CoCalc: an integrated environment for open science education in informatics and mathematics

Pavlo V. Merzlykin[1], Maiia V. Marienko[2] and Svitlana V. Shokaliuk[1]

[1]Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine

[2]Institute for Digitalisation of Education of the NAES of Ukraine, 9 M. Berlynskoho St., Kyiv, 04060, Ukraine

Abstract

CoCalc is a cloud-based platform that provides a variety of services and tools for open science education. It allows users to create, share, and collaborate on computational documents that can run various programming languages and frameworks. In this paper, we explore the potential of CoCalc as an integrator of services for learning informatics and mathematical disciplines within the context of open science. We aim to identify the structural elements of the CoCalc environment that are suitable for these disciplines and to examine the prospects of their use. We analyze the structure of the CoCalc kernel and highlight the features that can support different kinds of learning activities, such as interactive coding, data analysis, visualization, simulation, testing, and assessment. We also discuss the challenges and opportunities of using CoCalc in open science education, such as accessibility, reproducibility, transparency, and ethics. We conclude that CoCalc is a promising environment that can enhance the quality and effectiveness of informatics and mathematical education by providing a rich and flexible set of services and tools.

Keywords

CoCalc, open science, informatics, mathematics, education

1. Introduction

Programming skills are essential for many disciplines, not only for computer science. Therefore, it is important to introduce students to programming concepts and methods in different courses, especially those that involve practical applications. This way, students can learn how to use computational tools to solve problems in their fields of interest. However, learning programming is not only about writing code, but also about collaborating with others and sharing knowledge. This is in line with the principles of open education and open science, which are becoming more prevalent in higher education in Ukraine. In an open learning environment, students, researchers, and teachers are equal participants of the same information community, without

3L-Person 2022: VII International Workshop on Professional Retraining and Life-Long Learning using ICT: Person-oriented Approach, October 25, 2022, Kryvyi Rih (Virtual), Ukraine

linuxoid@i.ua (P. V. Merzlykin); popelmaya@gmail.com (M. V. Marienko); shokalyuk@kdpu.edu.ua (S. V. Shokaliuk)

https://kdpu.edu.ua/personal/pvmerzlykin.html (P. V. Merzlykin);
https://iitlt.gov.ua/eng/structure/departments/cloud/detail.php?ID=565 (M. V. Marienko);
https://kdpu.edu.ua/personal/svshokaliuk.html (S. V. Shokaliuk)

0000-0002-0752-411X (P. V. Merzlykin); 0000-0002-8087-962X (M. V. Marienko); 0000-0003-3774-1729 (S. V. Shokaliuk)

a rigid hierarchy. However, in reality, there is still a formal hierarchy in most universities. Therefore, the role of teachers is not only to deliver the course content, but also to manage the information environment and support students with technical issues. Moreover, teachers need to foster a culture of cooperation and communication among all users of the digital environment [1].

Another challenge in teaching programming across disciplines is the diversity of computing systems, methods, and concepts that are used in different fields. This makes it difficult to compare, reproduce, and communicate the results of computational tasks. It also hinders the development of common teaching standards and methodologies. Therefore, there is a need for integrating educational computer systems and research support systems that are tailored for specific disciplines. This will help to enhance the scientific component of students' education, not only in humanities but also in technical specialties. Furthermore, such an integration will facilitate the formation of a community of students and researchers within a single information space. The single digital environment can provide such integration tools.

Science is inherently a collaborative endeavor. Researchers often work in teams to conduct experiments, and this trend has been growing in recent years. Moreover, many experiments are performed using cloud services or platforms, which require suitable tools to support the experimental activities. However, not all tools are designed to facilitate collaboration among scientists. Some tools focus on computation, but neglect the communication and coordination aspects of group work. Even if a cloud-based environment offers some features for work or learning management, it may not address the specific pedagogical challenges of collaborative experiments. Therefore, our research aimed to explore the available tools for students to perform group tasks, conduct joint research, and share their results openly. We believe that conducting experimental research with a group of students, faculty, and researchers is a timely and important issue. We also think that it is necessary to examine every aspect of collaboration among these participants, and to identify the current problems and solutions in this area. In particular, we will present some evidence in the following paragraphs that shows how cloud service tools can be used as a means of open science.

SageMath is an open-source computer algebra system that has been widely used for research problems related to algebra and geometry. However, it has also evolved into an open-source cloud service that supports collaboration, and allows users to work with Python, R, Jupyter, LaTeX, and more. Furthermore, the CoCalc cloud service enables teachers to customize their own LMS environment. Programming, LaTeX usage, simulation – these are new skills for mathematics students, and such environments help them develop these skills [2].

2. Literature review

Klaßmann et al. [1] conducted a case study on the evolution of the digital learning and research environment at the Department of Musicology, University of Cologne. They analyzed 14 seminars from 2016 to 2020, and focused on the technological configuration of the digital environment and the curriculum development, which included digital literacy education and interdisciplinary connections [1].

De Assis Zampirolli et al. [3] explored MEGUA (Mathematics Exercise Generator, Universidadede Aveiro) 2 – an open source software that allows users to create databases of parameterized questions and answers in LaTeX. It works with the CoCalc mathematical software, which uses the Python programming language [3]. The databases of questions are called “Books’’ and are built with PDFLatex (for printing) or HTML and MathJAX (for web publications) [3]. The creation of questions takes place within the CoCalc tools, and consists of three steps:

1) a new worksheet is created, where a cell is used to import the MEGUA library and open or create a database to store questions;
2) another cell is used to write the question code, which consists of LaTeX text and Python code. The LaTeX block has sections for cataloging and describing the exercise, “% of the problem” (name and question), and “% of the answer’’ (solution);
3) CoCalc completes the computation part, which contains two functions: it generates random values for the variables, calculates the correct solution, and generates other multiple choices.

This cell produces two files: one in PDF format and another in text format [3, 4].

MEGUA also has a feature for adding parameterized graphs to exercises, but it does not have automatic correction for printed copies of questions, or a function for grading hundreds of users.

The problem of developing a curriculum for operations research courses has been addressed by Vlasenko et al. [5]. Their research focuses on the use of cloud computing for solving optimization problems. They confirm the suitability of using the CoCalc cloud environment in teaching students.

Bobyliev and Vihrova [6] examined the experience of implementing courses in Calculus and History of Mathematics for future mathematics teachers in the learning management system of Kryvyi Rih State Pedagogical University. They used a block-modular approach to design courses, which allows them to structure the online learning process of fundamental mathematical subjects, and to control the students’ pace and depth of learning. They also provide examples of laboratory classes on Calculus that students performed independently in the CoCalc computer mathematics system.

Gavrilyuk [7] discussed the challenges of using cloud services under quarantine conditions. They considered the possibilities of using cloud technologies for distance learning under preventive measures, and highlighted CoCalc as a key cloud service. They also gave an overview of cloud services that can be used to study Mathematics and Statistics related disciplines, and provided their brief characteristics.

The aim of the study is to identify the structural elements of the CoCalc environment that are appropriate to use in the educational process in the context of open science.

3. Results

CoCalc (**Co**llaborative **Calc**ulation and Data Science; mode of access cocalc.com) is a virtual online workspace (cloud-based environment) for calculations, research, authoring documents in collaboration mode.

The learning and scientific activities in the CoCalc environment involve working on a project. The elements of a project are folders and files in different formats.

It is through the project files that the student and/or scientist accesses the main components of CoCalc explicitly (figure 1) or through an "intermediary" (file type "X11 desktop", figure 2).

According to CoCalc's statistics over the last month, the most popular environment instrumental and applied components are Jupyter Notebooks, Sage Worksheets, LaTeX Documents

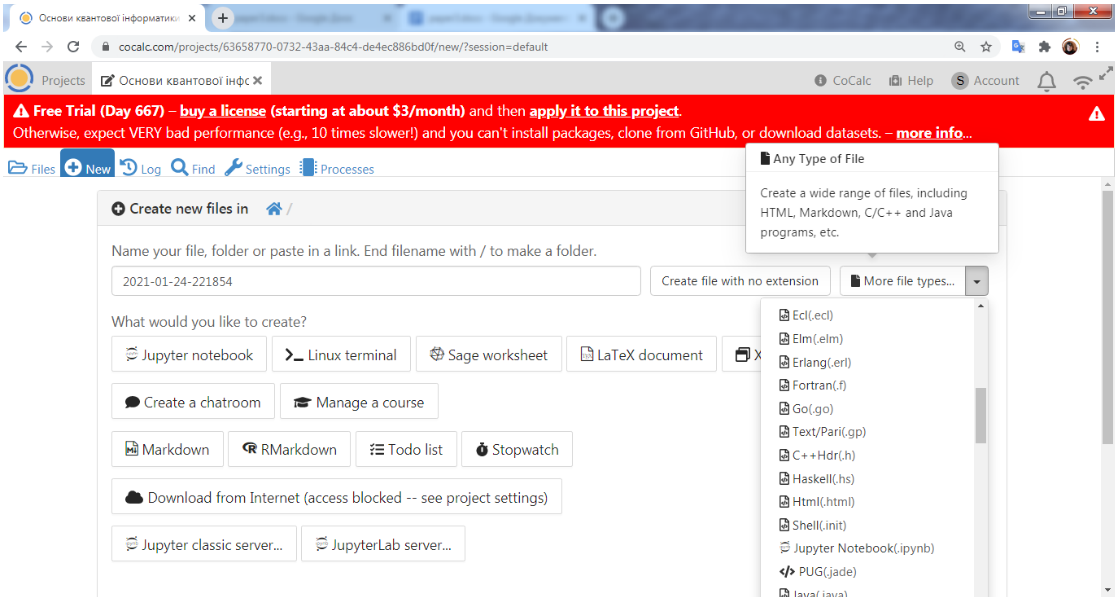

Figure 1: Page to create a new project file.

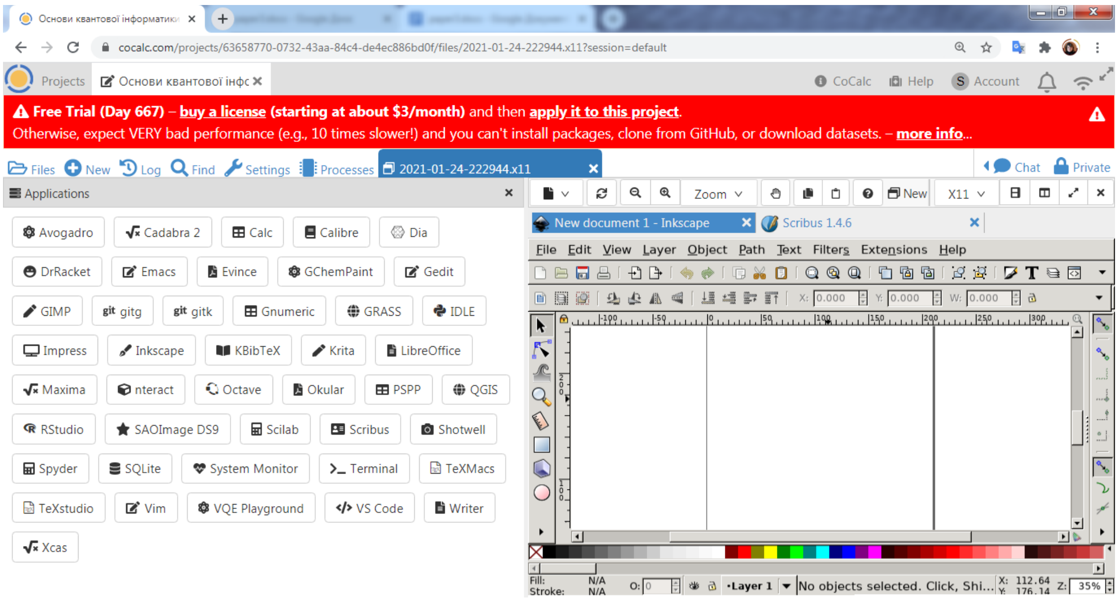

Figure 2: Page of a new file of type "X11 desktop".

and R Markdown Documents.

The popularity of Jupyter Notebooks is obvious. Because it is on Jupyter Notebooks that you can modeling (calculate, programming, etc.), with the functionality of SageMath or Python or R or Julia.

Before talking about the already popular tools (SageMath, Python, R, LaTeX), let's focus on the latter mentioned, Julia.

Julia is a high-level, high-performance programming language with dynamic typing for mathematical calculations. The syntax is similar to the matlab family, the language is written in C, C++ and Scheme, it is possible to call C libraries.

Julia was designed from the beginning for high performance. Julia programs compile for efficient native code for multiple platforms via LLVM.

Julia plays dynamically, is a scripting language and has good support for interactive use.

Playable environments make it possible to play the same Julia environment every time, on different platforms, with pre-built binaries.

Julia uses multiple sending as a paradigm that facilitates the expression of many object-oriented and functional programming patterns. Provides asynchronous I/O, metaprogramming, debugging, logging, profiling, package manager, and more. You can create entire programs and microservices in Julia.

Julia is an open source project with more than 1,000 authors. It is provided under MIT.

But first of the stages in the development of the CoCalc is a web Computer Mathematical System (web-CMS) *SageMath*.

SageMath is a free open-source mathematics software system based on many existing open-source mathematical packages – FLINT, GAP, Matplotlib, Maxima, NLTK, Numpy, Pandas, Scikit Learn, Scipy, Statsmodels, SymPy, and many others. They can be accessed using a generalised language based on Python, or directly through interfaces or shells.

The available web-CMS tools of SageMath version 4.6 (the latest version before the advent of CoCalc, even earlier than SageMathCloud) were not sufficient to organize all types of learning activities under distance learning or its elements. It was necessary either to organize training or with the involvement of two systems – web-CMS SageMath and any system to support distance learning, such as Moodle, or to integrate them. The first method proved to be inconvenient for neither teachers nor students, the second method – continues to be widely used [8], but it, with the advent and improvement of CoCalc, may lose relevance.

Since 2014, more than 80 students have completed the courses "Computer Technologies in Research" and "Computer Mathematics" for future computer science teachers with the additional qualification "applied programmer". The SageMath toolkit in CoCalc became especially popular with the advent of the ability to work on interactive Jupyter Notebooks instead of Sage Worksheets [9]. While the latter has the advantage of being able to work simultaneously (within one sheet) with different mathematical applications.

In addition, future teachers of mathematics and computer science were offered to master the tools of SageMath in CoCalc within the optional course "Using SageMathCloud in learning mathematics" (by Maiia V. Marienko), the course "Numerical Methods / Methods of Computing / Computational Mathematics", "Discrete Mathematics", "Operations Research", "Mathematical Programming", as well as to perform independent work on the courses "Linear Algebra and

Table 1
The main components (components, software) CoCalc: System software.

Type of software	Name of the software
Request and process user account information	accountsservice
FTP client	CFTP
VNC server	X11vnc
Archiver	7-ZIP, gzip, tar
Free command line utility for data compression	bzip2
Garbage collector	The Boehm-Demers-Weiser
Shell for GNU Screen and Tmux (application)	Byobu
Shell for Python GD library	gdmodule
Program for displaying a list of running processes	htop, ps
SageMath Notebook Server	SageMathNB
Operating System	Debian GNU/Linux

Numerical Systems", "Analytical and Differential Geometry", "Calculus", "Probability Theory and Mathematical Statistics".

The mathematical packages FLINT, GAP, Matplotlib, Maxima, NLTK, Numpy, Pandas, Scikit Learn, R, Scipy, Statsmodels, SymPy, TensorFlow are known as members of the *Python Scientific Computing Ecosystem* or more simply *Scientific Python* because they provides data processing (modeling, experiment control) and visualize results for quick analysis with high-quality metrics for reports or publications.

Among the tools mentioned, the packages *TensorFlow* and *R* are of particular note.

TensorFlow is a comprehensive open source platform for machine learning. It has a comprehensive flexible ecosystem of community tools, libraries, and resources that allows researchers to advance the latest advances in machine learning, and developers can easily create and deploy machine-based applications.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Among other things it has

- an effective data handling and storage facility;
- a suite of operators for calculations on arrays, in particular matrices;
- a large, coherent, integrated collection of intermediate tools for data analysis;
- graphical facilities for data analysis and display either directly at the computer or on hardcopy;
- a well developed, simple and effective programming language (called 'S') which includes conditionals, loops, user defined recursive functions and input and output facilities. (Indeed most of the system supplied functions are themselves written in the S language.)

R is very much a vehicle for newly developing methods of interactive data analysis. It has developed rapidly, and has been extended by a large collection of packages.

Since September 2018, almost 50 PhD candidates have been involved with the R toolkit in CoCalc and have successfully completed the Modern Information and Communication Technology in Research course.

To support cumbersome scientific calculations, there is a need to reduce the computational delay. Edge computations adopt a decentralized model that brings cloud computing capabilities closer to the user equipment to reduce computational latency. There are two types of projects in CoCalc: “trial (free) projects” and “participating projects”. Trial projects run on computers that share the same node with many other projects and system tasks. These nodes may also stop at any time, causing the current project to interrupt and restart.

Projects accepted by members are transferred to less loaded machines, which are reserved only for users who have purchased one of the proposed licenses (tariff plans). Those servers are not being restarted daily. The cluster is dynamically scaled to accommodate different numbers of member projects.

Work on members projects is much smoother because commands are executed faster with less delay, and heavy operations of the processor, memory and I/O work faster.

By default, free projects stop working after about 30 minutes of inactivity. This makes the calculations quite time-consuming.

There is an advanced license option to completely prevent downtime. Processes can still stop if they use too much memory, crash due to an exception, or or being restarted by the server on which they are running.

That is, for users who have purchased one of the proposed tariff plans, there are more opportunities to use edge calculations.

Also, it is possible to change the free tariff plan (default) Hub server by clicking “Reconnect” (figure 3). To some extent, this setting may also be considered as a practical use of edge computing [10].

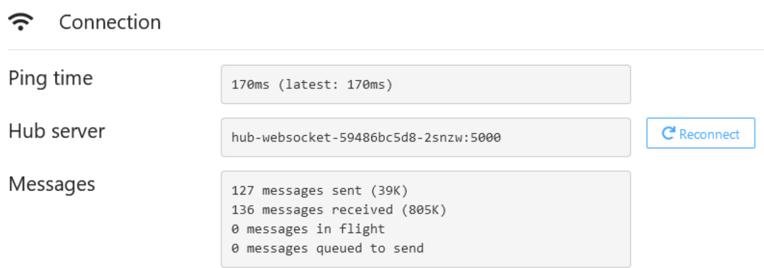

Figure 3: Pop-up settings “Connection”.

In addition, we should mention Big Data. The complexity arises from several aspects of the Big Data lifecycle, such as data collection, storage on cloud servers, data cleaning and integration. But edge computing solves this problem, which is an essential point for working with CoCalc.

CoCalc offers a wide collection of software environments and libraries (see tables 1-4).

A complete list of the current versions of CoCalc (1267 Python packages, 4472 R packages, 447 Julia libraries and more than 243sd files have been installed) can be obtained by using the command `$ sudo dpkg --get-selections`.

Detailed information on the specified in tables 1-4 and other CoCalc components (at the time of publication) can be obtained by direct link https://cocalc.com/help on the official website of

Table 2
CoCalc main components: General purpose application software.

Type of software	Name of the software
Analog screen for graphics programs	Xpra
Database of combinatorial graphs	Graphs
Library for rasterization of fonts and operations on them	FreeType
Library for working with raster graphics in PNG format	Libpng
GNOME tooltip browser	Yelp
File management and collaboration system	Mercurial
Electronic dictionary (thesaurus)	WordNet
Image viewer	GPicView
Interactive editor and macro support	Prerex
Programs for comparing the contents of text files and directories	Meld, diff
Services for reading e-books	Calibre, Evince
Document processing system in HTML, LaTeX or XML document formats	Docutils
Database management systems	RethinkDB, sqlite3
Text editors	GNU Emacs, Vim, nano, mcedit, AbiWord
Utility for finding differences between files	GNU patch
Cloud file storage	Dropbox

the CoCalc project.

Implementation of research projects, term papers with the use of CoCalc involves two ways:

1. Using the individual tools presented in CoCalc.
2. Execution, writing and registration of results of educational and research work in CoCalc without involvement of auxiliary software.

At the same time, teachers and a group of students can be involved in the research project.

The IPython interpreter in the process of training future mathematics teachers can be used to develop dynamic models with semi-automatic / automatic demonstration modes.

The first way involves creating a model (models) of the phenomenon under study on a worksheet using standard controls, HTML tags, LaTeX commands and using CSS.

The disadvantages of this use are that in the process of registration of the obtained results have to involve other software: text editor, software for creating presentations, video editor (if necessary). As a result, only a certain point of the research work was performed using the CoCalc toolkit. In addition, in the process of presenting scientific findings, the student will have to demonstrate to their colleagues in addition to the presentation of the developed model using a browser (or video editor). This can be avoided by using CoCalc tools not only to perform the research part of a particular job. Therefore, it is better to use the built-in LaTeX editor as a CoCalc tool.

LaTeX is a high-quality text document program.

LaTeX is a TeX-based macrosystem that aims to simplify its use and automate many common formatting tasks. This is the de facto standard for academic journals and books, and it offers one of the best free typography programs it has to offer.

Table 3
CoCalc main components: Special purpose application software.

Type of software	Name of the software
Automatic grid generator for geometric constructions	Gmsh
Software package for algebraic, geometric and combinatorial problems on linear spaces	4ti2
Library for performing problems in number theory	FLINT
Library for dynamic work with images	GD Graphics Library (GD)
Library for processing video and audio files	Ffmpeg
Library for working with graphs and other network structures	NetworkX
Library for solving linear programming problems	GLPK
Library for solving convex programming problems	CVXOPT
Library designed for applied and scientific mathematical calculations	GNU Scientific Library (GSL)
Libraries for determining and calculating elliptic curves defined over a field of rational numbers	eclib
Vector graphic editor	Inkscape
Sage versions	Sage.7, Sage.8, Sage.9, Sage.10
Client for Git repository	SparkleShare
Mathematical library	Cephes
Mathematical library for performing actions on complex numbers	GNU MPC
A set of libraries that extend the functionality of C++	Boost
SageTeX package extension	SageMathTeX
Software package for generating three-dimensional models	GenModel
Software package for scientific calculations	Scilab
Software packages for building phylogenetic trees	Phylip
System for mathematical calculations	GNU Octave
Computer algebra systems	Gias/Xcas, Axiom, GAP
Computer mathematics system	Maxima

Performing a term paper or a thesis in the LaTeX editor, the student has the opportunity to print it, preformed on the basis of a resource such as tex PDF-document.

That is, at the same time there is a process of registration of the obtained results, calculations, presentation and presentation of the main provisions of the study (using the presentation developed in the LaTeX editor) and demonstration of the created model. The student does not need to include additional software to perform, design or present the results, because all the work is completely unified within one cloud service – CoCalc.

```
\documentclass{article}
\usepackage[a5paper]{geometry}
\usepackage[utf8]{inputenc}
\usepackage[ukrainian]{babel}
\usepackage{sagetex}
\title{Sharing Sage and LaTeX}
\author{M. V. Marienko}
```

Table 4
CoCalc main components: Software tools.

Type of software	Name of the software
Interactive shell for programming	Jupyter Notebook
Python programming language interpreters	Python 2.x, Python 3.x, Python (Anaconda)
C ++ programming language compilers	C++
Interpreters	CPython, Java, Perl, bash
Compilers	Mono, Embeddable Common Lisp
Functional programming environments	DrRacket, MIT/GNU Scheme
Environment for statistical calculations, analysis and presentation of data in graphical form	R

```
\date {January 13, 2023}
\begin{document}
\maketitle
The easiest way to embed the results of Sage commands
in the tutorials created in LaTeX is to use the sage and
sageplot tags:”
a) finding the derivative:
$(x^3)’=$$\sage{diff(x^3,x)}$
b) plotting:
\sageplot{plot(sin(x),-pi,pi)}
\end{document}
```

You can of course offer an alternative to CoCalc – Jupyterhub and Zoom. However, they do not include the ability to synchronize with other community members in a text file, although Zoom has a basic real-time chat feature. Of course, you can offer to integrate the Markdown hypertext into the configuration by using the Jupyter Notebook, which seemed to be the ideal solution to enable collaboration in a browser-based text document in real time using Zoom, for example in workshops. In addition, HackMD Markdown files will be available to students at any time and will be used for notes during the workshop. In this way, you can create joint documents that implement synchronous and asynchronous discussions. In addition, HackMD will provide tools for documenting group work sessions so that it is easy to share with other users. In this way, you can create templates for courses that will be used later for notes, discussion of seminar topics outside the classroom. Currently, Jupyterlab does not allow real-time collaboration on real-time collaboration due to technical limitations.

CoCalc offers shared computing capabilities to small groups of users. It also includes basic chat and video conferencing features. CoCalc toolkit supports student projects and group assignments that require synchronous collaboration in computer science and math. Because CoCalc is also based on the Jupyter Notebook, integration with individual workspaces will be seamless, as users in the same group can easily transfer individual files between CoCalc to both the shared workspace and their own, private instance of Jupyterlab. Using the advanced configuration with Zoom, HackMD and CoCalc, seminars can be organized completely remotely [1].

Overall, this configuration is a good starting point for the further evolution of the digital environment and the management of a group of students to increase digital literacy in interdisciplinary research and the teaching of computer science and mathematics. To assess the cloud environment, it is necessary to take into account both the student's opportunities and interaction with them, as well as the success in achieving interdisciplinary learning goals and the level of discussion of the content achieved in seminars. CoCalc cloud service can be recommended to groups of students of all academic levels, from bachelor to doctoral and teachers of various fields of science. The use of a single cloud platform has certain advantages: it will help to form and hold regular meetings to discuss modern computational approaches in interdisciplinary research. This creates a digital environment for developing students and researchers that goes beyond weekly seminars. From the point of view of teaching, seminars conducted in one case study will confirm the potential of a common information environment for teaching computational interdisciplinary research. Thus, students with limited programming experience or no previous programming experience during distance learning workshops will be able to fully learn the basics of Python programming and gain skills in discussing and implementing high-level computational models [1].

The evolution of the configuration of the digital environment demonstrates clear progress, which is closely linked to the requirements of pedagogical and methodological practices within the developing free economic system, students and researchers. Thus, the resulting configuration for the introduction of computational thinking and digital literacy consists of the following tools that support the necessary functions in a single digital environment:

- Jupyter Notebook, which is serviced through Jupyterhub, will provide a basic environment for notes, programming and working with computational methods and concepts without the need for local installation and maintenance.
- GitHub, GitHub Pages, and GitHub Classroom will be used to track file versions, create a course website as an alternative communication channel, and support the logistics of issuing and submitting course assignments.
- Zoom will provide a tool for interactive synchronous social communication in distance and face-to-face learning.
- HackMD is used for synchronous co-writing of hypertext documents.
- CoCalc provides collaborative real-time programming based on the Jupyter Notebook.

4. Discussion

Ukraine has adopted a roadmap for its integration into the European Research Area (ERA-UA) by the decision of the Ministry of Education and Science of Ukraine No. 3/1-7 on March 22, 2018. One of the priorities of this roadmap is to promote the development of open science in Ukraine. Open science means making the research process transparent by publishing all its results and details on how they were obtained, and making them publicly accessible on the Internet.

The practical implementation of the open science paradigm involves [11]: sharing educational materials in open access (data, event program, abstracts, meeting minutes, didactic materials, data analysis files); publishing materials in open access journals; freely distributing and disseminating educational and scientific materials and data (for example, uploading content to an open repository).

According to Shyshkina [11], the principles of open science include [8]:

- open access to scientific sources;
- open access to electronic resources used during the research;
- free access to data sets obtained during a pedagogical experiment;
- open e-infrastructures.

A common example of open source is the large number of open source virtual learning environments used in the academic setting. The most notable example is Moodle, which is widely used in educational institutions [12].

Therefore, the introduction of open science standards in Ukraine should lead to more exchange, accountability, reproducibility and reliability of scientific materials, and affect the learning process as a whole. In the process of studying domestic and foreign experience, we identified the following benefits of using cloud services for mathematical purposes: resource saving; mobility of access; flexibility.

The use of cloud platforms and services in the educational process leads to the emergence and development of forms of education and research organization that are focused on collaborative learning activities, and create more opportunities for educational and research projects [13, 5, 14, 15, 16, 17, 18, 19]. The methods and approaches of open science have a significant impact on the educational process. Considering the above advantages of cloud-based tools in teaching mathematical disciplines, as well as the prospects of implementing the CoCalc cloud service in the educational process, we consider this service to be a potential cloud component of open science.

CoCalc is a cloud service that provides a virtual workspace for computation, research, collaboration and document creation [4]. It contains a cloud storage where researchers can share files with their colleagues. These include Jupyter notebooks, where multiple researchers can edit scripts in real time.

CoCalc [4] supports query, discovery and visualization subphases. This allows researchers to query the results and history of the experiment, among other data. Users can also visualize results using Jupyter notebooks and libraries, such as matplotlib. They can also use chats to discuss the experiment and its stages.

In this cloud service [4], the entire experimental environment is based on the principle of cloud operation. All changes are made directly in the cloud and synchronized with the user's browser via the Internet, without any blocking.

CoCalc [4] allows users to share various types of files, including scripts in different programming languages. The cloud service tools enable users to share documentation that can help researchers understand what has been done in the experiment and how to better use the shared data and workflows.

The cloud service [4] also allows users to store their interactions in a journal (chronology), but it is more like unstructured information that is hard to reproduce.

CoCalc [4] allows one to share a wide variety of files, including scripts in different programming languages. The cloud service toolkit allows you to share documentation that can help scientists understand what has been done in the experiment and help them make better use of shared data and scenarios.

The cloud service [4] makes it possible to store performed by scientists interaction in a journal (chronology), but it resembles more unstructured information that is difficult to reproduce.

CoCalc [4] enables users to share various kinds of files, including scripts in different programming languages. The cloud service tools also allow users to share documentation that can help researchers understand the experiment and how to use the shared data and workflows effectively.

The cloud service [4] also provides a way to store the interactions of the researchers in a journal (chronology), but it is more like unstructured information that is hard to reproduce.

Although the cloud service is fully ready for use in research [4], it requires a stable Internet connection to work. Working with the service is possible through the browser, but this may cause some difficulties when switching from the workspace, tools and development environments that the researcher is used to. Users can run code from the CoCalc environment, but this method is different from running files from the user's device. There are also some limitations on using a free cloud service account. Another problem worth mentioning is that CoCalc does not capture all the stages of the experiment adequately. It offers features such as time travel" and log'' that allow users to see the history of file changes and project activity. But these data are not detailed enough to ensure the reproducibility of the experiment.

We can conclude that CoCalc meets all the principles of open science. And CoCalc tools can be considered as open science tools that have didactic potential in the learning process.

5. Conclusions

The chronology presented in this paper shows the creation and adaptation of the digital environment based on the specific needs and practical tasks of a group of students, teachers and researchers in interdisciplinary research and education. As the digital environment is constantly evolving, the research cannot be considered conclusive. We plan to integrate the configuration of CoCalc and the curricula of individual disciplines for a deeper understanding of the learning material and to expand the means of forming professional competencies of future specialists in various fields of education and science. CoCalc tools enhance students' ability to organize and perform teamwork by implementing a joint project task. Thus, using the cloud service improves the indicators of scientific research, makes the educational process more open, relevant to human needs and content.

Given the growing popularity of free software and the wide range of applications and services offered by CoCalc, it is important to note that there is a need to develop teaching materials for Computer Science and Mathematics.

The use of cloud services leads to the emergence and development of learning forms that are focused on collaborative learning activities on the Internet. Cloud services should be used in Mathematics teachers training as a means of: communication; cooperation; data storage and processing, which should be the subject of further research. It is advisable to focus further research on the dissemination of open science approaches in Mathematics teachers training process.

References

[1] S. Klaßmann, N. Dahmen, U. Seifert, A digital habitat for interdisciplinary music research and teaching, in: D. Kayser (Ed.), Proceedings of the 13th International Conference of Students of Systematic Musicology (SysMus20), York, United Kingdom, 2020, pp. 59–69. URL: https://kups.ub.uni-koeln.de/51581/. doi:10.17605/OSF.IO/KAS63.

[2] O. Martines, CoCalc como herramienta de aprendizaje, 2020. URL: http://funes.uniandes.edu.co/22727/1/Martinez2020CoCalc.pdf.

[3] F. De Assis Zampirolli, F. Teubl, V. R. Batista, A generator and corrector of parametric questions in hard copy, in: S. Latifi (Ed.), 16th International Conference on Information Technology-New Generations (ITNG 2019), Springer International Publishing, Cham, 2019, pp. 269–275.

[4] E. Jandre, B. Diirr, V. Braganholo, Provenance in collaborative in silico scientific research: A survey, SIGMOD Rec. 49 (2020) 36–51. doi:10.1145/3442322.3442329.

[5] K. Vlasenko, O. Chumak, D. Bobyliev, I. Lovianova, I. Sitak, Development of an Online-Course Syllabus “Operations Research Oriented to Cloud Computing in the CoCalc System”, in: A. Bollin, H. C. Mayr, A. Spivakovsky, M. V. Tkachuk, V. Yakovyna, A. Yerokhin, G. Zholtkevych (Eds.), Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I: Main Conference, Kharkiv, Ukraine, October 06-10, 2020, volume 2740 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 278–291. URL: https://ceur-ws.org/Vol-2740/20200278.pdf.

[6] D. Y. Bobyliev, E. V. Vihrova, Problems and prospects of distance learning in teaching fundamental subjects to future mathematics teachers, Journal of Physics: Conference Series 1840 (2021) 012002. doi:10.1088/1742-6596/1840/1/012002.

[7] O. D. Gavrilyuk, The role of cloud services in quarantine, in: Proceedings VIII International scientific Internet conference Global and Regional problems of Informatization in Society and Nature Using ’2020, NULES of Ukraine, Kyiv, Ukraine, 2020, pp. 183–185.

[8] S. Shokaliuk, Y. Bohunenko, I. Lovianova, M. Shyshkina, Technologies of distance learning for programming basics on the principles of integrated development of key competences, CEUR Workshop Proceedings 2643 (2020) 548–562.

[9] O. Markova, S. Semerikov, M. Popel, CoCalc as a learning tool for neural network simulation in the special course ”Foundations of mathematic informatics”, CEUR Workshop Proceedings 2104 (2018) 388–403.

[10] S. O. Semerikov, A. M. Striuk, T. A. Vakaliuk, A. V. Morozov, Quantum information technology on the Edge, CEUR Workshop Proceedings 2850 (2021) 1–15. URL: http://ceur-ws.org/Vol-2850/paper0.pdf.

[11] M. P. Shyshkina, The use of the cloud technologies to support the educational research in the open science area, New computer technology 16 (2018) 105–115. doi:10.55056/nocote.v16i0.824.

[12] P. Nechypurenko, S. Semerikov, VlabEmbed - the New Plugin Moodle for the Chemistry Education, in: V. Ermolayev, N. Bassiliades, H. Fill, V. Yakovyna, H. C. Mayr, V. S. Kharchenko, V. S. Peschanenko, M. Shyshkina, M. S. Nikitchenko, A. Spivakovsky (Eds.), Proceedings of the 13th International Conference on ICT in Education, Research and

Industrial Applications. Integration, Harmonization and Knowledge Transfer, ICTERI 2017, Kyiv, Ukraine, May 15-18, 2017, volume 1844 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 319–326. URL: https://ceur-ws.org/Vol-1844/10000319.pdf.

[13] P. Nechypurenko, T. Selivanova, M. Chernova, Using the Cloud-Oriented Virtual Chemical Laboratory VLab in Teaching the Solution of Experimental Problems in Chemistry of 9th Grade Students, in: V. Ermolayev, F. Mallet, V. Yakovyna, V. S. Kharchenko, V. Kobets, A. Kornilowicz, H. Kravtsov, M. S. Nikitchenko, S. Semerikov, A. Spivakovsky (Eds.), Proceedings of the 15th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops, Kherson, Ukraine, June 12-15, 2019, volume 2393 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 968–983. URL: https://ceur-ws.org/Vol-2393/paper_329.pdf.

[14] S. Papadakis, A. E. Kiv, H. M. Kravtsov, V. V. Osadchyi, M. V. Marienko, O. P. Pinchuk, M. P. Shyshkina, O. M. Sokolyuk, I. S. Mintii, T. A. Vakaliuk, A. M. Striuk, S. O. Semerikov, Revolutionizing education: using computer simulation and cloud-based smart technology to facilitate successful open learning, CEUR Workshop Proceedings 3358 (2023) 1–18.

[15] S. Papadakis, A. E. Kiv, H. M. Kravtsov, V. V. Osadchyi, M. V. Marienko, O. P. Pinchuk, M. P. Shyshkina, O. M. Sokolyuk, I. S. Mintii, T. A. Vakaliuk, L. E. Azarova, L. S. Kolgatina, S. M. Amelina, N. P. Volkova, V. Y. Velychko, A. M. Striuk, S. O. Semerikov, Unlocking the power of synergy: the joint force of cloud technologies and augmented reality in education, CEUR Workshop Proceedings 3364 (2023) 1–23.

[16] V. Oleksiuk, O. Oleksiuk, The practice of developing the academic cloud using the Proxmox VE platform, Educational Technology Quarterly 2021 (2021) 605–616. doi:10.55056/etq.36.

[17] T. Vakaliuk, O. Spirin, O. Korotun, D. Antoniuk, M. Medvedieva, I. Novitska, The current level of competence of schoolteachers on how to use cloud technologies in the educational process during COVID-19, Educational Technology Quarterly 2022 (2022) 232–250. doi:10.55056/etq.32.

[18] V. Velychko, E. Fedorenko, N. Kaidan, V. Kaidan, Application of cloud computing in the process of professional training of physics teachers, Educational Technology Quarterly 2021 (2021) 662–672. doi:10.55056/etq.38.

[19] P. P. Nechypurenko, S. O. Semerikov, O. Y. Pokhliestova, Cloud technologies of augmented reality as a means of supporting educational and research activities in chemistry for 11th grade students, Educational Technology Quarterly 2023 (2023) 69–91. doi:10.55056/etq.44.