

TMCMC×TSM-ROM

Linearization management + analytical derivatives/JIT: program flow and key takeaways

Keisuke Nishioka
IKM_Hiwi / tmcme

January 17, 2026

Goals

- Provide a map from entry point `case2_tmcmc_linearization.py` to execution-critical modules
- Explain what dominates runtime and what dominates inference accuracy
- Highlight audit artifacts for reproducibility (`config.json`, `likelihood_meta`, diagnostics CSVs)

Key takeaways (paper / talk)

- **Method:** TMCMC (ESS-tempering) + TSM-ROM with linearization-point updates
- **Why it works:** robust exploration early (no linearization), fast/accurate later near MAP (linearization ON + θ_0 updates)
- **What to audit:** $\beta = 1$ reached, explicit likelihood definition, and per-stage diagnostics tables

End-to-end pipeline

One-liner

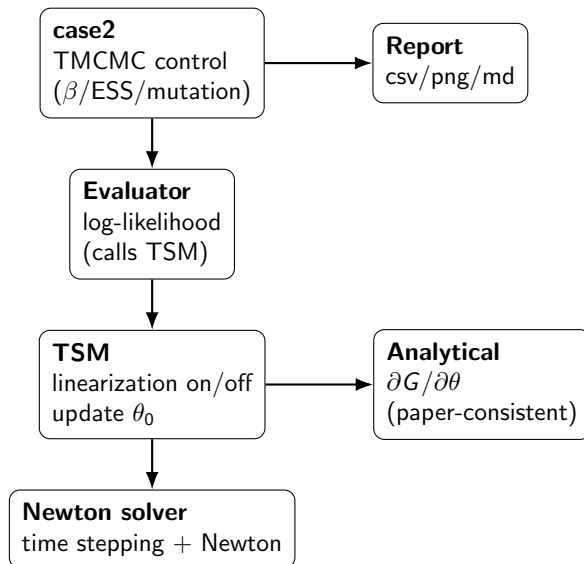
`run_pipeline.py` creates a run directory, executes the experiment runner, then builds `REPORT.md`.

- Progress logs: `subprocess.log` / `pipeline.log`
- Audit artifacts: `config.json`, `likelihood_meta_*.json`

Key modules

- Entry / control: `case2_tmcmc_linearization.py`
- Config: `config.py`
- TSM (linearization + analytical/JIT):
`demo_analytical_tsm_with_linearization_jit.py`
- Physical solver: `improved1207_paper_jit.py`
- Paper-mode derivatives: `paper_analytical_derivatives.py`
- Diagnostics/report: `mcmc_diagnostics.py`, `make_report.py`

Module map



- Tempering from prior ($\beta = 0$) to posterior ($\beta = 1$)
- Choose $\Delta\beta$ per stage based on target ESS (with min/max caps)
- Weight update \rightarrow ESS \rightarrow resample \rightarrow mutation (MCMC) for mixing

Critical check

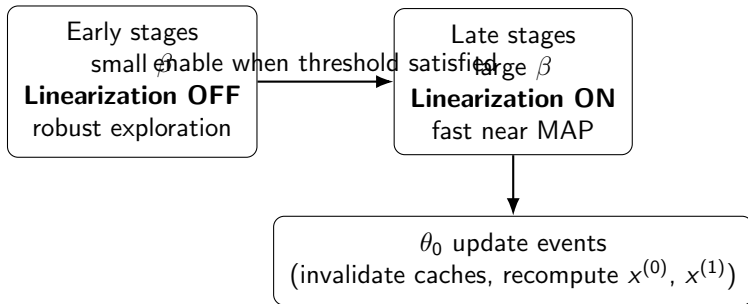
Confirm β **reaches 1.0** (log message like “ β reached 1.0”).

Local linearization

$$x(\theta) \approx x(\theta_0) + \left. \frac{\partial x}{\partial \theta} \right|_{\theta_0} (\theta - \theta_0) \quad (1)$$

- Early stages: linearization OFF (robust exploration)
- Later stages: linearization ON (fast and accurate near MAP)
- Update point: `update_linearization_point(θ_0)` invalidates caches and recomputes

When do we linearize? (concept)



- Enable linearization only after the posterior mass concentrates (avoid bias / instability).
- Track ROM error and $\|\Delta\theta_0\|$ to detect bad updates.

- `run_deterministic`: time integration with Newton solve per step
- `compute_Q_vector`, `compute_Jacobian_matrix`: residual and Jacobian
- Time-dependent antibiotics via `alpha_schedule` (`switch_time/step/frac`)

- **Largest:** likelihood definition (σ_{obs} , variance model; Cov inclusion in Var)
- **Large:** ROM validity (ROM error, linearization update rules, analytical derivatives)
- **Medium:** numerical stabilization (dt, Newton tolerances, clipping/penalties)
- **Medium:** TMCMC settings (particles, stages, mutation steps)

- **Largest:** `BiofilmTSM_Analytical.solve_tsm()`
- **Largest:** `BiofilmNewtonSolver.run_deterministic()` (Q/J + Newton)
- **Large:** sensitivity $x^{(1)}$ generation (esp. before linearization kicks in)
- **Medium:** TCMC mechanics (resample/mutation/ β update)

Reproducibility (audit artifacts)

- `config.json`: seeds and full configuration
- `likelihood_meta*.json`: explicit likelihood definition
- `diagnostics_tables/*.csv`: β , acceptance, ROM error, θ_0 history
- `REPORT.md`: PASS/WARN/FAIL summary

One-command pipeline:

```
python tmcmc/run_pipeline.py --mode paper --seed 123 --run-id  
paper_M1_seed123_fixed --models M1 --lock-paper-conditions  
--use-paper-analytical
```

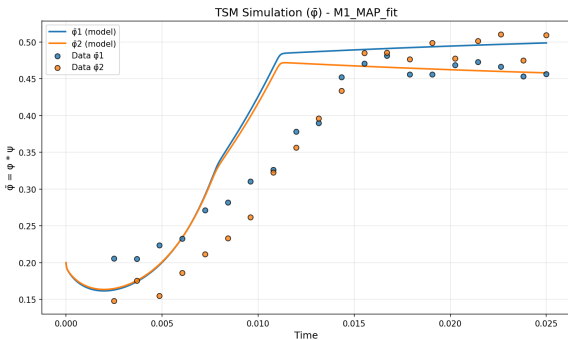
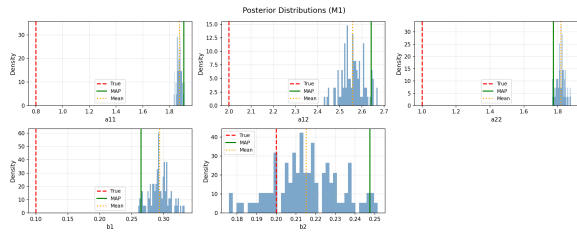
Paper-fixed conditions: `sigma_obs=0.01`, `cov_rel=0.005`

Figure ideas

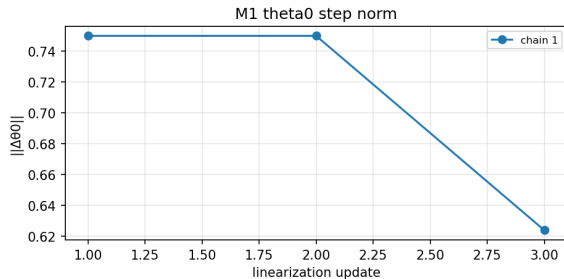
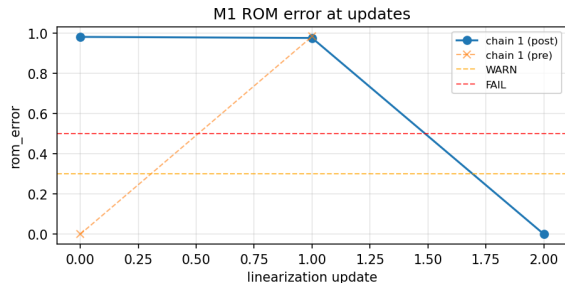
- β schedule (per chain)
- ROM error at linearization updates (pre/post)
- $\|\Delta\theta_0\|$ history
- Posterior plots and MAP/MEAN fits
- Cost-accuracy tradeoff

Example results (auto-picked best run)

Best run id: m1_check_np100_ns15. Figures are embedded only if present.



Example diagnostics (ROM validity)



- Pipeline: `run_pipeline` \rightarrow `case2` \rightarrow `make_report`
- Runtime dominated by `solve_tsm` and Newton time stepping
- Audit essentials: $\beta = 1$ **reached** and explicit likelihood definition

PASS checklist (before claiming results)

Minimum acceptance criteria

- β reached 1.0 for all chains (posterior reached)
- `likelihood_meta_*.json` archived (variance model is explicit)
- No NaN/Inf in solver logs; diagnostics CSVs exist
- ROM validity monitored (ROM error + $\|\Delta\theta_0\|$ are stable)

References



Fritsch, L., Geisler, H., Grashorn, J., Klempt, F., Soleimani, M., Broggi, M., Junker, P., Beer, M. *Bayesian updating of bacterial microfilms under hybrid uncertainties with a novel surrogate model*. tmcnc/Bayesian updating of bacterial microfilms under hybrid uncertainties with a novel surrogate model - Kopie.pdf.



Klempt, F., Geisler, H., Soleimani, M., Junker, P. *A continuum multi-species biofilm model with a novel interaction scheme*. arXiv:2509.01274v1 (2025). tmcnc/biofilm_simulation.pdf.



Junker, P., Balzani, D. *An extended Hamilton principle as unifying theory for coupled problems and dissipative microstructure evolution*. Continuum Mechanics and Thermodynamics (2021). DOI: 10.1007/s00161-021-01017-z. tmcnc/hamiltonian.pdf.



Heine, N. et al. *Influence of species composition and cultivation condition on peri-implant biofilm dysbiosis in vitro*. Frontiers in Oral Health (2025). DOI: 10.3389/froh.2025.1649419. tmcnc/Influence of species composition and cultivation condition on peri-implant biofilm dysbiosis in vitro.pdf.