáveis do pêndulo

        printf("Digite velocidade inicial:\n");

        scanf("%lf",&w0);

        printf("Digite o angulo inicial:\n");

        scanf("%lf",&teta);


        //interações

        for(i=0;i<n;i++){

            fprintf(Rx_pendulo,"%lf %lf\n", t, teta );


            w = w0 -((g/l)*sin(teta))*dt;

            teta = teta + w0*dt;
```

```

w0=w;

t= t+dt;

}

//fechando o arquivo

fclose(Rx_pendulo);

}if(p==2){

//arquivo

FILE *Rx_oscilador;

Rx_oscilador = fopen("posição (oscilador).txt", "w+");

//Variáveis do oscilador

printf("Digite velocidade inicial:\n");

scanf("%lf",&vx);

printf("Digite a frequência angular inicial:\n");

scanf("%lf",&w0);

printf("Digite posição inicial:\n");

scanf("%lf",&x);

//interações

for(i=0;i<n;i++){

fprintf(Rx_oscilador,"%lf %lf\n", t,x);

v = vx - (w0*w0*x)*dt;

x= x + vx*dt;

vx = v;

t= t+dt;

}

//fechando o arquivo

fclose(Rx_oscilador);

}

printf("seu arquivo foi criado com sucesso");

return 0;

```

```
}
```

EULER_CROMER

```
#include<stdlib.h>
```

```
#include<math.h>
```

```
#include<stdio.h>
```

```
int main(){
```

```
    //Definindo variáveis de movimento
```

```
    int n,i,p, g=10,j;
```

```
    double x, vx, w0, l=10,teta,t, dt=0.001, gama=0.9;
```

```
    //obtendo valores do usuário
```

```
    //quantidade de interações
```

```
    n=50000;
```

```
    //Seleção de qual programa executar
```

```
    printf("Selecione qual problema voce deseja resolver:\n01.Pendulo simples(1)\n02.Oscilador(2)\n");
```

```
    scanf("%d",&p);
```

```
    if(p==1){
```

```
        //arquivo
```

```
        FILE *Rx_pendulo;
```

```
        Rx_pendulo = fopen("posição (pendulo).txt","w+");
```

```
        //Variáveis do pêndulo
```

```
        printf("Digite velocidade inicial:\n");
```

```
        scanf("%lf",&w0);
```

```
        printf("Digite o angulo inicial:\n");
```

```
        scanf("%lf",&teta);
```

```
        //interações
```

```
        for(i=0;i<n;i++){
```

```
            fprintf(Rx_pendulo,"%lf %lf\n",w0, teta);
```

```
            w0 = w0 -((g/l)*sin(teta))*dt;
```

```

teta = teta + w0*dt;

t= t+dt;

}

//fechando o arquivo

fclose(Rx_pendulo);

}if(p==2){

//arquivo

FILE *Rx_oscilador;

Rx_oscilador = fopen("posição (oscilador).txt", "w+");

//Variáveis do oscilador

printf("Digite velocidade inicial:\n");

scanf("%lf",&vx);

printf("Digite a frequência angular inicial:\n");

scanf("%lf",&w0);

printf("Digite posição inicial:\n");

scanf("%lf",&x);

//interações

for(i=0;i<n;i++){

fprintf(Rx_oscilador,"%lf %lf\n", vx, x);

vx = vx - (w0*w0*x)*dt -(vx*gama)*dt + (cos(1*t))*dt; //força resistiva com gama e força motriz externa em cosseno

x= x + vx*dt;

t= t+dt;

}

//fechando o arquivo

fclose(Rx_oscilador);

}

```

MIDPOINT

```
#include<stdlib.h>

#include<math.h>

#include<stdio.h>

int main(){

    //Definindo variáveis de movimento

    int n,i,p, g=10;

    double x, xn, vx, v, w0, w, l=10, teta, tetaf, t, dt=0.001;


    //obtendo valores do usuário


    //quantidade de interações

    n=100000;


    //Seleção de qual programa executar

    printf("Selecione qual problema voce deseja resolver:\n01.Pendulo simples(1)\n02.Oscilador(2)\n");
    scanf("%d",&p);

    if(p==1){

        //arquivo

        FILE *Rx_pendolo;

        Rx_pendolo = fopen("posição (pendulo).txt", "w+");


        //Variáveis do pêndulo

        printf("Digite velocidade inicial:\n");

        scanf("%lf",&w0);

        printf("Digite o angulo inicial:\n");

        scanf("%lf",&teta);


        //interações

        for(i=0;i<n;i++){

            fprintf(Rx_pendolo,"%lf %lf\n",t, teta);


            w = w0 -((g/l)*sin(teta))*dt;

            tetaf = teta + 0.5*(w+w0)*dt;
```

```

t= t+dt;

w0=w;

teta=tetaf;

}

//fechando o arquivo

fclose(Rx_pendulo);

}if(p==2){

//arquivo

FILE *Rx_oscilador;

Rx_oscilador = fopen("posição (oscilador).txt", "w+");

//Variáveis do oscilador

printf("Digite velocidade inicial:\n");

scanf("%lf",&vx);

printf("Digite a frequência angular inicial:\n");

scanf("%lf",&w0);

printf("Digite posição inicial:\n");

scanf("%lf",&x);

//interações

for(i=0;i<n;i++){

fprintf(Rx_oscilador,"%lf %lf\n", t, x);

v = vx - (w0*w0*x)*dt;

x= xn + 0.5*(vx+v)*dt;

t= t+dt;

vx=v;

xn=x;

}

//fechando o arquivo

fclose(Rx_oscilador);

}

```

```
printf("seu arquivo foi criado com sucesso");
```

```
return 0;
```

```
}
```