Computer Programming and Physics

June 18, 2019

Why Computational Physics?

- Computational skills are critical for a career in both theoretical and experimental physics.

- Critical for other work:
  - Engineering
  - Finance and Social Systems
  - Teaching, Games
  - …

Computation in contemporary physics

- Numerical calculations: Solutions to well defined mathematical problems to produce numerical solutions.
- Computer simulation: Testing models of nature
  Examples: weather forecast, …
- Data collection and analysis: In experimental research
  Example: LabView.
- Symbolic manipulation: Examples: Maple, Mathematica, …
- Visualization and animation: The human eye + the visual processing power of the brain = very sophisticated tool.

Why Computational Physics?

"Being able to transform a theory into an algorithm requires significant theoretical insight, detailed physical and mathematical understanding, and a mastery of the art of programming."

From: *A Survey of Computational Physics*, R. Landau, M.J. Paez,C Bordeianu

Computation in contemporary physics

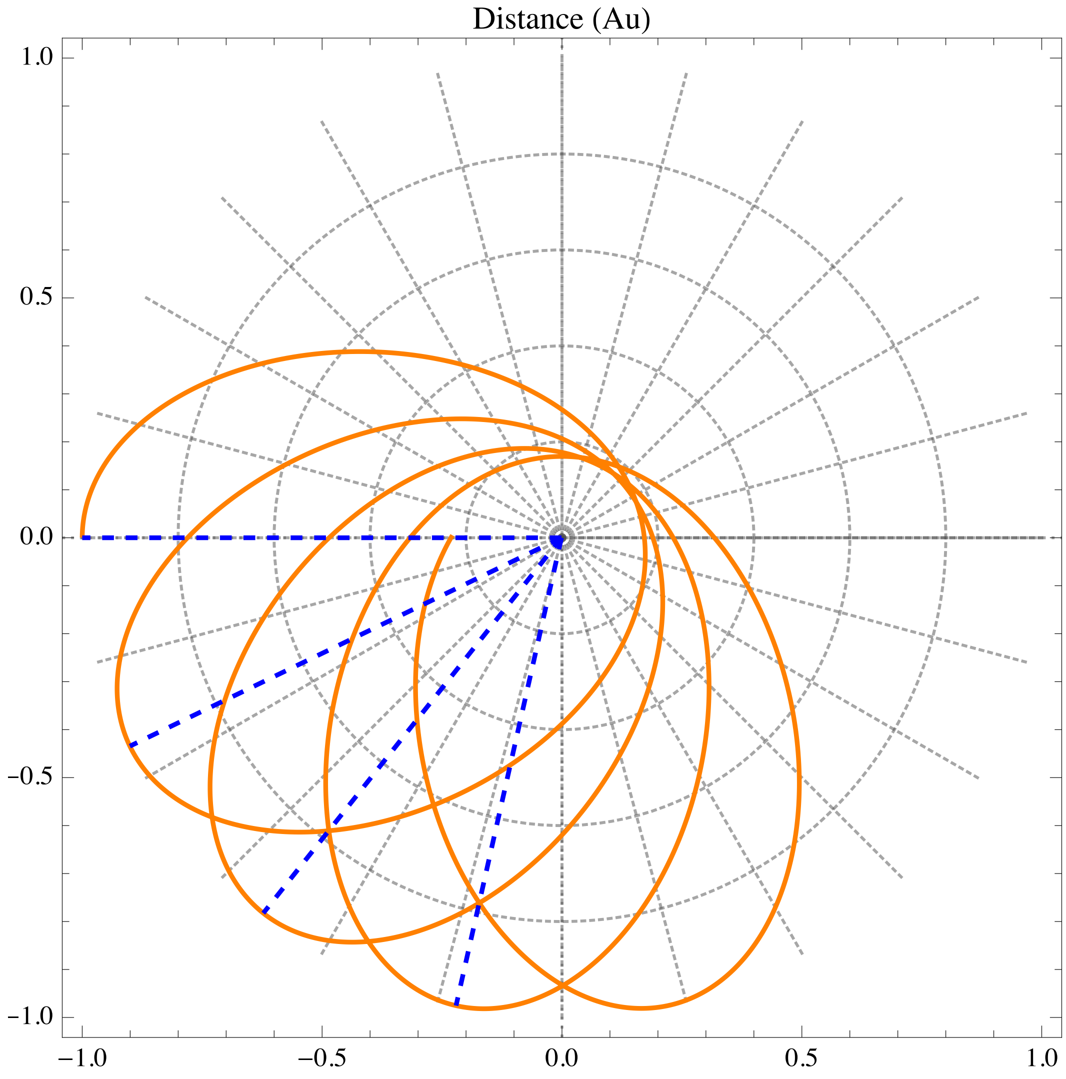
- Numerical calculations: Solutions to well defined mathematical problems to produce numerical solutions.

Differential Equations:

$$\frac{\mathrm{d}^2\vec{\mathbf{x}}}{\mathrm{d}t^2} = \frac{\vec{\mathbf{F}}}{m}$$

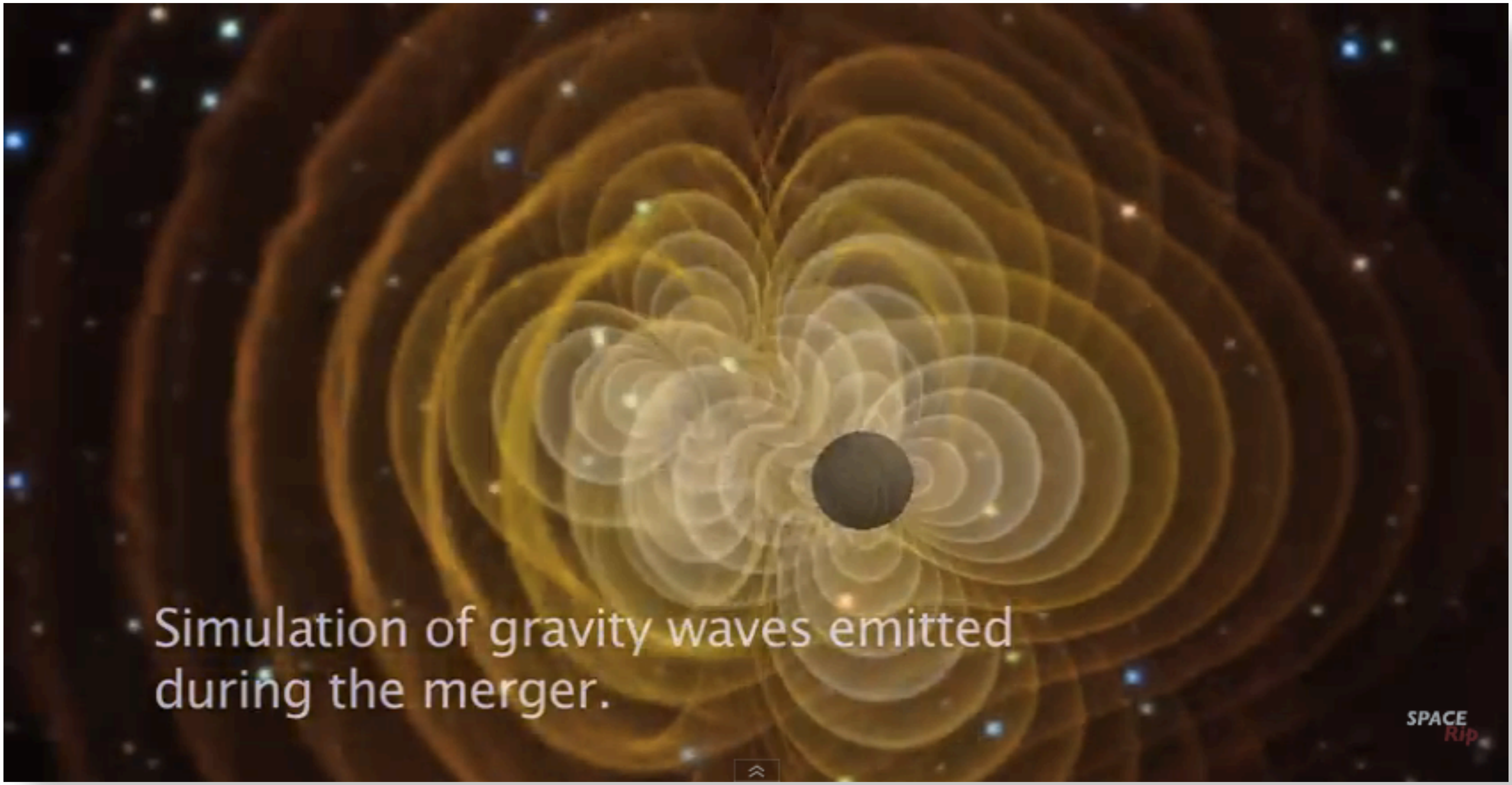
Familiar Example

Orbital Mechanics



Gravitation

$$R_{\mu\nu} - \frac{1}{2} R\, g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$



https://youtu.be/GVds0q0y5RM

Gravitation

http://www.astro.cardiff.ac.uk/research/gravity/tutorial/?page=4blackholecollisions

Computation in contemporary physics

- Numerical calculations: Solutions to well defined mathematical problems to produce numerical solutions.

- Computer simulation: Testing models of nature
  Examples: weather forecast, complex dynamical systems …

Classification of Models

Deterministic or Stochastic models

⇨ Deterministic models: Results of deterministic models depend on initial conditions.

⇨ Stochastic models: an element of chance exists.

Dynamic or Static models

⇨ A dynamic models changes in time.

⇨ A static model does not consider time

Computer Simulation (few examples)

- ✓ Molecular Dynamic Simulation
- ✓ Weather forecast
- ✓ Design of complex systems (aircrafts, …)
- ✓ Financial markets
- ✓ Traffic
- ✓ *Games!*
- ✓ …

Climate prediction	Materials science	Structural biology
Superconductivity	Semiconductor design	Drug design
Human genome	Quantum chromodynamics	Turbulence
Speech and vision	Relativistic astrophysics	Vehicle dynamics
Nuclear fusion	Combustion systems	Oil and gas recovery
Ocean science	Vehicle signature	Undersea surveillance

From: *A Survey of Computational Physics*, R. Landau, M.J. Paez,C Bordeianu

more …

- Many natural phenomena are *nonlinear*, and a small change in a variable might produce a large effect.
- Few nonlinear problems can be solved analytically.
- Interest in systems with many variables or many degrees of freedom

Millennium Simulation - the largest N-body simulation carried out thus far (more than 10^{10} particles).
A 3-dimensional visualization of the Millennium Simulation shows a journey through the simulated universe
http://www.mpa-garching.mpg.de/galform/millennium/

Visualization

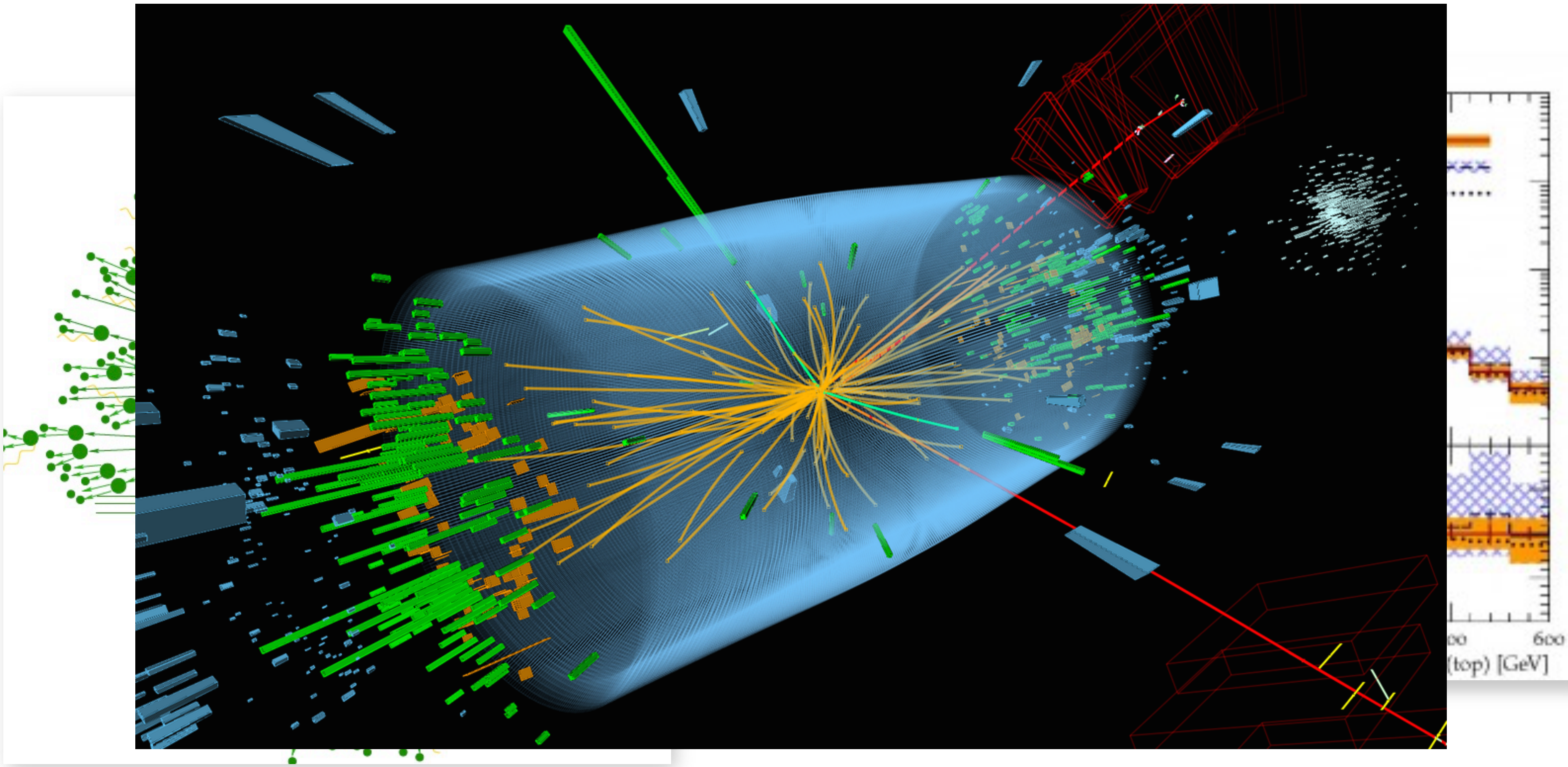
http://iopscience.iop.org/article/10.1088/1367-2630/10/12/125001/meta

"The effectiveness of visualization arises by exploiting the unmatched capability of the human eye and visual cortex to process the large information content of images. In a brief glance, we recognize patterns or identify subtle features even in noisy data, something that is difficult or impossible to achieve with more traditional forms of data analysis...

...Importantly, visualizations guide the intuition of researchers and help to comprehend physical phenomena that lie far outside of direct experience. In fact, visualizations literally allow us to see what would otherwise remain completely invisible. For example, artificial imagery created to visualize the distribution of dark matter in the Universe has been instrumental to develop the notion of a cosmic web"

From: "Focus on Visualization in Physics,"
New Journal of Physics
By Barry C Sanders, Tim Senden, and Volker Springel

Nuclear and Particle Physics



From: *Sherpa and Open Science Grid: Predicting the emergence of jets, 2014, International Science Grid This Week, Amber Harmon*

https://sciencenode.org/feature/sherpa-and-open-science-grid-predicting-emergence-jets.php

Computation in contemporary physics

- Numerical calculations: Solutions to well defined mathematical problems to produce numerical solutions.

- Computer simulation: Testing models of nature
  Examples: weather forecast, …

- Data collection and analysis: In experimental research
  Example: LabView.

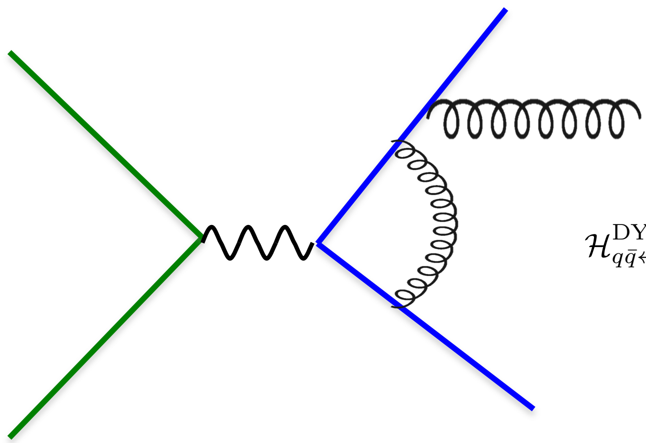
- Visualization and animation: The human eye + the visual processing power of the brain = very sophisticated tool.

http://aapt.scitation.org/doi/full/1

http://gamelab.mit.edu/games/a-slower-speed-of-light/

Computation in contemporary physics

- Numerical calculations: Solutions to well defined mathematical problems to produce numerical solutions.
- Computer simulation: Testing models of nature
  Examples: weather forecast, …
- Data collection and analysis: In experimental research
  Example: LabView.
- Visualization and animation: The human eye + the visual processing power of the brain = very sophisticated tool.
- Symbolic manipulation: Examples: Maple, Mathematica, …



Feynman Diagrams

$$
\begin{aligned}
\mathcal{H}^{\mathrm{DY}(2)}_{q\bar{q}\leftarrow qg}(x) = \frac{C_A}{4\pi^2}\Bigg\{ & -\frac{1}{12x}(1-x)\left(11x^2-x+2\right)\mathrm{Li}_2(1-x) \\
& + \left(2x^2-2x+1\right)\left(\frac{\mathrm{Li}_3(1-x)}{8} - \frac{1}{8}\mathrm{Li}_2(1-x)\ln(1-x) + \frac{1}{48}\ln^3(1-x)\right) \\
& + \left(2x^2+2x+1\right)\Bigg(\frac{3\mathrm{Li}_3(-x)}{8} + \frac{\mathrm{Li}_3\left(\frac{1}{1+x}\right)}{4} - \frac{\mathrm{Li}_2(-x)\ln(x)}{8} - \frac{1}{24}\ln^3(1+x) \\
& + \frac{1}{16}\ln^2(x)\ln(1+x) + \frac{1}{48}\pi^2\ln(1+x)\Bigg) + \frac{1}{4}x(1+x)\mathrm{Li}_2(-x) + x\mathrm{Li}_3(x) \\
& - \frac{1}{2}x\mathrm{Li}_2(1-x)\ln(x) - x\mathrm{Li}_2(x)\ln(x) - \frac{3}{8}\left(2x^2+1\right)\zeta_3 - \frac{149x^2}{216} \\
& - \frac{1}{96}\left(44x^2-12x+3\right)\ln^2(x) + \frac{1}{72}\left(68x^2+6\pi^2x-30x+21\right)\ln(x) + \frac{\pi^2x}{24} + \frac{43x}{48} \\
& + \frac{43}{108x} + \frac{1}{48}(2x+1)\ln^3(x) - \frac{1}{2}x\ln(1-x)\ln^2(x) - \frac{1}{8}(1-x)x\ln^2(1-x) \\
& + \frac{1}{4}x(1+x)\ln(1+x)\ln(x) + \frac{1}{16}(3-4x)x\ln(1-x) - \frac{35}{48}\Bigg\} \\
& + C_F\Bigg\{\left(2x^2-2x+1\right)\Bigg(\zeta_3 - \frac{\mathrm{Li}_3(1-x)}{8} - \frac{\mathrm{Li}_3(x)}{8} + \frac{1}{8}\mathrm{Li}_2(1-x)\ln(1-x) \\
& + \frac{\mathrm{Li}_2(x)\ln(x)}{8} - \frac{1}{48}\ln^3(1-x) + \frac{1}{16}\ln(x)\ln^2(1-x) + \frac{1}{16}\ln^2(x)\ln(1-x)\Bigg) \\
& - \frac{3x^2}{8} - \frac{1}{96}\left(4x^2-2x+1\right)\ln^3(x) + \frac{1}{64}\left(-8x^2+12x+1\right)\ln^2(x) \\
& + \frac{1}{32}\left(-8x^2+23x+8\right)\ln(x) + \frac{5}{24}\pi^2(1-x)x + \frac{11x}{32} + \frac{1}{8}(1-x)x\ln^2(1-x) \\
& - \frac{1}{4}(1-x)x\ln(1-x)\ln(x) - \frac{1}{16}(3-4x)x\ln(1-x) - \frac{9}{32}\Bigg\},
\end{aligned}
$$

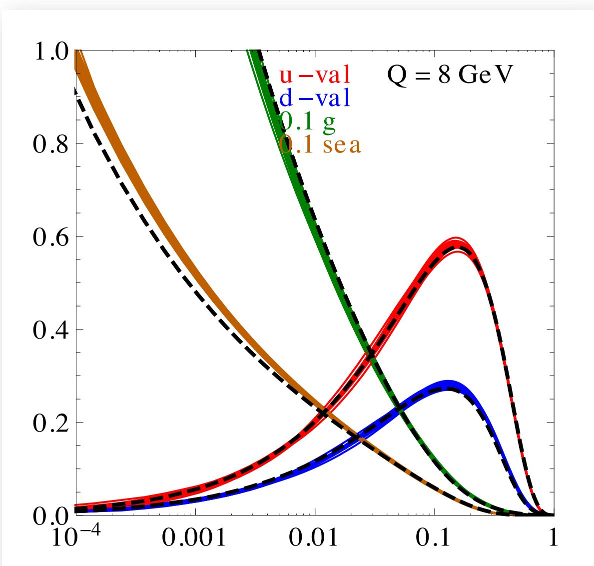
$$
\mathrm{Li}_2(z) = -\int_0^z \frac{\mathrm{d}\,t}{t}\ln(1-t)\,,
$$

$$
\mathrm{Li}_3(z) = \int_0^1 \frac{\mathrm{d}\,t}{t}\ln(t)\ln(1-zt)\,.
$$

Example: momentum of quarks?

x

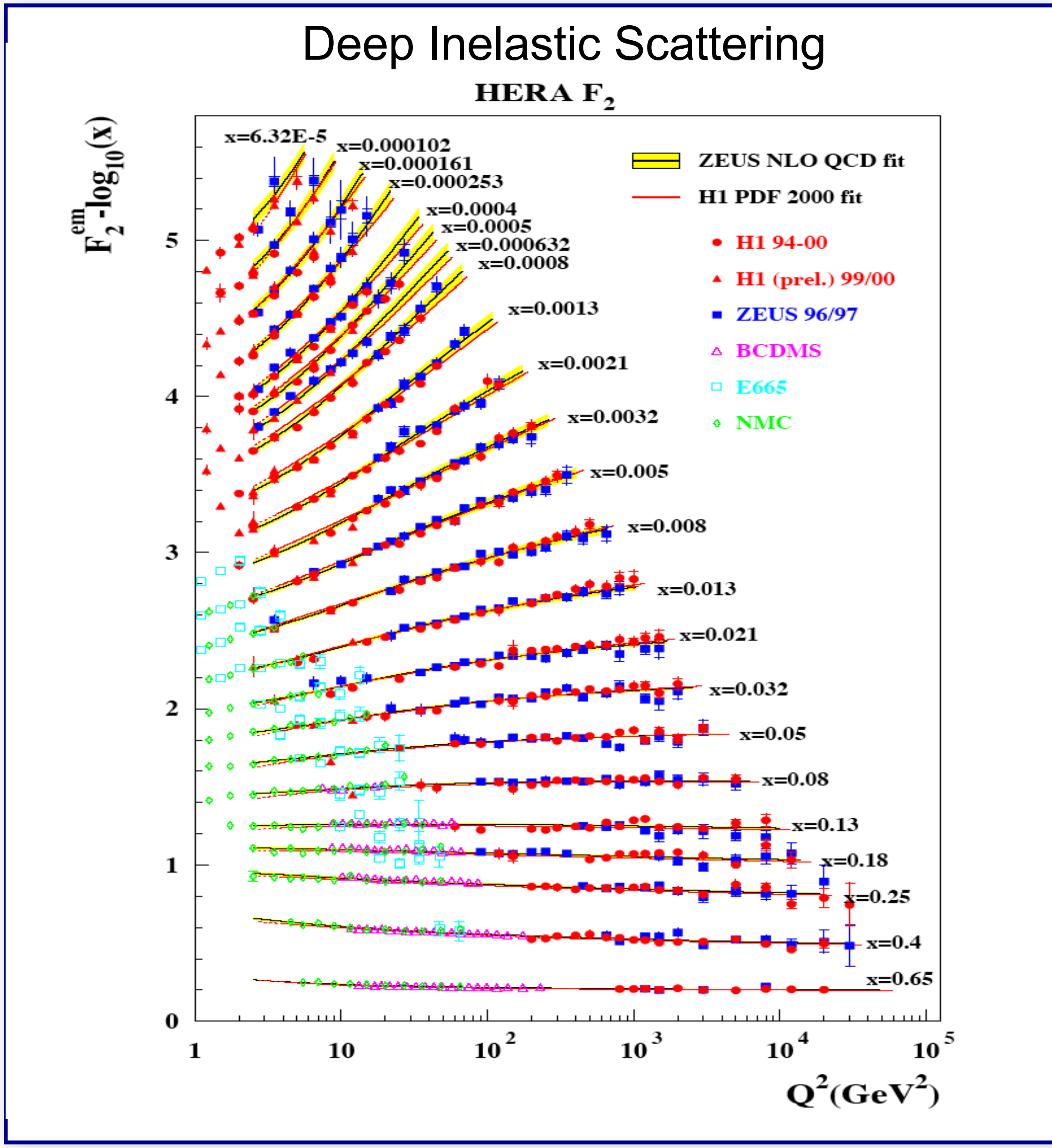
= Fraction of proton momentum carried by typical quark



http://www.physics.smu.edu/scalise/cteq/

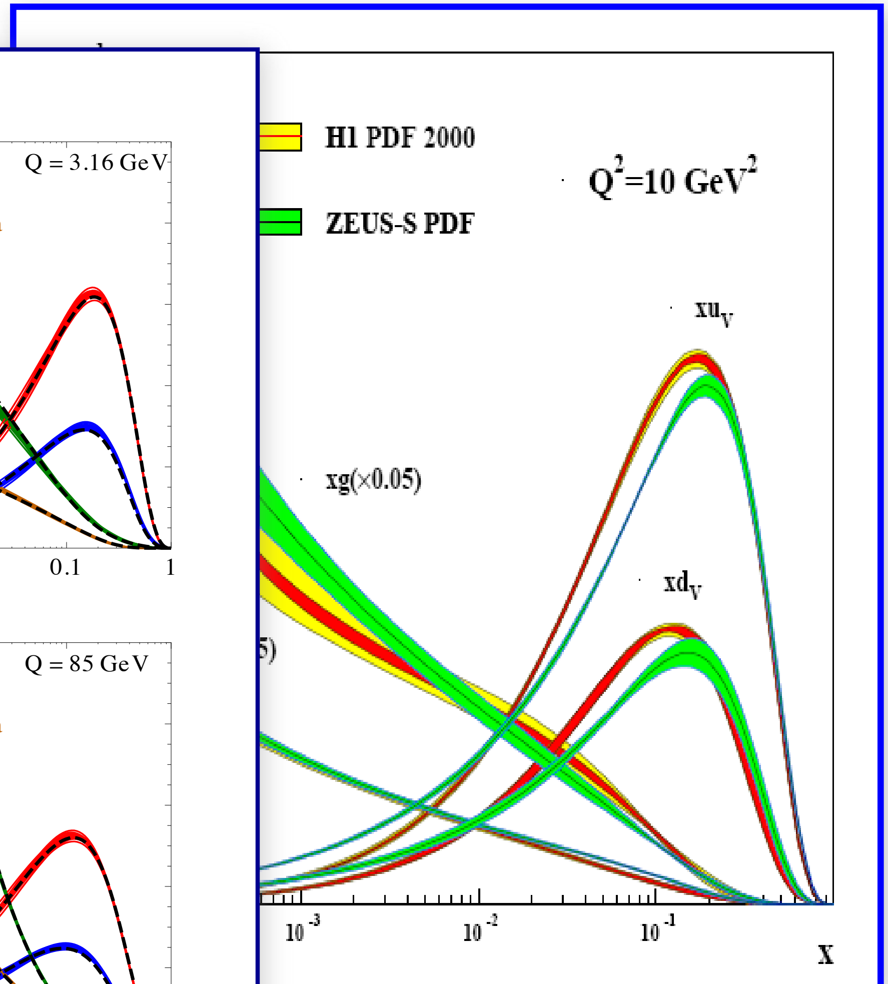
Determined from a combination of experimental fitting and numerical calculation

Quarks

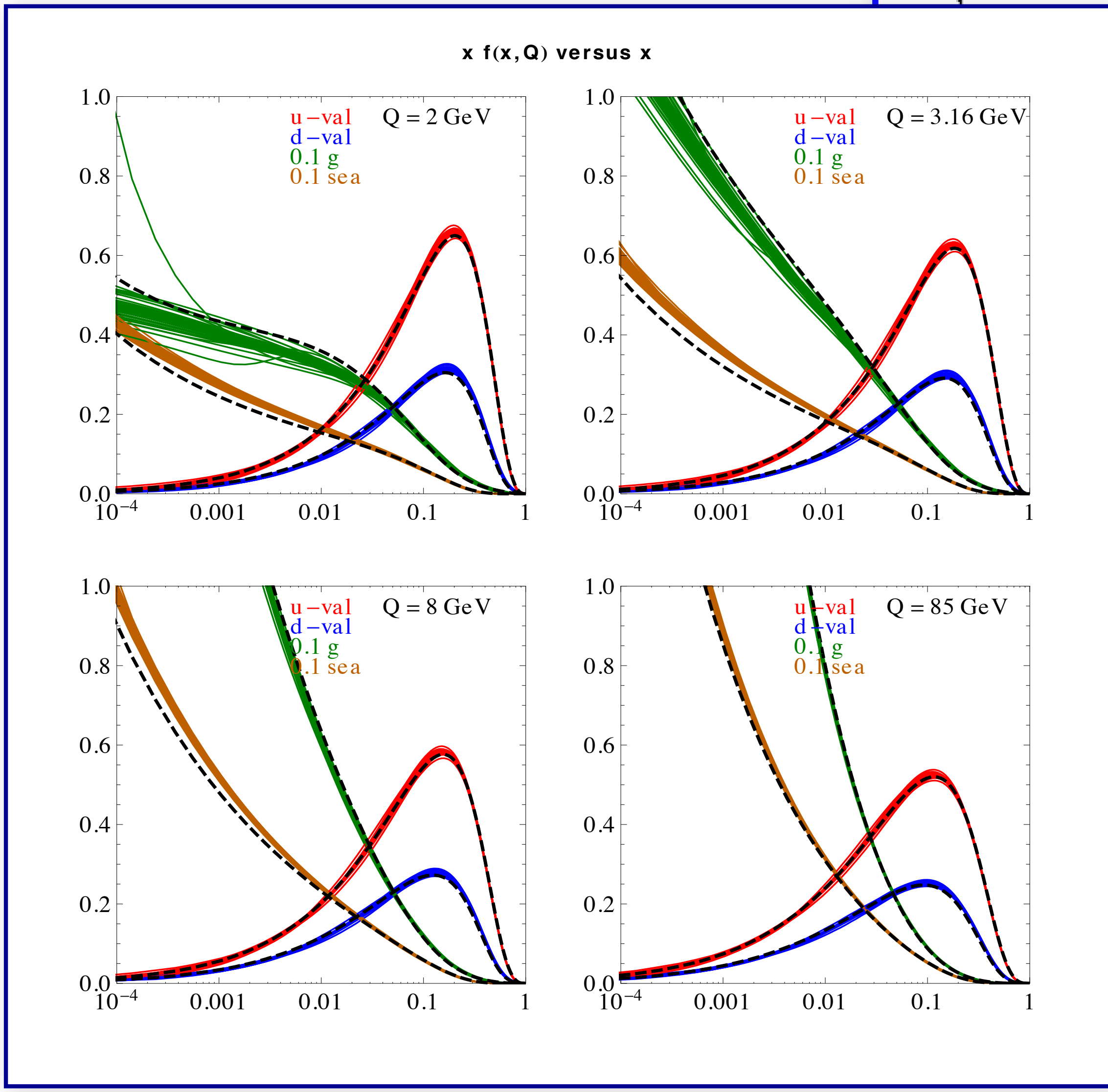


Fitting and Data Analysis, Statistics

Quarks

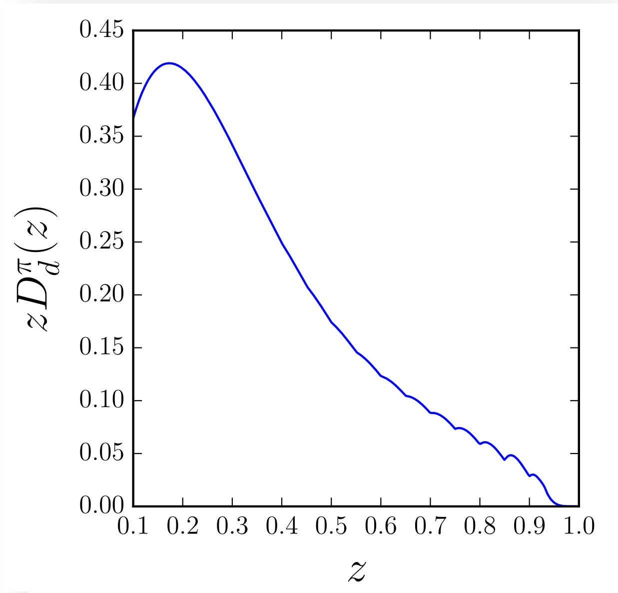


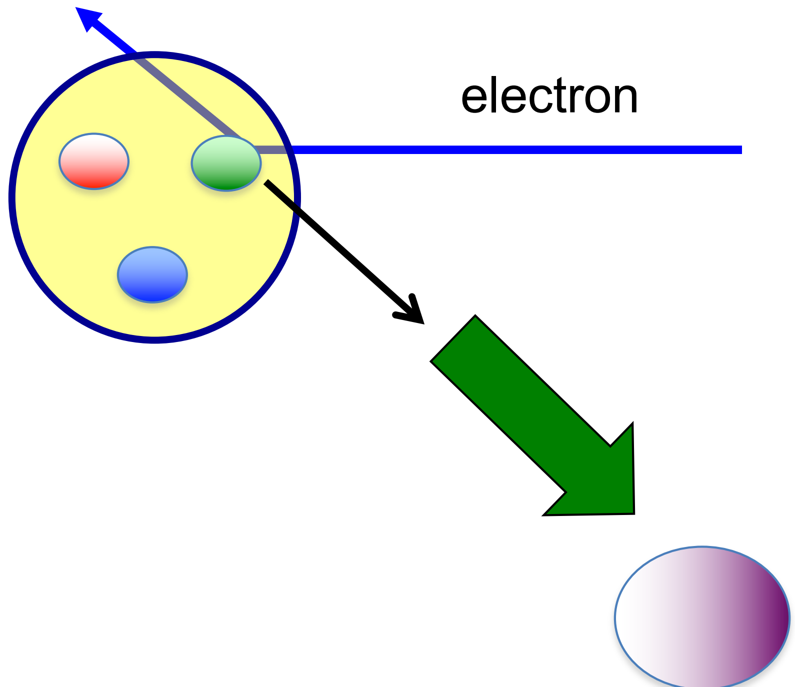
D67, 012007 (2003)



CTEQ10 NNLO **arXiv:1302.6246**

Example: momentum of quarks?



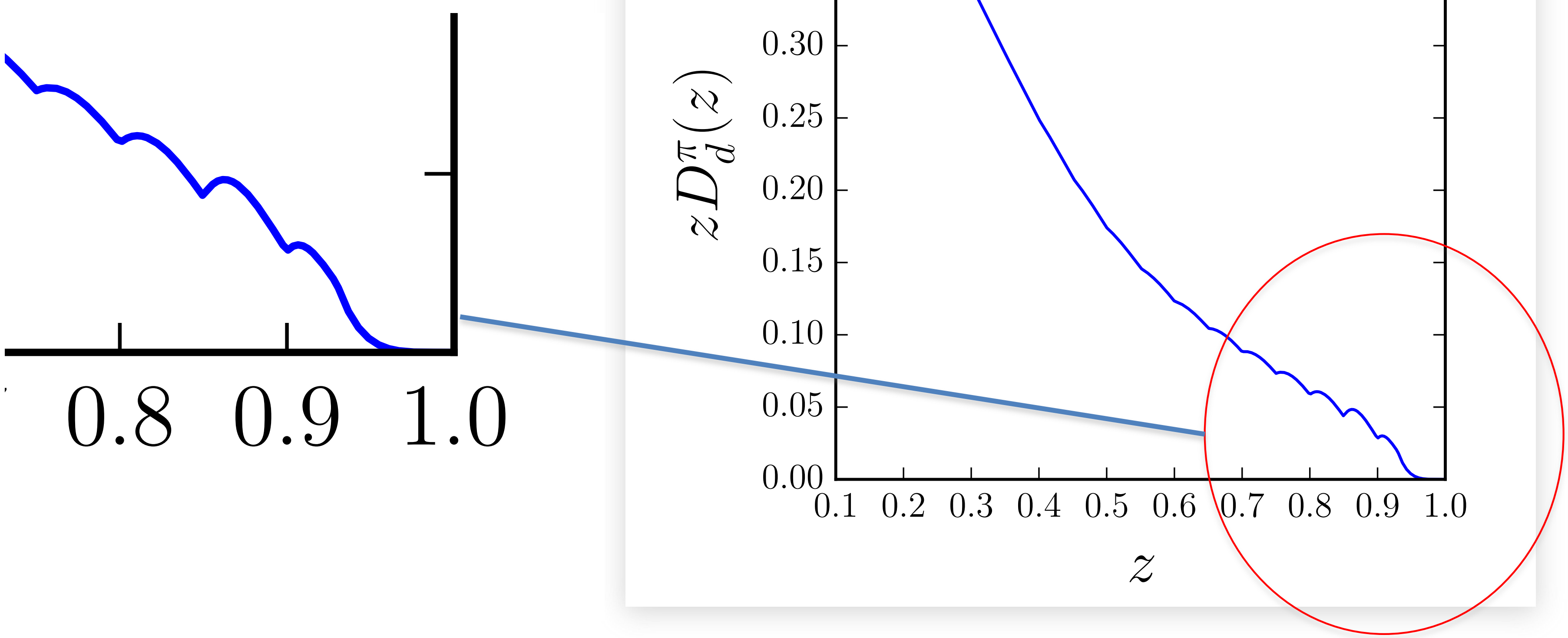


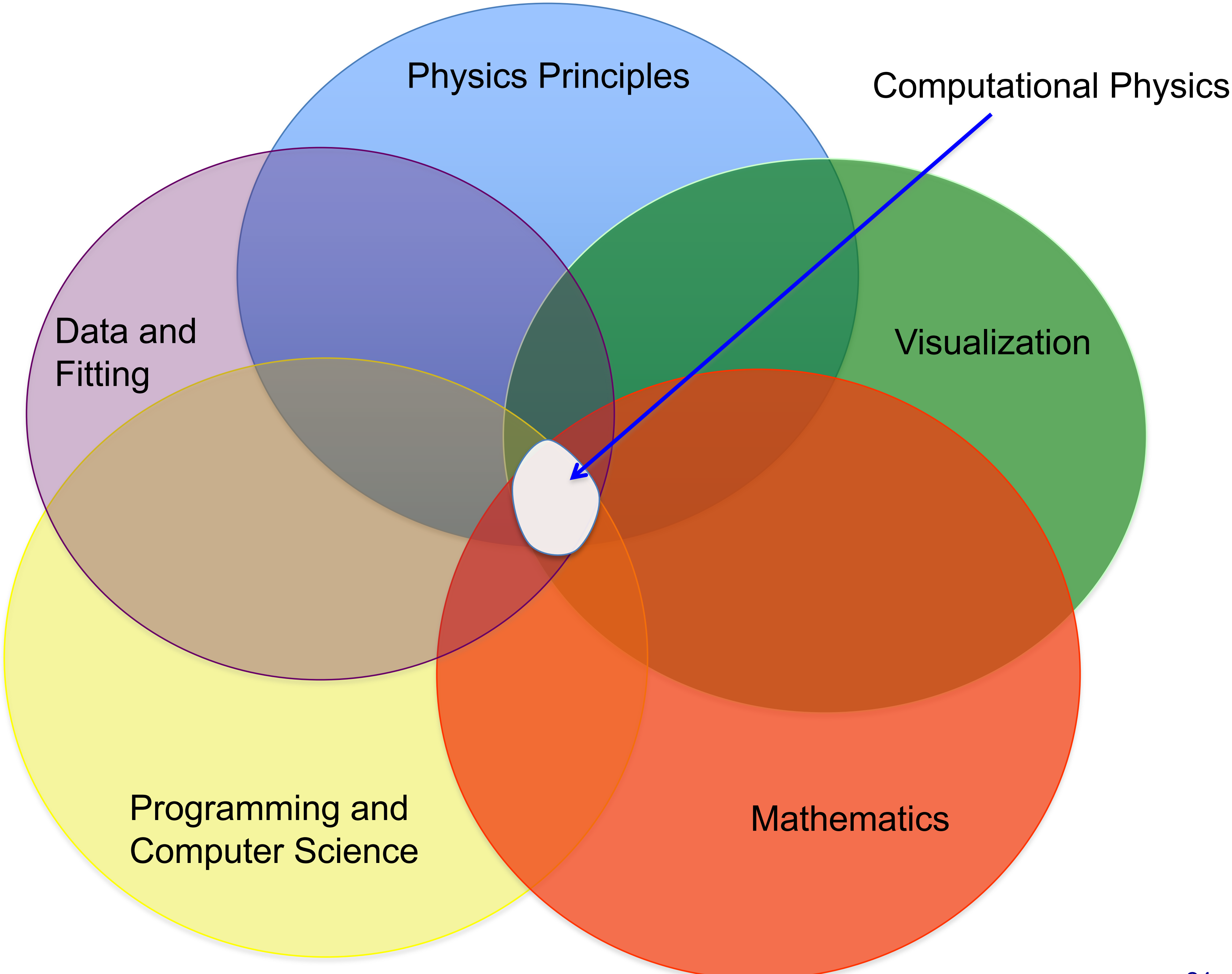
Final observed particle

z ***= Fraction of quark momentum carried by final observed particle***

Quarks

- What's going on here!?




Physics Principles
Computational Physics
Data and
Fitting
Visualization
Programming and
Computer Science
Mathematics

Tools For Computational Physics

High Performance Computing

- NERSC
  - www.nersc.gov
- Google
  - https://cloud.google.com/compute/
- Amazon
  - https://aws.amazon.com/hpc/
- High Throughput Computing
  - http://www.opensciencegrid.org/

Some Operating Systems

Alive

- Windows
- Linux
- Mac OS
- Unix

Dead by now

- DOS
- IBM OS/2
- VMS
- IBM OS/400

Languages and Compilers

For effective and efficient work you need to:

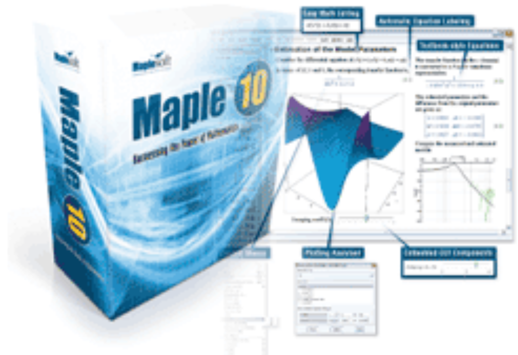
a) Select a language that is right for you now and in nearest future.
You may need to know/learn more than one language.

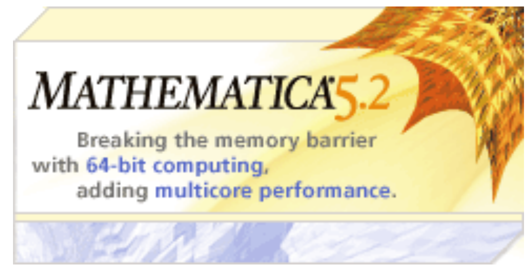
b) Have a good book (also google) to learn the language that you selected

c) Lots of libraries

Problem Solving Environments

- Maple
- Mathematica
- MatLab
- MathCad
- Derive





Problem Solving Environments

- A problem-solving environment is good for small and medium projects
- Prototyping
- Programming with compiled languages gives more control, flexibility

Libraries, Modules, existing programs etc...

- Avoid thinking of computational resources as black boxes
- How much do you need to customize?
- To what extent can you rely on existing resources?
- Computational physics libraries
- Numerical libraries and repositories

Project Design

- Identify physics ideas and principles
- Identify problem
  - Figure out which equations you need to solve, etc…
- Pseudocode and prototyping
- Draw diagrams
  - Always sketch a flowchart
- Code: Commenting and structuring
- Modularize
- Visualize output
- Validation:
  - Check simple cases and limiting cases. Compare with alternative methods.

Project Design

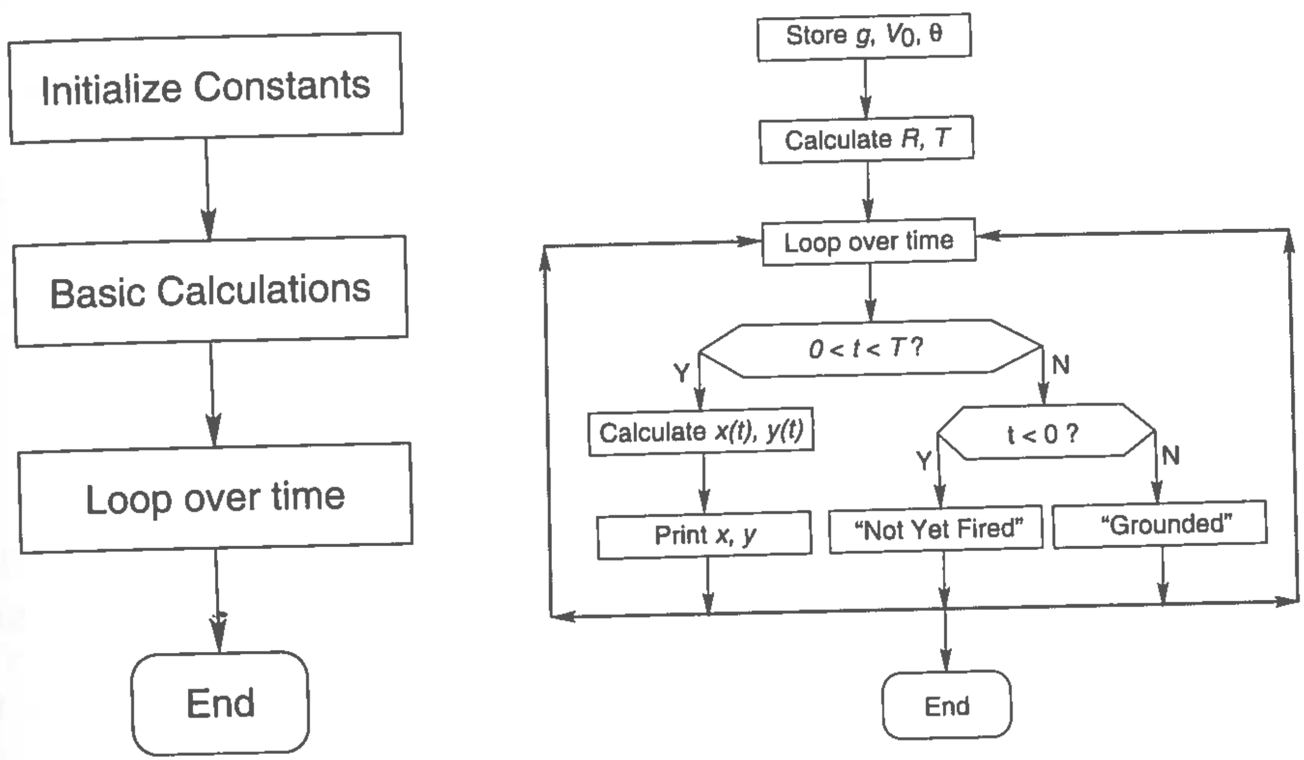


Figure 1.5 A flowchart illustrating a program to compute projectile motion. On the left are the basic components of the program, and on the right are some of its details. When writing a program, first map out the basic components, then decide upon the structures, and finally fill in the details. This is called *top-down programming*.

More on Style

- DO
  - Keep it clear and simple
  - Separate logic groups (structured programming)
  - *Internal* comments and *external* documentation and diagrams

- DON’T
  - Use complex logic
  - Tricky techniques
  - Use machine dependent features

More about style

Each program unit should be well documented

- Opening documentation (what program does, when the program was written and modified, list of changes, …)
- Comments to explain key program sections
- Meaningful identifiers
- Labels for all output data

Short Overview of Python

https://docs.python.org/3/tutorial/

- Operations
- Control Structures
- Lists
- Libraries
- Functions
- Jupyter

Operations

https://www.tutorialspoint.com/python/python_basic_operators.htm

- Arithmetic
  - +
  - -
  - *
  - /
  - **
  - %

Operations

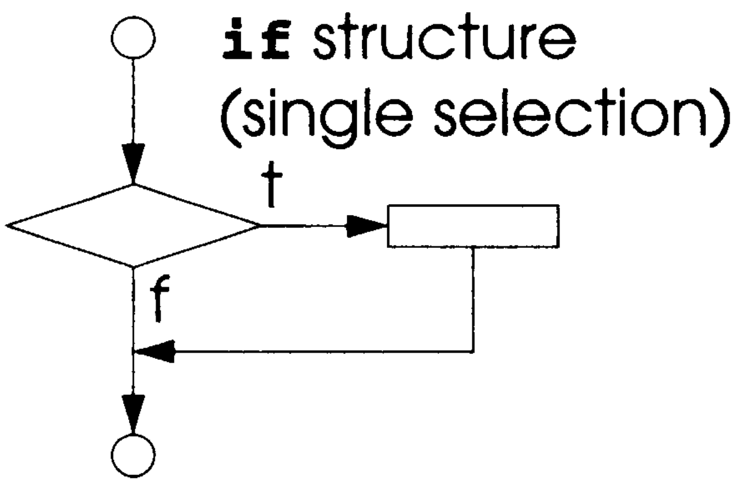
- Assignment
  - =
  - +=
  - -=
  - *=
  - /=
  - **=

Operations

- Comparison
  - ==
  - !=
  - >
  - <
  - >=
  - =<

Control Structures: <u>If</u>

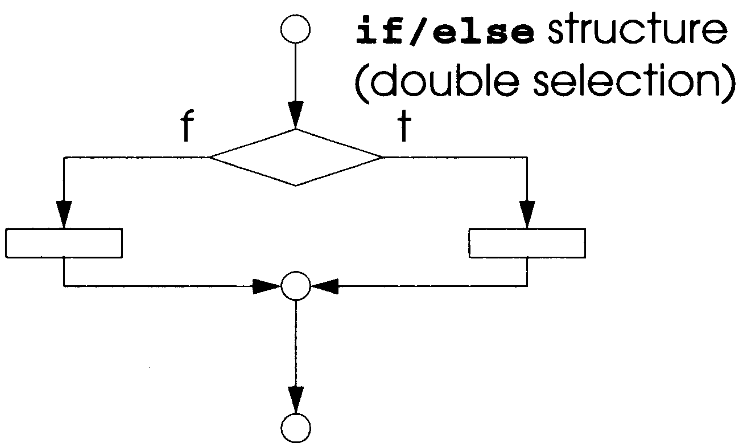
- The if selection structure performs an indicated action only when the condition if **true**; otherwise the condition is skipped




```
>>> x = 2
>>> if x > 0:
...     print('uh-oh!')
...
uh-oh!
>>> 
```

Control Structures: <u>If/elif</u>

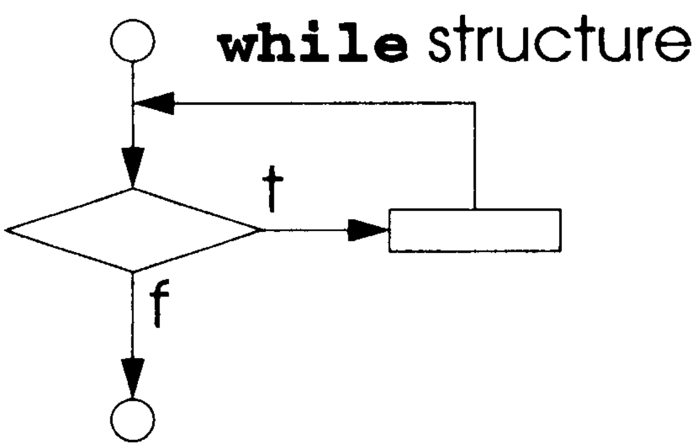
- if/elif selection: allows the programmer to specify that a different action is to be performed when the condition is **true** than when the condition is **false**.




```
>>> x = -1
>>> if x > 0:
...       print('uh-oh!')
... elif x == -1:
...       print('hooray!')
...
hooray!
>>>
```

Control Structures: <u>while</u>

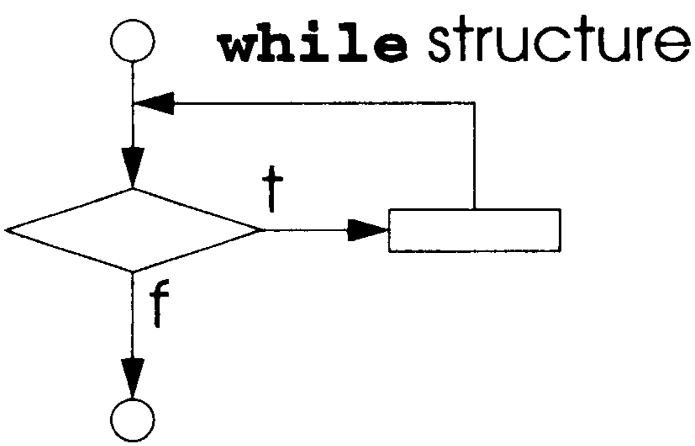
- A repetition structure allows the programmer to specify an action is to be repeated while some condition remains true




```
>>> i = 0
>>> while i < 10:
...     print(i)
...     i = i + 1
...
0
1
2
3
4
5
6
7
8
9
>>>
```

Control Structures: while

- A repetition structure allows the programmer to specify an action is to be repeated while some condition remains true




```
>>> i = 0
>>> while i <= 10:
...     print(i)
...     i = i + 1
...
0
1
2
3
4
5
6
7
8
9
10
>>>
```

Control Structures: for and range

- for iterates over items in a sequence

```
>>> for i in range(1,10):
...     print(i)
...
1
2
3
4
5
6
7
8
9
>>>
```

Lists

- Comma-separated items in square brackets.

```
>>> nums = [1, 5, 3, 2]
>>> nums[0]
1
>>> nums[1]
5
>>> nums|
2
```

```
>>> letts = ['a','b','c']
>>> letts[0]
'a'
```

Starts at 0!

```
>>> nums + letts
[1, 5, 3, 2, 'a', 'b', 'c']
>>>
```

```
>>> nums + letts
[1, 5, 3, 2, 'a', 'b', 'c']
>>> combined = nums + letts
>>> len(combined)
7
```

Functions

Convert a temperature from Celsius to Fahrenheit

```
def CeltoFahr(Ctemp):

    Tfahr = Ctemp*1.8 + 32

    return Tfahr
```

Define a function

Function name

Function arguments

Functions

Convert a temperature from Celsius to Fahrenheit

```
def CeltoFahr(Ctemp):

    Tfahr = Ctemp*1.8 + 32

    return Tfahr
```

Functions

Convert a temperature from Celsius to Fahrenheit

```
def CeltoFahr(Ctemp):

    Tfahr = Ctemp*1.8 + 32

    return Tfahr
```

Operations to get final result

Return final result

Python Libraries

- Many useful libraries available in python!

```
import numpy as np
import math
```

```
>>> import numpy as np
>>>
>>> pie = np.pi
>>> pie
3.141592653589793
>>>
>>>
>>> np.sin(pie/2)
1.0
>>>
```

  - math: https://docs.python.org/3/library/math.html

  - numpy: https://www.numpy.org

Anaconda

- From Wikipedia:
  - Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands... The following applications are available by default in Navigator:
    - JupyterLab
    - Jupyter Notebook
    - QtConsole
    - Spyder
    - Glueviz
    - Orange
    - Rstudio
    - Visual Studio Code

Jupyter

- From Wikipedia:

  “[A] Jupyter Notebook (formerly IPythonNotebooks) is a web-based interactive computational environment for creating… an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media.”

Some Warm-Up Exercises

- Transfer Dropbox exercises to Jupyter
  - Change colors on plots
  - Add titles and axis label
  - Change functions being plotted

- Projectile motion

- Elastic Scattering

Gaussian Probability Distribution

$$P(m) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(m-m_0)^2}{2\sigma^2}}$$

Mean

Variance

Kinematics

Constant acceleration – position as a function of time

$$x(t) = x_0 + v_0 t + \frac{a}{2} t^2$$

Constant acceleration – velocity as a function of time

$$v(t) = x'(t) = v_0 + at$$

Displacement

$$\Delta x(t) = x(t) - x_0$$

Eliminate time

$$v^2 = v_0^2 + 2a(x - x_0)$$

Kinematics

Free-Fall motion:

$$y(t) = y_0 + v_0^y t - \frac{g}{2} t^2$$

$$x(t) = x_0 + v_0^x t$$

$$g = 9.8\ \mathrm{m/s^2} = 32.0\ \mathrm{ft/s^2}$$

- 2D kinematics with constant acceleration: Parabolic

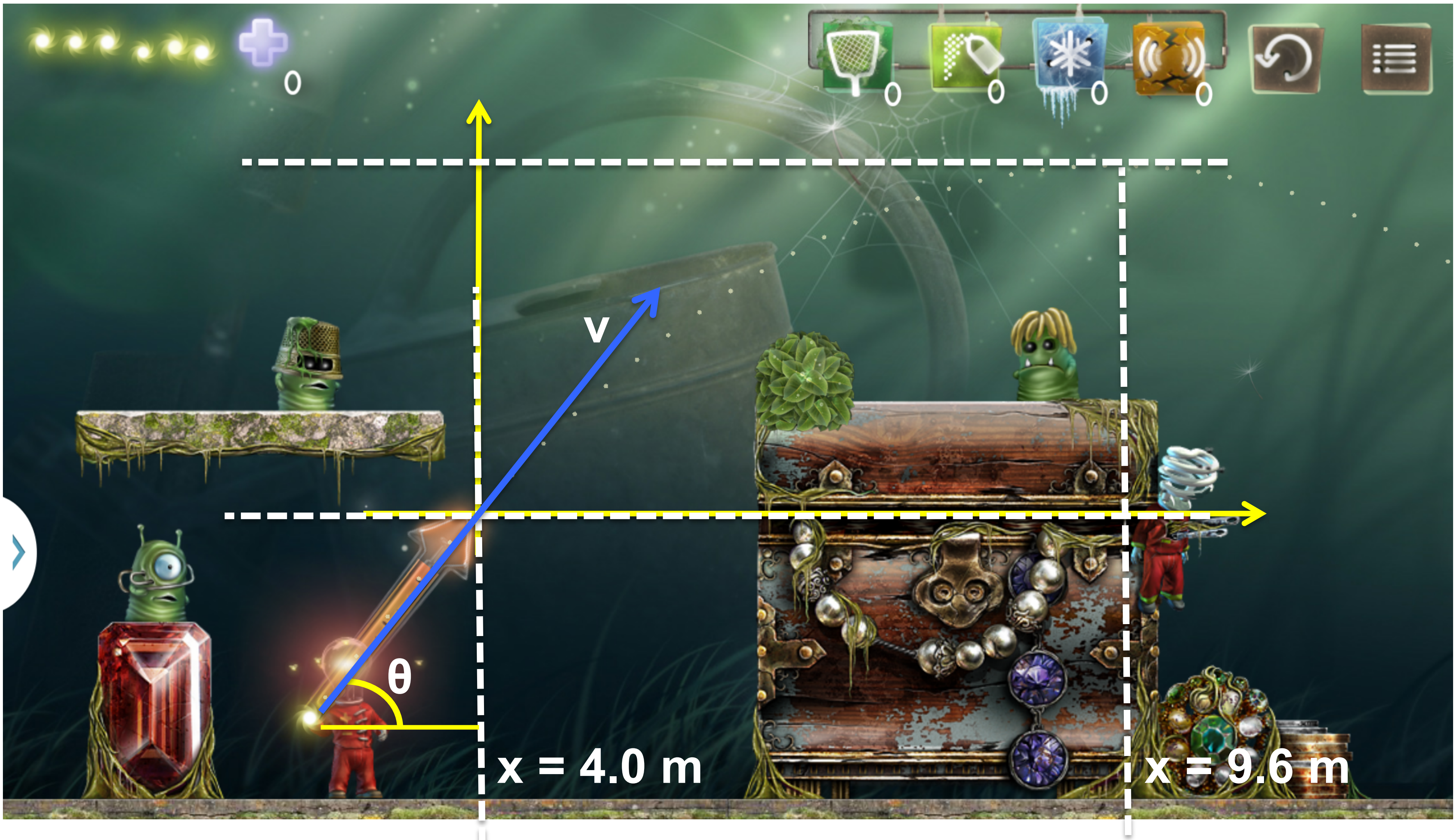
Your software company gave you a task to design a game in which you shoot monsters by controlling the launch velocity $\vec{v}$ and launch angle θ of a "projectile" (ball lightning).

1. Derive an equation for drawing the trajectory of the ball lightning as a function of $\vec{v}$ and θ

(And initial coordinates)

2. Which $\vec{v}$ and θ are needed to shoot the "blonde" monster? The thimble-hat monster?

2. Which $\vec{v}$ and θ are needed to shoot the "blonde" monster? The thimble-hat monster?



Project:

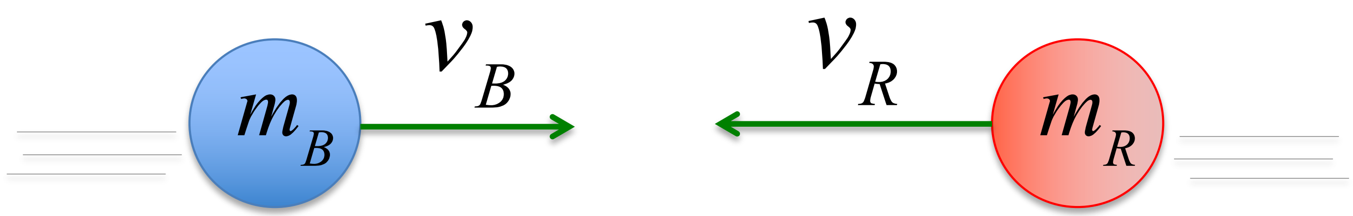
- Start assuming that v_0 is fixed. What angle is necessary?

Newtonian Momentum Conservation

$$\vec{p} = m\vec{v}$$

Elastic Collisions in 1D: Blue Marble vs. Red Marble

Before:

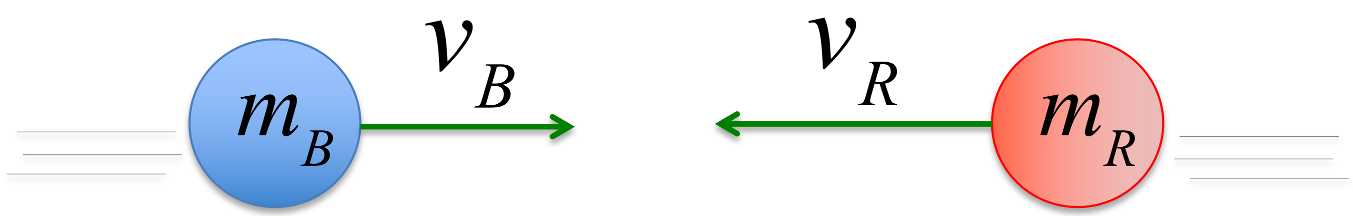


$$p = v_B m_B + v_R m_R$$

$$K = \frac{1}{2} m_B v_B^2 + \frac{1}{2} m_R v_R^2$$

Elastic Collisions in 1D: Blue Marble vs. Red Marble

Before:



$$p = v_B m_B + v_R m_R \qquad K = \frac{1}{2} m_B v_B^2 + \frac{1}{2} m_R v_R^2$$

After:

$$p' = v'_B m_B + v'_R m_R \qquad K' = \frac{1}{2} m_B v'^2_B + \frac{1}{2} m_R v'^2_R$$

Elastic Collisions in 1D

Momentum Conservation:

$$p = p'$$

Elastic Collisions in 1D

Momentum Conservation:

$$p = p'$$

$$v_B m_B + v_R m_R = v'_B m_B + v'_R m_R$$

$$m_B(v_B - v'_B) = m_R(v'_R - v_R)$$

Kinetic Energy Conservation:

$$K = K'$$

$$\frac{1}{2}m_B v_B^2 + \frac{1}{2}m_R v_R^2 = \frac{1}{2}m_B v_B'^2 + \frac{1}{2}m_R v_R'^2$$

$$m_B(v_B^2 - v_B'^2) = m_R(v_R'^2 - v_R^2)$$

$$m_B(v_B - v'_B)(v_B + v'_B) = m_R(v'_R - v_R)(v'_R + v_R)$$

$$v_B + v'_B = v'_R + v_R$$

$$v'_B = v'_R + v_R - v_B$$

$$\boxed{v'_R = v_B \frac{2m_B}{m_B + m_R} + v_R \frac{m_R - m_B}{m_B + m_R}}$$

$$\boxed{v'_B = v_B \frac{m_B - m_R}{m_B + m_R} + v_R \frac{2m_R}{m_B + m_R}}$$

Project:

- When do the marbles collide?

- Plot their trajectories before and after the collision.

- Make their initial positions and velocities probabilistic.