Example

The latest bit of code used is developed from `MaterialProperties`.

Using the developed interface here is some code for running a simulation of a cantilever rod subject to gravity and without viscosity.

```python
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from Simulations.Loads import Gravity, PointLoadBody, PointLoadFixed
import numpy as np
from Simulations.Rod import Rod, Series
from Simulations.Body import SimpleBody, FirstOrderMaterial, Cylinder


if __name__ == "__main__":

    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.view_init(elev=0, azim=-90)
    plt.axis('equal')
    dt = 0.01
    steps = 100
    N = 100
    loads = [Gravity(np.array([9.81, 0, 0]))]

    mat = FirstOrderMaterial(1e6, 1e6/3, 1e3, 0)
    cyl = Cylinder(1e-2, 10e-2)

    body = SimpleBody(mat, cyl)

    rod = Rod(body, N, xi_init=lambda s: np.array([0, 0, 0, 0, 0, 1]), loads=loads)

    ax = rod.plot(ax)
    plt.pause(0.01)
    for i in range(steps):
        print(i, "/", steps)
        rod.step(dt, np.array([]))
        ax = rod.plot(ax)
        plt.pause(0.01)
```

```
        plt.show()
```

Then an example where a 10g load has been attached to the center of the rod.

```python
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from Simulations.Loads import Gravity, PointLoadBody, PointLoadFixed
import numpy as np
from Simulations.Rod import RodFixedFree, Series

if __name__ == "__main__":

    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.view_init(elev=0, azim=-90)
    plt.axis('equal')
    dt = 0.01
    steps = 100
    N = 100
    l1 = [PointLoadFixed(np.array([0, 0, 0, 10e-3 * 9.81, 0, 0])), Gravity(np.array([9.81, 0
    l2 = [Gravity(np.array([9.81, 0, 0]))]

    mat = FirstOrderMaterial(1e6, 1e6/3, 1e3, 0)
    cyl = Cylinder(1e-2, 10e-2/2)

    body1 = SimpleBody(mat, cyl)
    body2 = SimpleBody(mat, cyl)

    r1 = Rod(body1, N // 2, xi_init=lambda s: np.array([0, 0, 0, 0, 0, 1]), loads=l1)
    r2 = Rod(body2, N // 2, xi_init=lambda s: np.array([0, 0, 0, 0, 0, 1]), loads=l2)


    rod = Series([r1, r2])
    ax = rod.plot(ax)
    plt.pause(0.01)
    for i in range(steps):
        print(i, "/", steps)
        rod.step(dt, np.array([]))
        ax = rod.plot(ax)
        plt.pause(0.01)
    plt.show()
```