Data Article

Python script for homogeneous aqueous chemical reaction analysis and associated data related to radiolysis simulations

Peter Doyle\textsuperscript{a,*}, David Bartels\textsuperscript{b}

\textsuperscript{a} University of Tennessee, Knoxville
\textsuperscript{b} University of Notre Dame

\textbf{A R T I C L E    I N F O}

\textbf{Article history:}
Received 22 April 2020
Accepted 13 May 2020
Available online 18 May 2020

\textbf{Keywords:}
Radiolysis
Homogeneous aqueous reaction
AECL model

\textbf{A B S T R A C T}

A script was developed to perform homogenous radiolysis calculations. It was used specifically to calculate radiolysis products under various neutron and gamma flux environments [1]. The routine may be used to calculate a single radiolysis condition, multiple independent conditions, or multiple conditions computed in series (the final concentration set of run i is the initial concentration of run i+1). While designed for radiolysis of water, the routine is easily adapted to a variety of aqueous reaction systems and may even be altered with minimal effort for more general homogenous chemical analysis. In the present article, the Python routine is explained along with various outputs and inputs. It and the relevant input and output text files are included as supplementary materials. They are the raw data used for calculation of figures in the associated journal article.

© 2020 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)

* Corresponding author.
E-mail address:pdoyle3@vols.utk.edu (P. Doyle).

https://doi.org/10.1016/j.dib.2020.105734
2352-3409/© 2020 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Specifications Table

| Subject                  | Analytical Chemistry          |
|--------------------------|-------------------------------|
| Specific subject area    | Chemical Reactions in Water   |
| Type of data             | 1 python script (text file with .py extension) |
| How data were acquired   | Relevant text files were used inputs to .py script run in Spyder 3.3.3 which generated the remaining text files |
| Data format              | All data is raw.              |
|                          | Output files are labelled to contain a parent name with either '_ConcDelta.txt', '_Results.txt', or '_Assembly.txt' suffixes. |
|                          | Input files are the parent name with a '.txt' extension. |
| Parameters for data collection | Data is obtained from the AECL model [2] for rate constants and reactions. |
|                          | An input file (similar to one in the repository) is needed to give parameters of the reactions, initial concentrations, G-values, simulation time, and energy deposition from radiation. Specifics are given in the description section. |
| Description of data collection | Most inputs are listed in a single file. Within the routine, input parent names are given as a list and certain solver-related parameters are set. |
| Data source location     | Institution: University of Tennessee, Knoxville |
|                          | City/Town/Region: Knoxville, TN |
|                          | Country: United States        |
| Data accessibility       | Repository name: Tennessee Research and Creative Exchange (TRACE) |
|                          | Direct URL to data: https://trace.tennessee.edu/utk_nuclpubs/5/ |
| Related research article | Peter J. Doyle, Kaichao Sun, Lance Snead, Yutai Katoh, David Bartels, Steven Zinkle, Stephen S. Raiman |
|                          | The Effects of Neutron and Ionizing Irradiation on the Aqueous Corrosion of SiC |
|                          | Journal of Nuclear Materials |
|                          | doi:10.1016/j.jnucmat.2020.152190 |

Value of the Data

- The routine has significant potential value in 2 ways: 1) as a rapid method for general radiolysis product estimation with significant control over the solution method and parameters, and 2) with minimal editing effort, for a swath of reactions in homogeneous aqueous solution. While it is formatted for use in radiolysis, the code may be easily altered to more general chemical systems where homogenous reactions must be computed. Individual data sets are useful as respects the associated article in probing the data behind figures 4–6 of that work in understanding equilibration times and decay rates of products following removal of radiation.
- Nuclear engineers and material scientists evaluating the impact of radiolysis products on corrosion, specifically have the greatest benefit from the exact data presented. However, the routine is versatile enough to find potential value in subfields of chemistry.
- This routine allows prediction of concentration profiles over a large swath of potential inputs and thus has significant experimental planning possibilities. The specific data is useful in designing future irradiation experiments to investigate the effects of changing radiation energy deposition rates on corrosion, specifically, which may affect life expectancy of materials in nuclear reactors.

1. Data

The input files and routine are described in detail in the next section. Broadly, the input files included in the repository are consistent with the AECL model’s reactions [2], with the exception of reaction 20, whose rate constant was taken from [3]. The input file corresponding to
Tables 3 and 4 in the associated article is the “Radiolysis-.decomp1.txt” where the ellipsis is either “Core”, “Above”, or “Decay” corresponding to the three major regions of the water loop. Other files indicate the effect of changing R21, corresponding to the data presented in Figure 6 of the parent article [1]. Data from “decomp1” files models R21 as written, and “decomp2” models H2O2 as directly decomposing to 2 OH radicals.

Output files are also described in the experimental section and contain the data presented in the various figures of the associate article [1]. The “_Results.txt” files contain the most pertinent information. These contain headers describing simulation conditions, include input G-values, calculated production rates from radiolysis, a notice that G-values of water and H radical have been adjusted to ensure mass balance, and a further notice if major errors were identified. The bulk of the file is a time series giving all concentrations (in M units) at each time step (logarithmically distributed with time) where the first entry is the initial concentrations. Three additional columns are included giving the material balances at each readout. At the end of the output, the final fluxes of each species are given (at t_final) with the difference in the concentration over the course of the simulation. Following this is a series of notices with any errors and description of the simulation conditions. The final list identifies each reaction used, its rate constant, and its initial and final rate.

Two other files are output for supplementary information. “_ConcDelta.txt” provides a time series of concentration fluxes (in M/s) at each output time in as the “_Results.txt” file. “_Assembly.txt”, assembles the reactions into differential form and lists all simulation equations and constants needed to also run the file in a separate differential equations solver.

2. Experimental Design, Materials, and Methods

Input files to the routine are required to be formatted specifically as follows and as demonstrated in the data files. All entries in a row are separated only by a white space. The first line contains a complete list of all chemical species to model. Subscripts should be written as a number next to the species (e.g., H2O2 is represented as H2O2) and charges may be represented as either “-“ and “+” symbols or with “m” and “p”. Representation of differing chemical systems can be done by adding and removing components from this list. However, it should be noted that mass balances only account for H and O. The program will allow different elements to be run, but will potentially generate error messages unless the code is changed to include accounting for these elements. For the “_Assembly.txt” file, which gathers the reactions into differential form all “-” and “+” signs are converted into “m” and “p”, for clarity. Both H2O and H must be in this first line for mass balance on the G-values or errors will be generated. If H is not relevant, it can be included as a dummy variable with all its inputs set to 0.

The next two lines contain as the first entry the energy deposition rate from each radiation source in Gy/s (exactly two sources are programmed into the routine). The rest of each line contains a list of numbers each corresponding to the G-value (in #/100 eV) of the chemical species listed in the first line, in order of their listing. In the fourth line, the first entry is the total simulation time (in seconds). This will be the last value in a logarithmically distributed time series beginning at 0. The rest of line 4 is a starting concentration series. Reactions are described by the remainder of the lines in the file, with each line corresponding to a single chemical reaction. These continue until all reactions are described (no limit). The first entry in each line is a number indicating the rate constant (M^-nS^{-1}) where n is the reaction order minus 1. Reaction components are described by the remainder of the line. Negative values indicate reactants, and positive values indicate products. The reaction order is taken to be elementary. Non-elementary reactions can be described by explicitly defining the CoeffsRate array (in the routine itself) with respect to each component. CoeffsRate has the same shape and meaning as the coefficient matrix defined by the reaction section of the input file (i.e., each column corresponds to a component (in order of components) and each row corresponds to each reaction).
Additional variables are described within the routine itself. “autobalance” is a Boolean (default “True”) that, when true, triggers the program to call ObtainComponents() which identifies all compounds, and populates mass and charge balance arrays. If “autobalance” is false, each of these arrays must be manually supplied. The redox balance arrays must be manually supplied regardless of autobalance. In their default form, they correspond to the reaction system in the corresponding paper. Buffer is an array (in this work it is zeros) that is only 1 for concentrations that are buffered to prevent change, and 0 otherwise. If other components are included or some are removed, these three arrays must be changed to the correct size and value, as relevant.

As this routine uses the SciPy differential solver, solve_ivp, the variables “abtol” and “reltol” tell the solver which error tolerances to set. As provided in the repository, they are set at values which provided the best stability in the current outputs. “tStart” is a power of ten that serves as the starting point of the logarithmic time distribution. As this value is really the time defined by the initial supplied concentrations, this value only serves to change the time interval of outputs, and not the actual solutions. The number of output points is given by “number”. “name” is a list of all parent input file names (no extensions). Where a common suffix is used, “suffix” is a string that is added to each parent name, that facilitates running large numbers of files. Finally, “sequentialfiles” is a Boolean (default “True”). If True, rather than using the given initial concentrations, after the first file has run, it imports the solution concentrations from the last time step of the previous problem set and uses these as the initial concentrations. Otherwise, provided concentrations from the input files are used. In the current dataset, “sequentialfiles” is True for all simulations with all simulations run in the order “Radiolysis-Core....txt”, “Radiolysis-Above....txt”, and “Radiolysis-Decay....txt”.

After the input file is read, a variety of arrays and variables are populated. These are documented in the routine and serve only to set up the functions for solution. As mentioned above, one of these, CoeffsRate, defines the order of reaction with respect to each component. Specifically, each element is the exponent on each component’s concentration for each reaction. Assigning non-elementary rates is thus accomplished by assigning relevant elements of CoeffsRate specific values between lines 344 and 353. The reaction will still cause elementary depletion and production of components per the input file.

Beyond such details, reactions are screened for charge, redox, and mass balance. If imbalance is found a major error is generated. Each reaction is then screened for water as a reactant. Except for the decomposition of water to H⁺ and OH⁻, water is removed as a reactant since it is the solvent and does not affect other reaction rates. The routine is not formatted for multiple water decomposition reactions (except implicitly through the use of G-values), which are not expected to be required. This does not affect the concentration flux of water (i.e, water is still consumed by reactions for which it is a reactant, it only does not contribute to the rate).

Production rates of each compound from radiation are calculated assuming all species come from water decomposition, which is the negative of the total oxygen formation rate (determined using G-values, water density, and the dose rate). Given that the number of time steps can be large, to ensure mass balance from G-value reactions, the production rate of the H radical is set as the difference between the hydrogen consumption rate from radiation (water production rate) and the production of radiolytic species. As a notice statement, the difference in the production of H as-input versus as-used is listed in the results file.

The differential equation system is then solved using the scipy.integrate.solve_ivp() function, which is a freely available routine whose documentation describes it sufficiently [4]. Concentration differences and final reaction rates, are calculated for output along with mass, charge, and redox balances. The final lines of the program create appropriate notices and output the previously described data to the relevant files.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
Acknowledgments

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow other to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

References

[1] P.J. Doyle, K. Sun, LL. Snead, Y. Katoh, D.M. Bartels, S.J. Zinkle, S.S. Raiman, The Effects of Neutron Irradiation on the Corrosion of SiC, Corrosion Science Accepted (2019).
[2] A.J. Elliot, D.M. Bartels, The reaction set, rate constants and g-values for the simulation of the radiolysis of light water over the range 20 deg to 350 deg C based on information available in 2008, Canada, 2009, p. 162.
[3] D. Swiatla-Wojcik, Hybrid method for numerical modelling of LWR coolant chemistry, Radiation Physics and Chemistry 127 (2016) 236-242.
[4] TheScipyCommunity, scipy.integrate.solve_ivp, 2019. https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html. (Accessed August 29 2019).