



Projeto 4: Equações Diferenciais

Anderson Araujo de Oliveira 11371311

2020

1 Parte A

1.1 Forma analítica

Vamos encontrar a forma analítica da velocidade e do deslocamento, chamando $a = \frac{dv}{dt}$.

$$\frac{dv}{dt} = a \quad (1)$$

Resolvendo isso (1) chegamos na equação de baixo (3).

$$\int_{v_0}^v dv = \int_0^t a dt \quad (2)$$

$$v = v_0 + at \quad (3)$$

Qbrindo $\frac{dy}{dt}$ e resolvendo

$$\frac{dy}{dt} = v_0 + at \quad (4)$$

$$\int_{y_0}^y dy = \int_0^t (v_0 + at) dt \quad (5)$$

$$y = y_0 + v_0 t + a \frac{t^2}{2} \quad (6)$$

1.2 Resultados

Utilizando a função (6) conseguimos saber qual dos ϵ tem uma menor incerteza de seus dados, usando o tempo $t=0,02886s$.

$$x = 100 + 10 \cdot 0,02886 - 10 \frac{0,02886^2}{2} = 100,284 \quad (7)$$

Vemos que o gráfico que está mais perto desse valor é o $\epsilon = 0.0001$

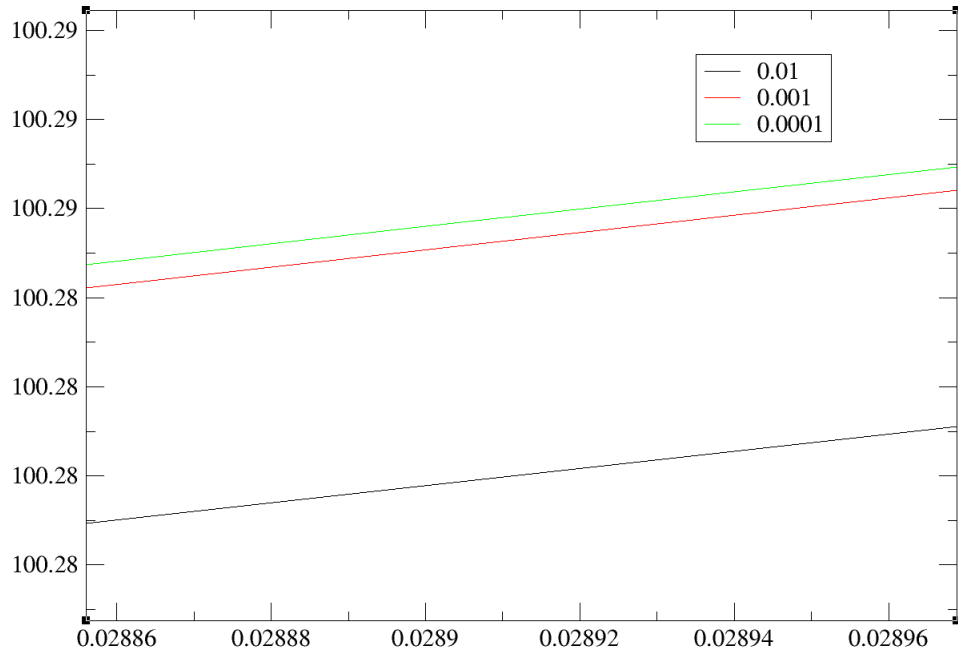


Figura 1: y versus tempo, para ver confiabilidade

Nas figuras abaixo foram utilizados $\epsilon = 0.0001$ por ter uma confiabilidade maior no dados. Usando as formulas que achamos anteriormente, conseguimos ver que os dados que obtemos condiz com realidade, se substituimos os valores e igualando a $y = 0m$ conseguimos saber qual o tempo a partícula bate no chão. Para o caso $V_0 = 10 \frac{m}{s}$ a partícula bate chega em $t = 5.58258s$ e no $V_0 = 0 \frac{m}{s}$ a partícula chega em $t = 4.47214s$, vemos que os resultados analítico condiz com o do programa.

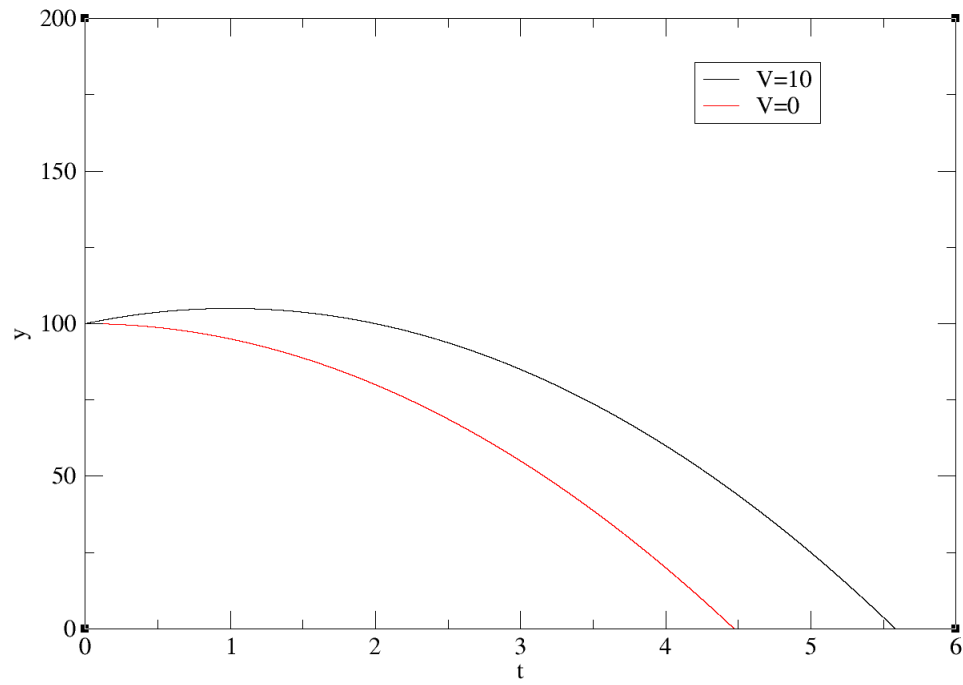


Figura 2: y versus tempo

Usando a mesma ideia que citamos anteriormente vamos colocar os resultados que obtivemos, mas dessa vez usaremos o tempo que obtivemos anteriormente para ver se bate com a velocidade final $t = 5.58258, vf = -4.82575 \frac{m}{s}$ e $t = 4.47214, vf = -4.82575 \frac{m}{s}$, assim vemos que a velocidade é mesma quando o objeto toca no chão.

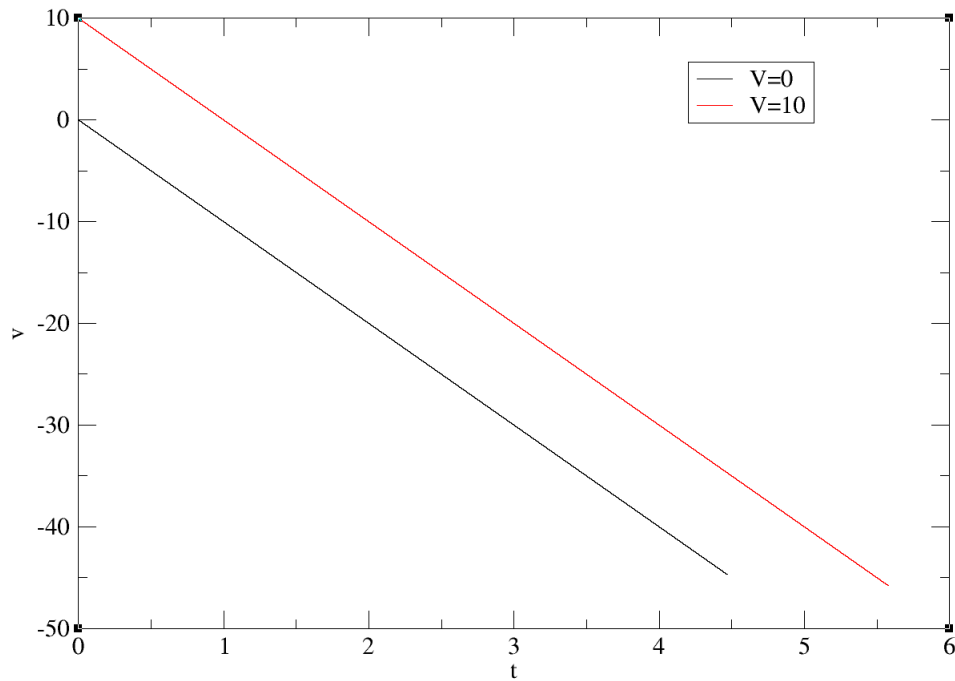


Figura 3: Velocidade versus tempo

1.3 Código A

```

1 program Qa
2 implicit real*8(a-h,o-z)
3 real*8,dimension(2)::v0
4 real*8,dimension(3)::e0
5 character::nome(12)*22
6 v0=(/10d0,0d0/)
7 e0=(/0.01d0,0.001d0,0.0001d0/)
8 !nome dos arquivos os valores impares s o y(t) e os arquivos pares v o v
   (t)
9 data nome/"saida1-a-11371311.dat","saida2-a-11371311.dat","saida3-a
   -11371311.dat","saida4-a-11371311.dat", &
10 &"saida5-a-11371311.dat", "saida6-a-11371311.dat","saida7-a-11371311.dat",
   "saida8-a-11371311.dat",&
11 &"saida9-a-11371311.dat","saida10-a-11371311.dat","saida11-a-11371311.dat"
   ,"saida12-a-11371311.dat"/
12 !condicoes iniciais
13 y=100
14 t=e0(i)
15 i=1
16 j=1
17 k=11
18 !primeiro e0
19 v=v0(j)-10*e0(i)/2
20 y=y+v*e0(i)
21 open(k,file=nome(k-10))
22 open(k+1,file=nome(k-9))

```

```

23 write(k,*)t,y
24 write(k+1,*)t,v
25 do while(y>=0 .and. j<=2)
26     v=v-10*e0(i)
27     y=y+v*e0(i)
28     t=t+e0(i)
29     if(y<=0 .and. i<3)then!caso y chegue a 0 e0 permuta
30         y=100
31         i=i+1
32         k=k+2
33         t=0
34         !primeiro e0
35         v=v0(j)-10*e0(i)/2
36         open(k,file=nome(k-10))!foi colocado k-10 por causa que quando
colocamos certos numeros para k ocorre um erro
37         open(k+1,file=nome(k-9))
38     elseif(y<=0 .and. i==3 .and. j<2)then!caso y chegue a 0 e0 permuta e j
=6 a velocidade troca
39         y=100
40         t=0
41         j=j+1
42         i=1
43         k=k+2
44         !primeiro e0
45         v=v0(j)-10*e0(i)/2
46         open(k,file=nome(k-10))!foi colocado k-10 por causa que quando
colocamos certos numeros para k ocorre um erro
47         open(k+1,file=nome(k-9))
48     endif
49     write(k,*)t,y
50     write(k+1,*)t,v
51 enddo
52 end program

```

2 Parte B

2.1 Forma analítica

Nesse projeto B iremos ver como é o deslocamento de uma partícula em uma região de uma dimensão, sofrendo aceleração da gravidade g com uma resistência do ar γ .

$$-\gamma v - g = a \quad (8)$$

Para saber a velocidade e altura da partícula vamos utilizar a mesma técnica que usamos na parte A para acharmos a forma analítica, $a = \frac{dv}{dt}$

$$\frac{dv}{dt} = -\gamma v - g \quad (9)$$

isolando o termo de v em um lado dos lados da equação

$$\int_{v_0}^v \frac{dv}{-\gamma v - g} = \int_0^t dt \quad (10)$$

$$\ln\left(\frac{-\gamma v - g}{-\gamma v_0 - g}\right) = -\gamma t \quad (11)$$

$$\frac{-\gamma v - g}{-\gamma v_0 - g} = e^{-\gamma t} \quad (12)$$

isolando v

$$v = \frac{(\gamma v_0 + g)e^{-\gamma t} - g}{\gamma} \quad (13)$$

A velocidade terminal é dada $t \rightarrow \infty$, se colocamos a função(13) tender ∞ chegamos na velocidade terminal

$$\lim_{t \rightarrow \infty} \frac{(\gamma v_0 + g)e^{-\gamma t} - g}{\gamma} = \frac{-g}{\gamma} \quad (14)$$

Para descobrimos o deslocamento chamamos $v = \frac{dx}{dt}$

$$\frac{dx}{dt} = \frac{(\gamma v_0 + g)e^{-\gamma t} - g}{\gamma} \quad (15)$$

$$\int_{y_0}^y dy = \int_0^t \frac{(\gamma v_0 + g)e^{-\gamma t} - g}{\gamma} dt \quad (16)$$

$$y = \frac{(\gamma v_0 + g)(1 - e^{-\gamma t}) - \gamma g t}{\gamma^2} + y_0 \quad (17)$$

2.2 Resultados

Em nenhum casos abaixo chegou na velocidade terminal, por causa que essa velocidade só encontrada quando o $t \rightarrow \infty$ como vimos na forma analítica $\frac{-g}{\gamma}$ se substituimos os valores que utilizamos em baixo chegamos em suas velocidades terminais para $\gamma = 0.1$ e $\gamma = 0.01$ são respectivamente $v_t = 100 \frac{m}{s}$ e $v_t = 1000 \frac{m}{s}$, para casos extremos conseguimos chegar perto da velocidade terminal se começamos a altura maior que $y = 110m$, $v = 99.98 \frac{m}{s}$, bem próximo do valor que calculamos para a velocidade terminal.

Como vemos na figura 4 a partícula demora mais para chegar no solo esse motivo se deve por causa da resistência do ar que faz a partícula sofrer uma menor aceleração, no gráfico mostra que ficou mais tempo que parte A vamos ver, analiticamente $v = 10 \frac{m}{s}$, $t = 5.90615s$ e $v = 0 \frac{m}{s}$, $t = 5.83183s$.

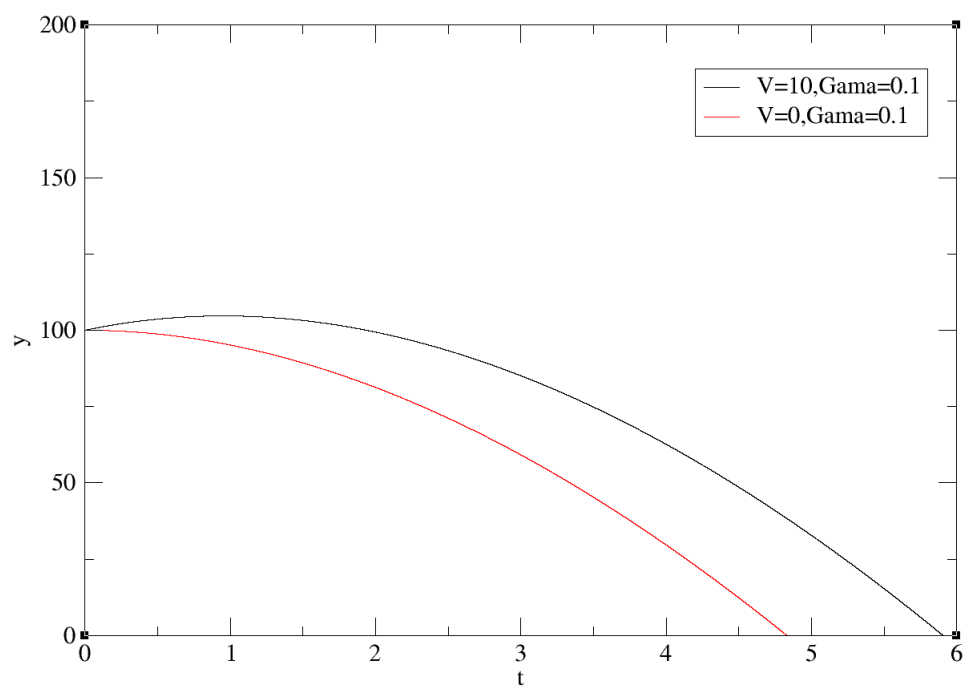


Figura 4: y versus tempo

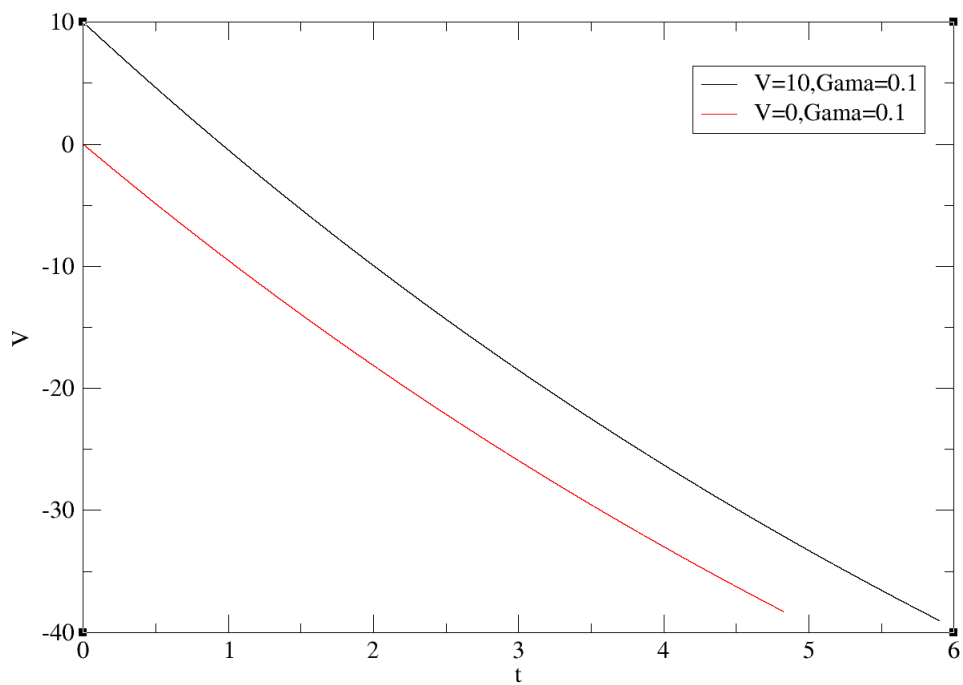


Figura 5: Velocidade versus tempo

Vamos ver os resultados batem também com resistência $\gamma = 0.001, v = 0 \frac{m}{s}, t = 4,505s$ e $v = 10 \frac{m}{s}, 5.612114s$

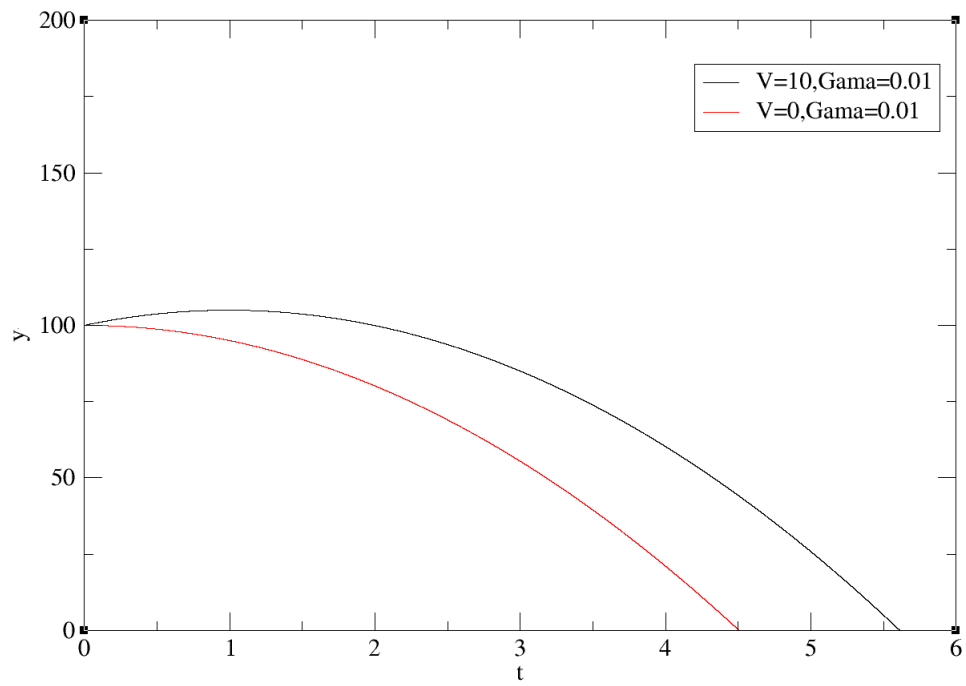


Figura 6: y versus tempo

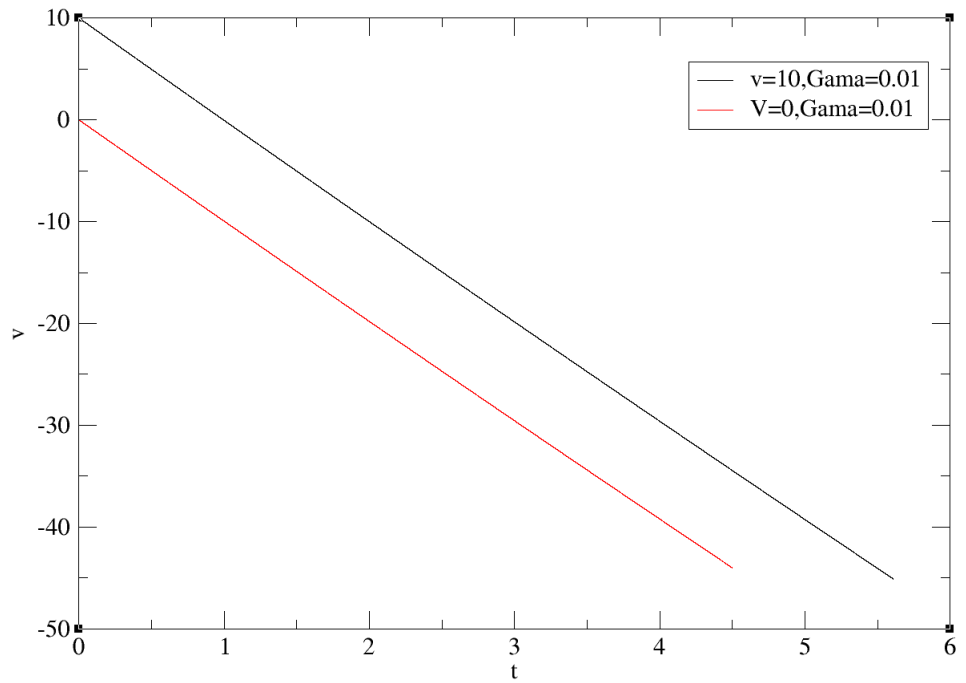


Figura 7: Velocidade versus tempo

2.3 Código B

```

1 program Qb
2   implicit real*8(a-h,o-z)

```

```

3  real*8,dimension(2)::v0
4  real*8,dimension(2)::gama
5  character::nome(8)*22
6  !nome dos arquivos os valores impares s o y e os arquivos pares v o
v
7  data nome/"saida1-b-11371311.dat","saida2-b-11371311.dat","saida3-b
-11371311.dat","saida4-b-11371311.dat", &
8  &"saida5-b-11371311.dat" ,"saida6-b-11371311.dat","saida7-b-11371311.
dat","saida8-b-11371311.dat"/
9
10 !condicoes iniciais no vetor
11 v0=(/10d0,0d0/)
12 gama=(/0.1d0,0.01d0/)
13
14 !condicoes iniciais
15 e0=0.0001
16 y=100
17 j=1
18 l=1
19 t=e0
20 k=11
21
22 !primeiro e0
23 a=-gama(1)*v0(j)-10
24 v=v0(j)+e0*a/2.0d0
25 y=y+e0*v
26 open(k,file=nome(k-10))
27 open(k+1,file=nome(k-9))
28 write(k,*)t,y
29 write(k+1,*)t,v
30
31 !loop para verifica
32 do while(y>=0 .and. j<=2)
33     a=-gama(1)*v-10
34     v=v+e0*a
35     y=y+v*e0
36     t=t+e0
37     if(y<=0 .and. j<2)then!caso y chegue a 0, v0 permuta
38         y=100!nesse caso ele verifica todos e0 antes de mudar v0
39         j=j+1
40         k=k+2
41         t=e0
42         !primeiro e0
43         a=-gama(1)*v0(j)-10
44         v=v0(j)+a*e0/2
45         open(k,file=nome(k-10))!foi colocado k-10 por causa que
quando colocamos certos numeros para k ocorre um erro
46         open(k+1,file=nome(k-9))
47         elseif(y<=0 .and. j==2 .and. l<2)then!caso y chegue a 0, gama
permuta e j=6 a velocidade troca
48         y=100!nesse caso ele verifica todos v0 antes de mudar gama
49         t=e0
50         l=l+1
51         j=1

```

```

52         k=k+2
53         !primeiro e0
54         a=-gama(1)*v0(j)-10
55         v=v0(j)+a*e0/2
56         open(k,file=nome(k-10))!foi colocado k-10 por causa que
quando colocamos certos numeros para k ocorre um erro
57         open(k+1,file=nome(k-9))
58     endif
59     write(k,*)t,y
60     write(k+1,*)t,v
61 enddo
62 end program

```

3 Parte C

3.1 Forma analítica

Vamos achar a função onde podemos obter y em função de x , como nessa parte não outra força agindo além da gravidade, portanto a velocidade na direção x será constante.

$$v_x = v_0 \cos(\theta) \quad (18)$$

$$x = v_0 \cos(\theta) t \quad (19)$$

Isolando t e substituindo em (21) obtemos y em função de x .

$$\frac{x}{v_0 \cos(\theta)} = t \quad (20)$$

$$y = y_0 + v_0 \sin(\theta) t + \frac{at^2}{2} \quad (21)$$

$$y = y_0 + tg(\theta)x + \frac{ax^2}{2v_0^2 \cos^2(\theta)} \quad (22)$$

3.2 Resultados

Vemos na figura 8 que representa o xXt vemos que de $\alpha=45^\circ$ e de $\alpha=-45^\circ$ as linhas ficam sobrepostas uma a outra por causa que a velocidade dos dois é a mesma no eixo x , mas os dois percorrem distâncias diferentes na direção de y por causa que a partícula de $\alpha=45^\circ$ fica mais tempo no ar do que a de $\alpha=-45^\circ$.

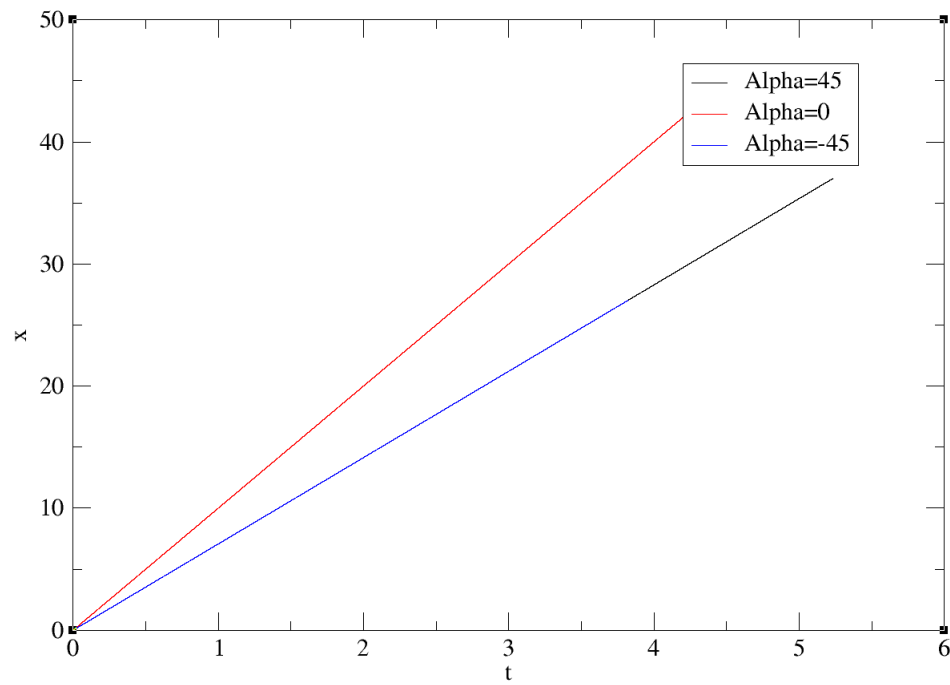


Figura 8: x versus tempo

Vemos nessa parte a altura e o tempo que partícula ficou no ar, algo interessante ocorre nesse parte que ajuda entender o próximo gráfico é o fato que partícula que percorreu a maior distância em X não é que ficou mais tempo no ar.

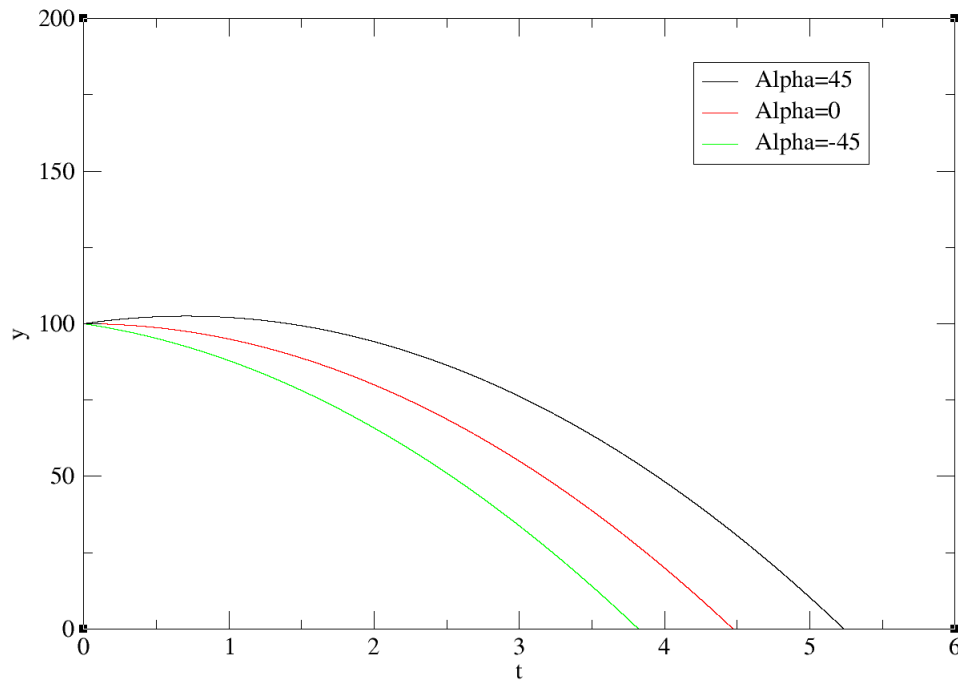


Figura 9: y versus tempo

Algo de interessante ocorre aqui pode se notar que partícula que foi mais longe isso se deve por causa $\alpha=0^\circ$ ter a maior velocidade na direção x, por isso chegou mais longe, mesmo caindo antes. Vamos ver o resultados que o obtemos com equações analíticas quando a partícula chega em $y=0$ para o ângulo 45° ele para em $x=37.016m$, para o ângulo 0° a partícula para em $x = 44.721m$, no ângulo -45° para em $x=27,016m$

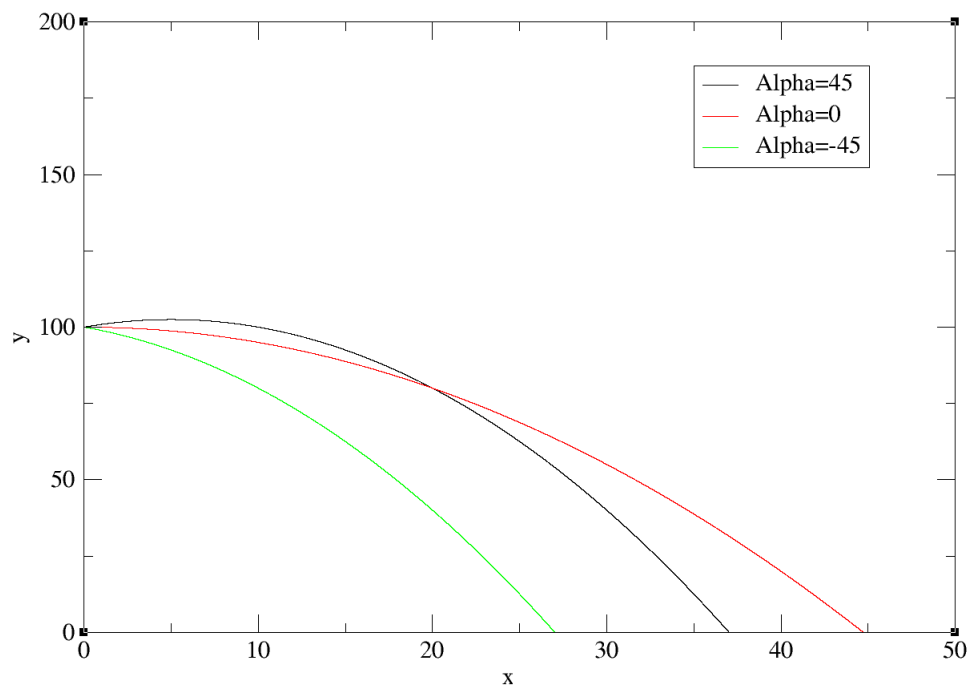


Figura 10: Velocidade versus tempo

3.3 Código C

```

1 program QC
2   implicit real*8(a-h,o-z)
3   real*8,dimension(3)::graus
4   character::nome(9)*21
5   data nome/"saida1-c-11371311.dat","saida2-c-11371311.dat","saida3-c-
6   -11371311.dat","saida4-c-11371311.dat", &
7   &"saida5-c-11371311.dat", "saida6-c-11371311.dat","saida7-c-11371311.
8   dat","saida8-c-11371311.dat",&
9   &"saida9-c-11371311.dat"/
10  graus=(/45.0d0,0.0d0,-45.0d0/)
11
12  !condi es iniciais
13  e0=0.0001
14  v=10
15  y=100
16  x=0
17  i=1
18  t=e0
19  k=11
20  rad=acos(-1.0)*graus(1)/180.0d0
21  vy=v*sin(rad)
22  vx=v*cos(rad)
23
24  !primeiro e0
25  vy=vy-10*e0/2.0d0
26  y=y+vy*e0

```

```

25     x=x+vx*e0
26     open(k,file=nome(k-10))!!foi colocado k-10 por causa que quando
colocamos certos numeros para k ocorre um erro
27     open(k+1,file=nome(k-9))
28     open(k+2,file=nome(k-8))
29     write(k,*)t,y
30     write(k+1,*)t,x
31     write(k+2,*)x,y
32
33     do while (y>=0)
34         x=x+vx*e0
35         y=y+vy*e0
36         vy=vy-10*e0
37         t=t+e0
38
39         if (y<=0 .and. i<3) then!caso y chegue a 0 e0 permuta
40
41             y=100
42             x=0
43             i=i+1
44             k=k+3
45             t=0
46
47             rad=acos(-1.0)*graus(i)/180.0d0
48             vy=v*sin(rad)
49             vx=v*cos(rad)!primeiro e0
50             vy=vy-10*e0/2.0d0
51             open(k,file=nome(k-10))!foi colocado k-10 por causa que
quando colocamos certos numeros para k ocorre um erro
52             open(k+1,file=nome(k-9))
53             open(k+2,file=nome(k-8))
54
55         endif
56
57         write(k,*)t,y
58         write(k+1,*)t,x
59         write(k+2,*)x,y
60     enddo
61 end program

```

4 Parte D

4.1 Forma analítica

Como agora estamos percorrendo uma partícula em uma região bidimensionalmente, assim teremos a resistência do ar agindo em cada componente, já calculamos a da componente y na parte B, vamos calcular agora no eixo x.

$$a = -v_x \gamma \quad (23)$$

$$\frac{dv}{dt} = -v_x \gamma \quad (24)$$

$$\int_{v_0}^{v_x} \frac{dv}{-v_x \gamma} = \int_0^t dt \quad (25)$$

$$v_x = v_0 e^{-\gamma t} \quad (26)$$

Vamos achar agora da posição

$$\frac{dx}{dt} = v_0 e^{-\gamma t} \quad (27)$$

$$\int_{x_0}^x dx = \int_0^t v_0 e^{-\gamma t} dt \quad (28)$$

$$x = \frac{v_0 \cos(\theta)}{\gamma} (e^{-\gamma t} - 1) + x_0 \quad (29)$$

Sabemos que a componente y é mesma da parte B, mas com umas mudanças sendo feitas.

$$y = \frac{(\gamma v_0 \sin(\theta) + g)(1 - e^{-\gamma t}) - \gamma g t}{\gamma^2} + y_0 \quad (30)$$

4.2 Resultados

Como vimos anteriormente na parte C percebemos como curva x versus tempo se comportava que uma fica sobre a outra, isso ocorre novamente na parte D, já que ambas tem velocidade idênticas e sofrem as mesmas forças durante o tempo no ar. Os resultados que ocorrem na parte C de diferente da parte D é o tempo mais longo que partícula fica no ar por causa da resistência do ar $\gamma = 0,1 s^{-1}$.

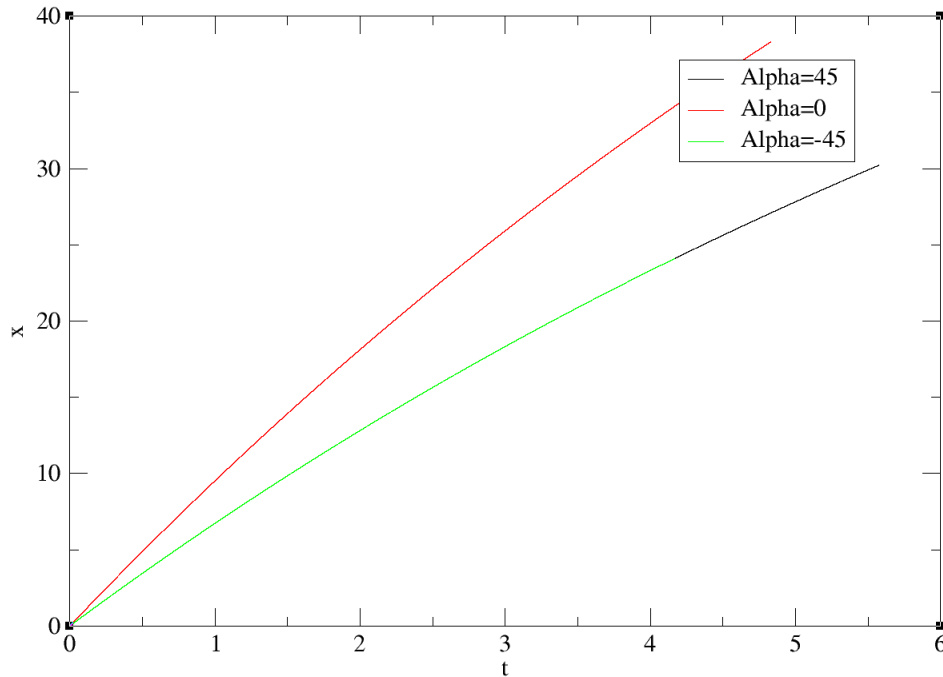


Figura 11: x versus tempo

Usando as formulas analíticas que obtemos lá em cima vamos ver se os resultados que obtemos são condizente, para $\text{Alpha} = -45^\circ$ o tempo que chega $y=0$ $t = 4.1666s$, $\text{alpha} = 45^\circ$ o tempo é $t = 5.577s$ e para $\text{Alpha} = 0^\circ$ o tempo é $t = 4.83183s$

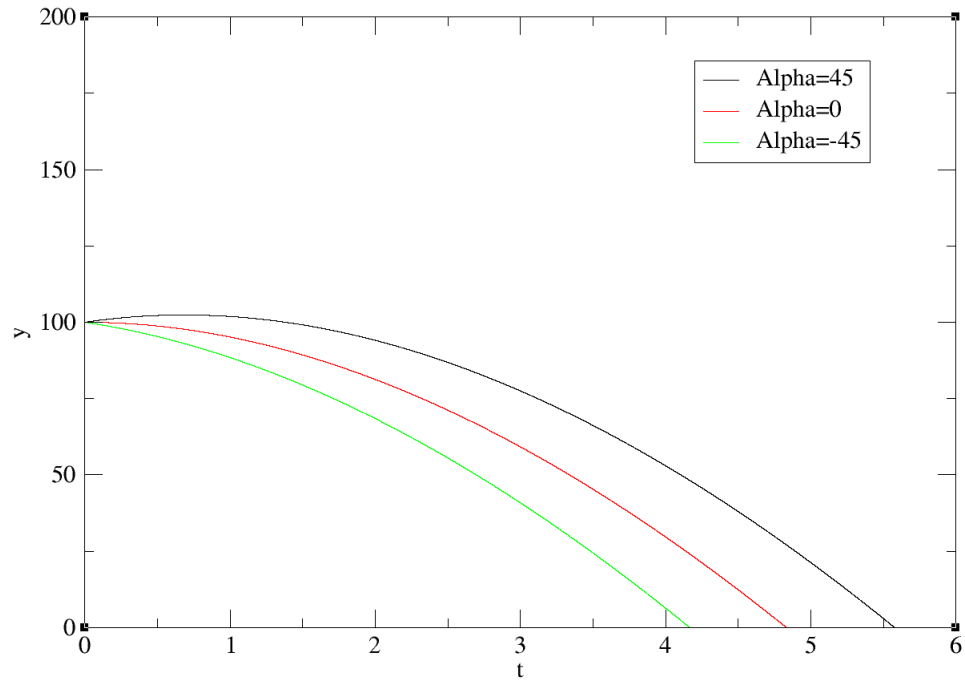


Figura 12: y versus tempo

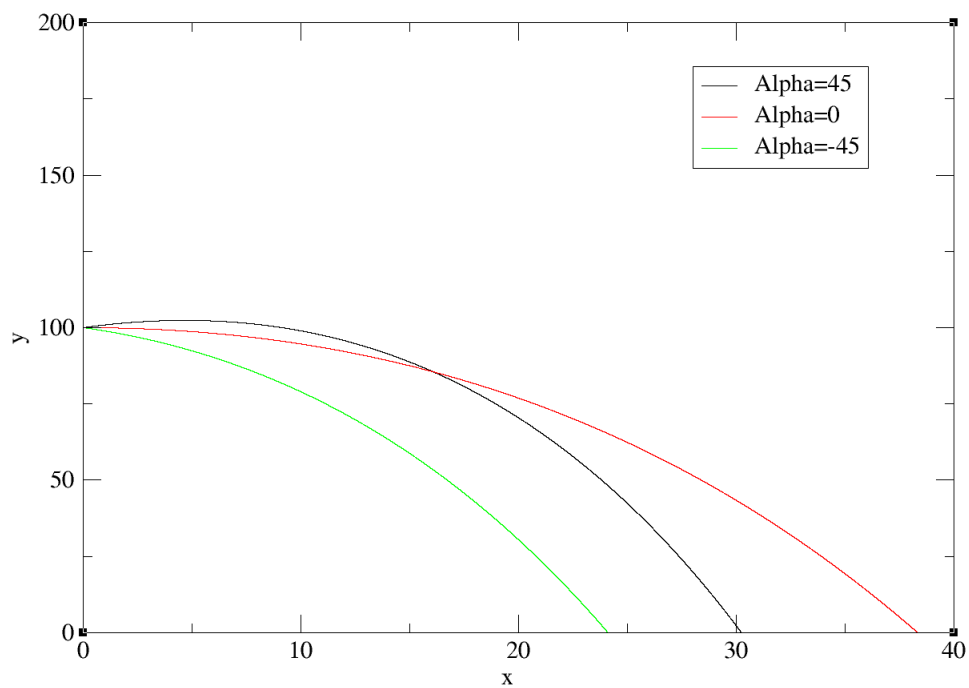


Figura 13: Altura(y) versus Distância(x)

4.3 Código D

```
1 program Qd
2 real*8::vx,vy,x,y,t
3 real*8,dimension(3)::graus
4 character::nome(9)*21
5 data nome/"saida1-b-11371311.dat","saida2-b-11371311.dat","saida3-b
   -11371311.dat","saida4-b-11371311.dat", &
6 &"saida5-b-11371311.dat", "saida6-b-11371311.dat","saida7-b-11371311.dat",
   "saida8-b-11371311.dat",&
7 &"saida9-b-11371311.dat"/
8 graus=(/45.0d0,0.0d0,-45.0d0/)
9 rad=acos(-1.0)*graus(1)/180.0d0
10 !condicoes iniciais
11 e0=0.0001
12 i=1
13 v=10
14 gama=0.1
15 y=100
16 x=0
17 t=0
18 k=11
19 vy=v*sin(rad)
20 vx=v*cos(rad)
21
22 !primeiro e0
23 ay=-10-gama*vy
24 ax=-gama*vx
25 vy=vy+ay*e0/2.0d0
26 vx=vx+ax*e0/2.0d0
27 x=x+vx*e0
28 y=y+vy*e0
29 open(k,file=nome(k-10))!foi colocado k-10 por causa que quando colocamos
   certos numeros para k ocorre um erro
30 open(k+1,file=nome(k-9))
31 open(k+2,file=nome(k-8))
32 write(k,*)t,y
33 write(k+1,*)t,x
34 write(k+2,*)x,y
35
36 do while(y>=0)
37     ay=-10-gama*vy
38     ax=-gama*vx
39     x=x+vx*e0
40     y=y+vy*e0
41     vy=vy+ay*e0
42     vx=vx+ax*e0
43     t=t+e0
44     if(y<=0 .and. i<3)then!caso y chegue a 0, graus permuta
45         y=100
46         x=0
47         i=i+1
48         k=k+3
49         t=0
```

```

50      !convercao de graus para radiano
51      rad=acos(-1.0)*graus(i)/180.0d0
52      vy=v*sin(rad)
53      vx=v*cos(rad)
54      !primeiro e0
55      ay=-10-gama*vy
56      ax=-gama*vx
57      vy=vy+ay*e0/2.0d0
58      vx=vx+ax*e0/2.0d0
59      x=x+vx*e0
60      y=y+vy*e0
61      open(k,file=nome(k-10))!foi colocado k-10 por causa que quando
colocamos certos numeros para k ocorre um erro
62      open(k+1,file=nome(k-9))
63      open(k+2,file=nome(k-8))
64  endif
65  write(k,*)t,y
66  write(k+1,*)t,x
67  write(k+2,*)x,y
68 enddo
69 end program

```

5 Parte E

5.1 Resultados

Vemos nos gráficos abaixo como se comporta o movimento da partícula que ao tocar no chão ($y = 0$) ela é impulsionada de volta com menos 30% da velocidade anteriormente de tocar na superfície.

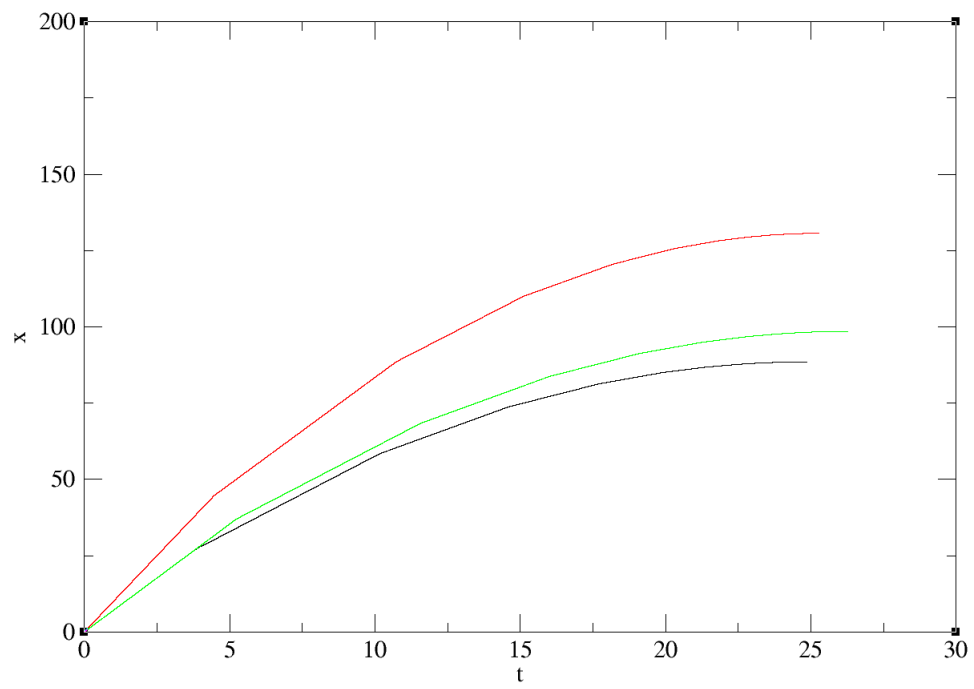


Figura 14: x versus tempo

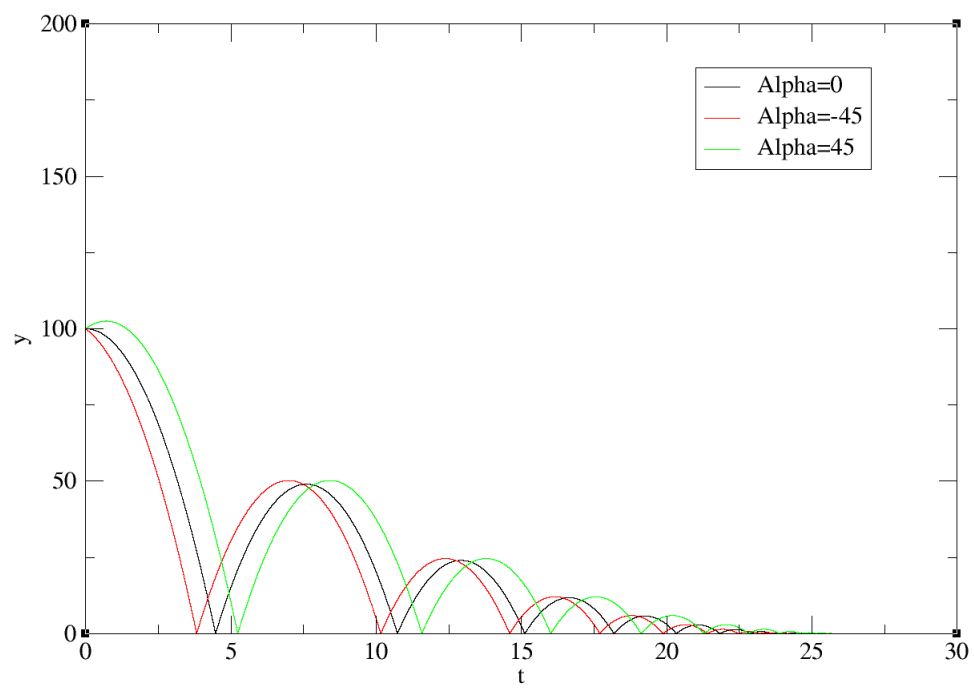


Figura 15: y versus tempo

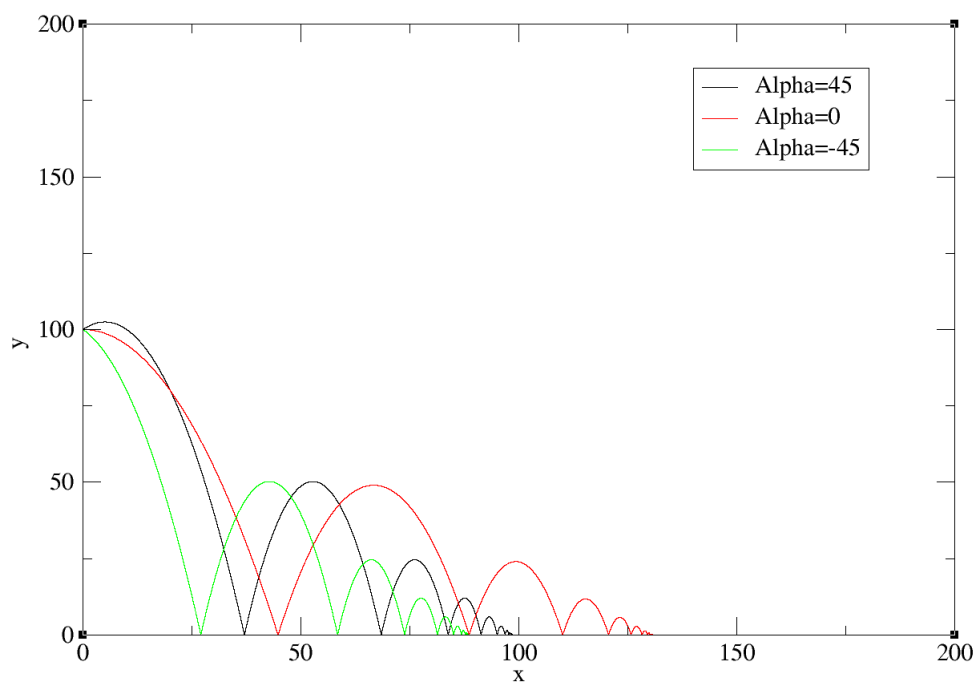


Figura 16: y versus x

Nos gráficos abaixo vamos mostrar o deslocamento, com uma perda maior quando partícula chega em $y=0$

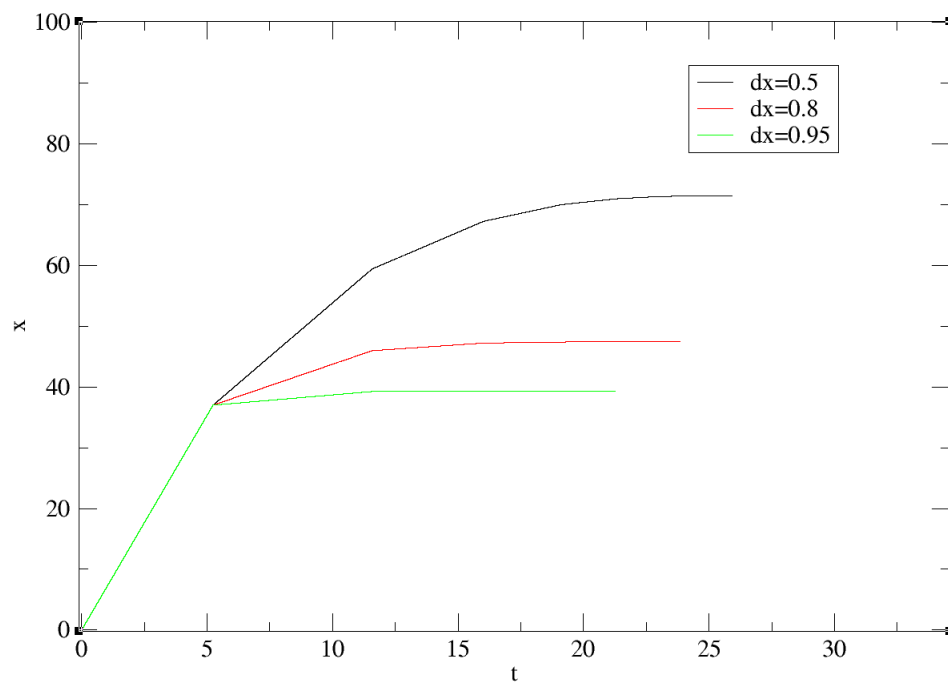


Figura 17: y versus tempo

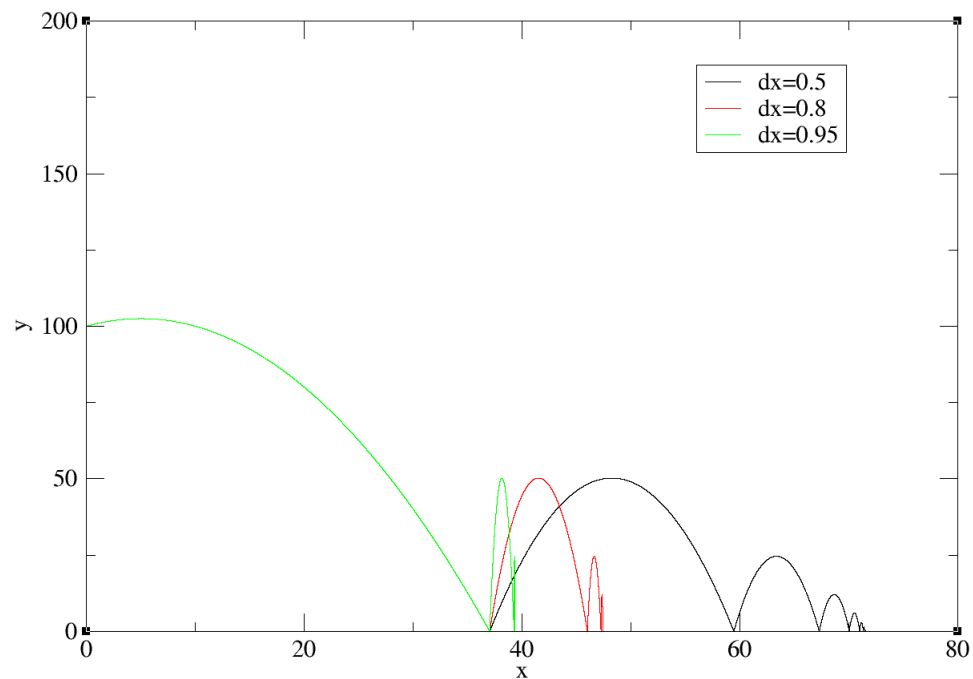


Figura 18: y versus x

5.2 Código E

```

1 program Qe
2     implicit real*8(a-h,o-z)
3     open(1,file="saida1-e-11371311.dat",status="replace")
4     open(2,file="saida2-e-11371311.dat",status="replace")
5     open(3,file="saida3-e-11371311.dat",status="replace")
6
7     !entrada de dados
8     write(*,*)"Coloque a velocidade"
9     read(*,*)v
10    write(*,*)"Coloque o valor de dx desejado"
11    read(*,*)dx
12    write(*,*)"Coloque o valor de alpha(Graus)"
13    read(*,*)graus
14
15    !convercao de graus para radiano
16    rad=acos(-1.0)*graus/180.0d0
17    !condicoes iniciais
18    e0=0.0001
19    y=100
20    x=0
21    t=0
22    vy=v*sin(rad)
23    vx=v*cos(rad)
24
25
26    !primeiro e0
27    vy=vy-10*e0/2.0d0

```

```

28     y=y+vy*e0
29     write(1,*)t,y
30     write(2,*)t,x
31     write(3,*)x,y
32
33
34
35     do while (0==0)
36         x=x+vx*e0
37         xr=x
38         y=y+vy*e0
39         vy=vy-10*e0
40         t=t+e0
41
42         if(abs(xr-x1)<=0.001 .and. y<=0) then
43             exit!caso a distância entre os pontos em que bolinha quicou
for menor 0.001 o loop para.
44
45
46         elseif(y<=0) then
47             vy=abs(vy)*0.7
48             vx=abs(vx)*(1-dx)
49             x1=x
50             !y=0 para caso y de negativo entrando aqui.
51             y=0
52         endif
53
54
55         write(1,*)t,y
56         write(2,*)t,x
57         write(3,*)x,y
58     enddo
59 end program

```

6 Parte F

6.1 Resultados

Nessa parte realizamos a mesmo procedimento anterior, porém com resistência do ar $\gamma = 0,1s^{-1}$

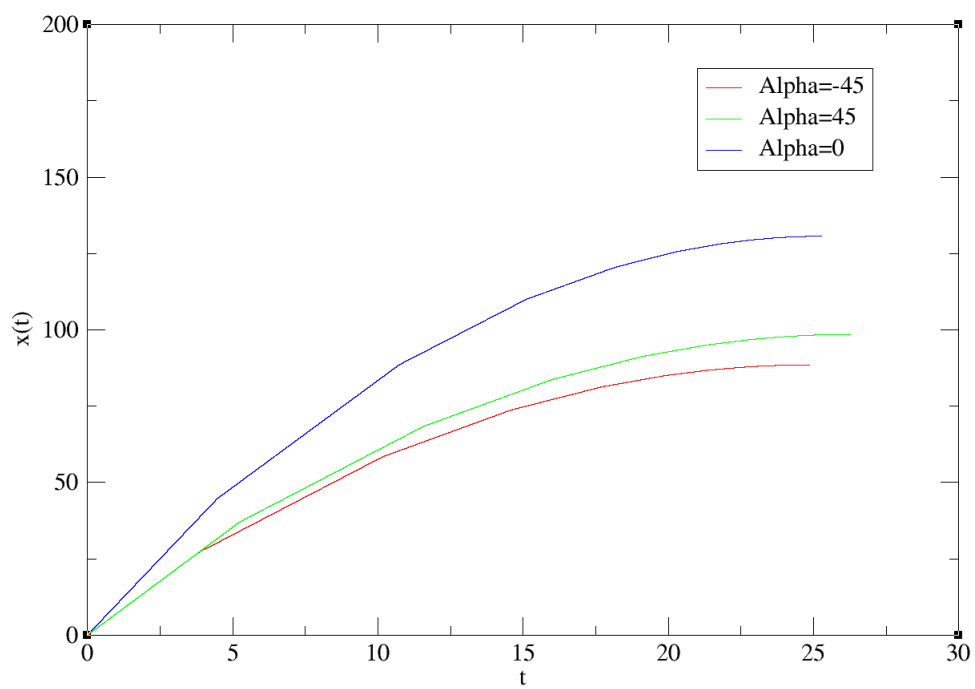


Figura 19: Distância(x) versus tempo

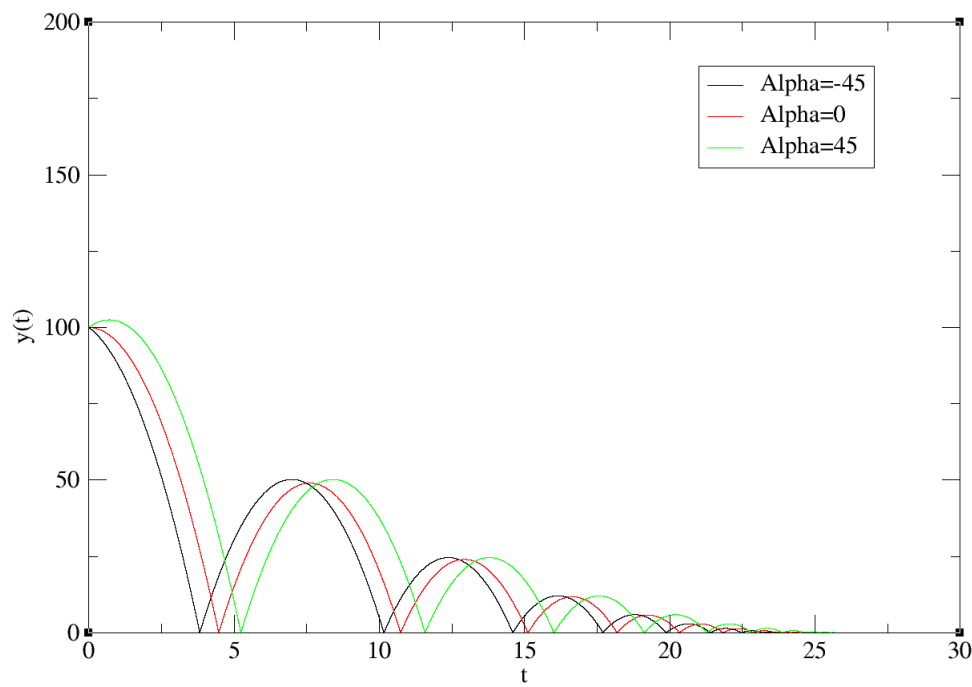


Figura 20: Altura(y) versus tempo

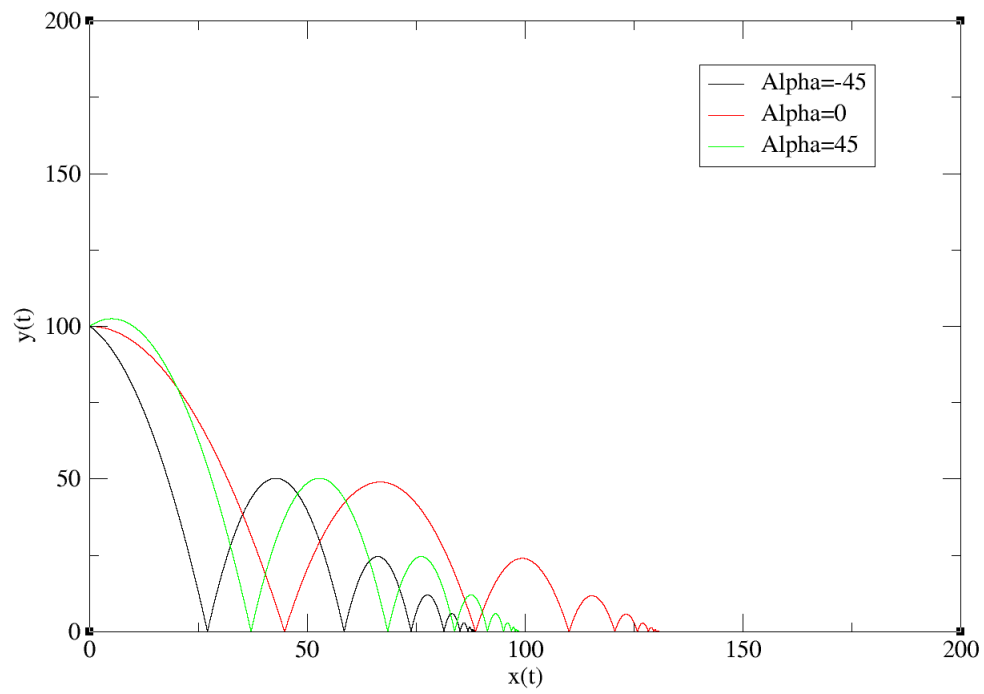


Figura 21: Altura(y) versus Distância(x)

Agora vamos colocar um perda maior na velocidade quando a bolinha chega no chão.

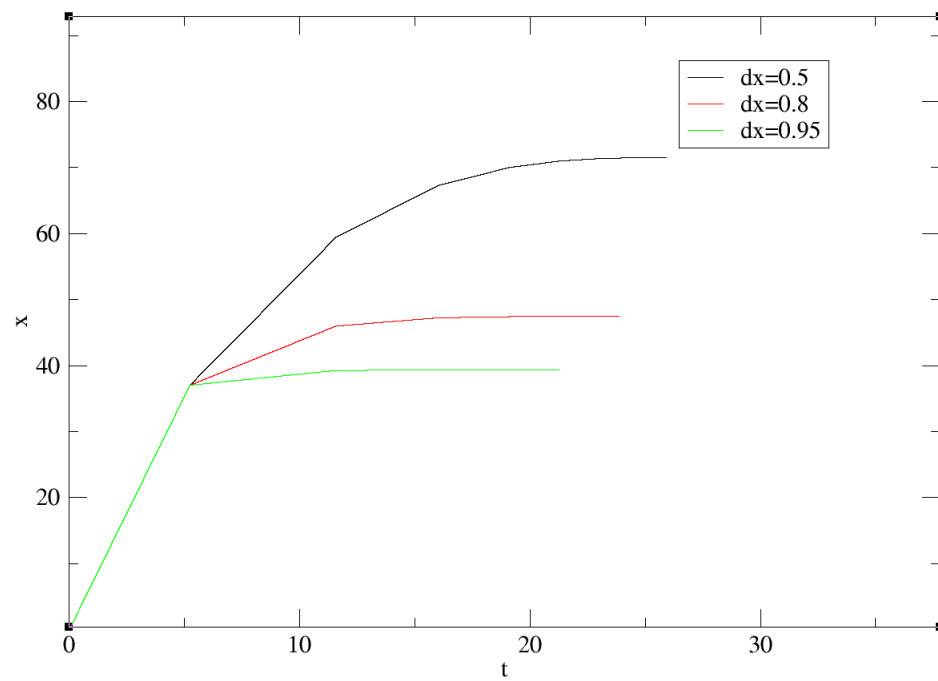


Figura 22: y versus tempo

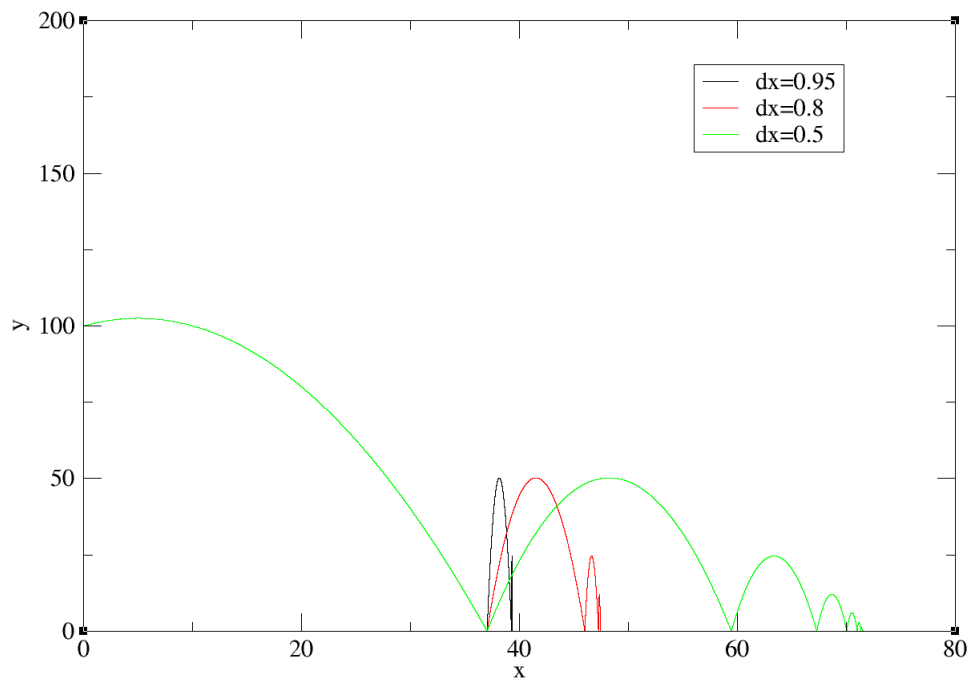


Figura 23: y versus x

6.2 Código F

```

1 program Qf
2 real*8::vx,vy,x,y,t
3 open(1,file="saida1-f-11371311.dat",status="replace")
4 open(2,file="saida2-f-11371311.dat",status="replace")
5 open(3,file="saida3-f-11371311.dat",status="replace")
6 !v0=(/10,0/)
7 !e0=(/0.01,0.001,0.0001/)
8 write(*,*) "Coloque a velocidade"
9 read(*,*)v
10 write(*,*) "Coloque o valor de dx desejado"
11 read(*,*)dx
12 write(*,*) "Coloque o valor de alpha(Graus)"
13 read(*,*)graus
14
15
16 rad=acos(-1.0)*graus/180.0d0
17 y=100
18 e0=0.0001
19 vy=v*sin(rad)
20 vx=v*cos(rad)
21 x=0
22
23
24 !primeiro e0
25 t=e0
26 ay=-10-gama*vy

```

```

27 ax=-gama*vx
28 vy=vy+ay*e0/2.0d0
29 vx=vx+ax*e0/2.0d0
30 x=x+vx*e0
31 y=y+vy*e0
32 write(1,*)t,y
33 write(2,*)t,x
34 write(3,*)x,y
35
36 !loop
37 do while(0==0)
38     x=x+vx*e0
39     y=y+vy*e0
40     xr=x
41     ay=-10-gama*vy
42     ax=-gama*vx
43     vy=vy+ay*e0
44     vx=vx+ax*e0
45     t=t+e0!caso a diferen a de entre os pontos onde a bolinha quicou for
menor 10**-3 o loop para
46     if(abs(xr-xl)<=0.001 .and. y<=0) then
47         exit
48     elseif(y<=0) then
49         xl=x
50         vy=abs(vy)*0.7
51         vx=abs(vx)*(1-dx)
52         y=0
53     endif
54     write(1,*)t,y
55     write(2,*)t,x
56     write(3,*)x,y
57 enddo
58 end program

```