Chesick Scholars - Intro to computer programming in STEM

Prof. Dan Grin

Deep Patel

Where

H204 - SCIENCE BUILDING

When

Sunday 8/21 9-10:30 AM

Monday 8/22 1-2:30 PM

Tuesday 8/23 1-2:30 PM

Wednesday 8/24 11-11:30AM

Mask please

Extra masks available at front of room

More about me

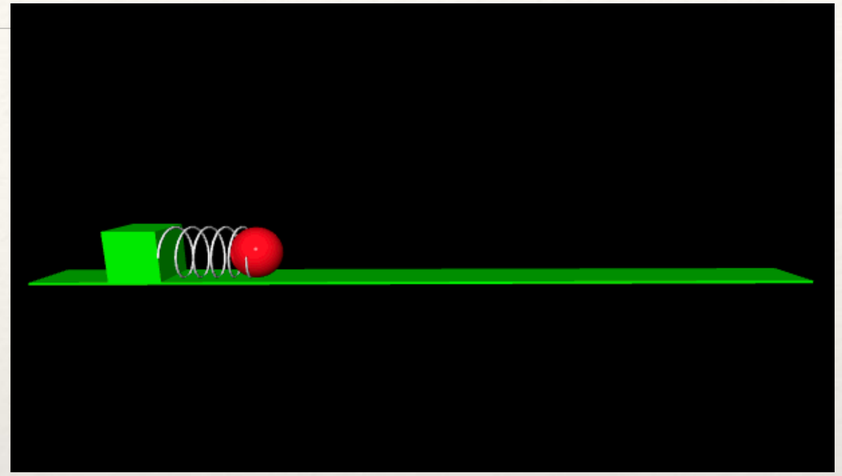
Welcome

- Grab notecard with your name and login -info
- Make name tag with name and pronouns
- Say hi to your neighbor
- Go around circle, show and recite tag - one cool fact about yourself

Agenda

- Why programming?
- Who programming?
- Coding basics?
- Intro examples + visualization
- Variables
- Loops and other program flow controls
- Ball example
- Planet example mini-project

Why/What?

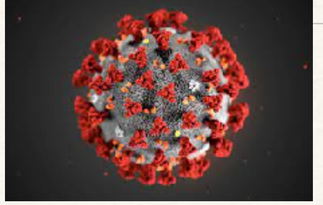


- Useful skill for jobs/research/lab
- Relevant to intro lab

- Computer programming to simulate solutions science questions
- VPython — a form of Python, no previous experience assumed

Galaxy collisions, dynamics formation

N~10^9 particles - many pairs, need Fourier or other techniques
Movie courtesy of Volker Springel



Covid-19

https://link.springer.com/content/pdf/10.1007/s11071-020-05743-y.pdf

https://link.springer.com/content/pdf/10.1140/epjp/s13360-020-00862-2.pdf

https://royalsocietypublishing.org/doi/pdf/10.1098/rsos.191187

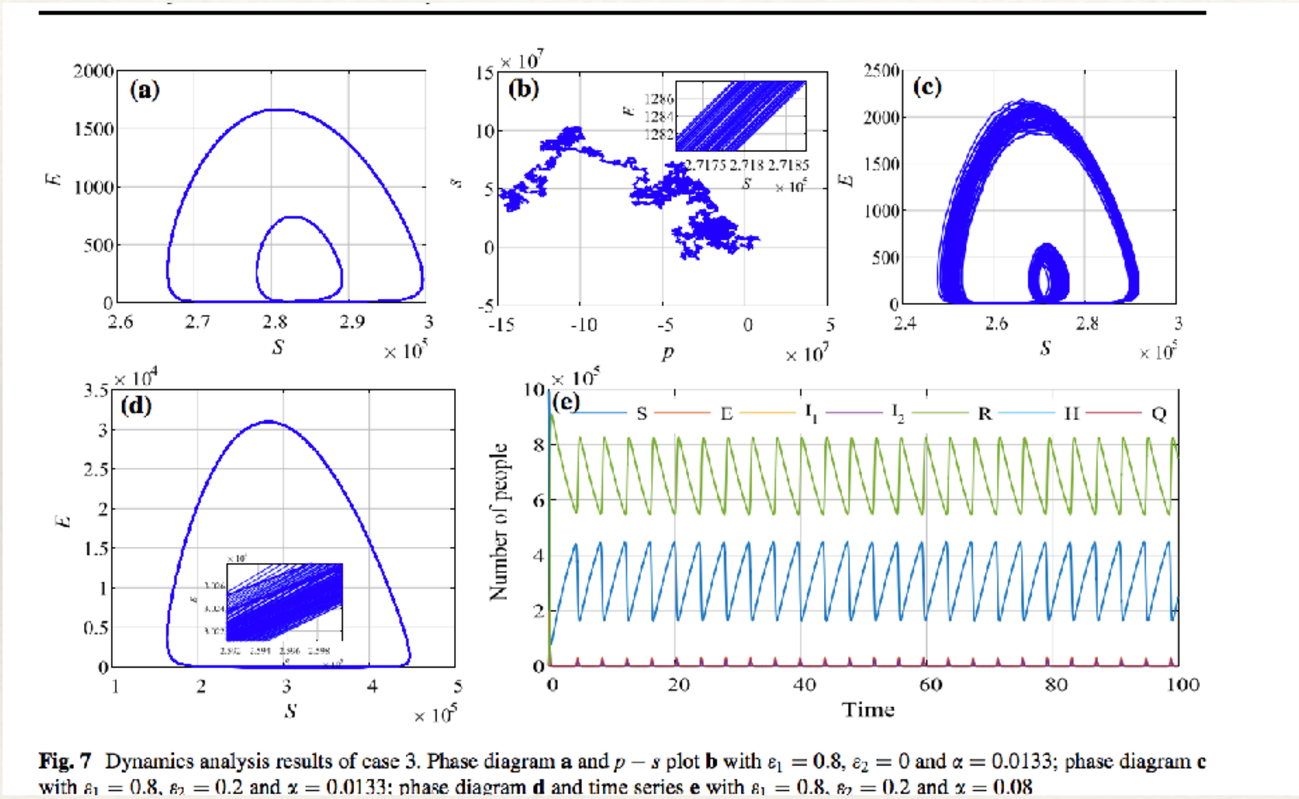
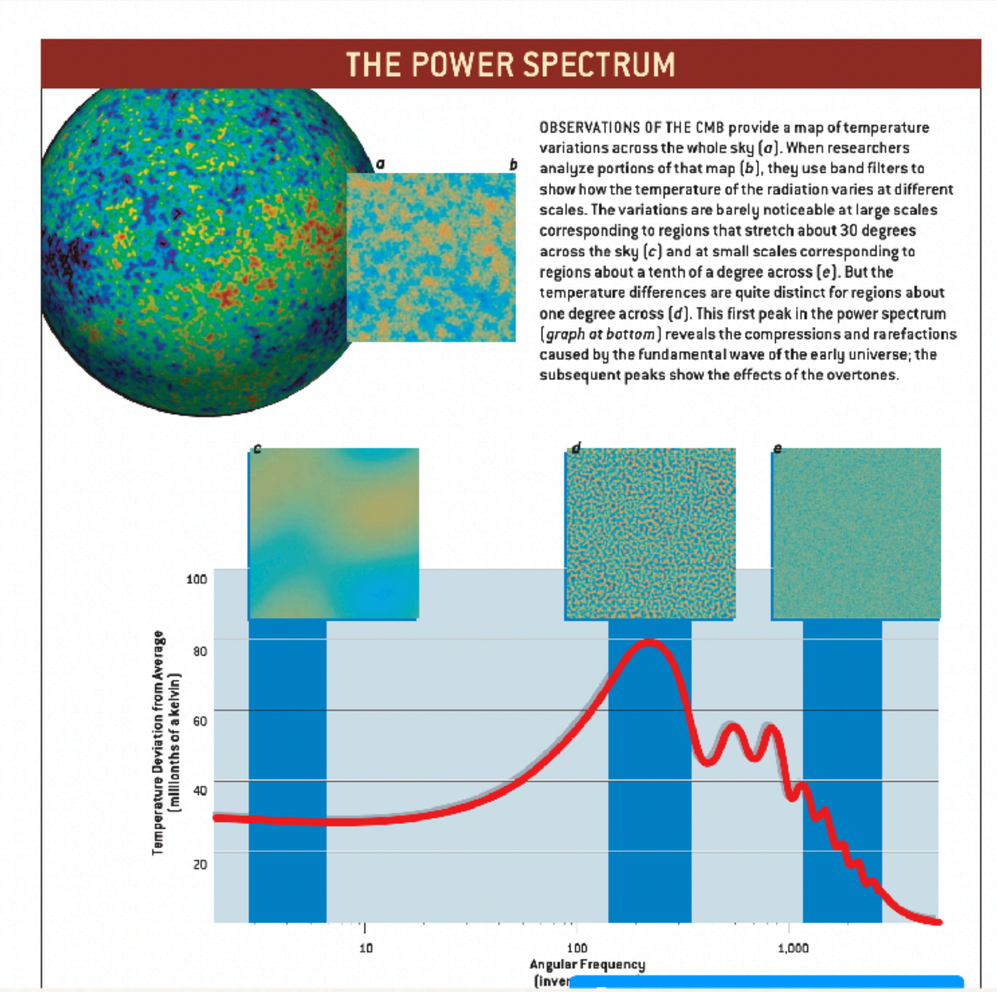


Fig. 7 Dynamics analysis results of case 3. Phase diagram **a** and $p - s$ plot **b** with $\varepsilon_1 = 0.8$, $\varepsilon_2 = 0$ and $\alpha = 0.0133$; phase diagram **c** with $\varepsilon_1 = 0.8$, $\varepsilon_2 = 0.2$ and $\alpha = 0.0133$; phase diagram **d** and time series **e** with $\varepsilon_1 = 0.8$, $\varepsilon_2 = 0.2$ and $\alpha = 0.08$

https://towardsdatascience.com/social-distancing-to-slow-the-coronavirus-768292f04296

Cosmic microwave background

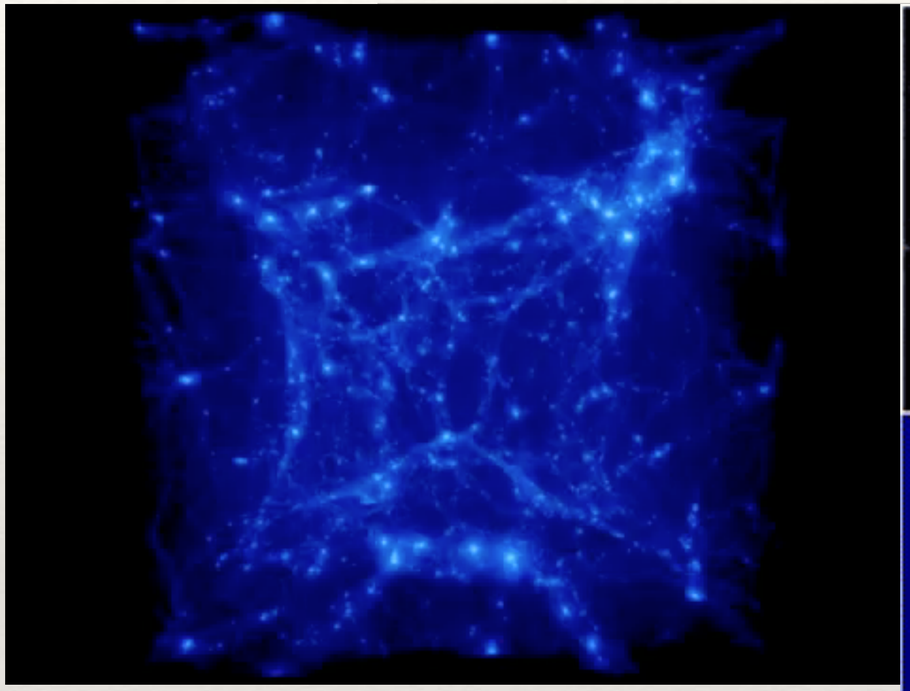


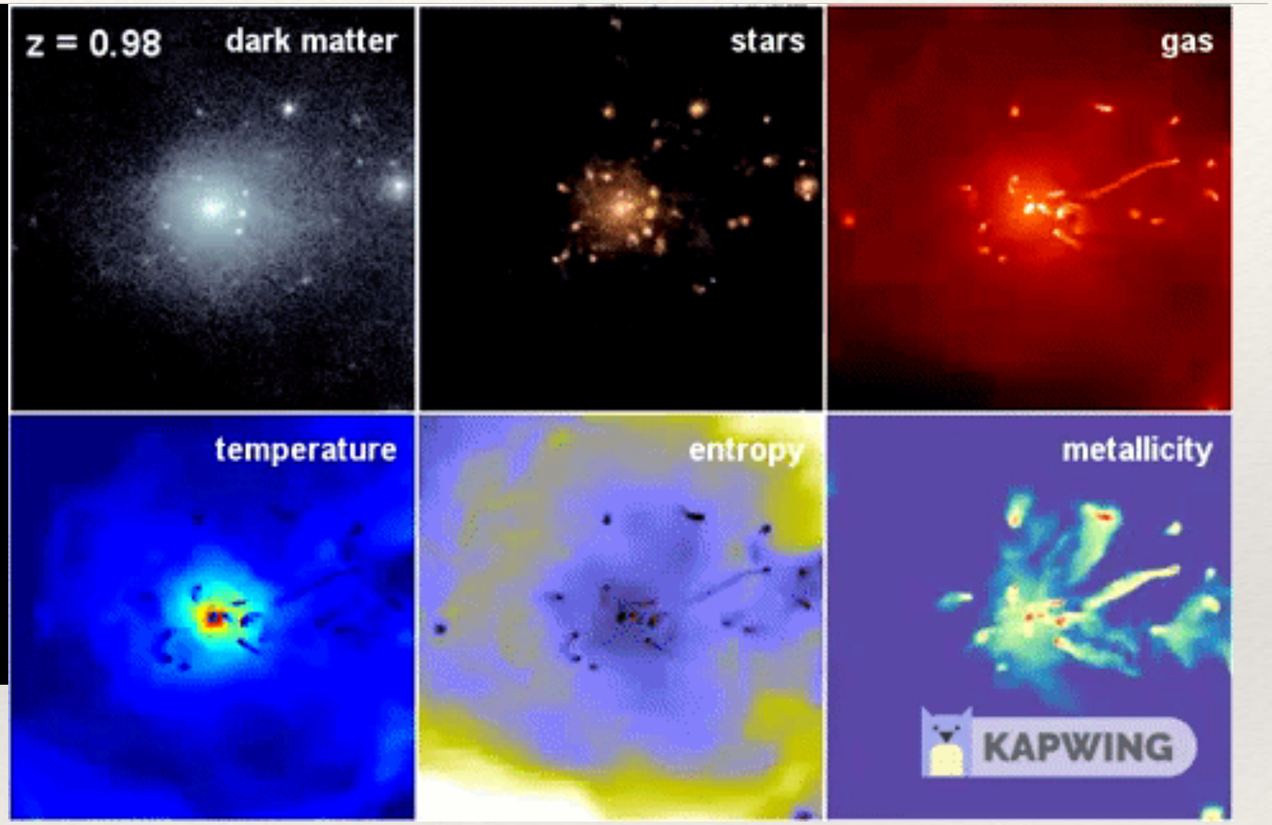
Wayne Hu, U Chicago

Uros Seljak, Berkeley

Marc Kamionkowski, JHU

Cosmological n-body simulations



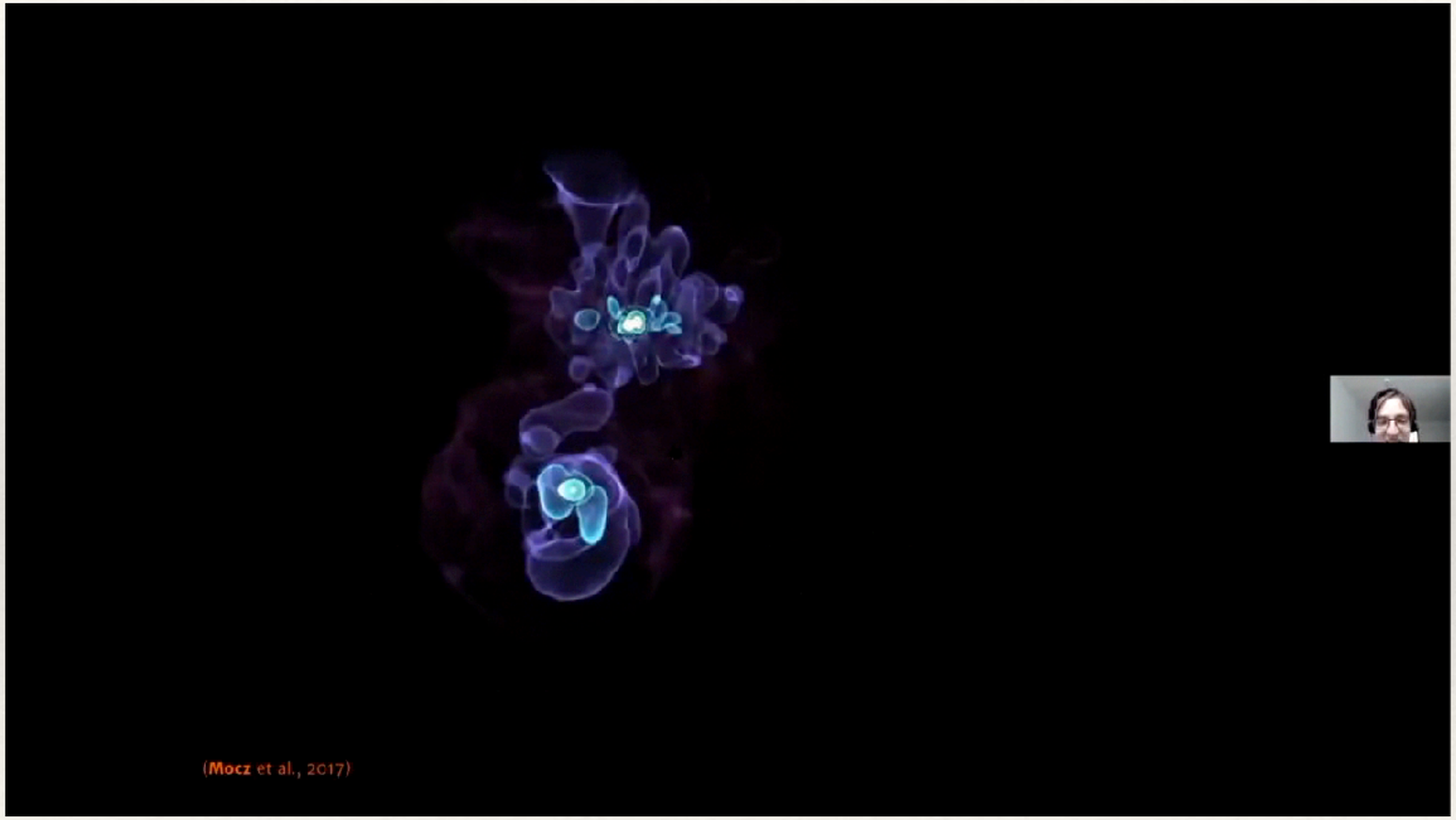


Camille Avestruz, U Mich

N~10^9 particles - many pairs, need Fourier or other techniques

Simulation courtesy of Ryan Wilkinson and Celine Boehm, University of Durham (UK)

Fuzzy dark matter



Simulation courtesy of P. Mocz, Lawrence Livermore National Labds

Numerical General Relativity

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} = 8\pi G T_{\mu\nu}$$

Baumgarte 2013

$$\begin{aligned} d_t \gamma_{ij} &= -2\alpha K_{ij} \\ d_t K_{ij} + \alpha\gamma^{kl}\partial_l f_{kij} &= \alpha M_{ij} \\ d_t f_{kij} + \alpha\partial_k K_{ij} &= \alpha N_{kij}. \end{aligned} \tag{4.12}$$

Here we have used the abbreviation

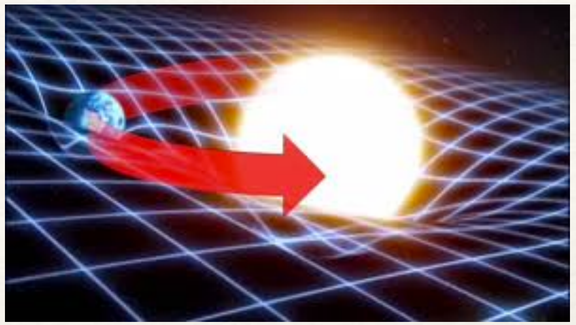
$$d_t \equiv \partial_t - \mathcal{L}_\beta \tag{4.13}$$

and the source terms M_{ij} and N_{ijk} are given by

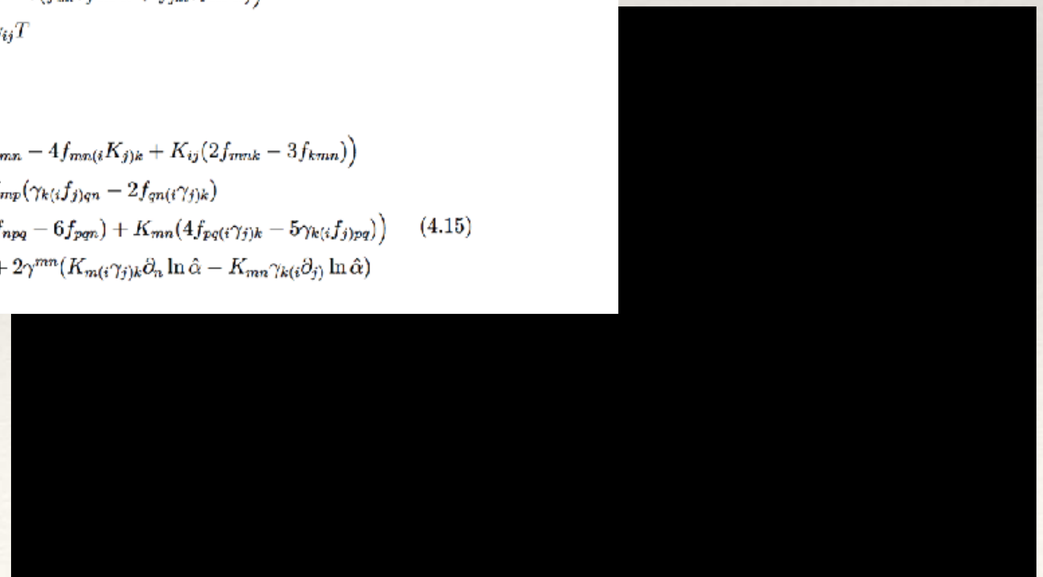
$$\begin{aligned} M_{ij} = {} & \gamma^{kl}(K_{kl}K_{ij} - 2K_{ki}K_{lj}) + \gamma^{kl}\gamma^{mn}(4f_{kmi}f_{[ln]j} \\ & +4f_{km[n}f_{l]ij} - f_{ikm}f_{jln} + 8f_{(ij)k}f_{[ln]m} + 4f_{km(i}f_{j)ln} \\ & -8f_{kli}f_{mnj} + 20f_{kl(i}f_{j)mn} - 13f_{ikl}f_{jmn}) \\ & -\partial_i\partial_j \ln\hat{\alpha} - (\partial_i \ln\hat{\alpha})(\partial_j \ln\hat{\alpha}) + 2\gamma_{ij}\gamma^{kl}\gamma^{mn}(f_{kmn}\partial_l \ln\hat{\alpha} \\ & -f_{kml}\partial_n \ln\hat{\alpha}) + \gamma^{kl}\big((2f_{(ij)k} - f_{kij})\partial_l \ln\hat{\alpha} \\ & +4f_{kl(i}\partial_{j)} \ln\hat{\alpha} - 3(f_{ikl}\partial_j \ln\hat{\alpha} + f_{jkl}\partial_i \ln\hat{\alpha})\big) \\ & -8\pi S_{ij} + 4\pi\gamma_{ij}T \end{aligned} \tag{4.14}$$

and

$$\begin{aligned} N_{kij} = {} & \gamma^{mn}\big(4K_{k(i}f_{j)mn} - 4f_{mn(i}K_{j)k} + K_{ij}(2f_{mnk} - 3f_{kmn})\big) \\ & +2\gamma^{mn}\gamma^{pq}\big(K_{mp}(\gamma_{k(i}f_{j)qn} - 2f_{qn(i}\gamma_{j)k}) \\ & +\gamma_{k(i}K_{j)m}(8f_{npq} - 6f_{pqn}) + K_{mn}(4f_{pq(i}\gamma_{j)k} - 5\gamma_{k(i}f_{j)pq})\big) \\ & -K_{ij}\partial_k \ln\hat{\alpha} + 2\gamma^{mn}(K_{m(i}\gamma_{j)k}\partial_n \ln\hat{\alpha} - K_{mn}\gamma_{k(i}\partial_{j)} \ln\hat{\alpha}) \\ & +16\pi\gamma_{k(i}j_{j)}. \end{aligned} \tag{4.15}$$



Caltech/Cornell NR group



Numerical General Relativity

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} = 8\pi G T_{\mu\nu}$$

Baumgarte 2013

$$\begin{aligned} d_t \gamma_{ij} &= -2\alpha K_{ij} \\ d_t K_{ij} + \alpha\gamma^{kl}\partial_l f_{kij} &= \alpha M_{ij} \\ d_t f_{kij} + \alpha\partial_k K_{ij} &= \alpha N_{kij}. \end{aligned} \tag{4.12}$$

Here we have used the abbreviation

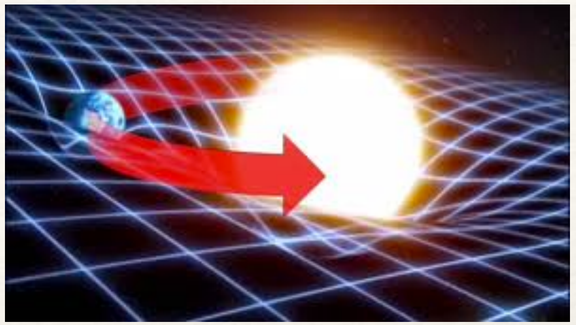
$$d_t \equiv \partial_t - \mathcal{L}_\beta \tag{4.13}$$

and the source terms M_{ij} and N_{ijk} are given by

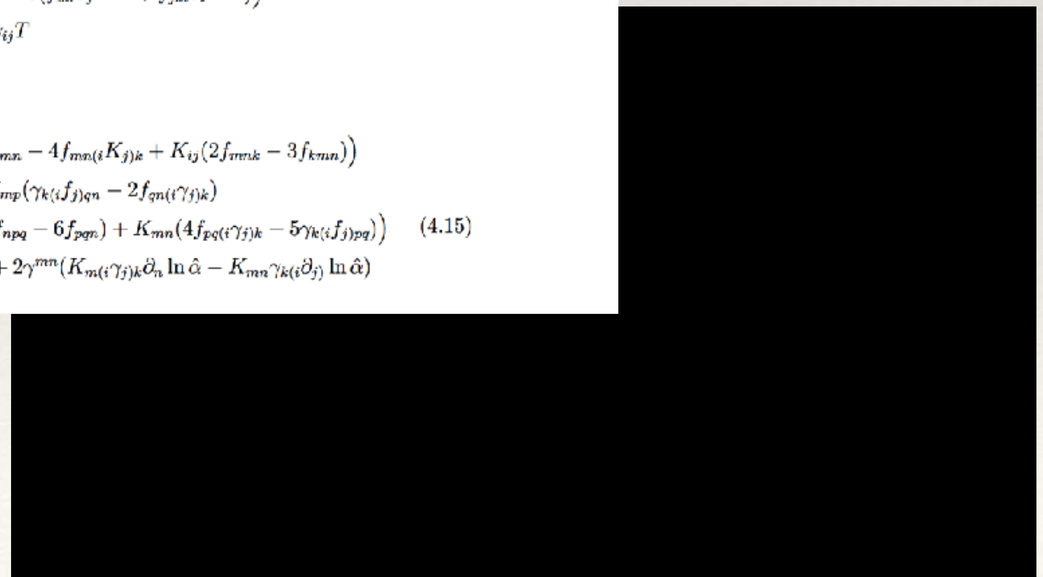
$$\begin{aligned} M_{ij} = {} & \gamma^{kl}(K_{kl}K_{ij} - 2K_{ki}K_{lj}) + \gamma^{kl}\gamma^{mn}(4f_{kmi}f_{[ln]j} \\ & +4f_{km[n}f_{l]ij} - f_{ikm}f_{jln} + 8f_{(ij)k}f_{[ln]m} + 4f_{km(i}f_{j)ln} \\ & -8f_{kli}f_{mnj} + 20f_{kl(i}f_{j)mn} - 13f_{ikl}f_{jmn}) \\ & -\partial_i\partial_j \ln\hat{\alpha} - (\partial_i \ln\hat{\alpha})(\partial_j \ln\hat{\alpha}) + 2\gamma_{ij}\gamma^{kl}\gamma^{mn}(f_{kmn}\partial_l \ln\hat{\alpha} \\ & -f_{kml}\partial_n \ln\hat{\alpha}) + \gamma^{kl}\big((2f_{(ij)k} - f_{kij})\partial_l \ln\hat{\alpha} \\ & +4f_{kl(i}\partial_{j)} \ln\hat{\alpha} - 3(f_{ikl}\partial_j \ln\hat{\alpha} + f_{jkl}\partial_i \ln\hat{\alpha})\big) \\ & -8\pi S_{ij} + 4\pi\gamma_{ij}T \end{aligned} \tag{4.14}$$

and

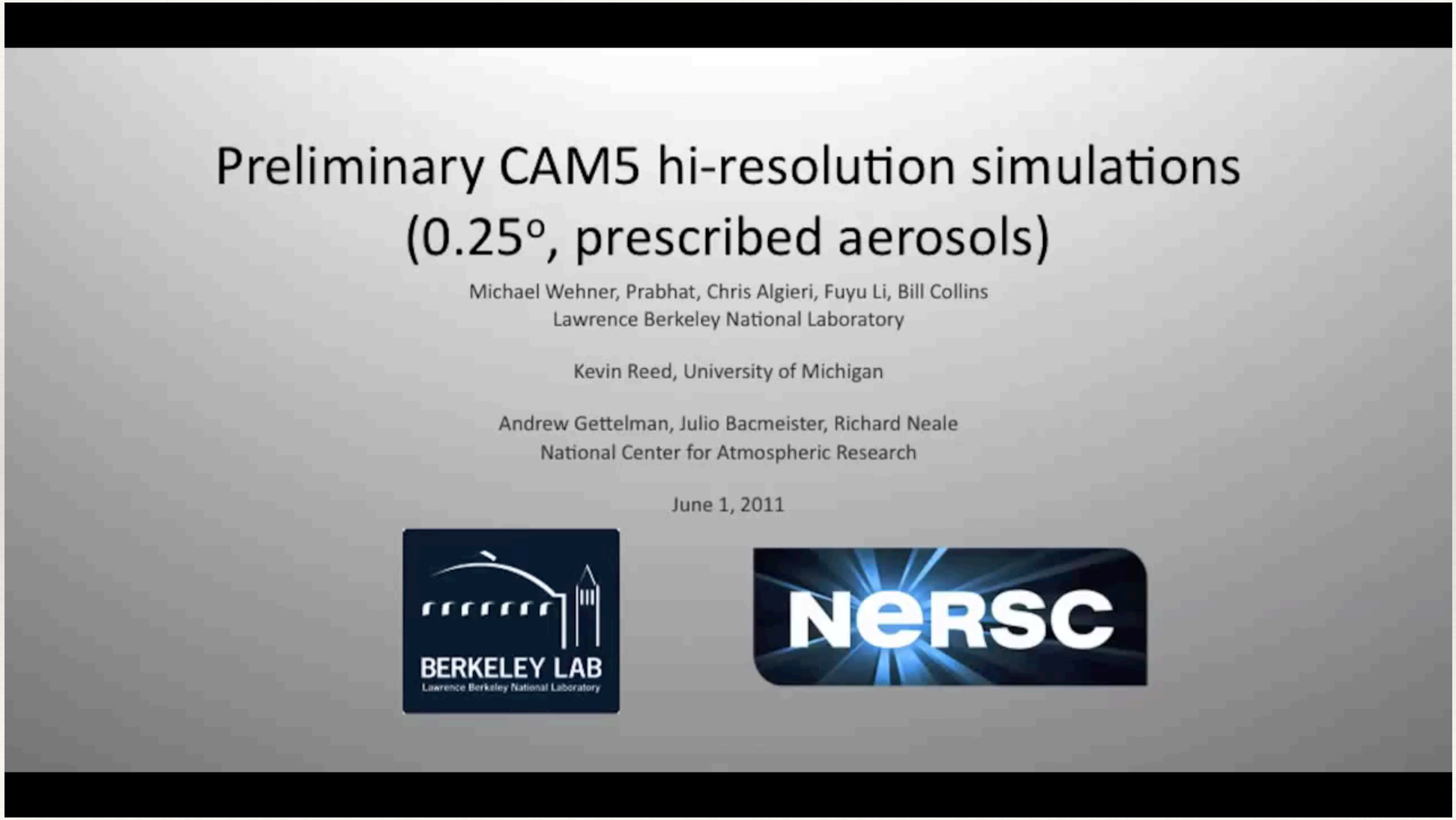
$$\begin{aligned} N_{kij} = {} & \gamma^{mn}\big(4K_{k(i}f_{j)mn} - 4f_{mn(i}K_{j)k} + K_{ij}(2f_{mnk} - 3f_{kmn})\big) \\ & +2\gamma^{mn}\gamma^{pq}\big(K_{mp}(\gamma_{k(i}f_{j)qn} - 2f_{qn(i}\gamma_{j)k}) \\ & +\gamma_{k(i}K_{j)m}(8f_{npq} - 6f_{pqn}) + K_{mn}(4f_{pq(i}\gamma_{j)k} - 5\gamma_{k(i}f_{j)pq})\big) \\ & -K_{ij}\partial_k \ln\hat{\alpha} + 2\gamma^{mn}(K_{m(i}\gamma_{j)k}\partial_n \ln\hat{\alpha} - K_{mn}\gamma_{k(i}\partial_{j)} \ln\hat{\alpha}) \\ & +16\pi\gamma_{k(i}j_{j)}. \end{aligned} \tag{4.15}$$



Caltech/Cornell NR group



Climate modeling and fluid dynamics



GCM!– courtesy of Warren Washington

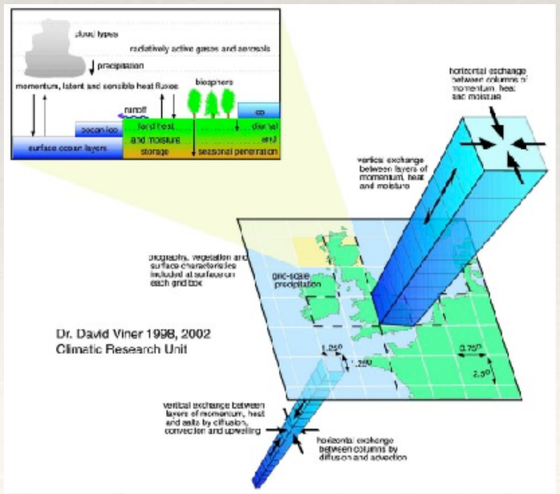
$$\frac{\partial u}{\partial t} + \mathbf{v}\cdot\nabla u + \omega\frac{\partial u}{\partial p} = fv + \frac{uvtan\theta}{a} - \frac{1}{acos\theta}\frac{\partial \Phi}{\partial \lambda} + S_{u,B}$$

$$\frac{\partial v}{\partial t} + \mathbf{v}\cdot\nabla v + \omega\frac{\partial v}{\partial p} = -fu - \frac{u^2 tan\theta}{a} - \frac{1}{a}\frac{\partial \Phi}{\partial \theta} + S_{v,B}$$

$$\frac{\partial \Phi}{\partial lnp} = -R_d T_v$$

$$\nabla\cdot\mathbf{v} + \frac{\partial \omega}{\partial p} = 0$$

$$\frac{\partial T}{\partial t} + \mathbf{v}\cdot\nabla T + \omega\frac{\partial T}{\partial p} = -\frac{\kappa T\omega}{p} - Q_R + Q_C + Q_B$$

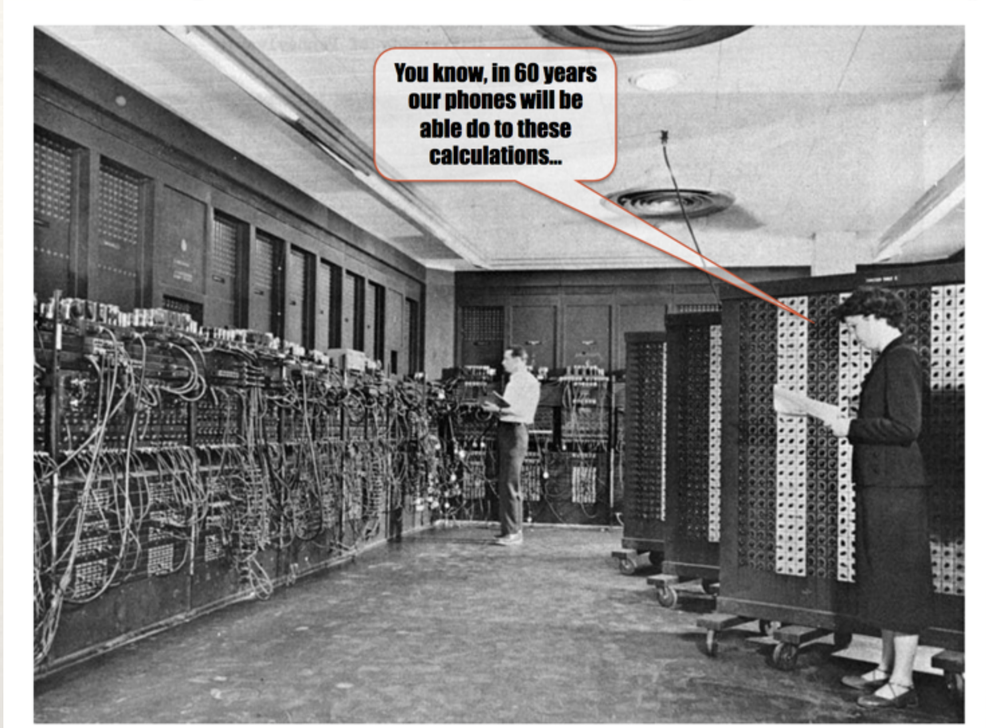


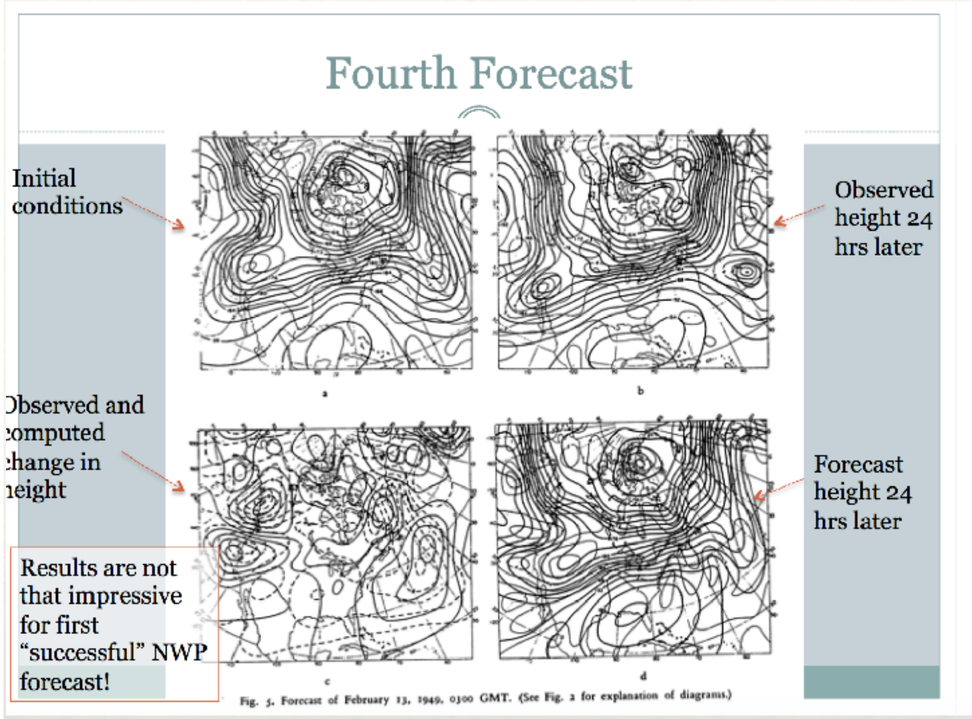
ipcc-data.org

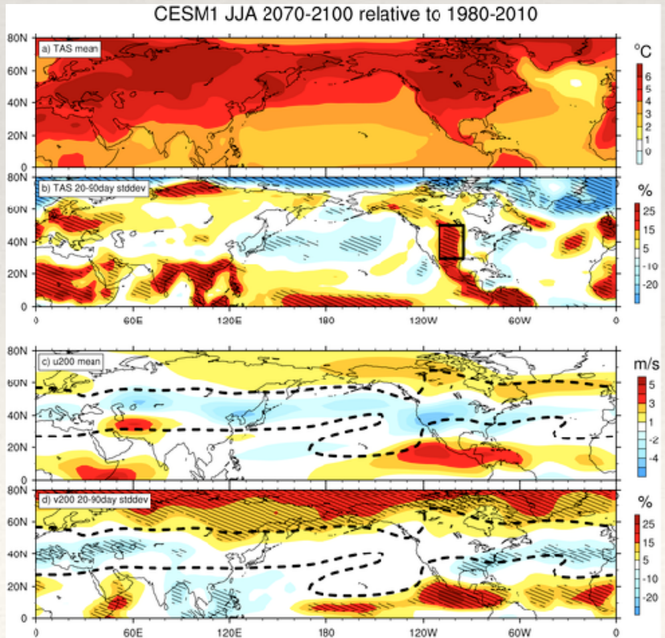
Photo Credit:
Ryan K. Photography and the National Science & Technology Medals Foundation.

https://agupubs.onlinelibrary.wiley.com/doi/10.1002/2015GL067574

https://www.atmos.washington.edu/~dargan/591/591_1.pdf

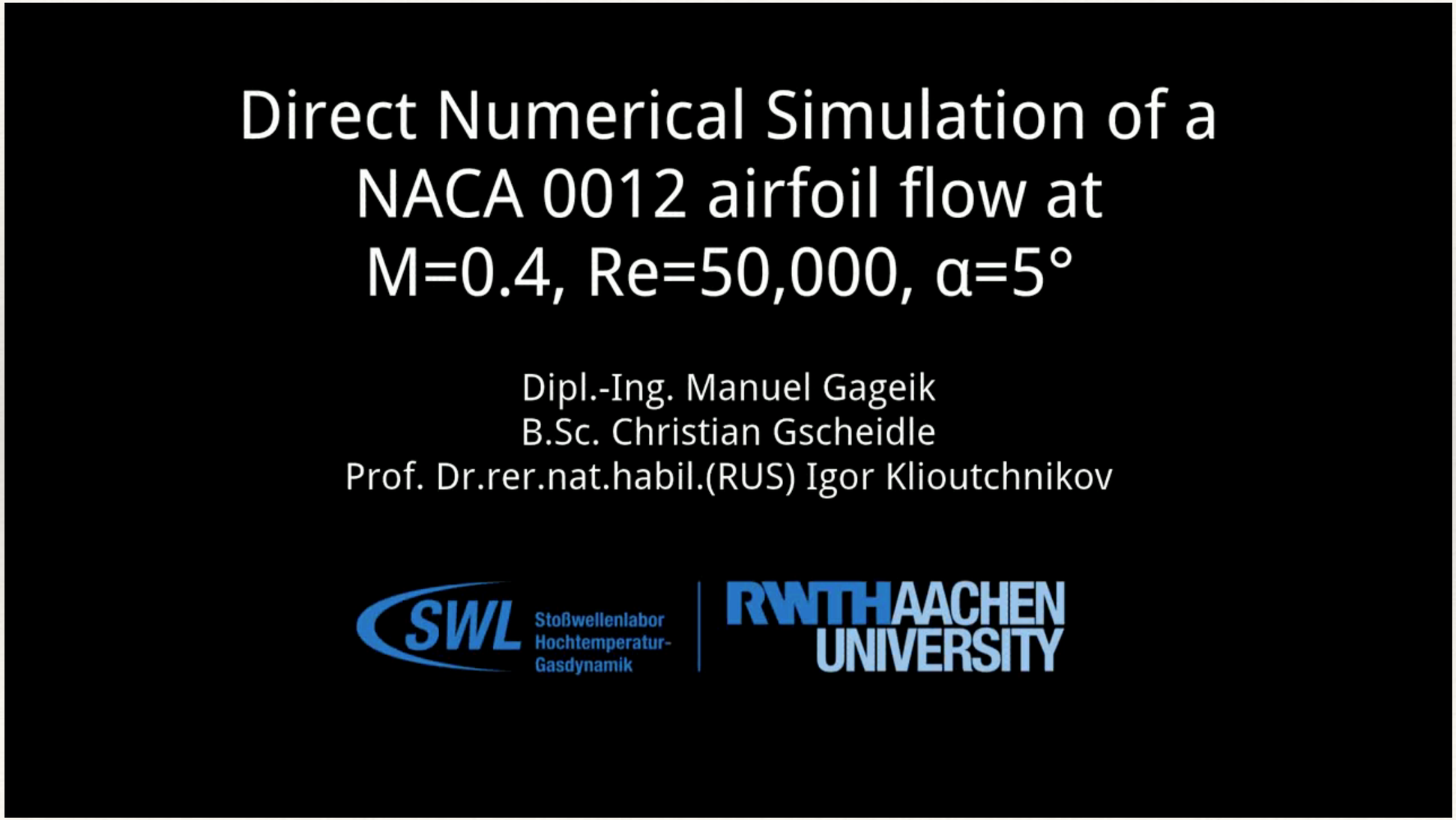






https://therevelator.org/warren-washington-climate/

Airplanes! (from Shock Wave Laboratory RWTH Aachen)



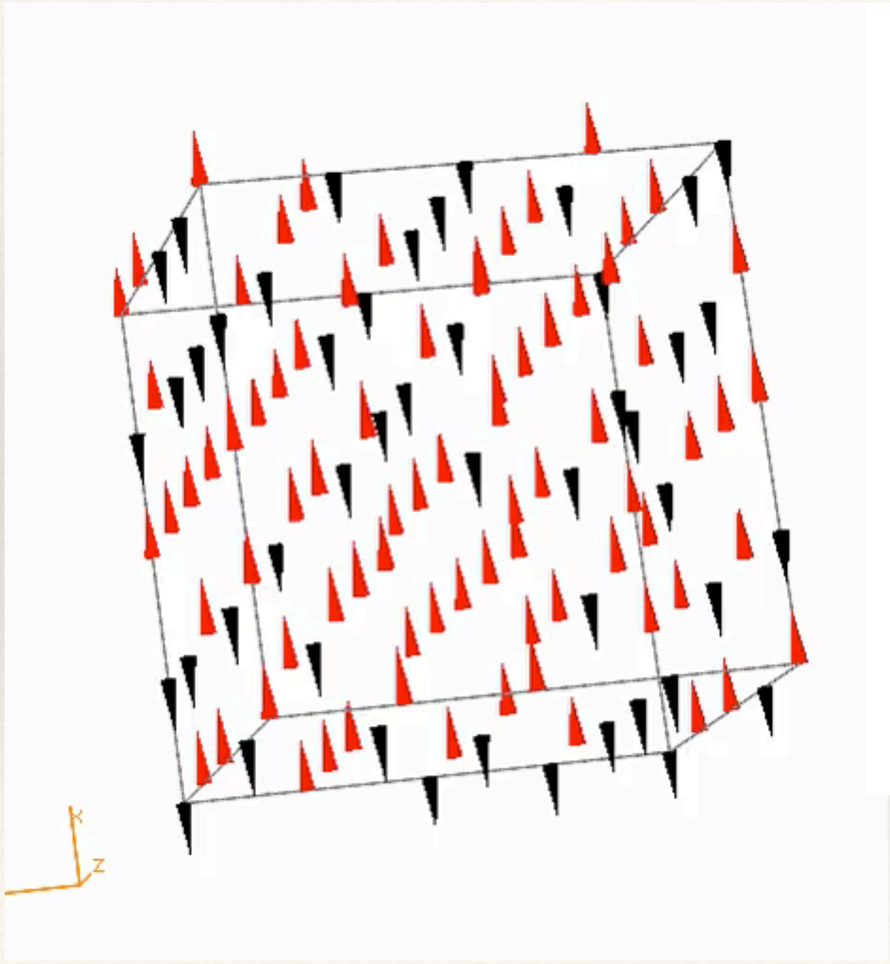
Protein Folding

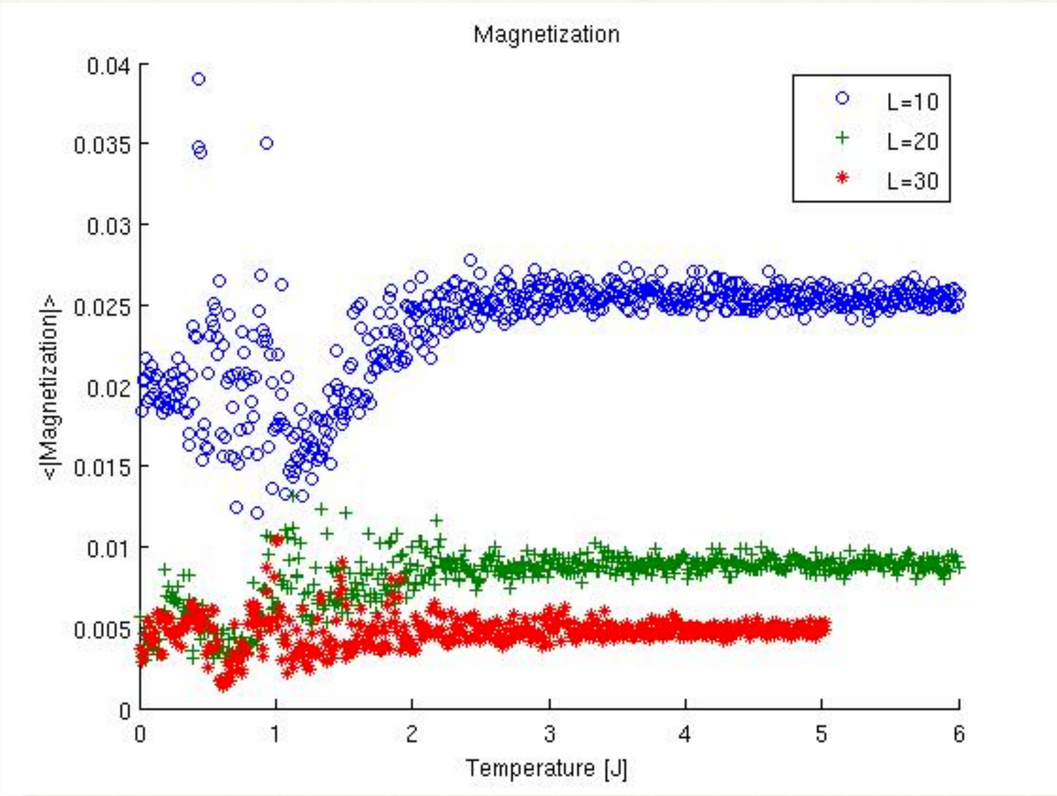
Eugene Shakhnovich, MIT + Harvard

Even 100 amino-acid sequences are hard to simulate –
computational cleverness is needed!

Magnetization of material

http://phelafel.technion.ac.il/~kostyas/results.html



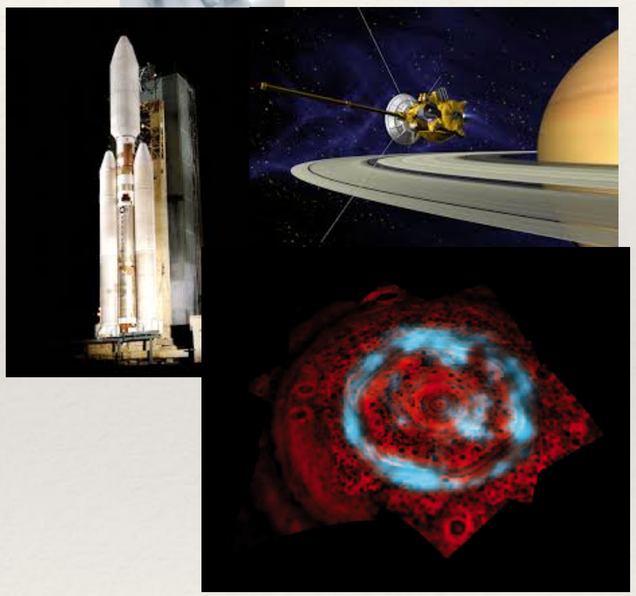


Some notable computational scientists

On the front cover: the large photograph is of Melba Roy Mouton. She calculated artificial satellite orbits as a Head Computer Programmer for NASA.

Cecilia Payne-Kaposhkin

Annie Easley



Katherine Johnson

Folks to talk to….scientific computing

Concentration Coordinators and Departmental Representatives

Jane Chandlee
Assistant Professor of Linguistics
Chase 103C
(610) 795-3371
jchandlee@haverford.edu
Profile
Homepage

Sara Mathieson
Assistant Professor
of Computer Science
Scientific Computing
Concentration Coordinator

Andrea Lommen
Professor of Physics and Astronomy
KINSC Link 108
alommen@haverford.edu
Profile

Robert Manning
The William H. and Johanna A. Harris Professor of Computational Science; Professor of Mathematics and Statistics; Associate Provost for Faculty Development and Support
KINSC H207C
(610) 896-1210
rmanning@haverford.edu

Casey Londergan
Professor
and Chair of Chemistry

Dan Grin
Assistant Prof. of Physics
+ Astronomy

Alvin Grissom II
Assistant Professor
of Computer Science

J. Dougherty
Prof. of Comp. Science

Giri Parameswaran
Associate Prof. of
Economics

Suzanne Amador Kane
Professor of Physics

Sorelle Friedler
Associate Professor of Computer Science

Professor of Physics and Astronomy;

Eric Miller
Assistant Professor of Biology

Clyde Daly
Assistant Professor
of Chemistry

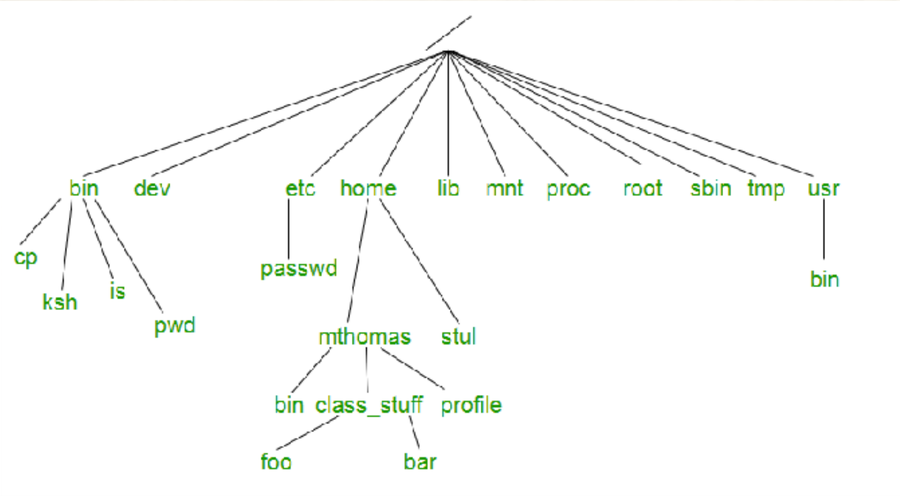
Carola Binder
Associate Professor of
Economics

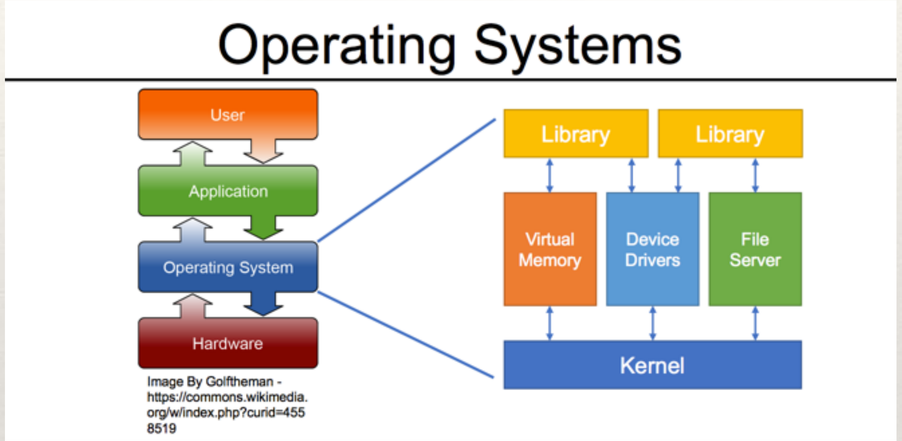
Ted Brzinski
Assistant Professor of Physics and Astronomy
KINSC L109
(610) 896-1196

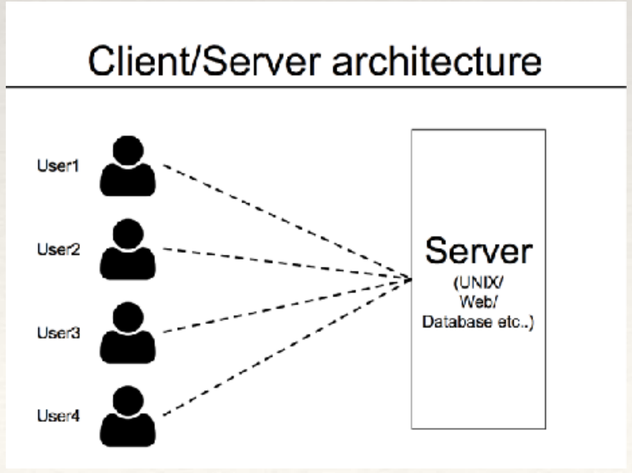
Rebecca Everett
Assistant Professor of Mathematics and Statistics

UNIX

file system







UNIX and its derivatives

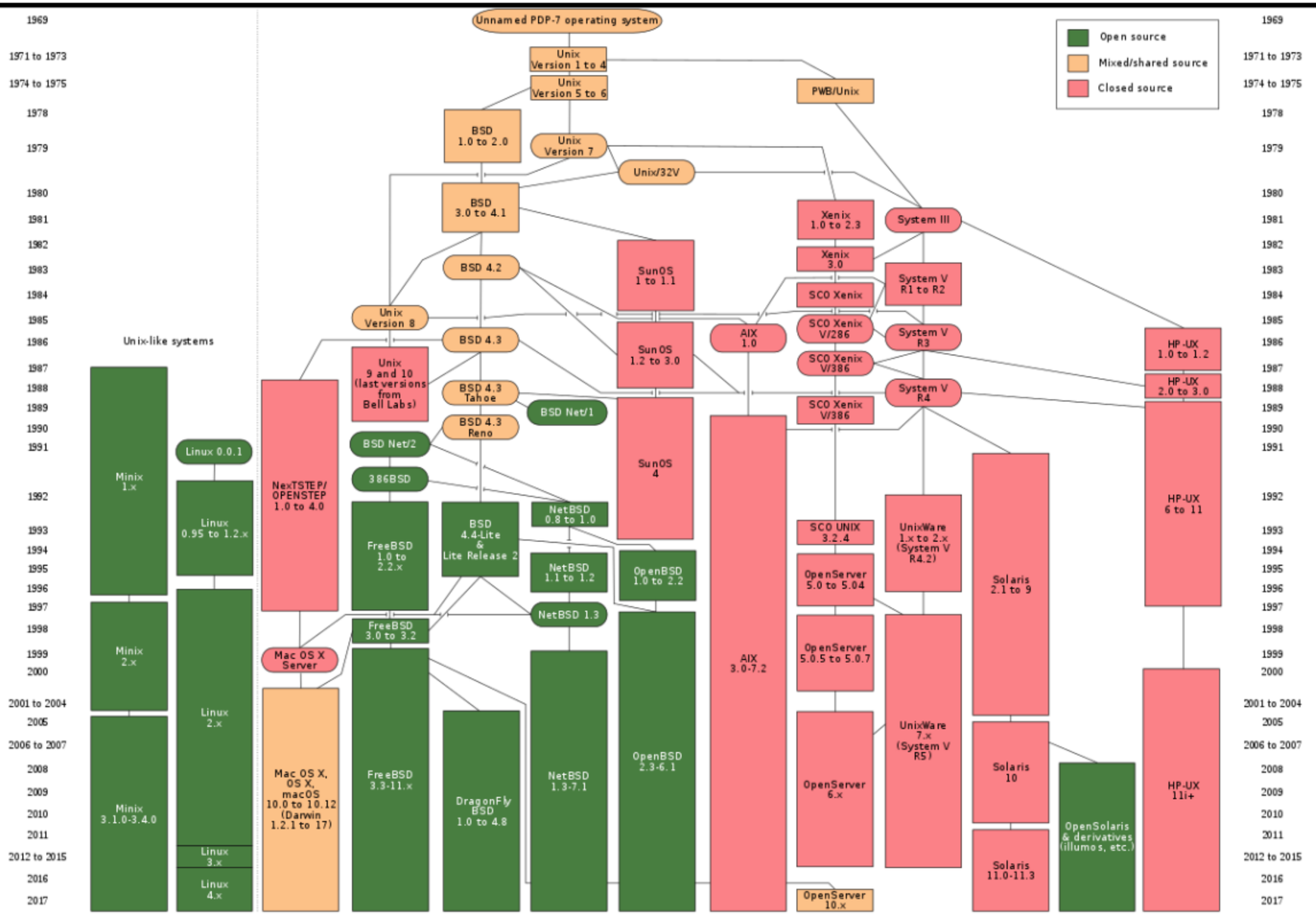


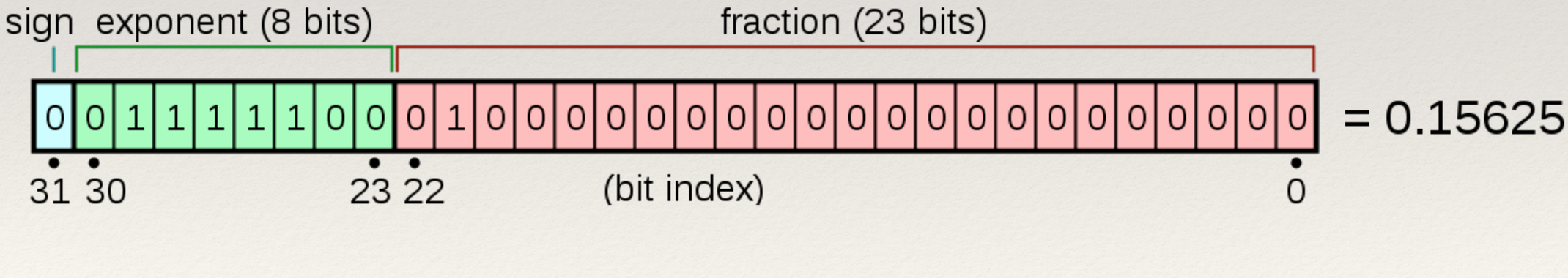
Image by Eraserhead1, Infinity0, Sav_vas - Levenez Unix History Diagram, Information on the history of IBM's AIX on ibm.com, https://commons.wikimedia.org/w/index.php?curid=1801948

Binary representation of numbers

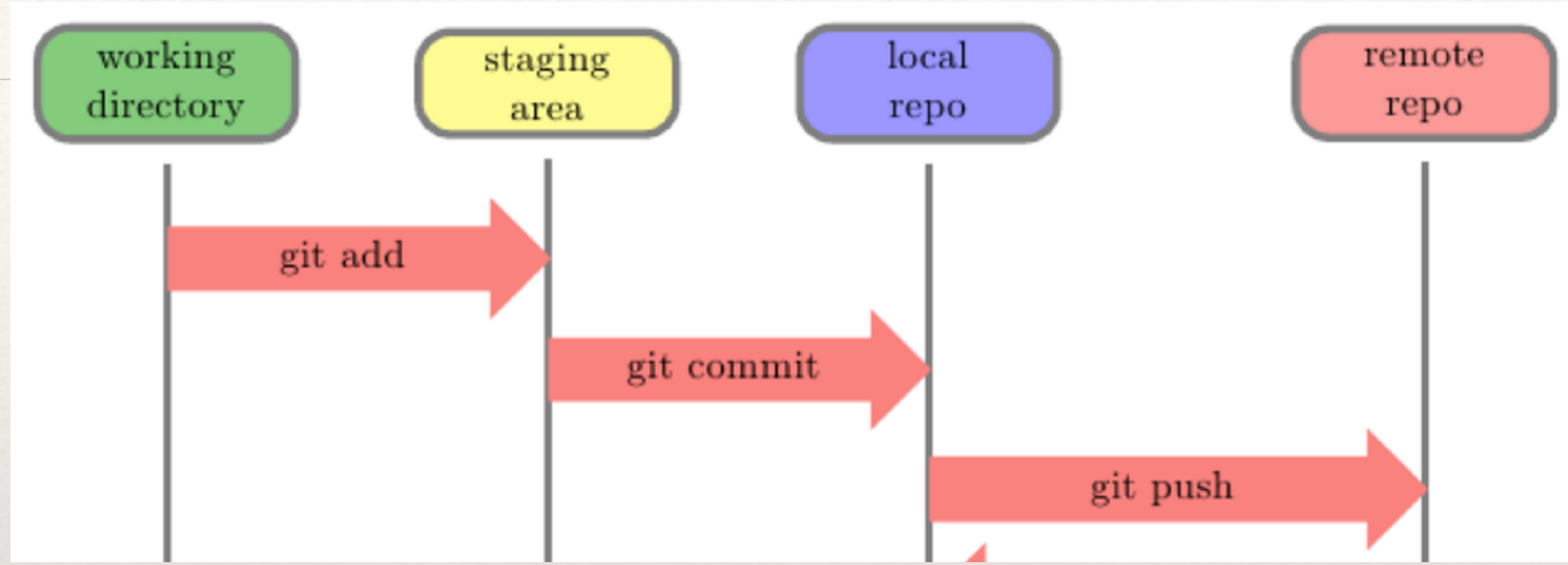
0	1	0	0	0	1	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
x128	x64	x32	x16	x8	x4	x2	x1

64 + 4 + 2

70



git

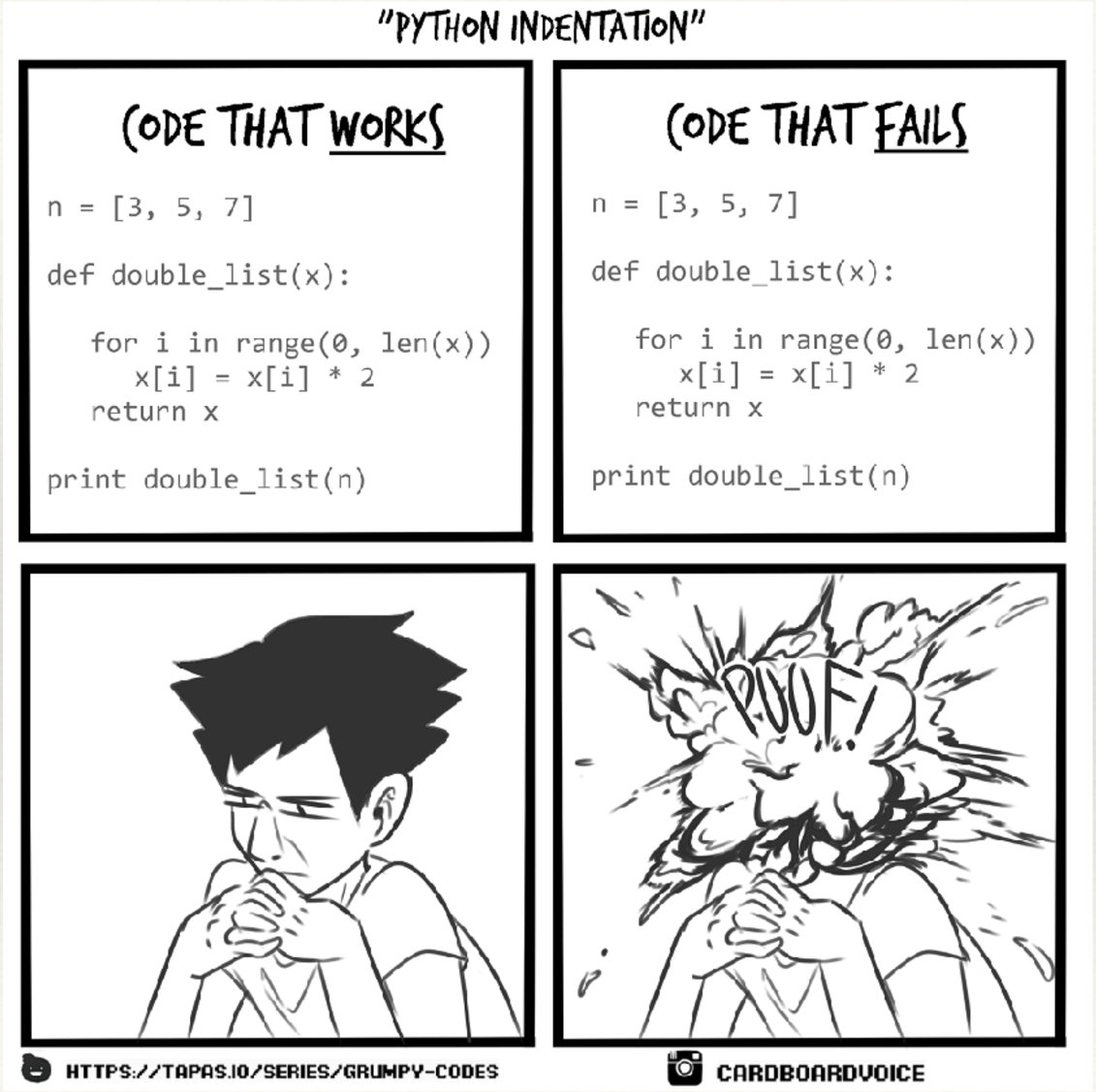


https://github.com/dgrin1/chesick_2022_coding

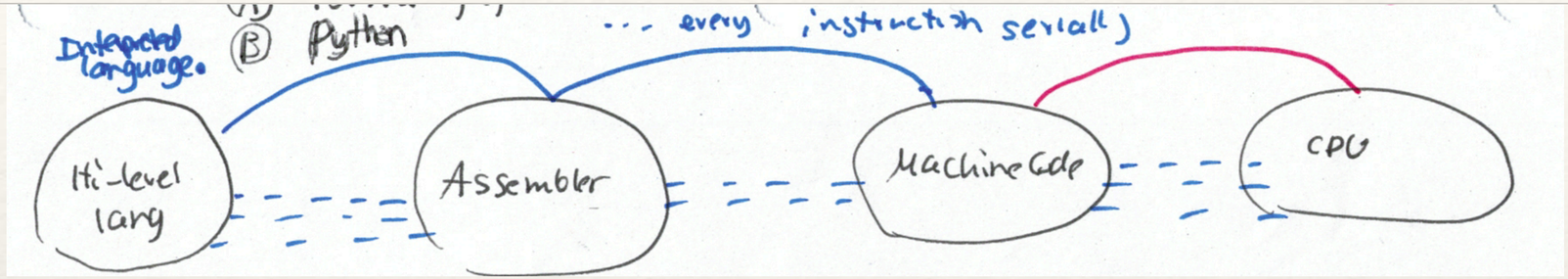
Python

- Many other programming languages good for scientific computing (e.g. Fortran, seriously, C, C++)
- Python lets us cut to the core of interesting scientific computing problems with minimum of reinventing the wheel
  - Access to useful, fast, well-tested scientific computing libraries (e.g. matplotlib, numpy, spicy)
  - No need to deal with innards (memory / variable definitions) of coding

Indentation



interpreted vs compiled languages

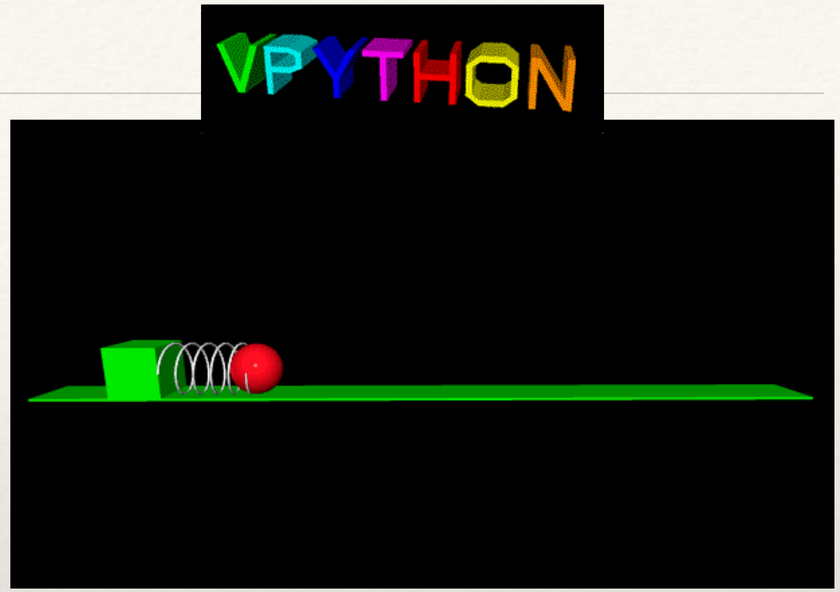


vPython

trinket ▶ Run ? Help

main.py

```
1  GlowScript 2.9 VPython
2
3  g=9.8
4  v=5
5  y=0
6  t=0
7  dt=0.1
8
9  while y>=0:
10   v=v-g*dt
11   y=y+v*dt
12   print("t = ",t," y = ",y)
13   t=t+dt
14 print("Hang time = ",t," s")
15
16
```



- Computer programming to simulate physics problems
- VPython — a form of Python, no previous experience assumed

GlowScript

Show how to access my programs and save their own

Hello world!

```
print("Hello, world!")
```

Function String

Function being called to display string

Try it! - now change it!

vpython – visual data types

```
box()
```

```
box(pos=vector(1,0,0),size=vector(.5,.3,.2),
    color=color.green)
```

mybox=box(length=0.5, height=0.5, width=0.5, color=color.yellow)

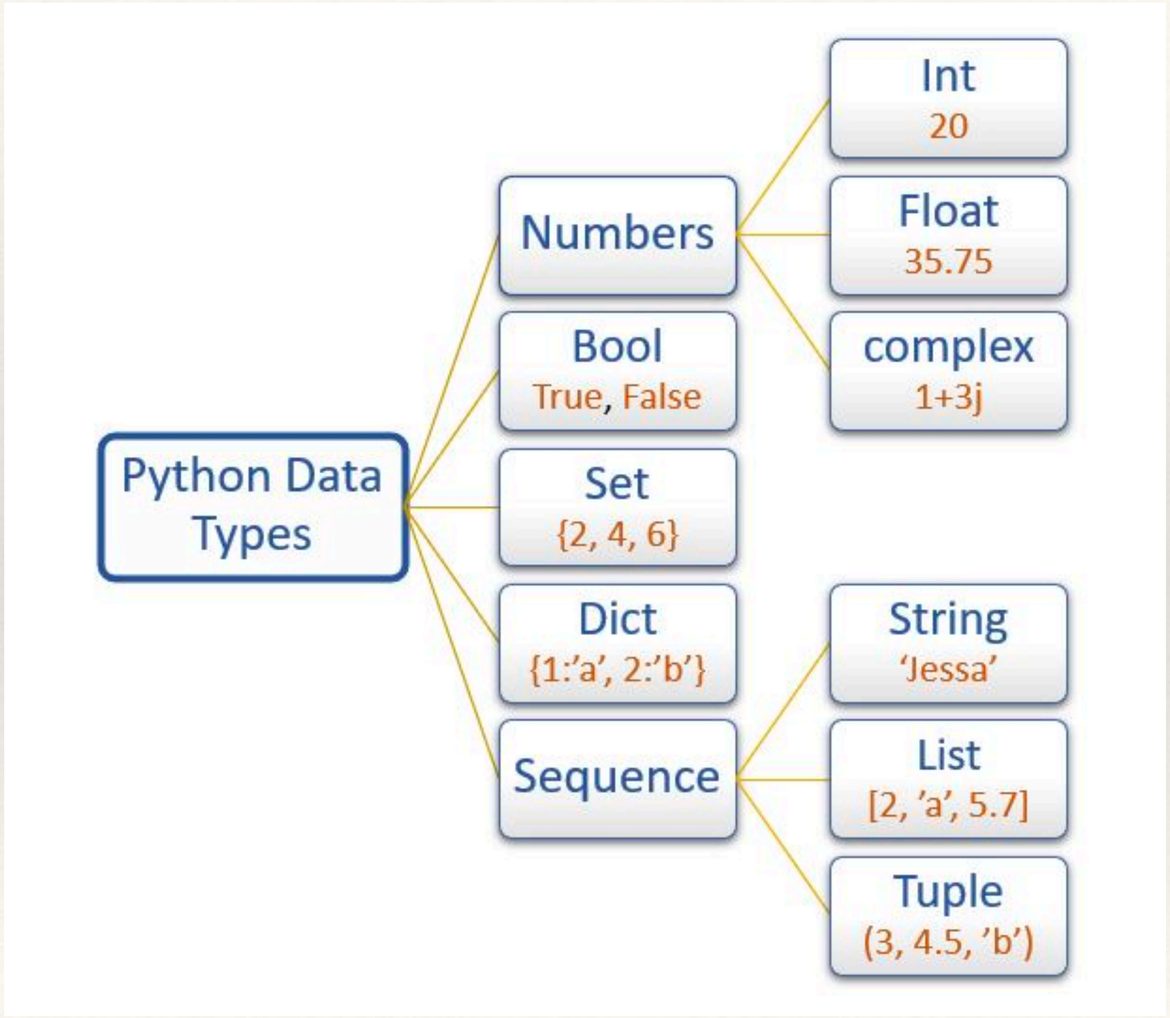
Computer takes care of world rendering

Try it! - now change it!

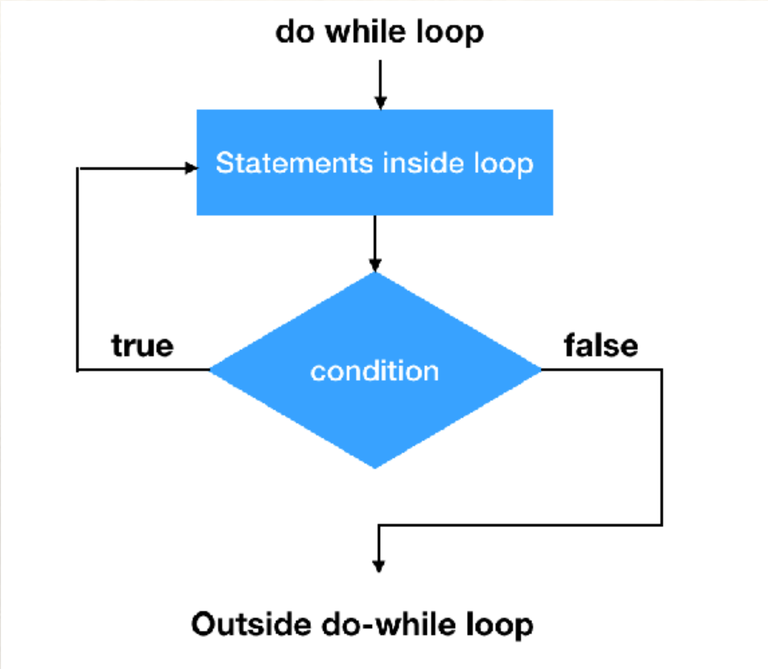
Calculating with python and using variables

```
R = 1

mysphere = sphere(radius = R, color = color.magenta)

volume = (4/3)*pi*R**3

print("The volume of the sphere is:", volume)
```

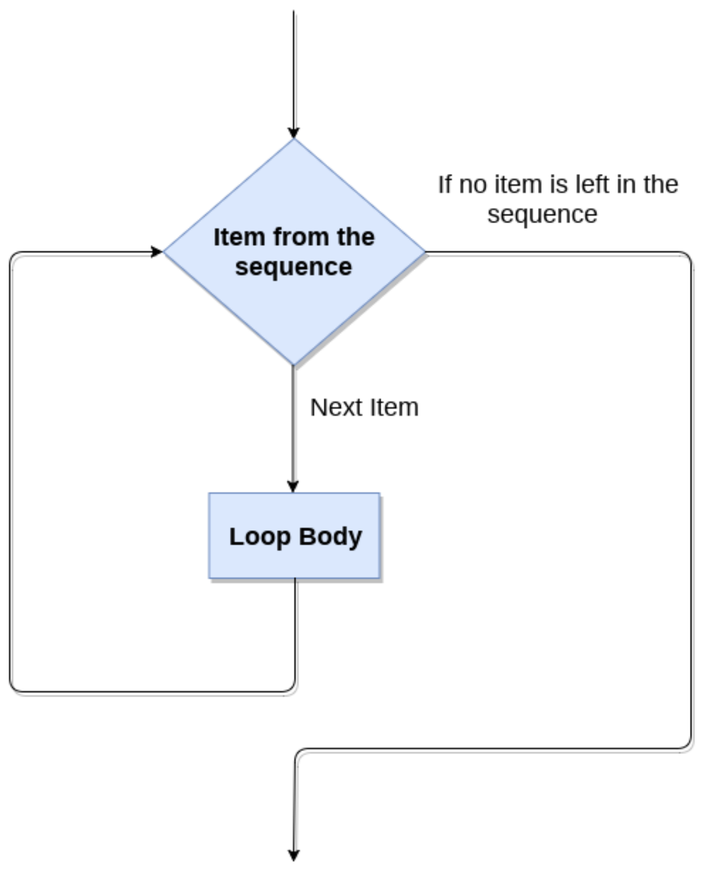
More Python data types



program flow



for loop



Loops, example

https://matter-interactions.trinket.io/00_welcome_to_vpython#/welcome-to-vpython/loops

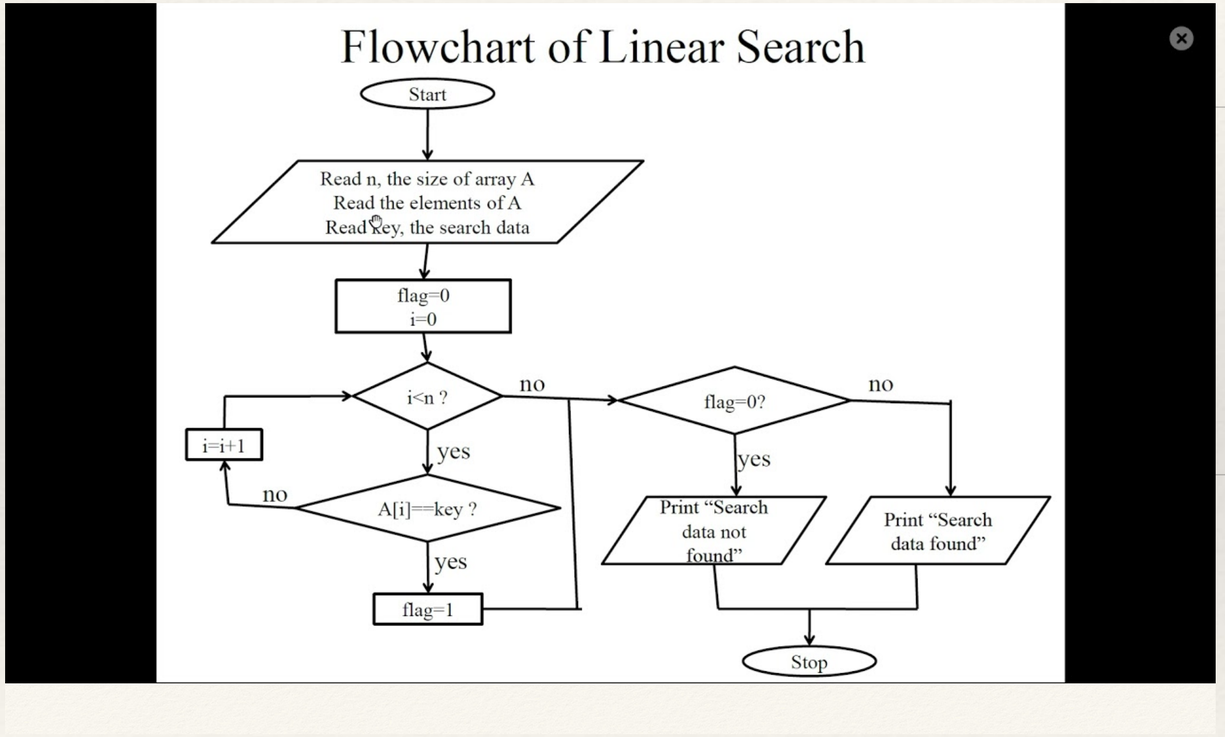
```
t = 0
print("t =", t)

t = t + 1
print("t =", t)

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

```
t = 0
while t < 11:
  print("t=", t)
  t = t + 2

print("The loop is finished")
```

dg loop examples

Debugging

Plan your code



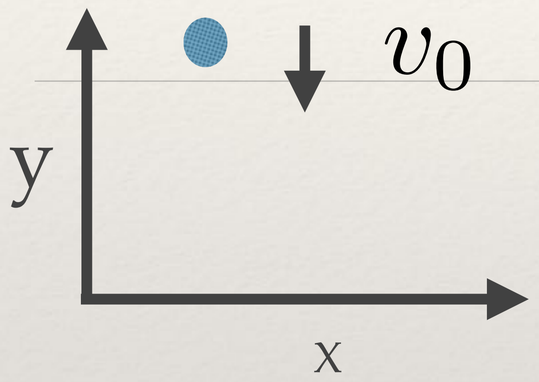
Work in pairs and plan out code on white board – check in with Deep or me before coding

Comment your code

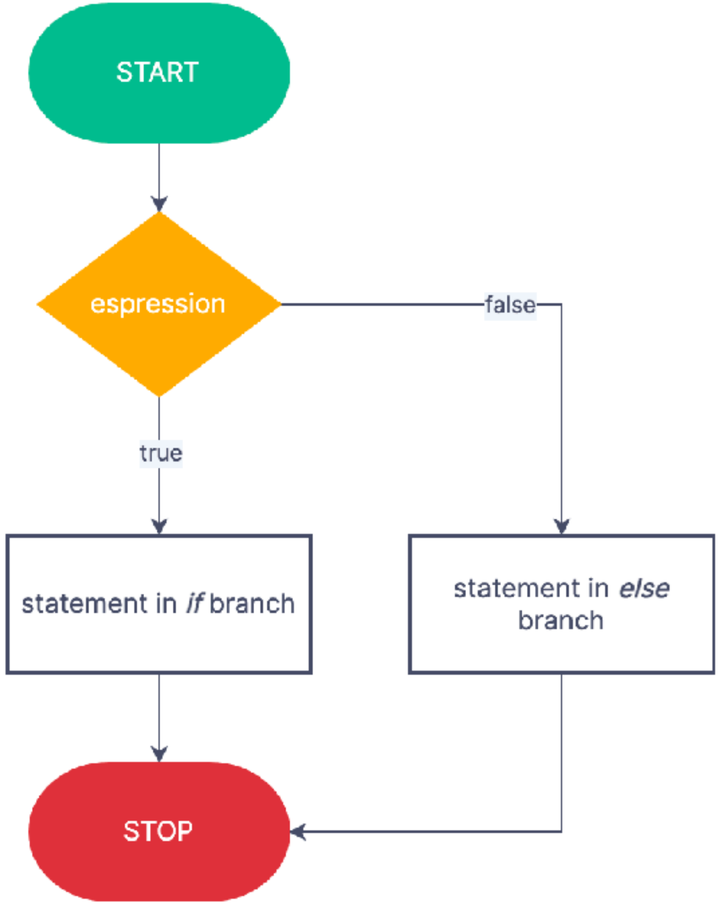
- Helps you remember what you were doing years later :)
- Helps colleagues expand and edit your code
- Helps debug (comment out and add back in code that is not working)

Moving ball example

$$y = y_0 + v_0 t$$



if/then program flow control

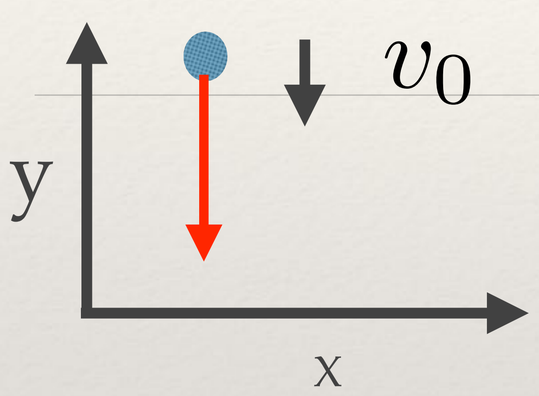


Use to impose table boundary

Falling ball with gravity— your turn! (Work in pairs)

$$y = y_0 + v_0 t - gt^2/2$$

$$v = v_0 - gt$$

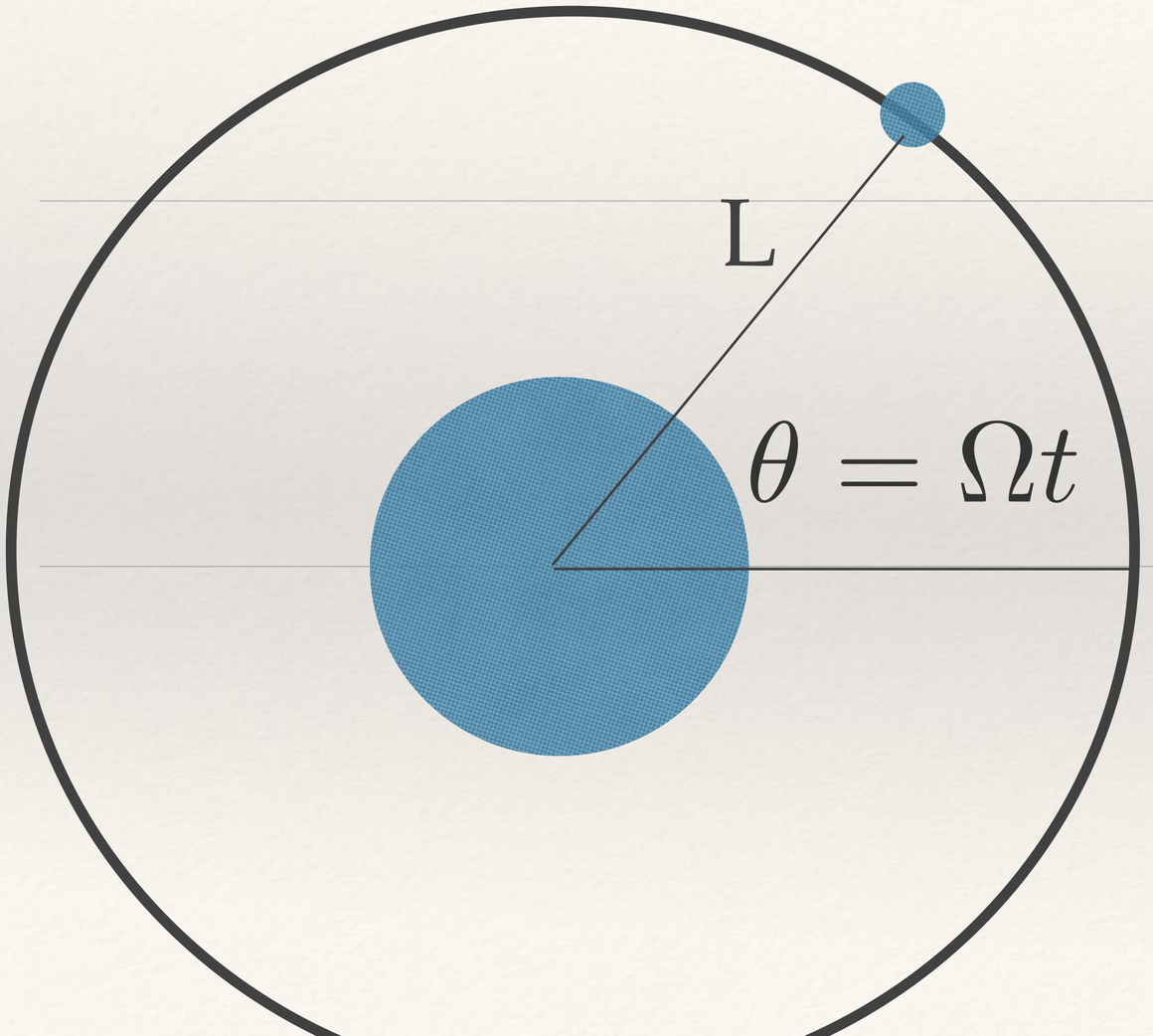


$$\vec{F} = m\vec{g}$$

Challenge — add a plot of v(t)

Example – orbits

$$x = L\cos(\Omega \times t)$$
$$y = L\sin(\Omega \times t)$$



Your turn – add the moon

Your turn — extrasolar planets

https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=PS