Department of Civil Engineering, Manipal University Jaipur

Interactive Lecture Materials Using Open-Source Technologies

Prabhas K. Yadav, Ph.D.
prabhask.yadav@jaipur.manipal.edu

Faculty Conclave 2017
18th July, 2017

Contents

Smart class is a place that encourages collaboration, engagement, and better learning aided by the technology.

Common features:
Student subgroups and many types of monitors

Education Technology: Refers to

1. Physical Equipments
2. Algorithm (codes)
3. Facilitating system between item 1 and item 2

History of Mechanical & Electronic Technology in Teaching

Year	Development
1920	Sidney Pressey's teaching machine
1940s	Overhead Projectors
1945	Hypertext
1950	Slide projectors
1960	Linked computer at University of Illinois
1970-80	Computer based learning
1990's	World wide web

A Requirement:
Interactive lecture contents for better learning

The Challenge:
The contents have to be dynamic (slides are mostly static) and be able to show "cause and effect"

Engineering Lecture Contents Normally Contains:

1. Principle (mostly text)
2. Geometry
3. Derivation and Formulation
4. Analysis (cause and effect)

For a better interactiveness, it is required that points 2–4 can be changed in real-time.

Clearly the only **text** based contents (e.g. slides) are not sufficient.

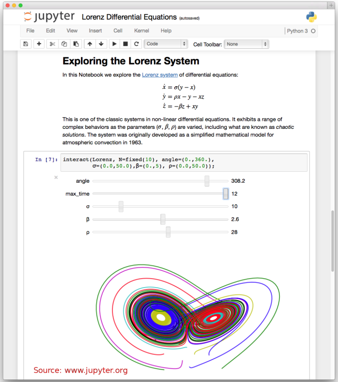
Contents Development is Software Issue, Do We:

1. have codes that can do symbolic manipulation.
2. have codes that present a notebook type interface.
3. have codes that can do instant numerical calculations
4. have codes that can present instant visualization.
5. have codes that are truly affordable and extensible

The answer is actually POSITIVE for all above points. These all exist as different packages. OPEN-SOURCE have intelligently put all of them together.

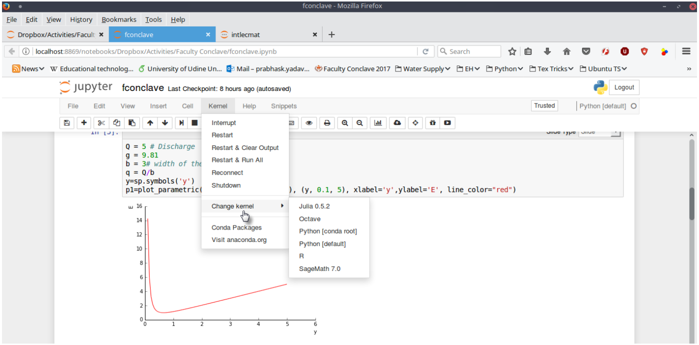
Introducing the JUPYTER Notebook

The Jupyter Notebook is an **open-source web application** that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. (source: www.jupyter.org)



Can *.ppt be replaced with *.ipynb?

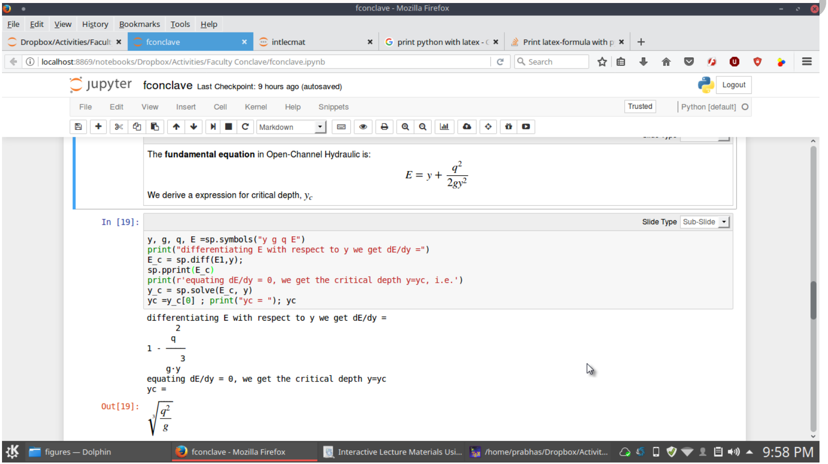
JUPYTER can execute several different codes in a same notebook. This is advantageous as it gives flexibility with the choice of code



The code is executed instantly, thus high interactiveness.

Demonstration of Interactive Contents
Using Jupyter

Next we use **SymPy**, a Symbolic Library for **Python** to derive an expression for critical depth y_c in Energy(E)-Depth(y) relationship.



Let us solve a **text-book** problem. We will use **SymPy**.

jupyter fconclave Last Checkpoint: 10 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Help Snippets

Trusted Python [default]

Code

A 2.5 m wide rectangular channel has a specific energy of 1.50 when carrying a discharge of 6.48 m^3/s. Calculate the alternate depths.

Solution:

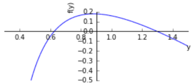
The fundamental equation is: $E = y + \frac{q^2}{2gy^2}$

In [3]:

Slide Type Slide

```
y, E, g, Q, b, q = sp.symbols("y E g Q b q")
eq1 = sp.Eq(E, y+q**2/(2*g*y**2))
print("We need to solve for y. The equation is non-linear")
print("Known are E = 1.5, g = 9.81, b=2.5, Q = 6.48 and q = Q/b or q = 6.48/2.5")
eq2 = eq1.subs({E:1.5, g:9.81, q:6.48/2.5}); sp.pprint(eq2) ;
sp.plot(1.5-y-(0.342/y**2),(y, 0.3, 1.5), ylim=[-0.5, 0.25]) ;
sol1 = sp.nsolve(1.5-y-(0.342/y**2), y, 0.5); print("sol1 =", sol1)
sol2 = sp.nsolve(1.5-y-(0.342/y**2), y, 1.4); print("sol2 =", sol2)
```

```
We need to solve for y. The equation is non-linear
Known are E = 1.5, g = 9.81, b=2.5, Q = 6.48 and q = Q/b or q = 6.48/2.5
          0.342429357798165
1.5 = y + ─────────────────
                  2
                 y
```




```
sol1 = 0.625288933447214
sol2 = 1.29655691384252
```

Complete Demonstration at:
https://github.com/prabhasyadav/Faculty_Conclave_2017

Slide Type Slide

Why Jupyter?

1. Suits the modern classroom requirements – many monitors.
2. Is highly extensible – can be personalized as required.
3. Is system independent – browser based
4. has extensive code basis – e.g., Python libraries, R, Julia. Most of them readily available in Smartphones
5. It is free of cost – lessens burden
6. Can be centrally implemented – cloud and server based
7. Is highly supported – large community of independent developers

Challenges with Jupyter Implementation

1. Requires basic programming knowledge for the content developers (e.g., lecturers).
2. Requires students to have some programming and computational skills.
3. Requires motivation
4. Requires support from management, particularly for changing examination pattern

Positive feedback were received when Jupyter based teaching was used in the classroom at Sharda University (2016).

It will be partly introduced in the class this semester at MUJ.

ACKNOWLEDGEMENTS:
Arvind, HOD Civil, Sagar, Chhavi, Sandhya and Mouly

Thank You!

Das Leben ist schön