3.1.5 Finger Exercise: Why did we deepcopy when initializing an object?

Bookmark this page

Finger Exercises 1 due Aug 3, 2023 05:00 IST Completed

MO2.2

In Section 8.4.2, we implemented our initial Python IVP class. In the `__init__` method, we chose to use a `deepcopy` when initializing the IVP object's data attribute `_p`. Similarly, we did a copy of the initial condition list member when initializing the object's data attribute `_uI`. In this Finger Exercise, we explore why that choice was made.

Recall the coffee IVP object example that was discussed in Section 8.4.3. Consider running the following code using that coffee IVP object to calculate the coffee temperature rates of change on a hot day, a cool day, and a hot day again:

```
# Create an IVP object for coffee on a hot day
p = {}
p['mc'] = 0.35 # kg
p['cc'] = 4200.0 # J / (kg C)
p['h']  = 5.0 # W/(m^2 C)
p['A']  = 0.04 # m^2
p['Tout'] = 25.0 # C

TcI = 85.0 # Initial temperature of coffee (C)
tImin = 0.0 # initial time to start simulation (min)
tFmin = 700.0 # final time to simulate to (min)
tI = tImin*60 # convert time to seconds
tF = tFmin*60

# Create an IVP object for coffee on a hot day
coffeeIVP_hotday = IVP([TcI], tI, tF, p, coffee_evalf)

# Calculate dTc/dt on a hot day with Tc=40.0 C
Tc = 40.0
dTcdt_hotday = coffeeIVP_hotday.evalf([Tc],0.)
print(f"On a hot day:        dTc/dt = {dTcdt_hotday[0]}")

# Create an IVP object for coffee on a cool day
p['Tout'] = 5.0 # C

coffeeIVP_coolday = IVP([TcI], tI, tF, p, coffee_evalf)

# Calculate dTc/dt on a cool day with Tc=40.0 C
dTcdt_coolday = coffeeIVP_coolday.evalf([Tc],0.)
print(f"On a cool day:       dTc/dt = {dTcdt_coolday[0]}")

# Calculate dTc/dt on a hot day with Tc=40.0 C one more time
dTcdt_hotday_again = coffeeIVP_hotday.evalf([Tc],0.)
print(f"On a hot day again:  dTc/dt = {dTcdt_hotday_again[0]}")
```

When that code is run with the original `__init__` implementation, the following is printed:

```
On a hot day:        dTc/dt = -0.002040816326530612
On a cool day:       dTc/dt = -0.0047619047619047615
On a hot day again:  dTc/dt = -0.002040816326530612
```

Suppose we consider the following alternative implementation of the `__init__` method in which we set `self._p = p`. Here's the new version of the `__init__` method:

```
class IVP():
    def __init__(self, uI, tI, tF, p, f):
        """
        Args:
            uI (float list): initial condition of state.
            tI (float): initial time.
            tF (float): final time.
            p (dictionary): set of fixed parameters.
            f (function): takes as input u,t,p and returns du/dt
        """

        self._uI = uI[:]
        self._tI = tI
        self._tF = tF
        self._p  = p # THIS IS THE ONLY DIFFERENCE
        self._f  = f
        self._M  = len(uI)
```

What will be printed with this new implementation of `__init__` when the coffee temperature rates of change are calculated? Your choices include:

```
Option A:
On a hot day:        dTc/dt = -0.002040816326530612
On a cool day:       dTc/dt = -0.0047619047619047615
On a hot day again:  dTc/dt = -0.002040816326530612

Option B:
On a hot day:        dTc/dt = -0.002040816326530612
On a cool day:       dTc/dt = -0.0047619047619047615
On a hot day again:  dTc/dt = -0.0047619047619047615
```

Problem: What will be printed?

1/1 point (graded)

- ○ Option A is printed
- ◉ Option B is printed ✔

Submit

ℹ Answers are displayed within the problem

SOLUTION: The solution will be available shortly after the due date in Section 3.2.5.

‹ Previous Next ›

edX

About
Affiliates
edX for Business
Open edX
Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap
Cookie Policy
Your Privacy Choices

Connect

Blog
Contact Us
Help Center
Security
Media Kit