
Using Python and Google Colab to Teach Physical Chemistry During Pandemic

Leonardo Baptista

Universidade do Estado do Rio de Janeiro, Faculdade de Tecnologia, Departamento de Química e
5 Ambiental

ABSTRACT

The COVID-19 pandemic brought the need to reinvent teaching pedagogical methods to engage
undergraduate students in remote classes. Synchronous and asynchronous meetings become routine
in the undergraduate courses, and web-based learning methods proved to be useful in the present
10 situation. In this scenario, the Google Colab platform could be a good tool to teach physical-chemistry.
Also, help the students develop different skills using collaborative tools, solve chemistry problems
using Python language, and apply mathematical knowledge learned in the first years' disciplines.
Seven Jupyter notebooks were developed and used, via Google Colab, in the physical chemistry course
of the Technological Faculty of the Rio de Janeiro State University to participate in the asynchronous
15 classes and help the students solve the assigned problems. The students' positive feedback pointed
out that this initiative was well accepted and should be continued in the next semesters.

KEYWORDS

Physical Chemistry, Web-Based Learning, Thermodynamics, Equilibrium, Python, Jupyter Notebook

20 Physical-Chemistry classes are known to be difficult and abstract due to the math background
required, and the content addressed. Due to the remote teaching caused by the COVID-19
pandemic¹⁻³, physical chemistry classes proved to be more tricky⁴. The instructors' additional
effort was necessary to guarantee the students' engagement^{5,6,7} and the desired academic
performance.

25 Several physical chemistry problems involve calculating derivatives, integrals, solution of
ordinary differential equations, and fit of mathematical models to a given data, which can be
conducted with software available on the internet. Simple regression models can be performed
in Microsoft Excel or Libre Office Calc, the first choice of most students. However, these

packages are not suitable for advanced calculations and are barely used by Chemists and
30 Chemical Engineering with this finality in their professional activities. For these reasons, the
physical chemistry courses can be the gateway to computational resources that can handle
advanced calculations, data treatment, and analysis.

Python is a simple and easy computational language, containing several scientific libraries like
NumPy⁸, SciPy⁹, Matplotlib¹⁰, and others that enable the users to perform simulations,
35 regression analysis, resolution of ordinary differential equations. The reader interested to know
more about Python capabilities can access the following articles¹¹⁻¹⁵. Additionally, Google's
Colab¹⁶ is a free web platform that enables users to run Python and Tensor Flow calculations
for free, using Google's infrastructure. The service requires no setup to use, providing free
access to computer resources without the necessity for the new user to download, install, and
40 configure Python and their libraries to your personal computer. These facilities help the
instructors implement Jupyter¹⁷ notebooks to write Python scripts in their remote physical
chemistry classes since they demand no previous knowledge, by the students, of Python
installation and setup. Google Colab is an inclusive platform since it requires only a device
with an internet connection, being possible Python coding and executing scripts with
45 smartphones and tablets.

Our objective is to introduce the student to the Python language and Google Colab platform as a
computational tool to solve physical chemistry problems, track, in real-time, the students' progress,
and solve their issues. Also, show students how to apply the knowledge acquired in the introductory
calculus courses and introduction to computation to solve physical chemistry problems with
50 computational tools.

PEDAGOGICAL FRAMEWORK

All scripts were written as Jupyter¹⁷ Notebooks in the iPython¹⁸ format and made available for the
students by the Colab platform, sharing the link to the notebooks in the Google Classroom used for
the discipline. Multiple users can share and use the notebooks simultaneously via Colab. According to
55 the permissions given, multiple users can edit and run the Python script. It is a native collaborative
environment, ideal for the period of remote classes and social distancing. The use of Google Colab

aims to be an additional tool to teach physical chemistry and engage the students in the discipline. Also, help the students to connect interdisciplinary concepts and gain other skills necessary to their professional life^{19,20}.

60 The students were free to save the notebooks to themselves, modify and make comments in the notebooks, which permits tracking the students' activities and their doubts. The notebooks were written to solve problems proposed by the course bibliography²¹⁻²⁴, using the libraries NumPy, Matplotlib, and SciPy. It was assumed no previous knowledge of the Python language by the students. For this reason, the scripts were simple as possible, and all instructions were commented on. It was
65 observed previously that the course modality impacts the students' outcomes²⁵. For this reason, it was chosen as a computational tool that will not add extra difficulty to physical chemistry learning. Since the notebooks are a part of the asynchronous classes, a tutorial video was made available together with the notebooks to explain the code and the method used to solve the proposed exercises. The Markdown²⁶ language was used to write the problems description and explanation of solutions.

70 Further, the students were assigned to academic work that needed the use of computational resources to solve the addressed problems. However, the students were free to choose the computational package to solve the proposed exercises.

STRUCTURE AND CONTENT

75 Google Colab provides a platform to write iPython notebooks containing interactive elements that can be shared by multiple users. The computational environment is ideal for data visualization, scientific calculation, and data processing. Additional texts can be added in a specific cell, using Markdown format, which introduces the problems and explains the addressed solution (Figure 1). The notebooks can be shared between users without any
80 additional repository, providing a useful manner to make available the codes and follow the students' progress. Also, the notebooks are stored directly in the Google Drive user

The screenshot shows a Google Colab notebook titled "Exercícios-Fis-Qui-Gases.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar (Comment, Share, Connect, Editing). The notebook content is as follows:

(1) Exercício 1

Para 1.0000 mol de N_2 a 0.00°C os seguinte volumes são observados em função da pressão:

p/atm	1.000	3.0000	5.000
V/cm ³	22405	7461.4	4473.1

(2)

calcule e faça o gráfico de pV/nT contra p para estes três pontos e calcule R quando p = 0.

(3)

```

1 # Definindo os vetores que serão plotados
2
3 p = np.array([1.000, 3.0000, 5.000])
4 V = np.array([22.405, 7.4614, 4.4731]) # Já alterado para litros
5
6 T = 273.15 # Kelvin
7
8 pV_nT = (p * V) / T
9
10 deg = 1
11 z = np.polyfit(p, pV_nT, deg)
12 y = np.poly1d(z)
13
14
15 print(z) # mostra os coeficientes do polinômio do ajuste na forma y = ax + b
16 print(y)
17

```

(4)

```

[-3.61522973e-05  8.20593996e-02]

```

executed by Vitor Fortes Martins
Monday, October 5, 2020
executed in 0.639s

(5)

Leonardo Baptista
1:16 PM Today

Comment...

Comment Cancel

Figure 1: Example of a Python script written in Google Colab platform. 1) Last time the script was edited, 2) Text cells to write the proposed problems, problems solutions, chemical and mathematical equations in Markdown format, 3) Code cell used to write the Python script, 4) Pop-up balloon showing who and when the code was executed, and 5) The right column shows the comment cells where workmates or instructors and students can share their doubts and issues related to the proposed tasks.

account, without having to download them. Users can add comments to each notebook cell to discuss doubts or issues related to the tasks. This tool eases the communication between workmates, instructors, and students, providing a suitable way of feedback by the students and solution of the related issues. The comments can be marked as "solved", indicating that the question addressed was solved. If necessary, the thread can be reopened for additional corrections, as illustrated in Figure 1, number (5).

The platform has other features that help the instructors to track the students' progress and activities. As shown in Figure 1(number 1), it is possible to see the last time the notebook was edited and who executed the code (number 4). These features are ideal for Web-based learning methods and teaching during pandemic times.

It is possible to install additional libraries that are not Colab by default, using native Linux commands. For example, it is possible to run the Cantera²⁷ suite on the Colab environment using the standard apt-get install instructions to install the package on the platform. The

95 environment supports panda, Jupyter Widgets and can read data from Google Sheets or Google Drive. This platform is a suitable tool for scientific calculation, enabling the users to perform such type of calculation without download, install, and configure any computational package. These characteristics are ideal for beginners in computational sciences or those non-familiars with computational tools. The notebooks are summarized below and made available Git-Hub
100 repository (although the notebooks are in English, the version used in our classes was written in Portuguese).

Notebook 1: Ideal Gas

The first notebook shows how to perform a linear fit from a given data, the procedures to
105 calculate the universal gas constant, the molecular mass of a gas, and how the ideal gas temperature scale can be obtained. Also, graphs were constructed with Matplotlib to show the quality of linear fits performed and introduce the students to graphs construction from experimental data and mathematical models.

Notebook 2: First Law

The second notebook was written to calculate enthalpies change and work of expansion after heating. The students can familiarize themselves with the Markdown format, used to write the problems, equations, and discussion of the solution of the proposed exercises. In this notebook, work is calculated for a reversible expansion of an ideal gas and a van der Waals
115 gas. Finally, the student is asked to estimate the work done by a gas that obeys the virial expansion and the Redlich-Kwong equation of state.

Notebook 3: Entropy

The third notebook introduces non-linear and polynomial fit procedures to obtain the heat
120 capacity as a temperature function. Also, plots of the experimental given data and fitted models are constructed. In a range of temperatures, the numerical integration of entropy is conducted using the SciPy library, and issues related to the entropy integration at low

temperatures are discussed. The aT^3 Debye's Law is introduced as a method to the entropy integral evaluation at low temperatures.

125

Notebook 4: Chemical Equilibrium

This notebook presents how to use the `scipy.optimize.fsolve` to cope with chemical equilibrium problems. This routine finds the root of a function or the root of a linear equations system defined by the user in form $f(x) = 0$. Simple equilibrium problems are addressed, and the equilibrium composition is estimated using the suggested routine. The notebook shows how to write a chemical equilibrium problem in Python language and solve it using Scipy. Scipy is used to find the minimum of the Gibbs free energy that corresponds to the equilibrium condition. The notebook address problems related to temperature and pressure effects on the chemical equilibrium and perform equilibrium calculations, including corrections due to non-ideal behavior on the equilibrium constant.

130

135

Notebook 5: Phase Equilibrium of pure substances

This notebook was used to solve problems involving the phase equilibrium of pure substances. A linear fit procedure was conducted to estimate the water's phase transition enthalpy using the Clausius-Clayperon equation and considering the constant enthalpy of transition in the temperature range. Further, the stability of phases and how the vapor pressure changes with the temperature were discussed.

140

Notebook 6: Real Gases

The scripts written in this notebook shows how to evaluate the coefficient of fugacity of a real gas by numerical integration, calculate the second virial coefficient, and the Boyle's temperatures for a gas. The student is introduced to two different methods to solve integrals numerically: the trapezoidal rule and Simpson's rule. The main idea is to show that both methods are easy to use and the chosen method depends on the problem addressed. The Boyle's temperature is estimated fitting a second-degree polynomial followed by roots finding.

145

150

Since the second-degree equation has two real roots, the choice of the right root is made by evaluating the virial coefficient versus temperature plot, by checking where the second virial data is nearly zero.

155 **Notebook 7: Phase equilibrium of mixtures**

This notebook introduces the student to more complex problems: phase equilibrium involving mixtures. The ideal solution and diluted ideal solution concepts are the subjects of this notebook. Problems involving one or more equilibrium processes are presented, and the methodology to solve each problem is addressed. One problem involves three simultaneous
160 equilibrium, and the proposed script shows how to use Cramer's rule to solve a system of linear equations.

ACTIVITIES

The physical-chemistry course had weekly evaluation tests and one major individual homework about several course topics. The students could choose any software of their liking to solve the
165 personal homework since the activity demands linear and polynomial fits, phase equilibrium plots, and entropy integration. As expected, the first choice of the majority of students was spreadsheets software, like Microsoft Excel. However, due to this type of software's limitations, the students who chose the spreadsheet as their main computational tool partially solved the homework manually or by using a second software. Of the 20 students that attended the physical-chemistry class last term, 55%
170 used partially or totally the Python language at Google Colab platform to solve the assigned problems.

EVALUATION

A survey issued on the final of the term, in December 2020, and answered by 15 students, indicates that the students considered the Colab platform user-friendly and the non-downloading and installation necessity the most likable character. The four main questions about Google Colab and
175 Python scripts are depicted in Figure 2. Nine of fifteen students considered the Python scripts a good supplementary material for the course. Not all students considered the Python scripts useful to learn physical chemistry or solve the course's problems. The complete acceptance of the proposed tool is not expected because many students have difficulties with programming and numerical calculus. For

180

these reasons, some insist on solving complex mathematical problems in spreadsheet software that are more user-friendly and previously known.

185

On the instructor side, the possibility to chat with the students using the comments balloons was useful to correct exercises and the code written. The notebook sharing fastens the communication and feedback between the instructor and students. Regarding the shared notebooks, it is possible to visualize the last person who edited and ran the script, providing a way to follow the students' progress and time spend with the proposed exercises

190

At this point, it is not possible to measure if the online course mediated by Google Colab has comparable or the same outcomes of a face-to-face format, as previously reported for the inorganic chemistry course²⁸. However, we can suggest the Google Colab as a resource to teach physical chemistry through internet means due to the students' positive feedback and simplicity to use and solve problems.

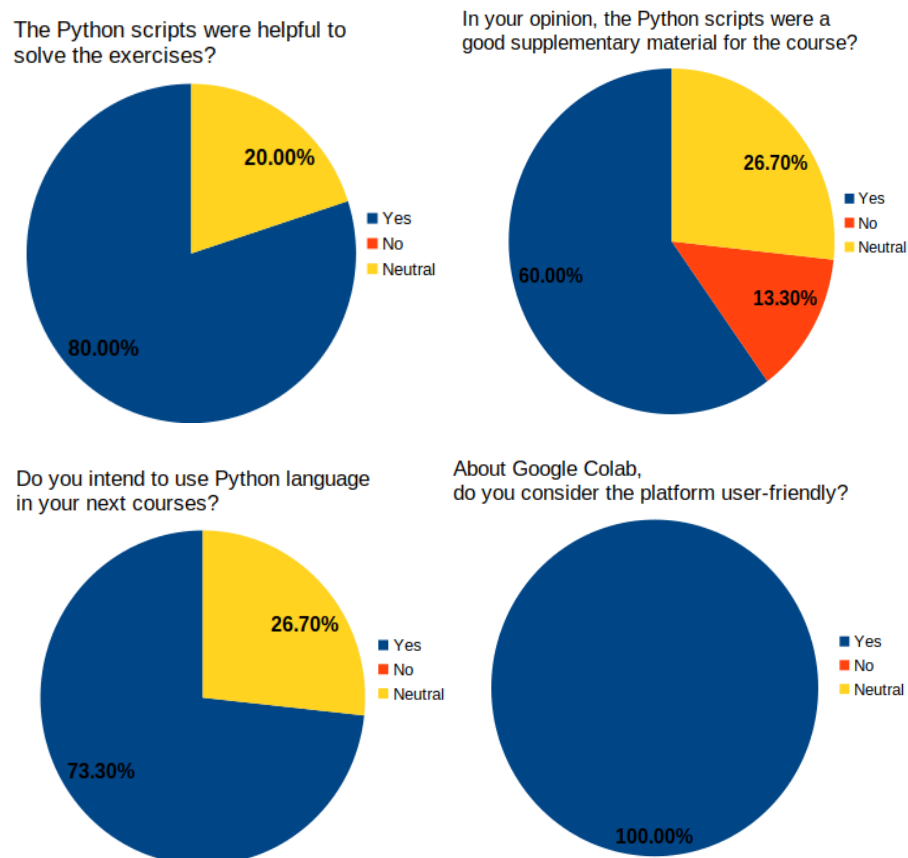


Figure 2. Survey about Google Colab and Python's scripts issued to the attendants of the second-semester physical-chemistry course. Fifteen students answered the survey.

Conclusions

Seven Jupyter notebooks were written and used as supplementary material of the physical chemistry course of the Technological Faculty of the Rio de Janeiro State University and made available by the students via the Google Colab platform. The platform was used as part of the asynchronous classes and chosen because it provides an environment to use Python language without installing, setup, and configuration of packages and libraries. All students considered a user-friendly platform, and the majority has chosen the Python language via Google Colab as their primary computational tool along the course. The platform's multi-user feature eases the communication between instructor and students and facilitates the students' feedback about his assignments. The platform proved to be a good tool for workgroups and assign exercises.

The notebooks were developed using a few capabilities of the Google Colab environment; however, they showed to be very valuable to the remote classes. More advanced features may be explored in the future, like pandas usage, integration with Google Sheets, and Jupyter Widgets, to improve the teaching method and physical chemistry learning.

The second term of 2020 was the first time that this resource was used as part of the physical-chemistry course, and due to the positive feedback, it will be continued in further terms. The platform will possibly become a regular part of the physical-chemistry course after the end of the pandemic.

Supporting Information

All seven Python scripts described in this manuscript are available as Jupyter notebooks in the iPython format at <https://github.com/leobap01/physical-chemistry>. All notebooks can be uploaded directly to the Google Colab website or executed in Jupyter installed locally.

AUTHOR INFORMATION

Corresponding Author

*E-mail: leobap@gmail.com

ACKNOWLEDGMENTS

I am very grateful to all students who help me test and debug the scripts used in our classes and their feedback relative to the Google Colab usage.

REFERENCES

- 225 (1) Soares, R.; De Mello, M. C. S.; Da Silva, C. M.; MacHado, W.; Arbilla, G. Online Chemistry Education Challenges for Rio de Janeiro Students during the Covid-19 Pandemic. *J. Chem. Educ.* **2020**, *97* (9), 3396–3399. <https://doi.org/10.1021/acs.jchemed.0c00775>.
- (2) Hwang, C. S. Using Continuous Student Feedback to Course-Correct during COVID-19 for a Nonmajors Chemistry Course. *J. Chem. Educ.* **2020**, *97* (9), 3400–3405. <https://doi.org/10.1021/acs.jchemed.0c00808>.
- 230 (3) Perets, E. A.; Chabeda, D.; Gong, A. Z.; Huang, X.; Fung, T. S.; Ng, K. Y.; Bathgate, M.; Yan, E. C. Y. Impact of the Emergency Transition to Remote Teaching on Student Engagement in a Non-Stem Undergraduate Chemistry Course in the Time of Covid-19. *J. Chem. Educ.* **2020**, *97* (9), 2439–2447. <https://doi.org/10.1021/acs.jchemed.0c00879>.
- (4) Wild, D. A.; Yeung, A.; Loedolff, M.; Spagnoli, D. Lessons Learned by Converting a First-Year Physical Chemistry Unit into an Online Course in 2 Weeks. *J. Chem. Educ.* **2020**, *97* (9), 2389–2392. <https://doi.org/10.1021/acs.jchemed.0c00678>.
- 235 (5) Lawrie, G. Chemistry Education Research and Practice in Diverse Online Learning Environments: Resilience, Complexity and Opportunity! *Chem. Educ. Res. Pract.* **2021**, *22* (1), 7. <https://doi.org/10.1039/d0rp90013c>.
- (6) Ranga, J. S. Online Engagement of Commuter Students in a General Chemistry Course during COVID-19. *J. Chem. Educ.* **2020**, *97* (9), 2866–2870. <https://doi.org/10.1021/acs.jchemed.0c00633>.
- (7) Emenike, M. E.; Schick, C. P.; Van Duzor, A. G.; Sabella, M. S.; Hendrickson, S. M.; Langdon, L. S. Leveraging Undergraduate Learning Assistants to Engage Students during Remote Instruction: Strategies and Lessons Learned from Four Institutions. *J. Chem. Educ.* **2020**, *97* (9), 2502–2511. <https://doi.org/10.1021/acs.jchemed.0c00779>.
- 245 (8) Numpy <https://numpy.org/> (accessed Jan 8, 2021).
- (9) SciPy.org <https://www.scipy.org/> (accessed Jan 8, 2021).
- (10) Matplotlib: Visualization with Python <https://matplotlib.org/> (accessed Jan 8, 2021).
- 250 (11) Weiss, C. J. Scientific Computing for Chemists: An Undergraduate Course in Simulations, Data Processing, and Visualization. *J. Chem. Educ.* **2017**, *94* (5), 592–597. <https://doi.org/10.1021/acs.jchemed.7b00078>.
- (12) Menke, E. J. Series of Jupyter Notebooks Using Python for an Analytical Chemistry Course. *J. Chem. Educ.* **2020**, *97* (10), 3899–3903. <https://doi.org/10.1021/acs.jchemed.9b01131>.
- 255 (13) Srncic, M. N.; Upadhyay, S.; Madura, J. D. A Python Program for Solving Schrödinger's Equation in Undergraduate Physical Chemistry. *J. Chem. Educ.* **2017**, *94* (6), 813–815. <https://doi.org/10.1021/acs.jchemed.7b00003>.
- (14) Jarvis, R. M.; Broadhurst, D.; Johnson, H.; O'Boyle, N. M.; Goodacre, R. PYCHEM: A Multivariate Analysis Package for Python. *Bioinformatics* **2006**, *22* (20), 2565–2566.
-

-
- 260 <https://doi.org/10.1093/bioinformatics/bt1416>.
- (15) Dahlgren, B. ChemPy: A Package Useful for Chemistry Written in Python Software • Review • Repository • Archive. <https://doi.org/10.21105/joss.00565>.
- (16) Google Colab colab.research.google.com (accessed Jan 8, 2021).
- (17) Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; Ivanov, P.; Avila, D.; Abdalla, S.; Willing, C.; development team, J. Jupyter Notebooks - a Publishing Format for Reproducible Computational Workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*; Loizides, F., Schmidt, B., Eds.; IOS Press: Netherlands, 2016; pp 87–90.
- 265 (18) Pérez, F.; Granger, B. E. {IP}ython: A System for Interactive Scientific Computing. *Comput. Sci. Eng.* **2007**, 9 (3), 21–29. <https://doi.org/10.1109/MCSE.2007.53>.
- 270 (19) Kurniawan, O.; Koh, L. L. A.; Cheng, J. Z. M.; Pee, M. Helping Students Connect Interdisciplinary Concepts and Skills in Physical Chemistry and Introductory Computing: Solving Schrödinger’s Equation for the Hydrogen Atom. *J. Chem. Educ.* **2019**, 96 (10), 2202–2207. <https://doi.org/10.1021/acs.jchemed.9b00068>.
- 275 (20) Hill, M. A.; Overton, T. L.; Thompson, C. D.; Kitson, R. R. A.; Coppo, P. Undergraduate Recognition of Curriculum-Related Skill Development and the Skills Employers Are Seeking. *Chem. Educ. Res. Pract.* **2019**, 20 (1), 68–84. <https://doi.org/10.1039/c8rp00105g>.
- (21) Levine, I. N. *Physical Chemistry*, 6th ed.; McGraw-Hill: New York, 2009.
- (22) Atkins, P.; Paula, J. de. *Físico-Química*, 10^o ed.; LTC: Rio de Janeiro, 2018.
- 280 (23) Nevers, N. de. *Physical and Chemical Equilibrium for Chemical Engineers*, 2nd ed.; John Wiley & Sons, Inc.: Hoboken, 2012.
- (24) Charleston, C. R. M. (College of. *Schaum’s Solved Problems Series - 2000 Solved Problems in Physical-Chemistry*; McGraw-Hill, Inc., 1990.
- (25) Faulconer, E. K.; Griffith, J. C.; Wood, B. L.; Acharyya, S.; Roberts, D. L. A Comparison of Online and Traditional Chemistry Lecture and Lab. *Chem. Educ. Res. Pract.* **2018**, 19 (1), 392–397. <https://doi.org/10.1039/C7RP00173H>.
- 285 (26) Markdown Guide <https://www.markdownguide.org/> (accessed Jan 8, 2021).
- (27) Goodwin, D. G.; Speth, R. L.; Moffat, H. K.; Weber, B. W. Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes.
- 290 <https://doi.org/10.5281/zenodo.1174508> (accessed Jan 8, 2021).
- (28) Nennig, H. T.; Idárraga, K. L.; Salzer, L. D.; Bleske-Rechek, A.; Theisen, R. M. Comparison of Student Attitudes and Performance in an Online and a Face-to-Face Inorganic Chemistry Course. *Chem. Educ. Res. Pract.* **2020**, 21 (1), 168–177. <https://doi.org/10.1039/c9rp00112c>.
-