Teaching Physics with Python

Louise Dash
louise.dash@ucl.ac.uk
@louisedash

Department of Physics and Astronomy, UCL

2 June 2015

Outline

Outline

Outline

Outline

Outline

Outline

1. Context
2. Why Python?
3. Challenges
4. The courses
5. More challenges (and successes)
6. The future

Outline

Outline

1. Context
2. Why Python?
3. Challenges
4. The courses
5. More challenges (and successes)
6. The future
7. Conclusions

Context

- Nearing the end of a total reorganization of our core computing courses

Context

- Nearing the end of a total reorganization of our core computing courses
- Previous courses were an “evolved over time” mishmash of Matlab, Excel, Mathematica, IDL

Context

- Nearing the end of a total reorganization of our core computing courses
- Previous courses were an "evolved over time" mishmash of Matlab, Excel, Mathematica, IDL
- 2013-2014: Introduction of brand new Python course for first year "Practical physics 1A" module

Context

- Nearing the end of a total reorganization of our core computing courses
- Previous courses were an “evolved over time” mishmash of Matlab, Excel, Mathematica, IDL
- 2013-2014: Introduction of brand new Python course for first year “Practical physics 1A” module
- 2014-2015: Introduction of follow-on Python course for second year “Practical physics 2B” module

Context

- Nearing the end of a total reorganization of our core computing courses
- Previous courses were an "evolved over time" mishmash of Matlab, Excel, Mathematica, IDL
- 2013-2014: Introduction of brand new Python course for first year "Practical physics 1A" module
- 2014-2015: Introduction of follow-on Python course for second year "Practical physics 2B" module
- Now integrating computing into other parts of the curriculum

Motivation

- Historically, computation has been a “poor relation” in the physics curriculum

Motivation

- Historically, computation has been a "poor relation" in the physics curriculum
- Not seen as particularly accessible to undergraduate students

Motivation

- Historically, computation has been a "poor relation" in the physics curriculum
- Not seen as particularly accessible to undergraduate students
- Hence of limited usefulness to students!

Motivation

- Historically, computation has been a “poor relation” in the physics curriculum
- Not seen as particularly accessible to undergraduate students
- Hence of limited usefulness to students!

However...

Modern languages like Python make computation much more accessible to the non-specialist.

Outline

Why Python?

- It's free, and open source

Why Python?

- It's free, and open source
- "Easy" to learn, user-friendly, gateway language

Why Python?

- It's free, and open source
- "Easy" to learn, user-friendly, gateway language
- Easy to install

Why Python?

- It's free, and open source
- "Easy" to learn, user-friendly, gateway language
- Easy to install
- Marketable, transferable skill beyond physics

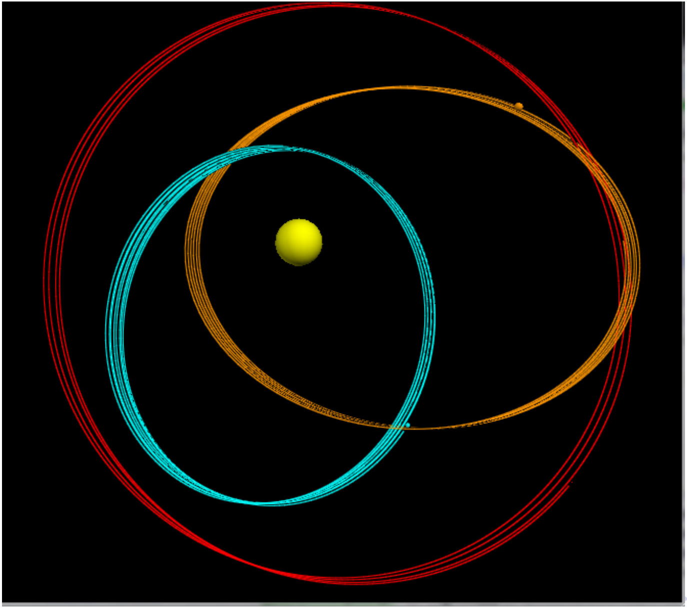
Why Python?

- It's free, and open source
- "Easy" to learn, user-friendly, gateway language
- Easy to install
- Marketable, transferable skill beyond physics
- See http://lorenabarba.com/blog/why-i-push-for-python/

Why Python?

- It's free, and open source
- "Easy" to learn, user-friendly, gateway language
- Easy to install
- Marketable, transferable skill beyond physics
- See http://lorenabarba.com/blog/why-i-push-for-python/
- Easy to create 3d animations and models with "visual" Vpython module

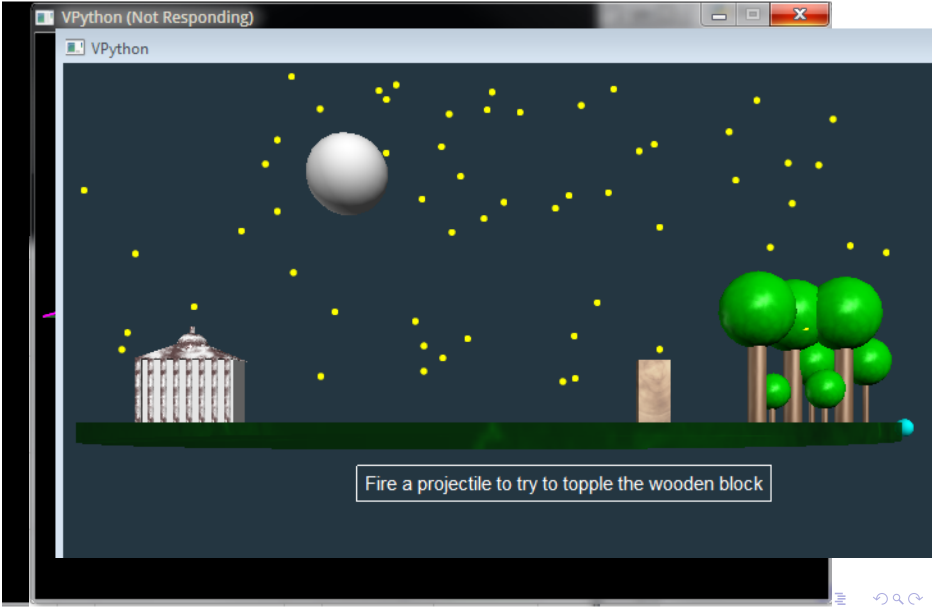
Vpython



Vpython


VPython (Not Responding)
Target at
x = 5.75 m
You hit the target, but failed to topple it!

Vpython


VPython (Not Responding)
VPython
Fire a projectile to try to topple the wooden block

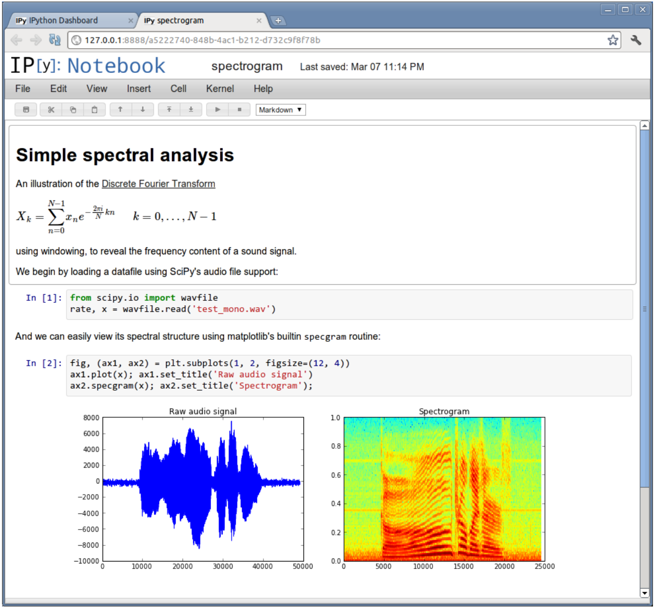
Why Python?

- It's free, and open source
- "Easy" to learn, user-friendly, gateway language
- Easy to install
- Marketable, transferable skill beyond physics
- See http://lorenabarba.com/blog/why-i-push-for-python/
- Easy to create 3d animations and models with "visual" Vpython module
- IPython Notebook

Why Python?

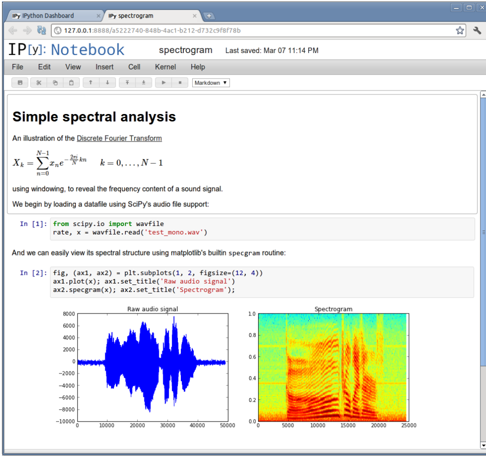
- It's free, and open source
- "Easy" to learn, user-friendly, gateway language
- Easy to install
- Marketable, transferable skill beyond physics
- See http://lorenabarba.com/blog/why-i-push-for-python/
- Easy to create 3d animations and models with "visual" Vpython module
- IPython Notebook

The IPython Notebook



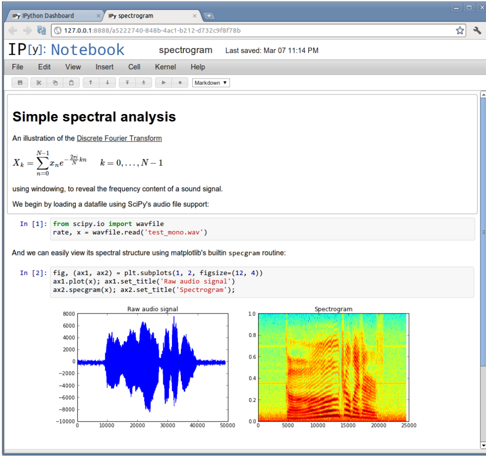
The IPython Notebook

- Text commentary encourages students to think clearly about what they are doing and *why*...
- ...from a coding and from a physics point of view.



The IPython Notebook

- Text commentary encourages students to think clearly about what they are doing and *why*...
- ...from a coding and from a physics point of view.
- Also ideal for use as an interactive session script.



The IPython Notebook

- Text commentary encour
students to think clearly
what they are doing and
- …from a coding and fror
physics point of view.
- Also ideal for use as an
interactive session script.

In [5]:
```
guess = [75,10,3.5,18] # list of initial guess parameters

type(guess) # what type of object does the variable "guess" represent?
```
Out[5]:
```
list
```

Now we can retry the fit:

In [6]:
```
popt,pcov = curve_fit(gaussian,xdata,ydata,p0=guess)
print "popt :\n", popt
print "pcov :\n", pcov
```
```
popt :
[ 72.50930905    3.01525268    3.85742572   13.40680376]
pcov :
[[  2.02507205e-03  -3.77758184e-10   4.14321753e-12   1.66043777e-09]
 [ -3.77758184e-10   2.22561784e-03  -6.30519920e-04  -4.05620885e-03]
 [  4.14321753e-12  -6.30519920e-04   1.98236679e-03  -1.40174499e-03]
 [  1.66043777e-09  -4.05620885e-03  -1.40174499e-03   3.10175058e-02]]
```

This has worked (or it should have done)! We can use the information from the matrix of covariance to calculate the error on each parameter, just as we did in the previous session for the polynomial coefficients. Remember, the errors on the parameters are given by the *square roots* of the diagonal elements of the matrix of covariance.

In the cell below, you should:

- calculate the errors on the parameters
- output each parameter with its error and an appropriate text string
- plot the original data and the fitted line on a single, appropriately labelled graph

In [7]:
```
### STUDENT COMPLETED CELL ###
```

If you've done this correctly, you should obtain a good fit to the data.

Outline

Challenges

- No prerequisite for computing!

Challenges

- No prerequisite for computing!
- Wide range of abilities, backgrounds and experiences

Challenges

- No prerequisite for computing!
- Wide range of abilities, backgrounds and experiences
  - 70% have no prior programming experience

Challenges

- No prerequisite for computing!
- Wide range of abilities, backgrounds and experiences
  - 70% have no prior programming experience
- Similarly, wide range of confidence with computers

Challenges

- No prerequisite for computing!
- Wide range of abilities, backgrounds and experiences
  - 70% have no prior programming experience
- Similarly, wide range of confidence with computers
  - 65% say they are confident or very confident with computing

Challenges

- No prerequisite for computing!
- Wide range of abilities, backgrounds and experiences
  - 70% have no prior programming experience
- Similarly, wide range of confidence with computers
  - 65% say they are confident or very confident with computing
- No obvious correlation between achievement and confidence

Challenges

- No prerequisite for computing!
- Wide range of abilities, backgrounds and experiences
  - 70% have no prior programming experience
- Similarly, wide range of confidence with computers
  - 65% say they are confident or very confident with computing
- No obvious correlation between achievement and confidence
- Large gender effect in confidence (no gap in outcomes)

Challenges

- No prerequisite for computing!
- Wide range of abilities, backgrounds and experiences
  - 70% have no prior programming experience
- Similarly, wide range of confidence with computers
  - 65% say they are confident or very confident with computing
- No obvious correlation between achievement and confidence
- Large gender effect in confidence (no gap in outcomes)
- At the beginning they don't know much physics or maths either...

Challenges

- No prerequisite for computing!
- Wide range of abilities, backgrounds and experiences
  - 70% have no prior programming experience
- Similarly, wide range of confidence with computers
  - 65% say they are confident or very confident with computing
- No obvious correlation between achievement and confidence
- Large gender effect in confidence (no gap in outcomes)
- At the beginning they don't know much physics or maths either...
- Computing notoriously hard to teach, and to assess (see work of Mark Guzdial, https://computinged.wordpress.com/

Outline

What we did: First year course

- Runs in first term of the first year

What we did: First year course

- Runs in first term of the first year
- Emphasis on basic skill requirements:

What we did: First year course

- Runs in first term of the first year
- Emphasis on basic skill requirements:
  - Present and analyse lab data:

What we did: First year course

- Runs in first term of the first year
- Emphasis on basic skill requirements:
    - Present and analyse lab data:
    - Plotting, straight line fit from first principles

What we did: First year course

- Runs in first term of the first year
- Emphasis on basic skill requirements:
  - Present and analyse lab data:
  - Plotting, straight line fit from first principles
  - Computational models using Vpython...

What we did: First year course

- Runs in first term of the first year
- Emphasis on basic skill requirements:
  - Present and analyse lab data:
  - Plotting, straight line fit from first principles
  - Computational models using Vpython...
  - ...mostly based on classical mechanics

What we did: Second year course

- Runs in second term of the second year

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...
- More advanced data analysis

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...
- More advanced data analysis
    - Non-linear fitting, χ^2 analysis, analysis of residuals

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...
- More advanced data analysis
    - Non-linear fitting, χ^2 analysis, analysis of residuals
- Computational physics

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...
- More advanced data analysis
    - Non-linear fitting, χ^2 analysis, analysis of residuals
- Computational physics
    - Matrix calculations (but not quantum mechanics...)

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...
- More advanced data analysis
    - Non-linear fitting, χ^2 analysis, analysis of residuals
- Computational physics
    - Matrix calculations (but not quantum mechanics...)
    - Runge Kutta methods

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...
- More advanced data analysis
  - Non-linear fitting, χ^2 analysis, analysis of residuals
- Computational physics
  - Matrix calculations (but not quantum mechanics...)
  - Runge Kutta methods
  - Fast Fourier transforms

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...
- More advanced data analysis
  - Non-linear fitting, χ^2 analysis, analysis of residuals
- Computational physics
  - Matrix calculations (but not quantum mechanics...)
  - Runge Kutta methods
  - Fast Fourier transforms
- Make use of both inbuilt libraries as black boxes *and* coding from scratch.

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...
- More advanced data analysis
  - Non-linear fitting, χ^2 analysis, analysis of residuals
- Computational physics
  - Matrix calculations (but not quantum mechanics...)
  - Runge Kutta methods
  - Fast Fourier transforms
- Make use of both inbuilt libraries as black boxes *and* coding from scratch.
- Eg Fourier transforms: start with hand-coded discrete Fourier transform...

What we did: Second year course

- Runs in second term of the second year
- They've forgotten most of what they did in the first year...
- More advanced data analysis
  - Non-linear fitting, χ^2 analysis, analysis of residuals
- Computational physics
  - Matrix calculations (but not quantum mechanics...)
  - Runge Kutta methods
  - Fast Fourier transforms
- Make use of both inbuilt libraries as black boxes *and* coding from scratch.
- Eg Fourier transforms: start with hand-coded discrete Fourier transform...
- ...Compare and contrast with inbuilt (numpy) FFT libraries

Outline

More challenges!

- Computing *really* hard to assess objectively and consistently,

More challenges!

- Computing *really* hard to assess objectively and consistently,
  - especially with a team of markers...

More challenges!

- Computing *really* hard to assess objectively and consistently,
  - especially with a team of markers...
  - Moodle / Turnitin rubrics very useful but not perfect

More challenges!

- Computing *really* hard to assess objectively and consistently,
  - especially with a team of markers...
  - Moodle / Turnitin rubrics very useful but not perfect
- First year course in particular challenging

More challenges!

- Computing *really* hard to assess objectively and consistently,
  - especially with a team of markers...
  - Moodle / Turnitin rubrics very useful but not perfect
- First year course in particular challenging
  - How to stretch experienced students without freaking out the less confident?

More challenges!

- Computing *really* hard to assess objectively and consistently,
  - especially with a team of markers...
  - Moodle / Turnitin rubrics very useful but not perfect
- First year course in particular challenging
  - How to stretch experienced students without freaking out the less confident?
  - Wide availability of “self-teach” material a double-edged sword

More challenges!

- Computing *really* hard to assess objectively and consistently,
  - especially with a team of markers...
  - Moodle / Turnitin rubrics very useful but not perfect
- First year course in particular challenging
  - How to stretch experienced students without freaking out the less confident?
  - Wide availability of "self-teach" material a double-edged sword
  - Students considerably more anxious about marks than understanding the concepts

More challenges!

- Computing *really* hard to assess objectively and consistently,
  - especially with a team of markers...
  - Moodle / Turnitin rubrics very useful but not perfect
- First year course in particular challenging
  - How to stretch experienced students without freaking out the less confident?
  - Wide availability of "self-teach" material a double-edged sword
  - Students considerably more anxious about marks than understanding the concepts
- IPython Notebooks and Vpython don't work well together, frustratingly!

Successes

- Real progress seen by end of 2nd year

Successes

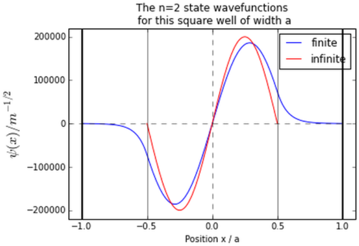
- Real progress seen by end of 2nd year
- Students demonstrate enhanced understanding of physics via computing

Successes

- Real progress seen by end of 2nd year
- Students demonstrate enhanced understanding of physics via computing

```
plt.xlim(-1.1,1.1)
plt.ylim(min(iwf)*1.1, max(iwf)*1.1)
plt.legend(loc='best')
plt.title("The n=2 state wavefunctions \n for this square well of width a")
```

[72]: <matplotlib.text.Text at 0x17002dd8>



Same observations as in the $n = 1$ state can be seen here, which include the sinusoidal wave form similar to the $n = 2$ 'infinite' wave inside the finite walls, and the decay to zero outside. Differences between the two above waves are due to their different energies and boundary conditions. These points are also exhibited in all higher energy states. Take $n = 3$ state for example, with which we'll, firstly, calculated the energy level E_3:

[73]:
```
# Define the initial conditions
E1 = Eisw[2]*0.3 # guess 1 of n=2 state energy (J)
E2 = Eisw[2]*0.6 # guess 2 of n=2 state energy (J)

# solve for the first and second guesses
psi1 = RungeKutta2d3(xpoints,E1)[N]
psi2 = RungeKutta2d3(xpoints,E2)[N]

### now for the secant method to converge on the right answer
```

Outline

Integrating computing within the curriculum

- Already using IPython notebooks within lecture courses as interactive lecture notes (David Bowler)

Integrating computing within the curriculum

- Already using IPython notebooks within lecture courses as interactive lecture notes (David Bowler)
- 1st year, term 2: Waves and optics (mostly plots)

Integrating computing within the curriculum

- Already using IPython notebooks within lecture courses as interactive lecture notes (David Bowler)
- 1st year, term 2: Waves and optics (mostly plots)
- 3rd year: Quantum mechanics - making use of matrix mechanics

Integrating computing within the curriculum

- Already using IPython notebooks within lecture courses as interactive lecture notes (David Bowler)
- 1st year, term 2: Waves and optics (mostly plots)
- 3rd year: Quantum mechanics - making use of matrix mechanics
- Some more challenges here:

Integrating computing within the curriculum

- Already using IPython notebooks within lecture courses as interactive lecture notes (David Bowler)
- 1st year, term 2: Waves and optics (mostly plots)
- 3rd year: Quantum mechanics - making use of matrix mechanics
- Some more challenges here:
  - Students on Natural Sciences program (and students from other departments) haven't done these computing courses

Integrating computing within the curriculum

- Already using IPython notebooks within lecture courses as interactive lecture notes (David Bowler)
- 1st year, term 2: Waves and optics (mostly plots)
- 3rd year: Quantum mechanics - making use of matrix mechanics
- Some more challenges here:
    - Students on Natural Sciences program (and students from other departments) haven't done these computing courses
    - Partial solution: run "catch-up" conversion workshop from Matlab to Python

The Future

- Plan to extend the use of IPython Notebooks within lecture courses

The Future

- Plan to extend the use of IPython Notebooks within lecture courses
- Early stages of planning new lecture course where the physics will be entirely assessed by IPython Notebook computation rather than written exam.

Outline

Conclusions

- Have rewritten core computing courses in years 1 and 2

Acknowledgments:

- Demonstrator teams on PHAS1240 and PHAS2441
- Dr David Bowler (lecturing with IPython Notebooks)

Conclusions

- Have rewritten core computing courses in years 1 and 2
- Extensive use of IPython notebooks enable enhanced understanding of physics

Acknowledgments:

- Demonstrator teams on PHAS1240 and PHAS2441
- Dr David Bowler (lecturing with IPython Notebooks)

Conclusions

- Have rewritten core computing courses in years 1 and 2
- Extensive use of IPython notebooks enable enhanced understanding of physics
- Extending this to fully embed computational physics within the curriculum

Acknowledgments:

- Demonstrator teams on PHAS1240 and PHAS2441
- Dr David Bowler (lecturing with IPython Notebooks)