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

Intructor: Miguel A. Bessa and **TA**: Martin van der Schelling

Brown University

Homework 1

Deliver **a short PDF report** of this assignment containing the answers to the questions listed here.

Due date: September 13, 2023

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, you are welcome to keep Anaconda. The only difference is that every time you install a Python package with Mamba you use the `mamba` command, while if you use Anaconda you use the `conda` command. **However, do not install both packages at the same time** – that is a bad idea.

   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/Mambaforge-$(uname)-$(uname -m).sh"
   bash Mambaforge-$(uname)-$(uname -m).sh
   ```

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

   ```
   mamba install -c anaconda notebook nb_conda rise ipywidgets
   mamba create -n 3dasm -c anaconda python=3 numpy scipy matplotlib pandas scikit-learn
   ```

   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 and load jupyter notebook (it will open in your internet browser):

   ```
   jupyter notebook
   ```

   1.6. Open notebook (3dasm_course/Lectures/Lecture1/3dasm_Lecture1.ipynb)

   1.7. Choose the `3dasm` kernel by going to Kernel > Change Kernel > Python [conda env:3dasm]. Note: if your computer does not recognize the `3dasm` virtual environment it should be related to the installation of the `nb_conda` package. One way to get around this is to install all the packages in 1.3. inside the `3dasm` virtual environment.

   1.8. Run the 3dasm_Lecture1.ipynb notebook and see if you do not get any errors.

2. Install Tensorflow. Depending on the Operating System you use and your Hardware, it can be non-trivial to install Tensorflow such that it uses your GPU (which would make training neural networks much faster). In order to help, we share below the installation commands that worked for a system with **Linux Ubuntu 22.04 LTS** and a **recent NVIDIA GPU**. (If you have a Mac with M1 GPU go here)

   2.1. Make sure you have recent NVIDIA drivers installed and running. In Linux, hit the "Show applications" button and type "Additional drivers".

   2.2. In the window that opened, click on the option: "Using NVIDIA driver metapackage from nvidia-driver-535 (proprietary, tested)". If your driver number is different, choose the most recent one that mentions "(proprietary, tested)".

2.3. Reboot your computer.

2.4. Now, follow the instructions to install Tensorflow here. Everywhere in that website where you see `conda` replace by `mamba`. Do not forget to install Tensorflow in the `3dasm` virtual environment. For your convenience, the commands used are included below but it's **better to type the commands or copy them directly from the Tensorflow page with the appropriate changes** (here some of the commands need to go to the next line):

```
mamba activate 3dasm

mamba install cudatoolkit=11.8.0

pip install nvidia-cudnn-cu11==8.6.0.163

python3 -m pip install nvidia-cudnn-cu11==8.6.0.163 tensorflow==2.13.*

mkdir -p $CONDA_PREFIX/etc/conda/activate.d

echo 'CUDNN_PATH=$(dirname $(python -c "import nvidia.cudnn;print(nvidia.cudnn.__file__)"))' >> $CONDA_PREFIX/etc/conda/activate.d/env_vars.sh

echo 'export LD_LIBRARY_PATH=$CONDA_PREFIX/lib/:$CUDNN_PATH/lib:$LD_LIBRARY_PATH' >> $CONDA_PREFIX/etc/conda/activate.d/env_vars.sh

pip install --upgrade pip

pip install tensorflow==2.13.*
```

2.5. That's it. Now run the following command to verify the CPU setup to see if it returns a tensor (in that case you installed Tensorflow CPU successfully):

```
python3 -c "import tensorflow as tf; print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
```

2.6. Also run the following command to verify the GPU setup to see if it detects your GPU (in that case you also installed Tensorflow GPU successfully):

```
python3 -c "import tensorflow as tf; print(tf.config.list_physical_devices('GPU'))"
```

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 and load jupyter notebook (it will open in your internet browser):

```
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 Tensorflow.

   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 Tensorflow 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 tensorflow as tf

print(tf.reduce_sum(tf.random.normal([1000, 1000])))
```

   5.4. Run the cell. Do not worry if you see some red warning messages. You can get rid of them if you run the cell again. Otherwise, just ignore them. Tensorflow outputs several warnings when it's run for the first time (NUMA node, lack of TensorRT, etc.). What matters is the actual output of the code.

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

```
print(tf.config.list_physical_devices('GPU'))
```

   5.6. Run the cell.

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).**