Lesson 0: Preparing Your Computer for the Course

Aaron A. King   Edward L. Ionides   Kunyang He

2025-12-25

Table of contents I

Table of contents II

Table of contents III

Section 1

Introduction

Introduction

This document provides instructions for setting up your computer to run the exercises in this course. We will be using **pypomp**, a Python implementation of partially observed Markov process (POMP) models.

Subsection 1

What you will need

What you will need

- Python 3.8 or higher
- pip or conda package manager
- A text editor or IDE (e.g., VS Code, PyCharm, Jupyter)
- Basic familiarity with Python programming

Section 2

Python Installation

Subsection 1

Checking your Python version

Checking your Python version

First, check if you have Python installed and verify the version:

```
python --version
```

or

```
python3 --version
```

You should have Python 3.8 or higher. If not, proceed to install Python.

Subsection 2

Installing Python

Option 1: Using Anaconda (Recommended for beginners)

1. Download Anaconda from https://www.anaconda.com/download
2. Follow the installation instructions for your operating system
3. Anaconda includes Python and many scientific packages

Option 2: Using official Python distribution

1. Download Python from https://www.python.org/downloads/
2. Follow the installation instructions for your operating system
3. Make sure to check "Add Python to PATH" during installation (Windows)

Section 3

Setting Up a Virtual Environment

Setting Up a Virtual Environment

It is highly recommended to use a virtual environment to avoid conflicts with other Python projects.

Subsection 1

Using venv (built-in)

Using venv (built-in)

```
# Create a virtual environment
python -m venv sbied-env

# Activate the environment
# On macOS/Linux:
source sbied-env/bin/activate

# On Windows:
sbied-env\Scripts\activate
```

Subsection 2

Using conda

Using conda

```
# Create a conda environment
conda create -n sbied-env python=3.10

# Activate the environment
conda activate sbied-env
```

Section 4

Installing Required Packages

Subsection 1

Installing pypomp

Installing pypomp

The main package we will use is **pypomp**. Install it using pip:

```
pip install pypomp
```

Subsection 2

Installing additional dependencies

Installing additional dependencies

Install the following packages that are commonly used in the course:

```
pip install numpy scipy matplotlib pandas jupyter
```

If you are using conda, you can install these packages using:

```
conda install numpy scipy matplotlib pandas jupyter
```

Subsection 3

Optional: GPU acceleration with JAX

Optional: GPU acceleration with JAX

For faster computation, especially for large models, you can install JAX for GPU/TPU acceleration:

```
# For CPU-only (default)
pip install jax jaxlib

# For GPU support (CUDA)
# See https://github.com/google/jax#installation for detailed
pip install jax[cuda12]
```

Subsection 4

Optional: Additional plotting libraries

Optional: Additional plotting libraries

For enhanced visualizations:

```
pip install seaborn plotly
```

Section 5

Setting Up Jupyter

Setting Up Jupyter

Jupyter notebooks provide an interactive environment for running Python code.

Subsection 1

Installing Jupyter

Installing Jupyter

If you haven't already installed Jupyter:

```
pip install jupyter notebook jupyterlab
```

Subsection 2

Launching Jupyter

Launching Jupyter

To start Jupyter Notebook:

```
jupyter notebook
```

To start JupyterLab (more modern interface):

```
jupyter lab
```

Subsection 3

Registering your virtual environment as a Jupyter kernel

Registering your virtual environment as a Jupyter kernel

If you created a virtual environment, register it with Jupyter:

```
# Make sure your virtual environment is activated
pip install ipykernel
python -m ipykernel install --user --name=sbied-env --display-
```

Section 6

Verifying Your Installation

Subsection 1

Test basic Python functionality

Test basic Python functionality

Create a test Python script or run in a Python interpreter:

```python
import sys
print(f"Python version: {sys.version}")

import numpy as np
import scipy
import matplotlib.pyplot as plt
import pandas as pd

print("All basic packages imported successfully!")
```

Subsection 2

Test pypomp installation

Test pypomp installation

Create a simple test to verify pypomp is working:

```python
import pypomp

print(f"pypomp version: {pypomp.__version__}")
print("pypomp installed successfully!")
```

Subsection 3

Test with a minimal example

Test with a minimal example

Run this minimal POMP model to ensure everything is working:

```python
import numpy as np
import pypomp

# Create a simple random walk model
def step_fn(x, t, params, **kwargs):
    return {'X': x['X'] + np.random.normal(0, params['sigma'])

def rinit_fn(params, **kwargs):
    return {'X': 0.0}

def rmeas_fn(x, t, params, **kwargs):
    return {'Y': x['X'] + np.random.normal(0, params['tau'])}

def dmeas_fn(y, x, t, params, log=True, **kwargs):
    ll = -0.5 * np.log(2 * np.pi * params['tau']**2) - \
```

Section 7

Downloading Course Materials

Subsection 1

Using Git (recommended)

Using Git (recommended)

If you have Git installed:

```
git clone https://github.com/[repository-url]/sbied-pypomp.git
cd sbied-pypomp
```

Subsection 2

Direct download

Direct download

Alternatively, download the course materials as a ZIP file from the course website and extract them to your working directory.

Section 8

Troubleshooting

Subsection 1

Common issues and solutions

"pip: command not found"

- Make sure Python is installed and added to your PATH
- Try using `python -m pip` instead of `pip`

"Permission denied" errors

- On macOS/Linux, try using `pip install --user package_name`
- Or use a virtual environment (recommended)

Import errors

- Make sure your virtual environment is activated
- Verify package installation: `pip list | grep package_name`
- Try reinstalling: `pip uninstall package_name && pip install package_name`

Jupyter kernel issues

- Make sure ipykernel is installed in your environment
- Re-register the kernel (see "Registering your virtual environment" above)
- Restart Jupyter after adding a new kernel

Subsection 2

Getting help

Getting help

If you encounter issues:

1. Check the pypomp documentation:
   https://github.com/pypomp/pypomp
2. Search for error messages online
3. Ask for help during the course sessions
4. Check the course discussion forum

Section 9

Summary

Summary

By now, you should have:

- Python 3.8+ installed
- A virtual environment created and activated
- pypomp and required packages installed
- Jupyter set up (if using notebooks)
- Verified your installation with test code

You are now ready to begin the course!

Section 10

Additional Resources

Additional Resources

- **Python Tutorial**: https://docs.python.org/3/tutorial/
- **NumPy Documentation**: https://numpy.org/doc/
- **Matplotlib Tutorial**:
  https://matplotlib.org/stable/tutorials/index.html
- **Jupyter Documentation**: https://jupyter.org/documentation
- **pypomp Repository**: https://github.com/pypomp/pypomp