PAPER • **OPEN ACCESS**

Computer-aided Linear Algebra Course on Jupyter-Python Notebook for Engineering Undergraduates

To cite this article: Chang Tang 2021 *J. Phys.: Conf. Ser.* **1815** 012004

Computer-aided Linear Algebra Course on Jupyter-Python Notebook for Engineering Undergraduates

Chang Tang*

School of Computer Science, China University of Geosciences, Wuhan 430074, China

*Corresponding author email: tangchang@cug.edu.cn

Abstract. As an important basic course, linear algebra plays critical role for engineering undergraduates. However, traditional purely theoretical explanation makes students difficult to deeply understand the essence of linear algebra. In this paper, we introduce a new form of computer-aided linear algebra course with the application of a set of Python computer programs implemented on the Jupyter notebooks platform, and also aim to improve the development of innovative teaching methodologies for this course. In addition, in order to help students understand the intrinsic physical meanings of some concepts, we design some visual examples which demonstrate the practical applications of linear algebra. A serious of user-friendly Jupyter notebooks workflows developed in this work allow teachers–students to intuitively teach and learn theoretical concepts, description of practical problems, programming code, and simulation results with a visual output. The designed environment can help teachers/students to implement their simulations in Python language.

1. Introduction

Linear algebra course is generally taught in the first year of engineering undergraduates, it is very important due to its widely application in solving practical engineering problems. For example, the Singular Value Decomposition (SVD) and Eigen Decomposition are commonly used techniques in computer science, especially in artificial intelligence field. On one hand, the pure theories of linear algebra are hard to understand for the first year undergraduates, which hinders them to apply linear algebra concepts for solving real problems. On the other hand, in many machine learning algorithms of computer science, a serious of linear algebra concepts are often implemented by computer programming languages such as Python, C++ and Matlab, just list a few. Among those programming languages, python language is at the forefront of scientific computation for engineers, and there are a large number of artificial intelligence algorithms implemented by Python.

In this paper, we design and introduce a new linear algebra course by using Jupyter Notebooks for engineering undergraduates, which aims to help students understand the intrinsic physical meaning of some linear algebra concepts. This course is designed to give students a solid foundation in linear algebra in terms of the most useful aspects in an engaging and easy to follow way. In addition, some visual examples which demonstrate the practical applications of linear algebra knowledge are designed for undergraduates to strengthen the learned concepts.

After learning this course, the students should be able to understand the physic meaning of vector, matrix, multiplication of vectors, multiplication of matrices, linear transformation, SVD, eigen decomposition develop, inverse matrix, and finally, their implementation in several practical applications. As to teachers, the Jupyter-notebook environment provide a good platform to showcase theoretical concepts description of problems, programming code, and graphical simulation results. For student, the Jupyter-notebook environment allows them to iteratively read the theory, modify inputs/outputs, and see the immediate effects in an intuitive manner. Moreover, the environment allows students to encode his/her simulations in Python language.

The main workflow of the proposed online-supported linear algebra course based on Jupyter notebook is shown in Figure 1. The whole course can be used by teacher and different students in an interactive modeling computer environment.

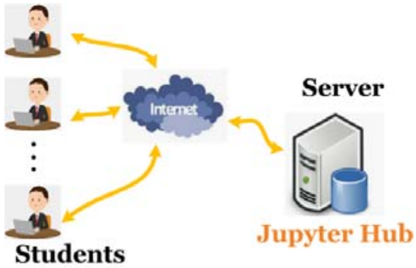

Figure 1. The main workflow of the online environment.

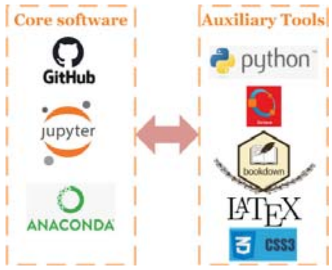

Figure 2. Core software and tools for the implementation of the online course.

2. Related Works

As an open-source software, Jupyter-notebook has been increasingly used for coding and education, especially in higher education about computer science majors. In addition, due to its simplicity, and active support community with plotting and numerical libraries such as IPython, math, scipy, numpy, os, time, and matplotlib [1], the Jupyter-notebook environment has been used as one of the most popular programming platform which can conveniently process programming languages such as Python and Octave [2]. Especially, by using the IPython open-source module, interactively working can be easily realized with the web-cloud interface and Python language.

Table 1. Examples of courses based on Jupyter-notebook in different higher education majors.

Major	Course contents
Artificial intelligence	Artificial intelligence techniques about sustainable development [5]
Computer science	Python programming on Jupyter-notebook [6]
Electronics	Photonic integrated circuits, including Design, simulation, and fabrication on Jupyter notebooks [7]
Electronics	Digital signal processing course based on Jupyter-notebooks [18]
Manufacturing	Optimization problems of manufacturing teaching with code components of a Jupyter notebook [14]
Transport phenomena	Analysis of transport phenomena and reaction in porous catalyst pellet based on a set of Jupyter notebooks [15]
Control	Laboratory oriented control theory courses [16]
Engineering	Computational science education MOOC [8]
Archives management	generous archive interfaces designed with Jupyter notebook [12]
Astronomy	Citation practices in the astronomy community [13]
Environment	Jupyter notebook computing environment for participatory Green infrastructure design, visualization, and evaluation [9]
Chemical	Computational chemistry includes simulations, data processing, and visualization for undergraduate [10]
Agriculture	Data mining and machine learning techniques for agriculture data analysis [11]
Mechatronics	Embedded systems in mechatronics for the engineering curricula [17]

As to higher education, Guerra et al. applied the Jupyter-notebook to a CS2-based course [3], which demonstrates how the Jupyter-notebook can be used for computer course. In [3], the well-known methodologies in industry and manufacturing are used to the teaching of laboratory classes in a computer science course. The "student-centered learning" paradigm is designed to present theoretical and laboratory computer courses. Interactive online course for learning the Python language is also implemented by using Jupyter-notebook [4], which consists of three levels, including basic, intermediate, and advanced levels, and these levels are developed in Jupyter-notebook. Both the students and the teachers can benefit from this course. As to students, they can review any forgotten knowledge and gain new knowledge. As to teachers, they can follow the progress of students and check which part presented them with considerable difficulty. The students and teachers can act in an interactive manner, i.e., messages from students can be easily obtained by teachers while teachers can give the feedback to students timely. In Table 1, we list some examples of courses designed based on Jupyter-notebook in different higher education fields.

3. Detailed Course Implementation

The kernel of the course implementation is a visualized web browser with the Jupyter-notebook Integrated Development Environment (IDE). The environment for programming allows the integration of several tools in addition to Python language, which is helpful to realize an intuitive online course.
The function of different elements are as follows:
(1) The bookdown package [19] allows writing the course document, which includes the insertion of code for figures and quotations.
(2) The stands for Cascading Style Sheets CSS [20] code is included to modify the appearance of the website.
(3) The Latex code allows us to write the mathematical equations conveniently.
(4) The Github client allows us to upload our course documents and Python code to the website, then students can download the materials from the server.
The Python or Octave languages are used for coding, which implement some computation of linear algebra concepts.
For each concept of linear algebra, we first introduce its detailed theory with intuitive visual description, then it is implemented by Python language with some practical applications. Finally, the documents with Python code are uploaded to Github and students can learn the theory and run the code downloaded from Github. Figure 3 shows an example of the course structure.

4. Experimental Results

In this section, we give two course examples with simulations and qualitative satisfaction study of the designed course.

4.1. Course Examples

In each simulation, we use the Jupyter-notebook to running the Python code and the corresponding visual results can be timely obtained. The course examples are presented as follows:
1. Linear equation;

Figure 3. The whole structure of a linear algebra course example.

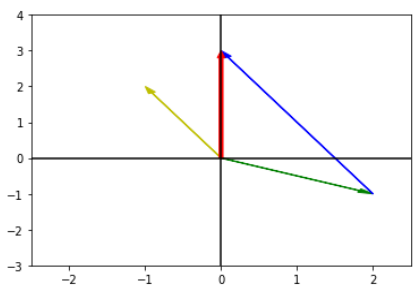

Figure 4. Visualization of column vector of Eq. (2)

2. Principal component analysis;
a) Linear equation. Supposing we have a linear equation with two unknown variables need to be solved, which is formulated as follows:

$$\begin{cases} 2x - y = 0 \\ -x + 2y = 3 \end{cases}, \tag{1}$$

where x and y are two variables which we need to solve. By re-writing the variable coefficients of Eq. (1) into vector form, we have:

$$x\begin{bmatrix} 2 \\ -1 \end{bmatrix} + y\begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}. \quad (2)$$

Then we plot the three column vectors as follows: As can be seen from Figure 3, the red vector, i.e., $[0,3]^T$ can be obtained by adding the green vector, i.e., $[2,-1]^T$ to two times of the yellow vector, i.e., $[-1,2]^T$. Therefore, x and y can be intuitively understood as the linear combination coefficients of two vectors to construct a vector, which is the intrinsic meaning of linear equations.

b) Principal component analysis. As a popular machine learning algorithm, principal component analysis (PCA) is frequently used in many practical problems. In order to give a deep understanding for students, we first introduce the theoretical derivation of PCA and then give a real example on digits de-noising. Due to the page limitation, the document of theoretical derivation is omitted here and we directly show the visual example by using Jupyter-notebook. The detailed process of this course example are as follows:

(1) Import Python Libraries. The implementation codes are shown in Table 2.

(2) Read digits data. The implementation codes for data loading are shown in Table 3.

Table 2. Python code for importing python libraries.

```
# Import Python Libraries
import matplotlib.pyplot as plt
import numpy as np
# Import PCA Algorithm
from sklearn.decomposition import PCA
# Import Dataset
from sklearn import datasets
```

Table 3. Python code for loading data.

```
# Read digits data
digits=datasets.load_digits()
x=digits.data
y=digits.target
noisy_digits=x+np.random.normal(0,2,size=x.shape)
ex=noisy_digits[y==0,][:10]
for num in range(1,10):
    x_num=noisy_digits[y==num,:][:10]
    ex=np.vstack([ex,x_num])
# Print the size of digits
print(ex.shape)
```

(3) Show original digits with noise. Firstly, we give an intuitively showing of original noisy images, and the Python codes are shown in Table 4. The results are shown in Figure 4. As can be seen, original digits images are with noise.

Table 4. Python code for showing original noisy data.

```
# Intuitively showing of the digits
def plot_digits(data):
fig,axes=plt.subplots(10,10,figsize=(10,10),
 subplot_kw={"xticks":[],"yticks":[]},
 gridspec_kw=dict(hspace=0.1,wspace=0.1))
    for i ,ax in enumerate(axes.flat):
        ax.imshow(data[i].reshape(8,8),
cmap="binary",interpolation="nearest",
              clim=(0,16))
    plt.show()
plot_digits(ex)
```

Table 5. Python code for de-noising with PCA algorithm.

```
# Call the PCA algorithm for de-noising
pca = PCA(0.5)
pca.fit(noisy_digits)
a=pca.transform(ex)
b=pca.inverse_transform(a)
plot_digits(b)
```

(4) De-noising by using PCA algorithm. Finally, we call the PCA algorithm for de-noising of the digits images. The Python codes are shown in Table 5.

4.2. Qualitative Satisfaction Study

In order to qualitatively evaluate the designed linear algebra course, we carry out a satisfaction study test. In this experiments, 30 students of the first-year computer science major are chosen from the school of computer science, China University of Geosciences, Wuhan, China. These students have no prior knowledge about linear algebra.

Firstly, a linear algebra course instructor explain how the linear algebra course works to the students. Then we test if the designed course is a satisfactory complementary learning tool for undergraduate students. Therefore, students are asked to fill out a survey which consists of six questions after they finish the online course [18]. For each question, there are five choices, including (1) no, (2) rather no, (3) do not know, (4) rather yes, and (5) yes.

The six questions are as follows:

- Is it useful of the proposed course to understand the linear algebra concepts?
- Easiness of learning with the linear algebra course.
- How the developed linear algebra course motivates you to learn other traditional courses?
- Does it increase your willingness to study at home?
- Dose this linear algebra course allows you to simulate comfortably?
- Do you think this linear algebra course is a complementary tool for traditional course?

The survey results are presented in Table 6. As can be seen from the results, the students are satisfied with the new linear algebra course to a large extend. The designed course is regarded as a useful resource for learning linear algebra.

Table 6. Survey results.

Questions	No	Rather no	Do not know	Rather yes	Yes
Q1	0%	0%	0%	5%	95%
Q2	0%	0%	2%	86%	12%
Q3	0%	1%	2%	83%	14%
Q4	0%	0%	1%	6%	93%
Q5	0%	0%	0%	3%	97%
Q6	0%	0%	1%	11%	88%

5. Conclusions

In this paper, we introduce a linear algebra course based on Jupyter-notebook. The course is implemented on the Jupyter notebooks platform, and aims to improve the development of innovative teaching methodologies for linear algebra. In order to help students understand the intrinsic physical meaning of some concepts, we design some visual examples which demonstrate the practical applications of linear algebra. By learning the designed course, students can understand the hard concepts of linear algebra more easily and this course also improve their learning enthusiasm for other courses.

References

[1] J. F. Pimentel, L. Murta, V. Braganholo and J. Freire. A large-scale study about quality and reproducibility of Jupyter notebooks. In 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), 2019, pp. 507–517.

[2] P. Prathanrat and C. Polprasert.Performance prediction of jupyter notebook in jupyterhub using machine learning. In 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), vol. 3, 2018, pp. 157–162.

[3] H. Guerra, L. M. Gomes and A. Cardoso. Agile approach to a CS2-based course using the Jupyter notebook in lab classes. 2019 5th Experiment International Conference (exp.at'19), Funchal (Madeira Island), Portugal, 2019, pp. 177–182.

[4] M. Djokic-Petrovic, D. Pritchard, M. Ivanovic and V. Cvjetkovic, IMI Python: Upgraded CS circles web-based python course, Comput. Appl. Eng.Educ.24(2016), no. 3, 464–480.

[5] G. Mota-Valtierra, J. Rodrıguez-Resındiz and G. Herrera-Ruiz, Constructivism-based methodology for teaching artificial intelligence topics focused on sustainable development, Sustainability11(2019), no. 17, 4642.

[6] G. Vial and B. Negoita. Teaching programming to non-programmers: The case of python and Jupyter notebooks. International Conference on Information Systems 2018, ICIS, 2018, 2018.

[7] W. Bogaerts, Teaching photonic integrated circuits with Jupyter notebooks: Design, simulation, fabrication, Fifteenth Conference on Education and Training in Optics and Photonics: ETOP 2019 (A. S. Poulin-Girard, and J. A. Shaw, eds.), SPIE, Quebec, 2019, pp. 78–87.

[8] A. Cardoso, J. Leitão and C. Teixeira, Using the jupyter notebook as a tool to support the teaching and learning processes in engineering courses, Adv. Intell. Syst. Comput. 917 (2019), 227–236.

[9] L.Leonard, B.Miles, B.Heidari, L.Lin, A.M.Castronova, B. Minsker, J. Lee, C. Scaife and L. E. Bandh, Development of a participatory Green Infrastructure design, visualization and evaluation system in a cloud supported Jupyter notebook computing environment, Environ. Model. Software 111(2019), 121–133.

[10] C. J. Weiss, Scientific computing for chemists: An undergraduate course in simulations, data processing, and visualization, J. Chem. Educ.94(2017), no. 5, 592–597.

[11] C. N. Vanitha, N. Archana and R. Sowmiya. Agriculture analysis using data mining and machine learning techniques. 2019 5th International Conference on Advanced Computing Computing & Communication (ICACCS), Coimbatore, India 2019, pp. 984–990.

[12] M. Wigham, L. Melgar and R. Ordelman.Jupyter notebooks for generous archive interfaces. 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA 2018, pp. 2766–2774.

[13] M. Wofford, B. M. Boscoe, C. Borgman, I. Pasquetto and M. Golshan, Jupyter notebooks as discovery mechanisms for open science: Citation practices in the astronomy community, Comput. Sci. Eng.22(2019), 5–15.

[14] A. Suárez, M. A. Alvarez-Feijoo, R. Fernández-González and E. Arce, Teaching optimization of manufacturing problems via code components of a Jupyter Notebook, Computer Applications in Engineering Education 26(2018), no. 5, 1102–1110.

[15] B. Golman, A set of Jupyter notebooks for the analysis of transport phenomena and reaction in porous catalyst pellet, Comput. Appl. Eng. Educ.27(2019), no. 3, 531–542.

[16] E. Cruz-Miguel, J. Rodriguez, J. García-Martínez, K. Camarillo-Gómez and G. Pírez-Soto,Field-programmable gate array-based laboratory oriented to control theory courses, Comput. Appl. Eng. Educ.08(2019), no. 27, 1253–1266.

[17] M. Marttınez-Prado, J. Rodriguez, R. GómezLoenzo, K. Camarillo-Gómez and G. Herrera-Ruiz, Short informative title: Towards a new tendency in embedded systems in mechatronics for the engineering curricula, Comput. Appl. Eng. Educ. (2019), no. 27, 603–614.

[18] Zúñiga-López, Arturo, and Carlos Avilés-Cruz. "Digital signal processing course on Jupyter-Python Notebook for electronics undergraduates." Computer Applications in Engineering Education 28.5 (2020): 1045-1057.

[19] Y. Xie, Bookdown: Authoring books and technical documents with R markdown (TF Group, ed.), CRC Press, Boca Raton, FL, 2016.

[20] www.w3.org, Style/CSS, 2020, available at https://www.w3.org/Style/CSS/.