Review of the manuscript *"Exploiting Physics-Based Machine Learning to Quantify Geodynamic Effects – Insights from the Alpine Region"* by Denise Degen, Ajay Kumar, Magdalena Scheck-Wenderoth, and Mauro Cacace.

This paper presents the construction of surrogate models for vertical surface velocity and topography in the Alpine region, using the non-intrusive reduced basis method - a technique that falls within the broader category of physics-informed machine learning. The study investigates three training datasets, each representing either isolated variations in density, viscosity, or a combination of both. Forward simulations used to generate these datasets were carried out using the parallel staggered-grid finite-difference thermo-mechanical code LaMEM (Kaus et al., 2016).

The development of surrogate models represents a noteworthy and valuable contribution. However, it is important to express concerns regarding the reliability of the underlying forward models. Several simulations terminated prematurely due to a crash in a third-party library (details are provided in Appendix A). These failed simulations were not correctly identified and were instead interpreted as successful, albeit exhibiting "instabilities" such as low topography and high vertical surface velocities. In reality, these anomalies are attributable to the premature termination of the simulation, which prevented the system from reaching isostatic equilibrium. Substituting the crashing solver component with an alternative resolves the issue entirely.

Although the affected simulations are explicitly reported only for the combined training set, it cannot be definitively ruled out that similar problems may exist in the other sets. Therefore, it is strongly recommended that all forward simulations be recomputed to ensure consistency and reliability. As this task exceeds the scope of a standard major revision, the manuscript should be reconsidered for resubmission after the complete regeneration of the training datasets.

To reduce the computational cost associated with regenerating these datasets, an optimized set of input parameters and solver configurations may be employed. Appendix B summarizes a series of such optimizations, which together yield a reduction in run time by more than two orders of magnitude.

Rather than including the datasets themselves, the supplementary material should provide scripts used to extract relevant information from the simulation outputs. It is also recommended to include automation scripts for input file generation and forward model execution. Appendix C presents an example Python implementation of this functionality.

**Detailed Comments**

Section 4.3

Original statements:

*"For the combined scenario (Figure 5), we obtain unrealistically low heights"*

*"For the vertical surface velocities (Figure 6), we obtain extremely high uplift rates"*

The simulations corresponding to realizations 82 and 95 (Figures 5 and 6) crashed prematurely due to issues with a third-party solver library (see Appendix A). As a result, the models did not evolve to a state of isostatic equilibrium, naturally producing low topography and elevated uplift rates. These results were misinterpreted as valid, while in fact they represent incomplete simulations. Notably, LaMEM emitted error messages during these runs, which appear to have been overlooked.

Section 5

*"A last note on the observed instabilities (Section 4.3)."*

Referring to these outcomes as “instabilities” may be misleading, as no supporting evidence is provided for such a classification. The patterns shown in Figures 5 and 6 do not resemble commonly known numerical instabilities in geodynamic modeling (e.g. checkerboard pressure or drunken sailor instability, see van Zelst et al., 2022). Instead, the anomalies have a straightforward explanation: the simulations were interrupted before equilibrium could be reached. Replacing the problematic solver (see Appendix A) allows the simulations to complete normally and yield expected results.

*“Considering the randomness of the observed numerical instabilities …”*

This interpretation seems speculative, as the unexpected results are unrelated to the advection scheme, free-surface stabilization method, or any internal numerical configurations. The root cause lies in the third-party solver failure.

*“To test that the observed instabilities are associated …”*

Comparisons made between LaMEM and other geodynamic codes, such as ASPECT, are not meaningful if the LaMEM simulations being compared are incomplete due to solver crashes. Such comparisons cannot yield reliable insights.

*“We note that, for the combined scenario, vertical velocities …”*

Again, this observation is not valid for the reasons mentioned above. Since the LaMEM models did not run to completion, their outputs cannot be considered representative or comparable.

*“The instabilities are a consequence of …”*

The unexpected results should not be attributed to model complexity or software limitations. Instead, they stem from the failure to verify the integrity of simulation outputs. It is crucial to check for error messages and to take appropriate corrective actions, such as contacting developers or exploring alternative solver configurations.

**Appendix A: Solver Crash Details**

LaMEM supports multiple sparse direct solvers for coarse-grid solutions, including PaStiX, MUMPS, UMFPACK, MKL PARDISO, and SuperLU_DIST. It is advisable to configure PETSc with more than one of these solvers, allowing for fallback options when issues arise.

In particular, MUMPS - used as the default coarse-grid solver in LaMEM - occasionally fails when used within a custom staggered-grid multigrid framework, resulting in a loss of numerical precision. This manifests as the PETSc error DIVERGED_NANORINF.

```
--------------------------------------------------------------------------
  0 SNES Function norm 1.206512042730e-01
  0 PICARD ||F||/||F0||=1.000000e+00
    Linear js_ solve did not converge due to DIVERGED_NANORINF iterations 0
--------------------------------------------------------------------------
**************   NONLINEAR SOLVER FAILED TO CONVERGE!   ******************
--------------------------------------------------------------------------
SNES Divergence Reason  : the linear solve failed

===================================================================================
=   BAD TERMINATION OF ONE OF YOUR APPLICATION PROCESSES
=   PID 1040679 RUNNING AT [redacted]
=   EXIT CODE: 9
=   CLEANING UP REMAINING PROCESSES
=   YOU CAN IGNORE THE BELOW CLEANUP MESSAGES
===================================================================================
YOUR APPLICATION TERMINATED WITH THE EXIT STRING: Killed (signal 9)
This typically refers to a problem with your application.
Please see the FAQ page for debugging suggestions
```

It must be emphasized that this issue does not reflect any inherent flaw in MUMPS, which remains a robust and widely used library. The problem arises from the specific configuration used within LaMEM and may be resolved by exploring custom solver settings.

The simulations in question (realizations 82, 89, and 95 from the combined training set) all exhibit this crash. An example input file from one of the failed simulations (m82_crash.dat) is included with this review. Using an alternative solver, such as SuperLU_DIST, resolves the problem. A modified input file (m82_stable.dat) demonstrates this fix:

MGCoarseSolver = superlu_dist

**Appendix B: Input File Optimization**

The reference input file (alps_lith_slabs_400km_HR_reg.dat) provided in Kumar (2025) employs several suboptimal settings. The following adjustments, reflected in the optimized file m82_fast.dat, result in substantial performance improvements:

- Increase maximum time step from 200 to 20000 years (dt_max = 0.02)
- Use forward Euler advection scheme (advect = basic)
- Enable subgrid marker control (mark_ctrl = subgrid)
- Disable unnecessary output fields
- Use low-level solver options
- Skip nonlinear iteration for linear material models (-snes_type ksponly)
- Use absolute tolerance for linear solver (-js_ksp_atol 1e-3)
- Increase number of multigrid levels from 3 to 5 (-gmg_pc_mg_levels 5)
- Reduce number of smoothing sweeps (-gmg_mg_levels_ksp_max_it 5)
- Use less MPI processes for a small coarse-grid problem (PETSc Telescope)

These changes dramatically reduce simulation time without compromising result quality. For example, the runtime of realization 82 was reduced from 1 day 17 hours to approximately 14 minutes. This corresponds to a speedup factor of 182.

```
-----------------------------------------------------------------
========================== SOLUTION IS DONE! ====================
-----------------------------------------------------------------
Total solution time : 148947. (sec)
-----------------------------------------------------------------
```

```
-----------------------------------------------------------------
========================== SOLUTION IS DONE! ====================
-----------------------------------------------------------------
Total solution time : 816.364 (sec)
-----------------------------------------------------------------
```

**Appendix C: Automation Framework**

Kumar (2025) does not include tools for automated input file generation and job submission. Given the large number of simulations involved, manual handling increases the risk of human error.

The provided Python script (submit_runs.py) automates this process via a run_model function, which takes the following inputs:

- A template input file without material parameters
- A parameter table file with one row per realization
- The realization index (starting from 0)
- Number of MPI tasks
- Path to LaMEM executable
- A dry-run flag

The script appends material parameters, creates a directory, generates a slurm batch script, and, if not in dry-run mode, submits the job. It can be easily modified to accommodate alternative workload managers if slurm is not available.

Three template input files are included: crash.dat, stable.dat, and fast.dat, corresponding to different solver setups discussed in this review.

## References

Ajay Kumar. (2025). LaMEM and ASPECT input and data files corresponding to Exploiting Physics-Based Machine Learning to Quantify Geodynamic Effects – Insights from the Alpine Region [Data set]. Zenodo. https://doi.org/10.5281/zenodo.15478977

Kaus, B. J. P., Popov, A. A., Baumann, T. S., Pusok, A. E., Bauville, A., Fernandez, N., Collignon, M., 2016. Forward and Inverse Modelling of Lithospheric Deformation on Geological Timescales. NIC Series, 48, 978–3.

van Zelst, I., F. Crameri, A.E. Pusok, A.C. Glerum, J. Dannberg, C. Thieulot (2022), 101 geodynamic modelling: how to design, interpret, and communicate numerical studies of the solid Earth, Solid Earth, 13, 583–637, doi:10.5194/se-13-583-2022