Bisection Method Example Tutorial/Tests

This document provides five example problems related to mechanics that can be solved using the bisection method solver.

**Note: Due to how the code is written, the unknown variable needs to be changed to "w" for each example.*

1. Beam Deflection (Cantilever Beam)

Problem Statement: A cantilever beam of length is subjected to a force at its free end. Find w for given values of F (force),E (Elastic Modulus),I (2nd moment of area),L (length).

Function for Solver:

```
func_str = "(3 * E * I / L**3) * w - F"
variables = {"E": 200e9, "I": 1e-6, "L": 2, "F": 1000}  # Example values
a, b = 0, 0.1  # Example Bounds
tol = 1e-6
```

2. Buckling Load of a Column (Euler's Buckling)

Problem Statement: A pinned-pinned column will buckle under a compressive load . The critical buckling load is given by Euler's formula. Find the smallest L (length) that causes buckling for given P (load),E (Elastic Modulus),I (2nd moment of area).

**Replaces L with w*

Function for Solver:

```
func_str = "(np.pi**2 * E * I / w**2) - P"
variables = {"E": 200e9, "I": 5e-6, "P": 5000}  # Example values
a, b = 1, 10  # Example Bounds
tol = 1e-6
```

3. Natural Frequency of a Simple Pendulum

Problem Statement: A pendulum of length and mass has a natural frequency. Find L (length) such that the frequency is 1 Hz.

**Replaces L with w*

Function for Solver:

```
func_str = "(1 / (2 * np.pi)) * np.sqrt(g / w) - 1"
```

```
variables = {"g": 9.81}  # Gravity
a, b = 0.1, 2  # Example Bounds
tol = 1e-6
```

4. Projectile Motion – Finding Time of Flight

Problem Statement: A projectile is launched at an angle, θ, with initial velocity, v0. Find the time, t, when the projectile returns to the ground.
**Replaces L with w*

Function for Solver:

```
func_str = "v0 * np.sin(theta) * w - 0.5 * g * w**2"
variables = {"v0": 20, "theta": np.pi / 4, "g": 9.81}  # Example values
a, b = 0.1, 5  # Example Bounds
tol = 1e-6
```

5. Torsional Vibration of a Shaft

Problem Statement: A shaft undergoing torsional vibration has a natural frequency, omega, defined by the torsional stiffness, K, and the polar moment of inertia, J. Find the polar moment of inertia, J, for a given frequency, omega.
**Replaces J with w*

Function for Solver:

```
func_str = "np.sqrt(K / w) - omega"
variables = {"K": 1000, "omega": 50}  # Example values
a, b = 0.01, 1  # Example Bounds
tol = 1e-6
```