Introduction to Computational Models Using Python

CS4491

José M. Garrido C.

Department of Computer Science
College of Computing and Software Engineering
Kennesaw State University

June, 2016
A general process of problem solving involves the following steps:

1. Understanding the problem
2. Describing the problem in a clear, complete, and unambiguous form
3. Designing a solution to the problem (algorithm)
4. Developing a computer solution to the problem.
5. Test
An essential modeling method is to use mathematical entities such as numbers, functions, and sets to describe properties and their relationships to problems and real-world systems.

Such models are known as *mathematical models*.

A *computational model* is an implementation in a computer system of a mathematical model and usually requires high performance computational resources to execute.

The computational model is used to study the behavior of a large and complex system.
Developing a computational model consists of:

- Applying a formal software development process
- Applying *Computer Science* concepts, principles and methods, such as:
  - Abstraction and decomposition
  - Programming principles
  - Data structures
  - Algorithm structures
  - Concurrency and synchronization
  - Modeling and simulation
  - Multi-threading, parallel, and distributed computing for high performance (HPC)
Developing Computational Models

Real-world level

Real system

Abstract level

Conceptual model

Mathematical model

Computational model
Is a **computer model** that implements the solution to a (scientific) complex problem.

It usually requires a **mathematical model** or representation that has been formulated for the problem.

The software implementation often requires **high-performance computing** (HPC) to run.
Computational science is an emerging area (or discipline) that includes concepts, principles, and methods from **Applied Mathematics** and **Computer Science**.

These are applied in various areas of **science** and **engineering** to solve large-scale scientific problems.

**Computational modeling** and **computational thinking** are the foundational components of computational science and focuses on reasoning about problems.
Computational Thinking

- **Reasoning** about computer problem solving
- The ability to describe the **requirements** of a problem and, if possible, design a solution that can be implemented in a computer
- **Multi-disciplinary** important problems require team work
- Implementation of a solution results in a **computational model**
Multi-disciplinary Development of Computational Models

Applying interdisciplinary and multi-disciplinary approaches

Including perspectives from the various sciences such as:

- physics
- biology
- chemistry
- other disciplines.
What Does Computational Science Involve?

- Reasoning about computer **problem solving**
- The problem domain is **multidisciplinary**
- The ability to describe the requirements of a problem and, if possible, design a solution that can be implemented in a computer
- Developing computational models needs the application of computer science principles, concepts, and methods.
Both are general approaches that are very useful in developing computational models with teams.

- **Multidisciplinary** computing is a non-integrative mixture of approaches from the various disciplines; each discipline retains its methodologies and assumptions without change or development from other disciplines within the multidisciplinary relationship.

- **Interdisciplinary** computing involves the combining of two or more approaches each from an academic field into one single process. An interdisciplinary approach crosses traditional boundaries between academic disciplines.
A Multidisciplinary team approach utilizes the skills and experience of individuals from different disciplines, with each discipline approaching the problem from their own perspective.

Most often, this approach involves separate individual analysis. Multidisciplinary teams provide more knowledge and experience than disciplines operating in isolation.

An interdisciplinary team approach integrates separate discipline approaches into a single analysis. It is a common understanding and holistic view of all aspects of the problem.
The term *multi-dimensional abstraction* is proposed to emphasize the need to improve the development of computational models with an interdisciplinary approach. For practical purposes, it involves a *combined* approach. Every member of the team must use abstraction to reduce the complexity of the real system to fit into the conceptual model. Each individual will use his/her own skills and knowledge to determine the essential aspects of the system to model, within that part of the system that relates to their field.
computing numerical values of variables that define properties of a system

optimization of parameters of a system giving some constraints (Linear and non-linear)

showing the dynamical behavior of a system with time via simulation (continuous or discrete)
Some Challenges of Computational Modeling

- May need to apply several notations, techniques and methods, such as: flowcharts, pseudo-code, UML diagrams, and/or others
- Usually applies important CS/SE concepts, such as: **abstraction** and **decomposition**
- Requires a **mathematical representation** (a formal mathematical model)
- Not all problems can be easily modeled with mathematics
- Many mathematical models do not have a formal (closed) solution
Modeling at a high level of abstraction:

- General purpose scientific computing environments/tools: Matlab, Octave, FreeMat, SciLab
- PyLab is an extension to the Python language, its five core components are Python, NumPy, SciPy, Matplotlib, and IPython.
- Cython is an optimizing compiler for Python.
- Environments/tools for continuous simulation: Simulink (Matlab), Scicos (Scilab), Stella, Vensim, SimCAD, acsIX, Dymola, JMCAD, and others
For improved performance, a more efficient implementation of the computational model is necessary.

Two relevant programming languages that provide very efficient implementations are:

- C
- Fortran.

These have been used for very efficient numerical and scientific programming.
The process of developing a computational model can be enhanced by including three levels of computer implementation:

1. **Prototyping**, using the Matlab programming language
2. **Improved performance implementation**, using: C or Fortran programming languages.
3. Include techniques, methods, and tools for **high performance computing** (HPC).
Abstraction is recognized as a fundamental and essential principle in problem solving and software development. Abstraction and decomposition are extremely important in dealing with large and complex systems. Abstraction is the activity of hiding the details and exposing only the essential features of a particular system. In modeling, one of the critical tasks is representing the various aspects of a system at different levels of abstraction. A good abstraction captures the essential elements of a system, and purposely leaving out the rest.
The process of developing a computational model is illustrated in this section with an extremely simple problem: the temperature conversion problem.

The mathematical representation of the solution to the problem is the formula expressing a temperature measurement $F$ in Fahrenheit in terms of the temperature measurement $C$ in Celsius:

$$F = \frac{9}{5} C + 32$$

Here $C$ is a variable that represents the given temperature in degrees Celsius, and $F$ is a derived variable, whose value depends on $C$. 

José M. Garrido C.  
Introduction to Computational Models Using Python
Another simple problem is formulated and requires computing the area and circumference of a circle, given its radius.

The mathematical model is:

\[
\text{cir} = 2\pi r \\
\text{area} = \pi r^2
\]
Categories of Computational Models

From the perspective of how the model changes state in time, computational models can be divided into two general categories:

1. Continuous models
2. Discrete models

A continuous model is one in which the changes of state in the model occur continuously with time.

A model that represents the temperature in a boiler as part of a power plant can be considered a continuous model.
Continuous Model

\[ y = f(t) \]
A discrete model represents a system that changes its states at discrete points in time, i.e., at specific instants.

The model of a simple car-wash system is a discrete-event model because an arrival event occurs, and causes a change in the state variable that represents the number of cars in the queue that are waiting to receive service from the machine (the server).

This state variable and any other only change its values when an event occurs, i.e., at discrete instants.
Discrete Model

Number of cars waiting vs Time

- **Y-axis:** Number of cars waiting
- **X-axis:** Time

- The graph shows the number of cars waiting over time, with a peak around the 30-40 time interval.
A programming language includes a defined set of syntax and semantic rules.

There are several integrated development environments (IDE) that facilitate the development of programs. Examples of these are: Eclipse, Netbeans, CodeBlocks, and Codelite.

Other tools include IDEs that are designed for numerical and scientific problem solving that have their own programming language.

There are several IDEs available for developing scientific applications with Python. Some of these are: Spyder, IEP, Eric, PyDev, WingIDE, Canopy, Komodo IDE, and Pycharm.
An interpreter is a special program that performs syntax checking of a command in a user program written in a high-level programming language and immediately executes the command.

Examples of interpreters are the ones used for the following languages: Python, MATLAB, Octave, PHP, and Perl.

The Python interpreter reads a command written in the Python programming language and immediately executes the command.

A Python program is a file of Python commands, so the program is also known as a *script*.

In a terminal or command window, all the interaction with a user takes place by typing Python commands.
The **command prompt** is the symbol that the Python interpreter displays on the window to alert the user that it is waiting for a command.

The Python prompt is `>>>` and is used for interactive mode of computing with Python.

In a terminal window, the user starts the Python interpreter simply by typing `python` at the Linux prompt. The interpreter starts and displays the Python prompt `>>>`.

Note how the Python interpreter immediately responds to a command; this is the interactive mode of using Python.
$ python
Python 2.7.8 (default, May 20 2016, 15:20:10)
[GCC 4.9.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> num = 15
>>> num
15
>>> y = 20.6
>>> y
20.6
>>> x = 2.56
>>> j = 200
>>> j
200
>>>
```python
>>> z = y * x + 125.25
>>> z
177.986
>>> z / 12.5
14.23888
>>> 
```