Active investigation and publishing of calculation web based applications for studying process

A A Sutchenkov and A I Tikhonov
Electrical Materials and Components Department, National Research University "Moscow Power Engineering Institute", 14 Krasnokazarmennaya st, Moscow, 111250, Russian Federation

E-mail: mont2@mail.ru

Abstract. The article discusses STEM technologies used in technical specialties students’ teaching to conduct scientific and technical calculations and development of calculation applications. As the platform ecosystem Python, Jupyter Notebook and JupyterLab, Visual Studio Code are used. The use of the listed software tools allows not only to carry out multivariate calculations and calculative experiments but also to describe solved problems formulation, simply and quickly design the user interface and to visualize the calculation results. On the basis of the VirtualBox virtual machine an environment has been developed which allows to develop applications quickly, publish them using Plotly Dash, Voila, Streamlit technologies, control users, content and create electronic active content educational resources with moderate labor intensity.

1. Introduction
The use of STEM (Science, Technology, Engineering, Mathematics) in education involves a large number of visual and interactive applications usage [1, 2] with which students could conduct calculation experiments solving not only direct problems (calculating output parameters by given input parameters) but also inverse problems involving model input parameters selection according to output parameters restrictions. Even for relatively simple inverse problems their solution is practically impossible by input parameters enumerating combinations and requires relationships knowledge between the parameters of the model under study [3]. At the same time applications usage allowing to solve inverse problems makes it possible to introduce play and competition elements into learning process [2, 4].

Nowadays such mathematical software systems as Mathematica, Mathcad, Maple, Matlab, Octave, SMath, Scilab, Sage are used as a tool platform for the interactive STEM applications development and use in educational process. This article considers the possibility of using the Python ecosystem for this purpose. The Python ecosystem is widely used in machine learning, data science [5, 6] and the programming language itself according to some estimates is the leader in educational field [7]. The main reasons for the ecosystem popularity are the huge number of libraries presence for solving scientific and technical problems including computational mathematics, scientific visualization, system administration, web applications; free character, simplicity and low learning curve. It is also important to have distributions that allow you to easily deploy a workplace environment for a wide range of scientific and technical problems development within minutes. Such distributions include portable WinPython [8] for the Windows operating system and Anaconda [9] for MacOS, Linux, and Windows.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd
2. Development environment requirements and use of interactive STEM applications

Interactive STEM applications development and use environment must meet a number of requirements during educational process. "Pure" Python allows to develop scripts simply and with low labor intensity – applications launched from the command line, the results of script execution are displayed in text form in a terminal window, and the results are visualized in separate windows. To change the input data you need to make changes in the scripts source text. In addition, problems arise with calculations results saving organization and reports formation on them. The ecosystem has a large number of tools for building graphical user interfaces [10, 11]. For applications used in the educational process building user interfaces complexity can exceed creating applications complexity themselves. The same situation takes place talking about web applications [12, 13].

Let us formulate the basic requirements for interactive calculation applications development and use during educational process.

- Development and use simplicity and low labor intensity, developed graphical interface, built-in visualization means of results.
- Applications development and use possibility by teachers and students in the process of preparing for classes without programmers and information technology specialists involvement in the educational process. Developed applications’ have easy deployment on servers.
- Electronic textbooks and interactive calculation applications integration is desirable to support hands-on exercises and virtual laboratory work.

The main purpose of this work is to develop a set of technologies and an environment for simple and fast creation, web applications on Python use deployment in educational process that meet requirements above.

It should be noted that technologies for calculation web applications rapid creation and publication are highly developing at present. It is sufficient to mention Mathcad Calculation Server, MATLAB Web App Server, a cloud version of the Mathematica scientific calculations system and finally R Shiny which allows to combine R statistical calculation system power with interactive calculation web applications easy building and publishing. Of the above-mentioned systems only Shiny is free of charge.

3. Jupyter ecosystem

To support calculation activities and practical exercises it is very convenient to use Jupyter ecosystem tools [14] based on the concept of calculating document (CD), sometimes CDs are called notebooks. CD is understood as an environment in which one can:

- describe problem conditions, include formatted text description, formulas, images, audio and video;
- carry out multivariate calculations;
- submit input and output data in a user-friendly form;
- visualize obtained results including in the form of animation;
- present and publish obtained results with minimal additional labor intensity.

Traditionally CD concept is associated with S. Wolfram who first implemented it in Mathematica system. At present times it is generally accepted and to one degree or another is implemented in scientific and technical calculations systems. CDs are most widely used in data science [15, 16] and are convenient environment not only for research but also for information exchange.

The acronym Jupyter is made up of the programming languages names Julia + Python + R symbolizing support for multiple language environments – kernels. Currently not only the listed environments are supported but also many others (more than 30) including Sage, Octave and Scilab scientific and technical calculating systems. The number of kernels supported is constantly growing.
Jupyter CD consists of cells’ two types: calculating and documental. Calculating cells consist of input and output areas. The input area includes supported kernels language constructs as well as "magic" commands used to interact with operating system, estimate runtime, profiling and so on. The output area appears only after cell execution and contains calculations results. CD can contain cells for different kernels for example Python and R and data exchange between cells is possible. You can execute cells either sequentially or in any order. Obtained data in previously executed cells is available in calculating cells forming a single context. Thus, CD is a non-linear media and the user can execute cells in arbitrary sequence at his own risk. On the one hand this makes it possible to make multivariate calculations simply and conveniently and on the other hand it requires keeping in mind cells execution sequence and their variable values. Here we note that it is precisely these CD’s capabilities that are rejected by software developers.

Documental cells can include markup languages HTML constructions, Markdown and Latex, they can embed rich text, pictures, videos. Naturally, things listed above can be built into calculating cells output areas but this is done by means of language kernels.

Users can work with two frameworks: Jupyter Notebook (JN) [15] and Jupyter Lab (JL) [17], in which CDs are called Jupyter notebooks. JN and JL are single-user web applications that run in any modern browser, run on the local machine, and interact with the kernel through the embedded web server. They can perform the full cycle of work with CD including creation, development, execution, interaction with the operating system. CD support is implemented in the integrated development systems JetBrains PyCharm and Microsoft Visual Studio Code (VS Code). In educational process we mainly use JN and VS Code.

Enhancements significantly increase JN and JL capabilities. This is how ipywidgets allows to design user’s interface. The library includes a large number of widgets including input fields, sliders, buttons as well as various containers for constructing layouts of user web interfaces. In addition, interact and interact_manual decorators allow to automate the construction of user interfaces by associating function input values with widgets. Decorators differ in that the first one tracks all changes in the input parameters running decorated function while the second one starts the function only when the automatically generated Run button is pressed. In figure 1 it’s showed a simple application for rectangle corners rounding, the Run and Save buttons are designed to launch the application for a given parameters combination and save the picture to the file system respectively. Parameters are set using three sliders. The application includes two cells: documental cell with task description and output area of a calculating cell with a user interface and results visualization.

![Figure 1. Simple user interface to simulate rectangle corner rounding.](image1.png)

![Figure 2. Jupyter notebook cell as slide of slideshow.](image2.png)
Enhancement nbconvert allows to save CDs in HTML and PDF formats which is convenient when publishing calculation results. The rise enhancement allows CD to be turned into a slideshow. The viewing order and if necessary skipping CD cells that are converted to slides are set directly in the JN, JL user interface. It is possible to change interactively CD’S cells layout by presenting the application as a control panel.

4. Publishing calculation applications
JN and JL are handy tools for visual interactive Python applications development. However they are not always convenient to be used in educational process. Since they contain applications source code which distracts from making calculations accidental changes can make application inoperable. Thus, one needs working environment allowing to publish and work with finished applications.

Virtual machines (VMs) VirtualBox running by Ubuntu Server 18.04 are used as an environment in educational process. Applications execution is carried out under Littlest JupyterHub (TLJH) control [18] allowing dozens of users’ simultaneous operation. TLJH has been supplemented with utilities to facilitate publishing and user management. Applications are accessed through the NGINX web server which also acts as a reverse proxy. CM size does not exceed 10 GB and it works in university internal network, external access is carried out via VPN. The CM has built-in web application that contains training materials and serves as a publishing front-end. In figure 3 it’s showed a fragment of an electronic textbook devoted to fractals.

In the context of educational process it is desirable that JN and Python applications transformation requires minimal modifications, is carried out on Python and does not require study web technologies need.

We have experimented with publishing Python calculation applications since 2016 using our own development which used input declarative description and output objects from which the user interface is generated to transform Python applications into web applications. However after free technologies Plotly Dash [19], PyViz Panel [20], Voila [21] and Streamlit [22] appearance we started working with them. Dash and Streamlit support converting Python applications to web applications, Panel and Voila allow converting Jupyter notebooks.

It should be mentioned that the least time-consuming conversion technology is Voila which allows Jupyter notebooks to run without modification. To launch notebooks one just need to change the uniform resource locator. In addition, Voila supports JN enhancements including dashboards and slideshows,
hides source notebook cells by default. At the same time Voila is the most resource-intensive technologies listed above. This technology is the most popular among students. Panel requires some work on applications and running Bokeh Server, we currently do not support Panel in CMs due to resource constraints. Newly developed Python applications are the easiest to convert to web applications using Streamlit. This technology appeared in 2019 and is currently undergoing intensive development and is not yet stable.

Finally, Dash is the most advanced technology with its own evolving widget system allowing to build web applications with flexible layouts, interactive rendering and the ability to embed Dash applications into Flask and Django web applications. It is possible to develop UI elements in React, making Dash a promising technology for creating virtual labs. At the same time creating Dash applications is associated with high labor intensity, the need to learn new technologies and libraries.

5. Conclusion
Considered technologies make it possible to organize interactive calculation applications development and use in educational process ranging from the description of tasks to carrying out multivariate calculations, results visualizing, generating reports, presenting them in various formats, publishing ready-made web applications. The easiest way to publish web applications is using Voila technology.

The software set designed in CM form supports the simultaneous work of up to dozens of users with Jupyter notebooks publishing web applications using Voila, Streamlit and Dash technologies. This allows students to develop and publish calculation web applications including graphical user interfaces, scientific visualization, electronic educational resources with active content publication and users and content management. Applications and user and content management are developed.

References
[1] Xun G, Ifenthaler D and Spector J M 2015 Emerging Technologies for STEAM Education (Switzerland: Springer International Publishing)
[2] Stohlmann M 2019 Integrated STEM Education through Open-Ended Game-Based Learning Journal of Mathematics Education 12 24-38
[3] Batischew D A 1975 Search methods for optimal design (Moscow: Sovetskoe Radio) (in Russian)
[4] Cabanela J, Gordon A L and Holen S 2019 Introductory Astronomy Interactives developed with Undergraduates American Astronomical Society AAS Meeting 233 id.257.06
[5] Johansson R 2019 Numerical Python Scientific Computing and Data Science Applications with Numpy SciPy and Matplotlib (New York: Apress Media)
[6] Pine D J 2019 Introduction in Python for Science and Engineering – (London: CRC Press)
[7] Schlothauer S 2019 Top programming languages in 2019: Python sweeps the board https://jaxcenter.com/python-ranking-2019-161880.html last accessed 10/09/2020
[8] WinPython The easiest way to run Python, Spyder with SciPy and friends out of the box on any Windows PC, without installing anything! https://winpython.github.io/ last accessed 10/09/2020
[9] Anaconda Individual Edition The World's Most Popular Python/R Data Science Platform https://www.anaconda.com/distribution/, last accessed 10/09/2020
[10] Moore A D 2019 Mastering GUI Programming with Python. Develop impressive cross-platform GUI applications with PyQt (Birmingham-Mumbai: Packt Publishing)
[11] Ulloa R 2013 Kivy: Interactive Applications in Python. Create cross-platform UI/UX applications and games in Python (Birmingham-Mumbai: Packt Publishing)
[12] Django Crash Course: Covers Python 3.8 and Django 3.x. (Two Scoops Press) https://www.feldroy.com/collections/everything/products/django-crash-course, last accessed 10/09/2020
[13] Aggarwal S 2019 Flask Framework Cookbook Over 80 proven recipes and techniques for Python web development with Flask (Birmingham-Mumbai: Packt Publishing)
[14] Project Jupyter https://jupyter.org/
[15] Calea A 2018 *Applied Data Science with Python and Jupyter* (Birmingham-Mumbai: Packt Publishing)

[16] Toomey D 2018 *Jupyter Cookbook* Packt Publishing (Birmingham-Mumbai: Packt Publishing)

[17] JupyterLab 1.0: Jupyter’s Next-Generation Notebook Interface https://jupyter.org/ last accessed 10/09/2020

[18] The Littlest JupyterHub https://tljh.jupyter.org/en/latest/, last accessed 10/09/2020

[19] Tikhonov A 2018 *Dash – a platform for development of Python scientific calculations web applications* (Beau Bassin, Mauritius: LAP Lambert Academic Publishing)

[20] Pane A high-level app and dashboarding solution for Python https://github.com/holoviz/panel last accessed 10/09/2020

[21] Voila-dashbords https://github.com/voila-dashboards/voila last accessed 10/09/2020

[22] Awesome Streamlit https://awesome-streamlit.readthedocs.io/ last accessed 10/09/2020