North Carolina State University
CCEE Department, Spring 2024
CE339 – Civil Engineering Systems

Software Handout

[Follow if you want to run models and solver in your computer instead of using Google Colab]

1) Anaconda Python Distribution

Anaconda is an open-source distribution for Python and R, designed for data science. It focuses on easing package management and deployment. The core of Anaconda's package management is 'conda', a system that checks the current environment to ensure new installations don't conflict with existing packages. This distribution includes over 250 pre-installed packages and offers access to over 7500 additional packages from PyPI and conda. Anaconda also features Anaconda Navigator, a GUI that allows users to manage packages and environments without command-line commands. Navigator enables package installation, running, and updating within environments.

A key distinction of Anaconda is how it handles package dependencies, particularly when compared to pip, another package manager. While pip installs dependencies without considering potential conflicts with existing packages, conda carefully analyzes the environment and existing installations. This approach helps maintain compatibility and prevents issues like a working TensorFlow installation breaking due to conflicting NumPy versions installed by pip. Users can install packages from various sources, including the Anaconda repository, Anaconda Cloud, or private repositories, using the 'conda install' command. Anaconda Inc. ensures the availability of compiled and built packages for Windows, Linux, and MacOS. Additionally, while conda tracks both its own installations and those made by pip, it ensures compatibility and stability within the conda environment.

Because third-party libraries for Python like NumPy and MatPlotLib are so useful, a distribution like Anaconda is a convenient way to get them. It is also freely available and runs on Windows, Linux, and Mac. The Gurobi website has instructions for obtaining and installing Anaconda and for combining it with Gurobi so these tools can be used together. If you already have Anaconda from a previous installation, the links below are useful for you as well because they show how to add Gurobi to your existing Anaconda distribution. Please install the version associated with Python 3.8.

- After Anaconda is installed in the terminal you should install pip:

```
python get-pip.py
```

- And upgrade its version by:

```
pip install -U pip
```

More details:

- Installing the Anaconda Python distribution: : **Windows**, **Linux**, **Mac**
- Anaconda download: https://www.continuum.io/downloads
- Installing and upgrading pip: https://pip.pypa.io/en/stable/installing/
- Anaconda packages search engine: https://anaconda.org

2) Pyomo

Pyomo is an open-source software developed in Python, offering a broad spectrum of optimization functionalities for the creation, resolution, and examination of optimization models. Its primary strength lies in structuring optimization applications. Unlike other algebraic modeling languages such as AMPL, AIMMS, and GAMS, Pyomo stands out by integrating its modeling components within a comprehensive, high-level programming language, enriched with extensive libraries.

This software is versatile in defining symbolic problems in general terms, generating specific instances of these problems, and solving them with both open-source and commercial solvers. Pyomo's diverse problem-solving capabilities include:

- Linear programming
- Quadratic programming
- Nonlinear programming
- Mixed-integer linear programming
- Mixed-integer quadratic programming
- Mixed-integer nonlinear programming
- Stochastic programming
- Generalized disjunctive programming
- Differential algebraic equations
- Bilevel programming
- Mathematical programs with equilibrium constraints

Moreover, Pyomo facilitates iterative analysis and scripting within a robust programming environment. It has also demonstrated its efficiency as a framework for creating advanced optimization and analysis tools. An example of this is the PySP package, which offers generic solvers for stochastic programming. PySP capitalizes on Pyomo's integration within a high-level programming language, enabling the transparent parallelization of subproblems through Python's parallel communication libraries.

A number of assignments in the course require that you use Pyomo for mathematical modeling. It is freely available and runs on Windows, Linux, and Mac computers. You can obtain Pyomo after you install the Anaconda Python Distribution.

After you installed pip:

```
pip install pyomo
pip install 'pyomo[optional]'
```

3) Gurobi Optimizer

The Gurobi Optimizer is a sophisticated, commercial solver used for a variety of optimization problems. It excels in solving linear programming (LP) and quadratic programming (QP) problems, along with their mixed-integer variants and quadratically-constrained programming challenges. This versatility makes it a powerful tool in fields where optimization is crucial, such as logistics, finance, energy, telecommunications, and manufacturing. One of the key strengths of Gurobi Optimizer is its compatibility with multiple programming languages. This feature allows it to seamlessly integrate into various software environments and workflows. Users can implement Gurobi in popular programming languages such as Python, C++, Java, .NET, and several others, facilitating its adoption in diverse projects and applications.

Gurobi is renowned for its speed and efficiency, often delivering solutions faster than other solvers, especially in complex and large-scale optimization problems. This efficiency is a result of continuous research and development in optimization algorithms and computational performance. In addition to its core solving capabilities, Gurobi provides a range of tools and features to enhance the user experience. These include intuitive interfaces, detailed documentation, and robust support for model development and result analysis. The software also offers advanced features like distributed optimization and cloud computing support, enabling users to tackle large-scale optimization problems more effectively.

The integration of Pyomo with Gurobi Optimizer presents a powerful combination for tackling complex optimization problems. Pyomo, with its flexible modeling capabilities and Python-based environment, serves as an ideal platform for formulating and structuring optimization models. When these models are solved using Gurobi, known for its speed and efficiency in handling large-scale and intricate problems, users benefit from a highly effective optimization workflow. This integration leverages Pyomo's ability to construct sophisticated models, including linear, nonlinear, and mixed-integer problems, and pairs it with Gurobi's robust solving algorithms. The seamless connection between Pyomo's Python-based environment and Gurobi's solver enhances the accessibility and usability of advanced optimization techniques. Users can easily define their models in Pyomo and then employ Gurobi for high-performance solving, making this combination particularly valuable in academic research, complex industrial applications, and data-intensive sectors where precision and computational efficiency are necessary.

A number of assignments in the course require that you use the Gurobi Optimizer with Pyomo. It is freely available and runs on Windows, Linux, and Mac computers. You can obtain Gurobi Optimizer here:

http://www.gurobi.com/academia/for-universities

In the upper righthand corner of the page, you'll see a blue "Register" button. You'll need to click on it and create an account using your university email address. For "account type," select "Academic" (not Commercial), which will be free for you.

Once you've created an account and signed in, go back to the link I pasted above and follow the directions for Obtaining a Free Academic License to get a named-user academic license, or see below for those directions:

To obtain a free named-user academic license (Named-User, Unlimited-Use, Single-Machine academic license):

1. After logging in, visit the **Download Gurobi Optimizer** page, select the version you want to download and click the related 'Download' button. You should also download README.txt.
2. After downloading, visit the **Free Academic License** page and follow the instructions on that page.
3. Follow the instructions in README.txt to install the software.
4. Once the software has been installed, run 'grbgetkey' using the argument provided when you created your license (ex: grbgetkey ae36ac20-16e6-acd2-f242-4da6e765fa0a). The 'grbgetkey' program will retrieve your license key and prompt you to store it on your machine. It will also validate your academic license eligibility by confirming that you are connected to the Internet from a recognized academic domain (e.g., any '.edu' address).

The current version of Gurobi Optimizer is 7.0. Once you get your license installed, which you must do from an ncsu.edu network, you will be able to start the Gurobi shell. You can invoke the shell with gurobi.bat (on Windows) or gurobi.sh (on Linux/Mac):

```
    $ gurobi.sh
    Python 3.8 (default, Oct 23 2021, 18:05:06)
    [GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.0.59.5)]
on darwin
    Type "help", "copyright", "credits" or "license" for more
information.

    Gurobi Interactive Shell (mac64), Version 6.5.0
    Copyright (c) 2021, Gurobi Optimization, Inc.
    Type "help()" for help

    gurobi>
```

At the Gurobi prompt (gurobi>) you have access to a Python 2.7 interpreter

that you can use like any other Python interpreter, but you'll also have access to one of the most powerful optimizers available at any price.
More details:

- Gurobi Optimizer reference manual: http://www.gurobi.com/documentation/
- Getting started with Gurobi webinar (building a model within Python) http://www.gurobi.com/resources/seminars-and-videos/seminars-videos#Starting

4) Coin-OR

The Computational Infrastructure for Operations Research (COIN-OR) is an initiative for the development of open-source software for Operations Research (OR). Drawing inspiration from the success of open-source projects like Linux and Apache, a team at IBM Research advocated for open source as a parallel approach to disseminating software, models, and data. COIN-OR started as a response, fostering open-source development within the computational operations research community. It provides essential online resources and hosting services, enabling others to launch and maintain their open-source software projects.

Launched in 2000 during the 17th International Symposium on Math Programming in Atlanta, Georgia, the COIN-OR website began as an exploratory venture. By 2007, it had expanded to include 25 application projects, encompassing tools for linear programming (like COIN-OR CLP), nonlinear programming (such as IPOPT), integer programming (including CBC, Bcp, and COIN-OR SYMPHONY), algebraic modeling languages (e.g., Coopr), and more. This growth continued, reaching 48 projects by 2011, and 73 as of 2023. Hosted by the Institute for Operations Research and the Management Sciences (INFORMS) and managed by the educational, non-profit COIN-OR Foundation, COIN-OR stands as a testament to the value of open-source in academia and research. This open-source approach not only facilitates transparency and reproducibility in computational research but also accelerates innovation by allowing researchers worldwide to collaborate, refine, and build upon existing tools and models.

For the purpose of this course the use of the solvers: CLP (for linear programming), CBC (for integer programming) and for IPOPT (for nonlinear programming) are encouraged. The COIN-COIN-CLP, CBC, and IPOPT, are integral part of this open-source initiative. CLP is designed to solve large-scale linear problems, making it essential for both academic and industrial use. CBC is able to effectively tackle complex mixed-integer programming problems, commonly found in logistics, resource management, and many other fields. IPOPT is key for solving nonlinear optimization, in application areas that span energy management to other discipline specific as well as multi-disciplinary problems in engineering.

Example of how to install Coin-CBC in your computer (running MAC or Linux):

- Install COIN-CBC (https://projects.coin-or.org/Cbc)
- Download sources

- Create a folder
- Inside the folder
    - mkdir build
    - cd build
    - ../configure
    - make
    - make test
    - make install
- Update PATH variable (.bash_profile) to point also to /Cbc-2.9.8/build/bin

* It is necessary to have installed gcc and gfortran compilers to install Coin-OR solvers.

** Once Coin-CBC is installed the libraries for Coin-CLP are also ready to be used