30 Years of Citcom, Ellipsis and Underworld

Louis Moresi

Published January 16, 2024

Citation

Keywords
Underworld Code, Geophysics, Auscope

Abstract
CITCOM is a geodynamics modelling code based on the finite element method that is designed for planetary evolution modelling where large spatial variations and strong non-linearities occur in the material properties.

Copyright
Copyright © Louis Moresi 2024. Distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
CITCOM is a geodynamics modelling code based on the finite element method that is designed for planetary evolution modelling where large spatial variations and strong non-linearities occur in the material properties. CITCOM was originally conceived by Louis Moresi working at Caltech with Professor David Stevenson but quickly developed into an open-source community software effort with the work of Shijie Zhong, Slava Solomatov, Mike Gurnis and many others. CITCOM spun off many novel software tools including CITCOM-S, CITCOM-SVE, Ellipsis and Underworld. In this article, Louis Moresi reflects on a 30 year community-software development journey.

We mark January 17, 1994 (the day better known for the Northridge Earthquake) as the birthday of the CITCOM family of codes. I was shaken awake early in the morning, and spent the rest of the day nervously waiting for the next aftershock while I checked that the code was really solving all the test problems quickly and accurately. CITCOM took more than a year to get right and many deadlines went by un-met as a result including the pre-Christmas deadline of the 1993 AGU Fall Meeting.

There were two notable papers that eventually came out of this early version of CITCOM: 1) Moresi and Solomatov (1995) where we were able to model fully-stagnant lid convection for the first time and demonstrate how all the theoretical scalings really worked; and 2) Moresi, Zhong & Gurnis (1996) where we proved that the method itself was accurate using analytical solutions. In the following two years we expanded on these topics in numerous other papers, Shijie Zhong developed a parallel version of the code, and we formally declared the code to be formally open source and encouraged the community to use it.

When Shijie Zhong and his team developed CITCOM-S (Zhong et al, 2000), the parallel, spherical code that inherited all the robust solvers from the Cartesian version, the community embraced the code and it became one of the most widely used software tools in the geodynamics community and continues to be used today.

The other branch of this family tree took a different path: the Ellipsis code was developed to study history-dependent fluid-dynamics problems (e.g. materials with damage, anisotropic flow, viscoelastic convection) using Lagrangian particles embedded in the finite element mesh (Moresi et al, 2003). Ellipsis became Underworld when we adopted the PETSc parallel computing framework and received funding from AuScope to develop a community of users and support their research needs.

As the CITCOM codes celebrate 30 years, and active development of Underworld passes the 20 year mark, we will be unveiling our next generation of software, Underworld3.
Underworld3 is integrated with Jupyter and with sympy. The development environment is easy to use, close to the original mathematical formulation and highly interactive. Once ready for deployment, parallel execution is efficient and scalable - no modification of the notebooks is needed.

Underworld3 continues the long collaboration with the PETSc team. We have completely re-engineered the internal structure of the code to leverage the PETSc computational infrastructure while focusing on ease of use for geophysical fluid dynamics problems. We introduce a pure symbolic interface to the conservations equations that builds the python sympy system — this provides automatic differentiation of PDEs (required for non-linear problems and optimisation for example) but also simplification, code-compilation, and mathematical verification. This system puts Lagrangian and Eulerian formulations on a completely equal footing and makes the two representations inter-operable. Parallel scaling is robust and reliable thanks to PETSc and, at the same time, the code is very straightforward to follow because the built-in documentation is perfectly integrated with Jupyter.

Stay tuned for a series of posts introducing Underworld3 and, this time (fingers crossed) no natural disasters to accompany the release.
References


