Modelação de Sistemas Físicos - Aula Prática nº9

Realização e resolução de problemas sobre:

- Energia e movimento, integração numérica

Problemas do Capítulo 5: Serviço da bola de ténis

Um jogador de ténis treina o serviço, sacando a bola da linha de base diagonalmente para a sua frente, como ilustrado no diagrama.

O jogador saca a bola do ponto $\mathbf{r}_0 = (x,\, y,\, z) = (0,\, 2,\, 3)\ \mathrm{m}$ com velocidade $\mathbf{v}_0 = (v_{0x},\, v_{0y},\, v_{0z}) = (160,\, 20,\, -20)\ \mathrm{km/h}$.

O serviço é válido se a bola passa por cima da rede (posicionada a $x = 11.9\ \mathrm{m}$, com uma altura $z_{\mathrm{rede}} = 1\ \mathrm{m}$), e cai dentro da área de serviço ($11.9\ \mathrm{m} \leq x \leq 18.3\ \mathrm{m}$ e $4.1\ \mathrm{m} \leq y \leq 8.2\ \mathrm{m}$).

1. Determine o movimento da bola usando o método de Euler a 3D. Considere a força de gravidade e a resistência do ar.

2. Faça um gráfico da trajetória da bola de ténis. Em que ponto a bola cai no solo? O serviço é válido, sim ou não?

3. Calcule a energia mecânica de $t_0 = 0$ até o momento em que a bola bate no solo ($t_{\mathrm{s}} \approx 0.4\ \mathrm{s}$).

4. Calcule o trabalho realizado pela força de resistência do ar até às posições nos três instantes $t_0 = 0$, $t_1 = 0.2\ \mathrm{s}$ e $t_2 = 0.4\ \mathrm{s}$

$$W_{\mathrm{res}} = \int_C \mathbf{F}_{\mathrm{res}} \cdot \mathrm{d}\mathbf{r} = \int_{t_0}^{t_1} \mathbf{F}_{\mathrm{res}} \cdot \mathbf{v}\, \mathrm{d}t$$

Use a aproximação trapezoidal para calcular os integrais. A massa da bola é $M = 57\ \mathrm{g}$.

5. Calcule o trabalho realizado pela força de resistência do ar usando a conservação de energia

$$W_{\mathrm{res}} = E_{\mathrm{c1}} + E_{\mathrm{p1}} - E_{\mathrm{c0}} - E_{\mathrm{p0}}$$

6. Nas mesmas condições do problema anterior, repete o calculo do trabalho realizado pela força de resistência do ar até ao instante $t_1 = 0.4\ \mathrm{s}$, pelos dois métodos, usando os seguintes valores para o intervalo de de tempo $\delta\tau = \{0.1,\, 0.01,\, 0.001,\, 0.0001,\, 0.00001\}$.

Faça um plot em escala semilog do erro dos valores do trabalho calculado, em função de $\delta\tau$.

Solução do ponto 1:

Determine o movimento da bola usando o método de Euler a 3D. Considere a força de gravidade e a resistência do ar.

Iniciamos com a descrição das forças que atuam na bola durante a trajetória, nomeadamente a força gravítica ($\mathbf{F}_{\mathrm{g}}$) e a força da resistência do ar ($\mathbf{F}_{\mathrm{res}}$),

$$\mathbf{F} = \mathbf{F}_{\mathrm{g}} + \mathbf{F}_{\mathrm{res}} = -Mg\,\hat{\mathbf{j}} - MD\,v\,\mathbf{v},$$

onde M é a massa da bola, $D = g/v_{\mathrm{T}}^2$ é o coeficiente de resistência do ar, e $\mathbf{v}$ o vetor velocidade com amplitude v.

Podemos então obter o vetor aceleração (em 3D),

$$\mathbf{a} = -g\,\hat{\mathbf{k}} - D\,v\,\mathbf{v},$$

o que nos permite resolver as equações dinâmicas usando o método de Euler,

$$\mathbf{a}_i = -g\,\hat{\mathbf{k}} - D\,v_i\,\mathbf{v}_i$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i\,\delta\tau$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_i\,\delta\tau$$

em que assumimos as seguintes condições,

- $\mathbf{r}_0 = (0,\, 2,\, 3)\ \mathrm{m}$
- $\mathbf{v}_0 = (160,\, 20,\, -20)\ \mathrm{km/h} = (160,\, 20,\, -20) \times 1000/3600\ \mathrm{m/s}$
- $v_{\mathrm{T}} = v_0$

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt

t0 = 0.0                                  # condição inicial, tempo [s]
tf = 0.5                                  # limite do domínio, tempo final [s]
dt = 0.001                                # passo [s]

r0 = np.array([0.0, 2.0, 3.0])            # condição inicial, posição inicial [m]
v0 = np.array([160.0, 20.0, -20.0]) * 1000 / 3600   # condição inicial, velocidade ini

g = 9.8                             # aceleração gravítica [m/s^2]
v_T = np.linalg.norm(v0)            # velocidade terminal [m/s]
D = g / v_T ** 2                    # coeficiente de resistência do ar [m^-1]

# inicializar domínio [s]
N = int((tf - t0) / dt + 1)
t = np.linspace(t0, tf, num=N)

# inicializar solução, aceleração [m/s^2]
a = np.zeros([np.size(t), 3])           # aceleração resultante
a_res = np.zeros([np.size(t), 3])   # aceleração devido à resistência do ar
a_grv = np.zeros([np.size(t), 3])   # aceleração devido à gravidade

# inicializar solução, velocidade [m/s]
v = np.zeros([np.size(t), 3])
v[0, :] = v0

# inicializar solução, posição [m]
r = np.zeros([np.size(t), 3])
r[0, :] = r0


for i in range(np.size(t) - 1):
    # calcular aceleração
    a_grv[i, :] = - g * np.array([0.0, 0.0, 1.0])
    a_res[i, :] = - D * np.linalg.norm(v[i, :]) * v[i, :]
    a[i, :] = a_grv[i, :] + a_res[i, :]
```

```
    v[i + 1, :] = v[i, :] + a[i, :] * dt
    r[i + 1, :] = r[i, :] + v[i, :] * dt
```

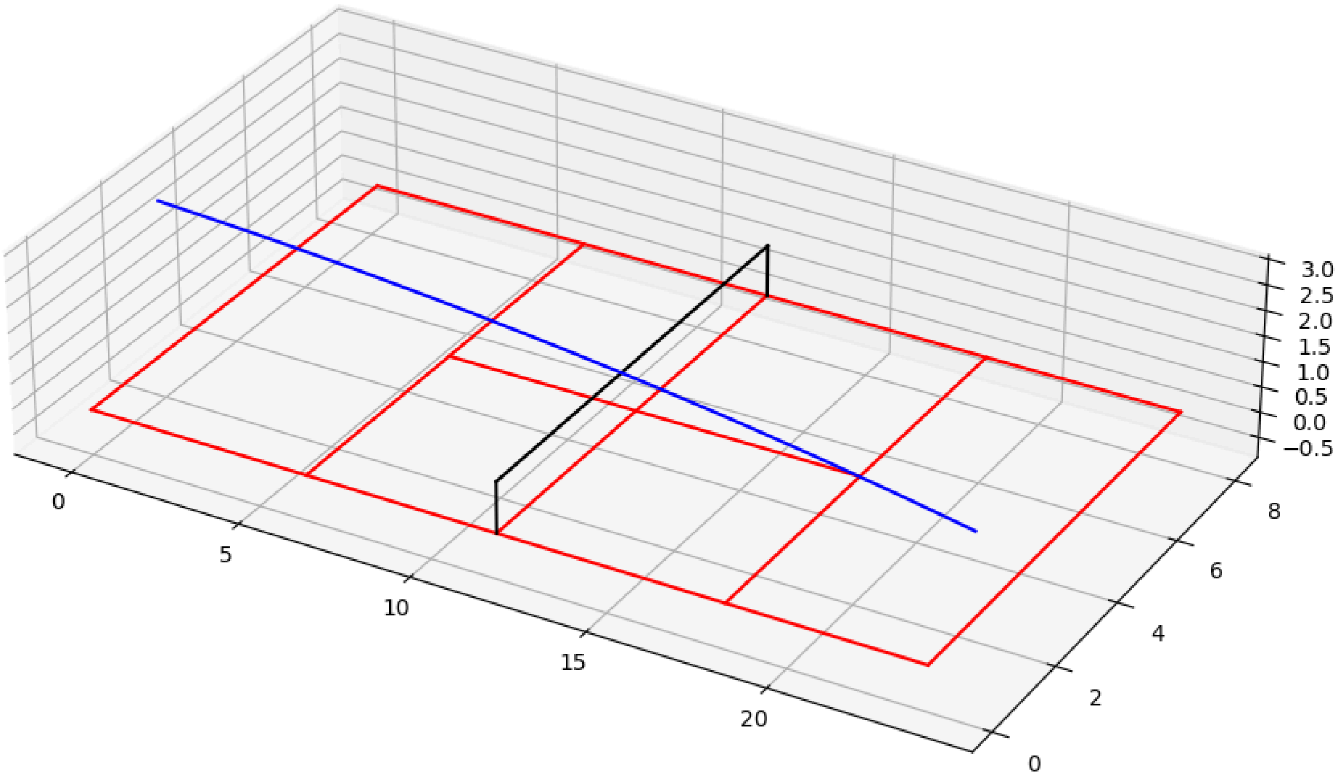
Solução do ponto 2:

Faça um gráfico da trajetória da bola de ténis. Em que ponto a bola cai no solo? O serviço é válido, sim ou não?

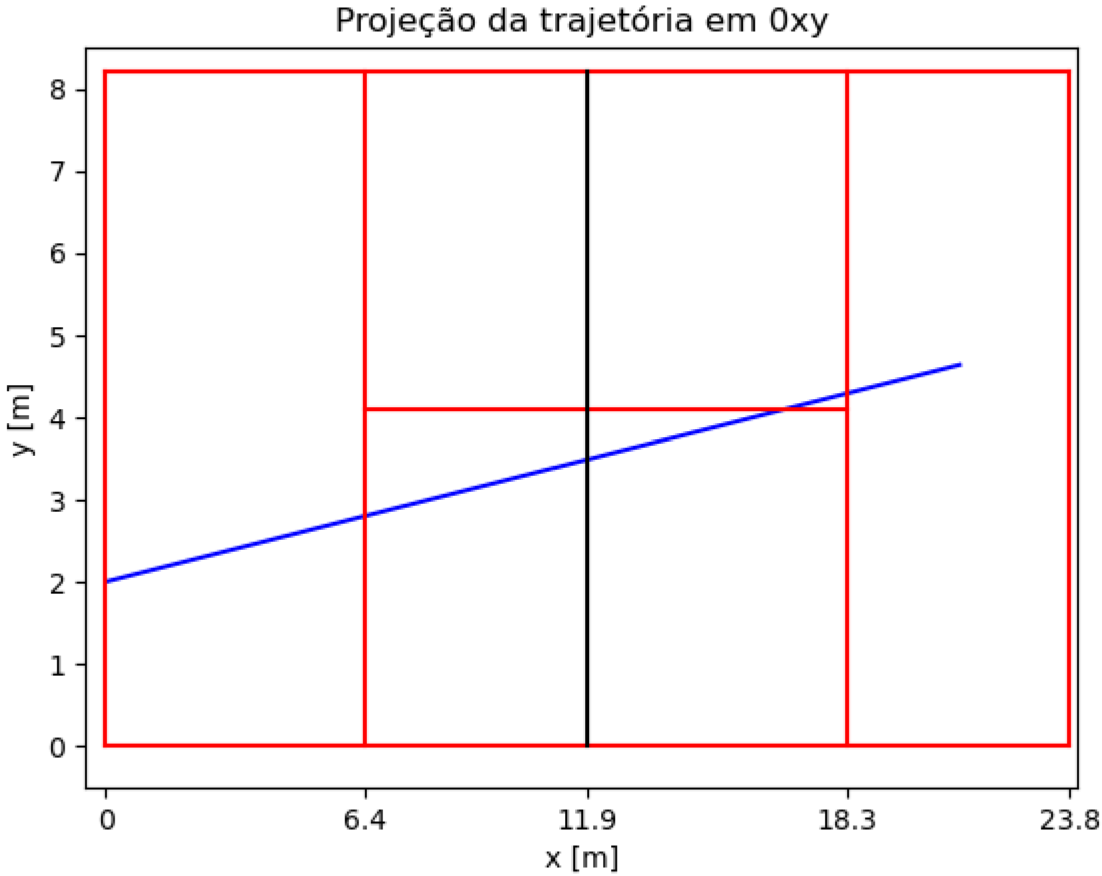
Em baixo representamos a trajetória da bola a 3-dimensões, incluindo as linhas do campo e rede.

In [2]:
```
fig = plt.figure(figsize=(10, 10))
ax = plt.axes(projection='3d')
# Coordenadas das linhas de campo e rede
x0 = 0
x1 = 18.3 - 11.9
x2 = 11.9
x3 = 18.3
x4 = 2 * 11.9
y0 = 0
y1 = 4.1
y2 = 8.2
z0 = 0.0
z1 = 1.0
ax.plot3D(np.array([x0, x4]), np.array([y0, y0]), np.array([z0, z0]), 'r-')
ax.plot3D(np.array([x1, x3]), np.array([y1, y1]), np.array([z0, z0]), 'r-')
ax.plot3D(np.array([x0, x4]), np.array([y2, y2]), np.array([z0, z0]), 'r-')
ax.plot3D(np.array([x0, x0]), np.array([y0, y2]), np.array([z0, z0]), 'r-')
ax.plot3D(np.array([x1, x1]), np.array([y0, y2]), np.array([z0, z0]), 'r-')
ax.plot3D(np.array([x2, x2]), np.array([y0, y2]), np.array([z0, z0]), 'r-')
ax.plot3D(np.array([x3, x3]), np.array([y0, y2]), np.array([z0, z0]), 'r-')
ax.plot3D(np.array([x4, x4]), np.array([y0, y2]), np.array([z0, z0]), 'r-')
ax.plot3D(np.array([x2, x2]), np.array([y0, y2]), np.array([z1, z1]), 'k-')
ax.plot3D(np.array([x2, x2]), np.array([y0, y0]), np.array([z0, z1]), 'k-')
ax.plot3D(np.array([x2, x2]), np.array([y2, y2]), np.array([z0, z1]), 'k-')
# Trajetória da bola
ax.plot3D(r[:, 0], r[:, 1], r[:, 2], 'b-')
ax.set_title('Trajetória da bola de ténis em 3D')
ax.set_box_aspect(aspect = (14, 8, 3))
plt.show()
```

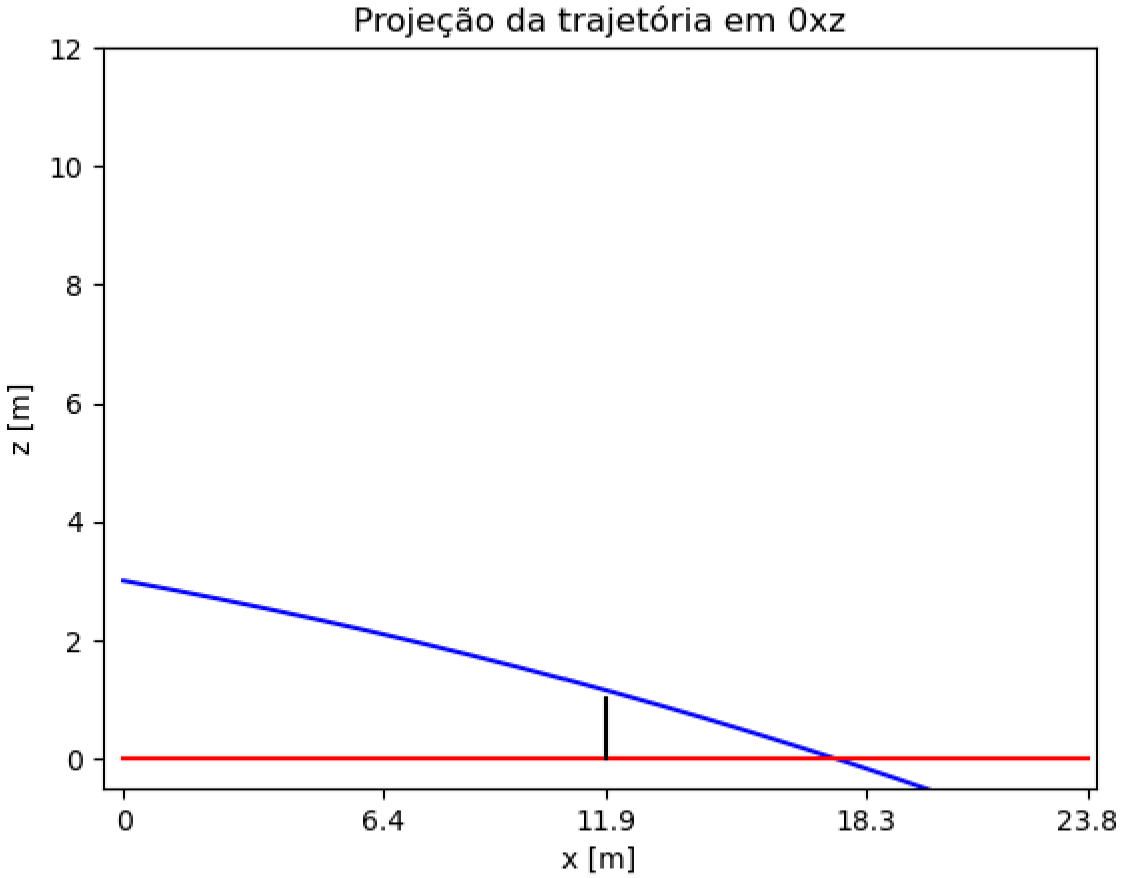
Trajetória da bola de ténis em 3D



In [3]:
```
plt.plot(r[:, 0], r[:, 1], 'b-')
plt.axis([-0.5, 24, -0.5, 8.5])
plt.xlabel("x [m]")
plt.ylabel("y [m]")
plt.plot(np.array([x0, x4]), np.array([y0, y0]), 'r-')
plt.plot(np.array([x1, x3]), np.array([y1, y1]), 'r-')
plt.plot(np.array([x0, x4]), np.array([y2, y2]), 'r-')
plt.plot(np.array([x0, x0]), np.array([y0, y2]), 'r-')
plt.plot(np.array([x1, x1]), np.array([y0, y2]), 'r-')
plt.plot(np.array([x3, x3]), np.array([y0, y2]), 'r-')
plt.plot(np.array([x4, x4]), np.array([y0, y2]), 'r-')
plt.plot(np.array([x2, x2]), np.array([y0, y2]), 'k-')
xtk = [x0, x1, x2, x3, x4]
xlbl = ['0', '6.4', '11.9', '18.3', '23.8']
plt.xticks(xtk, xlbl)
plt.title('Projeção da trajetória em 0xy')
plt.show()
```



In [4]:
```
plt.plot(r[:, 0], r[:, 2], 'b-')
plt.axis([-0.5, 24, -0.5, 12])
plt.xlabel("x [m]")
plt.ylabel("z [m]")
plt.plot(np.array([x0, x4]), np.array([z0, z0]), 'r-')
#plt.plot(np.array([x1, x3]), np.array([y1, y1]), 'r-')
#plt.plot(np.array([x0, x4]), np.array([y2, y2]), 'r-')
#plt.plot(np.array([x0, x0]), np.array([y0, y2]), 'r-')
#plt.plot(np.array([x1, x1]), np.array([y0, y2]), 'r-')
#plt.plot(np.array([x3, x3]), np.array([y0, y2]), 'r-')
#plt.plot(np.array([x4, x4]), np.array([y0, y2]), 'r-')
plt.plot(np.array([x2, x2]), np.array([z0, z1]), 'k-')
xtk = [x0, x1, x2, x3, x4]
xlbl = ['0', '6.4', '11.9', '18.3', '23.8']
plt.xticks(xtk, xlbl)
plt.title('Projeção da trajetória em 0xz')
plt.show()
```



In [5]:
```
# i-ésimo elemento do vetor r cuja componente z é superior a 0.
# É em t(i_ts) que a bola atinge o campo (z = 0).
i_ts = np.size(r[r[:,2] > 0], 0) - 1
ts = t[i_ts]
rs = r[i_ts, :]
print("Tempo no impacto com o campo, t_s = {0:.2f} s".format(ts))
print("Coordenada de impacto no campo, r(t = {0:.2f} s) = ({1:.2f}, {2:.2f}, {3:.2f})
```

```
Tempo no impacto com o campo, t_s = 0.41 s
Coordenada de impacto no campo, r(t = 0.41 s) = (17.54, 4.19, 0.00) m
```

Podemos assim concluir que o serviço é válido e a bola atinge a zona de serviço.

Solução do ponto 3:

Calcule a variação da energia mecânica de $t_0 = 0$ até ao momento em que a bola bate no solo ($t_\mathrm{s} = 0.41$ s).

A energia mecânica é dada por:

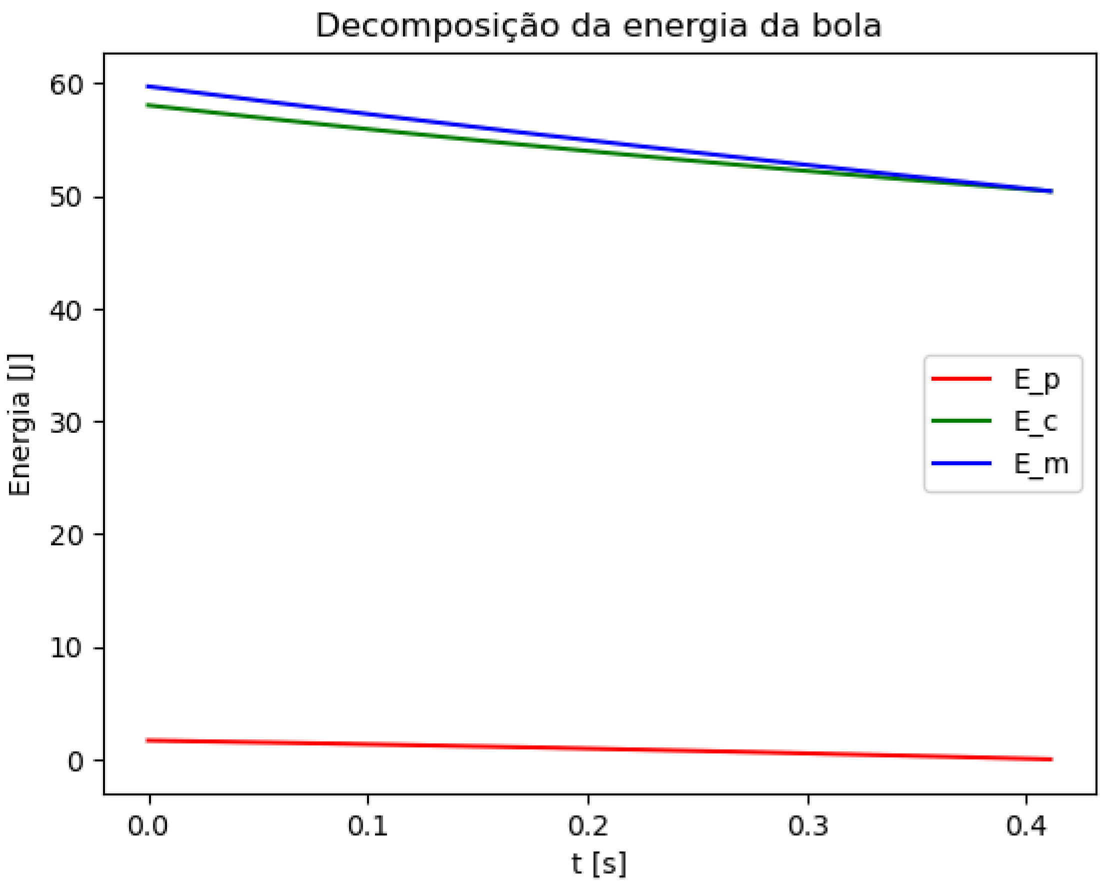
$$E_\mathrm{m} = E_\mathrm{p} + E_\mathrm{c} = M\,g\,z + \frac{1}{2}Mv^2$$

onde $M = 57\ \mathrm{g} = 57\ \mathrm{kg}/1000 = 0.057\ \mathrm{kg}$

In [6]:
```
M = 0.057                                # massa da bola, [kg]
v_norm = np.linalg.norm(v, axis=1)       # norma da velocidade, |v(t)|
E_p = M * g * r[0:i_ts, 2]               # energia potencial, [J]
E_c = 0.5 * M * v_norm[0:i_ts] ** 2      # energia cinética, [J]
E_m = E_p + E_c                          # energia mecânica, [J]

# representação gráfica das energias
plt.plot(t[0:i_ts], E_p, 'r-', t[0:i_ts], E_c, 'g-',  t[0:i_ts], E_m, 'b-')
plt.legend(['E_p', 'E_c', 'E_m'])
plt.xlabel("t [s]")
plt.ylabel("Energia [J]")
```

```
plt.title('Decomposição da energia da bola')
plt.show()
```



- A energia potencial diminui porque a altura da bola diminui;
- A energia cinética diminui porque a velocidade diminui (resistência do ar);
- A energia mecânica diminui até atingir o campo. Estamos na presença de uma perda de energia (campo não conservativo).
- Podemos também concluir que até atingir o campo, a bola perdeu (dissipou) uma quandidade de energia dada por:

In [7]:
```
print("E_m(t = t_s) - E_m(t = 0) = {0:.2f} J".format(E_m[-1] - E_m[0]))
```

```
E_m(t = t_s) - E_m(t = 0) = -9.30 J
```

Solução do ponto 4:

Calcule o trabalho realizado pela força de resistência do ar até às posições nos três instantes $t_0 = 0$, $t_1 = 0.2\,\mathrm{s}$ e $t_2 = 0.4\,\mathrm{s}$

$$W_{\text{res}} = \int_C \mathbf{F}_{\text{res}} \cdot \mathrm{d}\mathbf{r} = \int_{t_0}^{t_1} \mathbf{F}_{\text{res}} \cdot \mathbf{v}\,\mathrm{d}t$$

Use a aproximação trapezoidal para calcular os integrais. A massa da bola é $M = 57\ \mathrm{g}$.

Vamos usar a aproximação trapezoidal para calcular o integral, e para isso definimos uma função que calcula:

$$\int_a^b f(t)\,\mathrm{d}t \approx \sum_i \frac{f_i + f_{i+1}}{2} \delta\tau,$$

onde $f_i = f(t_i)$ e o domínio de integração está equipartido segundo $a = t_0 < t_1 < \cdots < t_{N-1} = b$, em que $t_i - t_{i-1} = \delta\tau$.

In [8]:
```
def integral(f, D, a, b):
    """
    Integração de uma função entre dois limites
    https://en.wikipedia.org/wiki/Trapezoidal_rule
    f: array, [f_0, f_1, ..., f_N-1]
    D: Limites do domínio da função f
    D: array, [D0, D1], f(D0) = f_0, f(D1) = f_N-1
    a, b: float, limites de integração
    """
    # https://en.wikipedia.org/wiki/Trapezoidal_rule
    dt = (D[1] - D[0]) / len(f)
    i_a = int((a - D[0]) / dt)
    i_b = int((b - D[0]) / dt)
    sum = 0.0
    for i in range(i_a, i_b):
        sum += (f[i] + f[i + 1]) / 2.0 * dt
    return sum


t1 = 0.2
t2 = 0.4
D = np.array([t0, tf])
F_res = M * a_res
F_dot_v = np.sum(F_res * v, axis = 1)
W0 = integral(F_dot_v, D, t0, t0)
W1 = integral(F_dot_v, D, t0, t1)
W2 = integral(F_dot_v, D, t0, t2)
```

Assim, o trabalho realizado pela força de resistência do ar até às posições nos três instantes $t_0 = 0$, $t_1 = 0.2$ s e $t_2 = 0.4$ s é dado por

In [9]:
```
print("W_res(t0 = 0.1 s) = {0:.2f} J".format(W0))
print("W_res(t1 = 0.2 s) = {0:.2f} J".format(W1))
print("W_res(t2 = 0.4 s) = {0:.2f} J".format(W2))
```

```
W_res(t0 = 0.1 s) = 0.00 J
W_res(t1 = 0.2 s) = -4.76 J
W_res(t2 = 0.4 s) = -9.06 J
```

Solução do ponto 5:

Calcule o trabalho realizado pela força de resistência do ar entre $t_0 = 0$ e $t_2 = 0.4$ s usando a conservação de energia

$$W_{\rm res} = E_{\rm c}(t_2) + E_{\rm p}(t_2) - E_{\rm c}(t_0) - E_{\rm p}(t_0)$$

In [10]:
```
i_t0 = int(t0 / dt)
i_t2 = int(t2 / dt)
print("A perda de energia mecânica e dE_m = {0:.2f} J".format(E_m[i_t2] - E_m[i_t0]))
```

```
A perda de energia mecânica e dE_m = -9.08 J
```

Solução do ponto 6:

Nas mesmas condições do problema anterior, repete o calculo do trabalho realizado pela força de resistência do ar até ao instante $t_2 = 0.4$ s, pelos dois métodos, usando os seguintes valores para o incremento do tempo $\delta\tau = \{0.1,\, 0.01,\, 0.001,\, 0.0001,\, 0.00001\}$.

Faça um gráfico em escala semilog do erro dos valores do trabalho calculado, em função de $\delta\tau$.

Para resolver este problema, necessitamos em primeiro lugar de um valor de referência para a energia dissipada (trabalho da força da resistência do ar $W_{\rm res}$ entre t_0 e t_2), com uma precisão mais elevada do que a obtida com $\delta\tau = 10^{-5}$ s. Atribuindo o valor $\delta\tau = 10^{-6}$ s obtemos $W_{\rm res,ref} = -9.078705$ J.

Assim, executamos tudo novamente, alterando o valor da variável `dt` na célula `[1]`. colocando o resultado na célula em baixo de forma a podermos representar graficamente num gráfico em escala semilogarítmica.

In [11]:
```
W_res_ref = -9.078705  # valor de referência para W_res, J
F = M * a_res
F_dot_v = np.sum(F * v, axis = 1)
W_res_trab = integral(F_dot_v, D, t0, t2)   # W_res pelo trabalho dissipado
W_res_emec = E_m[i_t2] - E_m[i_t0]         # W_res pela perda de energia mecânica
print("W_res_trab = ({0:.6f} ± {1:.6f})  J".format(W_res_trab, np.abs(W_res_trab - W_
print("W_res_emec = ({0:.6f} ± {1:.6f})  J".format(W_res_emec, np.abs(W_res_emec - W_
```

```
W_res_trab = (-9.060397 ± 0.018308)  J
W_res_emec = (-9.079159 ± 0.000454)  J
```

In [12]:
```
dt_vec = np.array([0.1, 0.01, 0.001, 0.0001, 0.00001])
Erro_res_trab = np.array([2, 0.1, 0.02, 0.001, 0.000006])
Erro_res_emec = np.array([0.05, 0.004, 0.0005, 0.00004, 0.000001])

# representação gráfica do erro
plt.loglog(dt_vec, Erro_res_trab, 'ro')
plt.loglog(dt_vec, Erro_res_emec, 'bo')
plt.legend(['Erro_res_trab', 'Erro_res_emec'])
plt.xlabel("dt [s]")
plt.ylabel("Erro [J]")
plt.show()
```