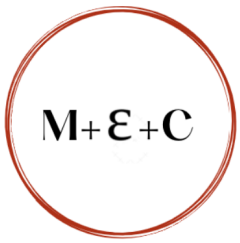

Math+Econ+Code Masterclass

January 2022

Optimal Transport and Economic Applications

Setting up Gurobi and Docker[1]

1 Getting started with Gurobi

In this class we are going to exploit the power behind the commercial-level optimizer Gurobi. Gurobi provides a fast and powerful mathematical programming solver available for a number of problems, including constrained linear and quadratic programming problems. It allows us to quickly work out the solution to (some of) the optimal transport programs we are going to see in this class.

While Gurobi is commercially available, academics and students have access to it for free through an academic license — keep this in mind for your future work. For our present purposes, Gurobi has recently launched a different kind of license for containers such as those running on Docker, called Web License Service (WLS). This WLS academic license is what you will need to run the code in the class container. As long as you correctly reference the license when launching the container and remain connected to the internet, Gurobi will work smoothly without the need to install any other tool. Here are the steps you need to follow in order to get your license.

1.1 Obtaining your license

To create a WLS license:

1. Register for a gurobi.com account as an academic. You can do this starting at this link, making sure to use your institutional email address.

[1]This tutorial is based on the work of Jules Baudet, Gabriele Buontempo and Giovanni Montanari for the 'math+econ+code' masterclass. Support from the ERC CoG-866274 EQUIPRICE grant is acknowledged.

2. Note you must be connected to a recognized academic institution network when the license request is made. If you are currently working from home, you need to use a VPN and connect to your university network. For NYU students, you can get started about using a VPN at this link, looking at the "Top support links" for your operating system. You will have to download the Cisco Anyconnect Secure Mobility Client and login with your credentials to access the "NYU VPN: All Traffic" network through the VPN.

3. Once connected to the university network, you are ready to request a license. Visit the Web License Manager site, where you can create (and in the future, extend) the WLS academic license.

4. At the end of the process, download your license file, which by default will be called "gurobi.lic", and place it in a directory of known path of your choice. In my case, this is

   ```
   /Users/alfredgalichon/Desktop/gurobi-wls-license
   ```

2 What is Docker ?

Docker is a technology that allows developers to:

1. deliver softwares in bundles called "**containers**",

2. make their application run smoothly on different operative systems,

3. save a significant amount of computing resources.

To do so, Docker uses **containers** and **images**.

2.1 Docker containers

A **container** is a standard unit of software. It is a "package" containing an application and all of its dependencies (libraries, settings, etc.) allowing it to run reliably on different environments.

Containers allow to isolate applications from the computer environment. They run on *Docker Engine*, a runtime compatible with most Linux and Windows-based systems.

This tutorial will allow you to download a container gathering all the material for the course (Jupyter notebooks) and the dependencies needed to run them. **You will thus have everything you need in one place!**

2.2 Docker images

A **Docker image** is an immutable file containing the source code, libraries, dependencies, etc. needed for the application to run. It can be built (i.e. created) through the command **docker build**, or downloaded from an online storage (e.g. DockerHub) through the command **docker pull**. We will use the latter option.

An image is a **read-only file**, whereas the container is a virtualized **run-time environment** where your application will actually be running. Launching a container is done with the command **docker run**.

For more details on Docker, you may read: https://docs.docker.com/get-started/overview/.

3 Using Docker for the M+E+C Masterclass

3.1 Installing Docker

To download the course container, you will first need to install Docker. Follow the instructions corresponding to your machine:

- For Mac: https://docs.docker.com/docker-for-mac/install/,
- For Windows: https://docs.docker.com/docker-for-windows/wsl/ (Please make sure that you meet the prerequisites! In particular, you need to install WSL 2 https://docs.microsoft.com/en-us/windows/wsl/install-win10).

Once you have installed Docker, open the Docker Desktop application.

3.2 Downloading the image

You are now ready to download the **image** of the course.

1. Setup a work directory on your local hard drive (avoid synced folders). For me, that is:

   ```
   /Users/alfred/Desktop/mec_jan
   ```

2. Open the shell, navigate to folder created for the course. For me, that is:

   ```
   cd /Users/alfred/Desktop/mec_jan
   ```

3. Download the course image from docker hub with the following command:

   ```
   docker pull alfredgalichon/mec_optim:2022-01
   ```

The last instruction will start the process of downloading the image (which may take a while). Make sure you download the image before the beginning of the first lecture, or you won't have enough time to follow live.

3.3 Running the container

Now that you have installed the image, you are ready to run the container. To do so, I use the following command in the shell:

```
docker run -it --rm -p 8888:8888 -v /Users/alfred/Desktop/
    mec_jan://src/notebooks/my-work -v /Users/alfred/Desktop/
    gurobi-wls-license/gurobi.lic:/opt/gurobi/gurobi.lic:ro
    alfredgalichon/mec_optim:2022-01
```

NOTE: you have to use the command above with **the path to the work directory in your hard drive**, i.e. you have to replace

```
/Users/alfred/Desktop/mec_jan
```

with the path to the folder you created for the course on your machine.

The flag -v ADDRESS 1:ADDRESS 2 creates a *volume*, i.e. a folder shared by your local system at the address ADDRESS 1 and your container at ADDRESS 2. It will allow you to send files from your computer to your container and vice versa.[2]

3.4 Opening Jupyter

After running the command docker run, your terminal should print a few URLs. In my case, these lines are as follows:

```
To access the notebook, open this file in a browser:
    file:///root/.local/share/jupyter/runtime/nbserver-1-
        open.html
Or copy and paste one of these URLs:
    http://6f1c393b5767:8888/?token=
        c2560f08864c8fd878563438068be123e541c25e5552ffcb
 or http://127.0.0.1:8888/?token=
    c2560f08864c8fd878563438068be123e541c25e5552ffcb
```

Copy one of the URLs displayed at the bottom in a browser. You should now have access to Jupyter running in your container. You will see that the notebook directory includes two folders: the class folder, and the previously created *my-work* folder, which is linking the container path to your work folder. It provides a space to save your work to your physical machine and avoid losses upon exiting the container.

While running your Docker container, make sure not to have any other notebook running locally on your computer or you might have troubles with the token authentication.

4 Good practices

Throughout the class, we advise you to work as follows:

- Whenever you want to work on a notebook, navigate to your work directory and use the **docker run** command presented in section 3.3.

[2] You can find out the meaning of the other flags -p, -it, etc. here: https://docs.docker.com/engine/reference/run/.

- When you are done working on a notebook, save your modifications and make sure to copy this notebook in your volume! **Otherwise, when you will close your container, all your progress will be lost!**

You now have everything you need to start the class!

Please make sure you go through the content of this file before the beginning of the class, so that you are up and running by then.