Visualizing Physical Models (and their consequences)

Ruth Chabay
Bruce Sherwood

Department of Physics
North Carolina State University
Physical models: Not just about data

• Structure and function
  • protein folding, molecular conformations
  • nanotube electronic and physical properties
• Consequences of physical models: time evolution
  • Solar system evolution
  • Neutrino flux in supernovas
• Emergent behavior in complex systems
  • Chaos (phase space)
  • Molecular dynamics
  • Plasma dynamics (solar events)
• Experimental event identification
  • Monte Carlo simulations and neural networks in high energy event identification
  • Numerical relativity: prediction of gravitational radiation pulse shapes
What do we need to visualize?

- **Physical models** (ontology and action)
  - ball–spring model of a solid
  - magnetic domains in an Ising magnet
  - lattice gas model of fluid dynamics
- **Abstract quantities**
  - force, momentum, fields, flux, energy
- **Spatial and temporal dynamics**
  - fields, states, reactions, currents
“...anyone can imagine a simple radial inverse square field without the help of a picture.”

E. Purcell,

*Electricity and Magnetism*

2d edition, p. 18
The design and operation of a cyclotron is discussed in Section 20.1.4.

(a) Show that the "period" of the motion, the time between one kick to the right and the next kick in the same direction, does not depend on the current speed of the proton (at speeds small compared to the speed of light). As a result, we can place across the dees a simple sinusoidal potential difference having this period and achieve continual acceleration out to the maximum radius of the cyclotron. See Figure 20.80.

(b) One of Ernest Lawrence's first cyclotrons, built in 1932, had a diameter of only about 30 cm and was placed in a magnetic field of about 1 tesla. What was the frequency (= 1/period, in hertz = cycles/second) of the sinusoidal potential difference placed across the dees to accelerate the protons?

(c) Show that the equivalent accelerating potential of this little cyclotron was about a million volts! That is, the kinetic energy gain from the center to the outermost radius was \( K = e \Delta V_{eq} \), with \( \Delta V_{eq} = 1 \times 10^6 \) volts.

(d) If the sinusoidal potential difference applied to the dees had an amplitude of 500 volts (that is, it varied between +500 and -500 volts), show that it took about 65 microseconds for a proton to move from the center to the outer radius.

Figure 20.80 A cyclotron (Problem 20.4)
Simple examples

- Magnetic field of a moving particle
- Gauss’s Law
- Magnetic field of a current loop
  - Ball-spring model of a solid
  - Energy in a 3D mass-spring system
  - Ising magnet
Intervention topics

Charles (phase 2)

Position update

Momentum definition

Momentum principle

Force

$t, \Delta t$

Program Structure

I.C.

line number

0 100 200 300 400 500 600
Student Mechanics Programs

- VPython intro
- Motion with piecewise constant velocity
- Gravitational force vector in 3D
- Planet around fixed star; binary star system
- Spring-mass oscillator
- Energy graph for planet
- Energy graph for damped spring-mass oscillator
- Rutherford scattering (discovery of nucleus)
- Quantum statistical mechanics (temperature dependence of heat capacity)
E&M Programs

- VPython intro
- Electric field of point charge
- Electric field of dipole
- Electric field of a charged rod
- Magnetic field of a moving charge
- Moving charge in a magnetic field
- Positron in an electromagnetic wave
VPython

http://vpython.org
Visualizing a principle

• The momentum principle
  • a.k.a. Newton’s second law
    \[ \Delta p = \bar{F} \Delta t \]

• The superposition principle
The Newtonian Synthesis

Open-ended prediction of motion into the future

\[ \vec{F} = f(\vec{r}) \quad \text{Force as a function of position} \]

\[ \Delta \vec{p} = \vec{F} \Delta t \quad \text{The momentum principle} \]

\[ \dot{\vec{p}} \leftarrow \vec{p} + \Delta \vec{p} \quad \text{Update momentum} \]

\[ \vec{r} \leftarrow \vec{r} + \vec{v} \Delta t \quad \text{Update position} \]

do it again
The Momentum Principle

student program
Programming: Why?

- No black boxes
  - Student codes all the physics
- Same fundamental principles invoked in different situations
- Links multiple representations
  - Equations
  - Code
  - 3D Animation of motion / visualization
  - Graph
Matter & Interactions

New introductory calculus-based physics course for engineers and scientists emphasizing:

• Small number of fundamental principles
• Atomic nature of matter
• Modeling physical systems
  Including computational modeling
PY205/PY208 at NC State

- Calculus-based intro course
  - Engineering and science students
- 3 interactive lectures / week
  - 100 students per section
  - 12 sections in Spring 2005
- 1 two-hour studio lab / week
  - 24 students per section
- Computer homework system (WebAssign)
Interactive Lectures

- Computer visualizations
- Interactive lecture demonstrations
- Student response system
Q3: By thinking about the situation, make an argument to predict the magnitude of the electric field at the center of a uniformly charged ring of radius R carrying a charge +Q. Then utilize the formula we just derived to confirm this result.

1) \( E = \frac{1}{4\pi\varepsilon_0} \frac{Q}{Z^2} \)
2) \( E = \frac{1}{4\pi\varepsilon_0} \frac{Q}{Z^2 R^2} \)
3) \( E = 0 \)
4) \( E = \infty \)
Discussion of student responses
Interactive Studio Labs

- Teaching assistant (TA): physics graduate student
- Teaching assistant assistant (TAA): undergraduate who did well in course

Coaching 24 students who work in groups of two or three
Experiments closely tied to theory
Group work: solving large, difficult problems
Writing a computer program to model a system in 3D (VPython)
Arbor Scientific
Air Powered Projectile

Projectile Speed
- Erase Track

Initial Launch Angle

Position of Projectile
- \( x \): 95.06 m
- \( y \): 24.64 m
- \( \theta \): -26.83°

Velocity of Projectile
- \( V_x \): 24.75 m/s
- \( V_y \): -12.52 m/s
- \( |V| \): 27.73 m/s

Press "Run" to launch the projectile.
Press "Reset" to replace the projectile on the launch pad. Adjust the initial speed and direction by moving the sliders.
M.U.P.P.E.T.
University of Maryland 1980s

Turbo Pascal

Output: graphs only

Needed numerical analysis (Runga-Kutta, etc.) because computers were slow

Large amount of setup code provided to students

http://www.physics.umd.edu/perg/muppet


PROGRAM Projectile10;  (* proj10.pas *) 
{
***************
| Program to calculate motion of |
| a particle in 1D with gravity |
| and air resistance using RK2. |

USES
Cr, Do, Graph, Printer, MUPPET;

CONST
numData : integer = 200;  (* number of points to plot *)
g : Real = 9.8;  (* m/sec/sec *)
VAR
x, t : DataVector;  (* time, position *)
v, a : DataVector;  (* velocity, accel *)
X0, V0 : Real;  (* initial coords, velocity *)
m : Real;  (* mass *)
b : Real;  (* air reals coeff. *)
dt : Real;  (* time step *)
i : Integer;  (* loop variable *)
IC : screen;  (* data screen *)
act : CHAR;  (* control character *)
{
The types 'DataVector' and 'screen' are defined inside the unit MUPPET.}

(*--------------- Physics Procedures ---------------*)

FUNCTION Force(x, v, t : Real) : Real;
BEGIN
Force := -mg - b*v*vabs(v);
END;

(*--------------- Mathematics Procedures ---------------*)

(* Second order Runge-Kutta routine for stepping *)
(* from variables at time t (In variables) to *)
(* variables at time t + dt (Out variables) *)

PROCEDURE StepRK2(xn, vtn, ttn, st : Real);
VAR
xHalf, vHalf : Real;
BEGIN
xHalf := ttn + 0.5*st;
vHalf := vtn + 0.5*st*vtn;

xNew := xn + xHalf*st/m;
vNew := vtn + vHalf*st/m;
END;

(*--------------- Graphics Procedures ---------------*)

PROCEDURE GraphSetup;
BEGIN
GraphBackColor := DarkGray;
DefineViewport(1, 0.55, 0.55, 0.25, 0.25);  (* Define ViewPort *)

ViewPort1 := DefineViewport(1, 0.25, 0.35, 0.75, 0.25);  (* Define Scale 1 *)

ViewPort2 := DefineViewport(0.5, 0.5, 0.35, 0.35, 0.25);  (* Define Scale 2 *)

END;

PROCEDURE PlotIt(viewPort, color: Integer; x, y: DataVector; nameLabel: BigStr);
BEGIN
SetColor(color);
SelectScale(viewPort);
OpenViewport(viewPort);
Axis(0, 0, 1.2, 0.5);
PlotData(x, y, numData);
PutLabel(Inside, nameLabel);
END;

(*--------------- Data Screen Procedures ---------------*)

PROCEDURE MakeDataScreen;
BEGIN
DefInputPort(0.0, 0.45, 0.0, 0.5);
A[0] := "UM.P.R.I.T.";
A[1] := "University of Maryland";
END;

BEGIN
MUPPETInit;
MakeDataScreen;
GraphSetup;
REPEAT
GetScreenData(n, b, x0, v0, dt);
IF Escaped THEN BEGIN
   MUPPETdone := TRUE;
   EXIT;
END;
BEGIN
   \( m = 0.14 \) kg
   \( \text{Mass} \)
   \( \text{Coefficient, } b = 0 \) kg/m/s
   \( \text{Air Resistance} \)
   \( \text{Initial Conditions} \)
   \( \text{Time step, } dt = 0.05 \text{ sec} \)
   \( \text{Initial position, } x0 = 0 \text{ m} \)
   \( \text{Initial velocity, } v0 = 30 \text{ m/sec} \)
   LoadScreen(IC, 17);
END;
FOR i := 1 to numData DO
   Solve the equation:
   \( \text{StepRK2(x[i-1], v[i-1], t[i-1], \dots, x[i], v[i], t[i], \dots, a[i])} \)
   \( \text{Message('Press for new data, to quit?')}; \)
   PlotIC(1, lightGreen, t, x, 'X vs T');
   PlotIC(2, lightRed, t, v, 'V vs T');
   Source[1] := ReadKey;
   UNTIL ord(Source) = 27;
MUPPETdone := TRUE;
END;

(*--------------- Main Program ---------------*)
Press <ENTER> for new data, <ESC> to quit

M.U.P.P.E.T.
University of Maryland

PROJECTILE PROGRAM: 1D
F = -mg - bv*abs(v)

PARAMETERS
Mass \( m = 0.14 \) kg
Air Resistance
Coefficient, \( b = 0.005 \) kg/m
Time step, \( dt = 0.050 \) sec

INITIAL CONDITIONS
Position: \( x_0 = 0 \) m
Velocity: \( v_0 = 30 \) m/sec
Constraints

- Many students have never written a program before this
- Very little time can be spent on programming instruction

Therefore

- Teach minimal set of programming concepts
- Language and environment must be easy to learn and use (VPython)
What difficulties do students have with programming?
Interview Study
Matt Kohlmyer

• Paid volunteers from two M&I classes
  • Spring 2003: N=4
  • Fall 2003: N=5

• Three 1-hour-long interviews per student

• Work on computer programs

• Think-aloud protocols
  • For detailed data on student reasoning
  • Videotaped and transcribed

• If stuck, could ask questions, or look at VPython syntax help
Orbit problem:

- Moon orbits Earth
- Given: orbit is circular, period is 28 days, masses of moon and earth

Students had previously written an orbit program in class.
Quantitative analysis of dialogue

- Count lines of transcribed dialogue
- Interviewer gave more hints on force than on any other topic
3D force calculation

\[ \bar{r} = \bar{r}_{\text{planet}} - \bar{r}_{\text{moon}} \]
\[ |\bar{r}| = \sqrt{x^2 + y^2 + z^2} \]
\[ \hat{r} = \frac{\bar{r}}{|\bar{r}|} \]
\[ |\vec{F}| = G m_1 m_2 / |\bar{r}|^2 \]
\[ \vec{F} = |\vec{F}| \hat{r} \]

Steps encapsulated in:
\[ \vec{F} = \left( G m_1 m_2 / |\bar{r}|^2 \right) \hat{r} \]
Force as scalar

\[ \text{moon.rmag}=3.8\text{e}8 \]
\[ \text{Fnet}=6.7\text{e}-11* (\text{moon.m*earth.m})/\text{moon.rmag}^2 \]

Error on run: adding vectors and scalars when updating momentum

\[ \text{moon.p}=\text{moon.p}+\text{F*deltat} \]

Kyle, phase 2 (others made similar errors)
Force in constant direction

\[ F_{net} = \text{vector}(0, -F_{mag}, 0) \]

- Direction does not update with time
- Possible confusion with \( mg \)?
- Force in direction of motion?
- Two other students: \( F_{net} = \text{vector}(F_{mag}, 0, 0) \)

Kyle
Discrimination between vectors

I: Do you remember how we defined Fnet, so that it's always pointing towards the earth?
K: You take the, you take the uh, final position minus the initial position.
I: Yeah. Yeah, that's gonna be involved.
K: And I need to define, or I can say earth dot pos, minus moon dot pos.

\[ F_{net} = \text{earth.pos} - \text{moon.pos} \]

*Interviewer* explained: this is not the force, only a vector in the same direction as the force.
Need for unit vector

Kyle’s fix:

\[ F_{net} = (\text{earth.pos} - \text{moon.pos}) \times F_{mag} \]

Interviewer explains: Magnitude too large. Kyle does not understand. Interviewer shows a written numerical example, and explains \( \hat{r} \)-hat. Kyle then remembers \( \hat{r} \)-hat from lecture and homework.
Why is force difficult?

• Combines many different quantities and concepts
  • Force magnitude
  • Relative position vector
  • Magnitude of relative position vector
  • Unit vector
• Changing force (magnitude & direction)
• VPython syntax still not familiar
Physics or programming?

Computer program requires correctness in features that might be ignored in written work:

- Force is not a scalar
- Force is not constant in an orbit
- You can’t divide by a vector

\[ F = G \times (\text{moon.m}\times\text{earth.m})/r^{**2} \]

where \( r \) is a vector
Revised instructional sequence (S2005)

- Lab 1: VPython intro (objects, position vectors, simple loops)
- Lab 2: piecewise constant velocity motion, constant force motion
- Lab 3: gravitational force vector at multiple static locations
- Lab 4: bring it all together—planet in elliptical orbit around star
Physics for the 21st Century: New Content

• Microscopic (atomic-level) view of matter
  • No atoms in traditional course

• Computational modeling of physical systems
  • No computer modeling in traditional course

• Application of fundamental principles to a wide range of systems (from nuclei to stars)
  • Traditional course emphasizes plugging numbers into specific formulas for specific situations; not a good preparation for attacking new problems

• Solving complex, real-world problems
  • Traditional course emphasizes sanitized unrealistic situations
**Setup**

3D graphics

Create objects, give initial pos.

Constants

Initial momentum

Timestep

Initialize time

```python
from visual import *

Sun = sphere(pos=vector(0,0,0), radius = 1e10, color=color.yellow)
Earth = sphere(pos=vector(1.5e11,0,0), radius = 5e9, color=color.cyan)
Earth.trail=curve(color=Earth.color)

Earth.m = 6e24
Sun.m = 2e30
G=6.67e-11

Earth.p = Earth.m*vector(0,2e4,0)

deltat = 1e4

t=0
```
Physics loop

```python
while t < 3e7:
    r = Earth.pos - Sun.pos
    rmag = sqrt(r.x**2 + r.y**2 + r.z**2)
    rhat = r/rmag

    Fmag = G*moon.m*planet.m/rmag**2
    F = -Fmag*rhat

    Earth.p = Earth.p + F*deltat
    Earth.pos = Earth.pos + Earth.p/Earth.m*deltat

    Earth.trail.append(pos=Earth.pos)
    t = t + deltat
```
3D Vector Force Calculation

<table>
<thead>
<tr>
<th>Rel. pos. vector &amp; unit vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grav. force vector</td>
</tr>
</tbody>
</table>

```python
while t<28*24*60*60:
    r=planet.pos-moon.pos
    rmag=sqrt(r.x**2 + r.y**2 + r.z**2)
    rhat=r/rmag

    Fmag=G*moon.m*planet.m/rmag**2
    F=Fmag*rhat

    moon.p=moon.p+F*deltat
    moon.pos=moon.pos+moon.p/moon.m*deltat

    moon.trail.append(pos=moon.pos)
    t=t+deltat
```
The traditional calculus-based introductory physics course

- Where are the fundamental concepts?
  - Force: chapter 5
  - Energy: chapter 7
  - Momentum: chapter 9
  - Angular momentum: chapter 12

- What do students see as most fundamental?

\[ x = \frac{1}{2} at^2 \]
3D Vectors

\[ \vec{r} = \langle 6, -19, -23 \rangle m - \langle -23, 5, -25 \rangle m \]

\[ |\vec{r}| = \sqrt{29^2 + (-14)^2 + (-58)^2} m \]

\[ \hat{r} = \vec{r} / |\vec{r}| \]
Typical rationale for introductory physics

- Learn systematic problem solving
- Learn to separate world into system & surroundings
- Practice applying mathematics

- See the unity of physics?
- See the power of fundamental principles?
The traditional calculus-based introductory physics course

Instruction focuses on solutions to classes of problems (constant acceleration, circular motion at constant speed, static equilibrium, parallel resistors, RC circuits…) rather than reasoning from fundamental principles.

Therefore, students see the course as a collection of unrelated problem types.
Physics Education Research (PER) has focused on teaching the traditional course more effectively. However, we need to ask:

What *should* we teach?

Research shows that a large investment by teachers and students is required for effective learning.

What is important enough to be worth a large investment on the part of students and teachers?

We need clear goals on which to base decisions.
Physics for the 21st Century

- Emphasize a small number of fundamental principles
  (unify mechanics & thermal physics; electrostatics & circuits)

- Integrate contemporary physics
  (atomic viewpoint; connections to chemistry, biology, materials science,
   nanotechnology, electrical engineering, nuclear engineering, computer
   engineering, ...)

- Engage students in physical modeling
  (idealization, approximation, assumptions, estimation)

- Introduce computational physics
  (now a partner of theory and experiment)

- Omit topics that do not contribute to this goal.
Modeling the physical world

- **Students** should see clearly that a small number of fundamental principles can explain a very wide range of phenomena.

- **Students** should see the place of classical physics in the larger physics framework (including the atomic nature of matter, quantum mechanics, relativity).
Research Supporting Development

Theoretical
  New views of standard physics
  Cognitive task analyses
  Predictions based on models of learning

Experimental
  Analysis of students’ written work
  Think-aloud protocol analysis (video)
  Fine-grained assessment
  Large scale assessment

Time Scale
  14 years (and still going…)
Supporting text:

- **Matter & Interactions I: Modern Mechanics**
  mechanics; integrated thermal physics

- **Matter & Interactions II: Electric & Magnetic Interactions**
  modern E&M; physical optics

John Wiley & Sons, 2002
Bobby (1)
Norman (1)
Paul (1)
Richard (1)
Andrew (2)
Charles (2)
Kyle (2)
Nick (2)
3D Vector Force Calculation

```python
while t < 3e7:
    r = Earth.pos - Sun.pos
    rmag = sqrt(r.x**2 + r.y**2 + r.z**2)
    rhat = r/rmag

    Fmag = G*moon.m*planet.m/rmag**2
    F = -Fmag*rhat

    Earth.p = Earth.p + F*deltat
    Earth.pos = Earth.pos + Earth.p/Earth.m*deltat

    Earth.trail.append(pos = Earth.pos)
    t = t + deltat
```
Force calculation

\[ \vec{r} = \vec{r}_{Sun} - \vec{r}_{Earth} \]

\[ |\vec{r}| = \sqrt{x^2 + y^2 + z^2} \]

\[ \hat{r} = \frac{\vec{r}}{|\vec{r}|} \]

\[ |\vec{F}| = \frac{GMm}{|\vec{r}|^2} \]

\[ \vec{F} = -|\vec{F}| \hat{r} \]
2\textsuperscript{nd} session interview

- Moon orbit program
- Took place after about 6 weeks
- Students had completed several programming assignments
  - Including a model of a planet orbiting a star, which was similar to interview task
Introductory Calculus-Based Physics for Engineers & Scientists:

*Why computation?*

- Authentic physics
  - Theory + Experiment + Computation
- Modeling complex systems
  - No analytical solutions
- Fundamental principles
  - Time evolution (Newtonian Synthesis)
  - Vectors as tools
- 3D visualization
Two kinds of dynamics problem

- Given known motion, deduce unknown forces.
- Given force law (and initial conditions), predict unknown motion.
Traditional Problems

Given known motion, deduce force
Open-ended Problems

Given initial conditions and force law, predict unknown motion

- Binary star
- Rutherford scattering
E&M: 3D Fields

- Superposition
- Variation in time and space
- 3D vectors as tools
Programming: How?

Many students have never written a program before

- Must be easy to learn
  - Minimum set of programming concepts
  - No interface or graphics coding
  - Student concentrates on physics
- No fancy algorithms
  - Computers are now very fast!
  - Just take very small steps
VPython:
3D programming for ordinary mortals

Python programming language
IDLE interactive development environment
Visual 3D rendering module
Numeric fast array manipulation module

Free
Open source
Multiplatform: Windows, Linux, MacOSX
Originated by David Scherer

http://vpython.org
Write a VPython program

VPython

Produces 3D real-time navigable animations as a side effect of physics computation

Mean free path of a gas molecule

Write a VPython program
Visualizing a principle

• The momentum principle

• The superposition principle
  • Finding the net field at a location in space
The superposition principle

To calculate the field due to many charged particles:

- Add up the contribution of each particle or group of particles
- These contributions are not changed by the presence of other particles