Generic formulation of BVP

$$F\left(x, f(x), \frac{df(x)}{dx}, \frac{d^2 f(x)}{dx^2}, \frac{d^3 f(x)}{dx^3}, \ldots, \frac{d^{n-1} f(x)}{dx^{n-1}}\right) = \frac{d^n f(x)}{dx^n},$$

- x in a region [a,b], we need n boundary conditions at value a and b.
- For 2nd order case, we have different cases
  - f(a) and f(b) are given
  - f′(a) and f′(b) are given
  - f(a) and f′(b) are given or f(b) and f′(a) are given

Two-point BVP

Python BVP solver

- scipy.integrate.solve_bvp

solve_bvp(*fun*, *bc*, *x*, *y*, *p*=None, *S*=None, *fun_jac*=None, *bc_jac*=None, *tol*=0.001, *max_nodes*=1000, *verbose*=0, *bc_tol*=None)

- fun: similar to ivp, fun(x,y) or fun(x,y,p)
- bc: boundary conditions
- x: initial mesh
- y: initial guess at the mesh nodes

Example

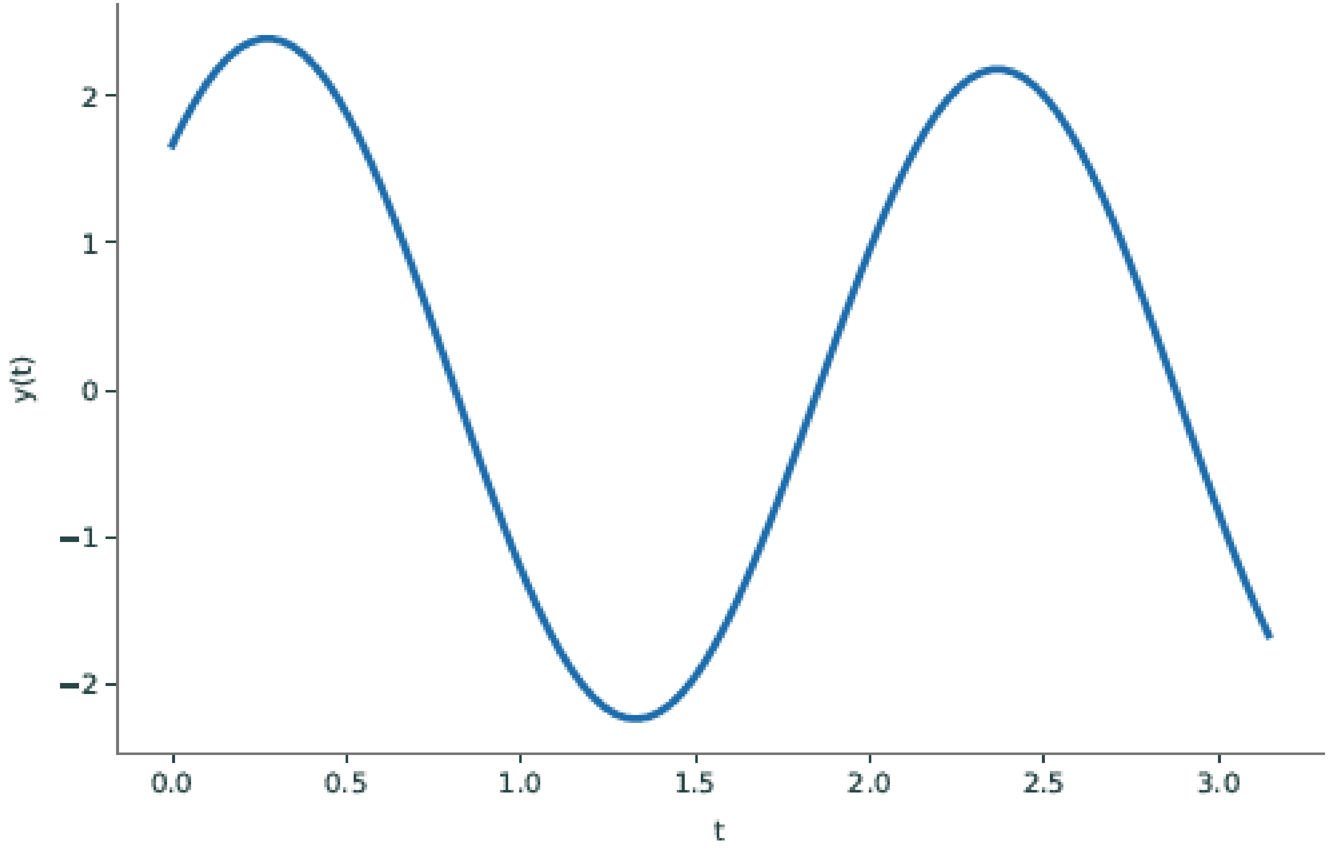
$y'' + 9y = \cos(t),\ y'(0) = 5,\ y(\pi) = -5/3$

S = [y, y'], S' = [y', y'']

```
from scipy.integrate import solve_bvp
import numpy as np
# element 1: the ODE function
def ode(t,S):
    ''' define the ode system '''
        return [S[1], np.cos(t) - 9*S[0]]
# element 2: the boundary condition function
def bc(ya,yb):
    ''' define the boundary conditions '''
    # ya are the initial values, value of S at t0
    # yb are the final values, i.e., value of S at tf
    # the returned array will be set to zero
    return np.array([ya[1] - 5, yb[0] + 5/3])
```

```
# element 3: the time domain.
t_steps = 100
t = np.linspace(0,np.pi,t_steps)
# element 4: the initial guess.
y0 = np.ones((2,t_steps))
# Solve the system.
sol = solve_bvp(ode, bc, t, y0)

import matplotlib.pyplot as plt
# here we plot sol.x instead of sol.t
plt.plot(sol.x, sol.y[0])
plt.xlabel('t')
plt.ylabel('y(t)')
plt.show()
```



- fun remains the same as ivp problem
- Must provide bc: 2 arrays representing initial and final values. bc evaluate to zero.
- Pass a linspace of $[t_0, t_f]$
- Pass an initial guess for all values

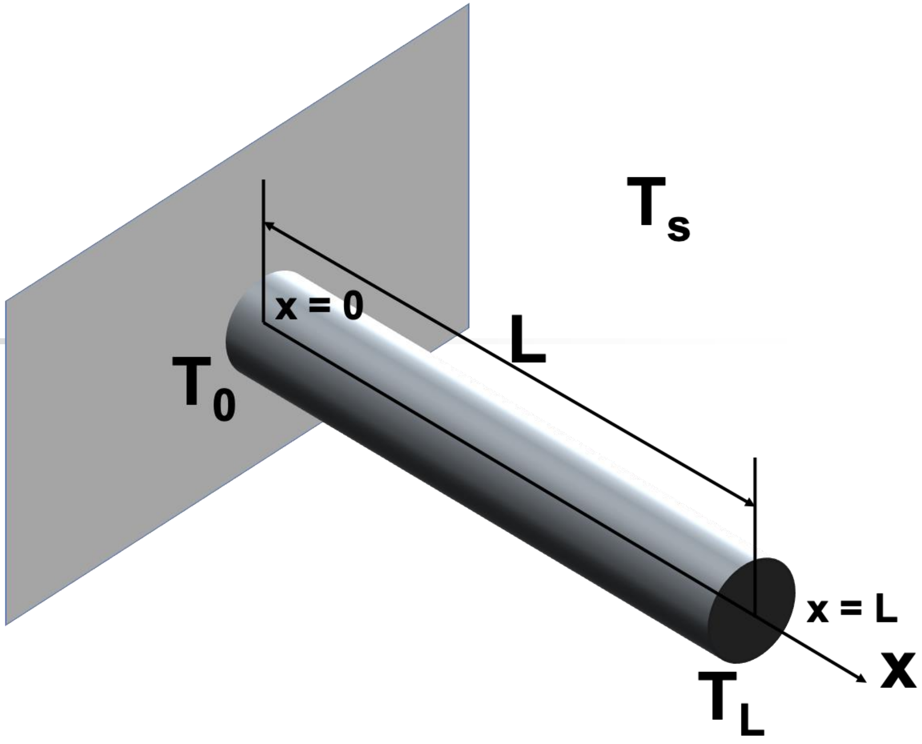
Notes

- sol.sol is a callable function. Plug in any value or numpy array, e.g., sol.sol(np.linspace), sol.sol(float), sol.sol(list).

- Pay attention to the initial values. Small changes can lead to large difference in the final approximations.

- BVP with free parameters can also be addressed.

Example: cooling pin fin



- Steady state temperature distribution T(x)

$$\frac{d^2T}{dx^2} - \alpha_1(T - T_s) - \alpha_2(T^4 - T_s^4) = 0$$

$$\alpha_1 = \frac{h_c P}{k A_c} \qquad \alpha_2 = \frac{\epsilon \sigma_{SB} P}{k A_c}$$

- $L = 0.2$ m
- $T(0)=T_0 = 475$ K
- $T(0.1) = 290$ K
- $T_s = 290$ K
- Determine T(x)

h_c=40 W/m²/K	Convective heat transfer coefficient
P = 0.015 m	Perimeter bounding the cross section
K = 240 W/m/K	Thermal conductivity
A_c=1.55e-5 m²	Cross section area
ε = 0.4	Radiative emissivity
σ_{SB} = 5.67e-8 W / (m²K²)	Stefan-Boltzmann constant