

Gurobi Seminar 1

Installation and Introduction

Demetrios Papazaharias

UB INFORMS Student Chapter
Department of Industrial & Systems Engineering
University at Buffalo, SUNY

September 13, 2019



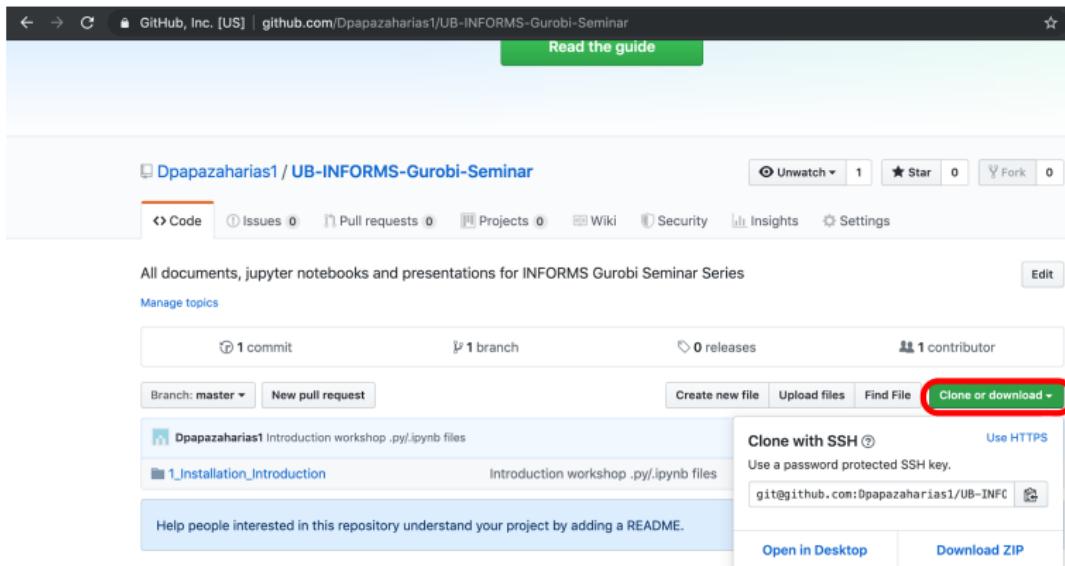
Welcome

- This is the first installment of our Gurobi seminar series!
- All documents, codes, slides, etc. will be held on GitHub:
<https://github.com/Dpapazaharias1/UB-INFORMS-Gurobi-Seminar>
- Lectures will be recorded and posted on YouTube throughout the semester

Downloading Repository

- Go to:

<https://github.com/Dpapazaharias1/UB-INFORMS-Gurobi-Seminar>



The screenshot shows the GitHub repository page for 'Dpapazaharias1 / UB-INFORMS-Gurobi-Seminar'. The page includes a 'Read the guide' button, a navigation bar with links for Code, Issues, Pull requests, Projects, Wiki, Security, Insights, and Settings, and a summary section showing 1 commit, 1 branch, 0 releases, and 1 contributor. A red box highlights the 'Clone or download' button in the top right corner of the main content area. Below this, there are two repository items: 'Dpapazaharias1 Introduction workshop .py/.ipynb files' and '1_Installation_Introduction'. A blue box at the bottom encourages adding a README. On the right side, there are options for cloning with SSH or HTTPS, a copy link, and buttons for 'Open in Desktop' and 'Download ZIP'.

Information

- Gurobi is a popular commercial optimizer used for solving LP, MIP and QP.
- Throughout this series we will cover many classic OR problems, their solution techniques and how to implement them using Gurobi
- Although the focus of this series is mathematical programming we will briefly cover data structures in Python, algorithms and heuristics as needed

Tentative Schedule

1. Installation & Introduction
2. Inventory Problems
3. Network Flow Problems
4. Vertex Packing - User Cuts
5. TSP Part 1 - Lazy Cuts
6. TSP Part 2 - Lagrangian Relaxation and Heuristics
7. Cutting Stock Problem - Column Generation

Contents

1. Installation

2. Mathematical Modeling

3. Basic Syntax

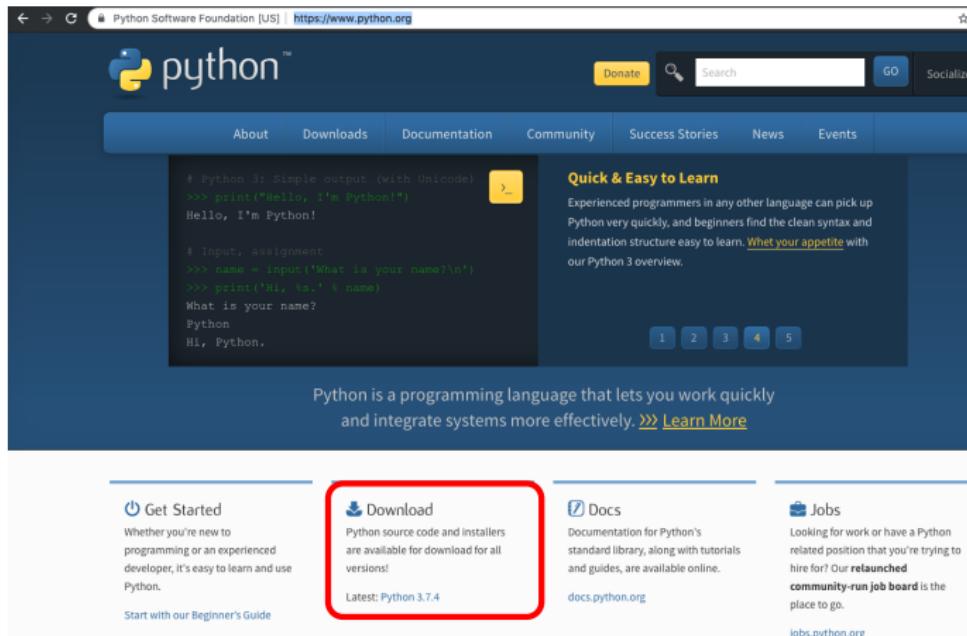
Required Software

The required software for this series:

- Python
- Jupyter
- Atom
- Gurobi

Installing Python

- Visit <https://www.python.org>



The screenshot shows the Python Software Foundation's website. At the top, there is a navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. Below the navigation bar, there is a dark sidebar containing Python code examples:

```
# Python 3: Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!

# Input, assignment
>>> name = input('What is your name?\n')
>>> print('Hi, %s' % name)
What is your name?
Python
Hi, Python.
```

To the right of the sidebar, there is a section titled "Quick & Easy to Learn" with the following text:

Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. [Whet your appetite](#) with our Python 3 overview.

Below this section are five numbered buttons (1, 2, 3, 4, 5). The main content area below the sidebar contains the following text:

Python is a programming language that lets you work quickly and integrate systems more effectively. [»» Learn More](#)

At the bottom of the page, there are four main sections:

- Get Started**: Whether you're new to programming or an experienced developer, it's easy to learn and use Python.
[Start with our Beginner's Guide](#)
- Download**: Python source code and installers are available for download for all versions!
Latest: Python 3.7.4
- Docs**: Documentation for Python's standard library, along with tutorials and guides, are available online.
[docs.python.org](#)
- Jobs**: Looking for work or have a Python related position that you're trying to hire for? Our [relaunched community-run job board](#) is the place to go.
[jobs.python.org](#)

Installing Python

- Scroll down and select the proper version for your computer

[Full Changelog](#)

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debd3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

Installing Python

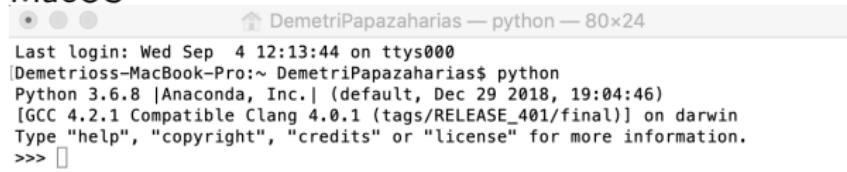
- For Windows users, select “Add Python 3.7 to PATH”
- If not selected you, go to Environment Variables
- Then add “C:\Python37” to PATH



Testing Python

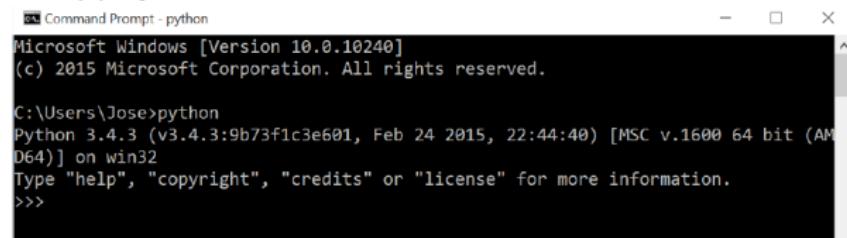
- To test if Python was installed correctly open a command prompt (Windows) or terminal (MacOS)
- Type: “python” and hit enter

- MacOS



```
Last login: Wed Sep  4 12:13:44 on ttys000
[Demetrios-MacBook-Pro:~ DemetriPapazaharias$ python
Python 3.6.8 |Anaconda, Inc.| (default, Dec 29 2018, 19:04:46)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

- Windows

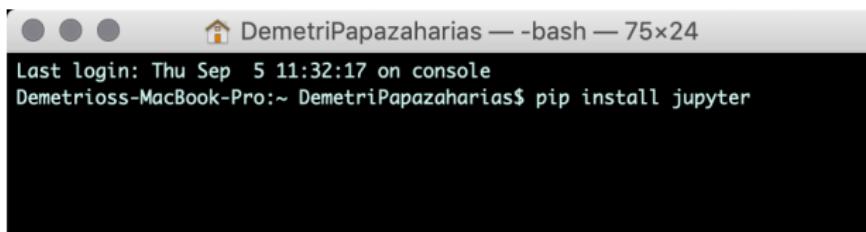


```
Command Prompt - python
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Jose>python
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

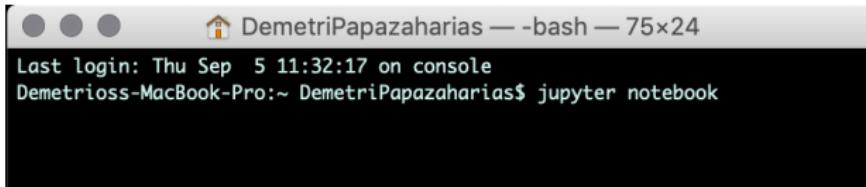
Installing Jupyter

- Once Python is installed correctly we can download Jupyter
- In the command prompt/terminal, type “pip install jupyter”



```
● ● ●     🏠 DemetriPapazaharias — -bash — 75x24
Last login: Thu Sep  5 11:32:17 on console
Demetrios-MacBook-Pro:~ DemetriPapazaharias$ pip install jupyter
```

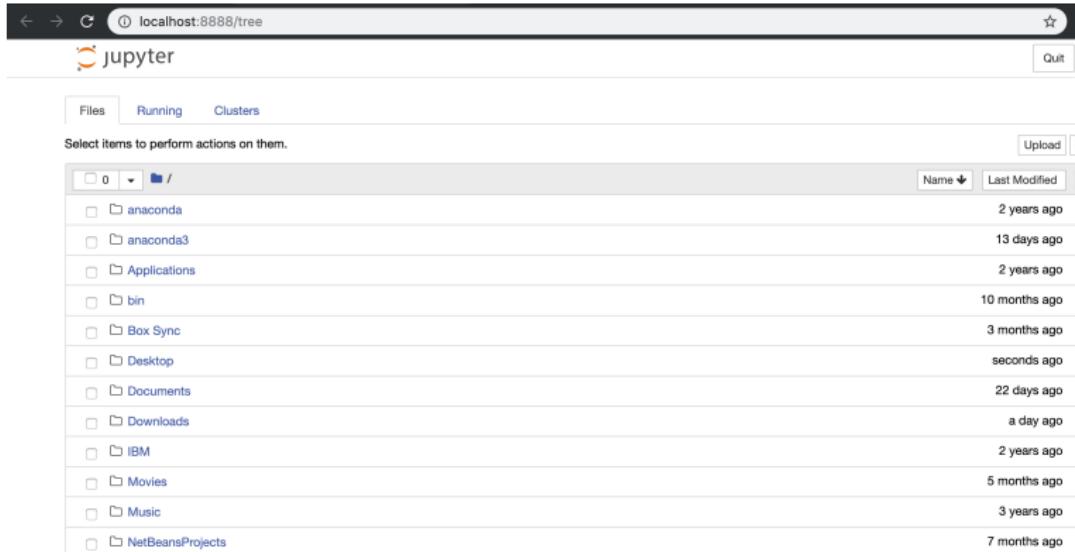
- Once Jupyter is installed, type “jupyter notebook”



```
● ● ●     🏠 DemetriPapazaharias — -bash — 75x24
Last login: Thu Sep  5 11:32:17 on console
Demetrios-MacBook-Pro:~ DemetriPapazaharias$ jupyter notebook
```

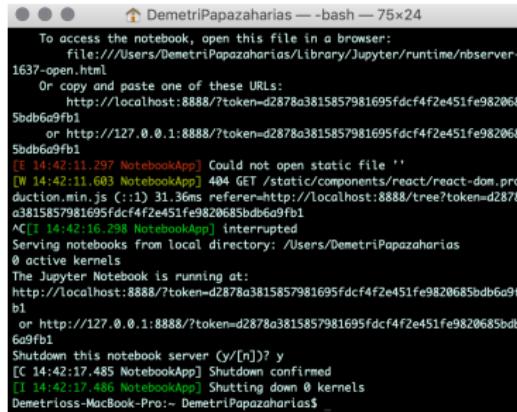
Installing Jupyter

- Once you enter “jupyter notebook” your browser should open



Closing Jupyter

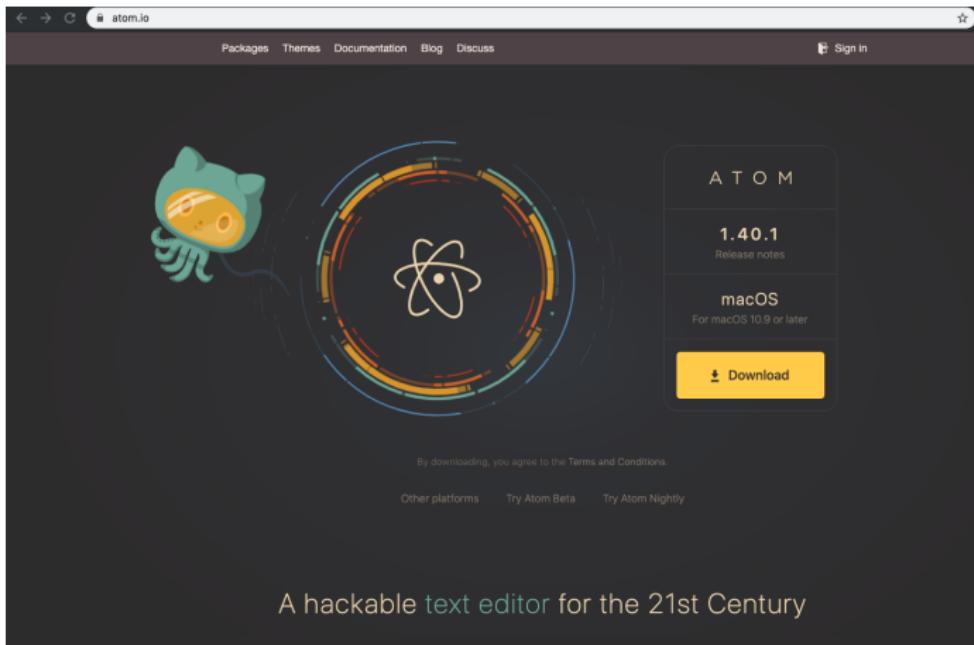
- While a Jupyter session is active, you cannot use the terminal which you entered “jupyter notebook”
- To close the Jupyter app, close the terminal or hit “ctrl + c” to interrupt the server



```
DemetriPapazaharias — bash — 75x24
To access the notebook, open this file in a browser:
    file:///Users/DemetriPapazaharias/Library/Jupyter/runtime/nbserver-
1637-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=d2878a3815857981695fdc4f2e451fe982068
5bdb6a9fb1
    or http://127.0.0.1:8888/?token=d2878a3815857981695fdc4f2e451fe982068
5bdb6a9fb1
[E 14:42:11.297 NotebookApp] Could not open static file ''
[W 14:42:11.603 NotebookApp] 404 GET /static/components/react/react-dom.pro-
duction.min.js (::1) 31.36ms referer=http://localhost:8888/tree?token=d2878
a3815857981695fdc4f2e451fe9820685bdb6a9fb1
^C [I 14:42:16.298 NotebookApp] interrupted
Serving notebooks from local directory: /Users/DemetriPapazaharias
0 active kernels
The Jupyter Notebook is running at:
http://localhost:8888/?token=d2878a3815857981695fdc4f2e451fe9820685bdb6a9f
b1
    or http://127.0.0.1:8888/?token=d2878a3815857981695fdc4f2e451fe9820685bdb
6a9fb1
Shutdown this notebook server (y/[n])? y
[C 14:42:17.485 NotebookApp] Shutdown confirmed
[C 14:42:17.486 NotebookApp] Shutting down 0 kernels
Demetrios-MacBook-Pro:~ DemetriPapazaharias$
```

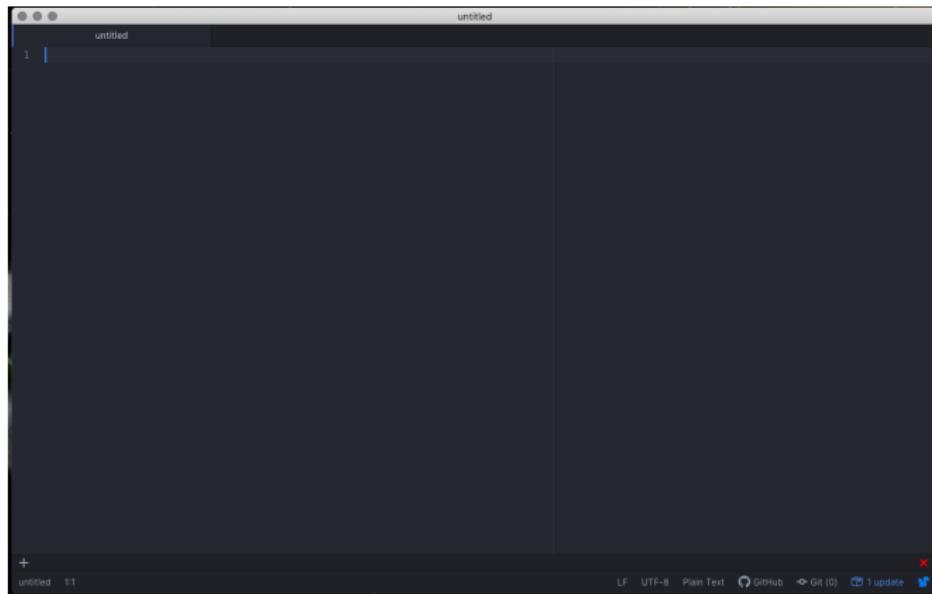
Downloading and Installing Atom

- Go to <https://atom.io/>



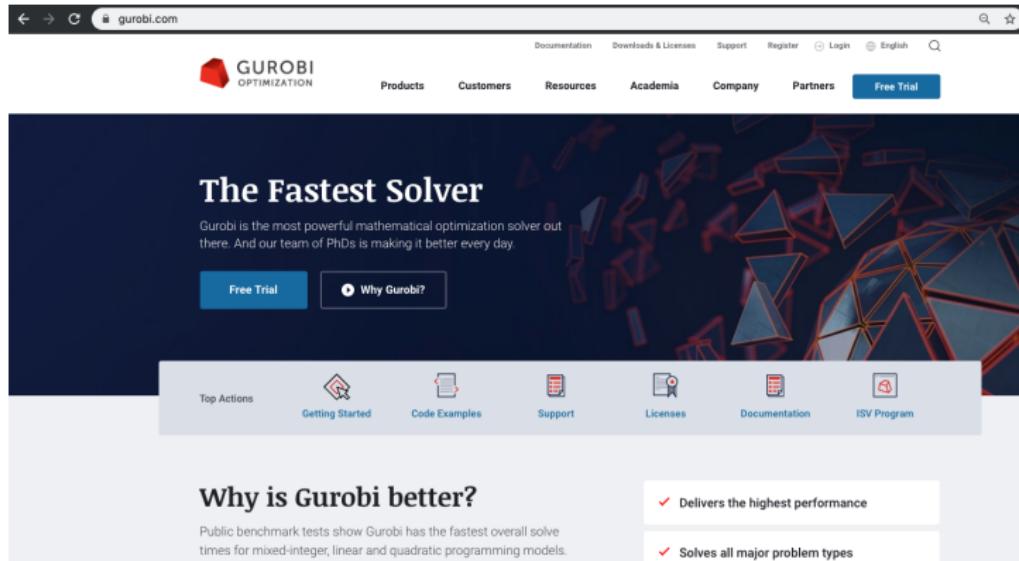
Downloading and Installing Atom

- Once installed, open Atom and it should look like this



Downloading and Installing Gurobi

- Go to <https://www.gurobi.com>



The screenshot shows the Gurobi Optimization website. At the top, there's a navigation bar with links for Documentation, Downloads & Licenses, Support, Register, Login, English, and a search icon. Below the navigation is a main banner with the text "The Fastest Solver" and a subtext about Gurobi being the most powerful solver. It features a "Free Trial" button and a "Why Gurobi?" button. The background of the banner is a dark blue with abstract geometric shapes. Below the banner is a navigation menu with icons for Top Actions, Getting Started, Code Examples, Support, Licenses, Documentation, and ISV Program. A callout box highlights two features: "Delivers the highest performance" and "Solves all major problem types".

Why is Gurobi better?

Public benchmark tests show Gurobi has the fastest overall solve times for mixed-integer, linear and quadratic programming models.

✓ Delivers the highest performance

✓ Solves all major problem types

Create an Account

- Click on Register
- Select academic and fill the form with your information

Account Type: Academic Commercial

First Name: *

Last Name: *

Email Address: *

University/Company: *

Academic Position:

Phone Number: *

Check this box if you also consult with commercial businesses:

Country: *

*Required: The information you provide to us will be used in accordance with the terms of our [Privacy Policy](#).

[Access Now](#)

Download Gurobi Optimizer

- Return to home page and select Downloads & Licenses
- Select “Gurobi Optimizer - Download Software”
- Select the appropriate version for your operating system



The screenshot shows the Gurobi Optimization website. At the top, there is a navigation bar with links for Documentation, Downloads & Licenses, Support, My Account, English, and a search icon. Below the navigation bar, there is a main menu with links for Products, Customers, Resources, Academia, Company, Partners, and a Free Trial button. The main heading "Gurobi Optimizer – Get the Software" is displayed prominently.

Gurobi Optimizer – Get the Software

Get the software

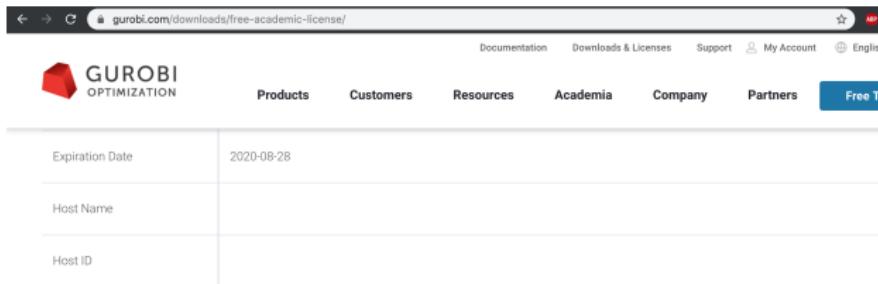
Gurobi Optimizer is the Gurobi optimization libraries. In addition to the software, the corresponding README file contains installation instructions.

Here is the list of bug fixes for each release.

Current version		64-bit Windows		32-bit Windows		64-bit Linux		64-bit macOS		64-bit AIX	
8.1.1	README	Gurobi 8.1.1-win64.msi				gurobi 8.1.1-linux64.tar.gz		gurobi 8.1.1-mac64.pkg		gurobi 8.1.1-power64.tar.gz	
Old versions											
8.0.1	README	Gurobi 8.0.1-win64.msi				gurobi 8.0.1-linux64.tar.gz		gurobi 8.0.1-mac64.pkg		gurobi 8.0.1-power64.tar.gz	
7.5.2	README	Gurobi 7.5.2-win64.msi		Gurobi 7.5.2-win32.msi		gurobi 7.5.2-linux64.tar.gz		gurobi 7.5.2-mac64.pkg		gurobi 7.5.2-power64.tar.gz	
7.0.2	README	Gurobi 7.0.2-win64.msi		Gurobi 7.0.2-win32.msi		gurobi 7.0.2-linux64.tar.gz		gurobi 7.0.2-mac64.pkg		gurobi 7.0.2-power64.tar.gz	

Getting a License

- Under Downloads & Licenses, select “Academic Licenses”
- Accept the conditions of use
- Copy and paste the license into a command prompt/terminal (including grbgetkey)



The screenshot shows the Gurobi Optimization website with the URL gurobi.com/downloads/free-academic-license/. The page has a navigation bar with links to Documentation, Downloads & Licenses (which is highlighted), Support, My Account, and English. Below the navigation bar, there's a main menu with tabs for Products, Customers, Resources, Academia, Company, Partners, and a blue 'Free' tab. A table below the menu contains three rows: 'Expiration Date' with the value '2020-08-28', 'Host Name' (empty), and 'Host ID' (empty).

Installation

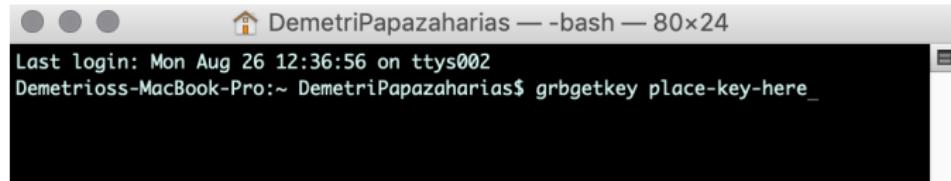
To install this license on a computer where Gurobi Optimizer is installed, copy and paste the following command to the Start/Run menu (Windows only) or a command/terminal prompt (any system):

grbgetkey [REDACTED]

The `grbgetkey` command requires an active internet connection. If your computer has no internet access, or you get no response or an error message such as "Unable to contact key server", Please click [here](#) for additional instructions.

Activating License

- Copy and paste the license (including grbgetkey) into terminal

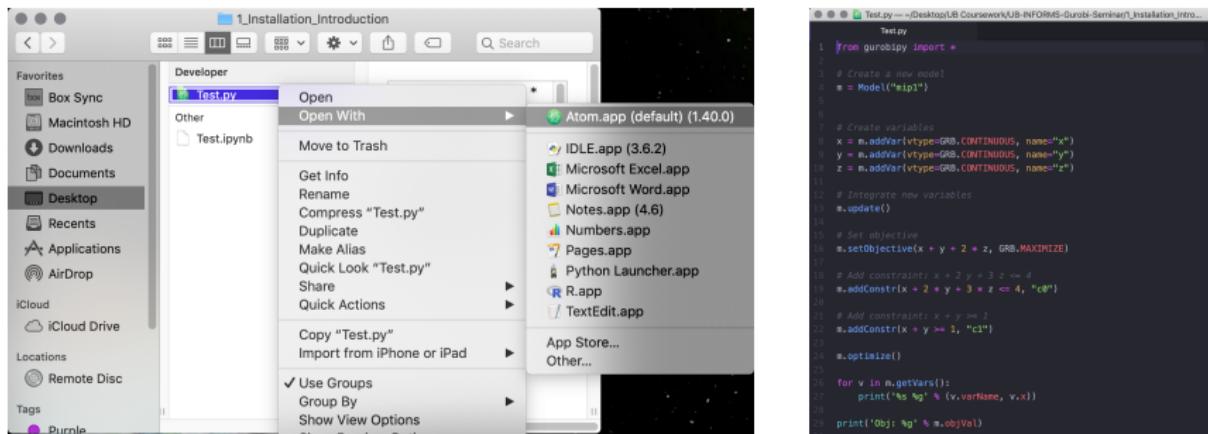


The screenshot shows a terminal window on a Mac OS X desktop. The window title is "DemetriPapazaharias — bash — 80x24". In the terminal, the user has typed the command "grbgetkey place-key-here_". The terminal interface includes a menu bar with "File", "Edit", "View", "Shell", "Help", and "About", along with standard window controls.

- If done correctly, this will activate your Gurobi license

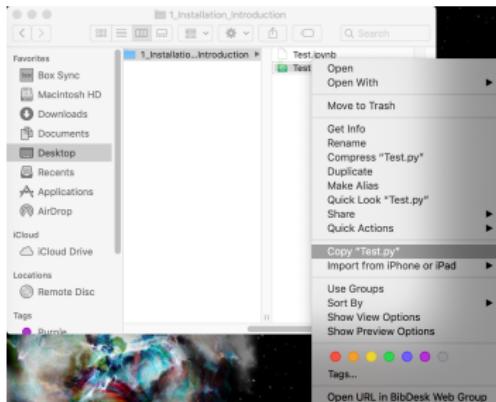
Testing Atom and Gurobi

- We can check to see if we installed our software correctly
- Go to “1_Installation_Introduction” in the folder you downloaded from GitHub
- Right click “Test.py” and open in Atom



Testing Atom and Gurobi

- Open a terminal/command prompt and type
 - `gurobi.sh` on MacOS/UNIX
 - `gurobi` on Windows
- Copy the file “`Test.py`” and paste into the terminal/command prompt and hit enter



Two terminal windows are shown side-by-side. Both windows have the title bar 'DemetriPapazaharias — bash — 75x25'.

Terminal 1 (Left):

```
Last login: Thu Sep  5 11:32:17 on console
Demetrios-MacBook-Pro:~ DemetriPapazaharias$ gurobi.sh /Users/DemetriPapazaharias/Desktop/U8\ Coursework/gurobi_seminar/1_Installation_Introduction/Test.py
```

Terminal 2 (Right):

```
Last login: Thu Sep  5 11:32:17 on console
Demetrios-MacBook-Pro:~ DemetriPapazaharias$ gurobi.sh /Users/DemetriPapazaharias/Desktop/U8\ Coursework/gurobi_seminar/1_Installation_Introduction/Test.py
```

Academic license - for non-commercial use only
Copyright © 2012, Gurobi Optimization, Inc.

Coefficient statistics:
Matrix range [1e+00, 3e+00]
Objective range [-1e+00, 2e+00]
Bounds range [-9e+00, 9e+00]
RHS range [-1e+00, 4e+00]

Presolve removed 2 rows and 3 columns
Presolve time: 0.000s

Presolve: All rows and columns removed
Iteration Objective Primal Inf. Dual Inf. Time
0 4.0000000e+00 0.000000e+00 0.000000e+00 0s

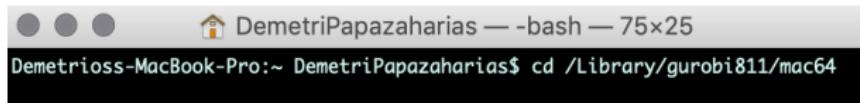
Solved in 0 iterations and 0.01 seconds
Optimal objective 4.0000000e+00

x 4
y 0
z 0
Obj: 4

- You should be able to use Python and Gurobi now!

Jupyter and Gurobi

- In order to use Jupyter and Gurobi we need one extra step
- Open terminal and type:
 - On Windows: `cd C:/gurobi811/win64`
 - On MacOS: `cd /Library/gurobi811/mac64`

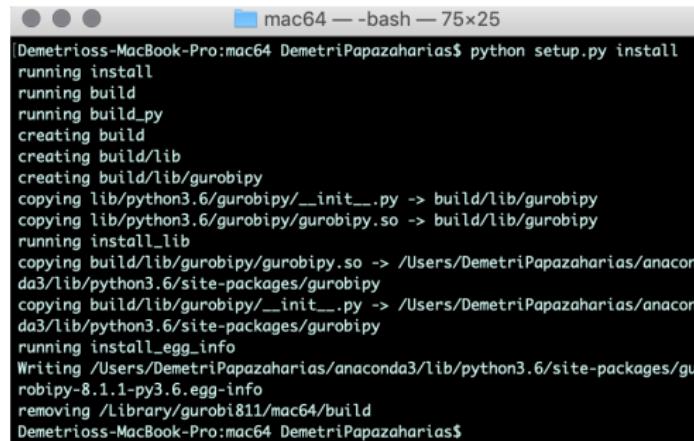


A screenshot of a macOS terminal window. The title bar says "DemetriPapazaharias — bash — 75x25". The command line shows the user's path as "Demetrios-MacBook-Pro:~ DemetriPapazaharias\$" followed by the command "cd /Library/gurobi811/mac64".

- If this does not work then find where the folder `gurobi811` is located
- Inside `gurobi811` copy the folder `mac64` or `win64`
- Open the terminal/command prompt and type `cd`
- Paste the folder into the terminal and hit enter

Jupyter and Gurobi

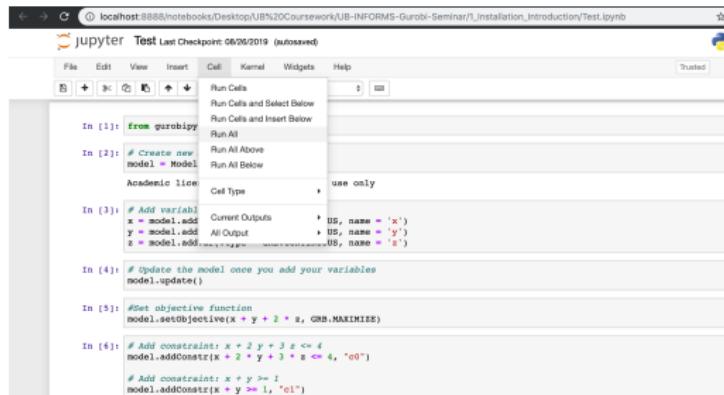
- Now your directory is set to mac64 or win64
- Type python setup.py install and hit enter
- The output should look something like this



```
[Demetrios-MacBook-Pro:mac64 DemetriPapazaharias$ python setup.py install
running install
running build
running build_py
creating build
creating build/lib
creating build/lib/gurobipy
copying lib/python3.6/gurobipy/__init__.py -> build/lib/gurobipy
copying lib/python3.6/gurobipy/gurobipy.so -> build/lib/gurobipy
running install_lib
copying build/lib/gurobipy/gurobipy.so -> /Users/DemetriPapazaharias/anaconda3/lib/python3.6/site-packages/gurobipy
copying build/lib/gurobipy/__init__.py -> /Users/DemetriPapazaharias/anaconda3/lib/python3.6/site-packages/gurobipy
running install_egg_info
Writing /Users/DemetriPapazaharias/anaconda3/lib/python3.6/site-packages/gurobipy-8.1.1-py3.6.egg-info
removing /Library/gurobi811/mac64/build
Demetrios-MacBook-Pro:mac64 DemetriPapazaharias$
```

Jupyter and Gurobi

- Open a new terminal/command prompt
- Start a new Jupyter session (jupyter notebook)
- In the Jupyter app, go to “1_Installation_Introduction”
- Open Test.ipynb, click on Cell and select “Run All”



The screenshot shows a Jupyter Notebook interface with several code cells. The code is written in Python and uses the Gurobi library to define a linear programming model. The cells are numbered In [1] through In [6]. A context menu is open over In [1], showing options like Run Cell, Run Cells and Select Below, Run Cells and Insert Below, Run All, Run All Above, Run All Below, and Cell Type. The code in the cells is as follows:

```
In [1]: from gurobipy import *
In [2]: # Create new model
model = Model("Academic license")
In [3]: # Add variables
x = model.addVar(name='x')
y = model.addVar(name='y')
z = model.addVar(name='z')
Current Outputs:
  > US, name = 'x'
  > US, name = 'y'
  > US, name = 'z'
All Output:
  > US, name = 'x'
  > US, name = 'y'
  > US, name = 'z'
In [4]: # Update the model once you add your variables
model.update()
In [5]: # Set objective function
model.setObjective(x + 2 * y + 3 * z, GRB.MAXIMIZE)
In [6]: # Add constraint: x + 2 * y + 3 * z <= 4
model.addConstr(x + 2 * y + 3 * z <= 4, "c0")
# Add constraint: x + y >= 1
model.addConstr(x + y >= 1, "c1")
```

- You should be able to use Gurobi and Jupyter now!

Contents

1. Installation

2. Mathematical Modeling

3. Basic Syntax

Introduction to Mathematical Modeling

- Mathematical models are idealized representations of a system using mathematical concepts and language
- Every mathematical model has four major components
 1. Decision Variables
 2. Constraints
 3. Objective Function
 4. Parameters
- General form of model

$$\begin{aligned} \max \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) \leq b \\ & \mathbf{x} \in X \end{aligned}$$

Introduction to Mathematical Modeling

- The **decision variables** are values which we control in a mathematical model , e.g. *how much of product X to produce, build facility at location A or not, etc.*
- The set of **constraints** represent restrictions of our decisions variables, e.g. *spending budget, resource scarcity, etc.*
- The **objective function** is a measure of performance based off these decisions, e.g. *profit, total path length, distance traveled*
- The **parameters** are constants in the objective function and constraints related to the decision variables, e.g. *profit per unit, cost of building facility at location A, distance between cities, etc.*

Contents

1. Installation

2. Mathematical Modeling

3. Basic Syntax

Overview

To use Gurobi with Python:

```
from gurobipy import *
```

Objects and methods to remember:

- Model
 - Model.addVar(lb, ub, obj, vtype, name)
 - Model.addConstr(constraint, name)
 - Model.update()
 - Model.modelSense or Model.SetObjective()
 - optimize()
- Var
 - Var.X ← Get value of variable in current solution
- Constr
 - Constr.Pi ← Dual value
- LinExpr
 - LinExpr.getValue()
- quicksum, multidict, tuplelist

Building a model

min 0
s.t.

```
# Create a new model
m = Model("mip1")
```

Building a model

$$\begin{aligned} \text{min} \quad & 0 \\ \text{s.t.} \quad & x, y, z \geq 0 \end{aligned}$$

```
# Create a new model
m = Model("mip1")

# Create variables
x = m.addVar(vtype=GRB.CONTINUOUS, name="x")
y = m.addVar(vtype=GRB.CONTINUOUS, name="y")
z = m.addVar(vtype=GRB.CONTINUOUS, name="z")

# Integrate new variables
m.update()
```

Building a model

$$\begin{aligned} \max \quad & x + y + 2z \\ \text{s.t.} \quad & x, y, z \geq 0 \end{aligned}$$

```
# Create a new model
m = Model("mip1")

# Create variables
x = m.addVar(vtype=GRB.CONTINUOUS, name="x")
y = m.addVar(vtype=GRB.CONTINUOUS, name="y")
z = m.addVar(vtype=GRB.CONTINUOUS, name="z")

# Integrate new variables
m.update()

# Set objective
m.setObjective(x + y + 2 * z, GRB.MAXIMIZE)
```

Building a model

$$\begin{aligned}
 \text{max} \quad & x + y + 2z \\
 \text{s.t.} \quad & x + 2y + 3z \leq 4 \\
 & x + y \geq 1 \\
 & x, y, z \geq 0
 \end{aligned}$$

```

# Create a new model
m = Model("mip1")

# Create variables
x = m.addVar(vtype=GRB.CONTINUOUS, name="x")
y = m.addVar(vtype=GRB.CONTINUOUS, name="y")
z = m.addVar(vtype=GRB.CONTINUOUS, name="z")

# Integrate new variables
m.update()

# Set objective
m.setObjective(x + y + 2 * z, GRB.MAXIMIZE)

# Add constraint: x + 2 y + 3 z <= 4
m.addConstr(x + 2 * y + 3 * z <= 4, "c0")

# Add constraint: x + y >= 1
m.addConstr(x + y >= 1, "c1")

```

Building a model

$$\begin{array}{ll}\text{max} & x + y + 2z \\ \text{s.t.} & x + 2y + 3z \leq 4 \\ & x + y \geq 1 \\ & x, y, z \geq 0\end{array}$$

Solution: $(x, y, z) = (4, 0, 0)$

```
# Create a new model
m = Model("mip1")

# Create variables
x = m.addVar(vtype=GRB.CONTINUOUS, name="x")
y = m.addVar(vtype=GRB.CONTINUOUS, name="y")
z = m.addVar(vtype=GRB.CONTINUOUS, name="z")

# Integrate new variables
m.update()

# Set objective
m.setObjective(x + y + 2 * z, GRB.MAXIMIZE)

# Add constraint: x + 2 y + 3 z <= 4
m.addConstr(x + 2 * y + 3 * z <= 4, "c0")

# Add constraint: x + y >= 1
m.addConstr(x + y >= 1, "c1")

m.optimize()
```