Environmental Science
Water Research & Technology

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Check for updates

Cite this: *Environ. Sci.: Water Res. Technol.*, 2022, **8**, 2289

QSDsan: an integrated platform for quantitative sustainable design of sanitation and resource recovery systems†

Yalin Li, ‡*[ab] Xinyi Zhang, ‡[c] Victoria L. Morgan, §[a] Hannah A. C. Lohman, [c] Lewis S. Rowles, ¶[a] Smiti Mittal, [d] Anna Kogler, [e] Roland D. Cusick, [c] William A. Tarpeh [ef] and Jeremy S. Guest [abc]

Sustainable sanitation and resource recovery technologies are needed to address rapid environmental (*e.g.*, climate change) and socioeconomic (*e.g.*, population growth, urbanization) changes. Research prioritization is critical to expedite the development and deployment of such technologies across their vast system space (*e.g.*, technology choices, design and operating decisions). In this study, we introduce QSDsan – an open-source tool in Python for the quantitative sustainable design (QSD) of sanitation and resource recovery systems. As an integrated platform for system design, simulation, techno-economic analysis (TEA), and life cycle assessment (LCA), QSDsan can be used to enumerate and investigate the expansive landscape of technologies under uncertainty, while considering contextual parameters that are critical to technology deployment. We illustrate the core capabilities of QSDsan through two distinct examples: (i) evaluation of a complete sanitation value chain that compares three alternative systems; and (ii) dynamic process modeling of the wastewater treatment plant described in the benchmark simulation model no. 1 (BSM1). Through these examples, we show the utility of QSDsan to automate design, enable flexible process modeling, achieve rapid and reproducible simulations, and to perform advanced statistical analyses with integrated visualization. We strive to make QSDsan a community-led platform with online documentation, tutorials (explanatory notes, executable scripts, and video demonstrations), and a growing ecosystem of supporting packages (*e.g.*, DMsan for decision-making). This platform can be freely accessed, used, and expanded by researchers, practitioners, and the public alike, ultimately contributing to the advancement of safe and affordable sanitation technologies around the globe.

Received 15th June 2022,
Accepted 26th July 2022

DOI: 10.1039/d2ew00455k

rsc.li/es-water

Water impact

Robust and agile tools are needed to support the research, development, and deployment (RD&D) of sanitation and resource recovery technologies. This work introduces QSDsan – an open-source Python tool that integrates system design, simulation, and sustainability characterization (techno-economic analysis and life cycle assessment) to quickly identify critical barriers, prioritize research opportunities, and navigate multi-dimensional sustainability tradeoffs for technology RD&D.

[a] *Institute for Sustainability, Energy, and Environment, University of Illinois Urbana-Champaign, 1101 W. Peabody Drive, Urbana, IL 61801, USA. E-mail: yalinli2@illinois.edu; Tel: +1 (217) 300 3097*

[b] *DOE Center for Advanced Bioenergy and Bioproducts Innovation, University of Illinois Urbana-Champaign, 1206 W. Gregory Drive, Urbana, IL 61801, USA*

[c] *Department of Civil and Environmental Engineering, 3221 Newmark Civil Engineering Laboratory, University of Illinois Urbana-Champaign, 205 N. Mathews Avenue, Urbana, IL 61801, USA*

[d] *Department of Bioengineering, Stanford University, 129 Shriram Center, 443 Via Ortega, Stanford, California 94305, USA*

[e] *Department of Civil and Environmental Engineering, Stanford University, 311 Y2E2, 473 Via Ortega, Stanford, California 94305, USA*

[f] *Department of Chemical Engineering, Stanford University, 129 Shriram Center, 443 Via Ortega, Stanford, California 94305, USA*

† Electronic supplementary information (ESI) available. See DOI: **https://doi.org/10.1039/d2ew00455k**

‡ Y. Li and X. Zhang contributed equally to this work.

§ Present address: Hazen and Sawyer, 2420 Lakemont Avenue, Suite 325 Orlando, FL 32814, USA.

¶ Present address: Department of Civil Engineering and Construction, Georgia Southern University, 201 COBA Drive, BLDG 232 Statesboro, GA 30458, USA.

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

1. Introduction

With the increasing pace of technology development[1–3] and growing complexity of sustainability challenges,[4–7] there is a need for robust and agile tools to quickly identify critical barriers, prioritize research opportunities, and navigate multi-dimensional sustainability tradeoffs in the research, development, and deployment (RD&D) of technologies.[8–10] This need is particularly pressing for the field of sanitation as it concerns one of the most basic human rights. Ensuring the right to sanitation also directly addresses the sixth sustainable development goal (SDG) proposed by the United Nations (universal sanitation by 2030), and is connected to many other SDGs (*e.g.*, resource circularity, carbon neutrality).[11] To improve sanitation service coverage in resource-limited communities, a portfolio of technologies are needed, including those that do not require large capital investment (*e.g.*, *via* non-sewered sanitation) and that lower costs and environmental impacts through resource recovery.[12,13]

To support technology RD&D, multiple high-fidelity commercial software are available for the design and simulation of water and wastewater systems as well as resource recovery technologies (*e.g.*, GPS-X™,[14] SUMO©,[15] BioWin,[16] WEST,[17] Aspen Plus®,[18] SuperPro Designer[19]). However, these tools primarily focus on conventional, centralized technologies rather than early-stage RD&D of novel, decentralized systems. Additionally, the current approach of segregating system design, simulation, and sustainability characterization into multiple tools creates challenges in execution and maintaining transparency. For example, GPS-X™ can support design and simulation, CapdetWorks[20] can support techno-economic analysis (TEA), and SimaPro[21] can be used for life cycle assessment (LCA). This segregated approach requires data to be organized and transferred between tools, motivating the development of new tools to streamline this workflow.[9] Further, the lack of support for uncertainty and sensitivity analyses often limits the scope of existing studies to a narrow set of simulation inputs (design and/or control decisions, technological parameters, contextual parameters). In contrast, early-stage technologies are characterized by high levels of uncertainty with limited information on these simulation inputs, thus creating a mismatch that undermines the utility of these tools for early-stage technologies. Although features have been included in some software to enable the incorporation of uncertainty to a limited degree (*e.g.*, by allowing advanced simulation settings in programming languages like C# or Python), functionalities beyond batch simulation (*e.g.*, parameter sampling, calculation of sensitivity indices, statistical analysis) still need to be executed externally. Therefore, it remains challenging to perform robust uncertainty and sensitivity analyses with these high-fidelity commercial tools.

Herein, we present QSDsan – an open-source tool that leverages the quantitative sustainable design (QSD) methodology for integrated design, simulation, and sustainability evaluation of sanitation and resource recovery systems.[10] Built under the object-oriented programming (OOP) paradigm using Python (3.8+), QSDsan aims to address the lack of supporting tools for the RD&D of early-stage sanitation technologies. QSDsan is a community-led platform that provides flexible, transparent, and freely accessible modules for modeling and evaluation. With a rich collection of Python libraries and the embracement of open source by the rapidly growing scientific programming community, it has the potential to continuously evolve and advance with emerging sanitation and resource recovery technologies.

The main features of QSDsan include bulk property calculations of waste streams, equilibrium and dynamic process modeling, user-defined unit operation design, automated system simulation, integrated TEA and LCA, and advanced uncertainty and sensitivity analyses with built-in visualization functions. In addition to introducing the underlying structure of QSDsan, we illustrate its usage through two example implementations under different simulation modes (equilibrium and dynamic). In the first implementation (equilibrium mode), we simulate three alternative sanitation systems under uncertainty and characterize their sustainability *via* TEA and LCA, where each alternative includes human excreta input, user interface and onsite storage, conveyance, centralized treatment, and reuse of treated and recovered excreta-derived products.[22,23] In the second implementation (dynamic mode), we evaluate the system described in the benchmark simulation model no. 1 (BSM1),[24,25] which consists of a five-compartment activated sludge reactor and a secondary clarifier. The activated sludge reactor was modeled as two anoxic tanks and three aerobic tanks with the activated sludge model no. 1 (ASM1),[26] and the clarifier was modeled as a 10 layer non-reactive unit.[27] Through these example implementations, we validate the algorithms in QSDsan and highlight its novel capacity to provide insight for technology RD&D. Finally, we discuss how QSDsan can be continuously developed to better contribute to the advancement of sustainable sanitation and resource recovery systems (*e.g.*, by connecting with other tools for decision-making and system optimization).

2. Methods

2.1. Structure and capacities of QSDsan

To enable the modeling of any sanitation technologies and systems, QSDsan leverages the mechanism of "inheritance" in the OOP paradigm in the programming language of Python. The OOP paradigm in Python has two core concepts – "classes" and "instances", both of which can be referred to as "objects". Different classes can be established to provide pre-defined sets of data and/or methods (*i.e.*, functional algorithms), which are collectively referred to as "attributes". For each class, subclasses can be created to inherit and modify these attributes (*e.g.*, a lagoon class can have anaerobic and facultative subclasses). In application,

 This journal is © The Royal Society of Chemistry 2022

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

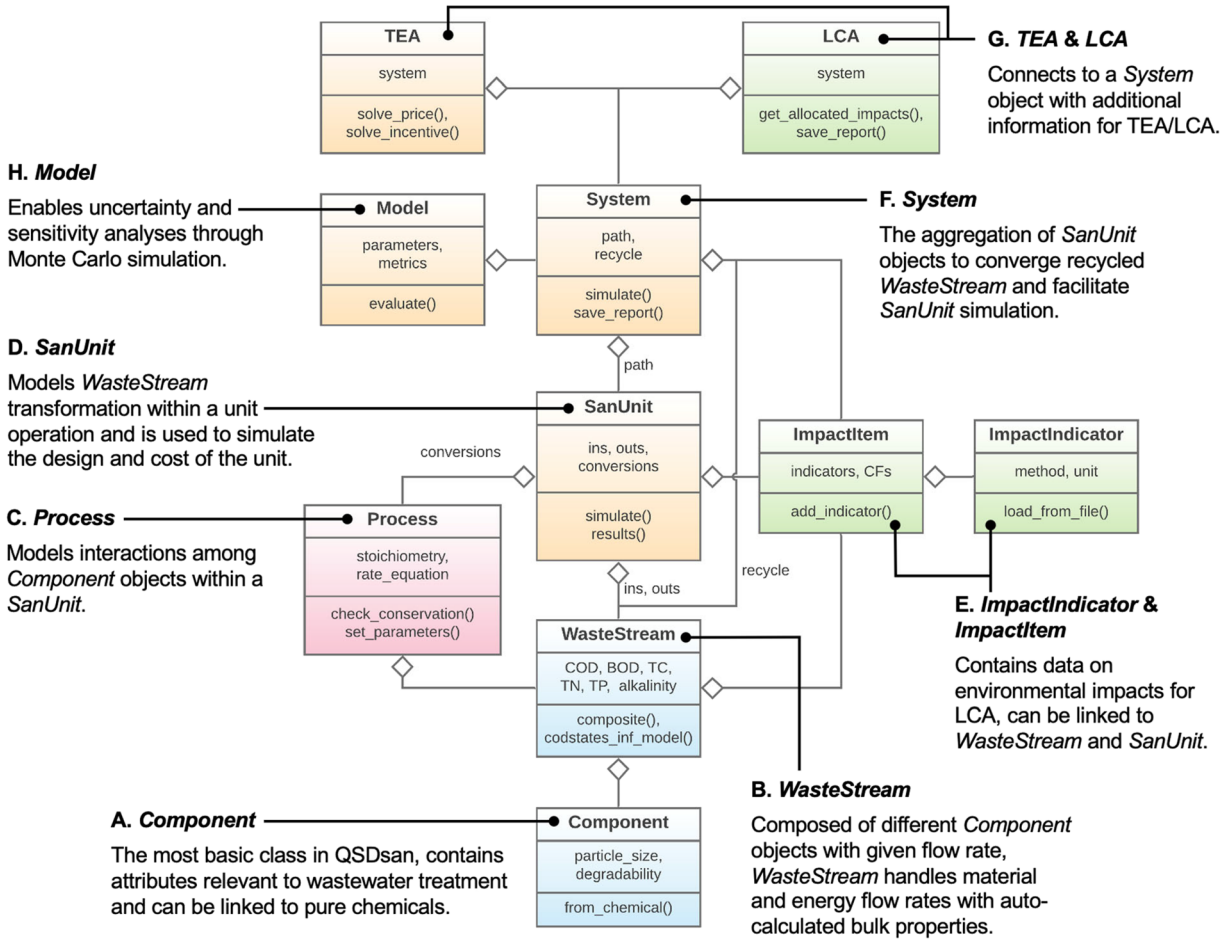


Fig. 1 Simplified unified modeling language (UML) diagram showing the structure and core Python classes implemented in QSDsan. Each class is represented by a box containing the class name (bold, top part of the box) with select data (middle part of the box) and method (end with parentheses, bottom part of the box) attributes. Letters A–H represent the class hierarchy from lower (*i.e.*, more fundamental) to higher (*i.e.*, more advanced) levels. The *Component* and *WasteStream* classes in blue are inherited from the *Chemical* and *Stream* classes in Thermosteam[30,31] with the addition of wastewater-related attributes. The *Process* class in red enables dynamic simulation of *Component* objects' transformation during kinetic processes (*e.g.*, degradation of substrates). The *SanUnit, System, TEA,* and *Model* classes in yellow are inherited from BioSTEAM[28,29] with added capacities for dynamic simulation and handling of construction inventories. Green boxes including *ImpactIndicator, ImpactItem,* and *LCA* are implemented in QSDsan to enable LCA functionalities. Refer to section 2.1 for detailed explanation on the structure and capacities of QSDsan.

"instances" of classes are created in the systems established by the user, where these "instances" are the actual implementation of the classes with inherited attributes. For example, users can design a system with any number of anaerobic lagoons, and each of these lagoons will be an instance of the anaerobic lagoon class. Using the OOP paradigm, QSDsan inherits pertinent classes in existing tools (*e.g.*, BioSTEAM[28,29] and Thermosteam[30,31]), while adding a range of capacities that are critical for sanitation and resource recovery applications (*e.g.*, wastewater property calculation, dynamic process modeling; Fig. 1). With this structure, QSDsan enables a rigorous, automated, and integrated workflow of design, simulation, and sustainability characterization (explained in following sections).

2.1.1. Tracking mass and energy flows. When using QSDsan, users start with creating *Component* objects (italic words denote modules, classes, or attributes in Python hereinafter), which contain attributes relevant to wastewater treatment (*e.g.*, nitrogen content, total suspended solids, chemical oxygen demand [COD]-to-mass ratio). *Component* objects can be linked to pure chemicals (*e.g.*, acetic acid) in databases[30–33] to enable thermodynamic property calculation and simulation (*e.g.*, density, phase equilibrium and transition). Alternatively, users can select from the built-in set of typical components in wastewater treatment modeling,[34] or create new ones from scratch (*i.e.*, set all properties manually). Regardless of how the *Component* is created, users would be able to tailor it to the needs of the RD&D application.

With *Component* objects, *WasteStream* objects can be created to handle mass and energy flows by tracking the quantity, phase, temperature, and pressure of individual components in a stream (*e.g.*, influents and effluents of a unit operation). With this mass and energy flow information, bulk properties of the stream can also be calculated with embedded algorithms (*e.g.*, the concentration of volatile

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

suspended solids). A *WasteStream* object can be created by either (i) defining the flowrates of individual *Component* objects, or (ii) through established influent characterization models with the default component set (*e.g.*, based on the total COD and COD fractions).[35]

Kinetic interactions among components are captured using *Process* objects, which store data on stoichiometries and rate equations. The *Process* class is equipped with algorithms for automatic calculations of unknown stoichiometric coefficients based on conservation of materials (*e.g.*, carbon, nitrogen, COD, charge). With multiple *Process* objects, kinetic models (*e.g.*, ASM1 (ref. 26)) can be compiled and used in reactor models to describe the rates of change of state variables as ordinary or partial differential equations (ODEs or PDEs).

2.1.2. Design and simulation of unit operations and systems. The *SanUnit* class is used to design and simulate unit operations (*e.g.*, a bioreactor). Influents and effluents of a *SanUnit* object are represented by *WasteStream* objects. Transformation of *Component* from influent to effluent *WasteStream* can be modeled in either equilibrium (by defining conversions) or dynamic (through *Process* objects) mode. Design (*e.g.*, reactor height, volume), cost (*e.g.*, capital, operational), and utility usage (*e.g.*, heating, cooling) of *SanUnit* objects are stored as attributes of *SanUnit*. These attributes can be fixed at certain values or calculated with respective algorithms using inputs from *WasteStream* (*e.g.*, calculate reactor volume based on flowrate and retention time; calculate reactor cost based on the design dimensions and selected materials) and/or *Process* (*e.g.*, calculate electricity usage based on the air flowrate modeled by the aeration *Process*). In essence, the *SanUnit* class is an "umbrella" class containing generic algorithms (*e.g.*, total cost equals the cost of all equipment within this unit), where subclasses of *SanUnit* can be created with specific algorithms for different technologies (*e.g.*, calculation of the friction head for a pump).

SanUnit objects can be connected by *WasteStream* objects and aggregated into *System* objects. In the equilibrium mode, mass and energy of *SanUnit* influents and effluents (represented by *WasteStream*) are converged at the given conditions. In the dynamic mode, ODEs representing the accumulation rate of components in each *SanUnit* are compiled into *System*-wide ODEs and integrated from the initial conditions over the desired period. After convergence of the *System*, design algorithms provided by the user are simulated to update the unit costs and system inventory (including chemical and material usage as well as emissions and wastes), which are used in TEA and LCA (discussed in the following section).

2.1.3. Performing TEA and LCA. With the established *System* objects, cost analysis can be performed through the *TEA* class with additional user inputs (*e.g.*, income tax, discount rate). Upon simulation, capital and operating costs (including materials, utilities, labor, and maintenance) of each *SanUnit* are determined by the cost algorithms of that specific unit, and the cost of the *System* will be calculated as the sum of the costs of all *SanUnit* within the *System*. With these cost data, algorithms included in the *TEA* class can be used to calculate cost and profitability indicators such as net earnings and payback period. Discounted cashflow rate of return analysis can then be performed to calculate rate of return, net present value, and other indicators of interest. Further, users can modify the *TEA* class to include algorithms for other indicators of interest.

Similarly, LCA is performed through the *LCA* class with the auto-generated inventory from *SanUnit* objects within the *System*, but two more classes – *ImpactIndicator* and *ImpactItem* – are needed when using the *LCA* class. *ImpactIndicator* carries information on the impact indicators of interest (*e.g.*, kg CO_2 equivalents for global warming potential, kg N equivalents freshwater eutrophication), and they should be selected based on the desired life cycle impact assessment (LCIA) methodology (*e.g.*, TRACI,[36] ReCiPe[37]). These indicators are then stored as attributes of *ImpactItem* objects, and these *ImpactItem* objects are used to represent inventory items such as construction materials, chemical inputs, and waste emissions. For each of the added indicators, *ImpactItem* objects also store the corresponding values of characterization factors (per functional unit of the impact item), which can be used to connect the foreground system inventory simulated by QSDsan with the background life cycle inventory from databases. Users can retrieve the life cycle inventory data and characterization factors externally (*e.g.*, from databases such as ecoinvent and/or existing literature) and code them in the script, or organize the data into a spreadsheet and import them into QSDsan. Additionally, through external packages Brightway2 (ref. 38) and BW2QSD,[39] users can directly retrieve the data from supported databases (see the example of retrieving data from ecoinvent in EXPOsan[40]) and use them in QSDsan. Moreover, *ImpactItem* can be linked to *WasteStream* and *SanUnit*, thereby enabling automatic updates of impact item quantities upon system simulation (*e.g.*, CO_2 emitted during operation, cement required in construction). In addition, *ImpactItem* objects can also be created in isolation with user-defined functions for automatic updates of item quantity upon simulation (*e.g.*, consider system-wise electricity usage). Similar to system design and process simulation, results of TEA and LCA (*e.g.*, total and breakdown of costs and environmental impacts) are accessible in the Python environment for further processing and can be output as data spreadsheets.

2.1.4. Executing uncertainty and sensitivity analyses. Finally, *Model* objects created in association with the *System* object of interest can be used to incorporate uncertainties in the system's design, simulation, TEA, and LCA through the Monte Carlo method.[41] Two core attributes of the *Model* class – *Parameter* and *Metric* – are used for this purpose. *Parameter* objects are used to specify the input parameters with uncertainty (*e.g.*, a component's nitrogen content, kinetic parameters of a process model, retention time of a reactor, chemical price, impact item characterization factor), and *Metric* objects are used to specify the output variables to be

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

evaluated (*e.g.*, effluent quality, total cost, environmental impacts). To perform an uncertainty analysis, users firstly select the desired sample size and sampling method (*e.g.*, random,[42] Latin hypercube[43]), then the *Model* object will generate a sample matrix using the probability distributions defined *via* the *Parameter* objects (*e.g.*, uniform, triangular). System simulation, TEA, and LCA are subsequently carried out for each sample within the sample matrix, and the result metrics defined by the user *via Metric* objects are recorded for later analyses.

Further, leveraging external Python libraries (*e.g.*, SALib,[44] Matplotlib,[45] seaborn[46]), QSDsan also includes a *stats* module with a wide range of global sensitivity analysis methods (*e.g.*, Spearman rank correlation, Morris one-at-a-time technique,[47] the Sobol method[48]) and visualization functions. All input parameters (*i.e.*, sample matrix), output results (*i.e.*, metrics), and generated figures can be accessed in Python and saved externally for additional processing.

2.2. Illustration of QSDsan applications

2.2.1. Evaluation of a complete sanitation value chain. To illustrate QSDsan's capacity in system design, simulation, TEA, and LCA, a complete sanitation value chain with three alternative systems was implemented using QSDsan. Details on the three systems can be found in Trimmer *et al.* (Fig. 2, top panel).[22] Briefly, all three systems included user interface, onsite storage, conveyance, centralized treatment, and reuse (land application and/or biogas combustion) units. In system A, pit latrines were used as the user interface and onsite storage, followed by transportation *via* tanker trucks to the existing treatment plant (sedimentation, anaerobic lagoon, facultative lagoon, and unplanted drying bed), and the recovered nutrients (N, P, and K) were sold as fertilizers. For system B, the same pit latrine and tank trucker transportation were used, but an anaerobic treatment plant (anaerobic baffled reactor, liquid treatment bed, and unplanted and planted drying beds) was modeled. Biogas recovered from the anaerobic baffled reactor was assumed to be sold as a cooking fuel (as a replacement for liquid petroleum gas). Lastly for system C, container-based toilet and storage facilities were used, which included urine-diverting dry toilets (UDDTs), urine storage tanks, and feces dehydration vaults. The containers were assumed to be collected through handcarts and transported by tanker trucks to the

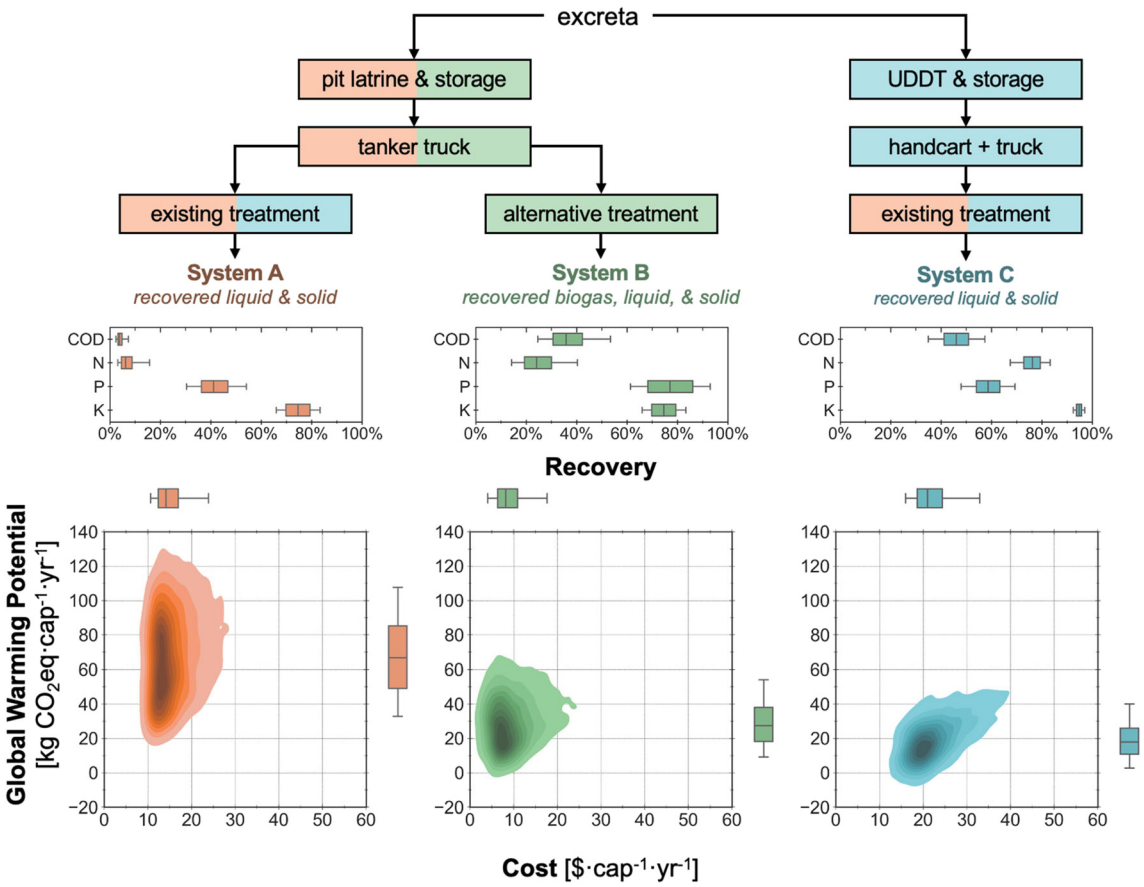


Fig. 2 (Top) Simplified process diagrams, (middle) COD and nutrient (N, P, and K) recoveries expressed as percentages of the initial inputs in the excreta, and (bottom) net annual cost and annual global warming potential (GWP) for the three sanitation systems as described in Trimmer *et al.*[22] For the box plots in the middle and bottom panels, middle line, edges, and whiskers are 50th, 25th/75th, and 5th/95th percentiles, respectively. Systems A, B, and C are color-coded in orange, green, and blue, respectively, in all panels.

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

treatment plant. For treatment and reuse, the same treatment and reuse processes as in system A was used, with the only exception being that the sedimentation unit was eliminated because solids were already separated from liquid in the UDDT. Solids recovered from all three systems were assumed to be used for land application.

Six indicators (*i.e.*, metrics) were included to evaluate the performance of the three systems: total recoveries of COD, N, P, and K, as well as annual user cost and emission. The recoveries were calculated using the mass flow data obtained upon system simulation and reported as the percentage of the excreted COD or nutrients; the annual user cost was calculated by normalizing the annualized net cost (including amortized capital cost) to a *per capita* basis and reported in $ per cap per year; and the emission (from the construction and operating of the system) was calculated by normalizing the annualized net global warming potential (GWP) and reported in kg CO_2eq per cap per year. For this study, TRACI was chosen as the LCIA method and the background inventory data were retrieved from ecoinvent to be consistent with the literature,[22] but example usage of external packages for direct data import from ecoinvent was also included in the module with ReCiPe as the LCIA method.

For the uncertainty analysis, probability distributions of the uncertain parameters followed those described previously.[22,23] A total of 137, 133, and 122 parameters were varied for systems A, B, and C, respectively. Latin hypercube sampling was used to generate 5000 samples (*i.e.*, sets of input data) for the simulation of each system. To enable pair-wise comparisons of results in the uncertainty analysis, in each sample, a parameter was assigned the same value across all systems that share this parameter (*e.g.*, the same pit latrine emptying time was used for both systems A and B, the same user caloric intake was used for all systems).

To validate QSDsan's structure and algorithms in equilibrium simulation, TEA, and LCA, a global sensitivity analysis was performed to provide additional insight on the key drivers of system sustainability. The analysis was performed using the Morris one-at-a-time technique[47] with 50 trajectories. Each trajectory represents one set of simulations that yield one evaluation of each parameter's "elementary effect" on the model outputs (*e.g.*, the change in the annual user cost of the system caused by the change in pit latrine emptying period, with other parameters fixed at certain values). In Morris analysis, the total number of simulations for one system (N_{Morris}) is:

$$N_{\mathrm{Morris}} = n_{\mathrm{trajectory}} \times (k + 1) \tag{1}$$

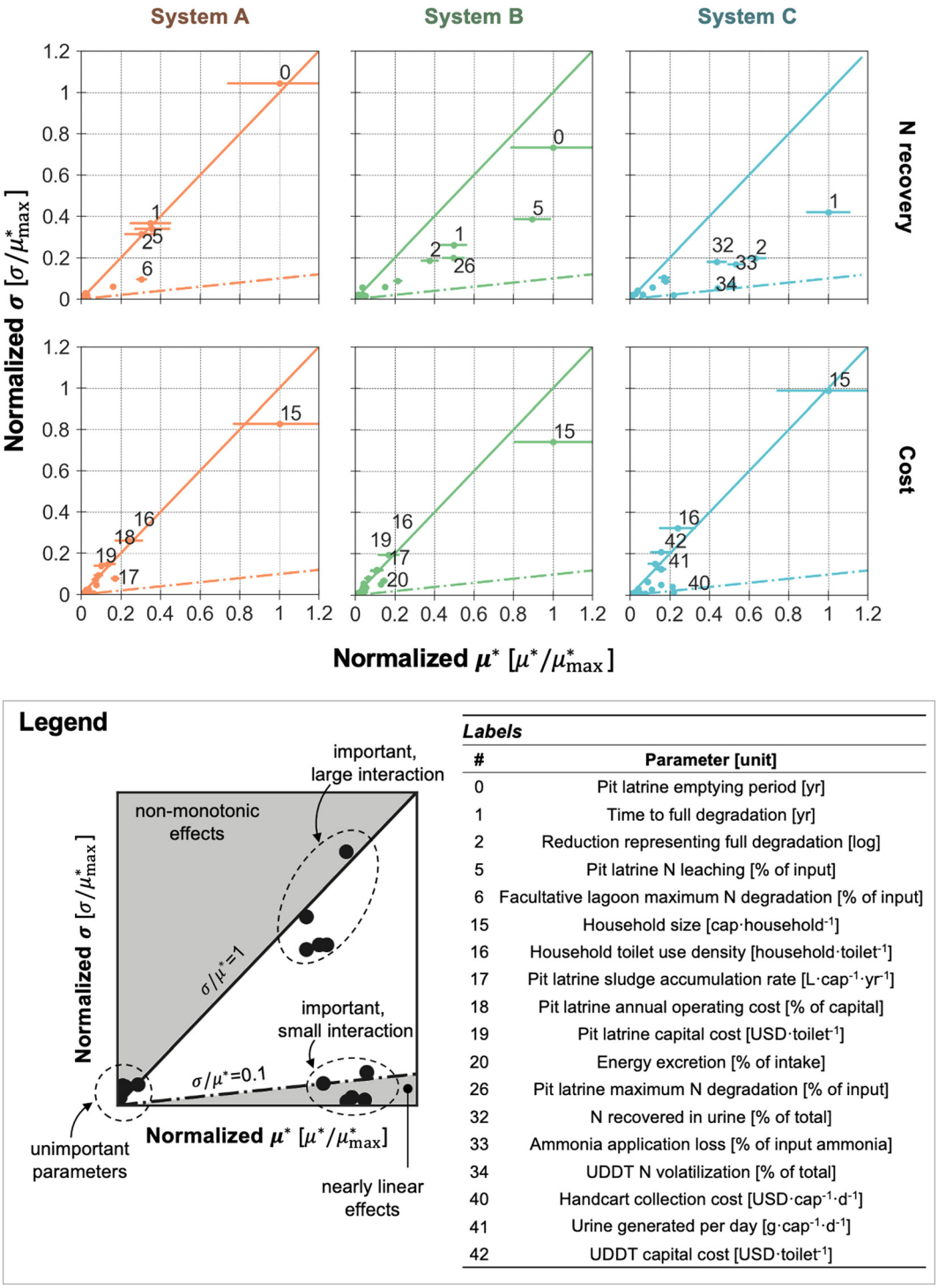
where $n_{\mathrm{trajectory}}$ is the number of trajectories and k is number of parameters with uncertainty. Therefore, 6900, 6700, and 6150 simulations were performed for systems A, B, and C, respectively. Results of the Morris analysis were reported as μ^* and σ values, which indicate the mean and the variance, respectively, of the evaluated "elementary effects" across trajectories. The μ^* and σ values of each parameter were normalized by the largest μ^* value of all input parameters for a particular indicator (*e.g.*, cost) of one system, which allows the results across the six indicators in the three systems to be presented on the same scale.

Uncertainty and sensitivity analyses were performed and visualized using the *stats* module in QSDsan (except for minor annotation of the QSDsan-generated figures to improve readability). Source codes of the three systems, scripts used to conduct uncertainty and sensitivity analyses and plotting, results and figures (generated with QSDsan v1.1.3 and EXPOsan v1.1.4), and a brief instruction, can be found in the online repository EXPOsan (the central location for systems developed using QSDsan).[40]

2.2.2. Process model benchmarking. To validate and demonstrate QSDsan's process modeling and dynamic simulation algorithms, the BSM1 system[24] was modeled with QSDsan following identical assumptions as the original MATLAB/Simulink implementation (Fig. 4A).[25,49] The BSM1 system consists of a five-compartment activated sludge reactor followed by a flat-bottom circular clarifier. The activated sludge reactor was modeled as five CSTRs in series, and a constant influent represented by a *WasteStream* object was used as the feed to the system. Eight *Process* objects were created to describe ASM1's kinetic processes[26] in all five CSTRs, which consisted of two anoxic reactors followed by three aerobic. Two additional *Process* objects were created to represent the diffused aeration processes in the three aerobic CSTRs. A simple one-dimensional 10 layer settling model[27] was built into the clarifier to model particulate components, whereas soluble components were modeled as if in a non-active CSTR (*i.e.*, one layer). Two recycle streams were included in the system, with one from the last aerobic CSTR to the first anoxic CSTR (*i.e.*, the internal recycle), and the other from the clarifier to the first anoxic CSTR (*i.e.*, the return activated sludge, RAS). To validate the dynamic simulation algorithms in QSDsan, the same parameter values as in the IWA MATLAB/Simulink implementation[25] were used as the baseline (future users can modify these values based on their needs). A simulation period of 50 days was selected, where steady state could be achieved for both implementations. In addition to the baseline validation between QSDsan and the published MATLAB/Simulink results, the BSM1 system was also tested with 100 different initial conditions in QSDsan, which were generated through Latin hypercube sampling from uniform distributions centered around the baseline initial concentrations (±50%).

Further, to inform system operation, a complete Monte Carlo simulation was performed, followed by Monte Carlo filtering to identify the top two influential decision variables affecting the effluent quality. A total of 28 uncertain parameters (including seven design and operational control decision variables) and seven metrics were included in the analysis, and a sample size of 1000 was used. The quality of the effluent was subsequently simulated across the full range of value combinations for the top two drivers (*i.e.*, across both their ranges of possible values) to visualize system

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.



#	Parameter [unit]
0	Pit latrine emptying period [yr]
1	Time to full degradation [yr]
2	Reduction representing full degradation [log]
5	Pit latrine N leaching [% of input]
6	Facultative lagoon maximum N degradation [% of input]
15	Household size [cap·household^{-1}]
16	Household toilet use density [household·toilet^{-1}]
17	Pit latrine sludge accumulation rate [L·cap^{-1}·yr^{-1}]
18	Pit latrine annual operating cost [% of capital]
19	Pit latrine capital cost [USD·toilet^{-1}]
20	Energy excretion [% of intake]
26	Pit latrine maximum N degradation [% of input]
32	N recovered in urine [% of total]
33	Ammonia application loss [% of input ammonia]
34	UDDT N volatilization [% of total]
40	Handcart collection cost [USD·cap^{-1}·d^{-1}]
41	Urine generated per day [g·cap^{-1}·d^{-1}]
42	UDDT capital cost [USD·toilet^{-1}]

Fig. 3 Morris sensitivity indices (μ^* and σ) of the key parameters for N recovery and cost of the three systems. Each plot represents the indices calculated for one indicator of a system. Error bars represent the 95% confidence interval of μ^*. For illustration purpose (*i.e.*, to use the same *x* and *y* axes scales across different indicators and systems), values of μ^* (*x*-axis) and σ (*y*-axis) were normalized by the maximum μ^* of all parameters in the analyzed system for the analyzed indicator. Key parameters were defined as parameters with a normalized μ^* value greater than or equal to 0.1 ($\mu^*/\mu^*_{max} \geq 0.1$). If there were more than five parameters meeting this criterion, only the top five parameters with the largest normalized μ^* values were considered as key parameters. All parameters were included in the plot, but only the key parameters were labeled (parameters with small normalized μ^* and σ values clustered at the origin point are thus indistinguishable). In all plots, the solid and dashed lines have slopes of 1 and 0.1 (*i.e.*, σ/μ^* = 1 and 0.1), respectively, where the parameters on the left side of the solid line were considered to have non-monotonic effects on the indicator values, and parameters on the right side of the dashed line were considered to have linear effects. A compiled figure with all indicators is included as Fig. S1 (scatter plots as this figure) and S2 (bubble plot) in the ESI;† all generated raw data and figures are available online.[40]

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

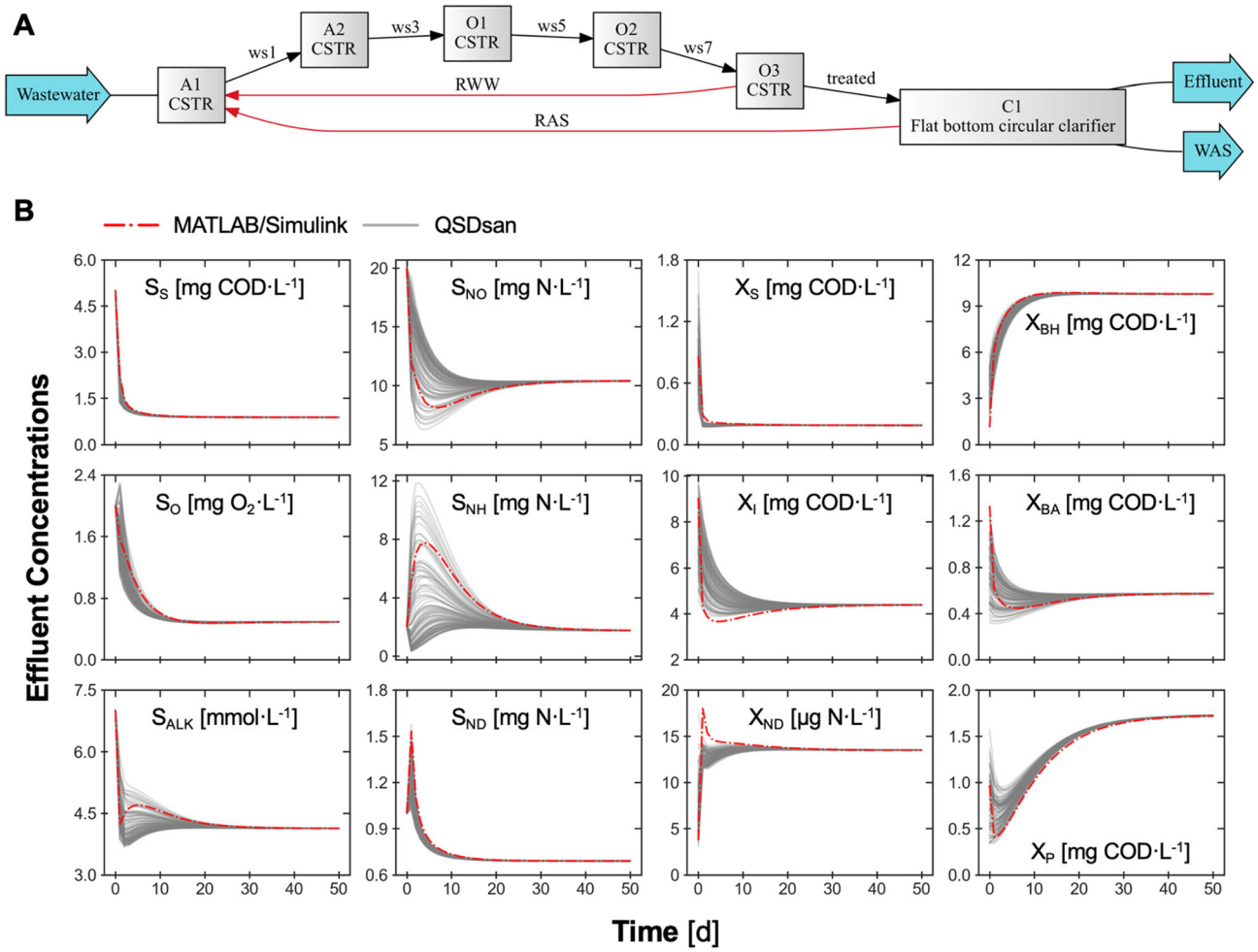


Fig. 4 (A) Configuration of the BSM1 system generated by QSDsan. A1 and A2 represent the anoxic reactors; O1, O2, and O3 represent the aerated reactors. RWW – return wastewater (*i.e.*, the internal recycle); RAS – return activated sludge; WAS – waste activated sludge. (B) Dynamics of BSM1 system's effluent state variables simulated by QSDsan (with 100 different initial conditions) and MATLAB/Simulink (with the baseline initial condition). Soluble inert organic matters, S_I, is omitted in this figure because it is not involved in any conversion processes, as defined in ASM1.

performance across the decision space. Specifics of simulation settings for the analyses above can be found in the ESI.† Complete scripts of BSM1 implementation in QSDsan, results and figures (generated with QSDsan v1.1.3 and EXPOsan v1.1.4), and a brief instruction can be found in the EXPOsan repository.[40]

3. Results and discussion

3.1. Groundwork toward an open and community-led platform

The core structure of QSDsan has been completed and released on the Python package index (PyPI) repository.[50] All of QSDsan's source codes and documentation are available online,[50,51] and the documentation allows the users to access instructions online or in Python as they are using QSDsan (*i.e.*, similar to a "help" function). Tutorials with step-by-step instructions from the installation of Python to the use of QSDsan have also been included in the documentation, and a complementary YouTube channel has been created with relevant video resources (*e.g.*, tutorial demonstrations, workshop recording).[52]

To date, several widely used process models including ASM1,[53] ASM2d,[53] and anaerobic digestion model no. 1 (ADM1)[54] have been added to QSDsan and verified against literature[53,55] and/or established implementations in other platforms (*e.g.*, MATLAB[25,56] and GPS-X™). A range of basic reactor configurations (*e.g.*, continuous stirred tank reactors, CSTRs), as well as conventional (*e.g.*, the activated sludge process) and emerging (*e.g.*, anaerobic membrane bioreactor) technologies have been implemented in QSDsan. Additionally, a growing number of sanitation and resource recovery systems (*e.g.*, the Biogenic Refinery,[57] the Reclaimer system[58]) have been developed and included in the EXPOsan repository. Lists of developed process models,[59] unit operations,[60] and systems[61] can be found in QSDsan's documentation page with links to the source codes. Continuous development and maintenance of QSDsan will be supported by members of the QSD group with regular updates and addition of new technologies and systems as

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

researchers and practitioners leverage the software and share their codes.

We also welcome adoption and contributions from the community (*e.g.*, as we have observed for our existing tools with >15 500 downloads in the first two years of the publication[28–31]), and we adhere to the Contributor Covenant Code of Conduct[62] for a collaborative environment. To encourage external contributions, QSDsan's documentation also includes a special section on contribution instructions and guidelines.[63] Briefly, the first contribution from a community developer will be on that developer's own "fork" (*i.e.*, copy) of QSDsan, and the developer can submit a "pull request" to QSDsan's root repository (hosted by the QSD group) to incorporate the changes into QSDsan's stable version. The pull request will be accepted if the developer's fork (i) contains meaningful contributions with documentation, (ii) has no conflicts with the root repository, and (iii) has passed all test modules to avoid compromising QSDsan's existing functionalities. Examples for such contributions can be found on QSDsan's GitHub repository (*e.g.*, for a chlorination process[64]). After the first contribution, the developer will be invited to join the QSD group and given writing access to the root repository.

3.2. Simulating a complete sanitation value chain

3.2.1. Uncertainty analysis of sanitation alternatives. QSDsan can be used to simulate complete sanitation value chains and characterize their sustainability *via* TEA and LCA, thereby providing guidance when navigating tradeoffs among engineering performance metrics (*e.g.*, nutrient recoveries) and sustainability indicators (*e.g.*, cost, global warming potential, eutrophication potential) among alternative systems. In this example implementation (Fig. 2), all three systems were able to recover the majority of the K in the excreta. Through urine separation *via* UDDT, system C was able to recover significantly more N than systems A and B. In contrast, system B achieved the highest P recovery as it avoided the loss of P to settled solids in system A's treatment processes and the precipitation of P (as struvite) in system C's decentralized urine storage unit. For COD recovery, systems B and C were able to recover more COD for beneficial use due to the generation and capturing of biogas (system B) or less degradation of organics in the sludge (system C). As for the user cost and GWP, system B was the most affordable system because of the relatively inexpensive facilities and the revenue from selling biogas, while system C had the highest user cost due to the more expensive UDDT and the higher operating cost (driven by the more frequent waste collection). However, GWP of system C was the lowest among the three alternatives because of the mitigation of fugitive emissions (less CH_4 and N_2O from organic degradation) and the offset of commercial fertilizers by recovered nutrients.

Overall, these results are consistent with the previously published analysis, validating the core functionality of QSDsan (a summary spreadsheet including full comparison of the results is available through the online EXPOsan repository[40]). Additionally, though the systems were modeled following the assumptions in the reference for validation,[22] users can easily tailor the systems to their needs (*e.g.*, update parameter values, modify system structures, use different LCIA methods).

3.2.2. Global sensitivity analysis of sanitation alternatives. To identify the drivers of system sustainability and understand the interactions between uncertain parameters, a global sensitivity analysis was performed using QSDsan's built-in functions. In this illustration, the Morris technique was chosen because of (i) its effectiveness toward screening the most impactful parameters with relatively small sample sizes, which is particularly relevant for the current example with large numbers of uncertain parameters (137, 133, and 122 parameters for systems A, B, and C, respectively); (ii) its ability to provide insights on the interactions among parameters; and (iii) its applicability to nonlinear and non-monotonic (*i.e.*, a model output can both increase and decrease with a model input, depending on the input value) system models. Results from the Morris analysis are commonly reported as two sensitivity indices, μ^* and σ. For a particular indicator, parameters with larger μ^* values are considered to have a more significant effect on the indicator than other parameters. A larger σ indicates the parameter having a nonlinear effect on the results and/or stronger interactions with other parameters (*i.e.*, the effect of this parameter on the result has a stronger dependency on the values of other parameters). Moreover, a parameter with a σ-to-μ^* ratio (σ/μ^*) greater than 1 is usually considered as having a non-monotonic effect on the results, whereas a parameter with $\sigma/\mu^* < 0.1$ is considered as having a nearly linear effect on the results.[65,66]

Results from the Morris analysis revealed different trends of the effect of key parameters on systems and indicators (Fig. 3 and S1 and S2 in the ESI†). For COD and N, recoveries for systems A and B were controlled by parameters associated with the pit latrine, and these parameters were found to have stronger interactions (*i.e.*, larger σ/μ^* values) than those of system C. In systems A and B, the pit latrine emptying period (*i.e.*, the time between two emptying events) was the most significant driver for COD and N recoveries, but its impacts were realized in combination with other parameters (*e.g.*, maximum degradation, the time to reach full degradation, the reduction at full degradation). For system C, however, because of the much less organic degradation in UDDT, the recoveries of COD and N were controlled by parameters associated with other units within the system (*e.g.*, drying bed retention time and maximum degradation), and the effects of these parameters were less prominent. For P, key parameters for systems A and B were related to leaching or the treatment processes (*e.g.*, P retention in the sedimentation tank, P removal in the lagoon), whereas for system C, dietary parameters (*e.g.*, P intake, P and Ca content in urine) were critical due to the potential precipitation

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

reactions in the urine storage tank that could lead to the loss of P. In the case of K, due to its better retention through the systems (as observed in the uncertainty analysis), its recovery was mostly sensitive to the amount lost to leaching (for pit latrines in systems A and B) or handling during conveyance and application of the recovered nutrients (for system C).

Regarding the annual cost, household size had the largest normalized μ^* and σ values compared to other parameters across all systems, indicating that it was the main driver of the overall system cost, and its impacts on the cost were realized in combination with other parameters. This is because the household size directly determined the number of toilets needed for the system, which affect the costs and GWP of not only the user interface units (*i.e.*, pit latrine or UDDT) but also downstream units (*e.g.*, storage and conveyance units). As the number of toilets was also dependent on the total population and household toilet use density (number of households served by a toilet), a larger interaction effect was observed (*i.e.*, larger σ value).

For GWP, household size remained the most impactful parameter for system C. However, for systems A and B, the percentage of caloric intake diverted to the excreta had the largest effect as it contributed to direct CH_4 emission (from the degraded organic matter in the excreta), which was the largest contributor to GWP as observed in the uncertainty analysis.

Finally, systems A and B – which have the same toilet, storage, and conveyance units, but with different treatment processes – had much more similar patterns of μ^* and σ (*e.g.*, σ/μ^* values, key parameters) than systems A and C (same treatment process, different toilet, storage, and conveyance units). These patterns in μ^* and σ values reveal that the selection of toilet (and therefore storage and conveyance units) was more impactful to the sustainability of the sanitation value chain than the choice of treatment process. Overall, this example illustrates QSDsan's capacity in system design, simulation, TEA, LCA, as well as the utility of its statistical module in carrying out integrated uncertainty and sensitivity analyses and providing visualization tools to guide the RD&D of technologies.

3.3. Benchmarking a pseudo-mechanistic process model

3.3.1. Dynamic simulation of BSM1. Simulation results of the BSM1 system validate QSDsan's capacity to implement complex process models with simulation speeds comparable to existing process simulation options (Fig. 4B). On a personal computer with an Intel Core i7-6700K CPU@4.00 GHz and 16.0 GB RAM, it took about 4 to 6 seconds for QSDsan to initialize system state, compile ODEs, and perform a 50 day simulation of BSM1 with any implicit ODE solver readily available in the SciPy

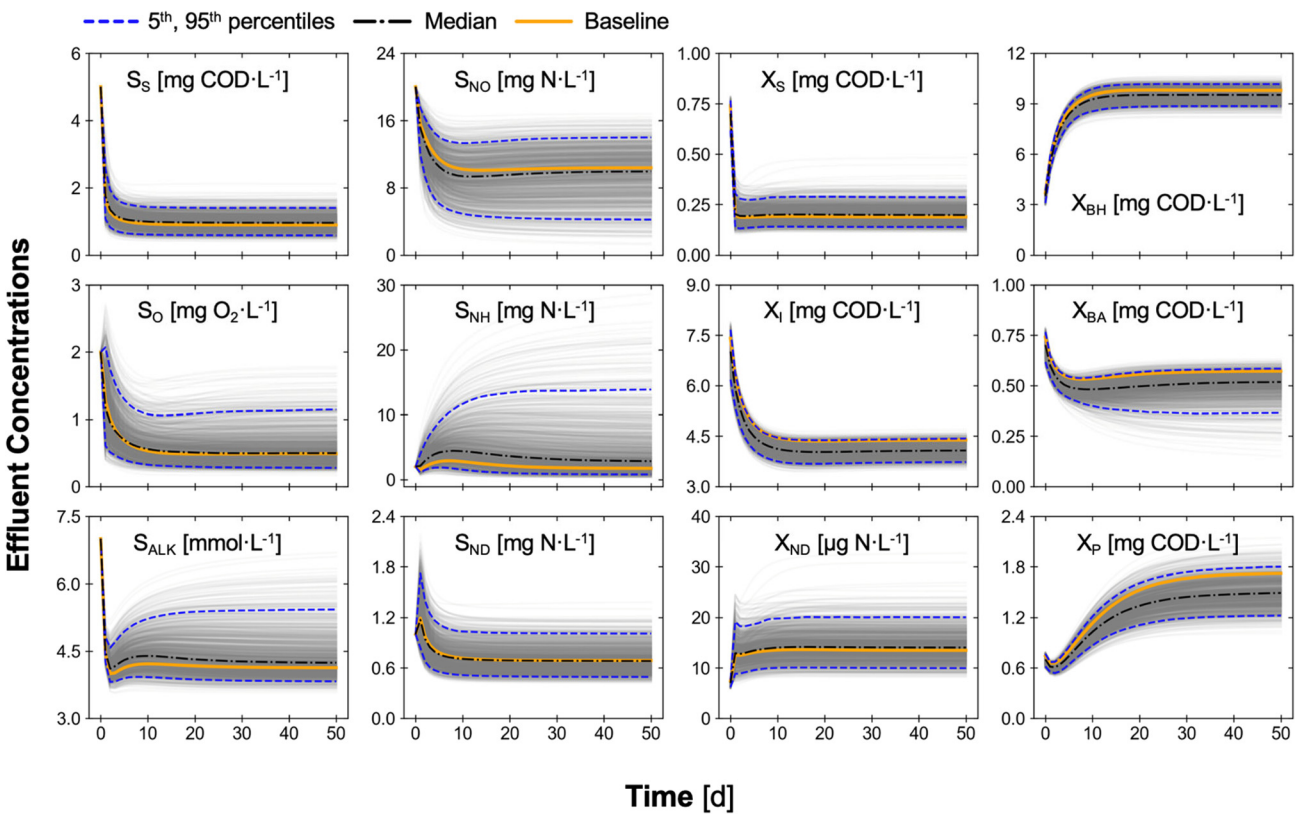


Fig. 5 Dynamics of the effluent state variables generated in Monte Carlo simulations of the BSM1 system for uncertainty analysis. Soluble inert organic matter, S_I, is omitted in this figure because it is not involved in any conversion processes, as defined in ASM1. Each solid grey line represents one sample. Median and 5th/95th percentiles at each time point were calculated based on the entire 1000 samples.

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

package.[67] Moreover, regardless of the initial conditions assigned to each simulation, the system consistently converged to the same steady state in QSDsan, which matched results from MATLAB/Simulink with a maximum relative error <1% for state variables across all unit operations in the system.

The transparent implementation of BSM1 provides an illustrative example of how QSDsan's basic structure can be leveraged and built upon for a wide range of process modeling applications. With built-in functions in QSDsan, users can quickly visualize the dynamics of state variables upon simulation. To implement new process models developed for novel technologies (*e.g.*, microalgal and cyanobacterial process models for photobioreactors[68]), apart from modifying an existing model in the corresponding spreadsheet or directly in Python, users can also import new stoichiometry, define functions for complex kinetics, and seamlessly incorporate them in system simulation with the *Process* module. New unit operations can be added as subclasses of *SanUnit* to the existing portfolio for dynamic simulations with essential attributes (*e.g.*, ODE algorithms) describing the mass balance. QSDsan's current structure also supports the implementation of more complex simulation settings, such as dynamic influent streams and active operational control with proportional–integral–derivative (PID) controllers.

3.3.2. Uncertainty and sensitivity analyses of BSM1. Following the same procedures as in the equilibrium simulation example, Monte Carlo simulation can also be used in the dynamic mode to propagate input uncertainties through the system to characterize the uncertainty in system model outputs. For this BSM1 implementation, the system was characterized considering uncertainties from decision variables (*e.g.*, aeration flowrate, designed reactor volume), technological parameters (*e.g.*, heterotrophic yield in ASM1), contextual parameters (*e.g.*, influent ammonium concentration, saturation dissolved oxygen [DO]), and modeling assumptions (*e.g.*, COD-to-mass ratio of particulate organic substrates). Visualization of the effluent dynamics over time and the converged steady state reveals that state variables closely related to the nitrification/denitrification processes (*i.e.*, nitrite/nitrate S_{NO}, and ammonia S_{NH}) were subject to the greatest uncertainty (Fig. 5). Additionally, by inspecting the distributions of key metrics (common effluent quality indicators, daily sludge production, sludge retention time [SRT]) against control or design objectives (*e.g.*, regulations on effluent quality, or sludge disposal costs), one can locate the performance gaps and make targeted decisions

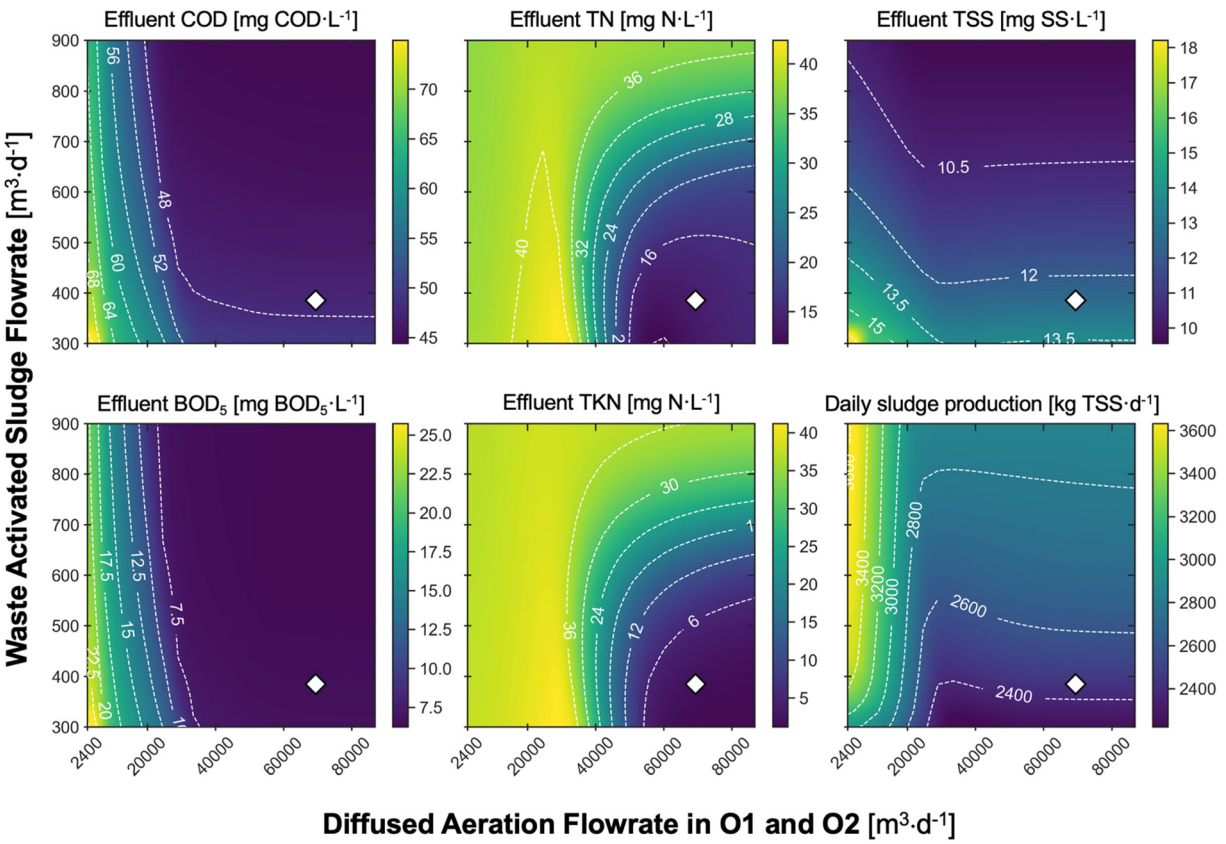


Fig. 6 Effluent quality metrics mapped across the decision space of aeration and sludge wastage. The diffused aeration air flowrate at field condition in the first two aerated CSTRs (*i.e.*, O1 and O2) was varied uniformly between 2400 $m^3\ d^{-1}$ and 86 765 $m^3\ d^{-1}$, corresponding to a K_La of 80–300 d^{-1}. The WAS flowrate was varied uniformly within 300–900 $m^3\ d^{-1}$, resulting in an SRT of 4.15–17.2 days. The white diamonds represent the baseline setting.

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

for improvement. For example, assuming the discharge limits are TN ≤ 18 mg-N per L and TKN ≤ 4.0 mg-N per L,[24] the BSM1 system was estimated to have 11.3% and 43.3% chances of violation, respectively, while the discharge limits of COD, BOD_5, and TSS (assumed to be 100 mg L^{-1}, 10 mg L^{-1}, and 30 mg L^{-1}, respectively) were estimated not binding for the BSM1 system (*i.e.*, the effluent was able to meet the limits for all simulations; Fig. S3 in the ESI†).

To identify the driving factors for compliance with the discharge limits, Monte Carlo filtering of the 28 uncertain parameters was performed with the simulation results from the uncertainty analysis described above (Table S4†). The filtering was performed by first dividing the samples into two groups (*i.e.*, above and below) based on the discharge limit of a particular metric (*e.g.*, effluent TN), followed by Kolmogorov–Smirnov tests to compare the distributions of each parameter between groups. With this analysis, the aeration rate to the first two aerobic CSTRs and the aerobic zone hydraulic retention time (HRT) were found to have the most significant differences in distribution between the two groups for TN (*i.e.*, they are the most impactful decision variables for effluent TN; Table S4†). For effluent TKN, in addition to the decision variables above, waste activated sludge (WAS) flowrate and RAS flowrate were also found to have significant impacts on the compliance with this discharge limit. These decision variables directly affect the DO levels and the SRT of the activated sludge system, which have been commonly recognized among the most important operational control and design parameters for biological nitrogen removal.[69,70] Overall, these analyses show QSDsan is equipped with robust algorithms for process simulation and uncertainty and sensitivity analyses, demonstrating its capacity for systematic identification of key decision variables and impactful parameters.

3.3.3. Mapping the decision space of activated sludge system operation. With the most impactful decision variables identified, systems can be simulated in QSDsan across the decision space of these variables to elucidate their implications on system performance (Fig. 6). Based on the results from Monte Carlo filtering, WAS flowrate and aeration flowrate to the first two aerobic CSTRs were chosen to illustrate the quantitative investigation of an optimal control strategy for the BSM1 system with QSDsan. Aerobic zone HRT, albeit important for effluent TN and TKN, was excluded from the analysis as it would be determined by the design of the reactors. Results show that effluent COD and BOD_5 were insensitive to the change of sludge wastage flowrate, whereas varying aeration rate had little impact on effluent TSS. Moderate reduction of both WAS and aeration flowrates from the baseline levels had the benefits of further lowering effluent TN and daily sludge production with only a marginal increase of effluent TSS and COD. This observation implies that the system can be operated at lower aeration energy demand and sludge disposal cost while meeting the discharge regulatory requirements. With the *TEA* and *LCA* classes in QSDsan, economic and environmental implications of such operational control changes can be further quantified and leveraged to understand potential trade-offs (*e.g.*, reducing cost *vs.* greater risk of violating limits).

4. Conclusions and future work enabled by QSDsan

In this work, we introduced QSDsan, which is an open-source platform that integrates system design, simulation, and sustainability characterization (TEA and LCA) under uncertainty for sanitation and resource recovery systems. With diverse and growing capacities, QSDsan can be used to prioritize research directions, facilitate technology deployment, and navigate decision-making across wide ranges of decision, technological, and contextual parameters. In particular, the OOP paradigm allows users to utilize existing process models, unit operations, and systems, tailor them to their needs (*e.g.*, modify process stoichiometry, adjust cost algorithms), and/or implement new ones. Uncertainties in every element of modeling assumptions, design, and operation (*e.g.*, material costs, technology performance) can be included in system simulation and sustainability analyses. This flexibility in system design and ability to perform rigorous uncertainty and sensitivity analyses (especially global sensitivity analysis) are critical to emerging technologies that are characterized by large uncertainties in their design and performance.

Further, the agility of QSDsan allows it to be easily connected to external packages for enhanced features. For example, through DMsan (decision-making for sanitation and resource recovery systems),[71] users can leverage QSDsan-generated simulation and sustainability analysis results in multi-criteria decision analysis. With Python being one of the most widely used programming languages (ranked #1 by IEEE Spectrum in 2021 (ref. 72)), QSDsan will benefit from the rapidly growing number of Python modules and libraries for future improvement (*e.g.*, incorporation of machine learning in mechanistic modeling;[73] implementation of digital twin in water/wastewater utilities[74]).

Moreover, we have laid the groundwork for a collaborative, community-led platform (*e.g.*, open-source, detailed documentation, executable tutorials with video demonstrations). This platform is poised to be adopted by researchers, practitioners (*e.g.*, the Container Based Sanitation Alliance[75]), and the public across the world to increase access to and sustainability of sanitation, which is particularly relevant to resource-limited communities where the largest need is expected for the coming decades.[76] Further, this growth of user base and contributors will enrich the collection of process models, unit operations, and systems in QSDsan, thereby enabling the use of QSDsan by users with less programming experience (*i.e.*, the users can directly use existing modules instead of developing from scratch). Additionally, QSDsan can be used in courses focusing on sustainable design to offer students hands-on opportunities without the cost barrier of a software license (*e.g.*, allowing

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

students to design their own systems and perform TEA and LCA, example workshop video in the YouTube channel[52]), and web-based platforms (*e.g.*, Binder,[77] Google Colab[78]) can be leveraged to develop interactive modules (some with a graphical user interface, see ref. 79) without the need to build a local Python environment. Altogether, QSDsan provides the field of sanitation and resource recovery with a valuable and timely tool for guiding technology RD&D, informing decision making, and fostering the stewardship of sustainability among the next generation, ultimately contributing to society's advancement toward a more sustainable future.

Conflicts of 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.

Acknowledgements

This publication is based on research funded in part by the Bill & Melinda Gates Foundation. The findings and conclusions contained within are those of the authors and do not necessarily reflect positions or policies of the Bill & Melinda Gates Foundation. We would like to thank Dr. Ulf Jeppsson (Lund University) for providing advice on the Benchmark Simulation Model and Yoel Cortés-Peña (University of Illinois Urbana-Champaign) for the suggestions and discussions on package development.

References

1 M. Roser and H. Ritchie, Technological Progress, *Our World in Data*.
2 U.S. Patent and Trademark Office, Issue Years and Patent Numbers, **https://www.uspto.gov/web/offices/ac/ido/oeip/taf/issuyear.htm**, (accessed 9 February, 2021).
3 D. J. C. Constable, What Do Patents Tell Us about the Implementation of Green and Sustainable Chemistry?, *ACS Sustainable Chem. Eng.*, 2020, **8**, 14657–14667.
4 A. Y. Hoekstra and T. O. Wiedmann, Humanity's unsustainable environmental footprint, *Science*, 2014, **344**, 1114–1117.
5 W. Steffen, K. Richardson, J. Rockström, S. E. Cornell, I. Fetzer, E. M. Bennett, R. Biggs, S. R. Carpenter, W. de Vries, C. A. de Wit, C. Folke, D. Gerten, J. Heinke, G. M. Mace, L. M. Persson, V. Ramanathan, B. Reyers and S. Sörlin, Planetary boundaries: Guiding human development on a changing planet, *Science*, 2015, **347**, 1259855.
6 D. W. O'Neill, A. L. Fanning, W. F. Lamb and J. K. Steinberger, A good life for all within planetary boundaries, *Nat. Sustain.*, 2018, **1**, 88–95.
7 J. B. Zimmerman, P. Westerhoff, J. Field and G. Lowry, Evolving Today to Best Serve Tomorrow, *Environ. Sci. Technol.*, 2020, **54**, 5923–5924.
8 UNEP/SETAC Life Cycle Initiative, *Towards a Life Cycle Sustainability Assessment*, 2011.
9 R. Mahmud, S. M. Moni, K. High and M. Carbajales-Dale, Integration of techno-economic analysis and life cycle assessment for sustainable process design – A review, *J. Cleaner Prod.*, 2021, **317**, 128247.
10 Y. Li, J. Trimmer, S. Hand, X. Zhang, K. Chambers, H. Lohman, R. Shi, D. Byrne, S. Cook and J. Guest, Quantitative Sustainable Design (QSD) for the Prioritization of Research, Development, and Deployment of Technologies: A Tutorial and Review, ChemRxiv, 2022, preprint, DOI: **10.26434/chemrxiv-2022-rjqbn**.
11 United Nations Department of Economic and Social Affairs, *Transforming our world: the 2030 Agenda for Sustainable Development*, United Nations, 2015.
12 C. Hyun, Z. Burt, Y. Crider, K. L. Nelson, C. S. S. Prasad, S. D. G. Rayasam, W. Tarpeh and I. Ray, Sanitation for Low-Income Regions: A Cross-Disciplinary Review, *Annu. Rev. Environ. Resour.*, 2019, **44**, 287–318.
13 H. A. C. Lohman, J. T. Trimmer, D. Katende, M. Mubasira, M. Nagirinya, F. Nsereko, N. Banadda, R. D. Cusick and J. S. Guest, Advancing Sustainable Sanitation and Agriculture through Investments in Human-Derived Nutrient Systems, *Environ. Sci. Technol.*, 2020, **54**, 9217–9227.
14 Hydromantis, *GPS-X*, **https://www.hydromantis.com/GPSX.html**, (accessed 12 April, 2021).
15 Dynamita, *dynamita*, **https://www.dynamita.com/the-sumo/**, (accessed 12 April, 2021).
16 EnviroSim, *BioWin*, **https://envirosim.com/products**, (accessed 12 April, 2021).
17 MIKE Powered by DHI, *WEST*, **https://www.mikepoweredbydhi.com/products/west**, (accessed 12 April, 2021).
18 AspenTech, *Aspen Plus*, **https://www.aspentech.com/en/products/engineering/aspen-plus**, (accessed 12 April, 2021).
19 Intelligen, Inc., *SuperPro Designer*, **https://www.intelligen.com/products/superpro-overview/**, (accessed 12 April, 2021).
20 Hydromantis, *CapdetWorks*, **https://www.hydromantis.com/CapdetWorks.html**, (accessed 17 February, 2022).
21 PRé Sustainability, *SimaPro*, **https://simapro.com/**, (accessed 12 April, 2021).
22 J. T. Trimmer, H. A. C. Lohman, D. M. Byrne, S. A. Houser, F. Jjuuko, D. Katende, N. Banadda, A. Zerai, D. C. Miller and J. S. Guest, Navigating Multidimensional Social–Ecological System Trade-Offs across Sanitation Alternatives in an Urban Informal Settlement, *Environ. Sci. Technol.*, 2020, **54**, 12641–12653.
23 J. T. Trimmer, *Bwaise-sanitation-alternatives*, **https://github.com/QSD-Group/Bwaise-sanitation-alternatives**, (accessed 2 September, 2021).
24 J. Alex, L. Benedetti, J. Copp, K. Gernaey, U. Jeppsson, I. Nopens, M. Pons, L. Rieger, C. Rosen and J.-P. Steyer, *Benchmark Simulation Model no. 1 (BSM1)*, 2008.
25 IWA Task Group on Benchmarking of Control Strategies for WWTP, Benchmark Simulation Models, **https://github.com/wwtmodels/Benchmark-Simulation-Models**, (accessed 28 November, 2021).
26 M. Henze, L. Grady Jr, W. Gujer, G. Marais and T. Matsuo, *Activated sludge model no.1*, International Association on Water Pollution Research and Control (IAWPRC), 1987.

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

27 I. Takács, G. G. Patry and D. Nolasco, A dynamic model of the clarification-thickening process, *Water Res.*, 1991, **25**, 1263–1271.
28 Y. Cortés-Peña, D. Kumar, V. Singh and J. S. Guest, BioSTEAM: A Fast and Flexible Platform for the Design, Simulation, and Techno-Economic Analysis of Biorefineries under Uncertainty, *ACS Sustainable Chem. Eng.*, 2020, **8**, 3302–3310.
29 BioSTEAM Development Group, *BioSTEAM: The Biorefinery Simulation and Techno-Economic Analysis Modules*, **https://github.com/BioSTEAMDevelopmentGroup/biosteam**, (accessed 17 April, 2020).
30 BioSTEAM Development Group, *Thermosteam: BioSTEAM's Premier Thermodynamic Engine*, **https://github.com/BioSTEAMDevelopmentGroup/thermosteam**, (accessed 17 April, 2020).
31 Y. Cortés-Peña, Thermosteam: BioSTEAM's Premier Thermodynamic Engine, *J. Open Source Softw*, 2020, **5**, 2814.
32 C. Bell, *Thermo*, **https://github.com/CalebBell/thermo**, (accessed 3 May, 2022).
33 C. Bell, *Chemicals*, **https://github.com/CalebBell/chemicals**, (accessed 3 May, 2022).
34 L. Rieger, S. Gillot, G. Langergraber, T. Ohtsuki, A. Shaw, I. Takcs and S. Winkler, *Guidelines for Using Activated Sludge Models*, IWA Publishing, 2012.
35 Hydromantis, *GPS-X Technical Reference - v8.0*, **https://www.hydromantis.com/help/GPS-X/docs/8.0/Technical/index.html**, (accessed 2 February, 2022).
36 J. Bare, TRACI 2.0: the tool for the reduction and assessment of chemical and other environmental impacts 2.0, *Clean Technol. Environ. Policy*, 2011, **13**, 687–696.
37 M. A. J. Huijbregts, Z. J. N. Steinmann, P. M. F. Elshout, G. Stam, F. Verones, M. Vieira, M. Zijp, A. Hollander and R. van Zelm, ReCiPe2016: A harmonised life cycle impact assessment method at midpoint and endpoint level, *Int. J. Life Cycle Assess.*, 2017, **22**, 138–147.
38 C. Mutel, *Brightway2*, **https://brightway.dev/**, (accessed 12 April, 2021).
39 Quantitative Sustainable Design Group, *BW2QSD*, **https://github.com/QSD-Group/BW2QSD**, (accessed 19 September, 2021).
40 Quantitative Sustainable Design Group, *EXPOsan*, **https://github.com/QSD-Group/EXPOsan**, (accessed 2 September, 2021).
41 N. Metropolis and S. Ulam, The Monte Carlo Method, *J. Am. Stat. Assoc.*, 1949, **44**, 335–341.
42 J. E. Gentle, *Random Number Generation and Monte Carlo Methods*, Springer Science & Business Media, 2003.
43 M. D. McKay, R. J. Beckman and W. J. Conover, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, *Technometrics*, 1979, **21**, 239–245.
44 J. Herman and W. Usher, SALib: An open-source Python library for Sensitivity Analysis, *J. Open Source Softw*, 2017, **2**, 97.
45 J. D. Hunter, Matplotlib: A 2D Graphics Environment, *Comput. Sci. Eng.*, 2007, **9**, 90–95.
46 M. L. Waskom, seaborn: statistical data visualization, *J. Open Source Softw*, 2021, **6**, 3021.
47 M. D. Morris, Factorial Sampling Plans for Preliminary Computational Experiments, *Technometrics*, 1991, **33**, 161–174.
48 A. Saltelli, Making best use of model evaluations to compute sensitivity indices, *Comput. Phys. Commun.*, 2002, **145**, 280–297.
49 *Benchmarking of Control Strategies for Wastewater Treatment Plants*, ed. K. V. Gernaey, U. Jeppsson, P. A. Vanrolleghem and J. B. Copp, IWA Publishing, 2014.
50 Quantitative Sustainable Design Group, *qsdsan*, **https://github.com/QSD-Group/QSDsan**, (accessed 4 September, 2021).
51 Quantitative Sustainable Design Group, *QSDsan documentation*, **https://qsdsan.readthedocs.io/en/latest/**.
52 Quantitative Sustainable Design Group, *QSDsan Video Resources*, **https://www.youtube.com/channel/UC8fyVeo9xf10KeuZ_4vC_GA**, (accessed 13 September, 2021).
53 M. Henze, W. Gujer, T. Mino and M. van Loosedrecht, Activated Sludge Models ASM1, ASM2, ASM2d and ASM3, Water Intelligence Online, 2006, 5, 9781780402369.
54 IWA Task Group for Mathematical Modelling of Anaerobic Digestion Processes, *Anaerobic Digestion Model No.1 (ADM1)*, 2005.
55 C. Rosen, D. Vrecko, K. V. Gernaey, M. N. Pons and U. Jeppsson, Implementing ADM1 for plant-wide benchmark simulations in Matlab/Simulink, *Water Sci. Technol.*, 2006, **54**, 11–19.
56 IWA Task Group on Benchmarking of Control Strategies for WWTP, *Anaerobic Digestion Models*, **https://github.com/wwtmodels/Anaerobic-Digestion-Models**, (accessed 20 July, 2021).
57 L. Rowles, V. Morgan, Y. Li, X. Zhang, S. Watabe, T. Stephen, H. Lohman, D. DeSouza, J. Hallowell, R. Cusick and J. Guest, Financial viability and environmental sustainability of fecal sludge treatment with Omni Processors, *ACS Environ. Au*, 2022, DOI: **10.1021/acsenvironau.2c00022**.
58 L. Trotochaud, R. M. Andrus, K. J. Tyson, G. H. Miller, C. M. Welling, P. E. Donaghy, J. D. Incardona, W. A. Evans, P. K. Smith, T. L. Oriard, I. D. Norris, B. R. Stoner, J. S. Guest and B. T. Hawkins, Laboratory Demonstration and Preliminary Techno-Economic Analysis of an Onsite Wastewater Treatment System, *Environ. Sci. Technol.*, 2020, **54**, 16147–16155.
59 Quantitative Sustainable Design Group, *Developed process models in QSDsan*, **https://qsdsan.readthedocs.io/en/latest/processes/_index.html**, (accessed 3 May, 2022).
60 Quantitative Sustainable Design Group, *Developed unit operations in QSDsan*, **https://qsdsan.readthedocs.io/en/latest/sanunits/_index.html**, (accessed 3 May, 2022).
61 Quantitative Sustainable Design Group, *Developed systems in QSDsan*, **https://qsdsan.readthedocs.io/en/latest/Developed_Systems.html**, (accessed 3 May, 2022).
62 Organization for Ethical Source, *Contributor Covenant: A Code of Conduct for Open Source and Other Digital Commons Communities*, **https://www.contributor-covenant.org/**, (accessed 4 September, 2021).

 This journal is © The Royal Society of Chemistry 2022

Open Access Article. Published on 22 August 2022. Downloaded on 10/6/2025 9:43:26 AM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

63 Quantitative Sustainable Design Group, *Contributing to QSDsan*, **https://qsdsan.readthedocs.io/en/latest/CONTRIBUTING.html**, (accessed 4 September, 2021).
64 P. Steiner, Example QSDsan contribution, https://github.com/QSD-Group/QSDsan, (accessed 3 May, 2022).
65 D. Garcia Sanchez, B. Lacarrière, M. Musy and B. Bourges, Application of sensitivity analysis in building energy simulations: Combining first- and second-order elementary effects methods, *Energy Build.*, 2014, **68**, 741–750.
66 M. Poirier-Pocovi and B. N. Bailey, Sensitivity analysis of four crop water stress indices to ambient environmental conditions and stomatal conductance, *Sci. Hortic.*, 2020, **259**, 108825.
67 P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa and P. van Mulbregt, SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nat. Methods*, 2020, **17**, 261–272.
68 B. D. Shoener, S. M. Schramm, F. Béline, O. Bernard, C. Martínez, B. G. Plósz, S. Snowling, J.-P. Steyer, B. Valverde-Pérez, D. Wágner and J. S. Guest, Microalgae and cyanobacteria modeling in water resource recovery facilities: A critical review, *Water Res.: X*, 2019, **2**, 100024.
69 Office of Research and Development, Water Supply and Water Resources Division, *Nutrient Control Design Manual*, U. S. Environmental Protection Agency, 2010.
70 G. Tchobanoglous, H. D. Stensel, R. Tsuchihashi and F. Burton, *Wastewater Engineering: Treatment and Resource Recovery*, McGraw-Hill Education, New York, 5th edn, 2013.
71 Quantitative Sustainable Design Group, DMsan: Leverages MCDA to compare technologies within a context and explore opportunity spaces for RD&D, **https://github.com/QSD-Group/DMsan**, (accessed 28 November, 2021).
72 IEEE Spectrum, Top Programming Languages 2021, **https://spectrum.ieee.org/top-programming-languages/**, (accessed 28 November, 2021).
73 W. Quaghebeur, E. Torfs, B. De Baets and I. Nopens, Hybrid differential equations: integrating mechanistic and data-driven techniques for modelling of water systems, *Water Res.*, 2022, 118166.
74 B. Valverde-Pérez, B. Johnson, C. Wärff, D. Lumley, E. Torfs, I. Nopens and L. Townley, *Digital Water: Operational digital twins in the urban water sector*, International Water Association, 2021.
75 Container Based Sanitation Alliance, *Container Based Sanitation Alliance*, **https://cbsa.global/**, (accessed 2 February, 2022).
76 World Health Organization, *Guidelines on sanitation and health*, World Health Organization, 2018.
77 The Binder Project, **https://mybinder.org/**, (accessed 20 July, 2022).
78 Google Colaboratory, **https://colab.research.google.com/**, (accessed 20 July, 2022).
79 Quantitative Sustainable Design Group, *QSDsan Workshop*, **https://github.com/QSD-Group/QSDsan-workshop**, (accessed 20 July, 2022).