ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume 28, Number 3, 2025, 286–298
doi: 10.59277/ROMJIST.2025.3.04

Effect of Random Variation of the Initial Values of Ordinary Differential Equations and its Web-based Application to Chemical Kinetics

Miguel CASQUILHO[1,*], Rui GALHANO[1], and Jorge BUESCU[2]

[1]Department of Chemical Engineering, Instituto Superior Técnico, University of Lisbon, and CERENA, Centro de Recursos Naturais e Ambiente ("Centre for Natural Resources and the Environment"), Ave. Rovisco Pais, IST, 1049-001 Lisboa, Portugal

[2]Department of Mathematics, Faculdade de Ciências, University of Lisbon, and CMAFCIO, Centro de Matemática Aplicada e Fundamentos, Computação e Investigação Operacional ("Centre for Applied Mathematics and Fundamentals, Computation and Operational Research"), Campo Grande, 1749-016 Lisboa, Portugal

Email: mcasquilho*, rui.galhano@tecnico.ulisboa.pt, jsbuescu@fc.ul.pt

* Corresponding author

Abstract. In this article, the initial values of ordinary differential equations (ODEs) are computationally subjected to random variation in order to assess its effect on the final values, through a Monte Carlo procedure, with a chemical kinetics problem as the mathematical illustration. As we use and uphold the Web as a computing medium, the results can be verified on constructed, public web pages. A random variation in the reactants' composition raises undesired variation, which we observe, in the product composition. In a preliminary part of the article: we overview the determination of ODEs' parameters from data, leading, with Python, to favor its 'minimize' function, against 'curve_fit'; and we use some reactions to find their rate constants, namely, the frequent $\mathrm{A} \to \mathrm{B} \to \mathrm{C}$ (or "ABC") reaction, as the one to produce furfural. In the main part of the article, we assess the variation in the final values, the product composition, in the ABC reaction using Monte Carlo simulation, by repeatedly solving its ODEs, through numerical integration to keep generality, the initial values being assumed Gaussian. The final values are simulated on the web pages for exploration, showing the Web as a medium for scientific computing and use in publications, also easing the academia-industry connection.

Key-words: Chemical kinetics; Monte Carlo simulation; ordinary differential equations; random initial values; scientific computing; web-based application.

1. Introduction

In solving differential equations, a random variation in their initial values induces variation on the successively computed and, thus, final values.[1] The mathematical problem addressed will be illustrated with an application in chemical reaction kinetics, the branch dealing with the speed or "rate" of reactions, using the Monte Carlo methodology. A simultaneous, clear goal in the study is, through results verifiable in web pages, to show the pertinence of web-based computing, which is rarely present in scientific publications and even in the university context.

In many production cases, such as in the broad reaction class adopted for illustration, the final values ought to match a target, and are thus the critical results, randomness making those values unpredictable. In the case of chemical reactions, typically described by a system of ODEs, an uncertainty in the reactants' composition ("initial values") inevitably causes deviations on the product's composition (final values). In an industrial context, variations are more liable than in the laboratory, as the reactants' composition can undergo incidental, perhaps small, random variations from their specifications. These variations, to be assessed computationally in this article, should desirably have a minor impact on the objective of the reaction, which is the specified composition. The variations in the final composition are unpredictable because the values in the reactants' composition not only are unknown but also change from case to case, at random. So, our two quantitative tasks are to create the basic uncertainty, and observe its effect.

As regards variations, deriving an analytical relation between the initial and final variations is unviable for ODEs resolved numerically. The problem, thus, fits the Monte Carlo methodology, with reliable results if a reasonable random behavior is assigned to the initial values.

The behavior of the initial values will be given by a set of Gaussian distributions centered on their correct values. For each variable in the ODEs (vs. time), its correct initial value will act as the mean, μ, of its Gaussian, and its variability will be assigned to the standard deviation, σ. The set of σ values will, therefore, be the problem variables to be explored.

Questions arise in handling the σ values, since the initial values, the reactants' composition, can be zero or near, affecting the Gaussian assumption, and may be quite diverse, e.g., 0.1, 20, orders of magnitude. This is discussed later, leading to shift from σ to coefficients of variation.

Regarding other approaches to this article in the literature, both the web-based environment endorsed and the methodological, mathematical problem addressed seem unreported, on chemical reactions or not, but some example references follow (chronologically). Higham [1] relates chemical reaction rates to their ODEs, giving Matlab scripts. Kuntsche et al. [2] give a website for modeling, allowing no execution, such as the hard Belousov-Zhabotinsky (MOSAIC [3]) reaction. Pozna and Precup [4] give results on the observation process modeling, with a case study. Liang et al. [5] present a kinetic study with ODEs similar to those of furfural. Ye et al. [6] list progress in furfural production, due to its broad application prospects. Şerban et al. [7] mention useful websites that, however, offer only programs to install. Misra et al. [8] say there are very few tools to test web-based applications, but their website gives no suggestion, and, on trying a proposed set of data, gave an error. Russo et al. [9] present an unrelated "kinetic Monte Carlo". Bogdos et al. [10] mention a web-based application, supplying only a repository. Abramov et al. [11] distinguish the dynamics of data from that of official statistics. Eşsiz et al. [12] devise deep-learning models to predict vitamin D status. Yan et al. [13] propose a semi-supervised ensemble fuzzy system dealing with missing values. Finally, Helo et al. [14] refer the impact of technology on supply chain, and show static explorations. A website akin to our web pages, in a

[1] Out of our focus are implicit or partial differential equations, boundary values, though similar questions would apply.

university, is Ponce's [15], on Hydraulics, but see also Arsham's [16].

The above works relate to our study through concepts like simulation, web-based, kinetics, Monte Carlo, Internet, but omit *web pages* (Cambridge Dictionary) for display or user verification, where a browser is the sole need to solve a class of problems, the user just supplying data to obtain results. We have endorsed web-based work, hence citing various works of ours ([17]–[21]). (We recognize the uneasy leap into the Web as a computing medium.) Some articles nowadays offer their programs, perhaps in languages strange to the reader. Citing the 'FAIR' initiative ([22] or addendum): "There is an urgent need to improve the infrastructure supporting the reuse of scholarly *data*." [our emphasis] Its 52 authors duly write "data", but the lack seems worse about working programs in the literature, whereas web pages can be a simple alternative.

In the present text, some hyperlinks are given, indicated in the usual 'underline' style.

In the next parts of the paper: Section 2. ("The Kinetic Problem") mentions the chemical reactions' ODEs used; Section 3. ("Resolution Approach") determines constants in ODEs and details the Monte Carlo approach; Section 4. ("Computational Structure") focuses the web pages setup; Section 5. ("Results and Discussion") shows numerical effects of the said variation; and Section 6. ("Conclusions"), summarizes the approach and findings.

2. The Kinetic Problem

Our main, illustrative reaction is the ABC reaction, with generic components. It is a case of consecutive, irreversible reactions, chosen because of its computational simplicity and frequency, namely, in organic chemistry. The reaction to produce furfural follows the same kinetics. These two reactions are now described, followed by a reversible reaction with 4 components, the ABCD reaction. Another model, peripheral to our study, an SIR-type for epidemiological evolution is shown only in a web page (in List 1, below).

The first case of consecutive irreversible reactions is the ABC reaction, written in (1) (or just $\mathrm{A} \to \mathrm{B} \to \mathrm{C}$), and its assumed first-order kinetics (no cross products), with $a = [\mathrm{A}]$, etc., is described by this system with the chemically obvious three ODEs:

$$\begin{aligned} \mathrm{A} &\xrightarrow{\mathrm{k_1}} \mathrm{B} \\ \mathrm{B} &\xrightarrow{\mathrm{k_2}} \mathrm{C} \end{aligned} \tag{1}$$

$$\begin{cases} \mathrm{d}a/\mathrm{d}t = -k_1 a(t) \\ \mathrm{d}b/\mathrm{d}t = -k_2 b(t) + k_1 a(t) \\ \mathrm{d}c/\mathrm{d}t = k_2 b(t) \end{cases} \tag{2}$$

Their initial concentrations (time, 0) are, with working values $(a_0,\ b_0,\ c_0) = (1.,\ 0.,\ 0.)$, thus, an "initial value problem" (IVP). The ODEs are (mathematically) explicit, so, the proposed exploration is viable in moderate run time, crucial for Monte Carlo simulation.

The second case is the reaction to produce furfural (from Latin "furfur", (wheat) bran[2]), short for furfuraldehyde, was reported [23] to have the previous kinetics. Furfural (IUPAC nomenclature furan-2-carbaldehyde), in Fig. 1, is a valuable biomass-derived chemical. The reactions to produce it are $\mathrm{HE} \xrightarrow{k_1} \mathrm{FU} \xrightarrow{k_2} \mathrm{DE}$, where HE, FU, DE are hemicellulose, furfural, degradation products, respectively. With (A, B, C) = (HE, FU, DE), the ODEs in (2) are precisely

[2] De *Kleie*, Es *salvado*, Fr *son*, Pt *farelo*, Ro *tărâțe*, Ru *отруби*.

applicable. The determination of the rate constants from experimental data will be given (with its web page) as a short addendum in Section 5. ("Results and Discussion").

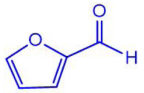

Fig. 1. Furfural ($C_5H_4O_2$).

For completeness, a case of a reversible reaction is included, though secondary to our scope. It is a numerical example (without an analytical solution) of a "reversible reaction with the dimerization of an intermediate" (component C). It is designated here the ABCD model, from "parameter estimation" [24], with unidentified components, A to D, $\mathrm{A} + \mathrm{B} \underset{\mathrm{k_2}}{\overset{\mathrm{k_1}}{\rightleftharpoons}} \mathrm{C}$ and $\mathrm{C} + \mathrm{C} \underset{\mathrm{k_4}}{\overset{\mathrm{k_3}}{\rightleftharpoons}} \mathrm{D}$. The following, reported ODEs, with given $(a_0, b_0) = (2.1, 3.1)$, and 4 rate constants, are readily computed in web page [25] with its default data (web pages listed in Section 3., 2.nd paragraph).

$$\begin{cases} \mathrm{d}C/\mathrm{d}t = k_1(a_0 - C - 2D)(b_0 - C - 2D) - k_2 C - 2k_3 C^2 + 2k_4 D \\ \mathrm{d}D/\mathrm{d}t = k_3 C^2 - k_4 D \end{cases} \tag{3}$$

3. Resolution Approach

The resolution approach, now described, relates to: (*i*) the determination, where pertinent, of the ODEs' parameters, which, in the application, are the reaction rate constants, done through a cautious choice of the regression optimizer; (*ii*) the detail of the Gaussians' simulation; (*iii*) the exploration by Monte Carlo; and (*iv*) the application to the ABC reaction.

The computations for the rates' determination (SIR, ABC, ABCD, furfural) and for the simulation can be freely verified in the following web pages: solving the SIR model direct, inverse; [26] determination of ABC rates; [25] determination of ABCD rates; [27] determination of furfural rates; [17] Monte Carlo simulation with the ABC reaction.

3.1. Determination of the rate constants

As experimental data are accessorily available for the reactions (ABC, ABCD, furfural), it became opportune to revisit the determination of reaction rates, done by typical regression. Useful findings arose on the better solver in Python, the obvious 'curve_fit' or 'minimize' [28].

The medium for the computation were the constructed web pages, written in PHP [29]. The regressions are conducted in Python, mostly, and Fortran 90. With the data in Table 1, the problem parameters, which are the 2 rate constants in (2), k_1, k_2, from an applicable reaction ([30], data seemingly made up), can be determined by adjusting the integrated model to the data. The number of components in the system of ODEs will be denoted by K and the number of instants by T. So, in Table 1, from [30], $K = 3$ and $T = 12$.

Some models in ODEs are analytically integrable, which is the present case. For such instances, e.g., Mathematica ® [31], Maple ® [32] are options. Notwithstanding, to assure generality, numerical methods are always used here.

In the case of Fortran, our minimization uses a Nelder-Mead search, one of the innumerable aids by Burkardt [33]. Another handy path would be to build a 'module', compiled via a mere call to 'f2py' [34], and directly callable from Python, with no auxiliaries.

Table 1. Experimental kinetic data

Instant	Time	[A]	[B]	[C]
1	0	1	0	0
2	2	0.88	0.12	0.003
3	6	0.69	0.29	0.03
...	...	...	...	...
10	120	0	0.12	0.88
11	150	0	0.06	0.94
12	200	0	0.02	0.98

Table 2. Rearranged data

Time	0	2	...	150	200	0	2	...	150	200	0	2	...	150	200
[A] [B] [C]	1	0.88	...	0	0	0	0.12	...	0.06	0.02	0	0.003	...	0.94	0.98

In the case of Python, the common option for curve fitting, if 'scipy.optimize' [35] is adopted, is its 'curve_fit' function. Yet, it has an adverse feature for application to multiple dependent variables, as is the case in systems of ODEs. In this type of problem, there is a single independent variable, but there are more than one dependent variables, one for each chemical species. Thus, in Table 1, there is one column for time, and $K > 1$ columns for the dependent variables.

If 'curve_fit' were chosen, the data would need rearrangement into strictly two columns: the independent variable column, for time, by concatenating $K = 3$ instances of the original vector; and the dependent variable column, as a single vector of size $T \,.\, K = 36$ by concatenating the K columns of the remaining data, as shown in Table 2 (landscape for shortness) and detailed in [21]. The first row (of length $T \,.\, K = 36$) would be the values of the new independent variable, and the second (same length) the values of the new sole dependent variable.

A simpler method (in the same Python module) was adopted, the 'minimize' function (dropping some 'curve_fit' features). In correspondence with the original (T, K) sized matrix of data: (*i*) a new matrix of same shape (T, K) of the calculated data is produced through the integrated ODEs with initial guesses of the parameters (i.e., k_1, k_2); and (*ii*) a sum of "distances", say S, between the two matrices is minimized. This scalar is a function of the two problem parameters, $S(k_1,\ k_2)$, which becomes a standard multivariate optimization. The choice of initial guesses is the typical difficulty of iterative processes, liable to nonconvergence, but the present problem is well-behaved in that sense. The usual sum of squares for S can of course be replaced by other criteria ("minimax", in absolute or relative terms, etc.), but such finer view is irrelevant here.

The multivariate optimization approach adopted was again the Nelder-Mead, the default in 'minimize'. In case constraints become necessary, as to avoid negative k values, the SLSQP [36] sequential quadratic programming is used.

3.2. Simulation of Gaussian random variables

Applying Monte Carlo with a Gaussian variable needs precautions due to the physical range of the variable and the random number generator (RNG), always recalling its range, typically $[0, 1)$.

The σ value for a variable, respecting its range, is difficult to set. Instead of a direct σ, the coefficient of variation, $c_v = \sigma/\mu$, becomes a simple, consistent option. This still poses an issue

in the frequent cases of $\mu \gtrsim 0$, but the relation is used reversely, with c_v as input: $\sigma = \mu\ c_v$.

The behavior of a Gaussian with positive mean near 0 ($\mu \gtrsim 0$) is arguable if negative values are nonvalid. This, though peripheral to our purpose, is attended below.

We relied on the native RNG in Python, which directly simulates a Gaussian distribution with given μ and σ (named 'loc', 'scale'). No reference was found on border value failure, which can arise from inverting the cumulative Gaussian. However careful the choice of σ may be as the counterpart to μ, the simulation of initial values must, at least, conform to their physical ranges as reactants' composition. A delicate case is the simulation of values for small $\mu \gtrsim 0$: with the chosen Gaussians, a way to avoid the (rare) negative values is to skip them. (Truncated Gaussian, Gamma or such are alternatives, out of our scope.) An RNG usually produces uniforms, $R \in [0, 1)$, and (using the "inversion method") its Gaussian x_R is $x_R = \mu + \sigma\Phi^{-1}(R)$, with Φ^{-1} the inverse cumulative.

The improbable $R = 0$ produces $x_R = -\infty$, thereby, an interruption ("abort") or at least an inordinate negative. (Excessive positive, rare values cause no interruptions.) For an (unreported) minimum, say $R = 3 \times 10^{-308}$, it is $\Phi^{-1}(R) = -37.5$, widely exceeding the 3 in the customary practical interval $\mu \pm 3\sigma$. To avoid interruption, negative initial values are skipped, counted, and shown in the end of execution. It may be seen that these values are absent or very rare, due to the realistically small variabilities (through c_v) used.

3.3. Methodology for the simulation

The assessment of the effect of the initial variation is solved in two steps: (*i*) if applicable, known concentrations (as in Table 1) are used to determine the rate constants; and (*ii*) the initial concentrations are made random with the Gaussians for the K components, $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ (non-standard, identical $\mathcal{G}(\mu, \sigma)$ would be more Engineering-like), for $i = m..n$ (concise notation for $\forall i \in [m, n]$, $i \in \mathbb{Z}$), where the various μ will be made equal to the desired (nominal, exact) initial compositions, $\boldsymbol{\mu} = (a_0,\ b_0,\ c_0)$, guarding, for small μ's, against liable negatives. Other distributions, as a said Gamma, might be used: $g(x; k, \theta) = \frac{1}{\theta\,\Gamma(k)}(x/\theta)^{k-1}\exp(-x/\theta)$, with (shape, scale) = (k, θ). Such option would, however, need considerations out of our scope to find values for its two parameters.

3.4. Exploration by Monte Carlo simulation

With the rate constants known, the effect of variation in the initial values (reactants) upon the variation in the final values (products) can be observed. As the computing is conjoined with its web page, a synthetic description follows those lines.

The reaction is computationally started (time, 0) and runs until a given final time, t_f. The rate constants, known or [21] previously determined, and the admittedly correct μ's are supplied, and the variations are substantiated in the c_v's. The reaction will now be run a large N (thousands) times, so that a statistic of final composition (at t_f) may be assessed. To make the results more concise, an "incumbent" component, k ($1 \leq k \leq K$), is chosen, with an assigned admissible tolerance, such as 5%, in the target concentration. Making variations on more than one component unnecessarily complicates the conclusions.

The number of Monte Carlo trials, N, and the random number generation seed are now inserted, with its repeatability, 'irepeat' = 1 (as for debugging) or 0 for pure random. Usual values of N are larger (millions), but it is to be noted that, for each trial, the whole system of ODEs

is (numerically) solved from the beginning, because it is the initial values that keep changing, following the Gaussians. Of course, no way was found to relate the initial variation size to that of the final variation. Further optimization of the numerical integration algorithm itself would be in order for more complex ODEs and longer Monte Carlo runs.

4. Computational Structure

The computational structure of this study will be described in parallel with its operation in the web pages, in which this track is not visible. The description follows the core, ABC case. In the pages, the user supplies the problem data, and control is passed to a Python script that does the calculations, returning the results in alphanumeric text and graphs. For these, we favored (free) 'gnuplot' neutrally callable by other languages, instead of the obvious Python 'matplotlib' module [37] to safeguard generality if using non-graphical languages (such as C, Fortran). Even with other languages, Python is a suitable manager, acting "as glue" [38]. This affinity may be crucial for speed, because Python is (mostly) interpreted, not compiled, a concern precisely in an extensive Monte Carlo simulation. The language versions are: Python 3.9, 'gnuplot' 5.4, and PHP 5.6, the versions in the public (Linux) system in the University.

In the web pages, the user submits the problem data, and clicks 'Execute' to obtain: (*i*) the rate constants, in the inverse problem (or [26]); or (*ii*) the behavior of the final composition, in the direct problem (or [17]). It is the latter, related to our main objective, that will be detailed. The execution, facilitated by the presence of default data, is in a simple style fit for Engineering. The dynamic results are built in HTML with the web-native language PHP [29], and the architecture consists of the web page as a front-end user interface, and a back-end for the computation.

The web interface is accessed by the user, accepts the task's arguments (data), processes them, schedules the task, and sends data to the back-end, which finally replies, thus building and showing the output in a new, temporary, dynamic web page.

The computed results are formatted via the HTML tag 'pre' (pre-formatted text), and displayed as the final web page on the user's terminal. The computations are executable on the University public system, maintained by its Computing Centre ([39]): a Linux operating system, reporting ('uname') 16 GB memory, amd64, Debian 5.10.70 (2021), x86_64 GNU/Linux, and ('lscpu') 8 CPU's, Xeon, 2 GHz.

The procedure, from the web page [17] and Fig. 2, is (problem particular names in italics):

a) The user reaches the web page, *P-ranKin.php*, and accepts the default data or inserts other in an HTML 'form', data to be sent to the next file via HTTP request ('P-' for Python).
b) Clicking 'Execute' sends the data to an intermediate PHP file, via 'form' with attribute 'action=*RanKin.php*', which sends the input to the Python script (next) as arguments. This PHP will get the results back from the Python execution via 'stdout', to create a temporary, dynamic PHP results web page.
c) The Python script, *ranKin.py*, is run through PHP 'shell_exec'.
d) Python's output ('stdout') is sent to the 'Results' page as HTML 'pre' tag with text and graphs ('png') just produced via Python 'base64' (leaving no files for deletion), and preventing clashes between concurrent users.
e) Auxiliary files — PHP environment files inserted by 'include', and cascading style sheet and images characterizing the website.

The italicized names suggest substitution for other problems, with the structure in Fig. 2.

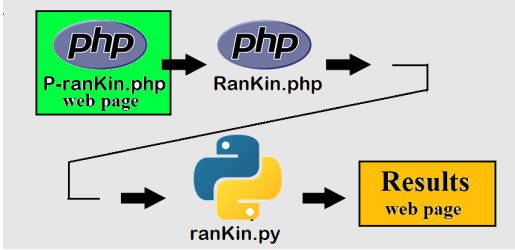

Fig. 2. Communication path from the 'input' web page [17] to the 'output' (Results) web page.

Web-based execution adds no difficulty to solving a problem, beyond its inevitable scientific hardship. We use the same executables on the command line for debug and long runs.

Writing programs in a compiled language (as C or Fortran) is more arduous than, e.g., user-friendly Matlab or Python, yet, if run times become large, compiled languages can be required. This happens in more complex problems, as suggested by the longer simulation runs to be shown.

In summary, to observe the said kinetic behavior, the data for the example are the following, giving results just by clicking 'Execute' if with default values (in parentheses):

a) Final time of the reaction, t_f (30) time units, as minutes;
b) Rate constants (0.06296 0.02115);
c) Concentrations' μ (1 0 0), of the respective 3 (K) components;
d) Concentrations' c_v (0.1, either one value if all equal or K values);
e) Incumbent (1, selected reactant for observation);
f) Tolerance (5%), selection from $10\%, 5\%, 2\%, 1\%$;
g) Trials (2.e+4) and seed (75357, used if 'Other') for the simulation;
h) Points (100) for both plots (evolution and simulation);
i) Show values (No) (of the plot coordinates).

Results are included (Section "Results and Discussion") for some data combinations. The final "behavior" is simply a histogram of the simulated values, estimating the probability density functions. The curves bear resemblance to Gaussians, but unraveling it is out of our scope.

5. Results and Discussion

The effects of the random variation are now presented, after a description of the strategy of the Monte Carlo simulation. The numerical results of the simulation are explored to assess the effect of the initial composition variation on the final composition, from the web pages (in List 1), where the results can be confirmed.

As an addendum to our results, we give just a summary of the determination of rate constants (peripheral to our scope) for the furfural reaction (Section 2.). The constants arise from (default data): compositions vs. time; initial guesses, $(0.008, 0.008)$; computational mode, 'optimize'. The result is (link[3]) $(k_1, k_2) = (0.00783, 0.00047)$, with absolute deviations $\lessapprox 0.05$.

[3]These "links" connect to web page executions that run for $\sim$ 10–30 s.

5.1. Strategy of the Monte Carlo simulation

In a Monte Carlo simulation, an inherent issue is its "run length", which is often solved doubly by computational trials, their repetition consolidating the length. A run length tentative estimation is, e.g., at the website [16]: for pilot run length of 500, composition estimate $1.$, variance 0.5, significance 5%, and relative error 5%, it gives $N \approx 4000$, whereas half the error (2.5%) of course quadruples it to $N = 16000$. A value of $N = 2 \times 10^4$ was used from this estimation, this length typically fitting in the prescribed time limit in the University's website. This length sufficed to yield "reproducible" results (in the sense of Monte Carlo). An arbitrary unique simulation seed of 75357 is used to permit verification of the results by a user.

To initiate the exploration itself, a small value of σ, say, $1/10$, as a fraction of μ, i.e., $c_v = 0.10$, was deemed reasonable. In order to improve (decrease) this variability, a criterion of successive reductions was envisaged. Since a "geometric" reduction appeared more adequate than, e.g., an "arithmetic" one, a Renard-type [40] "R5" reduction was adopted. This means that σ will successively be divided (the 5 in R5) by $\sqrt[5]{10} = 1.58489\ldots$. For the said first $\sigma = 0.1$, this gives the sequence (rounded) 0.1, 0.0631, 0.0398, 0.0251, 0.0158, 0.01, this last one a handy value, after the desired five reductions.

Values of tolerance in the final composition were chosen at usual 10%, 5%, 2%, and 1%. The question to be resolved is, thus, to assess the effect of, say, a variation of 5% (i.e., $c_v = 0.05$ or $\sigma = 0.05\,\mu$) in a reactant's composition upon the resulting *fraction nonconforming*.

The final concentrations will be classified as whether they fall inside acceptable intervals, the "fraction conforming", and in the opposite case the "fraction nonconforming". For a correct operation, this fraction nonconforming must be, of course, the smaller the better.

The expressions "fraction conforming", "fraction nonconforming" (or "defective"[4]) are from the Quality discipline ("fraction", attributive noun, and "conforming", "defective", adjectives as nouns. A fraction nonconforming, to be denoted by FNC, means that, e.g., for an FNC = 10% in final composition, this composition falls that often, 10% of the cases, outside a specified interval.

5.2. Numerical results of the simulation

The exploration of the effects of values of c_v and tolerance was done by the Monte Carlo simulation mentioned, applied to the ABC reaction (using web page [17]).

A test to the "reproducibility" of results was executed to perceive the sufficiency of the simulation run length, since its best value is not easy to set. The mentioned $N = 2 \times 10^4$ met that end without exceeding our web-based computing limit of $\sim$2 min. Private, direct, command line long runs, out of the Internet, have no practical time limit.

Long runs point to attain "smooth" curves, shedding light on due length. Thus, long runs of 0.1, 0.5, 0.75, and 1.5 million trials were executed on the command line (with exactly the same program). The test case has a (heavy) $c_v = 0.1$ and tol $= 5\%$. The execution times were, respectively, 1.5, 13, 20.5, and (double) 41 min. The curves for the first and last run lengths (the penultimate curves resembled the last) are in Fig. 3. The enormous fraction nonconforming FNC $\approx 62\%$ is a sheer consequence of the high c_v. Regarding run length, in order to permit verification by the user, the said $N = 2 \times 10^4$ trials (default) will be used.

Two cases were run from the command line with larger $N = 0.5$ million trials: the cases $c_v = (0.10,\ 0.04)$ took 15 min each (above the website limit), and the results, visible at [41],

[4] Deprecated, because a defective item can be effective in another context (or for semantic bias).

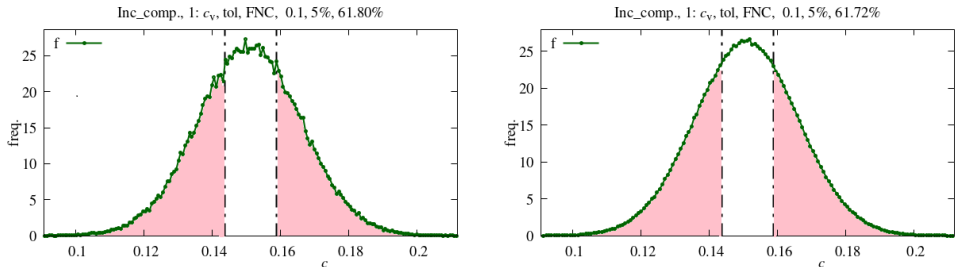

Fig. 3. Long runs, ABC reaction, with 0.1 (left) and 1.5 (right) million trials, giving FNC $\approx 62\%$, color signaling the FNC area.

were similar to those in Fig. 3, FNC $\approx$ 62%, and FNC $\approx$ 21%, respectively, sustaining our default N. Each integration of the problem ODEs, thus, took $15\,\text{min}/(0.5 \times 10^6) = 2.$ ms.

5.2.1. Exploring variability and tolerance

Starting the exploration proper, two introductory cases were computed, with $c_v = (0.10, 0.04)$ and default initial concentrations, (1, 0, 0), tol = 5%, and the adopted N trials and seed. The worse case, with $c_v = 0.10$, yields (link) FNC = 61.9%, and the better case, with $c_v = 0.04$, yields (link) FNC = 20.9%. These results for the reaction illustrate the effect of the variation of the initial compositions on the final ones. (Both FNC's, industrially very dissatisfactory, give an idea of the ranges of c_v to explore.) The final nonconforming (defective) compositions fall in tails of the distribution (histogram) that are undesirably outside the 5% tolerance, or 'tol', indicated by the two parallel vertical dash-dotted lines in the images. Reducing the initial variation — not an ordinary effort — is, of course, a way to improve final quality.

Led by the previous results, two improved cases are added (with values picked from data annex to Fig. 4, below) and tol = 5%: (*i*) $c_v = 0.0251$, giving (link) FNC = 4.74%; and (*ii*) $c_v = 0.0158$, giving (link) FNC = 0.15%, with graphs in the web pages.

Changing only the acceptable tolerance, in two cases now addressed, maintains the same behavior but alters the fraction nonconforming (quality is indeed relative). For a c_v kept at 0.0158, a lax tol = 5% yields (link) FNC = 0.15%, which seems good, but reducing to tol = 2% raises the fraction to a dismal (link) FNC = 20.4%, anyway a hardly predictable sensitivity.

5.2.2. Refinements

The answer to the question posed in this article is in the data annex to Fig. 4 (below), where, for instance, the values tol = 5% and $c_v = 0.1$ lead to an enormous FNC = 61.9%. To reduce (improve) this fraction, obviously, a smaller $c_v = 0.0631$ leads to a lower (better) FNC of 42.4%. An improvement would of course be expected, but its now known magnitude was not predictable.

All the results presented must be looked at prudently. The assumptions of the adherence of the ODEs to the real matter, and of the random behavior (Gaussian) are just reasonable hypotheses. Anyway, the tendencies and their magnitudes are certainly trustworthy.

Finally, a certain, common FNC can of course be obtained for different target tolerances: e.g., FNC = 8% (high, quite "unindustrial", for clearer results), for two tolerances, 5% and 10%. Solving the inverse problem semi-manually led to: (*i*) (link) (tol, c_v) = (5%, 0.02867); and (*ii*) (link) (tol, c_v) = (10%, 0.05734).

Various calculations were done to show how the initial variation, through c_v, affects the nonconformance, i.e., FNC, fraction nonconforming, defective, of the final composition. This was done for the values in data annex to Fig. 4: σ, through c_v, decreasing (improving) from 0.1 to 0.01, according to the Renard series R5, and tol $= (5\%, 2\%, 1\%)$. The results appear in Fig. 4, confirming the opposing tendencies: for a given c_v, a high tolerance, 5% (careless operation) leads to lower fraction nonconforming, FNC, while a low tolerance, 1% (strict operation) leads to higher FNC. Beyond obvious *tendencies*, it is the computation by simulation that provides the concrete values of acceptability and their sensitivity. For application, two opposite factors are at stake: initial variability and final quality. Both are "expensive", so, the economic decision to balance them will be possible with real economic data.

c_v	tol = 5%	tol = 2%	tol = 1%
0.1	0.616	0.839	0.919
0.0631	0.424	0.753	0.874
0.0398	0.208	0.613	0.799
0.0251	0.045	0.428	0.691
0.0158	0.001	0.204	0.526
0.01	≈0	0.046	0.321

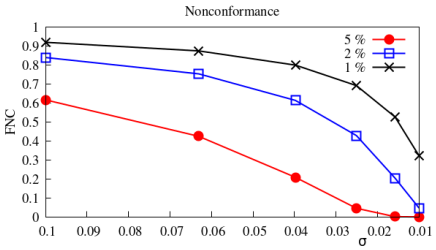

Fig. 4. Data (left) and fraction nonconforming ($\sigma \propto c_v$) for tolerances $5\%, 2\%, 1\%$ (right).

6. Conclusions

A question in ODEs was studied computationally using a chemical kinetics illustration to investigate the effect that an undesired random variation in the initial values (the reactants' composition) has on the variation in the final values (the product composition). Many other systems of ODEs in STEM (Science, Technology, Engineering, Mathematics) might obviously have been chosen.

All our computing resides in web pages as a computing medium. Monte Carlo simulation was adopted, since the fact that the underlying ODEs are, in general, solved numerically precludes finding an analytic relationship. The initial values were subjected to Gaussian behavior to observe the final values against their specification. The results show how much a variation in the initial values leads to a variation in the final values: high input variation, expressed as coefficients of variation, c_v, implies high, undesirable output variation, measured as fraction nonconforming (FNC). For a tolerance of, e.g., 2% admitted in final values, if c_v is relaxed from 0.01 to 0.025, FNC goes from a passable 4.6% to a huge 43%. Sensitivity is, thus, also reachable by this tool.

The languages PHP and mostly Python (with 'gnuplot') were operant, and our programs are the same off or on the Web. We show this client-server computing as a web-based access and execution, without any user need, and the web pages, thus, permit to verify the results. The architecture used serves an extent of other problems.

The study provided: an option for Python 'minimize' against 'curve_fit'; the sought assessment of the effect of initial values' variability; and the operation on web pages. We value direct web-based computation verifiability, scarcely used in articles. From our experience in academy and industry, Web computing is fit for both and their connection.

Acknowledgements. The work of Miguel Casquilho and Rui Galhano was supported by the projects FCT-UIDB/04028/2025 and FCT-UIDP/04028/2025. The work of Jorge Buescu was

supported by the project UIDB/04561/2020. CIIST (Informatics Centre of IST) supplied the computing system.

References

[1] D. J. HIGHAM, *Modeling and simulating chemical reactions*, SIAM Review **50**(2), 2008, pp 347–368.

[2] S. KUNTSCHE, T. BARZ, R. KRAUS, H. ARELLANO-GARCIA and G. WOZNY, *MOSAIC a web-based modeling environment for code generation*, Computers and Chemical Engineering **35**, 2011, pp 2257–2273.

[3] MOSAIC, Use of Basic Elements II—-Ordinary Differential Equation System. Accessed: Oct. 20, 2024.

[4] C. POZNA and R.-E. PRECUP, *Aspects concerning the observation process modelling in the framework of cognition processes*, Acta Polytechnica Hungarica **9**(1), 2012, pp. 203–223.

[5] C. LIANG, Y. HU, L. GUO, L. WU and W. ZHANG, *Kinetic study of acid hydrolysis of corncobs to levulinic acid*, Bioresources **12**(2), 2017, pp, 4049–4061.

[6] L. YE, Y. HAN, X. WANG, X. QI and H. YU, *Recent progress in furfural production from hemicellulose and its derivatives: Conversion mechanism, catalytic system, solvent selection*, Molecular Catalysis **515**, 2021, paper 111899.

[7] N.-M. ŞERBAN and R. HOBINCU, *Computer vision algorithm for detecting resistor color codes*, Romanian Journal of Information Science and Technology **24**(3), 2021, pp. 321–333.

[8] S. MISRA, L. R. BUTTAZONI, V. AVADIAPPAN, H. J. LEE, M. YANG and C. T. MARAVELIAS, *CProS: A web-based application for chemical production scheduling*, Computers and Chemical Engineering **164**, 2022, 107895.

[9] V. RUSSO, H. GRÉNMAN, T. SALMI and R. TESSER, *A novel approach to inulin depolymerization: A Monte Carlo based model*, Chemical Engineering Science **256**, 2022, paper 117712.

[10] M. BOGDOS and B. MORANDI, *EveRplot: A web-based shiny application for creating energy vs reaction coordinate diagrams*, Journal of Chemical Education **100**, 2023, pp, 3641–3644.

[11] S. ABRAMOV, S. TRAVIN, G. DUCA and R.-E. PRECUP, *New opportunities model for monitoring, analyzing and forecasting the official statistics on coronavirus disease pandemic*, Romanian Journal of Information Science and Technology **26**(1), 2023, pp. 48–63.

[12] U. EŞSIZ, Ç. ACI, E. SARAÇ and M. ACI, *Deep learning-based prediction models for the detection of vitamin D deficiency and 25-hydroxyvitamin D levels using complete blood count tests*, Romanian Journal of Information Science and Technology **27**(3–4), 2024, pp. 295–309.

[13] L. YAN, T. ZHAO, X. XIE and R.-E. PRECUP, *OSSEFS: An online semi-supervised ensemble fuzzy system for data streams learning with missing values*, Expert Systems with Applications **255**, 2024, paper 124695.

[14] P. HELO and V. THAI, *Logistics 4.0 — digital transformation with smart connected tracking and tracing devices*, International Journal of Production Economics **275**, 2024, paper 109336.

[15] V. M. PONCE, San Diego State University, https://uon.sdsu.edu/Accessed: Oct. 20, 2024.

[16] H. ARSHAM, Determination of Simulation Runs' Size (Systems Simulation), University of Baltimore, 2015. Accessed: Oct. 20, 2024.

[17] M. CASQUILHO, 2024, Random towards kinetics.

[18] M. CASQUILHO and J. BUESCU, *Standard deviation estimation from sums of unequal size samples*, Monte Carlo Methods and Applications **28**(3), 2022, pp, 235–253.

[19] M. CASQUILHO, *Computação científica, Internet, Indústria (Scientific computing, Internet, Industry)*, Proceedings of 1st Portuguese Meeting on Mathematics for Industry, FCUP, University of Porto, Portugal, 2013, pp. 1–6.

[20] M. BARROS and M. CASQUILHO, *Linear programming with CPLEX: an illustrative application over the Internet*, Proceedings of 14th Iberian Conference in Information Systems and Technologies, Coimbra, Portugal, 2019, pp. 1–6.

[21] M. CASQUILHO, P. PACHECO, R. GALHANO, I. PAULO, J. L. MIRANDA and J. BORDADO, *Fitting a system of ODEs to data in Python: select the 'function' and run on the Web*, Proceedings of 19th Iberian Conference on Information Systems and Technologies, Salamanca, Spain, 2024, pp. 1–6.

[22] M. WILKINSON, M. DUMONTIER, I. AALBERGSBERG et al., *The FAIR guiding principles for scientific data management and stewardship*, Scientific Data **3**, 2016, paper 160018.

[23] X. LI, J. YANG, R. XU, L. LU, F. KONG, M. LIANG, L. JIANG, S. NIE and C. SI, *Kinetic study of furfural production from Eucalyptus sawdust using H-SAPO-34 as solid Brønsted acid and Lewis acid catalysts in biomass-derived solvents*, Industrial Crops and Products **135**, 2019, pp. 196–205.

[24] Maplesoft, *Parameter Estimation for a Chemical Reaction*. Accessed: Oct. 20, 2024.

[25] M. CASQUILHO, 2023, Fitting in kinetic ODEs.

[26] M. CASQUILHO, 2019, Chemical reaction parameters.

[27] M. CASQUILHO, 2023, Furfural: fitting in kinetic ODEs.

[28] Python 'scipy.optimize.curve_fit', 'scipy.optimize.fmin', 2024, Accessed: Oct. 20, 2024.

[29] PHP, The PHP Group. Accessed: Oct. 20, 2024.

[30] B. WINKEL, *Parameter estimates in differential equation models for chemical kinetics*, International Journal of Mathematical Education in Science and Engineering **42**(1), 2011, pp. 37–51.

[31] Mathematica ("Wolfram Mathematica"). Accessed: Oct. 20, 2024.

[32] Maplesoft. Accessed: Oct. 20, 2024.

[33] J. BURKARDT, Code: 'asa047', 2021. Accessed: Oct. 20, 2024.

[34] F2PY user guide and reference manual. Accessed: Oct. 20, 2024.

[35] Python, 2023, 'scipy.optimize' (last update Dec. 31). Accessed: Oct. 20, 2024.

[36] D. KRAFT, A software package for sequential quadratic programming, 1988. Accessed: Oct. 20, 2024.

[37] J. D. HUNTER, *Matplotlib: A 2D graphics environment*, Computing in Science & Engineering **9**(3), 2007, pp. 90–95.

[38] The SciPy community, "Using Python as glue", accessed 2024/10/20.

[39] CIIST, 'Centro de Informática do IST' (*Informatics Centre of Instituto Superior Técnico*). Accessed: Oct. 20, 2024.

[40] ISO 3:1973, Preferred numbers — Series of preferred numbers, 1973. Accessed: Oct. 20, 2024.

[41] M. CASQUILHO, 2025, Figures.