‹ Previous ✓ Next ›

8.4.2 Designing an IVP Python class

Bookmark this page

MO2.2 **MO2.4**

Our goal is to write Python code to solve IVP as described in Section 8.3.1. To do this, we will design and implement our own Python objects. Recall that an object is defined as having:

- **Data attributes**: information needed to describe a specific instance of the object.
- **Methods (procedural attributes)**: functions that interact with the object

In Python, objects are implemented using a Python class. Before we show the actual Python code that has our IVP class definition, let's first determine what data attributes and methods will be needed for our IVP object. We will keep this a very simple IVP class for now, and then continue to expand it as we move through the rest of this course.

First, let's consider what attributes are needed to describe an IVP. From Section 8.3.1, we see that the definition of an IVP includes t_I, t_F, $\underline{u}_I$, $\underline{f}\left(\underline{u}, t\right)$, and M. In addition, while not given explicitly stated in Section 8.3.1, we know that all of our IVP examples also contain parameters in $\underline{f}$. So, we'll simply refer to those parameters as $\underline{p}$ and we will modify the functional dependence of $\underline{f}$ to be $\underline{f}\left(\underline{u}, t, \underline{p}\right)$. Thus, in summary, we have the following attributes that define an IVP:

- t_I: the initial time
- t_F: the final time
- $\underline{u}_I$: the initial condition
- $\underline{p}$: parameters
- $\underline{f}\left(\underline{u}, t, \underline{p}\right)$: the time rate of change of $\underline{u}$ as a function of $\underline{u}$, t, and $\underline{p}$.
- M: the number of states (i.e. the size of $\underline{u}$)

Next, let's think about which of the above are data attributes and which are functional attributes (methods). t_I, t_F, $\underline{u}_I$, $\underline{p}$, and M are data. However $\underline{f}$ is a function of $\underline{u}$, t, and $\underline{p}$ and so this will be a method.

An actual Python class which defines an IVP object is

Discussions

All posts sorted by recent activity

An actual Python class which defines an IVP object is given below and is also available for download here.

```
##############################################
##################################
# IntroToCSE
# IVPlib_rev0
#

"""
This Python library is useful in solving
Initial Value Problems (IVP).
```

< Previous

Next >

```
s
an IVP in the form:
```

© All Rights Reserved

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](#)

GET IT ON Google play

Download on the App Store

described in the video, we believe there are better implementations for this IVP class. We will explore this in the shortly in Section 8.5 when we introduce the