

Data-Driven Design & Analysis of Structures & Materials (3dasm)

Instructor: Miguel A. Bessa

Brown University

Homework 1

Deliver a **short PDF report** of this assignment containing the answers to the questions listed here. **UPLOAD to CANVAS in the Assignments section (Homework 1) by the due date.**

Due date: until 11:59pm of day announced in [course home page](#).

Installing Python Packages

1. Follow the instructions of Lecture 1 to install locally in your computer the necessary packages for this course. (This could be the easiest or one the most frustrating homework questions of the entire course!)

- 1.1. Decide whether you want to install Mamba or Anaconda. Mamba is the same as Anaconda, but Mamba is much faster. I recommend Mamba. However, if you already have Anaconda installed and don't want to uninstall it, then do not install Mamba. The only difference when using Mamba instead of Anaconda is that every time you install a Python package with Mamba you use the `mamba` command, while with Anaconda you use the `conda` command. **Again, do not install both packages at the same time** – that is a bad idea. Windows users tend to prefer using Anaconda instead of Mamba – see footnote for instructions.¹

- 1.2. Install Mamba as described [here](#) (again: if you prefer, you can use or install Miniconda/Anaconda instead). When asked a yes/no question, type always yes. If you are using **Linux** based system then you can install Mamba with just two command lines:

```
wget "https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-$(uname)=$(uname -m).sh"
bash Miniforge3-$(uname)=$(uname -m).sh
```

- 1.3. Install Jupyter notebook and extensions in base environment, and create a virtual environment for this course called **3dasm**:

```
mamba create -n 3dasm python==3.11 notebook nb_conda rise numpy scipy matplotlib pandas scikit-learn ipykernel ipywidgets
mamba activate 3dasm

pip install torch torchvision --index-url https://download.pytorch.org/whl/cu126

pip install f3dasm==2.0.2

pip install optuna
```

- 1.4. Install [git](#), open command window & clone the course repository to your computer (This will copy a folder called **3dasm_course**. You can save this folder anywhere in your computer):

```
git clone https://github.com/bessagroup/3dasm_course
```

- 1.5. Open a (mamba) command window, activate the **3dasm** environment and start jupyter notebook (it will open in your internet browser):

```
mamba activate 3dasm
jupyter notebook
```

¹Instructions for **Windows users** that prefer using Anaconda instead of Mamba: run everything in the Anaconda command prompt in **administrator mode**. If Anaconda is taking too much time to run the commands listed in 1.3, you can also install packages individually (one by one).

```
python -m ipykernel install -name=3dasm
conda create -n 3dasm python==3.11 notebook nb_conda rise numpy scipy matplotlib pandas scikit-learn ipykernel
ipywidgets
conda activate 3dasm
pip install torch torchvision --index-url https://download.pytorch.org/whl/cu126
pip install f3dasm==2.0.2
pip install optuna
```

- 1.6. Open a notebook (3dasm_course/Lectures/Lecture01/3dasm_Lecture1.ipynb) and run it. Make sure you do not get any errors.
2. **Note** about installing Pytorch such that it uses your GPU. Depending on the Operating System you use and your Hardware, it can be non-trivial to install Pytorch such that it uses your GPU (which makes training neural networks faster). **For this course, it is ok if you are only able to run Pytorch with your CPU because you won't need to train large models.** Nevertheless, here are a few tips to make sure that your Pytorch installation recognizes and uses your GPU.
 - 2.1. In general, it is recommended to use a recent driver for your GPU. However, sometimes the latest driver is less stable than the previous one.
 - 2.2. For example, I use a Linux operating system (I am currently using **Linux Ubuntu 22.04 LTS**). Installing the appropriate driver in this system for my NVIDIA GPU is very easy:
 - i. Click on the "Show applications" button and type "Additional drivers".
 - ii. In the window that opens, click on the option: "Using NVIDIA driver metapackage from nvidia-driver-550 (proprietary, tested)". Apply this option. Note: interestingly, my system was not stable when I used a more recent driver than this one.
 - iii. Reboot your computer.

Create your first Jupyter notebook for your Homework!

3. Now that everything is installed, create your first Jupyter notebook to solve this homework. The following instructions will help you to create it.
 - 3.1. Start by creating a folder called `your_Assignments` inside the folder: `3dasm_course/Assignments`. (Remember: the folder `3dasm_course` is located where you decided to clone the `3dasm_course` GitHub repository to your computer).
 - 3.2. Now that you have created the empty folder `3dasm_course/Assignments/your_Assignments` you can open a command window, activate the `3dasm` environment, and start a jupyter notebook (it will open in your internet browser):


```
mamba activate 3dasm

jupyter notebook
```
 - 3.3. Navigate to the folder you created `3dasm_course/Assignments/your_Assignments`
 - 3.4. Then click in the button on the upper left part of the screen: **New > Python [conda env:3dasm]**
 - 3.5. This opens an empty Jupyter Notebook with title "Untitled". Rename it to "your_Homework1" by clicking on **File > Rename**.
 - 3.6. You are ready to start! Take some time to familiarize yourself with the Jupyter Notebook.
 - 3.7. You will use mostly two types of Cells: (1) Markdown cells where you can write text; and (2) Code cells where you can write code. The Lectures were also written with these 2 types of cells. To convert a cell to a Markdown cell, click on the left part of the cell and hit "m" in your keyboard. If you want to create a new cell below, hit "b". If you want to convert a Markdown cell into a Code cell, hit "y". Search for simple tutorials about Jupyter notebooks and you will get used to it very quickly. We are also here to help!
 - 3.8. In your new and empty Jupyter notebook, convert the first empty Cell into a Markdown cell, and write in it: `# Solution to Homework 1 by (your_name)`. Hit Shift+Enter. This creates text with a large font (to create your title).
 - 3.9. The next cell should be a code cell (again, hit "b" after selecting your first Markdown cell to create a new cell below). You can see that this new cell is a Code cell when it says 'In []:' behind it (left part of the cell). In this and other Code cells you can write the code to solve your homework. When you hit "Shift+Enter" you run the code cell to see the output of the code.
4. Create a figure with the following two subplots: (1) the Gaussian Cumulative Distribution Function (CDF); and (2) the Probability Distribution Function (PDF) for the car stopping distance problem considered in

Lecture 1. **You can find the figure and the respective code in Cell 6 of 3dasm_Lecture1.ipynb.** Do not forget to import the correct packages for plotting and from scipy.stats (just like it was done in the Lecture)... In addition, this time consider the following information to obtain the distributions:

4.1. $z \sim \mathcal{N}(\mu_z = 3, \sigma_z^2 = 1.2^2)$.

4.2. Show the probability and probability density values, respectively in the appropriate subplot, when $z = 1.45$ seconds. As in the lecture, use a red star marker to highlight this point.

4.3. Plot both subplots within the domain $z \in [0, 6]$.

4.4. Adjust the legends of the subplots accordingly to the new values considered herein.

5. As a final task, include in the Jupyter notebook the simple CPU and GPU checks of Pytorch.

5.1. Create a new Markdown cell below the cell with the Gaussian cdf and pdf. Write the following text in that cell: "Check that Pytorch is running properly."

5.2. Then create a Code cell below the previous cell.

5.3. Write the following code in that Code cell:

```
import torch
x = torch.rand(5, 3)
print(x)
```

5.4. Run the cell. It should output a tensor with 5 rows and 3 columns and some random numbers.

5.5. Create another Code cell below the previous code cell, and write:

```
torch.cuda.is_available()
```

5.6. Run the cell. It should output "True", which means that Pytorch is can use your GPU. If the output is false, you can only use Pytorch with your CPU.

After you run the entire notebook (click on Kernel > Restart & Run All), create a PDF (File > Print Preview and then print PDF). **UPLOAD the PDF to CANVAS in the Assignments section (Homework 1) by the due date (until 11:59pm).**