MCMC method overview and rationale

What MCMC is?

Markov chain Monte Carlo (MCMC) is a general strategy to generate dependent random draws from a target distribution that may be high-dimensional, constrained, and non-Gaussian. A Markov chain is constructed so that its stationary distribution equals the target; after burn-in, successive states provide approximate samples. In practice, we specify the target via component-wise truncated normals (oxide means ± SD with non-negativity) plus soft sum penalties (mass-balance proximity), and then use a gradient-based sampler (NUTS/Hamiltonian Monte Carlo) to explore that distribution efficiently. This yields joint draws that honor bounds and constraints without requiring closed-form multivariate densities.

MCMC is characterised by the introduction of a Markov Chain, which ensures that the generation of each new sample depends only on the previous one, without considering other factors. An accessible analogy for the MCMC method is exploring a dark room, where at each step only a small area around the current position is visible, and a random direction is chosen without direct knowledge of the rest of the room. As the number of movements increases, the understanding of the room's layout becomes progressively clearer. Ultimately, the mapped layout represents the unknown "posterior distribution."

The Hamiltonian Monte Carlo (HMC) method is a specific case of the Markov Chain Monte Carlo (MCMC) method. While HMC method is a specific case of MCMC simulation, there are key differences between the two methods. Traditional MCMC simulations do not require knowledge of the prior distribution for each feature. Traditional MCMC methods start from a random initial point and repeatedly perform sampling, acceptance, and rejection within a vast data space. This often causes random samples in high-dimensional spaces to fall into regions where the target posterior distribution has low probability density, leading to low acceptance rates and substantial computational and time costs.

In contrast, the HMC method leverages the principles of Hamiltonian dynamics by simulating the motion of particles in potential and kinetic energy fields to efficiently explore the target probability distribution, making it particularly suitable for high-dimensional spaces. By introducing gradient parameters, HMC enables random sampling to make larger jumps in high-dimensional space and adjusts the sampling direction toward regions with higher probability density in the target posterior distribution. This improves sampling efficiency, reduces the correlation between consecutive samples, and avoids excessive sampling in local regions. Returning to the earlier analogy, this is akin to taking larger steps in a dark room , each step favoring unexplored areas within the visible range. By replacing complete randomness in this directional way, the true layout of the dark room can be understood more quickly.

Why not draw directly from a correlated distribution using a correlation matrix?

Why don't we just sample from a correlated multivariate normal distribution using a correlation matrix? Because the real data have constraints and features that a simple correlated normal cannot capture. Oxides are like parts of a whole—they sum to a fixed value, and must be non-negative. This creates skewed and bounded patterns, not a symmetric bell-shaped distribution. Also, different oxides have different levels of measurement uncertainty; some even have zeros, which require special handling. Trying to force all this into one correlation matrix is difficult—it can be unstable when built from mixed datasets and may not reflect real geological controls. Instead, we use MCMC because it naturally respects positivity, mass balance, and variable uncertainties. It also lets us add geologic rules later in a transparent way, and allows us to check that the results are reliable. In short, MCMC fits the problem better because it directly handles the constrained, non-Gaussian.

How MCMC is implemented?

1. Purpose and scope (fixed T–P)

What this method does: We use a simple MCMC procedure to create realistic "variants" of each experimental composition. For every sample, we make small random changes to the reported melt and plagioclase oxide contents, within their stated standard deviations (SD). All values stay non-negative, and the total oxide sum stays close to the given target sum. These variants are then used to show uncertainty, to test model robustness, or to lightly augment the dataset.

Fixed T–P: In our hydrous melt experiments, temperature (T) and pressure (P) are exact. We do not sample or change T and P. They are copied to every variant as constants.

Inputs (per sample)

1. Melt oxides (wt.%): SiO_2, TiO_2, Al_2O_3, FeOt, CaO, MgO, Na_2O, K_2O, each with a mean and an SD.
2. Plagioclase oxides (wt.%): the same eight oxides, each with a mean and an SD.
3. Target sums (wt.%): SUM_melt and SUM_plag (sum of the eight oxides used).
4. T (°C) and P (MPa): exact experimental values.
5. Optional: H_2O (not sampled).

Outputs (per sample)

A small set of joint draws (default: 10) of melt and plagioclase oxide compositions that:

1. are ≥ 0;
2. stay within the reported uncertainty bands;
3. keep the oxide sums close to the provided targets.
4. The same T, P, and (if present) H_2O are attached to each draw.

2. Inputs and pre-processing

This section lists the required columns, units, and simple cleaning rules before running the sampler. The goal is to pass clean, numeric inputs with sensible uncertainty to the MCMC. T and P are fixed constants and are not sampled.

2.1 Required columns (names must match)

(1) Identifiers: SAMPLE NAME

(2) Melt oxides (wt.% means and SDs)

Means: SIO2_melt, TIO2_melt, AL2O3_melt, FEOT_melt, CAO_melt, MGO_melt, NA2O_melt, K2O_melt

SDs: add to each of the above (e.g., _SD; SIO2_melt_SD; ⋯ ; K2O_melt_SD)

(3) Plagioclase oxides (wt.% means and SDs)

Means: SiO2_plag, TiO2_plag, Al2O3_plag, FeOt_plag, MgO_plag, CaO_plag, Na2O_plag, K2O_plag

SDs: add to each (e.g., _SD; SiO2_plag_SD; ⋯ ; K2O_plag_SD)

(4) Target sums (wt.%)

SUM_melt, SUM_plag

(These are the sums of the eight oxides listed above, not "total = 100 wt.%" unless your table is defined that way.)

(5) Fixed thermodynamic conditions

T (°C), *P*(MPa)

(exact experimental values; carried through unchanged.)

(6) Optional

H_2O (wt.%)

2.2 Units and basic ranges

Oxides in wt.%; must be $\geqslant$ 0.

T in °C and *P* in MPa (convert beforehand if needed).

SUM_melt and SUM_plag should be close to the sum of the eight oxides you provide (often near, but not necessarily exactly, 100 wt.% if only major oxides are included).

2.3 Make everything numeric

Convert all numeric columns to numbers (e.g., pd.to_numeric(..., errors="coerce")).

Ensure no negative values in means or SDs.

2.4 Simple rules for zero SDs and zeros (oxides only)

These rules stabilize sampling and reflect minimal, practical uncertainty.

Non-zero mean but SD = 0: set SD = 2% of the mean for that oxide.

Mean = 0 and SD = 0: temporarily set both mean and SD to 1^{e-6} as a placeholder to avoid numerical issues; after sampling, set the corresponding draws back to 0.

Do not apply these fixes to *T* or *P* (they are constants).

2.5 Check sums before sampling

Confirm that is consistent with the eight melt oxides you use, and with the eight plagioclase oxides.

The model uses a soft mass-balance penalty to keep sampled sums close to these targets, so unrealistic targets will produce unrealistic draws. Adjust or document if your represent only a subset of oxides.

2.6 Quick quality checklist

All required columns present with exact names.

Oxide means ⩾ 0; SDs > 0 after the simple fixes above.

SUM_melt and SUM_plag make sense relative to the listed oxides.

T and *P* present for every sample and in the correct units; they are not modified.

3. Probabilistic model and constraints

This section explains what we sample, how we set the distributions, and how we keep sums reasonable. Temperature and pressure are fixed constants and are not part of the sampling.

3.1 What is sampled

For each sample, we sample the eight melt oxides and the eight plagioclase oxides (all in wt.%). These 16 quantities form one joint draw. Temperature, pressure, and (if present) H_2O are simply carried along with each draw without change.

3.2 Priors for oxides: truncated normal

Each oxide is modeled with a simple normal distribution centered on its reported mean and spread by its reported standard deviation (SD), but with physical bounds:

Lower bound: 0 (no negative wt.%).

Upper bound: mean plus 3×SD (to avoid unrealistic tails).

We treat oxides as independent at the prior stage. This is a pragmatic choice that keeps the model simple and stable while still reflecting the reported uncertainty. If an oxide had zero SD in the table, we apply the small fixes described in Section 2 before sampling so the prior remains well behaved.

3.3 Soft mass-balance constraint (sums close to targets)

Real compositions have meaningful partial sums. To respect this, we keep the melt-oxide sum close to "SUM_melt" and the plagioclase-oxide sum close to "SUM_plag." We do this with a soft Gaussian penalty: the farther a sampled sum is from its target, the more weight it loses. Two tuning numbers control how tight this is (one for melt, one for plagioclase). By default they are set to allow about ±1 wt.% deviations. This acts like a gentle mass-balance pull, not a hard equality, so the sampler can still explore.

3.4 Boundaries and zero handling

Because the priors are truncated at zero, negative values cannot appear. For oxides that are truly absent, we use a tiny placeholder during sampling (as described in Section 2) only to avoid numerical problems; after sampling, those entries are set back to exactly zero in the saved draws.

3.5 What this achieves

These choices produce joint draws that: stay non-negative; respect the reported means and SDs; and keep sums close to the declared targets. The result is a set of realistic composition variants around each experiment, ready to propagate uncertainty into later steps, while temperature and pressure remain fixed as measured.

4. Sampling settings and diagnostics

This section explains how we run the sampler and how we check that the draws are reliable. Temperature and pressure remain fixed and are not sampled.

4.1 Default settings

Sampler: NUTS (Hamiltonian Monte Carlo with automatic tuning).

Warm-up (tuning): 1000 iterations per chain.

Kept draws: 1000 iterations per chain.

Number of chains: 4.

Random seed: 42 (for full reproducibility).

Initialization: each oxide starts at its reported mean.

Soft sum penalties: both set to allow about ±1 wt.% around SUM_melt and SUM_plag.

These settings are a good balance between stability and runtime for typical experimental datasets. They give enough draws to estimate uncertainty bands and to select a small set of variants for export.

4.2 What to look at after sampling

Run these basic checks before using the draws:

Convergence indicators: Effective sample sizes should be comfortably large. There should be few or no divergences.

Trace behavior: Each chain should move freely and mix well. Chains should overlap and settle to a stable band after warm-up.

Boundary pressure: Few draws should sit right at the lower bound 0 or at mean + 3×SD. If many values pile up at a bound, review the mean–SD pair or the truncation range.

Sum residuals: Check that the melt and plagioclase sums are close to their targets. The residuals (sampled sum minus target sum) should center near zero with a spread consistent with the chosen penalty width.

Sanity ranges: Oxide values should stay within reasonable geologic ranges for the studied materials. If not, revisit inputs and bounds.

4.3 If diagnostics are not satisfactory

Many divergences or slow mixing: Increase warm-up (for example, to 2000), or raise the target acceptance (for example, to 0.9). If needed, allow deeper trees. These changes make the sampler more cautious.

Sums drift too far from targets: Tighten the soft sum penalties slightly (smaller penalty width). Do this gradually to avoid over-constraining the chains.

Values press against zero: Recheck the reported SDs for very small numbers. For true zeros, keep the tiny placeholder during sampling, but confirm that zeros are restored on export.

Unrealistic tails near mean + 3×SD: Consider narrowing the upper bound for specific oxides if domain knowledge supports it, or review the SDs that set the spread.

Highly uneven SDs across oxides: Very large SDs on a few oxides can dominate the exploration. Confirm measurement entries and consider capping extreme SDs if they reflect data entry issues.

4.4 Reproducibility

Keep the random seed fixed for both sampling and the final selection of exported variants. Document software versions for PyMC and related packages. Save the settings used (warm-up, draws, chains, penalty widths) alongside the exported file. With these settings and checks, the chains provide stable, transparent draws that reflect the reported uncertainties in melt and plagioclase compositions, while T and P remain fixed as measured.

5. Post-processing and export

This section explains how we turn raw MCMC output into a small, usable set of variants for each sample, and how we save them for later work. Temperature and pressure stay fixed and are copied to every row.

5.1 Combine and select draws

After warm-up, combine all chains for each sample. Randomly select a small number of joint draws per sample (default: 10). This keeps files compact while still showing the reported uncertainty. Keep the random seed fixed so selection is reproducible.

5.2 Restore true zeros and attach constants

For oxides that were truly absent in the table (mean = 0 and SD = 0), set the sampled values back to exactly 0 in the exported file. The tiny placeholder used during sampling is only for numerical stability.

Copy the exact experimental T (°C) and P (MPa) to every row without change. If present, copy H_2O to every row without change.

5.3 Basic screening before saving

Check that all oxides are non-negative and within a reasonable range for the material studied. Check that the melt and plagioclase sums are close to their targets. A small symmetric spread around the target is expected. If many values sit exactly at a bound (0 or mean + 3×SD), revisit inputs for that sample (means, SDs, or bounds).

5.4 File format and columns

Export a single table with one row per selected draw. Recommended columns:

"SAMPLE NAME"

Melt oxides (wt.%): SiO_2, TiO_2, Al_2O_3, FeOt, CaO, MgO, Na_2O, K_2O

Plagioclase oxides (wt.%): same eight oxides

"SUM_melt", "SUM_plag" (optional: recomputed sums to verify mass balance)

"T" (°C), "P" (MPa) — copied as constants

Optional: "H_2O"

Optional: a simple draw index per sample (for traceability)

Use a clear filename with a date stamp. Excel or CSV are both fine; keep units in the header and be consistent across files.

5.5 Simple uncertainty summaries (optional)

If needed for figures or quick checks, compute per-sample summaries from the combined draws before selection, such as the median and the 16th–84th percentiles (or 2.5th–97.5th). These give clean uncertainty bands for each oxide while keeping T–P fixed.

5.6 Reproducibility record

Alongside the exported table, record: sampler settings (warm-up, kept draws, number of chains), penalty widths used for the sums, random seed for sampling and for selection, software versions.

6. Physical checks and quality control

This section lists quick, practical checks to confirm that the exported draws are geologically reasonable. Temperature and pressure stay fixed.

6.1 Mass-balance proximity

Compute the residuals "sum of melt oxides-SUM_melt" and "sum of plagioclase oxides-SUM_plag." The residuals should center near 0. With the default penalty width, most values should fall within about ±1–2 wt.%. If spreads are wide or biased, tighten the penalty slightly or review the target sums.

6.2 Non-negativity and bounds

All oxides must be ≥ 0. Check that very few draws sit exactly at 0 unless the oxide is truly absent. Check how often values touch the upper bound "mean + 3×SD." Heavy piling at the bound suggests that the SD is too small or the bound too tight. Review those inputs.

6.3 Per-oxide sanity ranges

Compare each oxide's draws with reasonable ranges for your material and experiment type (major oxides, wt.%). Use simple plots (histograms or boxplots) to see if any oxide is drifting to implausible levels. If needed, narrow the bound for that oxide or verify the reported SD.

6.4 Cross-phase consistency

Check that melt and plagioclase compositions are jointly sensible (for example, plagioclase alkalis and

CaO look reasonable for the studied assemblage). If any phase shows compositions that contradict basic petrographic expectations, review the inputs for that sample.

6.5 Zero handling

For oxides listed as truly 0 in the table, confirm they are restored to exactly 0 in the export. If many draws press to 0 for a nonzero mean, recheck the mean–SD pair or consider a slightly wider SD.

6.6 Outliers and flags

Flag any draw where a single oxide looks inconsistent with the rest of the sample (for example, extreme residual from the target sum or repeated bound hits). Consider removing those draws or revising the inputs for that sample, then rerun.

6.7 When to adjust settings

If sums wander too far from targets: tighten the sum penalty modestly. If mixing is slow or diagnostics look poor: increase warm-up or target a higher acceptance (for example, 0.9). If certain oxides pile at bounds: review their SDs and bounds, or apply a narrower, geologically informed upper bound.

7. Computing resources and reproducibility

This section gives simple, practical guidance so others can rerun the sampler and get the same results.

7.1 Software environment

Use a clean Python setup with fixed versions. The following are sufficient:

Python 3.10–3.12

scikit-learn

PyMC 5.18.2

ArviZ 0.20.0

PyTensor (installed with PyMC, version=2.26.3)

NumPy 1.24+

pandas 2.0+

7.2 Randomness and seeding

Set and record one seed for sampling and the same (or another recorded) seed for selecting the final draws.

Sampling seed: for example, 42.

Per-chain seeds: optional; if used, record all of them.

Draw-selection seed: fix it so the exported 10 draws per sample are the same every run.

7.3 Performance and runtime

Cost grows with: number of samples, number of parameters (16 oxides per sample), chains, and warm-up and kept draws.

Four chains with 1000 warm-up and 1000 kept draws work well for typical experimental tables. A multi-core CPU helps; a GPU is not required. If runtime is long, reduce kept draws or process samples in batches.

7.4 Common issues and quick fixes

Many divergences or slow mixing: Increase warm-up, aim for a higher target acceptance (for example, 0.9), or allow deeper trees. Recheck inputs for very small or zero SDs.

Values pressed to bounds: Review the mean–SD pair, widen SD slightly if justified, or narrow unrealistic upper bounds for specific oxides.

Sums far from targets: Tighten the sum penalty a little (smaller penalty width). Do this gradually.

Numerical errors or NaNs: Ensure all required columns are numeric, non-negative, and have SDs after the simple fixes in Section 2. Remove or correct rows with broken entries.

Memory or file size concerns: Export only a small number of draws per sample (default 10). Stream processing if the table is very large.

7.5 Clarification for readers

Temperature and pressure are fixed experimental values in this work. They are not sampled. Only oxide compositions are perturbed within their reported uncertainty, with a soft mass-balance pull toward the given sums.