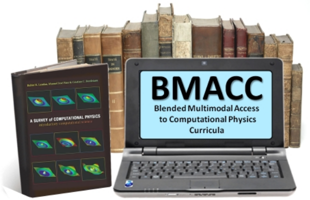
# Computational Physics Examples for Physics Courses

## Demonstrating Computational Scientific Thinking

Rubin H Landau

Oregon State University, Founder Emeritus
Computational Physics for Undergraduates Degree Program

SC12 Education Program, Salt Lake City



# Contributing Group

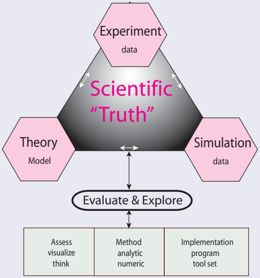
## In Addition to All the Suffering Students

- Manuel J Paez, University of Medellin, Colombia, SA, CoAuthor
- C. E. Yaguna, J. Zuluaga, Oscar A. Restrepo, Guillermo Avendano-Franco
- Cristian Bordeianu, University of Bucharest, Romania, CoAuthor
- Paul Fink, Robyn Wangberg, CoAuthors
- Justin Elser, Chris Sullivan (system support)
- Sally Haerer, Saturo S. Kano (consultants, producers)
- Melanie Johnson (Unix Tutorials)
- Hans Kowallik (Computational Physics text, sounds, codes, LAPACK, PVM)
- Matthew Ervin Des Voigne (tutorials)
- Bertrand Laubsch (Java sound, decay simulation)
- Jon J Maestri (vizualizations, animations, quantum wave packets)
- Al Stetz, David McIntyre (First Course)
- Juan Vanegas (OpenDX)
- Connelly Barnes (OOP, PtPlot)
- Phil Carter, Donna Hertel (MPI)
- Zlatko Dimcovic (Wavelets, Java I/O)
- Joel Wetzel (figures)
- Pat Cannan, Don Corliss, Corvallis HS (N-D Newton Raphson)
- Brian Schlatter
- Daniel Moore, (REU, Sum 98; Whitman College, WA)
- Justin Murray, (REU, Sum 98; Weber State University, Ogden, Utah
- Brandon Smith, (REU, Sum 97; Chico State/SDSC, CA)
- Paul D. Hillard, III (REU, Sum 96; Southern Univ, LA)
- Kevin Wolver, (REU, Sum 96; St Ambrose, IA)

# Computational Scientific Thinking (Philosophy)

## Δ: Teach Physics as Physics Is Done

- Solve + Compute in mind
- Why: how do research
- Not otherwise solvable
- Higher realism, complexity, precision
- Ed: PH + CS + Math in problem-solving context (cut classes)
- Using C teach P $\neq$ CP
- "OK for pedagogy" $\neq$ life

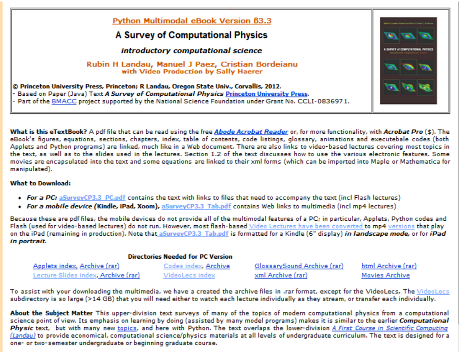



- Why we teach *Math Methods of Physics*?
- CS too important leave to CS to teach

# Available Resources (Free and Not): HTML Open Now!

## eText Book (*go there now!*)

http://www.science.oregonstate.edu/eBookWorking/



Python Multimodal eBook Version ß3.3

**A Survey of Computational Physics**

***introductory computational science***

**Rubin H Landau, Manuel J Paez, Cristian Bordeianu**
**with Video Production by Sally Haerer**

© Princeton University Press, Princeton; R Landau, Oregon State Univ., Corvallis, 2012.
- Based on Paper (Java) Text ***A Survey of Computational Physics*** Princeton University Press.
- Part of the BMACC project supported by the National Science Foundation under Grant No. CCLI-0836971.

**What is this eTextBook?** A pdf file that can be read using the free ***Abode Acrobat Reader*** or, for more functionality, with ***Acrobat Pro*** ($). The eBook's figures, equations, sections, chapters, index, table of contents, code listings, glossary, animations and executebale codes (both Applets and Python programs) are linked, much like in a Web document. There are also links to video-based lectures covering most topics in the text, as well as to the slides used in the lectures. Section 1.2 of the text discusses how to use the various electronic features. Some movies are encapsulated into the text and some equations are linked to their xml forms (which can be imported into Maple or Mathematica for manipulated).

**What to Download:**

- ***For a PC:*** aSurveyCP3.3_PC.pdf contains the text with links to files that need to accompany the text (incl Flash lectures)
- ***For a mobile device* (Kindle, iPad, Xoom),** aSurveyCP3.3_Tab.pdf contains Web links to multimedia (incl mp4 lectures)

Because these are pdf files, the mobile devices do not provide all of the multimodal features of a PC; in particular, Applets, Python codes and Flash (used for video-based lectures) do not run. However, most flash-based Video Lectures have been converted to mp4 versions that play on the iPad (remaining in production). Note that aSurveyCP3.3_Tab.pdf is formatted for a Kindle (6" display) ***in landscape mode,*** or for ***iPad in portrait***.

**Directories Needed for PC Version**

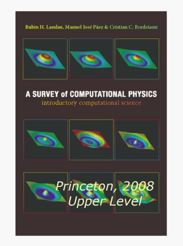
| | | | |
|---|---|---|---|
| Applets index, Archive (rar) | Codes index, Archive | GlossarySound Archive (rar) | html Archive (rar) |
| Lecture Slides index, Archive (rar) | VideoLecs index | xml Archive (rar) | Movies Archive |

To assist with your downloading the multimedia, we have a created the archive files in .rar format, except for the VideoLecs. The VideoLecs subdirectory is so large (>14 GB) that you will need either to watch each lecture individually as they stream, or transfer each individually.

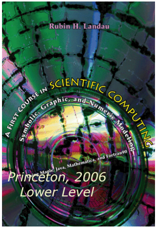
**About the Subject Matter** This upper-division text surveys of many of the topics of modern computational physics from a computational science point of view. Its emphasis on learning by doing (assisted by many model programs) makes it is similar to the earlier ***Computational Physic*** text, but with many new topics, and here with Python. The text overlaps the lower-division *A First Course in Scientific Computing (Landau)* to provide economical, computational science/physics materials at all levels of undergraduate curriculum. The text is designed for a one- or two-semester undergraduate or beginning graduate course.

| | | |
|---|---|---|
| PDF Book | PC/Mobile Size | Local/Web Media |
| Applets | Movies | Codes |
| Flash/mp4 Lectures | Lecture Slides | Sounds |

## Paper Texts ($$)

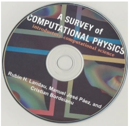



Princeton, 2008
Upper Level




Princeton, 2006
Lower Level

## Survey's CD

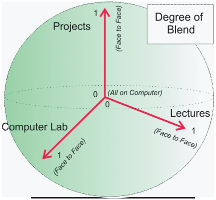



| | |
|---|---|
| Animations | Applet Libe |
| C Codes | Fortran77/95 |
| Java Codes | MPI codes |
| OpenDX | |

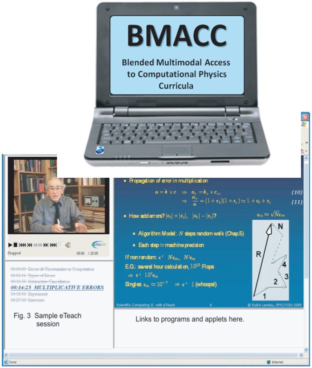
# Free BMACC eTextBook (Python), Abstract, Feedback

www.science.oregonstate.edu/eBookWorking/ → ComPADRE/Princeton

*The medium is the medium, the message is the message.*
*Different learners learn differently; compute to learn computing!*




- Web: as you like it
- Digital book (Linked pdf)
    - Live/search eqtns, figs
    - Present Technology



# How Incorporate Computation in Physics Courses?

## Philosophy (Words of Wisdom)

- Realistic/Classical Greek, correct math $\rightarrow$ CS
  Problem solving from basics now + computation
- Particular Python programs, to follow, book (Java, C, Fort)
- Coordinated Python exercises to build skills,
  to follow, local politics
- Survey of 3-D visualization techniques/vectors
  Very important, 10-20 $\rightarrow$ 50%
- Best tools common computation tasks
  Best religion? Compiled samples, Mathematica-like

# Skills Expected of Graduates

## AAPT Proposal

- Plot functions and data
- Numerical integration, diff
- Visualization complex data
- Limits of algorithms
- Programming*, compiled language
- Several operating systems
- ODEs, PDEs
- Matrix operations
- Fourier transforms, FFT
- Statistics, data fitting
- Computational thinking
- Symbolic programming
- LaTeX

Not programming experts, professionals (CS)!

# Our Scientific Python KISS

## Don't Go Whole 11 Yards (5835 packages)

- Doing science, not religion
- $\leq$ PhD CSE research
- P = Scripting Language
- No scripting
- No OS interactions (hard)
- No string manipulations
- No list manipulations
- Leave to CS

## What To Use (Python 2.6) (CiSE M/A 2011, M/J 2007)

- NumPy: ND arrays
- matplotlib: 2D/3D $\simeq$ Matlab
- Visual = Vpython: graphics, Vidle; 3D shapes
- SciPy: wrappers to other libes (LAPACK, FFT)
- (Sage > Mathematica): Symbols + py packages
- (Symbolic): Sympx, mpmath, Sage

# Organize Courses to Include



## Tools (Win Noble Prizes)

- **Basic Numeric Tools** (MathMeth)
- Integration $\Rightarrow$ Guassian
- Differentiation
- Floating Point Math [Errors]
- Search Techniques
- Linear Algebra [Libes]
- **ODEs Solutions (CM) $\Rightarrow$ rk4**
- Golf Balls
- Planets, 3-B Orbits*
- Classical Chaotic Scattering

- **Tools for Analysis**
- Visualization (Math, Orient)
- LaTeX* (Math, Orient)
- Fourier, DFT, FFT* (Lab)
- Wavelets* (MathMthds)
- Principal Components* (Math)
- Sonifications* (Lab)

# Organize Courses to Include — Applications

## Application

- **Explore Nonlinear Dynamics**
- Bifurcations, phase space (CM)
- Double & Chaotic Pendula (CM)
- Fractals, Stat Growth (StatMech)
- Integral Equations* (QM, CM)

- **Monte Carlo, Stochastic**
- Spontaneous Decay (QM)
- Random Walk (Thermo)
- Thermal Simulations (StatMech)
- Molecular Dynamics* ($\neq$ MC)
- Feynman Path Integrals* (QM)

- **PDEs (relax, t step, split t)**
- Laplace/Possion [space] (EM)
- Heat [x-t diffusion] (Thermo)
- Realistic Strings [waves] (CM)
- Quantum Wave Packets (QM)

- **Fluid Dynamics**
- Fluid Flow (> freshman)
- Shock Waves (CM)
- Solitons (CM)

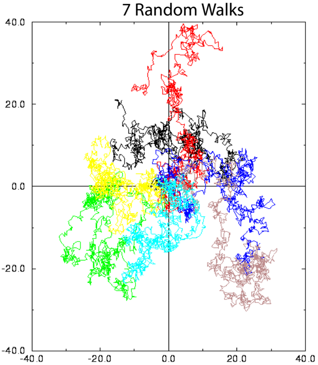
# Visualization Examples from Text

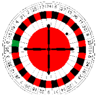
## Codes/PythonCodes/Visualizations

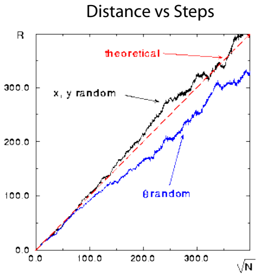
| | |
|---|---|
| *EasyMatPlot.py* | Import, create numpy x & y arrays |
| *GradesMat.py* | Two curves, one plots |
| *MatPlot2figs.py* | Two plots |
| *SurfacePlot.py* | Rotatable surface plot |
| *HarmosAnimate.py* | 2-D animation |
| *3Danimate.py* | 3-D animation |
| *3Dshapes.py* | Rotatable shapes |

# Monte-Carlo Random Walks

*Walk3D.py*






- No virgins: Should be "seen" at least once in college
- Stochastic processes & math: very interesting
- Random, probability, experimental statistics $\leq$ taught

# Monte-Carlo Decay Simulation

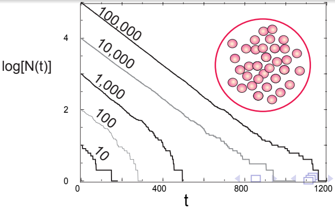
## Analytic $= e^{-t/\tau} \simeq$ Simulation = Closer Nature

$$\mathcal{P} = \frac{\Delta N(t)/N(t)}{\Delta t} = -\lambda \quad \text{(Law of Nature)} \tag{1}$$

$$\Rightarrow \quad \frac{\Delta N(t)}{\Delta t} = -\lambda N(t) \quad \text{(Real Physics)} \tag{2}$$

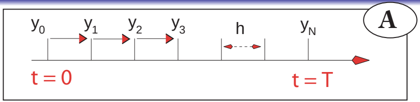
$$\not\Rightarrow \quad \frac{dN(t)}{dt} = \frac{dN}{dt}(0) e^{-\lambda t} \quad \text{(Approximate Physics)} \tag{3}$$

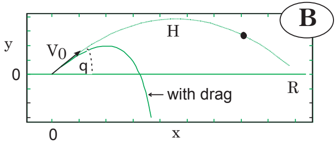
Links: Hear Geiger, *Hear Data* Applet, *DecaySound.py*

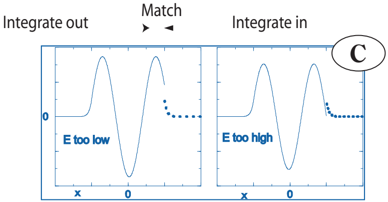


# Proper ODE Solve: Golf Balls & QM Binding

*ProjectileAir.py (V-V)* *rk4.py* *QuantumEigen.py*




## Can Easily Solve Any $V(r) \sim$ Same Technique ($r^{-3}$, $r^{-2.001}$)

- A, B: Euler's rule $\mathcal{O}(h^2)$: $y_{n+1} \simeq y_n + hf(t_n, y_n)$
- Euler not good for life
- rk4 $\mathcal{O}(h^4)$ cancels errors
- Velocity-Verlet in MD
- Bound $\stackrel{\text{def}}{=}$ confined $\Rightarrow$ B.C.
- C: Low $\mathcal{O}$: OK pedagogy?
- Low $\mathcal{O}$: Error Accumulates
- Error control in good science
- Math: both $e^{-\kappa r}$, $e^{+\kappa r}$

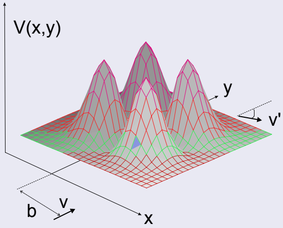
# CM Chaotic Scattering; 3-Bodies; Just Write ODE

## > Bound Orbits; Discovery, Creative, Stimulating, Personal

CM Scatt Applet 3 Body Applet

$$\mathbf{F} = m\mathbf{a}$$

$$\mathbf{F} = -\nabla\left(x^2y^2e^{-x^2-y^2}\right)$$




$$m\frac{d^2x}{dt^2} = 2y^2x(1-x^2)e^{-(x^2+y^2)}$$

$$m\frac{d^2y}{dt^2} = 2x^2y(1-y^2)e^{-(x^2+y^2)}$$

$$d\mathbf{y}/dt = f(t,\mathbf{y})$$

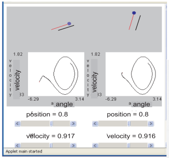
$$\mathbf{y} = [y^0, y^1, y^2, y^3]$$

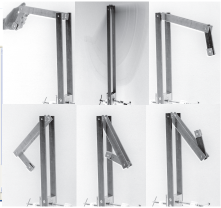
- 2nd order ODE → 2 × 1st order ODEs
- 2 Simultaneous → 4 × 1st order ODEs (↑ Dimension)

# Chaotic & Double Pendula

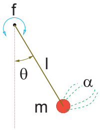
Applet

See Applets



DoublePend.mpg



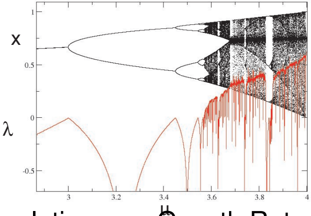

$$-mgl\sin\theta - \beta\frac{d\theta}{dt} + \tau_0\cos\omega t = I\frac{d^2\theta}{dt^2} \quad \text{(Single)} \tag{1}$$

$$\begin{aligned}(m_1+m_2)l_1\ddot{\theta}_1 + m_2 l_2\ddot{\theta}_2\cos(\theta_1-\theta_2) \\ +m_2 l_2\dot{\theta}_2^{\,2}\sin(\theta_1-\theta_2) + g(m_1+m_2)\sin\theta_1 = 0,\end{aligned} \tag{2}$$

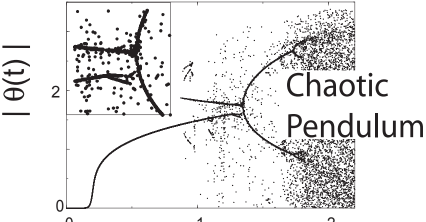
$$m_2 l_2\ddot{\theta}_2 + m_2 l_1\ddot{\theta}_1\cos(\theta_1-\theta_2) - m_2 l_1\dot{\theta}_1^{\,2}\sin(\theta_1-\theta_2) + mg\sin\theta_2 = 0. \tag{3}$$

# Bifurcations (NonLinear Analysis)

*Bugs.py* *LyapLog.py*



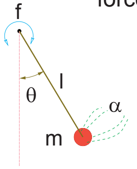

Logistic Map




Population x *vs* Growth Rate $\mu$

$$x_{i+1} = \mu x_i(1 - x_i) \qquad \text{Logistic}$$

$$x_i \sim x_{i-1} e^{\lambda\, i} \qquad \text{Lyapunov}$$




Hear FM Bifurcations

⇒ Universal

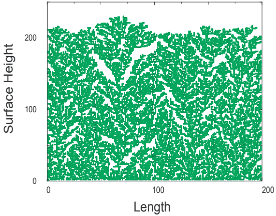
# Fractal Growth & Stat Physics

*Column.py* *Fern3D.py*

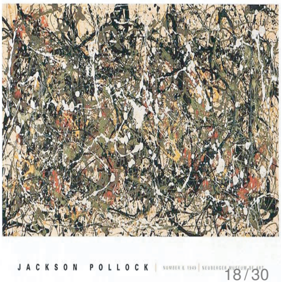
$$\text{Column Height} = h_r = \begin{cases} h_r + 1, & \text{if } h_r \geq h_{r-1} \quad h_r > h_{r+1} \\ \max[h_{r-1}, h_{r+1}] & \text{if } h_r < h_{r-1} \quad h_r < h_{r+1} \end{cases}$$

FractalGrowth.mpg

See Animation
Ballistic Depostion



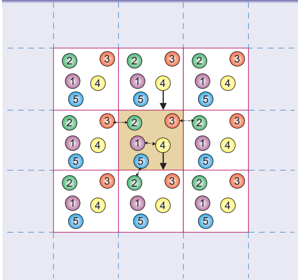

Jackson Pollock # 8

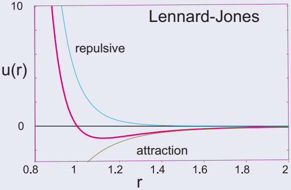


# Molecular Dynamics

Applet *MD1D.py* *MD2D.py*

## Straightforward, Obvious, Ridiculously Effective

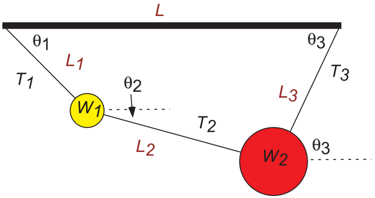





- Chem 101: walls $\Rightarrow PV = nRT$
- MD: atom-atom interactions
- QM: determines $V_{eff}(r)$
- Then $F_i = ma_i$
- HS physics problem from hell ($10^8$ particles $\neq 10^{23}$)
- Computers, boxes = finite
- Deterministic $\neq$ random, $\sim\ kT$

# HS Problem ⇒ HPC, Lin Alg (Parallel = Action)

*NewtonNDanimate.py*




- $T_i,\ \theta_i = ?$
- 9 nonlinear equations

$$
\begin{aligned}
L_1 \cos\theta_1 + L_2 \cos\theta_2 + L_3 \cos\theta_3 &= L, \\
L_1 \sin\theta_1 + L_2 \sin\theta_2 - L_3 \sin\theta_3 &= 0, \\
\sin^2\theta_1 + \cos^2\theta_1 &= 1 \\
\sin^2\theta_2 + \cos^2\theta_2 &= 1 \\
\sin^2\theta_3 + \cos^2\theta_3 &= 1
\end{aligned}
$$

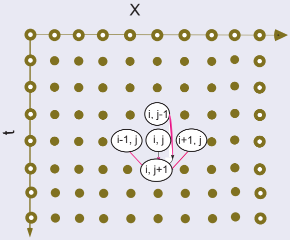
- Newton-Raphson search
- Use matrix libes

$$
\begin{aligned}
T_1 \sin\theta_1 - T_2 \sin\theta_2 - W_1 &= 0 \\
T_1 \cos\theta_1 - T_2 \cos\theta_2 &= 0 \\
T_2 \sin\theta_2 + T_3 \sin\theta_3 - W_2 &= 0 \\
T_2 \cos\theta_2 - T_3 \cos\theta_3 &= 0
\end{aligned}
$$

# Realistic Waves: Catenary + Friction *EqStringAnimate.py*

## PDE with Time Stepping (1 step at a time)

CatFrictionAnimate




$$c^2 \frac{\partial^2 y}{\partial x^2} - \frac{2\kappa}{\rho}\frac{\partial y}{\partial t} = \frac{\partial^2 y}{\partial t^2} \qquad \text{(with Friction)} \tag{1}$$

$$\frac{\partial T(x)}{\partial x}\frac{\partial y(x,t)}{\partial x} + T(x)\frac{\partial^2 y(x,t)}{\partial x^2} = \rho(x)\frac{\partial^2 y(x,t)}{\partial t^2} \qquad (\text{Variable } \rho \text{ \& } T) \tag{2}$$

# Ignored Time-Dependent Schrödinger Equation Wave Packets (Split Time Step PDE)

*HarmosAnimate.py*

$$i\frac{\partial\psi(x,t)}{\partial t} = -\frac{1}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\psi(x,t)$$

- Original SE; too hard
- Time stepping Applet
- 2 interacting particles $(x_1, x_2)$
- HTML Packet-Packet (AJP 2000)

click for 2slits

click for WavePacket$^2$

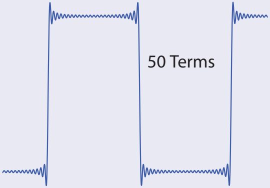
# Laplace, Poisson, Maxwell Eqns Get Realistic

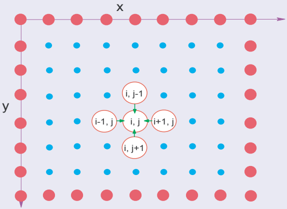
## Math Formulas/Series ≠ Algorithms *LaplaceLine.py*

$$\text{Poisson:} \quad \nabla^2 U(\mathbf{x}) = -4\pi\rho(\mathbf{x}) \tag{1}$$

$$\text{Soltn:} \quad U_{i,j} \simeq \frac{1}{4}\left[U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1}\right] + \pi\rho_{i,j}\Delta^2 \tag{2}$$

- Finite difference solution via relaxation
- E.G.: Gibb's Overshoot; also $j_l$ recurrence

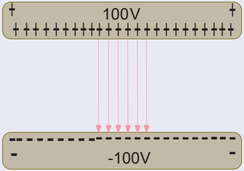


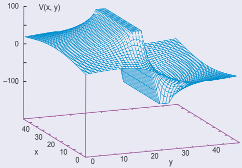



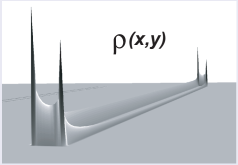
# Laplace, Poisson as Rubber Sheet Over BCs

## Charge on Realistic Capacitor? *LaplaceLine.py*

click for laplace




1. Solve + BC: $\nabla^2 U(\mathbf{x}) = 0$, 2. $\Rightarrow \rho(\mathbf{x}) = -\nabla^2 U(\mathbf{x})/4\pi$

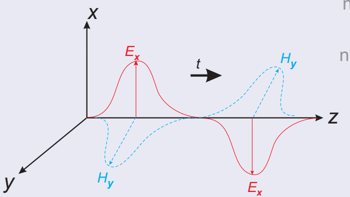
# Maxwell: Finite Difference Time Domain

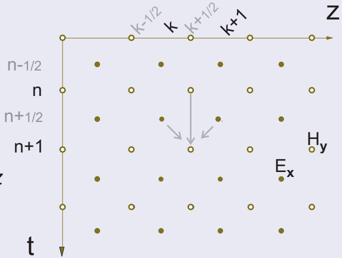
*FDTD.py*

## Split Time and Space Steps

$$\tilde{E}_x^{k,n+1/2} = \tilde{E}_x^{k,n-1/2} + \beta\left(H_y^{k-1/2,n} - H_y^{k+1/2,n}\right) \quad (1)$$

$$H_y^{k+1/2,n+1} = H_y^{k+1/2,n} + \beta\left(\tilde{E}_x^{k,n+1/2} - \tilde{E}_x^{k+1,n+1/2}\right) \quad (2)$$




- $z$ Propagation
- Coupled $E_x$, $H_y$ drive other
- $E_x$ filled, t/2; $H_y$ open, x/2
- Easy: Excell, *FDTD.py*

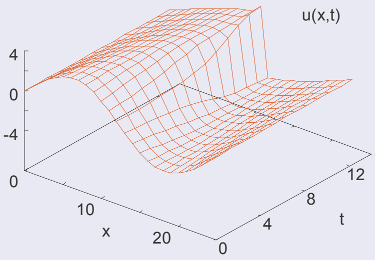
# Fluids Gone? Shock Waves

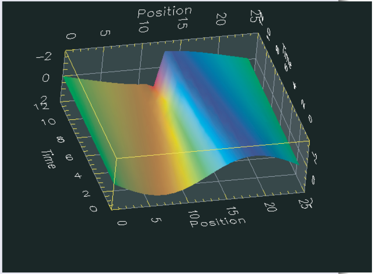
AdvecLax Applet

## Singular Math ⇒ Better Algorithm (Lax); Singular Nature?

$$\frac{\partial \rho(x,t)}{\partial t} + c\frac{\partial \rho(x,t)}{\partial x} = 0 \qquad \text{(Advection/Continuity)} \qquad (1)$$

$$\frac{\partial \rho(x,t)}{\partial t} + \epsilon\,\rho(x,t)\frac{\partial \rho(x,t)}{\partial x} = 0 \qquad \text{(Burgers' Eqn} \Rightarrow \text{Shock)} \qquad (2)$$

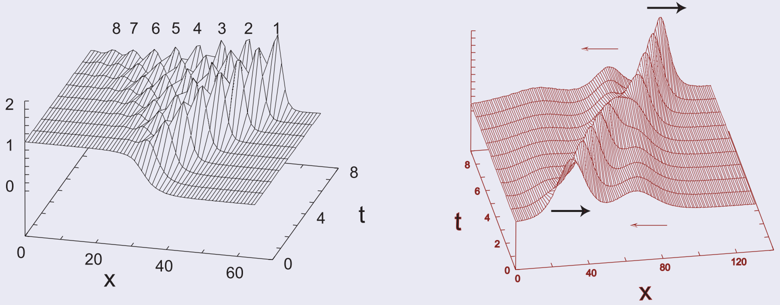


# (Shock Waves) Dispersion, Solitons

*SolitonAnimate.py*

## KdeV Equation:Dispersion Balances Shock



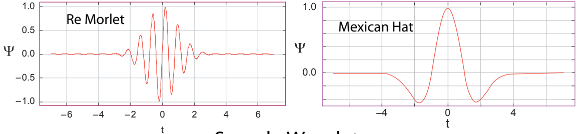

$$\frac{\partial \rho(x,t)}{\partial t} + \epsilon\, \rho(x,t) \frac{\partial \rho(x,t)}{\partial x} = 0 \qquad \text{Burgers' Equation} \qquad (1)$$

$$\frac{\partial \rho(x,t)}{\partial t} + \epsilon\, \rho(x,t) \frac{\partial \rho}{\partial x} + \mu \frac{\partial^3 \rho(x,t)}{\partial x^3} = 0 \qquad \text{Shock + Dispersion} \qquad (2)$$

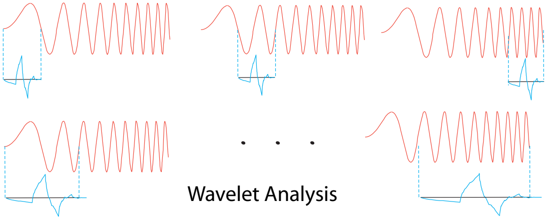
Applet

# Tools 2: Wavelet Analyses (jpeg)

*CWT.py* *FFT.py*




Sample Wavelets

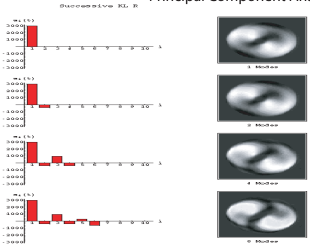


Wavelet Analysis

$$Y(s,\tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} dt\, \Psi^* \left( \frac{t-\tau}{s} \right) y(t) \quad (Wavelet\ Transform)$$

# Tools 3: (Wavelet &) Principal Component Analyses

Principal Component Analysis




(See Animation)

- Simplest eigenvector-based multivariate analysis
- Orthogonal linear transformation
- Multivariate data $\rightarrow$ uncorrelated principal components
- Matrix libes and powerful math (Animation)
- $\equiv$ singular value decomposition of data matrix

# Take Home Lessons: Need for Change Physics Ed

- Physics now done with computation
- Physics now done with other sciences
- Physics Ed now done with $\simeq$ 50-100 year old stuff
- American students now seriously behind
- Agree: bad math means wrong science?
- So bad computation means wrong science
- Computation too important to leave to CS
- www.science.oregonstate.edu˜rubin