RSE-QuantEcon
Computational Economics Workshop

An introduction to computational methods
for economics and finance

10:00am – 12:00am February 16th 2022

Personel

- Thomas J. Sargent (NYU)
- John Stachurski (RSE)

Topics

- Introduction to scientific computing
- Option pricing with Python
- Discussion of high performance computing
- Dynamic programming with Python

Acknowledgement of Country

We acknowledge and celebrate the First Australians on whose traditional lands we meet.

We pay our respect to the elders past and present.

Assumptions:

- econ/computer/maths/stats literate
- some basic familiarity with computers

Aims:

- Discuss options
- Review trends
- Learn techniques

Resources

- https://github.com/QuantEcon/rse_comp_econ_2023

Background: Language and Platforms

Proprietary

- Excel
- MATLAB, Mathematica
- STATA, Eviews, SPSS.

Open Source

- Python
- Julia
- R

closed and stable vs open and fast moving

Background — Language Types

Low level

- C/C++
- Fortran
- Assembly

High level

- Python
- Ruby

Low level languages give us fine grained control

Example. $1 + 1$ in assembly

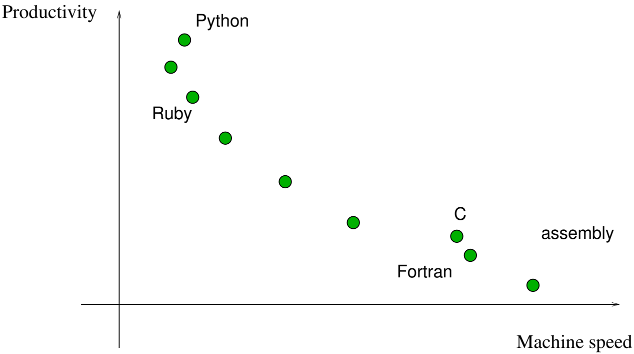
```
pushq   %rbp
movq    %rsp, %rbp
movl    $1, -12(%rbp)
movl    $1, -8(%rbp)
movl    -12(%rbp), %edx
movl    -8(%rbp), %eax
addl    %edx, %eax
movl    %eax, -4(%rbp)
movl    -4(%rbp), %eax
popq    %rbp
```

High level languages give us abstraction, automation, etc.

Example. Reading from a file in Python

```
data_file = open("data.txt")
for line in data_file:
    print(line.capitalize())
data_file.close()
```

Trade-Offs

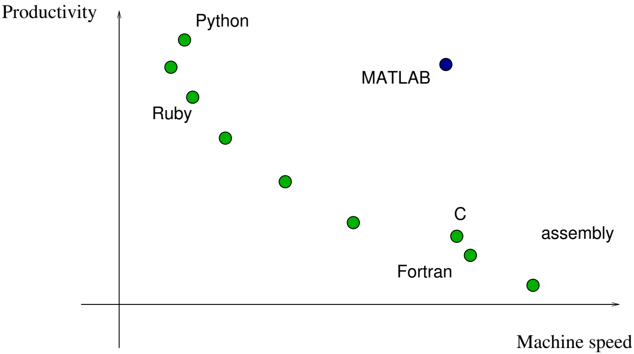

Productivity
Python
Ruby
C
assembly
Fortran
Machine speed

But what about scientific computing?

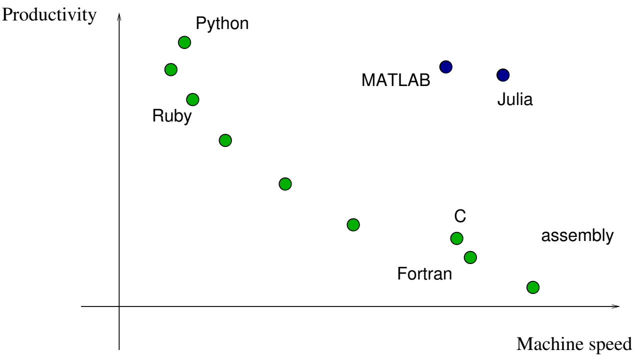
Requirements

- Productive — easy to read, write, debug, explore
- Fast computations

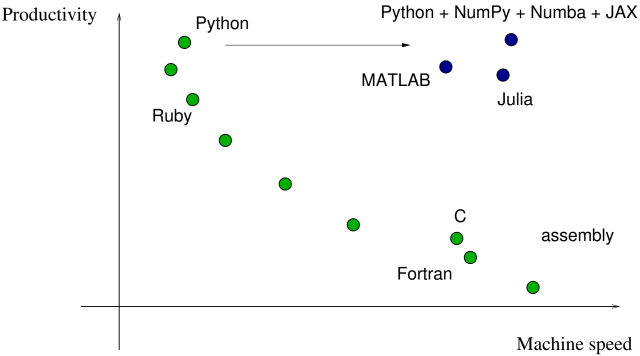
Trade-Offs


Productivity
Python
MATLAB
Ruby
C
assembly
Fortran
Machine speed

Trade-Offs

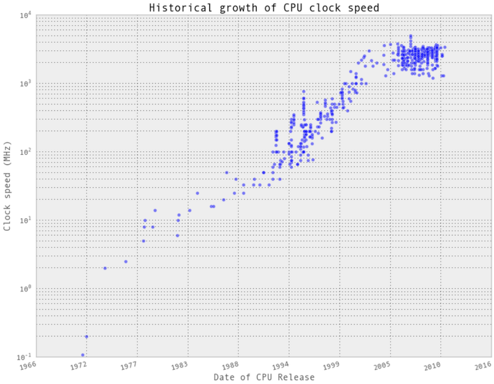

Productivity
Python
MATLAB
Julia
Ruby
C
assembly
Fortran
Machine speed

Trade-Offs

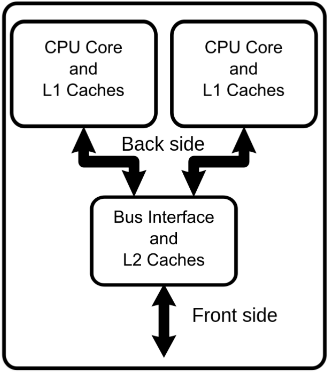

Productivity
Python
Python + NumPy + Numba + JAX
MATLAB
Julia
Ruby
C
assembly
Fortran
Machine speed

Parallelization

CPU frequency (clock speed) growth is slowing

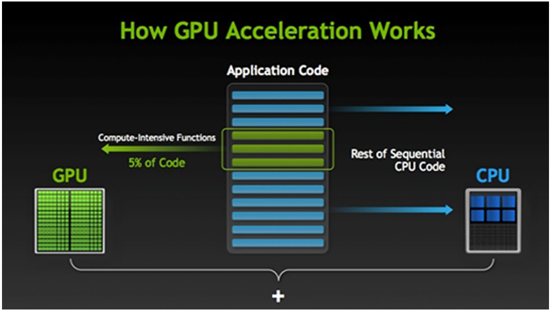


Chip makers have responded by developing multi-core processors



Source: Wikipedia

GPUs are becoming increasingly important


How GPU Acceleration Works
Application Code
Compute-Intensive Functions
5% of Code
GPU
Rest of Sequential CPU Code
CPU
+

Applications: machine learning, deep learning, etc.

Support for Parallelization

While scientific computing environments best support parallelization?

Parallelization requires different algorithms

- all have some support
- but which make it easy to harness its power?

Current winner:

- Google JAX (Python library)

Which Language

How about R?

- Specialized to statistics
- Huge range of estimation routines
- Popular in academia
- Loosing some ground to Python (AI, machine learning)

Julia

Pros:

- Fast and elegant
- Many scientific routines
- Julia is written in Julia

Cons:

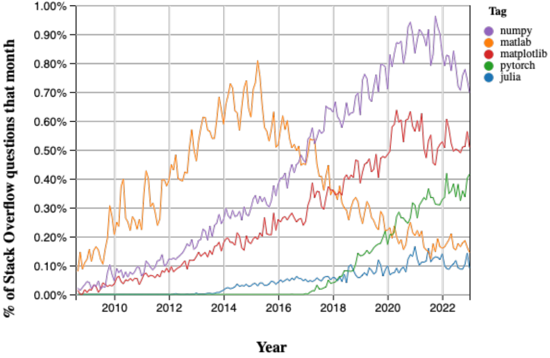
- Some stability issues
- Low rates of investment in some important libraries

Python

- Easy to learn, well designed
- Massive scientific ecosystem
- Heavily supported by big players
- Open source
- Huge demand for tech-savvy Python programmers

Example. Largest share of OpenAI (ChatGPT) code

Popularity:


% of Stack Overflow questions that month
1.00%
0.90%
0.80%
0.70%
0.60%
0.50%
0.40%
0.30%
0.20%
0.10%
0.00%
2010
2012
2014
2016
2018
2020
2022
Year
Tag
● numpy
● matlab
● matplotlib
● pytorch
● julia

Downloads / Installation / Troubleshooting

Install Python + Scientific Libs (Optional!)

- Install Anaconda from `https://www.anaconda.com/`
  - Select latest version
  - For your OS
  - Say “yes” at prompts
- Not plain vanilla Python

Remote options

- `https://colab.research.google.com`
- `https://www.pythonanywhere.com/`

Jupyter Notebooks

A browser based interface to Python / Julia / R / etc.

- Search for `jupyter notebook`

Useful for:

- getting started
- exploring ideas

Working with Notebooks

- Entry and execution
- Markdown
- Getting help
- Copy paste
- Edit and command mode