Exploring Integration of Python Libraries in Computation Intensive Mechanical Engineering Courses

Mr. Manish Patel
(Assistant Professor – Mechanical Engineering)
LJ Institute of Engineering & Technology

***Abstract* - Undergraduate courses specifically the one related to mechanical engineering curriculum have many computation intensive subjects. The curriculum also includes basic introduction to programming. Open source programming languages like python have powerful libraries which can be used to assist in complex engineering computation. This paper explores such libraries and their application in Mechanical Engineering. They can be integrated in the discipline curriculum using front end interactive open source platform like Jupyter. The benefits are twofold. It will encourage active learning among students. Instead of a traditional approach to learning the basics of a programming language, students will learn contextual applications of the programming language in problem solving while studying the subject matter. Instructors can reuse these learning modules and grade their students accordingly.**

Keywords – computation, module; python libraries; Jupyter

I.INTRODUCTION

Engineering undergraduate courses today, have one thing in common. In their curriculum, the subject of introduction to programming is included specifically during first year of the study. The subject is usually a bland introduction to the programming without hands on exploration into full fledged capabilities of the programming language. The same is true with the core engineering subjects involving complex engineering computation. The consequence is reflected in student's inability to solve the problem independently with vigor and passion. The lack of visualization nitty gritty of the problem as well as the solution makes the subject dull and not worth pursuing. Students end up mugging up the data and by the end of the year, they completely forget what they have learned. Visualizing the domain of the problem as well as the solution plays an important role in getting expertise in problem solving technique.

These two problems have one solution. Instead of usual teaching introduction to programming, this subject can be blended with the core discipline subjects. For example, in Mechanical Engineering discipline, the programming language like python can be integrated into the core subject with the help of its powerful libraries. These libraries serve as a tool to analyze and solve engineering computation. These subjects where one requires solving equations, visualizing graphs and networks can be easily done through python libraries.

II. INTRODUCTION TO PYTHON PROGRAMMING

Python is an open source, easy to use and powerful object-oriented high-level programming language with well-designed syntax. It includes large set of standard libraries used for assisting many routine programming tasks. It also serves as an all-purpose computational tool for science and engineering. Since it's an opensource programming language, it is free to download and use. One can modify it, use it in other applications and re distribute free of charge. Though slower than the compiled code, the code readability, inclusion of various packages for computation, fast code execution, efficient debugging and user friendliness contribute to its wider acceptance in the research community.

III.PYTHON LIBRARIES

Python has large database of libraries which have built in modules as a part of standard installation. Python libraries include different functionalities which are related to its library. The functionality of choice can be selected with the help of modules. Inclusive standard libraries help in routine computational and programming tasks. Third party libraries play an important role in extending the capability of python.

Few important standard libraries and modules and their functions are listed below –

a) Os

This module helps the user in interaction with operating system.

b) Statistics

Basic statistical calculations can be performed with help of this module.

c) Timer

Timer module from *timeit* library can be used to compare relative performance of code based on execution time.

d) Math

Math library provides basic mathematical calculation within python interactive development environment.

e) Datetime

Datetime library imparts date and time manipulation in different ways.

Apart from these standard libraries, the libraries listed below have contributed a lot in computational power of python.

a) NumPy

NumPy stands for Numerical Python. This library is dedicated to linear algebra calculations. It serves as a crucial tool for data science. It contains various built in functions for matrix computation.

b) Pandas

Pandas, derived from the term *panel data* is a very flexible and powerful tool used for data manipulation and efficient data indexing. The raw data can be arranged easily into rows and table for further operations.

c) Matplotlib

The objective of engineering computation is the insight and not some numbers. Matplotlib achieves the purpose. It creates graphical visualization of the data similar to commercial software like Matlab. It also generates statistical plots like histogram, boxplots, scatterplots, error charts and so on.

d) SymPy

SymPy stands for Symbolic Python. This module helps in tackling the problem with variable expressions. It assists in coding symbolically. This in turn helps in solving equations which involves symbolic variables.

e) PuLP

PuLP library can be used to solve linear optimization problems in Operations Research. This library can be used in solving linear programming problems.

f) PYro

This library has useful methods and classes for thermodynamic properties calculation of frequently used parameters.

g) SciPy

SciPy (Scientific Python) extends the functionality of python in numerical computation with inclusion of various algorithms and solvers. The library is built and based on NumPy.

These are some prominent libraries available for integration and use in educational curriculum setup. Other libraries and modules can be explored further to improve the lecture delivery in engineering institutions.

IV. INTRODUCTION TO JUPYTER

Jupyter is an open source browser-based application. Jupyter can be installed on personal computer for free and one can also distribute the documents created in it in different formats. In Jupyter application, one can create documents (notebooks) and share them with others. Jupyter Notebooks allows anyone to write programming code, use symbolic language to solve the equations, visualize the given mathematical model and its solution – all within one environment.

Within Jupyter application, the notebook consists of three components –

a) Jupyter Notebook IDE –

Jupyter application runs within interactive development environment (IDE).

b) Kernels –

The notebook runs based on programming language kernels like Julia Python or R.

c) Notebooks –

The notebooks are saved as .ipynb extension. These files can be converted into other formats for easy export and sharing.

The user can test the code and debug it interactively within the environment. These notebooks can run in integrated development environment based on various programming language kernels like Julia, Python and R. The notebooks can incorporate live code, visualization and equations. The key feature of the notebooks is that they are flexible and reusable. The user can share the notebooks in different formats ranging from pdf to html. The notebooks can be used as presentation in academic setup using Jupyter extensions. Students can visualize the result of live code, equations and interactive plotting.

V.INTEGRATION OF LIBRARIES

Mechanical Engineering course curriculum includes computational intensive subjects like Operations Research, Robotics, Computer Aided Design, Machine Design, Engineering Mathematics, Fluid Mechanics, Finite Element Method and many more. The computational part of these subjects can be taught by integrating python modules described above. These integrated modules can be created in Jupyter notebooks and can be loaded in the same platform. The course content can be converted into smaller notebook chunks within Jupyter application. Based on given chapters, these notebooks are categorized accordingly. Within the Jupyter notebook, the chapter content is again divided into topic wise code chunks. These notebooks can be reused in the formal course set up in Engineering curriculum. Many instructors have adopted the same approach to organize their course content.

For example, while teaching Linear Algebra in the class, the instructor can use notebooks integrated with NumPy library for live coding and solving linear equations. To improve the understanding of the concept, one can embed interactive two-dimensional and three-dimensional plots. Each step for solving equations can be coded and run in sequence to create collaborative environment within the class. The student can experiment with different numerical data to get the gist of the concept. Jupyter can help grading capability of each student with the help of its available extensions. In this way, students can clear the mathematical concept which can be difficult to grasp otherwise. It also helps them to use the contextual side of programming language. The simplicity of python language along with the feature rich interactivity and collaboration of Jupyter notebook can overcome the fear in students about the programming. Instead, they will test their programming and problem-solving abilities in different areas and hence this strategy creates fun learning environment with curiosity.

VI. CONCLUSION

The Jupyter notebooks integrated with python libraries can be prepared as modules for formal teaching. These modules can be reused depending upon the curriculum requirement. The results are fruitful to both students and educators. Students will be actively involved during lectures applying these libraries for problem solving. They will learn the practical applications of programming language as well. Instructors can grade students based on their problem-solving approach. Moreover, as these modules are reusable, they can apply them in different setups.

REFERENCES

[1] Van Rossum, G., & Drake Jr, F. L. (1995). Python reference manual. Centrum voor Wiskunde en Informatica Amsterdam.

[2] Barba, L. A., Barker, L. J., Blank, D. S., Brown, J., Downey, A. B., George, T., & Niemeyer, K. E. (2019). Teaching and Learning with Jupyter.

[3] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. Proceedings of the National Academy of Sciences, 111(23), 8410-8415.

[4] Cardoso, A., Leitão, J., & Teixeira, C. (2018, September). Using the Jupyter notebook as a tool to support the teaching and learning processes in engineering courses. In International Conference on Interactive Collaborative Learning (pp. 227-236). Springer, Cham.

[5] Fangohr, H. (2015). Python for Computational Science and Engineering. Faculty of Engineering and the Environment University of Southampton.

[6] Barba, L. A. (2020). Engineers Code: reusable open learning modules for engineering computations. Computing in Science & Engineering.

[7] A Nonprofit Supporting Open Code for Better Science. (2020, March 19). http://www.numfocus.org/.

[8] Van Rossum, G., & Drake Jr, F. L. (1995). Python reference manual. Centrum voor Wiskunde en Informatica Amsterdam.

[9] Barba, L. A., Barker, L. J., Blank, D. S., Brown, J., Downey, A. B., George, T., & Niemeyer, K. E. (2019). Teaching and Learning with Jupyter.

[10] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. Proceedings of the National Academy of Sciences, 111(23), 8410-8415.

[11] Cardoso, A., Leitão, J., & Teixeira, C. (2018, September). Using the Jupyter notebook as a tool to support the teaching and learning processes in engineering courses. In International Conference on Interactive Collaborative Learning (pp. 227-236). Springer, Cham.

[12] Fangohr, H. (2015). Python for Computational Science and Engineering. Faculty of Engineering and the Environment University of Southampton.

[13] Barba, L. A. (2020). Engineers Code: reusable open learning modules for engineering computations. Computing in Science & Engineering.

[14] A Nonprofit Supporting Open Code for Better Science. (2020, March 19). http://www.numfocus.org/.

[15] Van Rossum, G., & Drake Jr, F. L. (1995). Python reference manual. Centrum voor Wiskunde en Informatica Amsterdam.

[16] Barba, L. A., Barker, L. J., Blank, D. S., Brown, J., Downey, A. B., George, T., & Niemeyer, K. E. (2019). Teaching and Learning with Jupyter.

[17] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. Proceedings of the National Academy of Sciences, 111(23), 8410-8415.

[18] Cardoso, A., Leitão, J., & Teixeira, C. (2018, September). Using the Jupyter notebook as a tool to support the teaching and learning processes in engineering courses. In International Conference on Interactive Collaborative Learning (pp. 227-236). Springer, Cham.

[19] Fangohr, H. (2015). Python for Computational Science and Engineering. Faculty of Engineering and the Environment University of Southampton.

[20] Barba, L. A. (2020). Engineers Code: reusable open learning modules for engineering computations. Computing in Science & Engineering.

[21] A Nonprofit Supporting Open Code for Better Science. (2020, March 19). http://www.numfocus.org/.