The Journal of Open Source Software

chemsynthcalc: a software for chemical synthesis calculations and reaction balancing

Egor V. Syrov [1]

1 N. I. Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 02 January 2026
Published: unpublished

Summary

The problem of finding the amounts of reactants needed for a chemical synthesis (and the related problem of balancing reaction equations) is still relevant. In this article, we present `chemsynthcalc` - a Python package and GUI interface for automatic synthesis calculations and reaction balancing. The main advantages of this package over its competitors are operations with non-integer coefficients and atomic amounts, and a fast, rich, and well-documented API. Four linear algebra methods are implemented for performing operations on a chemical reaction matrix. Three of them are adopted from the literature and one of them (combinatorial or exhaustive search method) is original. The stability and performance of the calculations were measured against a large dataset of inorganic synthesis reactions.

Statement of need

The solid-state synthesis of inorganic compounds is a vital field in material science for the production of a variety of materials. The synthesis of photocatalysts, superconductors, ferroelectrics, phosphors, and other materials by calcination of solids, hydrothermal or sol-gel methods requires calculations of the precursor masses before weighing, grinding, and heating.

`chemsynthcalc` was created to help scientists calculate inorganic synthesis. There are three main aspects to this problem. First of all, a single student, scientist, postdoc, etc., works in their lab trying to synthesize a new material. The sample size of potential reactions is quite small in this case, and the scientist can use the old-fashioned pen-and-paper method. But why should one not try to automate this boring task? Secondly, while we are not quite there yet, one can imagine a robotic inorganic synthesis station, like Dr. Cronin devices (Sans et al., 2015). In this case, we need to calculate a large number of reactions fast and precisely, and we really should not hard-code all the masses beforehand. Finally, there are datasets of text-mined inorganic reactions (Kononova et al., 2019). They will surely expand and grow in size. With a sample size of thousands, we need our reaction balancing software to be extremely fast, robust, and flexible enough to balance as many reactions as it can. In this case, we need a free open-source solution that can be embedded in the data processing pipeline.

`chemsynthcalc` addresses all three of those cases. It is simple enough to use by a single scientist who is familiar with using Python packages and fast and robust enough to precisely calculate hundreds and thousands of reactions. There are already a large number of reaction balancing software, why do one need `chemsynthcalc` and why is it better than competitors? Here are some advantages of this package:

- It is completely free and open-source (under MIT license).
- It provides a rich and simple API for its functions.
- It can balance a huge variety of reactions.

- It can deal with formulas with float atom count (like $RbLa_{0.99}Eu_{0.01}Nb_2O_7$), both for molar mass and reaction balancing.
- It supports a huge number of nested parentheses in formulas (up to recursion depth).
- It supports addition notation (like $CuSO_4$*$5H_2O$).
- It is fast thanks to NumPy matrix operations.
- Finally, it is one of the few programs that can directly output precursor masses from a reaction string (in three lines of code!).

Algorithm

The calculation can be algorithmized as following:

1. Selection of synthesis *target* and *target mass*. (which substance will be synthesized and in what quantity).
2. Chemical equation construction (while taking available precursors into account).
3. Chemical equation balancing.
4. Calculation of molar masses of compounds.
5. Calculation of the amount of substance of the target compound *n* (mol) as $n = m/M$, where *m* is target mass (in grams), and *M* is the target molar mass (in g/mol).
6. Broadcasting this amount of substance to other compounds considering their stoichiometry and calculation of their masses as $m = cnM$ (where *c* is a ratio of stoichiometric coefficients of some substance and target substance).

Two most challenging steps here are, of course, the reaction balancing and the calculation of the molar mass. There is a set of algebraic or mathematical balancing methods. They are generally based on creation of a system of linear equations for every atom in the chemical equation and solving this system using different linear algebra techniques. These methods are based on the concept of chemical composition matrix or chemical reaction matrix (Blakley, 1982). For example, the chemical reaction matrix for a simple textbook reaction

$$KMnO_4 + HCl = MnCl_2 + Cl_2 + H_2O + KCl$$

will look like:

$$\begin{array}{c} \\ K \\ Mn \\ O \\ H \\ Cl \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 1 & 2 & 2 & 0 & 1 \end{bmatrix}$$

Where rows represent the atom type, and columns represent the compound. Now, this system of linear equations can be solved using linear algebra techniques. In `chemsynthcalc`, there are four such methods: the `inverse` method based on Thorne paper (Thorne, 2011); the general `pseudoinverse` and `partial pseudoinverse` methods based on Risteski papers (Ice B. Risteski, 2008, 2013; Ice B. Risteski, 2009). Finally, the `combinatorial` method is used to solve Diophantine matrix equation $Ax = By$ by brute force. Despite its limitations, the combinatorial method can achieve some unexpected and interesting results, for example, find a smaller set of coefficients for a well-known reaction. Detailed descriptions of these algorithms are presented in the docs.

There is also a user-friendly full-functional cross-platform GUI version of this package build with Go backend and web frontend, available for Windows, Linux, MacOS, Android and web (Syrov, 2025).

Testing

Beside standard unit tests, it is important to test and benchmark such package against huge variety of real-life examples of reactions. A dataset of text-mined solid-state reactions (Kononova et al., 2019) was used to test its capabilities. The reaction dataset used was a part of the `solid-state_dataset_20200713` file. The original dataset contains more than 30,000 entries. We filter this list of reactions, leaving only valid deduplicated reactions. These do not include reactions that contains non-stoichiometry symbols (like δ), reactions with unknown amounts of substance (like x or y), or reaction with different sets of atoms on right and left sides of the equation.

Thus, the list of 9181 valid reactions was formed. Then automatic balancing and calculations of the masses were performed for each reaction on this list. The results (benched on Ubuntu 24.04, AMD Ryzen 7 5700x, and 64 GB DDR4 RAM with Python 3.13.9, NumPy 2.4.0 and output to a .txt file) are 0.057 ms per formula and 0.49 ms per reaction. There are 69 million reactions in the Reaxys database (Elsevier, 2025), therefore all the reactions in the largest database can be balanced within 9.5 hours on a single consumer-grade PC!

References

Blakley, G. R. (1982). Chemical equation balancing: A general method which is quick, simple, and has unexpected applications. *Journal of Chemical Education*, *59*(9), 728. https://doi.org/10.1021/ed059p728

Elsevier. (2025). *Reaxys: An expert-curated chemistry database*. https://www.elsevier.com/solutions/reaxys

Kononova, O., Huo, H., He, T., Rong, Z., Botari, T., Sun, W., Tshitoyan, V., & Ceder, G. (2019). Text-mined dataset of inorganic materials synthesis recipes. *Scientific Data*, *6*(1), 203. https://doi.org/10.1038/s41597-019-0224-1

Risteski, Ice B. (2008). A new pseudoinverse matrix method for balancing chemical equations and their stability. *J. Korean Chem. Soc.*, *52*(3), 223–238.

Risteski, Ice B. (2009). A new singular matrix method for balancing chemical equations and their stability. *Journal of the Chinese Chemical Society*, *56*(1), 65–79. https://doi.org/https://doi.org/10.1002/jccs.200900011

Risteski, Ice B. (2013). A new topology of solutions of chemical equations. *J. Korean Chem. Soc.*, *57*(2), 176–203.

Sans, V., Porwol, L., Dragone, V., & Cronin, L. (2015). A self optimizing synthetic organic reactor system using real-time in-line NMR spectroscopy. *Chem. Sci.*, *6*, 1258–1264. https://doi.org/10.1039/C4SC03075C

Syrov, E. V. (2025). Chemsynthcalc-GUI: A web-based desktop GUI for the chemical synthesis calculator. In *GitHub repository*. GitHub. https://github.com/Syrov-Egor/chemsynthcalc-GUI

Thorne, L. R. (2011). *An innovative approach to balancing chemical-reaction equations: A simplified matrix-inversion technique for determining the matrix null space*. https://doi.org/10.48550/ARXIV.1110.4321