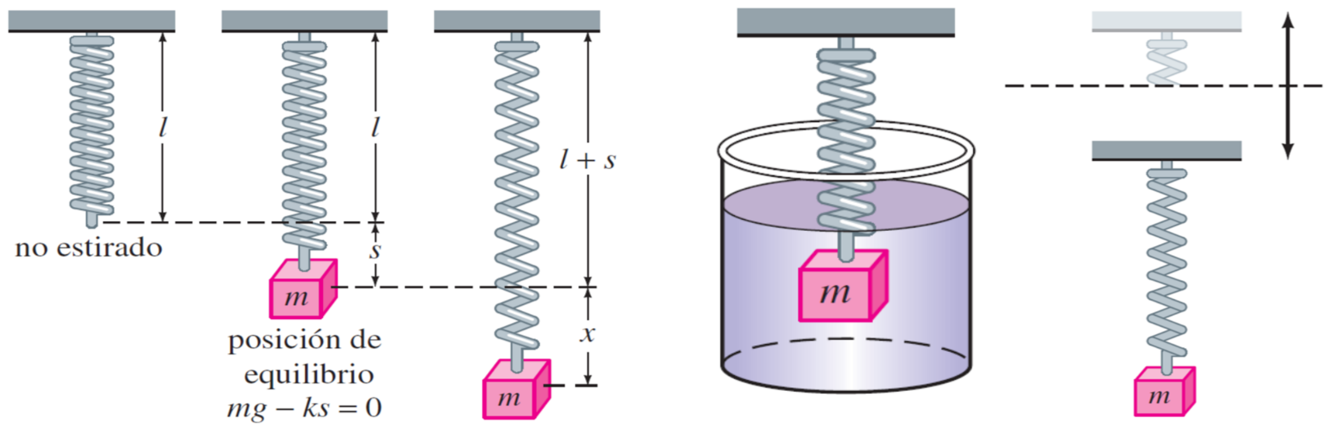
>

PROGRAMA PARA RESOLVER Y SIMULAR UN SISTEMA MASA-RESORTE



mX''+βX'+kX=f(t)

Ingrese los valores de m, β y k y f(t)

> $m\_ := \frac{1}{4};\ \beta\_ := 2;\ k\_ := 4;\ f\_ := 5\cdot\cos(4\cdot x)$

$$m\_ := \frac{1}{4}$$
$$\beta\_ := 2$$
$$k\_ := 4$$
$$f\_ := 5\cos(4\,x) \tag{1}$$

> $a := 1;\ b := convert\left(\frac{\beta\_}{m\_}, rational\right);\ c := convert\left(\frac{k\_}{m\_}, rational\right);\ F := convert\left(\frac{f\_}{m\_}, rational\right);$

$$a := 1$$
$$b := 8$$
$$c := 16$$
$$F := 20\cos(4\,x) \tag{2}$$

Ecuación diferencial

> $a\cdot X'' + b\cdot X' + c\cdot X = F$

$$\frac{d^2}{dx^2} X(x) + 8\left(\frac{d}{dx} X(x)\right) + 16\,X(x) = 20\cos(4\,x) \tag{3}$$

Solución ecuación complementaria

> $a\cdot m^2 + b\cdot m + c = 0$

$$m^2 + 8\,m + 16 = 0 \tag{4}$$

> $solve(a \cdot m^2 + b \cdot m + c = 0, m)$

$$-4, -4 \tag{5}$$

Conjunto de soluciones ecuación homogenea

> **if** $b^2 - 4 \cdot a \cdot c \geq 0$ **then**
> $m1 := \frac{-b + \sqrt{(b)^2 - 4 \cdot a \cdot c}}{2 \cdot a}$; $m2 := \frac{-b - \sqrt{(b)^2 - 4 \cdot a \cdot c}}{2 \cdot a}$ **elif** $b^2 - 4 \cdot a \cdot c < 0$
> **then** $\alpha := \frac{-b}{2 \cdot a}$; $\beta := \frac{\sqrt{|b^2 - 4 \cdot a \cdot c|}}{2 \cdot a}$
> **end if**

$$m1 := -4$$
$$m2 := -4 \tag{6}$$

> **if** $b^2 - 4 \cdot a \cdot c > 0$ **then** $X1 := \exp(m1 \cdot x)$; $X2 := \exp(m2 \cdot x)$ **elif** $b^2 - 4 \cdot a \cdot c = 0$ **then** $X1 :=$ $\exp(m1 \cdot x)$; $X2 := x \cdot \exp(m2 \cdot x)$ **elif** $b^2 - 4 \cdot a \cdot c < 0$ **then** $X1 := \exp(\alpha \cdot x) \cdot \cos(\beta \cdot x)$; $X2 := \exp(\alpha \cdot x) \cdot \sin(\beta \cdot x)$ **end if**

$$X1 := e^{-4x}$$
$$X2 := x\, e^{-4x} \tag{7}$$

Solución complementaria (transitoria)

> $Xc := c1 \cdot X1 + c2 \cdot X2$

$$Xc := e^{-4x} c1 + x\, e^{-4x} c2 \tag{8}$$

Calculo de wronskianos: w, w1 y w2

> $with(VectorCalculus) :$

> $W := Wronskian([X1, X2], x)$

$$W := \begin{bmatrix} e^{-4x} & x\, e^{-4x} \\ -4\, e^{-4x} & e^{-4x} - 4\, x\, e^{-4x} \end{bmatrix} \tag{9}$$

> $with(LinearAlgebra) :$

> $w := simplify(Determinant(W))$

$$w := e^{-8x} \tag{10}$$

> $W1 := \begin{bmatrix} 0 & X2 \\ F & \frac{d}{dx}(X2) \end{bmatrix}$

$$W1 := \begin{bmatrix} 0 & x\, e^{-4x} \\ 20 \cos(4x) & e^{-4x} - 4\, x\, e^{-4x} \end{bmatrix} \tag{11}$$

> $w1 := simplify(Determinant(W1), trig)$

$$w1 := -20\, x\, e^{-4x} \cos(4x) \tag{12}$$

> $W2 := \begin{bmatrix} X1 & 0 \\ \frac{d}{dx}(X1) & F \end{bmatrix}$

$$W2 := \begin{bmatrix} e^{-4x} & 0 \\ -4\,e^{-4x} & 20\cos(4x) \end{bmatrix} \tag{13}$$

> $w2 := combine(Determinant(W2), trig)$

$$w2 := 20\,e^{-4x}\cos(4x) \tag{14}$$

Calculo de funciones: u1 y u2

> $u1 := combine\left(\int \frac{w1}{w}\,dx, trig\right)$

$$u1 := -\frac{5}{2}\,x\,e^{4x}\cos(4x) - \frac{5}{2}\,e^{4x}\sin(4x)\,x + \frac{5}{8}\,e^{4x}\sin(4x) \tag{15}$$

> $u2 := combine\left(\int \frac{w2}{w}\,dx, trig\right)$

$$u2 := \frac{5}{2}\,e^{4x}\cos(4x) + \frac{5}{2}\,e^{4x}\sin(4x) - \frac{5}{2}\,e^{4x} + \frac{5}{2}\,(e^{x})^{4} \tag{16}$$

>

Solución particular (estado estable)

> $Xp := simplify(combine(u1\cdot X1 + u2\cdot X2, trig))$

$$Xp := \frac{5}{8}\sin(4x) \tag{17}$$

>

Solución general de la ecuación de posición

> $dsolve(\ a\cdot X'' + b\cdot X' + c\cdot X = F, \{\ X(x)\ \}\ )$

$$X(x) = e^{-4x}\,\_C2 + x\,e^{-4x}\,\_C1 + \frac{5}{8}\sin(4x) \tag{18}$$

> $X\_ := Xc + Xp$

$$X\_ := e^{-4x}\,c1 + x\,e^{-4x}\,c2 + \frac{5}{8}\sin(4x) \tag{19}$$

Solución general de la ecuación de velocidad

> $v := \frac{d}{dx}\,(X\_)$

$$v := -4\,e^{-4x}\,c1 + e^{-4x}\,c2 - 4\,x\,e^{-4x}\,c2 + \frac{5}{2}\cos(4x) \tag{20}$$

Condiciones iniciales

> $X\_0 := 0;\ V\_0 := -3$

$$X\_0 := 0$$
$$V\_0 := -3 \tag{21}$$

Sistema de ecuaciones

> $simplify(eval(X\_, x=0) = X\_0)$

$$c1 = 0 \tag{22}$$

> $simplify(eval(v, x=0) = V\_0)$

(23)

$$-4\,c1 + c2 + \frac{5}{2} = -3 \qquad (23)$$

Solución sistema de ecuaciones

> $solve(\{eval(X\_, x=0) = X\_0, (eval(v, x=0) = V\_0)\}, \{c1, c2\})$

$$\left\{c1 = 0, c2 = -\frac{11}{2}\right\} \qquad (24)$$

> $C1 := 0; C2 := -\frac{11}{2}$

$$C1 := 0$$

$$C2 := -\frac{11}{2} \qquad (25)$$

Función de posición respecto al tiempo

> $dsolve[':-interactive'](\,a\cdot X'' + b\cdot X' + c\cdot X = F)$

$$X(x) = -\frac{11}{2}\,x\,\mathrm{e}^{-4\,x} + \frac{5}{8}\,\sin(4\,x) \qquad (26)$$

> $x\_ = eval(X\_, \{c1 = C1, c2 = C2\})$

$$x\_ = -\frac{11}{2}\,x\,\mathrm{e}^{-4\,x} + \frac{5}{8}\,\sin(4\,x) \qquad (27)$$

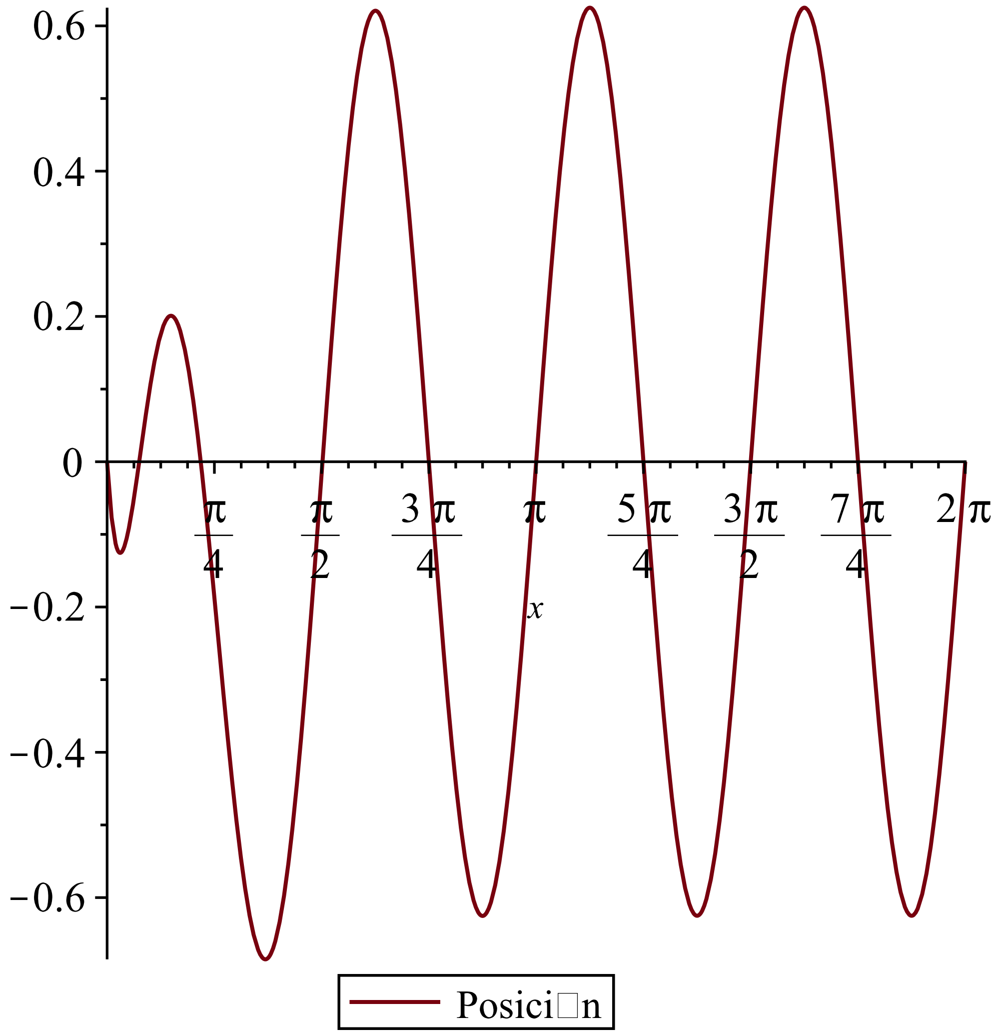
Función de velocidad respecto al tiempo

> $v\_ := eval(v, \{c1 = C1, c2 = C2\})$

$$v\_ := -\frac{11}{2}\,\mathrm{e}^{-4\,x} + 22\,x\,\mathrm{e}^{-4\,x} + \frac{5}{2}\,\cos(4\,x) \qquad (28)$$

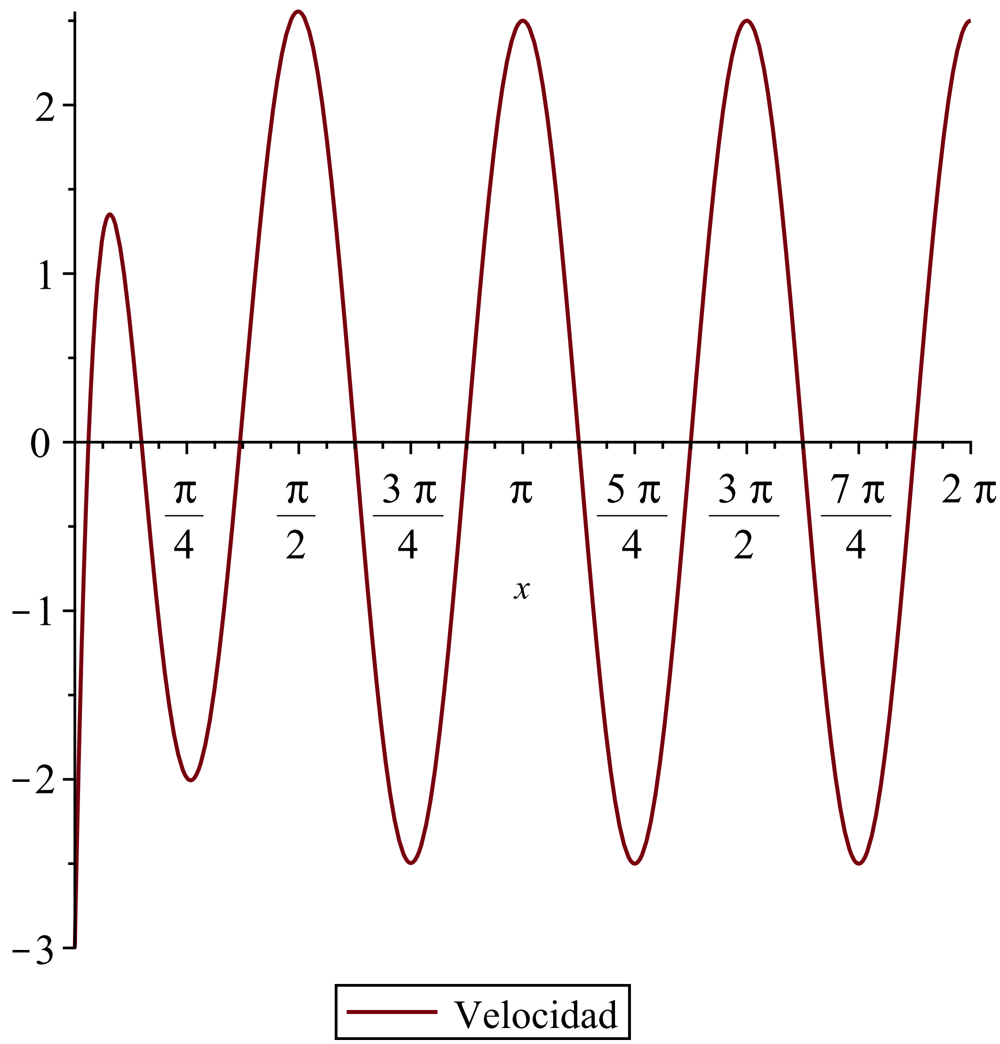
Grafica de la función de posición respecto al tiempo

> $plot(eval(X\_, \{c1 = C1, c2 = C2\}), x = 0 .. 2\cdot\mathrm{Pi})$



Grafica de la función de velocidad respecto al tiempo

> *plot*(*v_*, *x* = 0 .. 2 ·Pi)

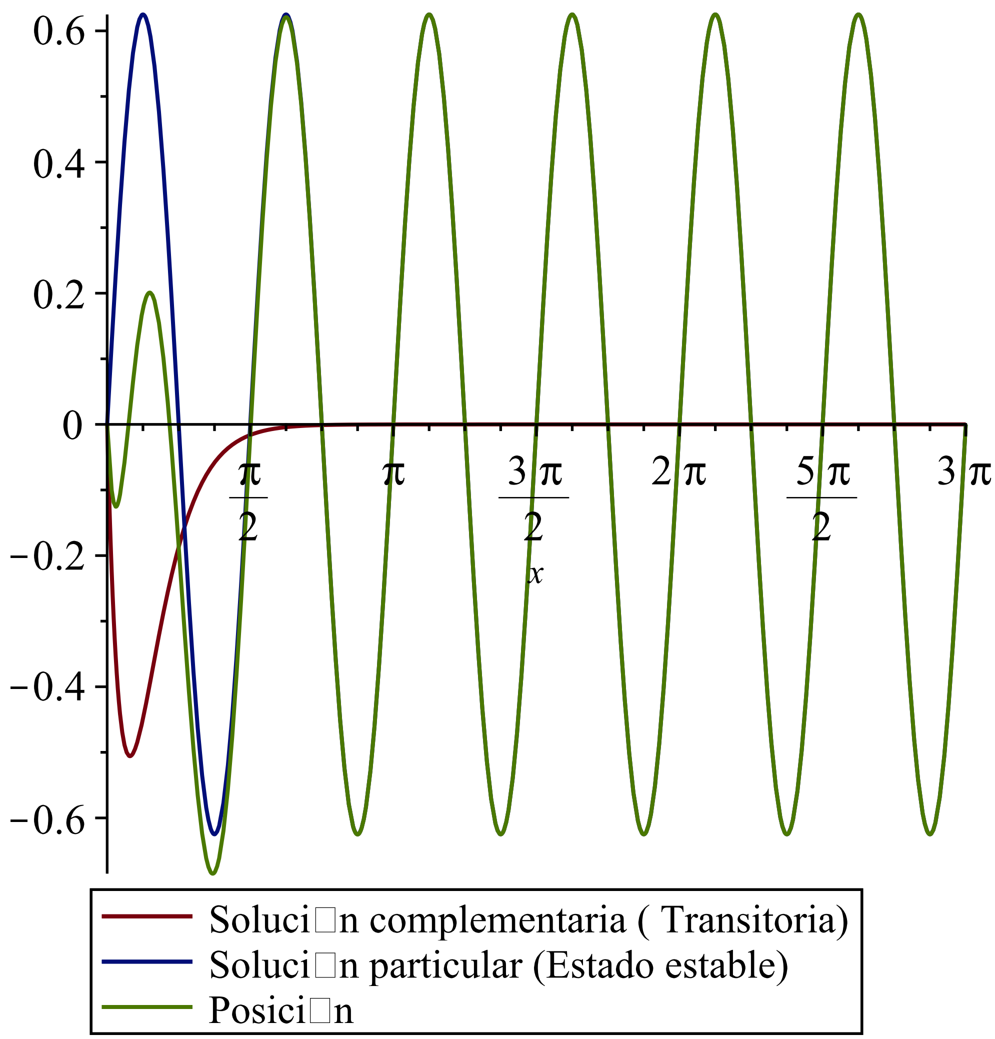


Grafica de la función de posición, solución transitoria y solucion particular

> $Xc\_ := eval(Xc, \{c1 = C1, c2 = C2\})$

$$Xc\_ := -\frac{11}{2}\, x\, e^{-4x} \tag{29}$$

> $plot(\{eval(X\_, \{c1 = C1, c2 = C2\}), Xc\_, Xp\}, x = 0..3 \cdot \mathrm{Pi})$



Simulación sistema masa-resorte

```
> mass_spring := proc(m, r, k, x1, x2, x0, xp0, tk, f, n )
   local mass, deq, init, sol, xk, xu, plt1, plt2, plt, pltxk, rect, base, spring,
   spring1, rod, cylinder, piston, fluid, dashpot;
   with(plots) : with(plottools) :
         spring := proc(x1, x2, n) #procedure for the spring
           local p1, p2, p3, p4, pn_1, pn, p;
           p1 := [x1, 0];
           p2 := [x1 + 0.25, 0];
           p3 := [x2 - 0.25, 0];
         p4 := [x2, 0];
           p := i → [x1 + 0.25 + (x2 - x1 - 0.5) / n * i, ( - 1) ^ (i + 1) * 0.5];
       plot( [p1, p2, seq( p(i), i = 1 ..n - 1), p3, p4], thickness = 2, color = aquamarine);
         end:
   spring1 := x2 → spring(x1, x2, 12);
   rect := t → translate(rectangle( [ - 0.2, - 0.2], [0.2, 0.2], color = red), t, xk(t) + x2 + 1);
   #rect: display the position of the mass on the displacement curve x(t)
   mass := x2 → rectangle( [x2, - 1], [x2 + 2, 1], color = red) :
         if x0 = 0 and xk(0.1) = 0 then
         xu := 1.5
         elif x0 = 0 and xk(0.1) ≠ 0 then
```

```
        xu := 5.0;
      else
        xu := x0;
      fi;
    rod := x2→plot( [ [x2 + 2, 0], [x2 + 2.5 + 2 * xu, 0]], color = grey, thickness = 2);
    cylinder := plot( [ [x2 + 2.5 + xu, -1], [x2 + 2.5 + xu, 1], [x2 + 3.5 + 3 * xu, 1], [x2 + 3.5
        + 3 * xu, -1],
    [x2 + 2.5 + xu, -1]], color = tan, thickness = 2);
    piston := x2→rectangle( [x2 + 2.5 + 2 * xu, -1], [x2 + 3 + 2 * xu, 1], color = grey);
    fluid := rectangle( [x2 + 2.5 + xu, -1], [x2 + 3.5 + 3 * xu, 1], color = blue);
    dashpot := x2→display(rod(x2), cylinder, piston(x2), fluid);
    deq := m* diff (x(t), t$2) + r* diff (x(t), t) + k* x(t) = f (t);
    print(deq);
    init := x(0) = x0, D(x)(0) = xp0 :
    sol := dsolve( {deq, init}, x(t) );
    print(combine(simplify(sol) ) );
    xk := unapply(rhs(sol), t);
    #xk=x(t) : the displacement of mass
    pltxk := plot( [xk(t) + x2 + 1, x2 + 1], t = 0 ..max(tk, x2 + 4 + 3 * xu) + 0.5,
    color = [blue, grey], numpoints = 400);
    base := plot( -1.1, t = 0 ..max(tk, x2 + 4 + 3 * xu) + 0.5, color = brown, thickness = 4);
    if r = 0 then #undamped case
    plt1 := x→translate(display(spring1(x2 + x), mass(x2 + x), base), 0, -3);
    else #with damping
    plt1 := x→translate(display(spring1(x2 + x), mass(x2 + x), dashpot(x2 + x), base), 0, -3);
    fi;
    plt2 := i→display( pltxk, rect(tk/n * i) );
    plt := i→display( plt1(xk(tk/n * i) ), plt2(i) );
    display(seq( plt(i), i = 0 ..n), insequence = true, args[11 ..nargs]);
    end:
> impulse_func := proc(m, r, k, a, eps, n)
    local h, dlign1, dlign2, sol1, sol2, plt1, plt2, pltd, plt, txt, txtd, kloss, e;
    with(plots) :
    alias(u = Heaviside) :
    h := eps→(u(t-a) - u(t-a-eps) ) / eps;
    dlign1 := eps→m* diff (y(t), t, t) + r* diff (y(t), t) + k* y(t) = 5 * h(eps);
    dlign2 := m* diff (y(t), t, t) + r* diff (y(t), t) + k* y(t) = 5 * Dirac(t-a);
    print(dlign1(eps), epsilon = eps);
    print(dlign2);
    sol1 := eps→simplify(dsolve( {dlign1(eps), y(0) = 0, D(y)(0) = 0}, y(t) ) );
    sol2 := dsolve( {dlign2, y(0) = 0, D(y)(0) = 0}, y(t) );
    e := textplot( [6, 1.3, convert( [101], bytes) ], font = [SYMBOL, 12]) :
    txt := eps→textplot( [6.9, 1.3, cat( ` = `, convert(evalf (eps, 2), string) ) ]);
    txtd := textplot( [4, 1.4, `Response to Dirac's delta function`], align = {RIGHT, ABOVE});
    pltd := eps→plot(h(eps), t = 0 ..5 * Pi, color = green, thickness = 2) :
    plt1 := eps→plot(rhs(sol1(eps) ), t = 0 ..5 * Pi, color = red, thickness = 2) :
    plt2 := plot(rhs(sol2), t = 0 ..5 * Pi, color = blue, thickness = 2) :
    plt := eps→display( pltd(eps), txt(eps), e, txtd, plt1(eps), plt2);
    display(seq( plt(eps- (eps-0.1) / n * i), i = 0 ..n), insequence = true, args[7 ..nargs]);
    end:
```

Ingrese las constantes de la siguiente manera

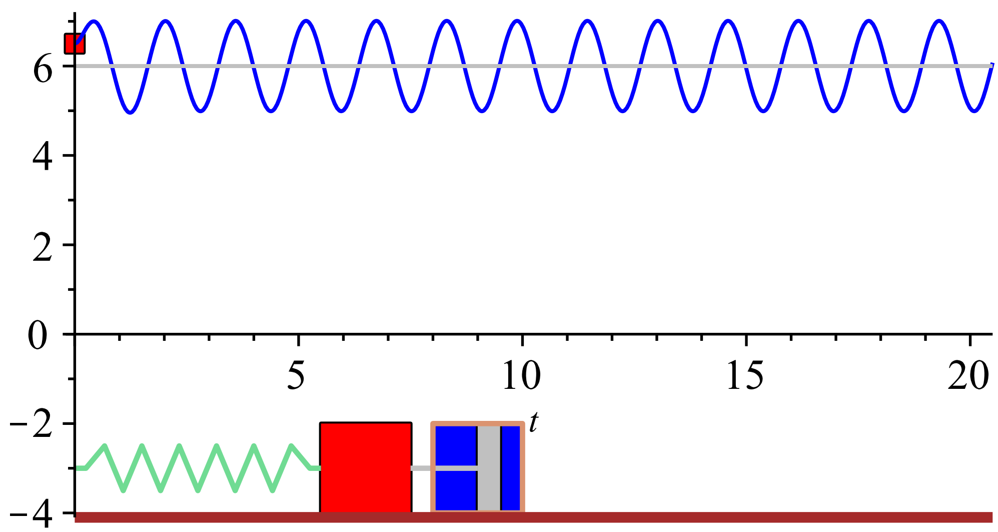
*(m: **masa**, r: **constante de amortiguamiento**, k: **constante elastica del resorte**, x1,x2: **posiciones del resorte** x0 = x(0), xp0=v(0): **condiciones iniciales**,tk:**longitud eje x**, f: **fuerza**, n: **numero de frames**)*

> $f\_ := t \rightarrow 5 \cdot \cos(4 \cdot t)$:

> $mass\_spring\left(\frac{1}{5}, 1.2, 2, 0, 5, \frac{1}{2}, 0, 20, f\_, 60, scaling = constrained\right);$

$$\frac{1}{5}\frac{d^2}{dt^2}x(t) + 1.2\left(\frac{d}{dt}x(t)\right) + 2\,x(t) = 5\cos(4\,t)$$

$$x(t) = -\frac{86}{51}e^{-3t}\sin(t) + \frac{38}{51}e^{-3t}\cos(t) - \frac{25}{102}\cos(4\,t) + \frac{50}{51}\sin(4\,t)$$



>