2072U Computational Science I Winter 2022

Week	Topic
1	Introduction
1–2	Solving nonlinear equations in one variable
3–4	Solving systems of (non)linear equations
5–6	Computational complexity
6–8	Interpolation and least squares
8–10	Integration & differentiation
10–12	Additional Topics

Outline

1. What is Computational Science?

2. Some applications

3. Some remarks
   - Symbolic vs. numerical computation
   - Operating systems

What is Computational Science?

- Also known as *Numerical Analysis* and *Scientific Computing*.
- Contains elements of mathematics and computer science.
- Main applications are in physics, engineering and data science.
- Origins are old but the rapid development of computers makes it an ever-changing field...
- Possible definitions:

What is Computational Science?

- Also known as *Numerical Analysis* and *Scientific Computing*.
- Contains elements of mathematics and computer science.
- Main applications are in physics, engineering and data science.
- Origins are old but the rapid development of computers makes it an ever-changing field...
- Possible definitions:
  A discipline concerned with the design, implementation and use of mathematical models to analyse and solve scientific problems. Typically, the term refers to the use of computers to perform simulations or numerical analysis of a scientific system or process. (Nature);

What is Computational Science?

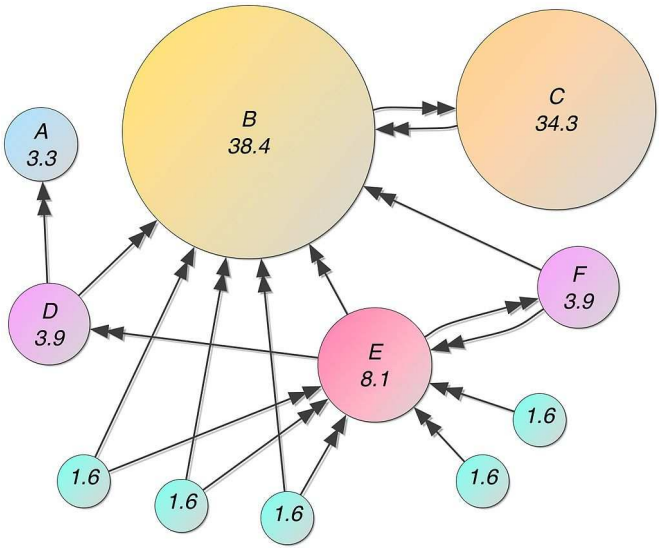
- Also known as *Numerical Analysis* and *Scientific Computing*.
- Contains elements of mathematics and computer science.
- Main applications are in physics, engineering and data science.
- Origins are old but the rapid development of computers makes it an ever-changing field...
- Possible definitions:
  Appropriate topics include the rigorous study of convergence of algorithms, their accuracy, their stability, and their computational complexity. (SIAM Journal on Numerical Analysis);

What is Computational Science?

- Also known as *Numerical Analysis* and *Scientific Computing*.
- Contains elements of mathematics and computer science.
- Main applications are in physics, engineering and data science.
- Origins are old but the rapid development of computers makes it an ever-changing field...
- Possible definitions:
  The process of producing and analyzing approximate, numerical solutions to mathematical or scientific problems.

Famous CS example: PageRank.

- Ranks nodes in a network according to number and quality of links.
- Used by Google for websites.
- Can be formulated as a linear system.



Wikipedia, public domain.

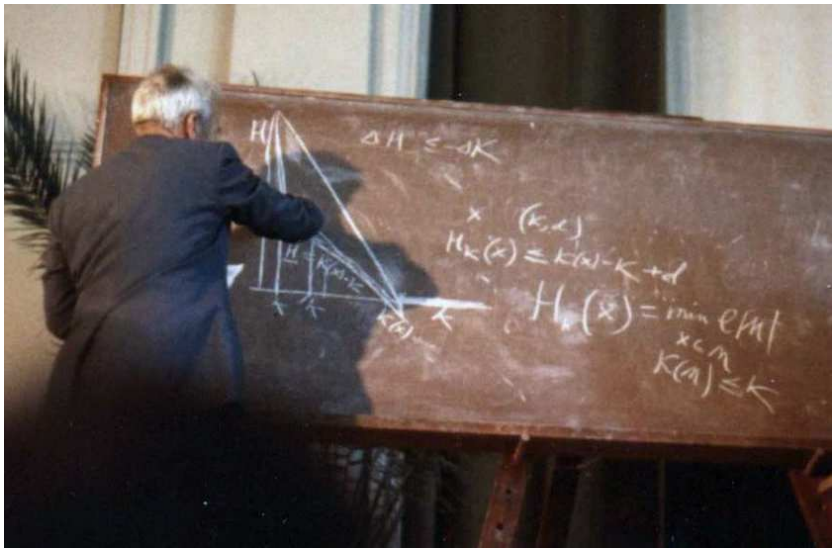
To solve:

$$Lw = w$$

where the entry L_{ij} is related to the number of links from node j to node i.

Computational science changed physics/engineering radically.

Before the 1970s the two pillars where *theory* and *experiment*.



Wikipedia, public domain.

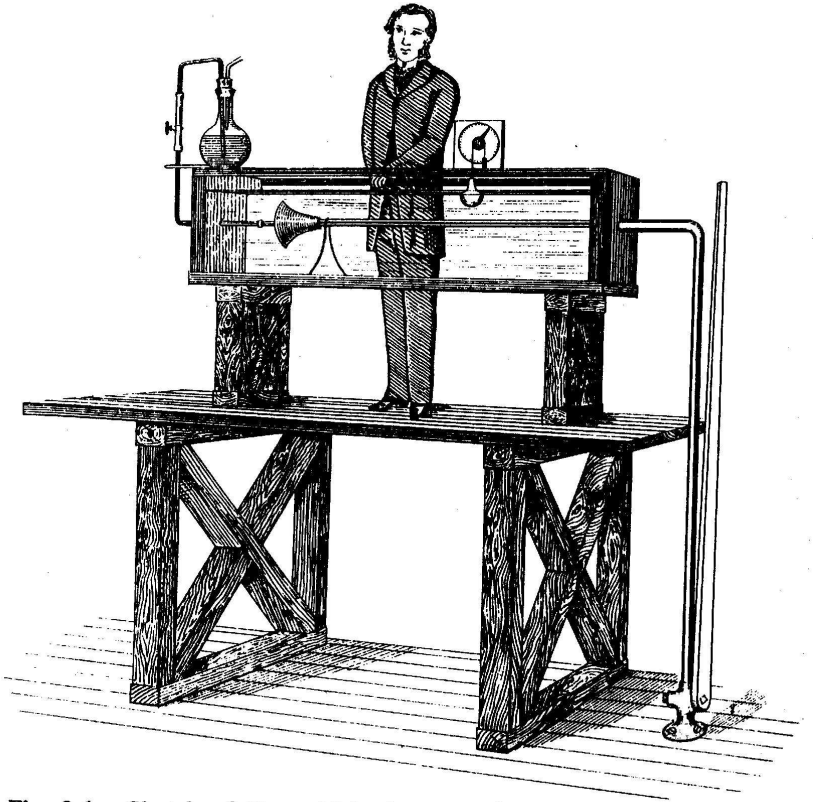


Fig. 9.1. Sketch of Reynolds's dye experiment, taken from his 1883 paper

In comparison, numerical approximation is often

- more feasible,
- easier to analyse,
- easier to manipulate,
- less expensive.

Figure 13: *Collapsed steel building during January 17,1995 Kobe Earthquake (by K. Meguro)*

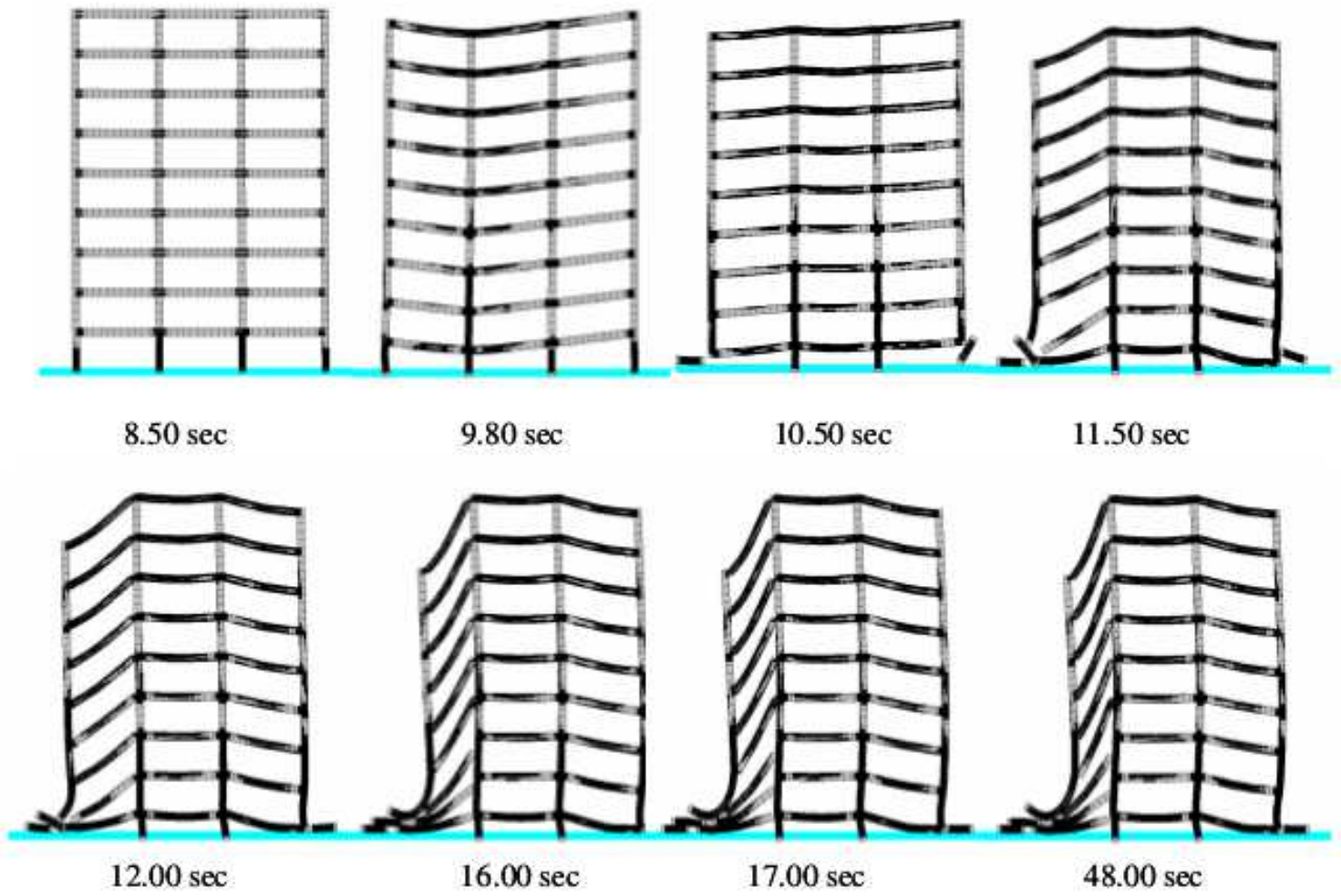
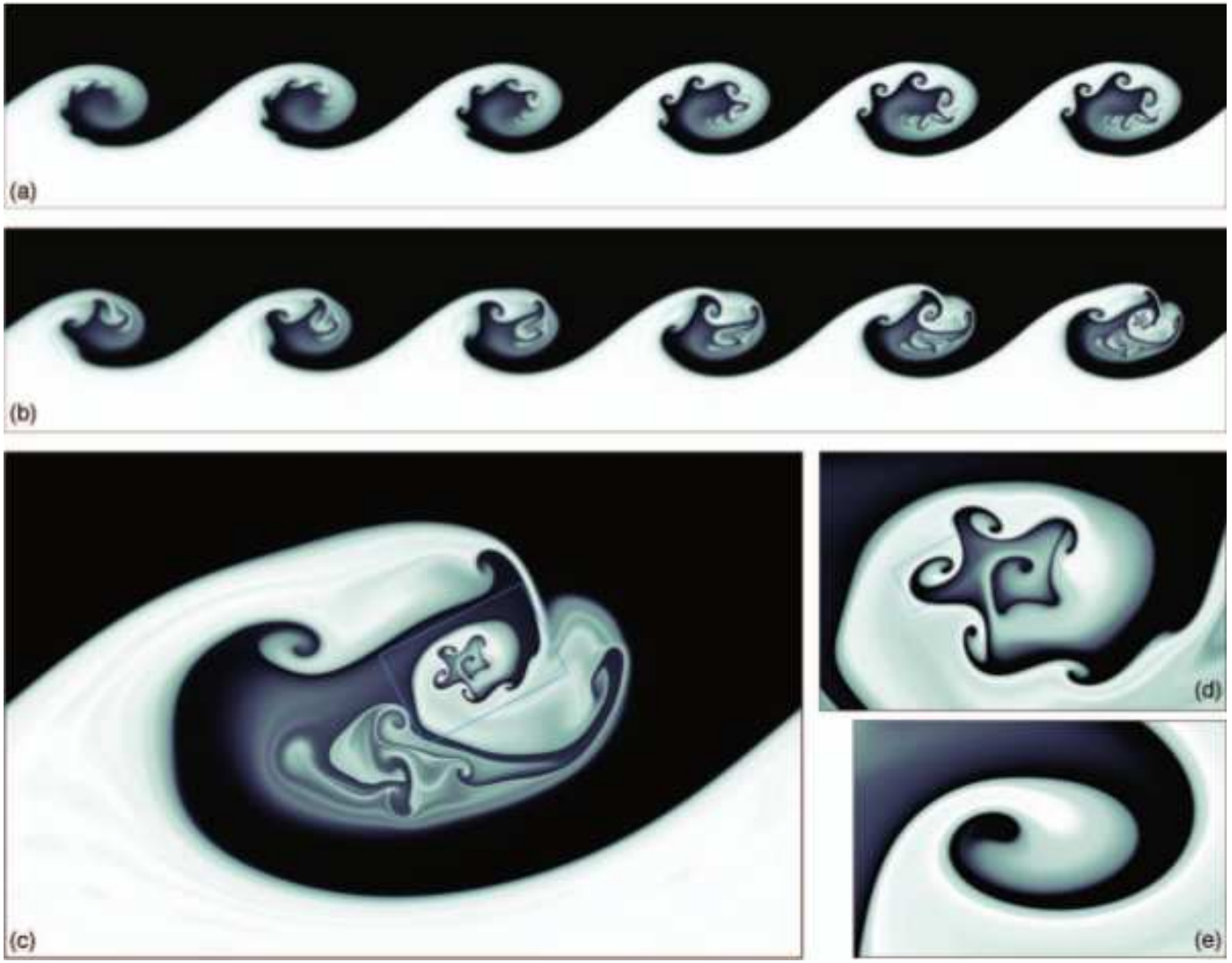
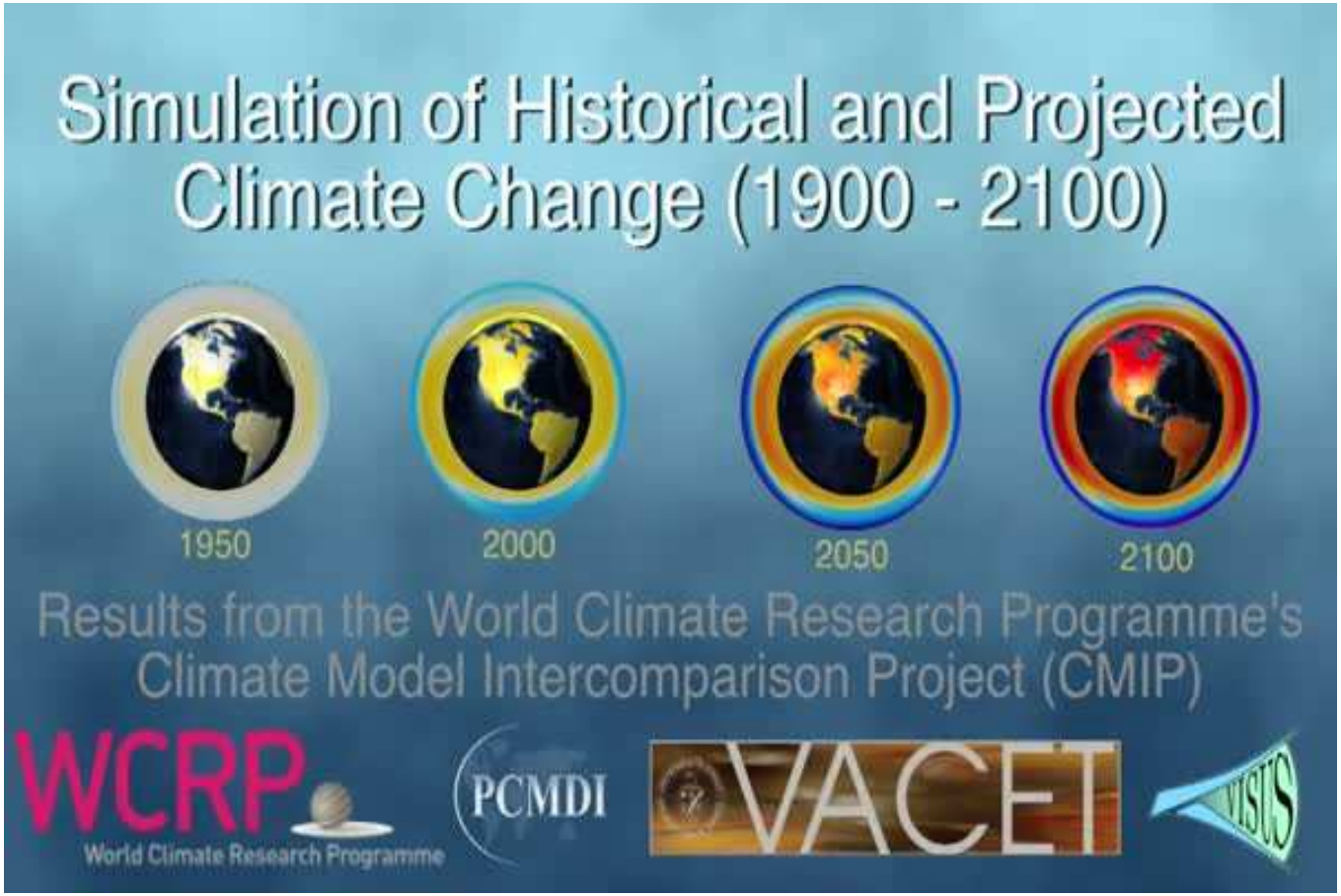


Figure 12: *Ground soft-story collapse mechanism*

Elkholy & Meguro, 13th World Congress on Earthquake Engineering



Fontane *et al.*, 25th Annual Gallery of Fluid Motion


Simulation of Historical and Projected Climate Change (1900 - 2100)
1950
2000
2050
2100
Results from the World Climate Research Programme's Climate Model Intercomparison Project (CMIP)
WCRP
World Climate Research Programme
PCMDI
VACET
VISUS

Symbolic computation: application of the rules of algebra and mathematical identities to manipulate an expression involving symbols and variables.
Example: $x \in \mathbb{R}$, $a \in \mathbb{R}$

$$\begin{aligned} 2\exp(-x^2 + a) = 1 &\Leftrightarrow \\ \ln(2) - x^2 + a = 0 &\Leftrightarrow \\ x = \pm\sqrt{\ln(2) + a} & \end{aligned}$$

Numerical computation: performing arithmetic on (approximate) numerical quantities. The result is a (set of) number(s) of finite precision.
Example: $x \in \mathbb{R}$, set $a = 2$

$$2\exp(-x^2 + 2) = x \;\Leftrightarrow\; \text{approximate } x \text{ from a sequence}$$
$$x_0 = 1.\underline{5},\; x_1 = 1.\underline{510}15398,\; x_2 = 1.\underline{51025458}4,\; \ldots$$

- MAPLE and MATHEMATICA are symbolic computation programs ("Computer Algebra Systems" or CAS) – but they can do some numerical computation, too.

- MAPLE and MATHEMATICA are symbolic computation programs ("Computer Algebra Systems" or CAS) – but they can do some numerical computation, too.
- Python and MATLAB are best suited for numerical computation – but can do some symbolic computation, too.

- MAPLE and MATHEMATICA are symbolic computation programs ("Computer Algebra Systems" or CAS) – but they can do some numerical computation, too.
- Python and MATLAB are best suited for numerical computation – but can do some symbolic computation, too.
- Lower-level programming languages like FORTRAN, C and C++ are commonly used for numerical computation and are generally faster than Python/Matlab.

- MAPLE and MATHEMATICA are symbolic computation programs ("Computer Algebra Systems" or CAS) – but they can do some numerical computation, too.
- Python and MATLAB are best suited for numerical computation – but can do some symbolic computation, too.
- Lower-level programming languages like FORTRAN, C and C++ are commonly used for numerical computation and are generally faster than Python/Matlab.
- Symbolic computations are exact.

- MAPLE and MATHEMATICA are symbolic computation programs ("Computer Algebra Systems" or CAS) – but they can do some numerical computation, too.
- Python and MATLAB are best suited for numerical computation – but can do some symbolic computation, too.
- Lower-level programming languages like FORTRAN, C and C++ are commonly used for numerical computation and are generally faster than Python/Matlab.
- Symbolic computations are exact.
- Numerical computations are approximate.

- MAPLE and MATHEMATICA are symbolic computation programs ("Computer Algebra Systems" or CAS) – but they can do some numerical computation, too.
- Python and MATLAB are best suited for numerical computation – but can do some symbolic computation, too.
- Lower-level programming languages like FORTRAN, C and C++ are commonly used for numerical computation and are generally faster than Python/Matlab.
- Symbolic computations are exact.
- Numerical computations are approximate.
- Which is appropriate depends entirely on the context.

Definition (Operating System)

The *Operating System* (OS) is the foundation systems software that controls the execution of computer programs and provides services like operating the internet connection, outputting to the screen, etc.

Examples: Unix, **Linux**, Mac OS, MS-DOS, Windows, etc.

Why Linux?

- It is free and always available.
  Your work will be portable, re-usable and independent of your environment.
- It is open-source.
- It is often faster.
- It is intended for scientific computing.
  Demo 1 (see the Top 500 Fastest Supercomputers).
- It strengthens your CV.
  Demo 2 (job listings on Workopolis).

Definition (Compiler)

A *compiler* is a computer program that reads high-level programming language and translates it to a low-level (machine) language (also called object code). The object code output by a compiler is typically put into a file called an executable that can be run (executed) at a later time.

Definition (Interpreter)

An *interpreter* is a computer program that reads input code in some high-level programming language and immediately executes the input program.
A programming language executed by an interpreter is called an interpreted programming language (contrast with compiled programming language).

- Interpreted code generally runs slower than compiled code because the interpreter reads and executes each statement as it goes along while the executable machine code output from a compiler need only execute each statement.

- Interpreted code generally runs slower than compiled code because the interpreter reads and executes each statement as it goes along while the executable machine code output from a compiler need only execute each statement.
- When testing/developing code, interpreted languages are usually easier.

- Interpreted code generally runs slower than compiled code because the interpreter reads and executes each statement as it goes along while the executable machine code output from a compiler need only execute each statement.
- When testing/developing code, interpreted languages are usually easier.
- Programs like MATLAB & MAPLE are interpreters. They allow for development of scientific algorithms with lots of debugging help.

- Interpreted code generally runs slower than compiled code because the interpreter reads and executes each statement as it goes along while the executable machine code output from a compiler need only execute each statement.
- When testing/developing code, interpreted languages are usually easier.
- Programs like MATLAB & MAPLE are interpreters. They allow for development of scientific algorithms with lots of debugging help.
- When developing a complicated program, we often use an interpreted language first, and then translate into a lower-level language like FORTRAN or c.
- Python is mostly an interpreter, but often calls compiled libraries for the actual computations and can be fast.