

Underworld2: Python Geodynamics Modelling for Desktop, HPC and Cloud

John Mansour¹, Julian Giordani², Louis Moresi^{3,2}, Romain Beucher², Owen Kaluza¹, Mirko Velic², Rebecca Farrington², Steve Quenette¹, and Adam Beall⁴

1 Monash eResearch Centre, Monash University, Clayton, Australia. **2** School of Earth Science, The University of Melbourne, Melbourne, Australia. **3** Research School of Earth Sciences, The Australian National University, Canberra, Australia. **4** School of Earth and Ocean Sciences, Cardiff University, Cardiff, UK.

DOI: [10.21105/joss.01797](https://doi.org/10.21105/joss.01797)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Leonardo Uieda](#) ↗

Reviewers:

- [@gassmoeller](#)
- [@gabersyd](#)

Submitted: 03 September 2019

Published: 06 March 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

The modelling of long time-scale solid-Earth processes presents challenges on many fronts. Typically a large range of length scales must be resolved, and computational requirements can easily go beyond a desktop machine's capability, even in 2-dimensions. Rheological behaviours are diverse and modellers often wish to investigate a spectrum of dynamics. With this also comes the requirement of robust solution strategies capable of capturing a system of large dynamic range. Time dependent aspects of a model also require careful consideration such that numerical artefacts do not subsume the required signal. `Underworld2` is a Python API which seeks to meet these challenges, while also providing a friendly and intuitive interface to model development.

Designed to work (almost) seamlessly across PC, cloud and HPC infrastructure, `Underworld2` consists of a set of Python classes, instances of which will be used to construct numerical geodynamics models. The API also provides the tools required for inline analysis and data management. `Underworld2` continues the legacy of `Underworld1` (Moresi, Dufour, & Mühlhaus, 2002, 2003; Moresi et al., 2007), sharing much of the underlying mathematical approach. But `Underworld2` is not a Python wrapper to `Underworld1`, and modelling paradigms have been completely redesigned to best fit the Python language.

`Underworld2` provides capacity for modelling 2- and 3-dimensional geodynamics processes, utilising a particle-in-cell finite element approach for solving Stokes flow type configurations. For the multi-processor scalability required of high-resolution models, we use MPI (Message Passing Interface) based domain-decomposition, and for performant operation all heavy computations are executed within a statically typed layer. In `Underworld2`, the finite element mesh can be static or dynamic, but it is not constrained to move in lock-step with the evolving geometry of the fluid. This hybrid approach allows for the accurate solution to the velocity problem (on the mesh) for a given material configuration, while simultaneously ensuring the accurate tracking of material interfaces and history information (using particle swarms). Templated systems are provided for solutions to Stokes flow, steady state heat (or Darcy) flow, and time-dependent advection-diffusion models.

A primary aim of `Underworld2` is to enable rapid and frictionless model development. To this end, the rich syntactic possibilities of Python have been leveraged empowering users with intuitive access to complex operations, while paternalistic patterns help to guarantee data integrity and ensure consistent parallel operation. By design, `Underworld2` provides only a minimal set of flexible core functionality, and leaves *user domain* concerns to the users themselves to construct. At the centre of this design is the `Function` class, which aims to

provide a natural interface for the description of problem dynamics. For example, the user may describe a viscosity which is piecewise constant, temperature dependent, or visco-elasto-plastic in behaviour. Simulation chronology is also fully exposed and the user is expected to explicitly specify when events should occur. This free-form approach allows users to define exactly the physics they require, and more importantly provides transparency.

Although not a requirement, `Underworld2` is very much at home within the Jupyter (Kluyver et al., 2016) environment. When coupled with the LavaVu (Kaluza, Moresi, Mansour, & Barnes, 2019) visualisation engine, users are able to interactively develop models piecemeal. Jupyter also provides a path to cloud amenability, with users accessing their model Notebooks via any standard web browser. Cloud based operation has been used successfully within teaching environments, and also enables our *live* user documentation and examples.

`Underworld2` is utilised by the `UWGeodynamics` Python module (Beucher et al., 2019), which provides a more structured interface to geodynamics model construction. It has enabled numerous recent publications, including Beall & Moresi (2018), Mériaux, May, Mansour, Chen, & Kaluza (2018) and Sandiford & Moresi (2019).

Acknowledgements

`Underworld` development was financially supported by AuScope (<https://www.auscope.org.au>) and the Australian Government via the National Collaborative Research Infrastructure Strategy (NCRIS). Additional funding for specific improvements and additional functionality has come from the Australian Research Council (<http://www.arc.gov.au>). The Python tool-kit was funded by the NeCTAR eResearch Tools program.

References

- Beall, A., & Moresi, C. C. M. (2018). Formation of cratonic lithosphere during the initiation of plate tectonics. *Geology*, *46*(6), 487–490. doi:[10.1130/G39943.1](https://doi.org/10.1130/G39943.1)
- Beucher, R., Moresi, L., Giordani, J., Mansour, J., Sandiford, D., Farrington, R., Mondy, L., et al. (2019). `UWGeodynamics`: A teaching and research tool for numerical geodynamic modelling. *Journal of Open Source Software*, *4*(36), 1136. doi:[10.21105/joss.01136](https://doi.org/10.21105/joss.01136)
- Kaluza, O., Moresi, L., Mansour, J., & Barnes, D. G. (2019). `OKaluza/LavaVu: V1.4.3`. Zenodo. doi:[10.5281/zenodo.3348451](https://doi.org/10.5281/zenodo.3348451)
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., et al. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. (F. Loizides & B. Schmidt, Eds.). IOS Press. doi:[10.3233/978-1-61499-649-1-87](https://doi.org/10.3233/978-1-61499-649-1-87)
- Mériaux, C. A., May, D. A., Mansour, J., Chen, Z., & Kaluza, O. (2018). Benchmark of three-dimensional numerical models of subduction against a laboratory experiment. *Physics of the Earth and Planetary Interiors*, *283*. doi:[10.1016/j.pepi.2018.07.009](https://doi.org/10.1016/j.pepi.2018.07.009)
- Moresi, L., Dufour, F., & Mühlhaus, H.-B. (2002). Mantle convection modeling with viscoelastic/brittle lithosphere: Numerical methodology and plate tectonic modeling. *Pure and Applied Geophysics*, *159*(10), 2335–2356. doi:[10.1007/s00024-002-8738-3](https://doi.org/10.1007/s00024-002-8738-3)
- Moresi, L., Dufour, F., & Mühlhaus, H.-B. (2003). A lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. *Journal of Computational Physics*, *184*(2), 476–497. doi:[10.1016/S0021-9991\(02\)00031-1](https://doi.org/10.1016/S0021-9991(02)00031-1)

- Moresi, L., Quenette, S., Lemiale, V., Mériaux, C., Appelbe, B., & Mühlhaus, H.-B. (2007). Computational approaches to studying non-linear dynamics of the crust and mantle. *Physics of the Earth and Planetary Interiors*, 163(1–4), 69–82. doi:[10.1016/j.pepi.2007.06.009](https://doi.org/10.1016/j.pepi.2007.06.009)
- Sandiford, D., & Moresi, L. (2019). Improving subduction interface implementation in dynamic numerical models. *Solid Earth*, 10, 969–985. doi:[10.5194/se-10-969-2019](https://doi.org/10.5194/se-10-969-2019)