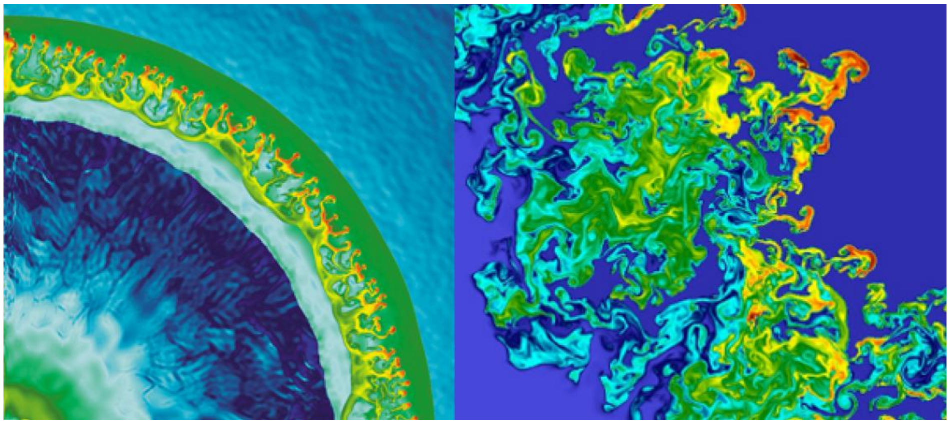
Computer Simulations in Physics



PHYS 409: Computational Methods in Physics

Three Ways of Doing Physics

Theory

$$\nabla \cdot \mathbf{D} = \rho_V$$

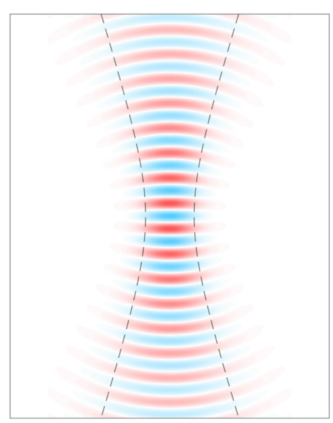
$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

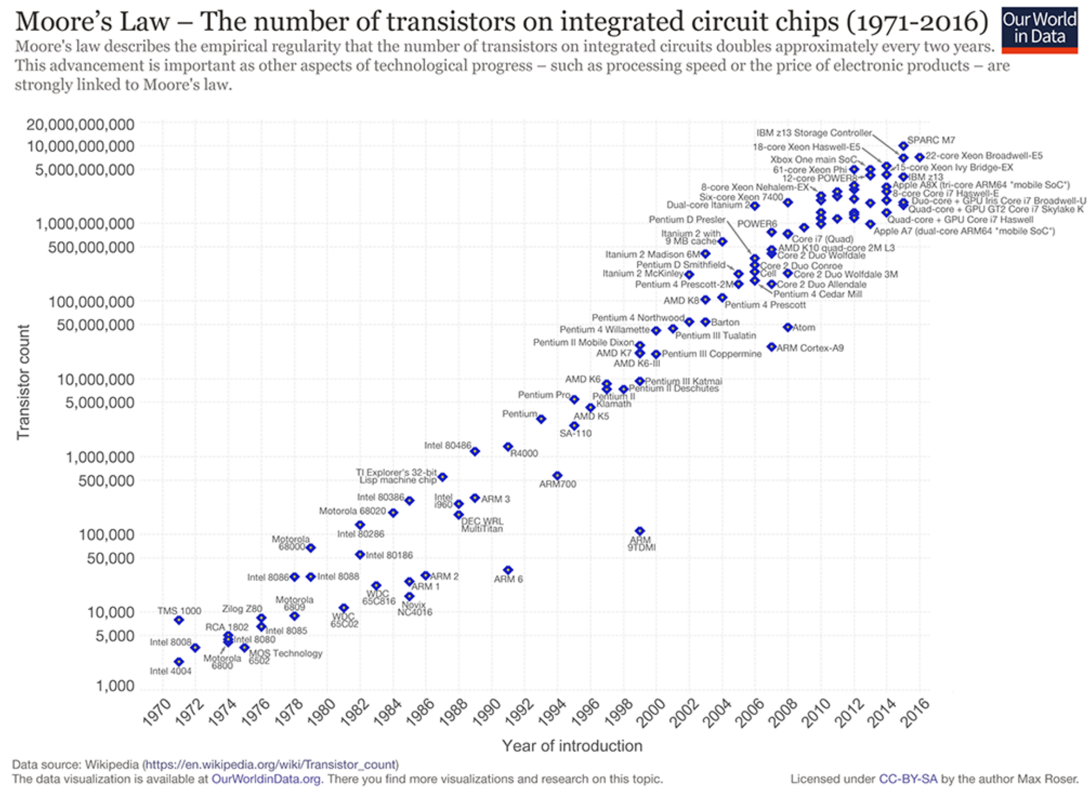
$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}$$

Experiment

Simulation

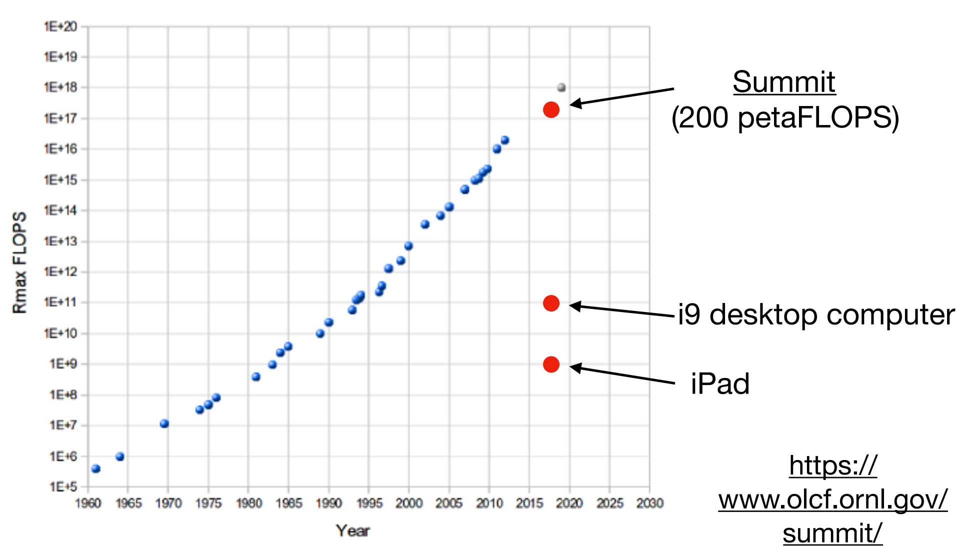


Moore's Law (Transistor Count)



Moore's Law (FLOPS)

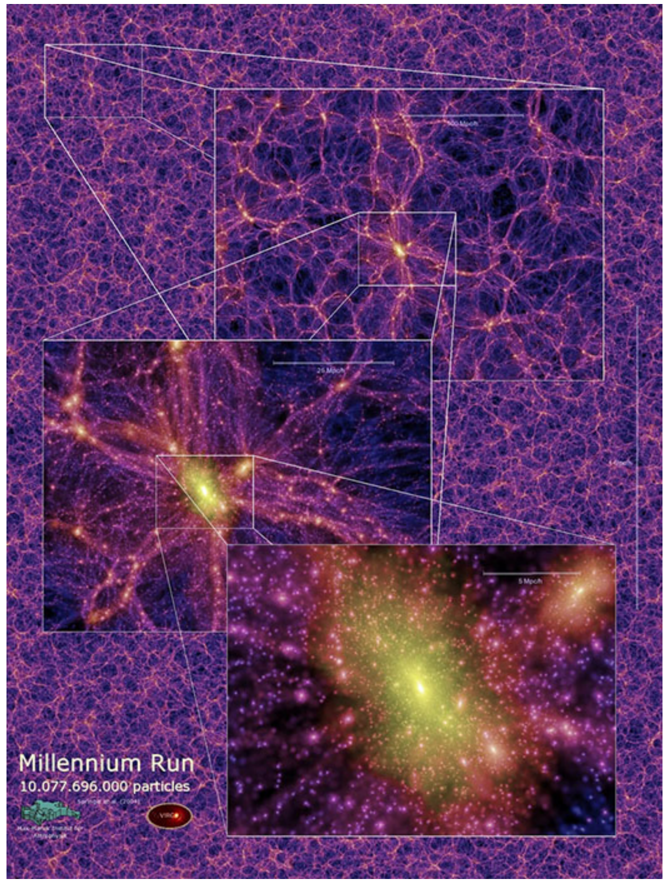
Number of Floating Point Operations (FLOPS) performed by the fastest supercomputer doubles every 1-2 years.



https://www.olcf.ornl.gov/summit/

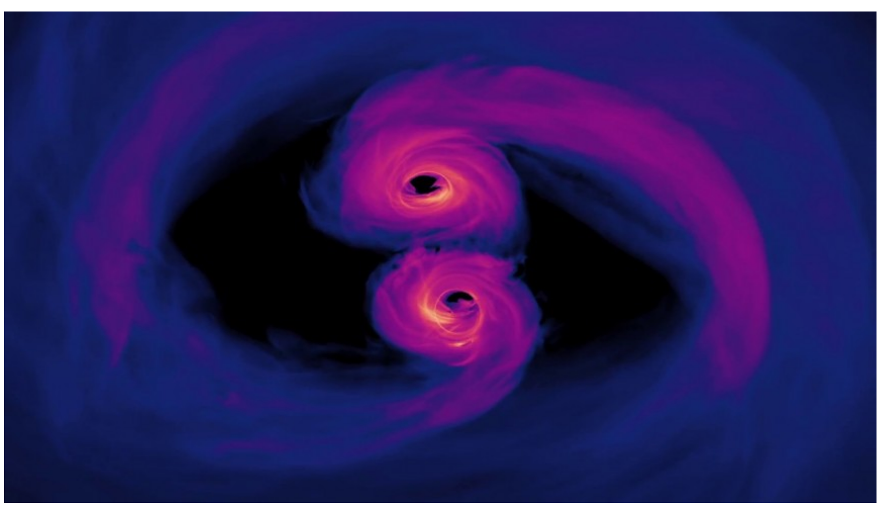
Cosmology: Millenium Simulation

- 10 billion particles
- 20 million galaxies
- 13.6 billion years



https://wwwmpa.mpa-garching.mpg.de/galform/virgo/millennium/

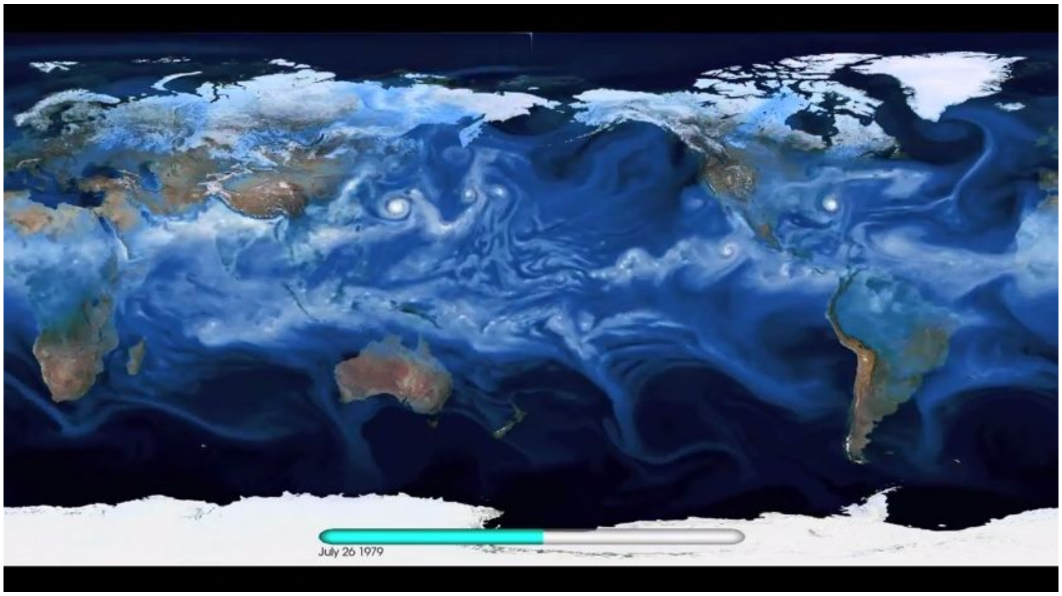
Astrophysics: Simulation of 2 Colliding Black Holes



- NASA/Goddard

https://www.nasa.gov/feature/goddard/2018/new-simulation-sheds-light-on-spiraling-supermassive-black-holes
https://www.youtube.com/watch?v=i2u-7LMhwvE

Climate Science: Global Circulation Models of the Earth

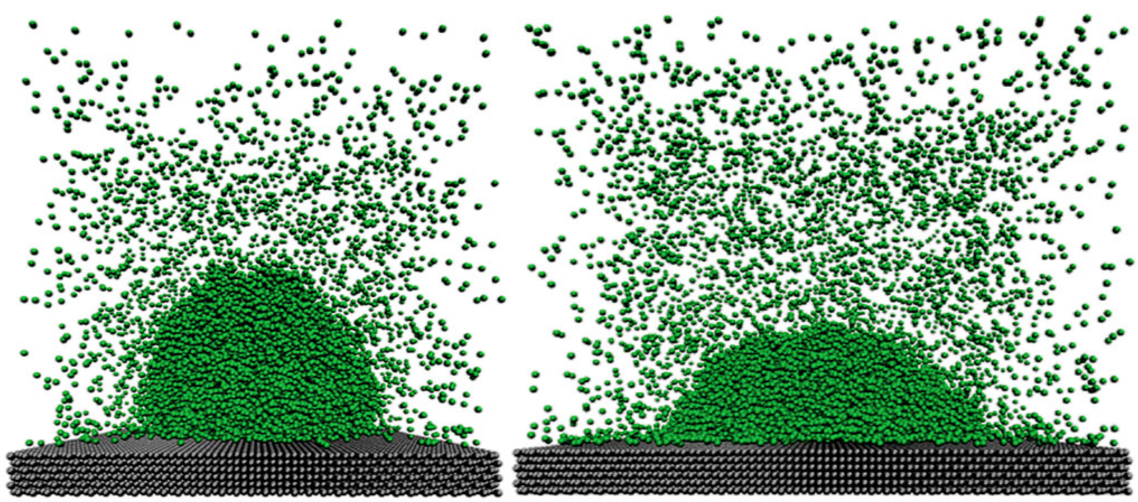


- Berkeley Labs NERSC
- Brookhaven E3SM sim.

https://phys.org/news/2014-11-latest-supercomputers-enable-high-resolution-climate.html

https://sciencesprings.wordpress.com/2018/04/27/from-brookhaven-lab-new-high-resolution-exascale-earth-modeling-system-for-energy/

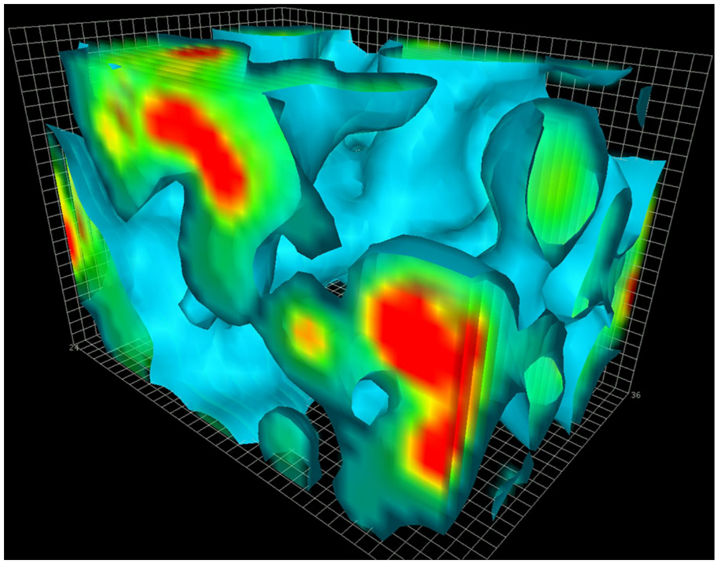
Molecular Dynamics: Evaporation of a Nanoscale Liquid Drop



Gauss Center for Supercomputing, Germany

http://www.gauss-centre.eu/gauss-centre/EN/Projects/MaterialsScienceChemistry/2016/horsch_fluids_at_interfaces.html?nn=1236240

High Energy Physics: QCD Simulations of Empty Space



QCD simulations (1fm)

http://www.physics.adelaide.edu.au/theory/staff/leinweber/

Why Simulations?

- Sometimes it is not possible to perform experiments (astrophysics, geophysics)
- Sometimes the theory is too complex to obtain analytic solutions (nonlinear problems with many degrees of freedom)
- Simulations can be cost effective (running 10 simulations might be cheaper than building 10 experiments)

How Trustworthy are Simulations?

A JPL study estimated that the number of computer “bugs” in a program decreases exponentially with time spent debugging.

This means you can never be sure your code is completely free of bugs 😫.

Always test your code!! Verification is Essential!!

- Run code for problems with known solutions.
- Test subcomponents of the code on a wide range of simple problems to make sure they behave appropriately.
- Monitor conserved quantities to make sure they stay conserved.
- Try to think up as many “reality checks” as you possibly can.

Computational Tools

Symbolic Calculations:

- Maple
- Mathematica

General Purpose Interpretive languages:

- Matlab - engineering, science, economics (nice integrated interface, but expensive)
- Python - open source, object-oriented (growing in popularity)

Low-Level compiled languages:

- C - custom applications (fluid flow, plasmas, EM,…)
- Fortran - mostly