AuRaCh: A tool for simulating radiolytic chemistry during liquid phase transmission electron microscopy

An Introduction

Neda Zargar Talebi · Birk Fritsch

June 13, 2022

Contents

1 Introduction

This document describes the operation and functionality of the Graphical User Interface (GUI) for AuRaCh, an **Au**tomated **Ra**diation **Ch**emistry simulation tool.

If you find our tool useful for your research please cite our related publication:

Fritsch *et al.*, Radiolysis-Driven Evolution of Gold Nanostructures – Model Verification by Scale Bridging *In Situ* Liquid-Phase Transmission Electron Microscopy and X-Ray Diffraction, *Advanced Science*, 2022, `https://doi.org/10.1002/advs.202202803`DOI:10.1002/advs.2

Naturally, some parts of this manual directly stem from this source. We encourage the reader to refer to this publication for a description of the working principle.

2 Package Dependencies

● python 3.7+ ● scipy 1.3.1+ ● numpy 1.19+ ● pandas 1.1+ ● matplotlib 3.3+ ● pyqt 5.9.2+

3 Description of the input files

For operating the GUI, two input text files need to be provided. First, a set of chemical reactions is required, second the initial concentrations, and, if applicable, generation bakues (g-values) are needed. The syntax of these two files is explained in the following.

3.1 Reaction file

The reaction file needs to be stored as '`.txt`' file. It consists of three columns separated by a ';'. The left column contains the chemical reaction, the middle column denotes the reaction constant, and the right column allows for an optional comment (i. e. the literature source or other remarks).

```
10  eh- + H+              --> H                      ; 2.3e10    ; Schneider
11  OH + OH-              --> O- +  H2O              ; 1.3e10    ; Schneider
12  O- + H2O             --> OH + OH-              ; 1.04e8    ; Schneider
13  OH                   --> O- + H+              ; 1.259e-1  ; Schneider
14  O- + H+              --> OH                    ; 1e11      ; Schneider
15  HO2                  --> O2- + H+              ; 1.346e6   ; Schneider
16  O2- + H+             --> HO2                   ; 5e10      ; Schneider
17  HO2 + OH-            --> O2- + H2O             ; 5e10      ; Schneider
18  O2- + H2O            --> HO2 + OH-             ; 18.62     ; Schneider
19  eh- + OH             --> OH-                   ; 3e10      ; Schneider
20  eh- + H2O2           --> OH + OH-              ; 1.1e10    ; Schneider
21  eh- + O2- + H2O      --> HO2- + OH-            ; 1.3e10    ; Schneider
22  eh- + HO2            --> HO2-                  ; 2e10      ; Schneider
23  eh- + O2             --> O2-                   ; 1.9e10    ; Schneider
24  2 eh- + 2 H2O        --> H2 + 2 OH-            ; 5.5e9     ; Schneider
25  eh- + H + H2O        --> H2 + OH-              ; 2.5e10    ; Schneider
```

Figure 1: Reaction file layout.

Syntax

Figure 1 provides an example of 15 reactions.

- Note, that **no headings** are given in the text file. The first line starts directly with a reaction.

- **Do not leave blank lines** in between the reaction set.

- Left column – Reaction layout:

  - Educts and products are separated by ' `-->` '. Take care to **include the spaces** so that the code recognizes it as an operator.

  - Reactants are separated by ' `+` '. Take care to **include the spaces** so that the code recognizes it as an operator.

  - **Stoichiometric factors** are written in front of the reactant **separated by a space**, e.g. `2 H`.

  - You can use whatever name you want for your agents, as long as you use **exactly the same name every time**. This, in turn, also means that **every name must be unique**. Note, that `H*` and `H` would be two different species! **Pay extra attention when copying content from another source**, e.g. a document or a website. Only because symbols look similar, they are not necessarily the same ASCII symbol. A major issue is '-' and '–'.

  - WARNING: While the kinetic simulation itself does not require meaningful names, the error handling (partially) requires a chemistry-like notation.

Hence, we suggest using the following convention that leans on the usual nomencalture for chemical formulae:

* Every **element** starts with a captial letter, e.g. `H`, `Au`, or `Cl`.

* **Indices** denoting the amount of an element in a chemical formula are written as arabic numbers in-line (e.g. `H2O2`, `O2` for H_2O_2, and O_2).

* **Charges** are written at the end of a sum formula. In case of a single charge, a sole '+' or '-' suffices. If more charges are present, this is indicated by a '^' followed by an arabic number (e.g. `O-`, `H+`, `Au2Cl6^2-` for OH^-, H^+, and $Au_2Cl_6^{2-}$).

* **Radicals** are not marked explicitly (write just `H` instead of `H*` or similar).

* Parentheses indicating oxidation states are fine (e.g. `Au(III)Cl-`).

* The **solvated electron** is written as '`eh-`'. It will be regarded for charge, but not for mass conservation checks (no solvation shell is taken into account).

* Other uses of () or similar symbols ([],{}, etc.) are not to be used, as notations of complexes or more complicated structures are not (yet) implemented.

- Middle column – Kinetic constant:

  - This column stores the **rate constant** in units of $\mathbf{mol^{-n+1}\,dm^{3(n-1)}\,s^{-1}}$ with n as reaction order.

  - However, **do not provide a unit**. Instead, just type an integer or floating point number (e.g. `20`, `0.13`).

  - You can abbreviate numbers with a large amount of digits by using **scientific notation** *via* an e, e.g. `5e9` $(= 5 \cdot 10^9)$ for 5000000000, or `1.3e-10` for 0.00000000013.

  - Use only '.' as decimal spacer and **do not type any spacer symbols** such as ' ' or ','.

  - TIP: By setting the rate constant to `0` (zero), a reaction can be disabled.

  - However, if the rate constant left from the decimal point is longer than one digit, it must not start with zero. This means that `00`, `01`, or `001` will not work, while `0.0`, `0.01` or `0.001` are unproblematic.

- Right column – comment:

  - **This column does not include any code-relevant information**, so in principle you can use it as you wish.

– Hence, you can leave it empty.

– However, we do not provide any warranty that some non-alphanumeric symbols used here may still cause bugs. Thus, use it wisely.

Examples

These would be valid lines in the reaction file:

- `eh- + eh- + H2O + H2O --> H2 + OH- + OH- ; 0.00055 ;`

- `eh- + eh- + H2O + H2O --> H2 + OH- + OH- ; 5500000000 ; a comment`

- `2 eh- + 2 H2O --> H2 + 2 OH- ; 5.5e9 ; much abbreviation, such wow`

- `H+ + eh- --> H ; 0 ; this reaction is disabled`

The following examples will cause **errors or unexpected behavior**. Keep in mind that this list is incomplete, as the possibilities of creating bugs are legion:

- `2 eh-+ 2 H2O --> H2 + 2 OH- ; 5.5e9 ; comment`

  The space between `eh-` and `+` is missing.

- `2 eh- +2 H2O --> H2 + 2 OH- ; 5.5e9 ; comment`

  The space between `2` and `+` is missing.

- `2 eh- + 2H2O --> H2 + 2 OH- ; 5.5e9 ; comment`

  This line would actually run, but `2H2O` would be assumed to be a species.

- `2 eh- + 2 H2O --> H2 + 2 OH- ; 5.5e9  comment`

- `2 eh- + 2 H2O --> H2 + 2 OH- ; 5.5e9M/s ; comment`

  In both cases, the text after the number will confuse the program, causing a warning and setting the rate constant to 0 (thus disabling the reaction). Ensure that the semicolon is placed behind the rate constant before writing additional text. Whitespace is fine.

- `2 eh- + 2 H2O --> H2 + 2 OH ; ; comment`

  No reaction rate is given. The programm will set it to zero.

- `2 eh- + 2 H2O --> H2 + 2 OH- ; 5 500 000 000 ; comment`

- `2 eh- + 2 H2O --> H2 + 2 OH- ; 5,500,000,000 ; comment`

- `2 eh- + 2 H2O --> H2 + 2 OH- ; 5,500,000,000.0 ;` `comment`

  Due to the decimal separators, the rate constant will not be understood. The programm will set it to zero and raise a warning.

- `2 eh- + 2 H2O --> H2 + 2 OH- ; 05 ;` `comment`

- `2 eh- + 2 H2O --> H2 + 2 OH- ; 00 ;` `comment`

- `2 eh- + 2 H2O --> H2 + 2 OH- ; 00.1 ;` `comment`

  The rate constant part left from the decimal point must not start with a zero. A single `0` as in `0.1` or `0.0002` is fine.

3.2 Concentration and g-value file

The second file is related to the initial concentrations and g-values of the species. It is also stored as '`.txt`' file. For convenience, only species where the initial concentration and/or the g-values are non-zero must be listed.

The programm uses this file as a **look-up table** for unique specie names as written in the reaction file (see above). Thus, **ensure identical notation** in both files. **If a species is not found in th concentration and g-value file, these values will be set to zero**. Likewise, if a species is denoted in the concentration and g-value file but not in the reaction set, this will have no influence on the simulation.

Syntax

An exemplary file is shown in Figure 2.

Figure 2: Concentration and g-value file layout

The file consists of three columns separated by tabstops. In contrast to the reaction file, the first line is a heading line that can be used for notations. In the example files, the

units of the second and third columns are given. Do not provide additional comments in this file, nor use blank lines in between. All restrictions to numbers given above are valid here, as well (no double-0, no units, etc., scientific notation works accordingly)

The first column directly starts with the species name. The second column denotes the initial concentration in M. The third column provides g-values in molecules/100 eV. Note that this can be negative.

4 GUI Window

The first window appearing after running the GUI.py is called "Settings" and looks as shown below:
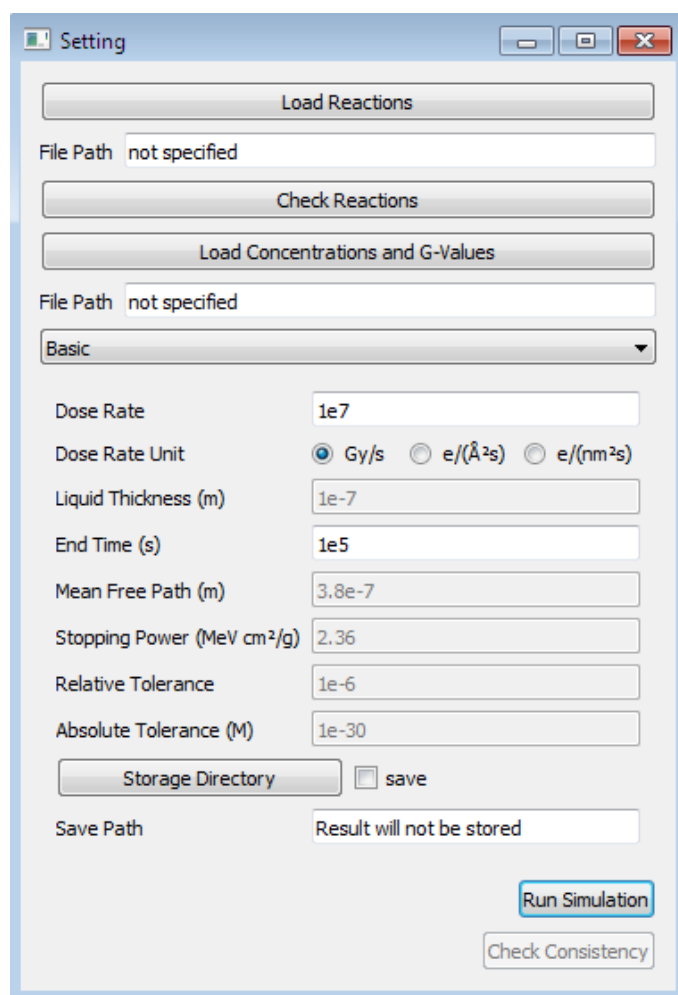
Figure 3: Settings window

By clicking "Load Reactions" button the user can select and upload the reaction set as ".txt" file. Its layout is specified in figure 2. As a result, the path containing the selected file will be appeared in the textbox "File Path". In order to check for errors regarding charge and mass balance of the individual reactions in the uploaded file, one can click on the "Check Reactions" button. If there is no error in the file, the following info message will be displayed.
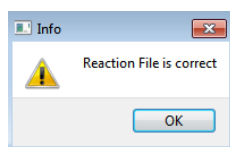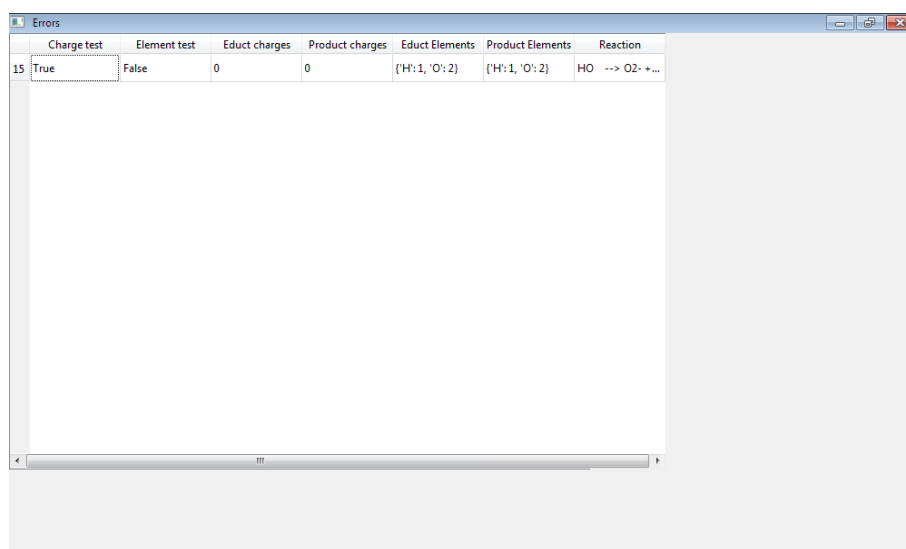
Figure 4: Message shown when the uploaded file is correct

Otherwise, a new window called "Errors" containing a table of faulty reactions will be opened. (see figure 4)

Figure 5: Table shown if the uploaded file contains faulty reactions

- Here, every row index references the respective line in the reaction file.

- The "Charge test" and "Element test" columns summarize whether the charge or mass balance is correct ("True") or not ("False).

- "Educt charges" and "Product charges" list the respective sum of charges on the educt/product site of the reaction.

- "Educt Elements" and "Product elements" list the respective elements.

- "Reactions" displays the original reaction, as provided by the Reaction file.

In the next step one needs to upload concentrations and g-values file in ".txt" layout using the corresponding button. Again, the required syntax of the file is shown inFigure 2. The file path of the uploaded file will be shown in the corresponding text field.

5 Setting Types

There are two options that a user can select from to determine the setting type, using the drop down list shown below:

Figure 6: Setting types: Basic/Advanced

Depending on which option is selected, different input values are required:

- Dose rate refers to the quantity of radiation absorbed per unit of time.

- Duration accounts for the time interval simulated.

- The inelastic mean free path is the average distance traveled by the radiator between inelastic interactions with the medium.

- The stopping power expresses the rate of energy loss of the (charged) radiation of the medium traversed.

- relative and absolute tolerances are parameters defining the accuracy of the simulation and can heavily affect the runtime. They are passed to the ODE solver `LSODA`. See `https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.LSODA.html` for details.

- Liquid density refers to the mass of the medium irradiated within a unit volume.

5.1 Basic settings window

In case of selecting "Basic", one can enter values for Dose Rate and Duration but other fields are deactivated.One can only enter values in the activated textbox.

Figure 7: Basic settings window

"Dose Rate Unit" can be selected by checking the corresponding checkbox. The default unit is 'Gy/s'. In case of selecting e/($Å^2$s) or e/(nm^{2}s), the "Liquid Thickness" field will become editable.
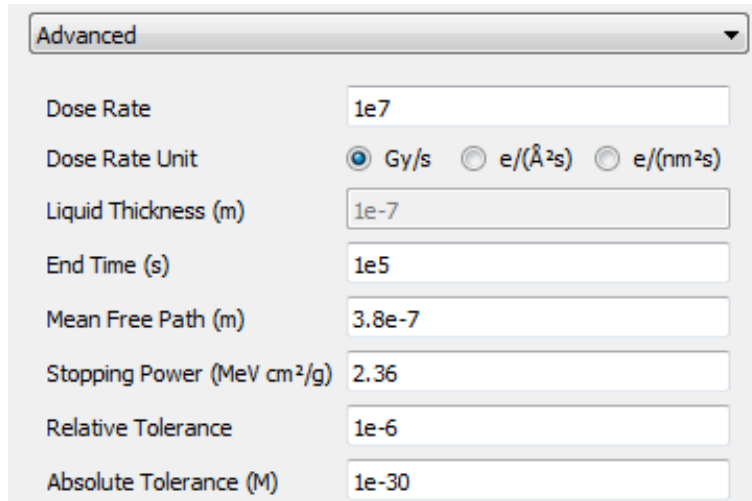
Figure 8: Selecting Dose Rate

5.2 Advanced setting window

If the "Advanced" option is chosen, the settings window will look like this:

Figure 9: Advanced settings window

It is worth noting that only numeric values are accepted for each field. In case of entering text, there will be a warning message. For example:
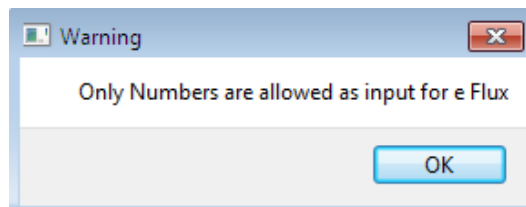

Figure 10: Warning Message in case of entering text as value for numeric fields

6 Simulation

6.1 Saving the simulation result

The user can select a storage directory by clicking on the "Storage Directory" button.

Figure 11: User can select storage directory of the result

The simulation result can be stored as a ".csv" table. Its storage directory can be provided via the button displayed in figure 11. By default, the result will be stored as "Simulation result.csv". By selecting the name of csv data include the ".csv" in it.
If saving the simulation outcome is not desired, the feature can be disabled by unticking the "save" tickbox.
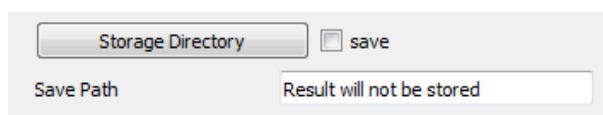

Figure 12: If the 'save' checkbox is not ticked, the numeric result will not be stored.

6.2 Simulation result visualization

To start simulating the kinetic model, "Run Simulation" must be clicked. In case of missing input files, a respective message is displayed.
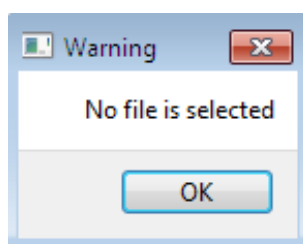

Figure 13: Warning shown when clicking Run button without having selected the required files

When simulation terminates, the output figure will be displayed in a separate window.As an example
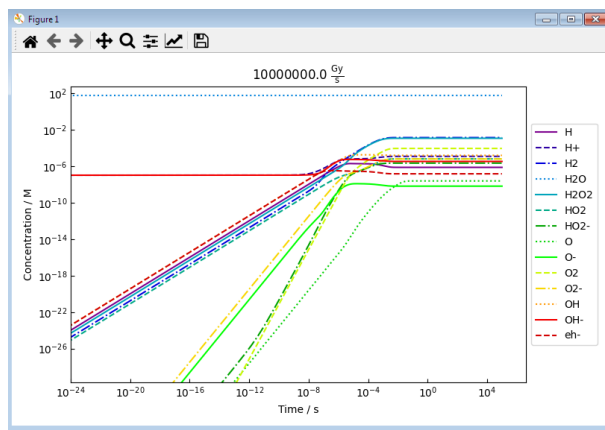

Figure 14: Sample result of simulation

On the top left corner, a navigation toolbar is visible, including numerous editing, zooming and saving options. A detailed documentation can be found at:
`https://matplotlib.org/3.2.2/users/navigation_toolbar.html`

6.3 Simulation result verification

Based on the generated result by simulation it is possible to verify that the law of conservation of mass is obeyed during the simulation via the associated button.

The option again relies on the usage of correct chemical formulate. Besides a backup check for frauds in the reactions, this option allows for checking for consistent G-Values.

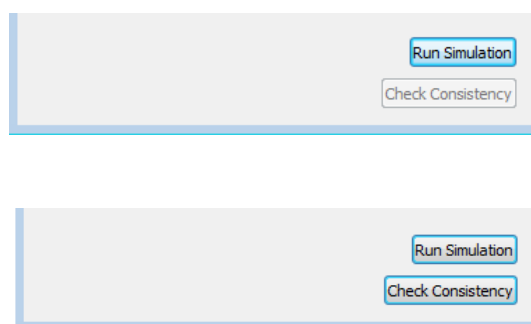This Button is disabled by default and can only be used when simulation is finished.

Figure 15: "Check Consistency" button which becomes active when simulation is finished