

Topics

- Discussion of scientific computing
- Overview of Python libraries
- Computing equilibria
- Google JAX
- Option pricing with Python
- High dimensional problems

Assumptions:

- basic econ/computer/maths/stats
- some programming?

Aims:

- Discuss options
- Review trends
- Learn techniques

Resources

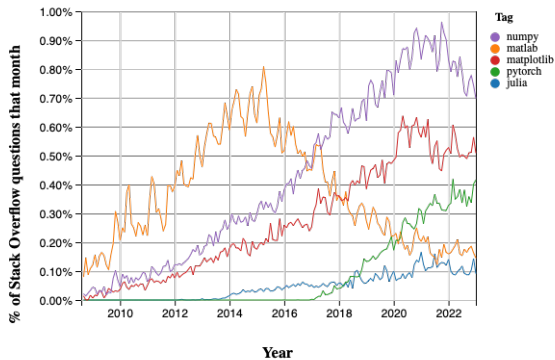
- https://github.com/QuantEcon/kobe_comp_econ_2023

Trends

What are the major trends in scientific computing?

- what's driving them?
- how can we benefit?

Popularity:



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
```

Example. 1 + 1 in C

```
#include <stdio.h>
int main() {
    int x = 1 + 1;
    printf("1 + 1 = %d\n", x);
    return 0;
}
```

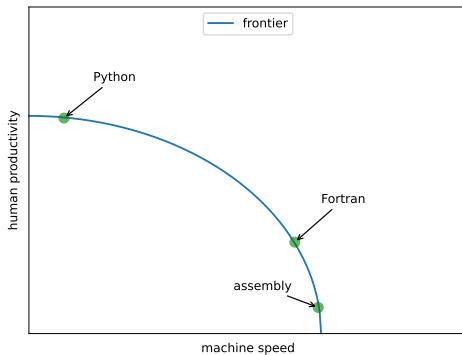
Example. $1 + 1$ in Fortran

```
PROGRAM ONE_PLUS_ONE  
INTEGER :: X = 1 + 1  
PRINT *, '1 + 1 = ', X  
END PROGRAM ONE_PLUS_ONE
```

Example. 1 + 1 in Python

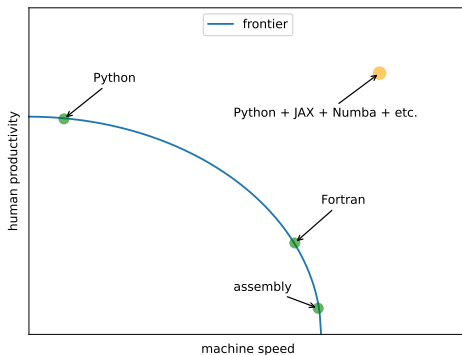
```
x = 1 + 1  
print("1 + 1 = ", x)
```

Trade-Offs:



New trend — a shifting frontier

Trade-offs:



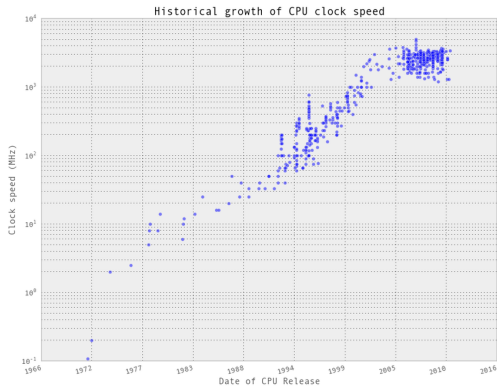
Example. What platforms/languages does OpenAI use?

In order (according to [repo stats](#)):

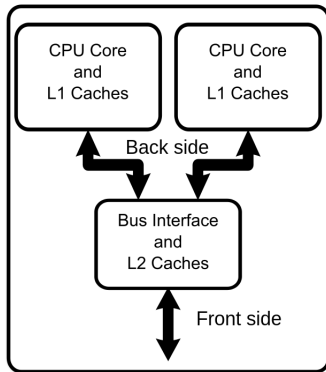
1. Python
2. C++
3. Javascript
4. Jupyter notebooks
5. Ruby

Trend 3: Parallelization

CPU frequency (clock speed) growth is slowing



Chip makers have responded by developing multi-core processors



Source: Wikipedia

GPUs are becoming increasingly important



Applications: machine learning, deep learning, etc.

Support for Parallelization

While scientific computing environments best support parallelization?

- Most 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)

How to Interact with Python?

Many options:

- **write** with VS Code / Emacs / Vim
- **run** with base Python, IPython, etc.

Or do both with **Jupyter notebooks / Jupyter lab**

- for simplicity we focus only on the last option

