Mine Closure 2026 – AB Fourie, G Boggs, J Heyes & M Tibbett (eds)

doi:10.36487/ACG_repo/2615_111

A probabilistic screening tool for long-term pit lake water balance estimation

Leonardo Navarro-Valdivia [a,*], Ryan Burgess [b], Paul Weber [a]

[a] *Mine Waste Management, New Zealand*

[b] *Hydrogeochem Group, New Zealand*

Abstract

Anticipating the filling behaviour of pit lakes is essential for effective mine closure planning and environmental management. Although sophisticated deterministic and probabilistic models are available (e.g. GoldSim), the industry also requires practical, high-level screening tools for the rapid assessment of long-term risks associated with pit lakes. To address this gap, a probabilistic water balance tool was developed to facilitate efficient preliminary pit lake assessments. This approach allows environmental and closure professionals to conduct robust preliminary assessments independently, thereby reducing the reliance on external resources.

The underlying computational engine calculates net volume changes by integrating direct precipitation, evaporation, catchment run-off, groundwater exchange and active pumping. The model employs a fourth-order Runge-Kutta (RK4) numerical method to handle stage-volume-area relationships. Optimised for rapid screening rather than granular simulation, the tool operates on a monthly timestep and assumes a repeating annual climate cycle, thereby optimising computational demands and simplifying user inputs.

To address the uncertainties inherent in long-term forecasting, a Monte Carlo simulation framework is integrated into the engine. By assigning statistical variances to key hydrological-hydraulic parameters, the model executes numerous iterations to produce probabilistic filling trajectories. This methodology captures statistical confidence intervals from the 5th to the 95th percentiles for both lake elevations and individual flux volumes.

While the tool is designed for future cloud-based deployment via a web interface to provide cross-platform accessibility, it is currently implemented as an open-source repository for local execution. This approach ensures immediate usability for environmental professionals while providing a foundation for automated, web-hosted reporting.

Keywords: *mine closure, pit lake, water balance, Monte Carlo simulation, Python*

1 Introduction

The transition from active mining to closure represents a critical phase in the life cycle of a mine. One of the most significant physical and chemical legacies of open pit operations is the formation of a pit lake. Predicting the filling rate, ultimate equilibrium elevation and long-term water balance of these terminal or flow-through sinks is a fundamental requirement for environmental impact assessments and closure planning. Accurate forecasting is essential to determine whether a pit lake will remain a terminal sink (where evaporation exceeds inflows, driving groundwater flow towards the pit) or become a source of potential contamination to the surrounding aquifer or surface water systems where pit water levels exceed surrounding groundwater levels. Historically, pit lake water balances have been constructed using deterministic spreadsheet models

* Corresponding author. Email address: leo.navarro@minewaste.com

utilising a simple explicit Euler integration method. While accessible, these models present several critical limitations such as:

- They rely on single-point estimates for highly variable inputs such as monthly rainfall, pan evaporation rates and regional groundwater heads. This deterministic approach fails to quantify the probability of extreme events, such as rapid filling due to consecutive wet years.
- While commercial software platforms (e.g. GoldSim) offer advanced probabilistic capabilities, their implementation typically requires specialised expertise and substantial time for model set-up and calibration.

For many projects, the high barrier to entry for commercial software platforms results in a reliance on inadequate deterministic tools during the critical early stages of closure planning. To overcome these constraints, a more accessible and simplified modelling approach is required. By applying probability distributions to key input parameters, environmental and closure professionals can evaluate a wide spectrum of potential future scenarios. This paper details the development and application of an interactive, Python-based Monte Carlo water balance model. Designed for rapid scenario testing, the tool enhances the predictive confidence of lake level evolution and provides a statistical foundation for subsequent long-term geochemical mass balance modelling and closure decision-making.

2 Methods

The mathematical basis of the predictive tool is a volumetric mass balance equation, evaluated continuously over the simulation period. The change in pit lake volume over time (dV/dt) is a function of the sum of all inflows minus the sum of all outflows.

2.1 The water balance equation

The conceptual model incorporates 6 primary flux pathways. The governing Equation 1 is defined as:

$$\frac{dV}{dt} = Q_{rain} + Q_{pit\_wall} + Q_{Catchment} + Q_{gw,in} - Q_{evap} - Q_{gw,out} - Q_{pump} \tag{1}$$

where:

Q_{rain} is direct precipitation onto the pit lake surface

Q_{pit_wall} is the run-off generated from the exposed, non-inundated pit walls

Q_{catch} is the run-off contribution from the external catchment that drains into the pit

$Q_{gw,in}$ and $Q_{gw,out}$ represent the bidirectional groundwater fluxes

Q_{evap} is the volumetric loss due to evaporation from the lake surface

Q_{pump} represents any anthropogenic water abstraction or addition (e.g. active dewatering, or filling).

2.2 Dynamic geometry and state variables

The model relies on a user-defined stage-volume-area curve. At any given time, *t*, the lake elevation (L_t) and the surface area (A_t) are dynamically calculated via linear interpolation of the current volume (V_t) against the geometric look-up tables, where:

$$L_t = f_{level}(V_t)$$

$$A_t = f_{area}(L_t)$$

2.3 Flux parameterisation

The specific flux terms are calculated as shown in Table 1.

Table 1 Surface inflows and outflows

Flux	Equation	Description
Precipitation	$Q_{rain} = A_t \cdot P_{monthly}$	Rainfall onto the pit lake's surface. $P_{monthly}$ is the monthly precipitation for the modelled month (January–December)
Pit wall run-off	$Q_{pit\_wall} = (A_{crest} - A_t) \cdot P_{monthly} \cdot C_{pit\_wall}$	Run-off generated on the pit walls A_{crest} is the surface area of exposed rock within the pit crest C_{pit_wall} represents the run-off coefficient of the pit walls
External catchment run-off	$Q_{catch} = A_{catch} \cdot P_{monthly} \cdot C_{catch}$	Run-off catchment draining into the pit (constant catchment represented by A_{catch}) C_{catch} represents the run-off coefficient of the catchment
Evaporation	$Q_{evap} = A_t \cdot E_{monthly}$	Evaporation from the lake (depending on monthly evaporation rates represented by $E_{monthly}$)
Groundwater	$Q_{gw,in,out} = K \cdot A_t(\frac{H_{gw} - L_t}{d_{gw}})$	Bidirectional groundwater flux H_{gw} – Regional groundwater elevation d_{gw} – Distance from the pitwall to the constant head (H_{gw}) K: Hydraulic conductivity of the rock The area of the lake (A_t) is assumed to be scaling proportionally to the pit wall saturated area as a simplification

2.4 Numerical approach: Runge-Kutta 4 integration

To resolve the non-linear feedback loop where changing volumes alter surface areas, which in turn alter flux rates, the model utilises an integration scheme. Runge-Kutta 4 (RK4) achieves fourth-order accuracy by evaluating the net flow rate at 4 trial points across the monthly timestep (Δt).

2.5 User interface

The user interface is constructed using Panel (Holoviz 2026), an open-source library that allows for the creation of interactive web applications directly from Python scripts. The application is categorised into distinct tabbed workflows: Parameters, Meteorology, Geometry, and Settings & Run.

- Parameters allows for real-time adjustment of scalar inputs (catchment areas, run-off coefficients, bulk hydraulic conductivity and regional head).
- Meteorology and Geometry use interactive tabulator widgets, enabling users to paste directly from external data sources into the application. Meteorology inputs are as monthly averages.
- Settings & Run allows the user to set up the variability of each parameter as a function of percentual variability (e.g. ±20%), set the number of runs for the Monte Carlo simulations, run the model and generate an HTML report.

Screenshots of the web application user interface are shown in Figure 1.

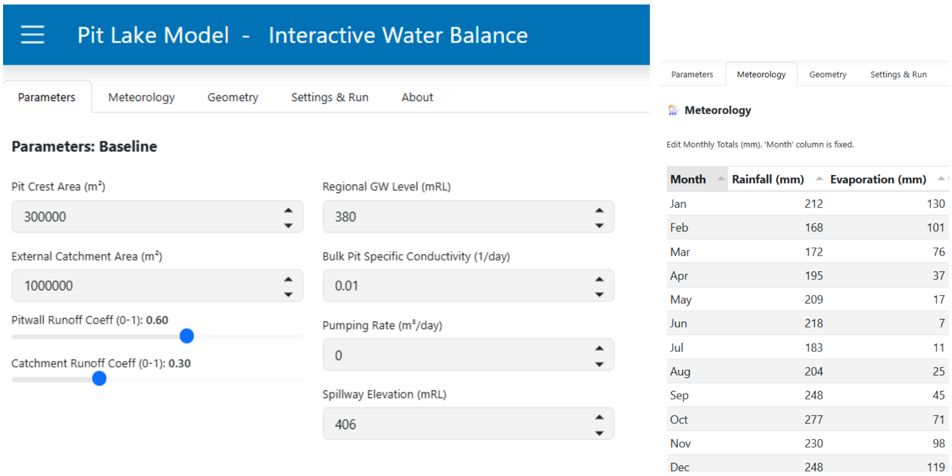

Figure 1 User interface screenshots

2.6 Stochastic modelling

The application utilises a Monte Carlo framework to execute N parallel simulations (where N is a modifiable number), perturbing input parameters based on user-defined standard deviations.

The parameter perturbation is included at the initialisation of each run, where the model applies a normal distribution scaling factor to the baseline parameters, for a given parameter X_{base} with a user-defined variation percentage σ.

This stochastic variation is applied to:

- Meteorological inputs. Monthly rainfall and evaporation rates are perturbed to simulate prolonged wet or dry climatic cycles
- Hydrological coefficients. Pit wall and external catchment run-off coefficients (C_{wall}, C_{catch}) are varied to account for uncertainties
- Hydrogeological conditions. Both the bulk hydraulic conductivity (K_{bulk}) and the regional groundwater head (H_{gw}) are treated as stochastic variables to reflect the inherent heterogeneity of fractured rock aquifers and the uncertainty in regional flow dynamics.

2.7 Data processing and aggregation

The model utilises NumPy (Harris et al. 2020) SciPy (Virtanen et al. 2020) and pandas (McKinney 2010) functions to handle all calculations.

Upon completion of the N runs, the resulting time-series data for lake volume, elevation and all flux pathways are compiled. The model outputs statistical quantiles across the simulation matrix, isolating the 5th (P05), 25th (P25), 50th (P50/median), 75th (P75) and 95th (P95) percentiles for every monthly timestep.

3 Results interpretation and visualisation

The output of the stochastic engine is a suite of interactive plots generated using the Plotly (2015) library, which allows users to pan, zoom and explore the data directly within the browser.

3.1 Probabilistic level forecasting

The primary output is the lake level forecast (see Figure 2). The visualisation renders the P50 median filling trajectory as a solid line surrounded by shaded confidence intervals. The inner dark band represents the P25–P75 interquartile range, while the outer light band represents the P05–P95 extremes. The inclusion of the target regional groundwater level on this plot visually confirms when the lake transitions from a groundwater sink to a potential groundwater source.

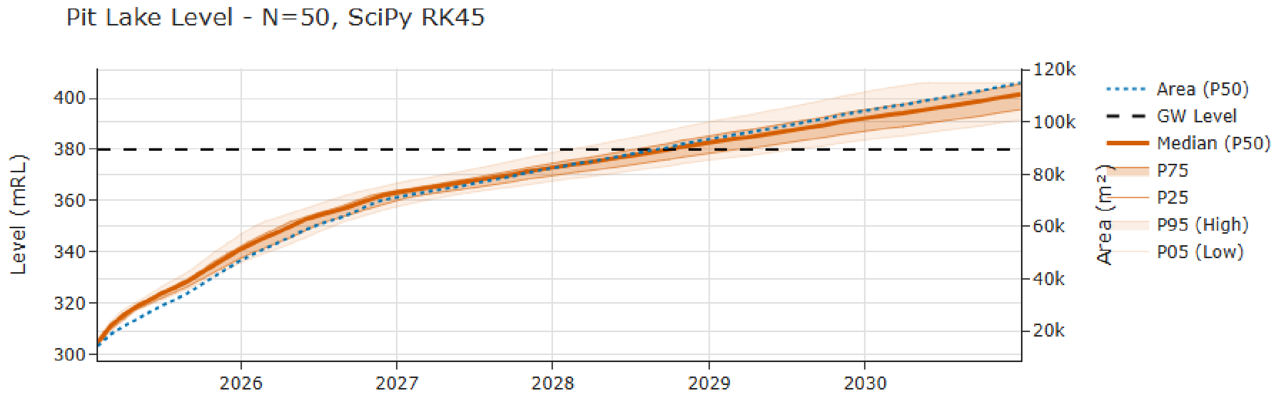

Figure 2 Pit lake level model results and percentiles

3.2 Flux disaggregation

To diagnose the dominant drivers of the water balance, the model generates stacked, multirow subplots for all inflow and outflow components (see Figure 3). By analysing the P05–P95 bands on these flux plots, users can identify the highest sources of volumetric uncertainty. For example, a wide probability band on pit wall run-off compared to a narrow band on direct rainfall immediately informs the user that refining the physical pit wall catchment area will yield the highest improvement in model confidence.

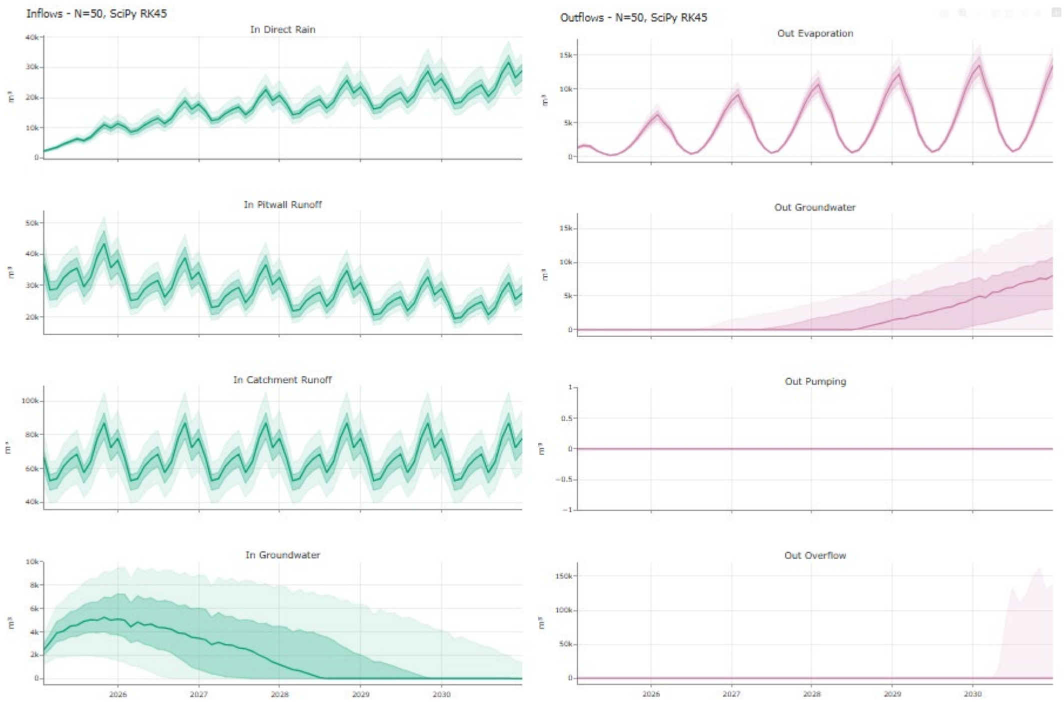

Figure 3 Inflow/outflow plots screenshot

4 Automated reporting

The application translates the raw outputs into processed tables and plots via the *report_generator.py* module as shown in Figure 4.

Upon successful completion of a simulation, the user can export a standalone HTML document that contains the following:

- Model inputs. The report provides the selected inputs
- Data outputs. The interactive Plotly figures are embedded, retaining their hover data and zoom capabilities within the final report
- Tabular summaries. The script automatically generates formatted probability summary tables and annual water balances (broken down by percentile).

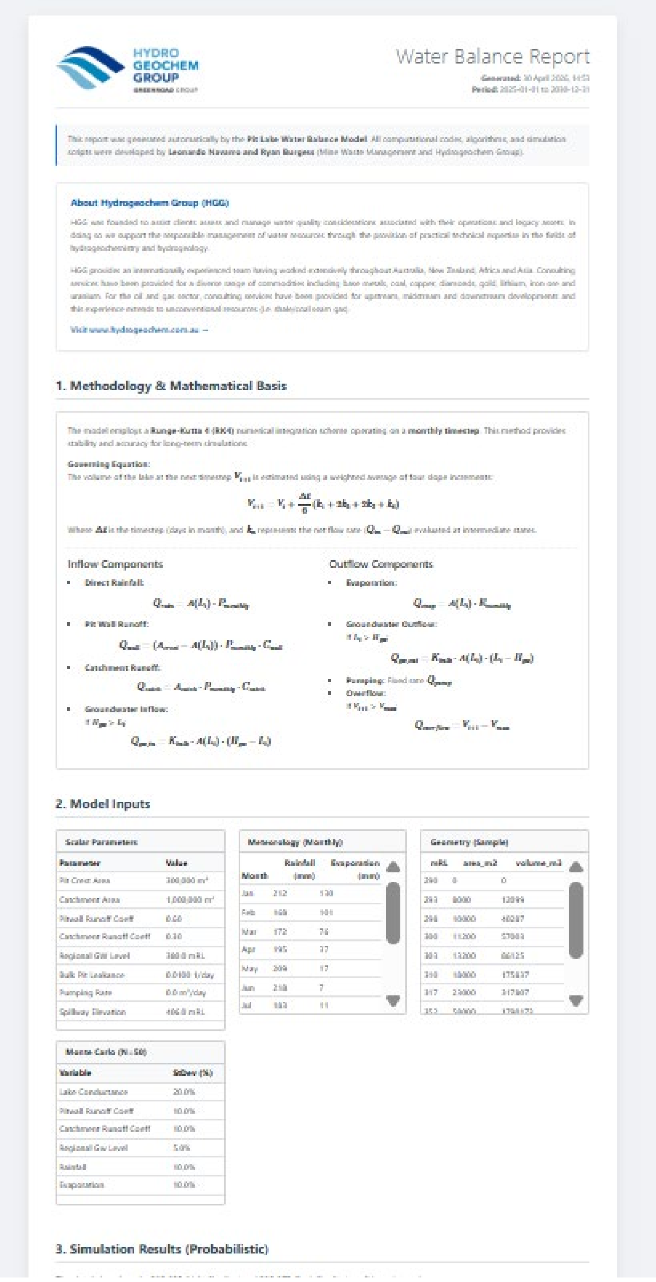

Figure 4 Generated HTML report screenshot

5 Deployment and implementation

5.1 Cloud-based access and data security

The model is architected to eventually function as a reactive web application hosted on cloud infrastructure. This planned deployment will allow users to access the tool via a standard web browser without requiring local Python installations or internal IT approvals. To ensure data privacy, the application is designed not to store any site-specific geometry and meteorological data uploaded by the user, as they are held only in temporary RAM during the session and are never written to permanent storage on any server.

5.2 GitHub repository

The source code for the model, and the report generation modules, are hosted on GitHub to support transparency and collaborative development (https://github.com/leonavarroMWM/pitlake-model). The repository structure is organised as follows:

- *pitlake_app.py* – the main dashboard user interface and application state management
- *pitlake_model.py* – *t*he numerical engine and stochastic sampling logic
- *report_generator.py* – the module responsible for compiling dynamic HTML summaries.

5.3 Local Windows implementation

This tool can be run locally on Windows. A standalone bootstrap script (run.cmd) is provided within the repository. This script automates the creation of an isolated, portable Python environment, pre-configuring all required scientific libraries. This 'zero install' approach ensures that environmental professionals can execute the model on corporate hardware without requiring administrative privileges or manual environment configuration Alternatively, users can by-pass the 'zero-install' approach by manually installing the Python package's requirements directly. A 'Readme.md' is provided in the GitHub repository for further installation instructions.

6 Limitations and next steps

6.1 Hydrogeological refinement

Currently, the model utilises a simplified linear scaling of groundwater flux against the lake's wetted surface area. For this screening-level tool, the wetted horizontal surface area (A_t) is used as a proxy for the submerged pit wall area. While this is a simplification for rapid water balance calculations, future iterations could implement a more precise geometric calculation of the 3D lateral wall area to improve flux accuracy. Furthermore, in highly heterogeneous environments, groundwater inflows may be heavily structurally controlled; for instance, by discrete fracture zones intersecting the pit at specific elevations. To address this, future versions of the model could allow users to define variable hydraulic conductivity parameters mapped to specific elevation intervals within the pit shell, rather than relying on a uniform bulk pit leakance.

6.2 Geochemical coupling

The volumetric water balance is only the first half of the prediction. The ultimate goal of this framework is to couple the probabilistic volumes with a geochemical mass balance engine for water quality predictions. By treating the P05–P95 volumetric fluxes as conservative transport mechanisms, future expansions will apply concentration profiles to each inflow term. This will enable the long-term simulation of solute evolution such as sulphate or any other solute.

6.3 Open-source collaboration

For transparency and continuous improvement, the full source code for this project is hosted on GitHub. We encourage peer review, contributions and collaboration from other geoscientists and developers. By maintaining an open-source framework, the tool can evolve to incorporate more complex modules, such as climate change sensitivity adjustments or high-resolution spatial groundwater modelling, ensuring it remains a relevant and robust resource for mine closure professionals globally.

6.4 Model validation and history matching

While model validation/calibration is a standard regulatory requirement, the screening-level nature of this probabilistic tool necessitates a tailored approach. Traditional automated calibration has been deliberately omitted from the current framework. In simplified modelling environments, automated calibration often suffers from non-uniqueness (equifinality), where multiple parameter combinations yield the same result, potentially generating mathematically valid but conceptually incorrect solutions.

To support validation without compromising the tool's accessibility, future iterations will integrate pit lake level data (input from the user) visualisation. Users will be able to overlay historical pit lake elevations directly onto the probabilistic output plots. This approach facilitates manual history matching, empowering practitioners to use their conceptual site knowledge to iteratively adjust input distributions and validate that the model's predictive envelope aligns with real-world observations.

Use of artificial intelligence disclosure statement

In the preparation of this work the authors used Gemini 3.1 (PRO) to refine the grammar of the manuscript and to assist in the development and optimisation of the Python-based numerical engine and the automated report generation module. The authors have reviewed the content and edited it where required, and take full responsibility for said content.

References

Harris, CR, Millman, KJ, van der Walt, SJ, Gommers, R, Virtanen, P, Cournapeau, D, … Oliphant, TE 2020, 'Array programming with NumPy', *Nature*, 585(7825), pp. 357–362. https://doi.org/10.1038/s41586-020-2649-2

HoloViz 2026, *Panel: A High-Level App and Dashboarding Framework for Python*, version 1.6.0, computer software, https://panel.holoviz.org

McKinney, W 2010, 'Data structures for statistical computing in Python', in S van der Walt & J Millman (eds), *Proceedings of the 9th Python in Science Conference*, pp. 56–61, https://doi.org/10.25080/Majora-92bf1922-00a

Plotly 2015, *Collaborative Data Science,* https://plot.ly

Virtanen, P, Gommers, R, Oliphant, TE, & SciPy 1.0 Contributors 2020, 'SciPy 1.0: Fundamental algorithms for scientific computing in Python', *Nature Methods*, vol. 17, no. 3, pp. 261–272, https://doi.org/10.1038/s41592-019-0686-2