Using Python and pdfLaTeX to Generate Customized Physics Problems

Bill Nettles
Geoffrey Poore

Union University

January 10, 2016

Multiple Exercises

- We routinely use exercises for emphasis and practice.
- Generate mental muscle memory, pattern recognition, collaboration, and teachable moments.
- How do we generate examples of favorite exercises with different starting values?
- How do we minimize time spent in generating solutions to multiple examples?
- Embed Python code into a LaTeX document

Python programming language

Python is an open-source, free programming language which is available for a variety of operating systems.

- Distributions, e.g., Anaconda, Python(xy), CPython, Active Python.
- Open-source $\rightarrow$ Many packages and libraries for scientific processing, symbolic processing, graphics processing, and web-based notebooks
- One interesting package in the PER community is `vPython`, developed by David Scherer and Bruce Sherwood and many others.
- `randassign` is a package we will address today.

UNION UNIVERSITY

LaTeX

An open-source software package for typesetting documents. Hundreds of packages are available for extending the ability of this package:

- special formats such as books, journals, calendars, dissertations
- graphics
- interfaces with programming languages
- interfacing Python via the PythonTeX package

UNION UNIVERSITY

The PythonTeX package

Provides a set of macros and environments for including Python code in a LaTeX document and allowing results to be typeset in the document.

```
\documentclass{beamer}
\usepackage{pythontex}
....
\begin{pycode}
print('Hello, AAPT. Welcome to Python\\TeX.')
\end{pycode} ....
```

Hello, AAPT. Welcome to PythonTeX.

UNION UNIVERSITY

The PythonTEX package

Calculations can be done either in a block environment or inline with macros. Values of variables can be typeset.

```
\pyc{from math import *}
\pyc{arad=round(25*pi/180,6)}
For $\theta$ = \ang{25}=\py{arad} radians,
$\cos\theta$ = $\py{round(cos(arad),5)}
and
$\sin\theta$ = $\py{round(sin(arad),4)}
```

For θ = 25°=0.436332 radians, cos θ =0.90631 and sin θ =0.4226

Motivation for PythonTEX

- Self-contained document with graphics (`matplotlib`)
- Typesetting symbolic calculations (`sympy` library)
- Easily changeable, reproducible scientific calculations
- Building other packages that require programming and file access (`nucleardata`)
- Randomized values within a standard formatting

An Aside on Using PythonTEX

- Using the PythonTEX package requires `pythontex.exe` to be called with the <*yourfile*>.`tex` file as the argument.
- `pythontex.exe` is automatically installed and correct path's are generated as part of the TEXLive distribution.
- LATEX document compiling sequence of mytest.tex:
  - `pdflatex mytest.tex`
  - `pythontex mytest.tex`
  - `pdflatex mytest.tex`
- Most editors for LATEX will allow users to create custom execution sequences like this or to designate shortcuts for each step of the sequence.

Using random numbers

By looping and using random number generation in Python, we can generate multiple exercises:

```
Find the $x$ and $y$ components
   of the following vectors:
\begin{pycode}
from numpy import random
for j in range(4):
  r1=random.uniform(3,11)
  theta=random.random_integers(0,360)
  thetarand=theta*pi/180
  x=r1*cos(thetarand)
  y=r1*sin(thetarand)
  print('$\\vec{r}$'+'={:3.2f} m at {:d}$^o$ \n'.format(r1,theta))
  print('\\hspace{.5cm}','x = {:3.2f} m'.format(x))
  print('\\hspace{.5cm}','y = {:3.2f} m \n'.format(y))
\end{pycode}
```

Using random numbers

Find the x and y components of the following vectors:

$\vec{r}$=3.93 m at 26^o

x = 3.53 m y = 1.72 m

$\vec{r}$=3.11 m at 223^o

x = -2.28 m y = -2.12 m

$\vec{r}$=10.32 m at 126^o

x = -6.07 m y = 8.35 m

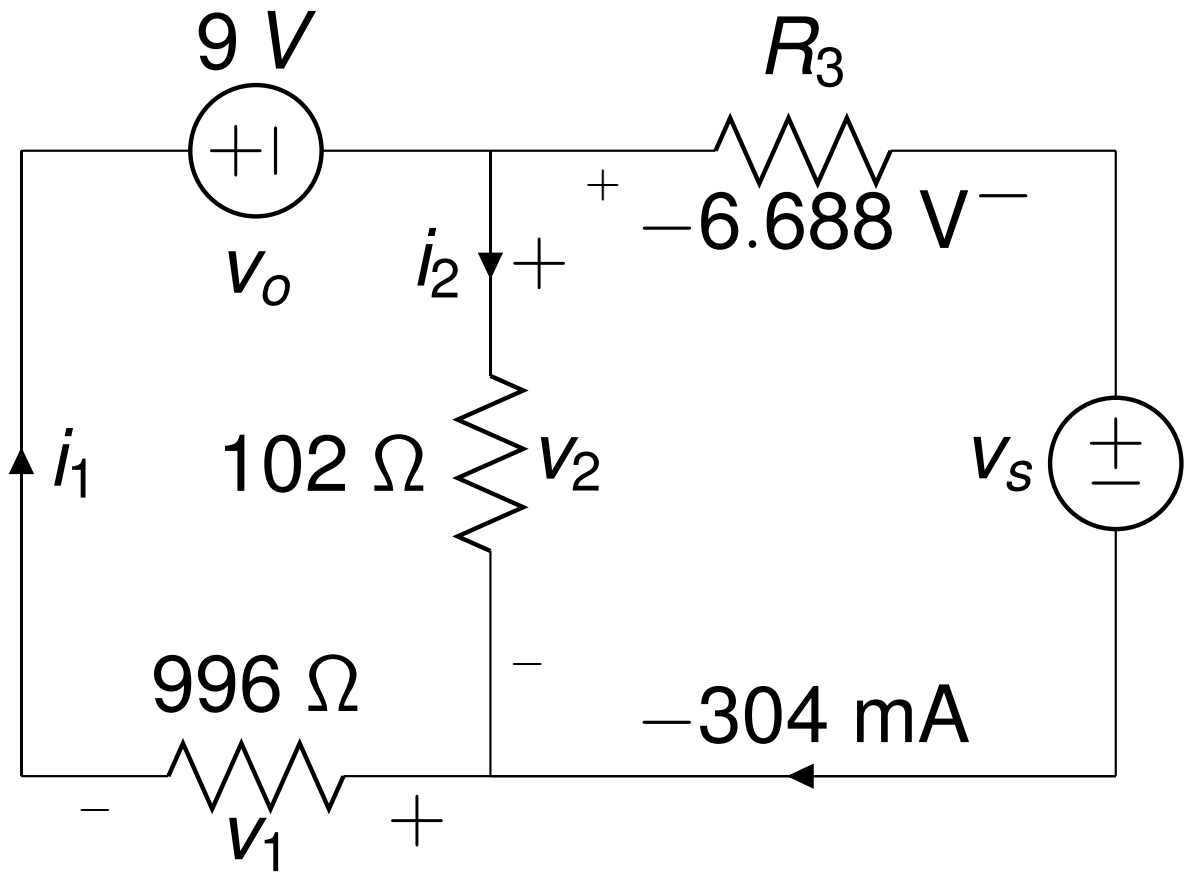
$\vec{r}$=6.54 m at 358^o

x = 6.54 m y = -0.23 m

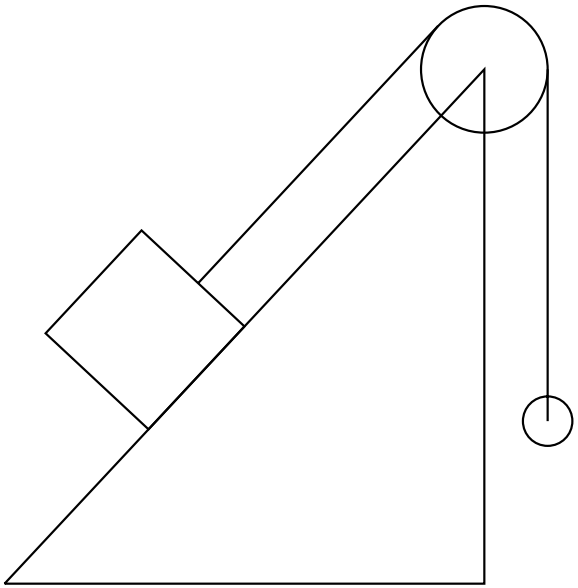
UNION UNIVERSITY

Graphics

Find all unknown currents, voltages, and the power consumed by every element.



Graphics



A box is
initially at rest on an inclined plane tilted at angle θ with respect to the horizontal. The box is attached by a low-mass, non-stretching rope to a freely-hanging mass. A uniform gravitational field $\vec{g}$ acts vertically downward. The pulley is low-mass and frictionless.
$\theta = 47°$
$m_{hanging}$=1.37 kg

m_{box}=3.62 kg
μ_s=0.381 μ_k = 0.325

UNION UNIVERSITY

The nucleardata Package

```
\begin{pycode}
z=random.randint(1,118)
alist=nuc.getIsotopes(str(z))#.split(",")
a=random.randint(int(alist[0]),int(alist[-1]))
randmass=nuc.getMass_u(str(z),a)
randsymb1=nuc.getSymbol(str(z))
randname=nuc.getName(str(z))
\end{pycode}
The random element is Z=\py{z},
which is \py{randname} with chemical symbol \py{randsymb1}.

The smallest isotope is \py{alist[0]}
and the largest is \py{alist[-1]}.

The mass of $^{\py{a}}$\py{randsymb1} is \py{randmass} u.
```

The random element is Z=88, which is Radium with chemical symbol Ra.
The smallest isotope is 202 and the largest is 234.
The mass of ^{206}Ra is 206.003827 u.

UNION UNIVERSITY

randassign

A Python program which automates generation of multiple randomized documents.

- Input a name from a file (`name.tex`)
- Runs the `pdflatex–pythontex–pdflatex` sequence on the LaTeX file
- Renames the output file and stores it in a subdirectory
- Reads a new name from a file (`students.txt`)

UNION UNIVERSITY

randassign package

A Python package which provides functions for storing and organizing answers or solutions in a separate file

```
\begin{pycode}
from numpy import random
from randassign import RandAssign
ra=RandAssign()
a=random.randint(1,100,2)
print(str(a[0])+ '+' + str(a[1])+'= _____ ')
ra.addsoln('Sum = '+str(a[0]+a[1]) )
\end{pycode}
```

82+20= ____ Sum = 102

UNION UNIVERSITY

Availability

- `pythontex` is part of the TeXLive and MikTeX distributions.
- `randassign` is available at PyPI via
  `pip install randassign`
  and direct download. Also from GitHub.

Future Libraries

We intend to develop a library of exercises. For our courses, we have already developed and used:

- E-field calculations in 2 dimensions with 2 randomly placed charges
- Two-loop DC circuits with random sources
- Box on an incline with random parameters
- Electrical potential due to multiple charges
- Drawing randomized ellipses for astronomy classes
- Randomizing telescope parameters
- Randomly selecting nuclides for halflife and binding energy exercises

UNION UNIVERSITY

Acknowledgements

- Union University, Jackson, TN
- Dr. Geoffrey Poore
- Python, TEX and LATEX developers
  - Comprehensive TEX Archive Network (CTAN)
  - Python.org
  - Software distributors like PyPI, GitHub, SourceForge