

Title	Development and classroom pilot of PK-SIMple: An open-source pharmacokinetics teaching tool for pharmacy students
Authors	Murray, Jack D.;Hoffmann, Sophia V.;Maguire, Ryan C.;Bennett-Lenane, Harriet;Vucen, Sonja
Publication date	2026-06-13
Original Citation	Murray, J D, Hoffmann, S V, Maguire, R C, Bennett-Lenane, H & Vucen, S 2026, 'Development and classroom pilot of PK-SIMple: An open-source pharmacokinetics teaching tool for pharmacy students', Currents in Pharmacy Teaching and Learning, vol. 18, no. 10, 102719, pp. 1-9. https://doi.org/10.1016/j.cptl.2026.102719
Type of publication	Article (peer-reviewed)
Link to publisher's version	https://www.scopus.com/pages/publications/105041503290 - 10.1016/j.cptl.2026.102719
Rights	© 2026, authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/) - https://creativecommons.org/licenses/by/4.0/
Download date	2026-10-24 21:05:21
Item downloaded from	https://hdl.handle.net/10468/19485



Contents lists available at ScienceDirect

Currents in Pharmacy Teaching and Learning

journal homepage: www.sciencedirect.com/journal/currents-in-pharmacy-teaching-and-learning



Teaching and Learning Matters

Development and classroom pilot of PK-SIMple: An open-source pharmacokinetics teaching tool for pharmacy students

Jack D. Murray^{a,*}, Sophia V. Hoffmann^a, Ryan C. Maguire^a,
Harriet Bennett-Lenane^a, Sonja Vucen^{a,b}

^a School of Pharmacy, University College Cork, Cork T12 K8AF, Ireland

^b SSPC, Research Ireland Centre for Pharmaceuticals, School of Pharmacy, University College Cork, Cork T12 K8AF, Ireland

ARTICLE INFO

Keywords:

Pharmacokinetics
Python
Educational technologies
Pharmacokinetic modelling
Universal design for learning

ABSTRACT

Background: Pharmacokinetics is a fundamental component of pharmacy education that students often find challenging due to its quantitative nature. Although instructional pharmacokinetics simulation tools have been previously reported, the requirements for installation, training, or licensing hinders their widespread adoption. Consequently, research demonstrates that the uptake of such software for pharmacy education is limited.

Educational activity: To address this gap, we developed PK-SIMple: a python-based Streamlit application that students can open directly in their browser to generate concentration-time profiles. The application was piloted in two 90-min workshops with third-year undergraduate pharmacy students. We detail the development and technical testing of the application to support pharmacy educators wishing to design their own instructional technologies.

Evaluation findings: Load-testing demonstrated that PK-SIMple performed quickly and reliably even under high user demand. This was corroborated by the application's real-world performance during the workshops, where the simulation time remained negligible. Both instructors and students were able to work through three case studies independently, demonstrating that the application does not require prior knowledge or specialist instruction for use.

Analysis of educational activity: PK-SIMple reflects principles of Universal Design for Learning by offering inherent flexibility in the level of theory that can be incorporated into a simulation. Our experience illustrates that the development of in-house technologies, though technically demanding, is both feasible and rewarding for pharmacy students and educators. An international, multi-institutional study is planned to assess the impact of PK-SIMple on students' enjoyment and understanding of pharmacokinetics.

Background

Competence in pharmacokinetics is central to the practice of pharmacy.¹ However, pharmacokinetics is widely recognised as a challenging subject for pharmacy students given its quantitative nature and the difficulty of linking mathematical models to clinical decision-making. This observation is reflected in the literature. In a recent survey of 318 students across 11 universities, which

* Corresponding author at: School of Pharmacy, University College Cork, Cork T12K8AF, Ireland.

E-mail addresses: jackmurray@ucc.ie (J.D. Murray), 122121868@umail.ucc.ie (S.V. Hoffmann), 118332343@umail.ucc.ie (R.C. Maguire), hbennettlenane@ucc.ie (H. Bennett-Lenane), svucen@ucc.ie (S. Vucen).

<https://doi.org/10.1016/j.cptl.2026.102719>

Available online 13 June 2026

1877-1297/© 2026 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

included 110 pharmacy students, Babey et al.² found the understanding of four pharmacokinetic concepts (namely bioavailability, clearance, volume of distribution, and steady-state concentration) to be poor. Each term was defined in a complete and correct manner in less than 30% of responses. Hughes et al.³ note that students frequently raised the difficulty of the pharmacokinetics course in mandatory exit interviews, and attributed this in part to its mathematics-intensive character. Persky and Pollack⁴ similarly remark that students dislike basic pharmacokinetics due to the high numerical content and difficulty in appreciating its clinical relevance.

These challenges have fuelled research into non-traditional teaching methods to improve understanding and enjoyment of the subject. Digital tools which simulate concentration-time curves are particularly popular in the pedagogical literature. For a comprehensive overview of early simulation tools, the interested reader is directed to Charles and Duffell⁵ who review developments from 1970s to 2000. In the recent literature, there is a notable diversification away from standalone applications which require installation. uSIMPk⁶ is a pharmacokinetic simulator built in Microsoft Excel using Visual Basic for Applications, implementing one- and two-compartment models, nonlinear pharmacokinetics, and repeated dosing. Although Excel is likely to be familiar to pharmacy educators, thereby removing a barrier to use, the authors noted significant compatibility issues with macOS which required students with Mac devices to use a Windows machine. Mehvar⁷ developed six web-based simulation modules to demonstrate a range of pharmacokinetic concepts, including bioavailability, the therapeutic window, and nonlinear pharmacokinetics. While this overcomes the problem of installation, the modules appear useful only within the narrow scope of predefined case studies. Furthermore, they were built using a commercially-available webpage design platform rather than open-source code. Han et al.⁸ report an application available both as a webpage and a standalone software built using JAVA and R. In this case, the input parameters appear fixed, thereby limiting the tool to a single, predefined level of pharmacokinetic complexity which may not meet the needs of all learners. Other novel modalities include a Physiologically-Based Pharmacokinetic (PBPK) model developed specifically for teaching⁹ and a mechanical patient simulator,¹⁰ though these are associated with many of the same barriers as the previously described technologies.

The uptake of these tools outside of the institutions where they are developed appears to be poor. Only 9 out of 82 pharmacy colleges in the United States used web applications to teach pharmacokinetics in 2017,¹¹ though there is a paucity of recent studies assessing teaching methodologies. This limited uptake is reflected in our local experience. Pharmacokinetics is integrated throughout the five-year Master of Pharmacy programme at the School of Pharmacy, University College Cork (UCC). The classic mathematical foundations of pharmacokinetic modelling and their clinical use, however, is delivered through a dedicated third-year module worth 10 European Credit Transfer and Accumulation System (ECTS) credits. "Pharmacokinetics: From Basic Principles to Clinical Applications" (PF3015) provides fundamental teaching on pharmacokinetic models, biopharmaceutics, the chemistry of drug metabolism, and the use of pharmacokinetics in clinical practice. In terms of core modelling approaches, the curriculum introduces compartmental modelling with first-order absorption and elimination, non-linear pharmacokinetics, and non-compartmental analysis. It covers intravenous bolus and infusion models, extravascular administration, repeated dosing, and application of the above models to dosage regimen design.

We have previously integrated both Excel-based exercises and PBPK Modelling software into pharmacokinetics instruction. Anecdotally, we found that the students' lack of prior experience with Excel significantly hinders their learning as difficulties in using spreadsheets overshadows the pharmacokinetic concepts. Our experience of PBPK modelling software was similar. Despite case reports of positive experiences with PBPK-based teaching in the literature,^{9,12,13} feedback from students at UCC indicates that this approach did not improve their understanding. These tools do not facilitate self-paced learning or differentiated instruction by design. In our local context, the cost associated with some PBPK software was prohibitive, posing an additional barrier to their adoption as a teaching tool.

The aim of the current work was to develop a simulation software designed for pharmacokinetics teaching that facilitates pharmacy students' learning and conceptual understanding, while overcoming limitations related to availability, cost, and usability. To this end, we created PK-SIMple, a Streamlit-based open source web application which is optimised for many concurrent users and has no requirement for specialist training or installation. We share lessons learned from piloting this novel tool in a classroom setting to assess its feasibility, highlighting practical considerations to ensure effective implementation. Importantly, the focus of this Teaching and Learning Matters piece is on the development and deployment process. This is something the authors feel is under-represented in the pharmacy education literature, despite being critical for other pharmacy educators with some coding knowledge who may wish to build similar tools. By outlining in detail how PK-SIMple was developed and tested, we aim to offer a framework for pharmacy educators to design their own web applications tailored to their specific teaching needs.

Educational activity

To support implementation of PK-SIMple at other Universities, its design and deployment are reported in line with the Template for Intervention Description and Replication (TIDieR) checklist.¹⁴ This work was deemed to not require ethical approval as the focus of this manuscript was the development and technical testing of the application. This exemption was confirmed by the local social research ethics committee. Sex and gender were not considered in this study as they are not relevant to the development or technical evaluation of the application.

Application development

The PK-SIMple application was developed in Python using Streamlit, a Python-based web development framework. The name PK-SIMple was inspired by the open-source PBPK modelling platform, PK-SIM. All code associated with the current study is available on GitHub via the following link: <https://github.com/jack-d-murray/pk-simple>. We prioritised a lightweight design and computational

efficiency to allow many students to generate simulations concurrently. Streamlit is a highly accessible library, and those with prior experience of coding in Python are likely to find the framework familiar and user-friendly. Less experienced coders and those with no coding background are encouraged to use our application as a foundation in developing their own resources.

The application implements an open, linear, one-compartment model with first order absorption and elimination, as this model forms the foundation of introductory pharmacokinetics teaching at UCC. Three different dosing methods (“intravenous bolus”, “intravenous infusion”, and “extravascular (oral/other)”) are supported. Aligning with typical undergraduate pharmacy curricula, both single-dose and multiple-dose simulations can be performed, with the latter concentration-time curve calculated via the superimposition principle. The key pharmacokinetic parameters used by PK-SIMple, including their abbreviations and units, are provided in Table 1. Closed-form equations (Fig. 1) describing each of the fundamental pharmacokinetic processes were selected over numerical simulation of differential equations to reduce the computational burden. Calculations were performed using the NumPy library in Python.¹⁵ The implementation of the major pharmacokinetic processes, namely Absorption, Distribution, Metabolism, and Excretion (ADME) are described below.

Absorption

Upon selection of “extravascular (oral/other),” the student is given the option of directly specifying a bioavailability, or having the bioavailability calculated from the fraction absorbed, fraction escaping gut wall metabolism, and the hepatic extraction ratio. The student is also prompted to enter an absorption rate constant. If the elimination rate exceeds the absorption rate, a warning appears notifying the student that flip-flop pharmacokinetics have been detected. The concentration-time plot produced by the extravascular concentration function automatically implements flip-flop pharmacokinetics and the terminal slope is governed by the absorption rate constant.

Distribution

The volume of distribution can be explicitly defined, or can be calculated from the plasma volume, the tissue volume, the fraction unbound in plasma, and the fraction unbound in tissue. The volume of distribution is recalculated dynamically as each of these values are changed, allowing students to interactively investigate the effects of disease or displacement from plasma proteins on distribution. A fraction unbound must be provided if the student wishes to calculate the hepatic extraction ratio or renal clearance from the glomerular filtration rate.

Table 1

Key pharmacokinetic parameters included in PK-SIMple and their standard abbreviations. Units correspond to those required when students input parameters to the application.

Parameter	Description	Unit
General		
t	Time	h
C _p	Plasma concentration	mg L ⁻¹
C _p (t)	Plasma concentration at time t	mg L ⁻¹
C _p (0)	Plasma concentration at t = 0	mg L ⁻¹
Absorption		
k _a	Absorption rate constant	h ⁻¹
F	Bioavailability	None
F _a	Fraction absorbed	None
F _g	Fraction escaping gut wall metabolism	None
AUC	Area under the curve	mg h L ⁻¹
C _{max}	Maximum plasma concentration	h
t _{max}	Time to maximum plasma concentration	mg L ⁻¹
Distribution		
V _d	Volume of distribution	L
V _p	Plasma volume	L
V _t	Tissue volume	L
f _u	Fraction unbound in plasma	None
f _{ut}	Fraction unbound in tissue	None
Metabolism and Elimination		
CL	Clearance	L h ⁻¹
CL _r	Renal clearance	L h ⁻¹
CL _h	Hepatic clearance	L h ⁻¹
CL _{int}	Intrinsic clearance	L h ⁻¹
CL _{tubular}	Renal tubular clearance	mL min ⁻¹
Q _h	Hepatic blood flow	L h ⁻¹
GFR	Glomerular filtration rate	mL min ⁻¹
k _{el}	Elimination rate constant	h ⁻¹
t _{1/2}	Half-life	h

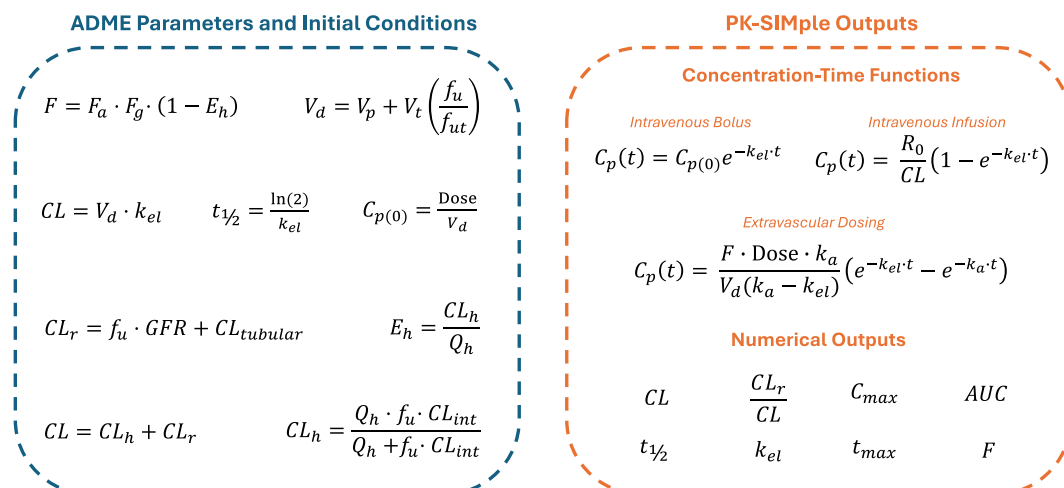


Fig. 1. Overview of equations implemented by PK-SIMple. The blue box describes the calculation of various ADME parameters. The orange box shows the concentration-time functions for intravenous bolus, intravenous infusion, and extravascular administration, and also lists the numerical outputs of PK-SIMple. For simplicity, some calculations have been omitted (e.g. the implementation of flip-flop pharmacokinetics for extravascular administration, calculation of plasma concentration after the end of an intravenous infusion, the superimposition principle, etc.). A complete description of the underlying calculations is provided in the source code. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(a) Absorption

Route of administration: Extravascular (oral/other)

Specify bioavailability directly or calculate from F_a , F_g , and E_h ?

☐ Directly specify F

☒ Calculate F

Fraction absorbed F_a : 0.86

Fraction escaping gut metabolism F_g : 0.90

⚠ Hepatic clearance must be calculated from CL_{int} and Q_h in order for E_h to modify oral bioavailability.

Absorption rate constant k_a (h^{-1}): 1.00

(b) Distribution

Volume of distribution V_d mode:

☐ Fixed V_d

☒ Calculate from f_u/f_{ut}

Plasma volume V_p (L): 3.00

Tissue volume V_t (L): 37.00

Fraction unbound in plasma f_u : 0.90

Fraction unbound in tissue f_{ut} : 0.80

Calculated $V_d = 44.62$ L

(c) Renal Clearance via GFR

GFR (mL/min): 100.00

Secretion/Reabsorption adj. (mL/min): 62.00

Calculated renal clearance = 9.12 L/h

Hepatic Clearance via Q_h and CL_{int}

Hepatic blood flow Q_h (L/h): 90.00

Intrinsic clearance CL_{int} (L/h): 200.00

Hepatic extraction ratio $E_h = 0.67$

(d) Hepatic Clearance by Individual Enzyme Contributions

▼ Add/remove enzyme contributions (L/h)

Enzyme name: CYP3A4

Enzyme CL (L/h): 9.00

Add enzyme

CYP2D6: 3.00 L/h Remove CYP2D6

CYP2C9: 4.00 L/h Remove CYP2C9

Fig. 2. Four static images of the user interface of PK-SIMple's input panel. The interface is rendered using browser-native Document Object Model elements, ensuring visual quality independent of zoom level. (a) The absorption panel showing the higher level of theory, where students must specify the fraction absorbed and fraction escaping gut wall metabolism by means of a slider. The student is notified that the hepatic metabolism must be calculated from the intrinsic clearance and hepatic blood flow rate in order for first-pass metabolism to impact bioavailability. (b) The automated calculation of volume of distribution. The parameter dynamically updates as the slider is moved. (c) Calculation of elimination through separate renal and hepatic components. (d) An alternative method of calculating hepatic metabolism where the student may directly specify their own metabolic enzymes.

Metabolism and excretion

Students can describe drug elimination at varying levels of complexity, as we felt it was important to provide flexibility to select the option which most closely suits a student's level of confidence and prior learning. The most basic implementation of elimination is through the “Global Parameter” option, where one of the half-life, total clearance, or elimination rate constant must be provided. Although this provides a high-level overview of elimination, it obscures the highly interdependent nature of pharmacokinetic parameters. For example, if total clearance is specified directly by the student, it will not change dynamically with the volume of distribution. Thus, for a deeper mechanistic view of elimination, the “Renal-Hepatic” option separates total clearance into renal and hepatic components. Renal clearance is calculated from the fraction unbound in plasma and the glomerular filtration rate, with a slider for tubular secretion and reabsorption incorporated. Hepatic clearance may be determined from the intrinsic hepatic clearance and hepatic blood flow, or, for additional detail, by specifying individual enzyme pathways, thereby enabling the student to simulate pharmacogenetic variability and drug-drug interactions.

Model output

Upon clicking “Run Simulation,” an interactive and downloadable concentration-time plot is generated using plotly.¹⁶ This can be converted to a log-linear plot to demonstrate that the exponential decrease in plasma concentration is linear when plotted on a logarithmic scale. A range of pharmacokinetic parameters listed as numerical outputs in Fig. 1, including Area Under the Curve and fraction excreted unchanged in urine, are also calculated.

Hosting and simulated load testing

The application was deployed using Streamlit Community Cloud, a free hosting platform for Python applications built in Streamlit. This enables the end user to access the application in the browser without the need for a local installation. We found that significant barriers to the widespread use of other simulation tools include the need for students to install software or obtain licences. PK-SIMple removes these steps and allows for more instructor time to be dedicated to pharmacokinetic understanding.

Performance under concurrent user load was tested using ThreadPoolExecutor in Python. To simulate many students concurrently accessing the application, 100 Hypertext Transfer Protocol (HTTP) requests were issued simultaneously. The request success rate, average response time, and total test duration were recorded.

PK-SIMple application

The final application is available at <https://pk-simple.streamlit.app/>. Users may need to reactivate the application after a period of inactivity. A selection of images of the parameter input panel is shown in Fig. 2. Each ADME process allows for varying levels of complexity to be integrated. For example, in the calculation of volume of distribution (Fig. 2 (b)), the parameter may be directly specified by the student, or it may be calculated using the interactive sliders representing plasma and tissue fractions unbound. An exemplar concentration-time profile is shown in Fig. 3. In the web application, this is an interactive vector-quality plot which can be downloaded as a .png image. Conversion to a semi-log plot is instantaneous.

Prior to introducing the application in the classroom, its compatibility across devices was analysed by running simulations on a Windows desktop, a Windows laptop, an Apple MacBook, and an iPhone. The simulations ran successfully on all devices tested in advance of the classroom pilot, including on mobile where the application adapts to the width of the device. This not only guarantees

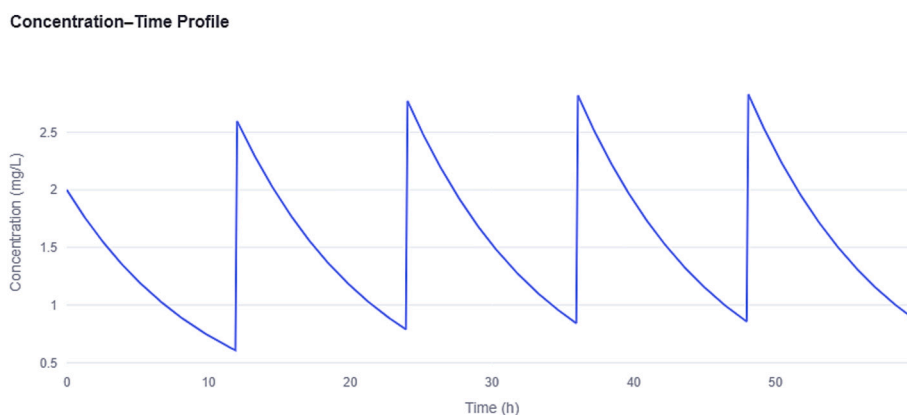


Fig. 3. Simulation output of five repeated bolus doses given at 12-h intervals with PK-SIMple. The resolution shown above reflects the resolution of the plot should the student choose to download it. The plot as generated in the browser is a vector image. Students additionally can zoom into an area of interest and select points on the plot.

accessibility in the classroom, but also reduces the technical barriers students may encounter when using the tool for revision.

Classroom deployment

PK-SIMple was piloted in 90-min workshops with two groups of 35 third-year undergraduate MPharm students at UCC. The purpose of this pilot was twofold: to demonstrate from a technical standpoint the ability of PK-SIMple to handle real-world concurrent users, and to evaluate the feasibility of integrating the tool more formally in future offerings of the pharmacokinetics module.

The workshops were designed as exploratory learning activities to supplement previously delivered lecture material, with no new theoretical concepts introduced. As such, no advance reading was assigned to students. After a short demonstration, students were provided with three cases to explore independently using PK-SIMple. These included constructing concentration-time curves for an intravenous bolus of benzylpenicillin, comparing the pharmacokinetics of three orally-administered investigational neuraminidase inhibitors, and exploring how different clinical and formulation factors influenced exposure to celecoxib. These cases were mapped to theoretical content already covered in the module, and, as such, no new theoretical material was introduced. Each case was accompanied by selected primary literature, including both preclinical and clinical studies, to build confidence in interpreting primary pharmacokinetic research. The cases were additionally aligned with the pre-existing Learning Outcomes of the module. This alignment was intended to demonstrate how PK-SIMple can be incorporated into an established curriculum without requiring substantial restructuring of content.

Three practicing pharmacists undertaking PhDs who contributed to the application's design (J.D.M., R.C.M., S.V.H.) were available to answer questions and provide support where necessary. The full cases provided to the students, with links to the accompanying primary literature and details of alignment with learning outcomes, are summarised in the Appendix.

Evaluation findings

The work in this paper culminated in a pharmacy student-centred interactive web application that allows users to explore foundational pharmacokinetic concepts with rapidly-generated simulations. This evaluation focuses on the technical performance and practical classroom implementation rather than formal assessment of learning or enjoyment.

Load and stability testing

All 100 requests were successful during the simulated load test, confirming that the application does not reject connections even with 100 simulated users. This indicates that the application can comfortably accommodate large class sizes or concurrent use in different institutions. The response time was 0.738 s and the total duration of the test was 6.32 s, reflecting both speed and efficiency. The simulation time was negligible, including for multiple dosing simulations. Han et al.⁸ report simulation times of five seconds for multidose simulations and 30 s for population pharmacokinetics simulations on their JAVA/R tool. The current software represents a clear improvement over this benchmark, though the functionality of the tools are not directly comparable. PK-SIMple's speed is attributable to the efficiency of numerical computing with NumPy and the caching of static elements of the application's interface. The rapid simulation is essential to the platform's goal of allowing pharmacy students to learn pharmacokinetic relationships through iterative experimentation.

Classroom pilot

No technical issues were reported during the pilot workshops, and simulations were efficient. This was true even when many students were concurrently inputting user-specified metabolic enzymes, which is the most memory-intensive feature of the application. The current pilot confirms that the application could handle 35 users simultaneously adding and editing up to four metabolising enzymes each without any change in responsiveness. PK-SIMple launched successfully across a diverse range of browsers, operating systems, and devices used by students, thus mitigating the need for the session to be run in a PC laboratory. The performance of the application was monitored by the instructor (J.D.M.) through the Streamlit Cloud Dashboard during both sessions and no outages occurred. We found the Streamlit dashboard to be user-friendly and can be monitored by an educator without coding expertise. This confirms that the software was optimised for concurrency and has a low memory footprint. A reliable application is essential to ensure uptake, as technical interruptions are likely to disrupt the flow and frustrate both students and educators. While performance was demonstrated under the conditions tested, scalability beyond this level has not been tested in real-world conditions. As such, the authors recommend that institutions with large student cohorts monitor application performance during its first implementation.

PK-SIMple was used without technical interruption by both instructors and students to implement the case studies listed in Appendix A. These cases reinforced theoretical content which was already covered in the module, and were intentionally chosen to cover both preclinical and clinical pharmacokinetic modelling. We found that a 90-min session was sufficient to cover all three. From an educator's perspective, we believe that PK-SIMple appears to be useful for teaching challenging aspects of pharmacokinetic theory. For example, PK-SIMple enables an increase in the absorption rate constant associated with food intake to be illustrated through concentration-time profiles. Though the aim of this study was not to demonstrate superiority over alternative digital approaches, we identified several practical advantages relevant to its implementation within an existing pharmacokinetics curriculum. The simple design reduces the prior knowledge requirements associated with Excel and PBPK software-based simulations. From our experience, demonstrating the same food effect concept in Excel requires exemplar concentration values for the fasted and fed state to be provided,

the absorption rate constant to be deconvoluted from the elimination rate constant, and then the area under the curve be determined using the trapezoidal rule. We find that this process is laborious and detracts focus from the core pharmacokinetic principles. PK-SIMple therefore places understanding of pharmacokinetic theory and clinical implications as the priority for all students, regardless of digital proficiency. A further example was the CYP inhibition exercise, where, in our experience, adjusting the metabolic activity of a single enzyme within PBPK software is not intuitive for students. PK-SIMple, however, allowed students to iteratively add, remove, or change the metabolic activity of CYP enzymes without the need for a separate interface. Additionally, we found PK-SIMple to be useful for designing figures for lecture materials, as it enables rapid generation of plots which directly align with the numerical values used in case studies. This supports alignment between teaching and workshop materials. These observations are based on educator experience during the pilot, rather than through formal evaluation. In practice, the authors do not believe that implementation at another institution would require substantial restructuring of a pre-existing curriculum. Rather, the case studies here are illustrative examples of how PK-SIMple can be integrated into teaching, and can be adapted or replaced to suit local learning outcomes.

Analysis of educational activity

This report describes the development and technical testing of the instructional application PK-SIMple, and our experience of piloting it with two cohorts of third-year undergraduate students. Despite evidence from the literature suggesting that pharmacy students' understanding of foundational pharmacokinetic concepts is poor² and that they find the subject challenging,^{3,4} didactic lectures remain the dominant mode of instruction in many institutions.¹¹ Although we do not wish to diminish the importance of manual calculation skills in this work, we recognise that some pharmacy students may better understand manual calculations after first exploring computer-generated simulations. The authors believe that PK-SIMple addresses a gap in the pedagogical toolkit by providing a freely available application that can be used without prior training or the need for installation. Given the limited number of publications addressing the digital tool development process in the pharmacy education literature, the scope of the present analysis is the design and implementation of the tool. It further details the lessons learned throughout the process and serves as a case study for other educators who wish to develop similar applications.

Theoretical frameworks and practical considerations in application design

It is well documented that a single classroom can have large variations in student confidence with quantitative tasks.^{17,18} We find this becomes particularly evident when teaching pharmacokinetics, which is the most mathematically challenging course in most pharmacy curricula. A "one-size-fits-all" approach to this subject may therefore fail to challenge those confident with calculations while simultaneously failing to support those who struggle with them. In response to this, PK-SIMple was intentionally designed in accordance with the Universal Design for Learning (UDL) framework,^{19,20} with specific interface and functionality choices aligned with its core principles. We hope that by sharing our design choices we might assist other educators who wish to incorporate UDL principles into digital tool design.

Firstly, by offering differing levels of theory for each of the ADME processes, and by automating key calculations (such as volume of distribution and hepatic extraction ratio) outside of the full simulation, PK-SIMple aligns with the UDL principle of multiple means of engagement. This is in contrast to Excel and PBPK-based approaches we have used previously at UCC, where the required inputs are often fixed and thus, in the authors' opinion, less suitable for a mixed ability classroom.

Similarly, the interface achieves the UDL principle of multiple means of representation by providing the output as both an interactive plot and calculated numerical parameters thereby accounting for diversity in information processing. The output panel was intentionally placed beside rather than below the input panel to enable immediate visual feedback and allow users to explore parameter changes in real time. This additionally supports multiple means of action and expression by providing a means for active, hypothesis-driven testing which is not hindered by barriers such as installation or long simulation times. Autonomy, a key goal of UDL, is further achieved by ensuring compatibility with all browsers and devices and thereby creating a new learning environment for students more comfortable using their mobile phone.

Aside from educational theory, many design choices were guided by practicality. Streamlit allows for information to be inputted in a variety of ways, such as number boxes, dropdown menus, and sliders. While the application initially featured only number boxes for numerical inputs, we found that incorporating sliders where applicable to constrain values to a physiologically meaningful range offered several advantages. Not only does this ensure the simulations remain realistic and reduce the risk of an illegal value being inputted, it may also help users develop an intuitive sense for appropriate values for these numbers.

An educator's perspective on application development

The development process itself was, on one hand, intellectually rewarding but on the other presented some unexpected technical demands. The author responsible for writing the code (J.D.M.) is a pharmacist pursuing a PhD in pharmaceuticals and had previous experience of using Python in a machine learning context. However, they are not a programmer by training and had not used coding previously for instructional design. While this made developing the numerically-focused parts of application design, such as writing the concentration functions, efficient, other considerations which were not anticipated from the outset only became evident as development progressed. To ensure widespread adoption, it is a fundamental requirement for the application to be stable when many users are using it at once. Similarly, the application needs safeguards to stop it from crashing if a student enters an illegal value, such as a negative concentration from which the code would then try to calculate the natural logarithm. The development process extended

beyond the need for numerical accuracy and stability, and the application was reviewed many times by educators for instructional suitability. We advise colleagues developing their own tools to expect but embrace this iterative process, as each round of testing improved the final product.

Open and honest communication regarding the constraints associated with a lightweight web application proved essential to maintain a shared vision of a technically feasible platform for teaching and learning. In our case, a key challenge during development was balancing desired functionality with the need for computational efficiency. While there was an initial “wish list” of potential features which included a therapeutic window function and the ability to display multiple simulations on one plot, incorporating these features risked reducing the responsiveness of the application. Particularly for the initial pilot session, there was a shared decision to avoid implementing features that were memory-intensive and likely to compromise performance. We addressed this by identifying clear priorities, with all authors agreeing that user-defined metabolism was the most important feature. We also focused on elements which had a low-memory footprint. The button to toggle between plasma concentration and the natural logarithm of the plasma concentration, for example, was implemented at minimal computational cost. Furthermore, we would encourage authors to explore existing Streamlit applications beyond the field of pharmacy, as these provide inspiration for interface design and functionality that may be adapted in the context of an educational tool.

One aspect of the application design that was not considered by the authors in advance was the importance of long-term stability. PK-SIMple leverages widely-used Python libraries, including NumPy for scientific computing and Plotly for visualisation. The authors recently observed that the application unexpectedly ceased functioning. The issue was traced to an update in NumPy where the `numpy.trapz` function had been removed in favour of `numpy.trapezoid`. `numpy.trapz` is used by PK-SIMple to calculate the AUC parameter on the output panel and, as such, NumPy's update caused the application to crash. NumPy had provided advance notice of this change and thus this incident could have been anticipated, but it had gone unnoticed during PK-SIMple's development. Though this issue was easily fixed, this experience highlights that stability over time is an important practical consideration for educators developing digital tools. An effective approach to mitigating this risk is to specify the versions of external packages to prevent breaking changes being introduced by package updates. This is crucial in the context of pharmacist education, where unanticipated technical issues can disrupt teaching and reduce confidence in novel teaching tools.

Our experience closely paralleled that of Stephens et al.²¹ who produced an R programme to generate practice questions in a pharmaceutical calculations course. We also believe that it is unrealistic to expect all instructors to learn how to code, but this does not minimise the contribution of subject experts who are much better placed to audit the application. We feel this separation of roles actually streamlined the development of PK-SIMple, as the familiarity of the coder with the application limits their ability to appreciate how a first-time user will interact with it. Stephens et al. note the time commitment required for application development is substantial and in some regards difficult to justify. If a team without any prior coding knowledge wished to develop a Streamlit application, the most significant time commitment would be learning to code for the first time. We are fortunate, however, to already have this expertise in-house. For those with prior Python experience, developing a Streamlit application is straightforward, and there is a very active community of Streamlit developers online. A key difference between pharmaceutical calculations and pharmacokinetics is the availability of alternatives to an in-house tool. In our experience, generating realistic concentration-time profiles in Excel is more laborious than writing classical pharmaceutical calculation questions (e.g. dilutions, unit conversions, etc), while the effort in coding applications for these tasks is likely similar. As such, the time investment associated with developing PK-SIMple can be justified by the marked gain in efficiency over existing approaches. Additionally, we hope that by integrating PK-SIMple into future offerings of the pharmacokinetics course as part of the undergraduate pharmacy programme at UCC we can benefit from this enhanced efficiency in years to come.

Limitations of PK-SIMple

In terms of limitations of PK-SIMple, the application only implements a one-compartment model with first-order absorption and elimination. This was deliberate to minimise computational load and maintain stability under concurrent use. The application can be easily extended to non-compartmental analysis, multicompartment modelling, or non-linear pharmacokinetics if required using the publicly available code. As previously suggested, the application could also be modified to support clinical examples through the addition of therapeutic windows for therapeutic drug monitoring. The enhanced functionality must, however, be balanced against the need for efficiency. While the initial load testing and classroom use suggests robust performance of the current application under moderate concurrent usage, scalability under substantially greater loads at larger institutions has not been characterised in real-world settings. As such, institutions adopting the tool at scale (i.e. >100 concurrent users) should monitor its performance on first use to ensure the operational limits of the application are not exceeded.

A further limitation is that PK-SIMple was only piloted at UCC, and no formal evaluation of user experience or impact on performance was performed. This paper instead serves as a technical proof-of-concept and a means of sharing our experience of digital tool design to equip educators with the guidance needed to create their own applications. Formal evaluation of PK-SIMple is planned at the multi-institutional level.

Summary of analysis

There was a clear gap in the availability of open-access pharmacokinetics teaching software aimed at pharmacy students. Thus, we developed PK-SIMple, a Streamlit-based open-source tool that simulates concentration-time profiles and can be used for instruction without any prior training. The application was integrated into two pilot teaching sessions without technical disruption. Its suitability

for use in large class sizes was verified through successful simulated stress testing. This work demonstrates more broadly that the development of in-house teaching applications by those with some prior coding experience is feasible. We found the process of creating the application to be demanding yet worthwhile. We hope that by describing the technical aspects of the development pipeline, including the design considerations and load testing, we have provided a roadmap for colleagues who wish to create their own bespoke educational technologies. An international, multi-institutional study is planned to evaluate the pedagogical advantages of PK-SIMple over conventional approaches to teaching and learning in pharmacokinetics.

Contribution to literature

This paper contributes a novel, browser-based pharmacokinetics simulation software developed specifically to aid pharmacy student learning. This research addresses an unmet need in the literature for a technical description of the tool development process from conceptualisation to classroom deployment. In doing so, this paper provides a pedagogically-grounded yet practical framework for educators seeking to build their own instructional technologies.

Declaration of competing interest

The authors declare no conflict of interest.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.cptl.2026.102719>.

References

- Dornblaser EK. A needs assessment of pharmacokinetic skills performed on advanced pharmacy practice experiences by student pharmacists. *Curr Pharm Teach Learn*. 2017;9(1):95–100. <https://doi.org/10.1016/j.cptl.2016.08.042>.
- Babey AM, Koenig J, Cunningham M, et al. Evaluating student understanding of core pharmacokinetic concepts. *Eur J Pharmacol*. 2025;990, 177256. <https://doi.org/10.1016/j.ejphar.2025.177256>.
- Hughes GJ, Cassagnol M, Wu W, et al. Modification of a pharmacokinetics course design to improve student performance. *Am J Pharm Educ*. 2014;78(5):93. <https://doi.org/10.5688/ajpe78593>.
- Persky AM, Pollack GM. A hybrid jigsaw approach to teaching renal clearance concepts. *Am J Pharm Educ*. 2009;73(3):49. <https://doi.org/10.5688/aj730349>.
- Charles BG, Duffull SB. Pharmacokinetic software for the health sciences: choosing the right package for teaching purposes. *Clin Pharmacokinet*. 2001;40(6):395–403. <https://doi.org/10.2165/00003088-200140060-00001>.
- Brocks DR, uSIMPK. An excel for windows-based simulation program for instruction of basic pharmacokinetics principles to pharmacy students. *Comput Methods Prog Biomed*. 2015;120(3):154–163. <https://doi.org/10.1016/j.cmpb.2015.04.006>.
- Mehvar R. Effects of simulations on the learning of pharmacokinetic concepts. *Curr Pharm Teach Learn*. 2012;4(4):278–284.e1. <https://doi.org/10.1016/j.cptl.2012.05.008>.
- Han S, Lim B, Lee H, Bae SH. Development of a user-friendly training software for pharmacokinetic concepts and models. *Transl Clin Pharmacol*. 2018;26(4):166. <https://doi.org/10.12793/tcp.2018.26.4.166>.
- Gabrielsson J, Andersson K, Tobin G, Ingvast-Larsson C, Jirstrand M. Maxsim2—real-time interactive simulations for computer-assisted teaching of pharmacokinetics and pharmacodynamics. *Comput Methods Prog Biomed*. 2014;113(3):815–829. <https://doi.org/10.1016/j.cmpb.2013.12.006>.
- Zuna I, Holt A. ADAM, a hands-on patient simulator for teaching principles of drug disposition and compartmental pharmacokinetics. *Br J Clin Pharmacol*. 2017;83(11):2426–2449. <https://doi.org/10.1111/bcp.13357>.
- Hughes GJ, Lee R, Sideras V. Design and delivery of clinical pharmacokinetics in colleges and schools of pharmacy. *Am J Pharm Educ*. 2018;82(9):6430. <https://doi.org/10.5688/ajpe6430>.
- Silva DA, Cor MK, Lavasanifar A, M. Davies N, Löbenberg R.. Using GastroPlus to teach complex biopharmaceutical concepts. *Pharm Educ*. 2022;22(1):336–347. <https://doi.org/10.46542/pe.2022.221.336347>.
- Morningstar-Kywi N, Morris DN, Romero RM, Haworth IS. Teaching of drug disposition using physiologically based pharmacokinetic modeling software: GastroPlus as an educational tool. *Adv Physiol Educ*. 2023;47(4):718–725. <https://doi.org/10.1152/advan.00051.2023>.
- Hoffmann TC, Glasziou PP, Boutron I, et al. Better reporting of interventions: template for intervention description and replication (TIDieR) checklist and guide. *BMJ*. 2014;348(mar07 3):g1687. <https://doi.org/10.1136/bmj.g1687>.
- Harris CR, Millman KJ, Van Der Walt SJ, et al. Array programming with NumPy. *Nature*. 2020;585(7825):357–362. <https://doi.org/10.1038/s41586-020-2649-2>.
- Sievert C. *Interactive Web-Based Data Visualization with R, Plotly, and Shiny*. 1st ed. Chapman and Hall/CRC; 2020. <https://doi.org/10.1201/9780429447273>.
- Conn KM, Birnie C, McCaffrey D, Brown J. The relationship between prior experiences in mathematics and pharmacy school success. *Am J Pharm Educ*. 2018;82(4):6257. <https://doi.org/10.5688/ajpe6257>.
- McCafferty R, Nadithe V. Student performance with standalone calculations course vs combined pharmaceuticals-calculations course. *Am J Pharm Educ*. 2018;82(7):6312. <https://doi.org/10.5688/ajpe6312>.
- Rose D. Universal design for learning. *J Spec Educ Technol*. 2000;15(4):47–51. <https://doi.org/10.1177/016264340001500407>.
- Doyle AJ, O'Toole M, Cassidy D, Condron CM. Universal Design for Learning (UDL) in simulation-based health professions education. *Adv Simul*. 2025;10(1):33. <https://doi.org/10.1186/s41077-025-00361-3>.
- Stephens B, Nguyen VD, Raake SE, Daugherty KK. R' you ready to save time? Using R to facilitate question creation in a pharmaceutical calculations course. *Curr Pharm Teach Learn*. 2023;15(12):1060–1065. <https://doi.org/10.1016/j.cptl.2023.10.009>.