EMANOEL MESSIAS DE LIMA BARBOSA

27 de março de 2021

1. BIBLIOTECAS

```
from sympy import*
from sympy.plotting import*
```

2. FUNÇÕES e SIMBOLOS FÍSICOS

```
#Funções
x = Function(input('Variável: '))
f = Function('f')
#Simbolos genéricos
P, Q, W, M = symbols('P Q W M')
#Simbolos de entes fisicos
t, g, m, a = symbols('t g m a',real=True, positive=True)
#Simbolos de entes fisicos para osciladores mecânicos
omega, omega1, beta, gamma, Fo, k = symbols('omega omega1 beta gamma Fo k', real = True, positiv
#Simbolos de entes fisicos para osciladores elétricos
R, L, C, epsilon = symbols('R L C epsilon')
print(t,omega, beta, m, omega1, Fo)
#Derivadas de primeira e segunda ordem, respectivamente.
def x1(t):
    return x(t).diff(t)
def x2(t):
    return x1(t).diff(t)
```

$$x1(t) = \frac{dx(t)}{dt}, \quad x2(t) = \frac{d^2x(t)}{dt^2}$$

3. EQUAÇÃO DIFERENCIAL "GERAL"

```
#Diferential Equation
#P*x2(t)+Q*x1(t)+R*x(t)=M
print('Equação Geral de Movimento com Coeficientes Constantes')
pprint(Eq(P*x2(t)+Q*x1(t)+W*x(t),M))
```

$$Eq(P * x2(t) + Q * x1(t) + W * x(t), M) \Rightarrow P\frac{d^2x(t)}{dt^2} + Q\frac{dx(t)}{dt} + Wx(t) = M$$

4. ENTRADA de DADOS para DEFINIÇÃO dos PARÂMETROS

```
print('Defina os parâmetros P, Q, W e M')
P = eval(input('P= '))
Q = eval(input('Q= '))
W = eval(input('W= '))
M = eval(input('M= '))
```

5. **SOLUÇÃO GERAL da EQUAÇÃO DIFERENCIAL de 2º ORDEM com COEFICIENTES CONSTANTES**

```
pprint(Eq(P*x2(t)+Q*x1(t)+W*x(t),M))
solve = dsolve(Eq(P*x2(t)+Q*x1(t)+W*x(t),M))
pprint(solve)
```

dsolve(eq, func = None, hint = 'default',
simplify = True, ics = None, xi = None, eta = None,
x0 = 0, n = 6, ** kwargs)
é a função que resolve a equação diferencial por séries de potência

6. **CONDIÇÕES DE CONTORNO**

```
#Condições iniciais
if P==0:
    ics={x(0):eval(input('x(0)= '))}
else:
    ics={x(0):eval(input('x(0)= ')),x1(t).subs(t,0):eval(input('x1(0)= '))}
```

Para EDO de 1º ordem

```
ics={x(0):eval(input('x(0)= '))}
```

$x(0) = x_0$

Para EDO de 2º ordem

```
ics={x(0):eval(input('x(0)= ')),x1(t).subs(t,0):eval(input('x1(0)= '))}
```

$x(0) = x_0, \quad \frac{dx(0)}{dt} = v_0$

7. **SOLUÇÃO PARTICULAR**

```
#Solução particular
solvepart = dsolve(Eq(P*x2(t)+Q*x1(t)+W*x(t),M),ics=ics)
pprint(solvepart)
```

8. **GRÁFICO DA SOLUÇÃO PARTICULAR**

```
#Graficos
#Definição de valores numéricos para os parâmetros gerais
print('Defina valores numéricos para os parâmetros')
Q = eval(input('Q= '))
W = eval(input('W= '))
M = eval(input('M= '))
solvepart2 = dsolve(Eq(P*x2(t)+Q*x1(t)+W*x(t),M),ics=ics)
print(solvepart2.args[1])
plot(solvepart2.args[1], legend=True, autoescale=True)
```

Quando a EDO é resolvida ela sai em formato de equalit Eq(x(t), Solução), para plotar eu seleciono Eq(x(t),Solução).args[1]=Solução.