Vis Geosci
DOI 10.1007/s10069-007-0006-3

ORIGINAL ARTICLE

Increasing the usability and accessibility of geodynamic modelling tools to the geoscience community: UnderworldGUI

Shea Goyette · Masa Takatsuka · Stuart Clark · R. Dietmar Müller · Patrice Rey · Dave R. Stegman

Received: 13 February 2007 / Revised: 22 April 2007 / Accepted: 27 April 2007

Abstract Geoscientists are faced with a number of complexities that represent obstacles to the development of realistic simulation of deep earth processes. Realistic 4D thermo-mechanical simulation using software packages like Underworld and Gale, when combined appropriately with geoscientific expertise, can lead to novel insights into the deformation of geological structures at a wide range of time and spatial scales. The challenge for end-user geoscientists lies in applying their knowledge within the framework of the software's input specification, including initial, internal, and boundary conditions and output visualization parameters. We have built a Graphical User Interface (GUI) to remove many of the difficulties related to editing the Extensible Markup Language (XML) encoded input files of Underworld/Gale geomodels and therefore, to greatly broaden the user base of these software packages. By helping Underworld/Gale to meet a large audience, we provide a tool to the geoscience community that helps to move from untested conceptual models to physically valid, properly scaled modelling. Furthermore, the UnderworldGUI offers a mechanism for storing and retrieving experimental models in a centralised database, thus providing the geoscience community with a means to share the outcomes of its experimental research. Further details of the UnderworldGUI are available at the web site http://www.wiki.vislab.usyd.edu.au/moinwiki/UnderworldGUI.

S. Goyette (✉) · M. Takatsuka
Vislab, University of Sydney,
Sydney, NSW 2006, Australia
e-mail: sgoyette@vislab.usyd.edu.au

S. Clark · R. D. Müller · P. Rey
Earthbyte Group, School of Geosciences,
University of Sydney, Sydney, NSW 2006, Australia

D. R. Stegman
School of Mathematical Sciences, Monash University,
Clayton, VIC 3800, Australia

Keywords Finite-element · Gale · Particle-in-cell · Underworld · User interface

Introduction

Computer simulation is used to solve highly non-linear problems whose complexity, size, or time span is so great that analytical solutions do no exist. This is particularly relevant for understanding geo-processes over geological time scales, such as plate tectonics, mantle convection and plate deformation. It is therefore, advantageous for geoscientists to employ 4D computer simulations in their research (Moresi et al. 2003). A number of codes for the modelling of long-term deformation of the Earth's lithosphere and convecting mantle have been developed. Many of these geodynamic modelling codes use a vast range of input parameters that allow for fine-tuned control of the layout of the model and simulation parameters. Input files for particle-in-cell, finite-element codes typically allow the user to select from a range of grid solvers, internal, boundary and initial conditions, geometric components and material thermo-mechanical properties, and to specify how and when simulation output is recorded onto the local disk (O'Neill et al. 2006).

Two codes in particular are discussed in this paper—Underworld and Gale. Underworld is a Lagrangian–Eulerian particle finite-element code, which incorporates Mohr-Coulomb and Drucker-Prager failure models in two and three dimensions. Its software stack incorporates the StGermain, StgFEM, PICellerator, and gLucifer libraries.

Its applications are formed out of building on top of StgFEM and PICellerator extension modules, which are assembled for solving a particular task.[1] Underworld allows the geoscientist to simulate geodynamic processes, and observe a variety of outcomes based on parameters that the user specifies. Gale is derived from Underworld, and also represents a parallel, two- or three-dimensional, finite-element, Lagrangian–Eulerian code for solving problems related to orogenesis, rifting, and subduction with coupling to surface erosion models.[2] Although Gale makes use of many of the same libraries as the Underworld code, its main point of departure from the Underworld code is that it does not use the gLucifer real-time visualization library (Stegman et al. 2005), instead outputting the simulation data in raw form. The UnderworldGUI was initially designed for manipulating Underworld input models, but it can also be used to manipulate Gale input models. At their current stage of development, the input specifications for Gale and Underworld hold a great deal in common. As Gale diverges from Underworld, some work will be needed to ensure that the UnderworldGUI will be compliant with the latest Gale input specification.

Both of these codes typically work with large parameter sets. In order to manage the many aspects of building the model, the user is required to have extensive knowledge of the input specifications. This includes an understanding of the input grammar and a grasp of the necessary vocabulary to express the features of the desired model, both of which evolve throughout the development of the code. Furthermore, even with this knowledge, the creation of models by direct manipulation of the input Extensible Markup Language (XML) files is a meticulous task. Software like the EllipsisGUI, designed for the ellipsis3D software (O'Neill et al. 2006), a non-parallel predecessor of Underworld, has aimed to simplify the process of building models for particle-in-cell simulation (Dyksterhuis et al. 2007). Simulation experiments that are carried out with software like Underworld and Gale are most powerful when employed as part of an iterative process. Due to the scale and sheer multiplicity of variables involved in modelling deep earth processes, honing in on the correct set of parameters is a fine art that relies heavily upon the expertise of the geoscientist. Experience and careful analysis guide the geoscientist to their initial model, but often adjustments need to be made, by observing the simulation output and hypothesizing about the effects of changing particular variables. By repeating this process, the geoscientist can develop a more accurate model of the deep earth processes they are investigating.

In order to address the needs of the geoscientific community, and facilitate the sharing of simulation tools, results, and ideas, we have developed a Graphical User Interface (GUI) to the Underworld code. Two main goals have been set out for the development of the UnderworldGUI:

- To allow the user to easily design and manage both model and visualization input parameters used by Underworld/Gale and gLucifer.
- To provide a means of sharing the outcomes of these simulations with other users.

The UnderworldGUI simplifies the process of creating input models for Underworld-driven simulations, submitting these models to the Underworld software, and retrieving the simulation results upon completion of the simulation. To facilitate scientific collaboration and dialogue, and to assist individual users in managing their work, the UnderworldGUI is able to store each model that is submitted to the Underworld as an XML input file, along with a sample of its simulation output, to a database of previously run models. Another consideration of the development process has been to maintain a modular code base that allows for easy adaptation to new input specifications, and integration with new simulation codes. The UnderworldGUI is currently deployed as a Java application that runs locally on the user's machine. It is platform-independent, and is being developed to accommodate its possible future implementation as a web-accessible scientific tool.

We will discuss the motivation behind including particular features, implementation and design choices, and explain how each pertains to the broad goals of the UnderworldGUI project, to simplify the modelling process and facilitate the sharing of simulation outcomes for the users of the Underworld code. Furthermore, as we have already stated, the UnderworldGUI has been designed to be adaptable to new codes. Given that Underworld and Gale share a large proportion of their code, it is highly likely that UnderworldGUI will also be deployed as a GUI for the Gale code in the near future. We will elaborate on this point in the next section, where we discuss, amongst other things, the protocol of communication between the UnderworldGUI and the Underworld software stack. Further information is also available from the UnderworldGUI web site, at http://www.wiki.vislab.usyd.edu.au/moinwiki/UnderworldGUI.

Interfacing with the user

UnderworldGUI offers the user a variety of methods for creating and editing geodynamic models, and in conjunc-

[1] See the APAC National Facility's Underworld Web site at http://www.nf.apac.edu.au/facilities/software/software.php?software = Underworld, last accessed 19 January 2007.

[2] See the Computational Infrastructure for Geodynamics' Gale Web site at http://www.geodynamics.org/cig/software/packages/long/gale/, last accessed January 24, 2007.

tion with the Underworld code, simulating changes in these models over time. The main window of the GUI consists of three parts: a menu, a toolbar, and an area for editing Underworld models. The menu and toolbar are essentially equivalent, and you can use whichever set of controls you prefer (Fig. 1). These controls allow the user to create, load, and save Underworld models; import and export models from and to a model library; select different editors; enable and disable debugging options; change the visual theme; and execute external processes to flatten input XML files and initiate Underworld simulations.

An Underworld model consists primarily of three types of data:

- a number of geo-materials involved in the model that need to be fully specified in terms of their geometry and distribution throughout the model, and their rheological nature (how the behaviour of their deformation is controlled by temperature, stress, strain, composition, and previous deformation);
- a number of visualisation components that define how the model is visualised during the simulation;
- a number of parameters that pertain to the systems of equations used in the simulation process (initial, internal and boundary conditions), and the numerical algorithms implemented for obtaining approximate solutions to the physical system.

All of these data are stored in an XML file that conforms to the StGermain specification.[3]

When the GUI is run for the first time, the area for editing Underworld models will be blank (Fig. 2). Here, the user can create a new Underworld model from scratch. However, the user may wish to load a pre-existing or template model. There are two methods for opening an Underworld model input file. The first is to open the XML file corresponding to the desired model directly, by loading an XML file from the user's local disk. This may be one of the sample input files included with the Underworld software, or any XML file that has been saved on the local disk. The second method of opening an Underworld model is to import the XML file corresponding to the desired model from a library of previous run models. The library

Fig. 1 UnderworldGUI menu and toolbar

[3] See the Victorian Partnership for Advanced Computing's StGermain Web site at https://www.csd.vpac.org/twiki/bin/view/Stgermain, last accessed 19 January 2007.

may be stored locally on the user's hard disk, or may be located remotely; however, from the user's perspective, there is no difference in the operation of retrieving a model from the library in either case. Conceptually, these two implementations of the library serve different purposes. Administering the library locally, on the user's hard disk, services the needs of an individual user to organize their own work. Administering the library remotely, where it can be publicly accessed, allows users to share their work with each other.

The model editing capabilities of UnderworldGUI consists of three editors. To reiterate, Underworld models are made up of three kinds of data—rheology/geometry components, visualisation components, and modelling parameters imposed on the model—and these data are stored in XML files. The three editors are capable of showing the same data, but display this data in different ways—in Figs. 4, 5 and 6, a lucViewport and a lucCamera object are displayed in XML form, as a list of parameters in name-value pairs, and in a dedicated visualisation component editor. Furthermore, changes in any one of the editors instantly results in an update of the content of the other editors.

The XML representation of the data can be viewed in the XML Tree sidebar (Fig. 3), and both viewed and edited in the Plain Text View (Fig. 4). This view shows the Underworld model in its raw XML format. Editing the model as plain text is powerful, but it is really only suited to expert users, or for making very particular adjustments to the model. For a more user-friendly approach to the data, the components and parameters can be viewed as name-value pairs in the Parameters View (Fig. 5). Editing the model in the Parameters View guarantees compatibility with a particular version of Underworld, and simplifies the process of locating specific components or parameters for adjustment. The design of this editor follows an object-oriented pattern, such that the data structures inherent in the input XML files are modelled as components in the GUI. Each of these components keeps a reference to the datum in the model that it represents, and registers itself as a listener to that datum, to be notified in the event of that datum being modified. This allows the component to modify that datum directly, and to respond instantly to any changes in that datum. This is particularly effective for manipulating the parameters that make up the geometry and material composition of the model. To edit the visualisation parameters of the model—parameters that are specific to the gLucifer component of the Underworld code – the user may prefer a more visual editor. This more visual representation of the visualisation components is available by selecting the Viewport Editor View (Fig. 6), which includes a preview of the camera perspective and controls for mapping simulation data to visualisation values.

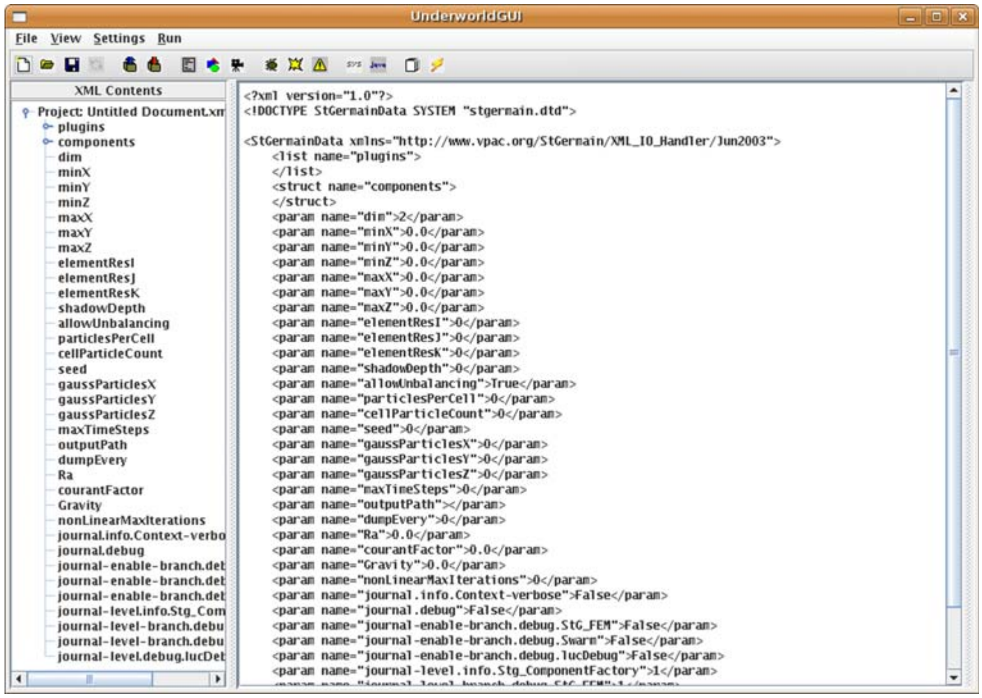


Fig. 2 Initial state of the UnderworldGUI

To run a simulation of the current model using the Underworld code, the GUI submits the currently open model as input to the Underworld software. This operation is called a job request. The actual location of Underworld could potentially be on the user's local disk, or on a remote server, but as with the model library, there is no ostensible difference in the operation of a job request in either case. The UnderworldGUI includes a package of networking classes, with a hierarchy of connection handlers. The parent connection handler defines a strict procedure for communicating with an external process, and child connection handler classes can be written to implement particular types of connections. A connection handler for accessing the local disk is included in the current version of UnderworldGUI, and a connection handler for accessing remote servers will be added in the near future. The simulation is instantiated by initiating the external process, so that any executable can be used. Furthermore, the XML parser is designed to accept any XML file that conforms with the StGermain specification. This makes the use of any StGermain-conformant geoscience simulation code possible, thus the UnderworldGUI can easily be made compatible with the Gale code. An example of running a simulation in the UnderworldGUI is covered in the example section below.

Scientific collaboration

When using simulation to investigate a geodynamic phenomenon, the geoscientist must narrow in on a set of

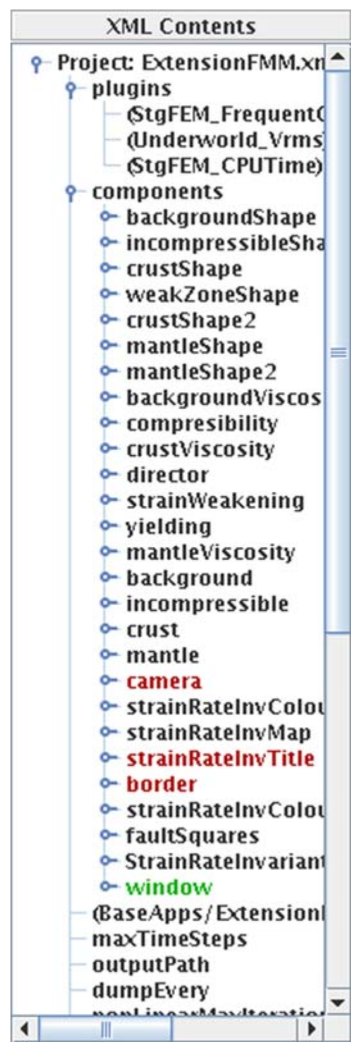
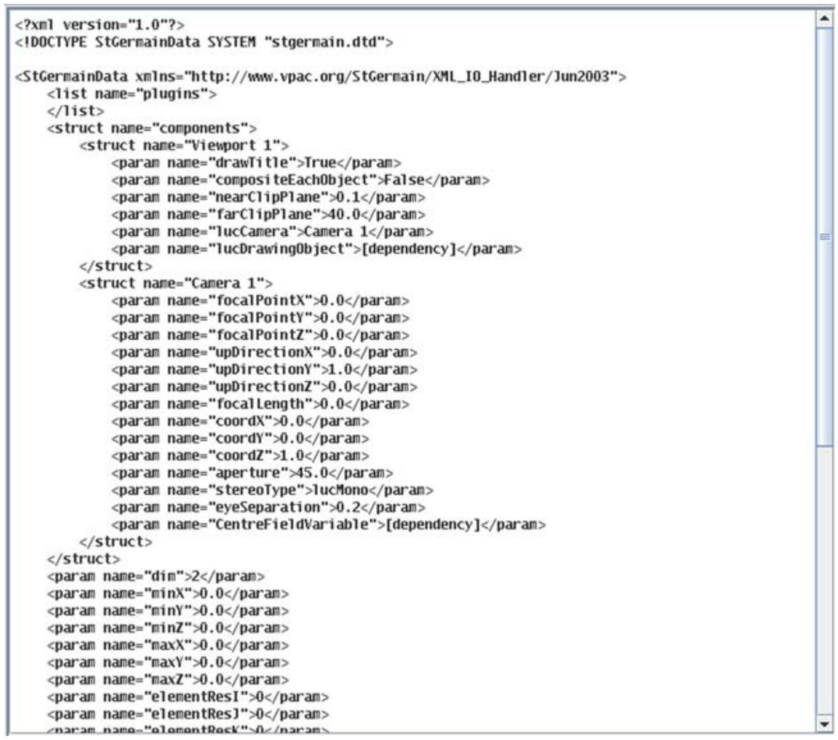


Fig. 3 XML Tree sidebar



```
<?xml version="1.0"?>
<!DOCTYPE StGermainData SYSTEM "stgermain.dtd">

<StGermainData xmlns="http://www.vpac.org/StGermain/XML_IO_Handler/Jun2003">
    <list name="plugins">
    </list>
    <struct name="components">
        <struct name="Viewport 1">
            <param name="drawTitle">True</param>
            <param name="compositeEachObject">False</param>
            <param name="nearClipPlane">0.1</param>
            <param name="farClipPlane">40.0</param>
            <param name="lucCamera">Camera 1</param>
            <param name="lucDrawingObject">[dependency]</param>
        </struct>
        <struct name="Camera 1">
            <param name="focalPointX">0.0</param>
            <param name="focalPointY">0.0</param>
            <param name="focalPointZ">0.0</param>
            <param name="upDirectionX">0.0</param>
            <param name="upDirectionY">1.0</param>
            <param name="upDirectionZ">0.0</param>
            <param name="focalLength">0.0</param>
            <param name="coordX">0.0</param>
            <param name="coordY">0.0</param>
            <param name="coordZ">1.0</param>
            <param name="aperture">45.0</param>
            <param name="stereoType">lucMono</param>
            <param name="eyeSeparation">0.2</param>
            <param name="CentreFieldVariable">[dependency]</param>
        </struct>
    </struct>
    <param name="dim">2</param>
    <param name="minX">0.0</param>
    <param name="minY">0.0</param>
    <param name="minZ">0.0</param>
    <param name="maxX">0.0</param>
    <param name="maxY">0.0</param>
    <param name="maxZ">0.0</param>
    <param name="elementResI">0</param>
    <param name="elementResJ">0</param>
```

Fig. 4 Plain text view of the Underworld model

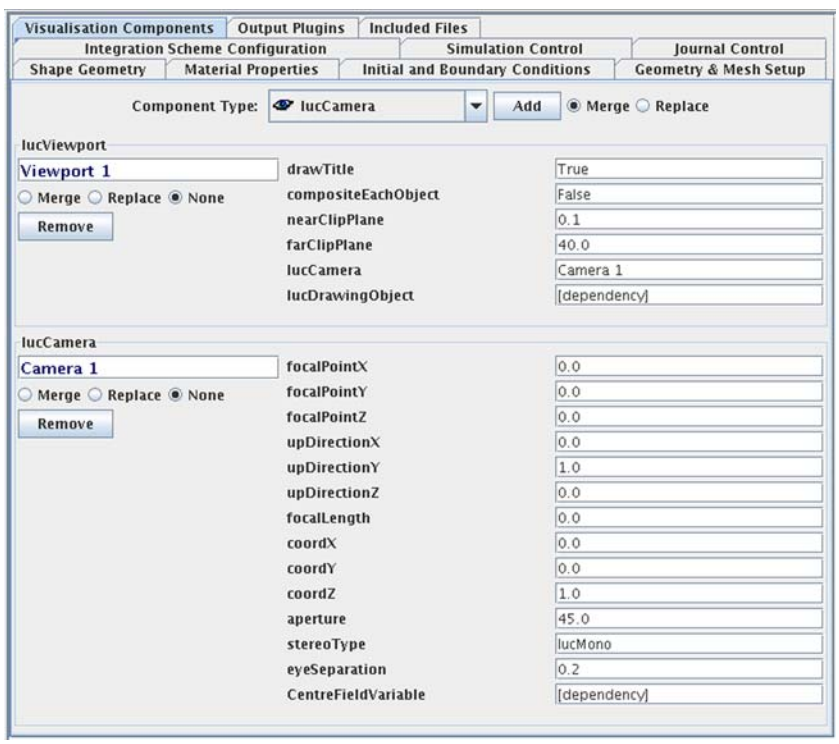


Fig. 5 Parameters view of the Underworld model

parameter values that generate that phenomenon. The process of determining these values begins with a parameter set that reflects an initial hypothesis. One way to enhance this process is formulate the initial hypotheses based on previous experience with simulation outcomes (Boschetti and Moresi 2001). In order to assist the geoscientist with this task, the UnderworldGUI provides a facility for managing previously run experimental models. More importantly, however, this library of models can be made accessible to entire communities of users by storing the models in an

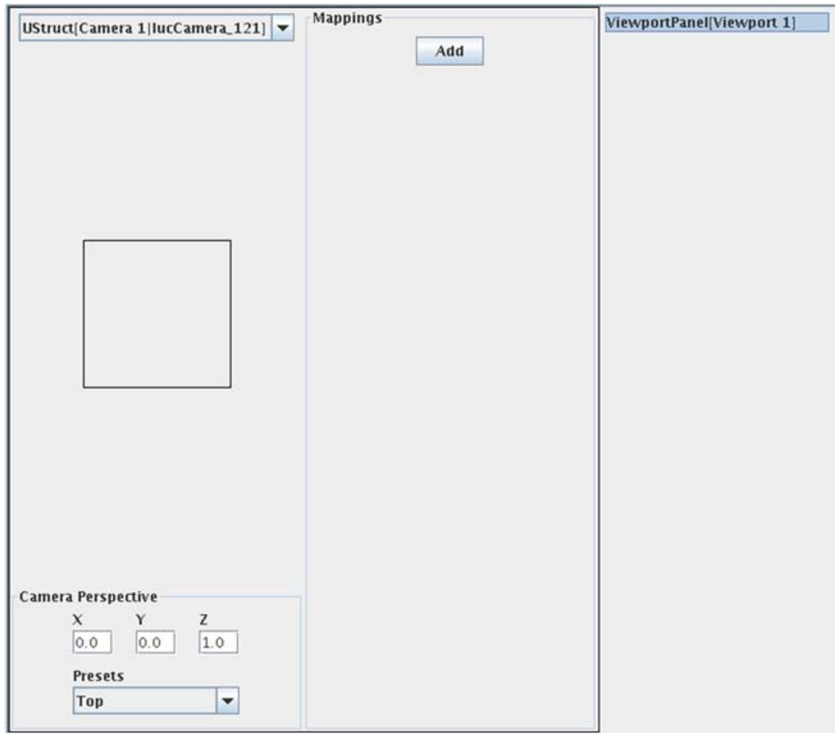


Fig. 6 View port editor view of the Underworld model

SQL database that allows remote connections, giving the geoscientist access to a much wider body of collective experience and expertise. To achieve this, a connection handler for accessing the library as a remote SQL database has been implemented in the UnderworldGUI.

Every time a model submitted to the Underworld code for simulation, a number of output files are generated. These files represent time slices of the simulation run or in the case of video files, the simulation run in its entirety—visualized in different ways by the gLucifer component of the code. When storing models in the library, it is these files that comprise the model's preview, giving the user a sense of the model's important features ''at a glance''. The generation of output files in image and video form is a specific feature of Underworld, however, and some simulation codes, like Gale, produce output data in its raw, numerical form. For this reason, the UnderworldGUI does not automatically store the model in the library, but allows the user to specify their own preview files, and hence to select their own method of visualization.

In the Underworld Library Exporter, users can select up to three images and one video as previews for a model (Fig. 7). These previews assist the user in identifying the important features of that particular simulation run when importing models from the library. Models can be imported in the Underworld Library Importer, where the ''<<'' and ''>>'' buttons allow the user to scroll through up to three preview images that have been taken at different time intervals during the model's simulation (Fig. 8). Display of an optional video preview may also be a useful feature in a future version of the UnderworldGUI.

It is important to note that this model library is designed to be used as a shared library, where a community of geoscientists can collaborate across any distance by sharing their experimental outcomes. Such collaboration may be intended or otherwise; a simulation may not produce useful outcomes for the geoscientist who ran it, but it may expose features that are of some interest to another geoscientist working elsewhere. By indexing these simulation outcomes with project and model descriptions, and preview image and video files that the user themselves chooses, the GUI allows the expert knowledge of its geoscientist users to guide the evolution of experimental models (Boschetti and Moresi 2001).

Example

Underworld subduction modelling has been used to correlate trench width and rollback rates (Stegman et al. 2006). The input template *SlabSubduction.xml*, which is packaged with Underworld, enables the novice to run subduction models. Our aim was to couple subduction modelling with an overriding plate. We have created three models as case studies to test the effect of subduction on the overriding plate. The first model attempts to simulate the first order conditions of the Kamchatka subduction zone in the northwest Pacific (model A), while the second

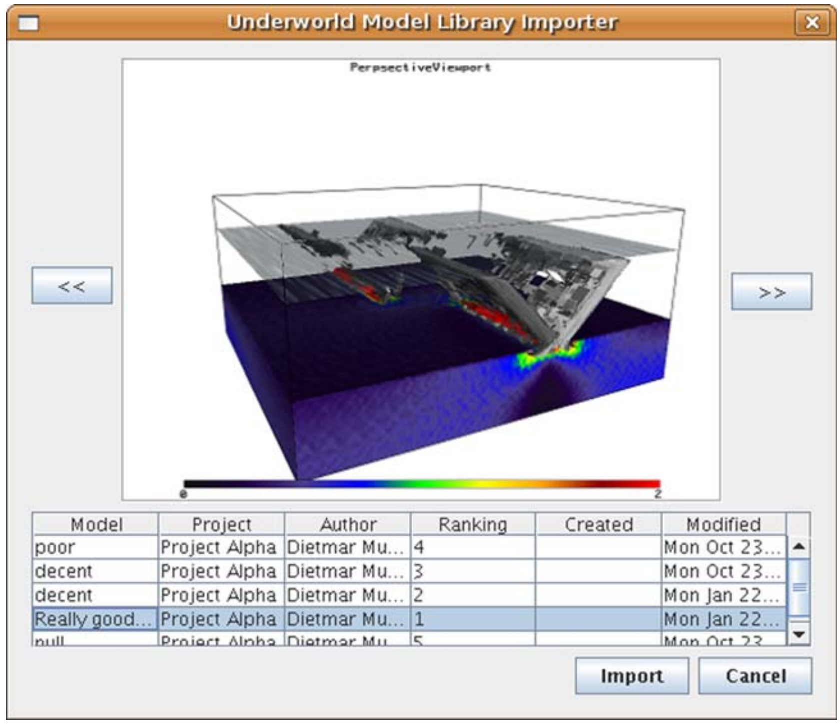


Fig. 7 Dialog for exporting models to the Underworld model library

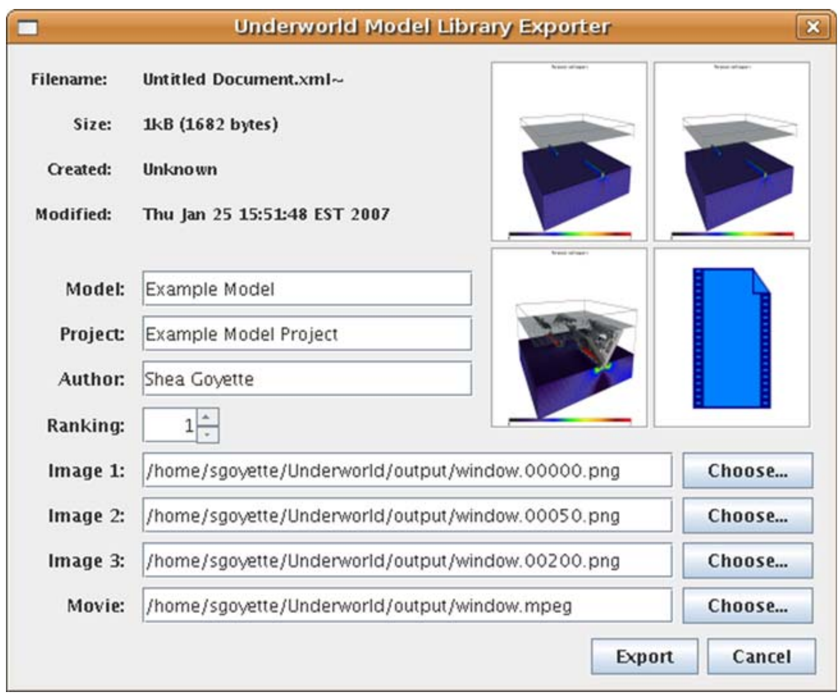


Fig. 8 Dialog for importing models from the Underworld model library

is based on the Caribbean Antilles trench (model B). The last model has been designed to replicate the evolution of the short Hellenic trench in the Mediterranean (model C). To demonstrate the UnderworldGUI's real-world application, we have also run simulations on these models using the UnderworldGUI, and exported the results into a model library.

The three models have been set up with some initial conditions that are common to the set (Fig. 9). In all models, the subduction is simulated to have begun after subduction has been initiated, thereby avoiding complex issues related to subduction initiation. This has been implemented by placing a perturbation of slab material extending from the trench into the mantle at a 60° dip angle

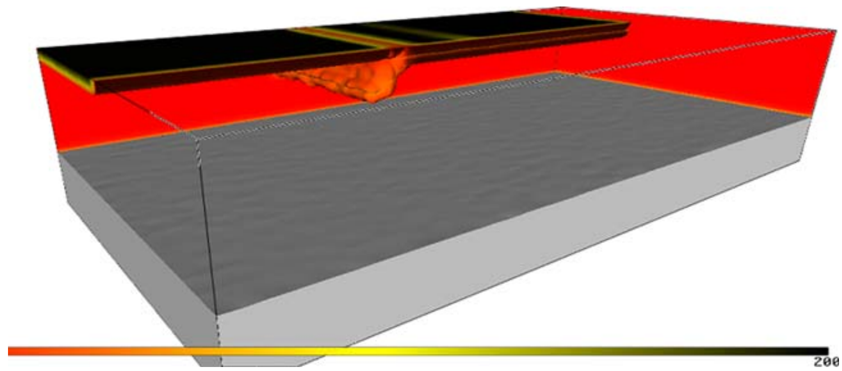


Fig. 9 Initial conditions for the models A–C

until about 600 km depth. The overriding plate and the subducting plate are 80 kg/m^3 denser than the surrounding mantle material and 200 times the upper-mantle viscosity. This density variation between the subducting plate and surrounding materials translates into a buoyancy force that drives subduction. In all three models, the overriding plate has been fixed to the wall (no-slip boundary condition) opposite the trench. This represents the confined ability of the overriding plate to move in all three models, constrained respectively by the Eurasian plate, Central America, and Europe. The timescale of the runs is approximately 30 Myr.

The models have progressively smaller widths. Model A has a wide trench (2,000 km), approximate to the width of the Kuril-Kamchatka trench, while Model B is 1,200 km wide, appropriate for the Antilles trench (Giunta et al. 2006), and Model C is only 600 km, representing the narrowness of the Hellenic trench (Bonneau 1984). In model A, the subducting plate is pushed at 6 cm/year, approximating the fast convergence of the Pacific plate relative to Kamchatka (Peyton et al. 2001). In model B, this is reduced to 2 cm/year to account for the slower South American plate. The overriding plate in the case of model C is also fixed to its opposite wall since the overriding plate in the Hellenic system is the North African plate (Wortel and Spakman 2000).

The models' outputs provide clues to the dynamics of their respective plates. In the Kamchatka case (Fig. 10), as a result of the fast Pacific plate, the model shows initial compression of the overriding plate and the absence of whole-scale trench rollback, reflecting the lack of a back-arc in the Kuril-Kamchatka region. The overriding plate undergoes a high level of strain and uplift until trench rollback initiates. Additionally the slab folds over itself a number of times, as the trench position stays relatively constant relative to the underlying mantle. This folding leads to periods of relative extension and compression on the overriding plate, although the extension is not enough to lead to the formation of a back-arc or spreading centre.

In the Caribbean model (Fig. 11), there is still an initial period of compression, but this is followed by rollback as the subducting slab folds over on itself. The stress-guide through the slab slows the encroachment, even though the slab is pushed, albeit at a slow rate. As the trench rolls back, the initial condition is nearly met again (although the model does not run for long enough) and presumably

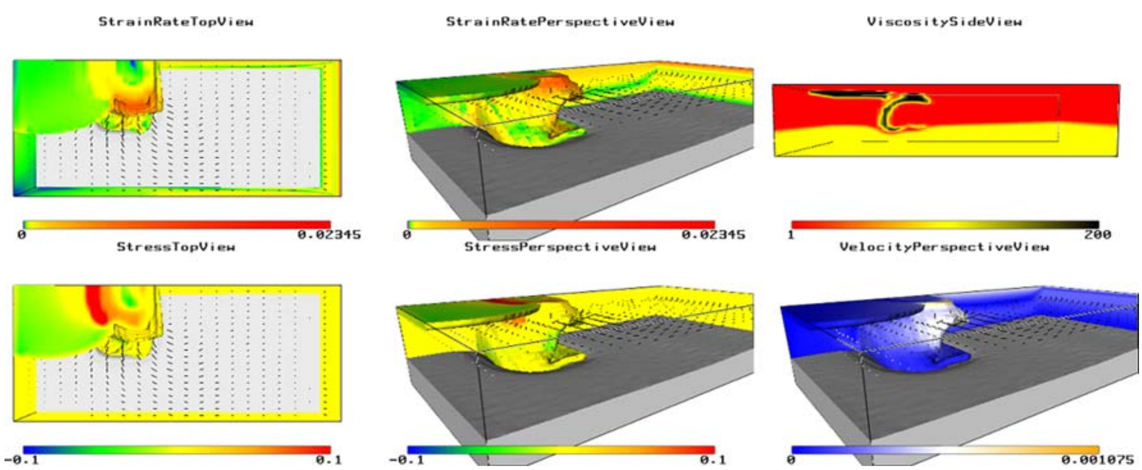


Fig. 10 Model A (Kuril-Kamchatka): strain rate, stress, viscosity and velocity (amplitude) and velocity (vectors) Trench rollback initiates at the edge of the model while the centre is still under compression

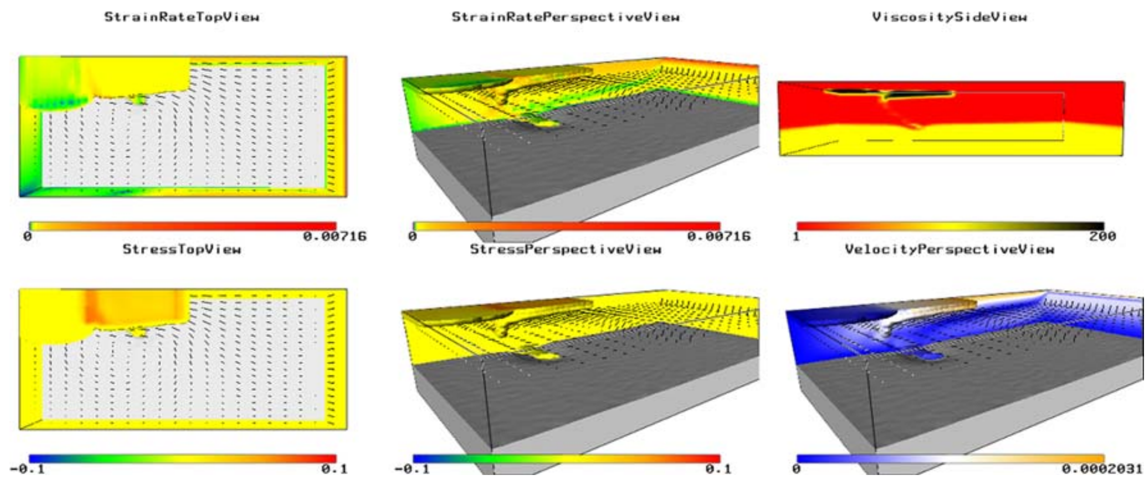


Fig. 11 Model B (Caribbean): strain rate, stress, viscosity and velocity (amplitude) and velocity (vectors) shown for the Caribbean. The overriding plate undergoes a high level of strain and uplift (shown on previous image) until trench rollback initiates

trench encroachment may resume or at least back-arc spreading may stop. This may explain, for example, the Tonga trench, where back-arc spreading initiates at around 34 Ma in the South Fiji Basin, ceasing at 25 Ma, followed by extension of the Norfolk Basin, which may have ceased around 20 Ma, only to resume in the Lau basin at around 7 Ma (Sdrolias and Müller 2006).

The Hellenic system does not experience ''ridge-push'', and this causes trench-rollback to be initiated nearly immediately (Fig. 12). The overriding plate is pulled by the subducting slab as it rolls back causing broad-scale extension on the overriding plate. The subducting slab is laid out too flat to fold over on itself and thus rollback is consistent throughout the period of the model. Thus in more constrained systems, such as those in which the Hellenic trench is surrounded by continental blocks, subduction falls into a pattern of fast rollback. This pattern matches the fast rollback relative to convergence for the Aegean back-arc basin, where the convergence is only 0.5 to 1 cm/year, but the rollback is up to 3 cm/year (Wortel and Spakman 2000).

The simulation results shown in Figs. 10, 11 and 12 were obtained by opening the corresponding XML input files in the UnderworldGUI and submitting a job request (Fig. 13). Once the simulation process was complete and the output data had been visualized, the results were

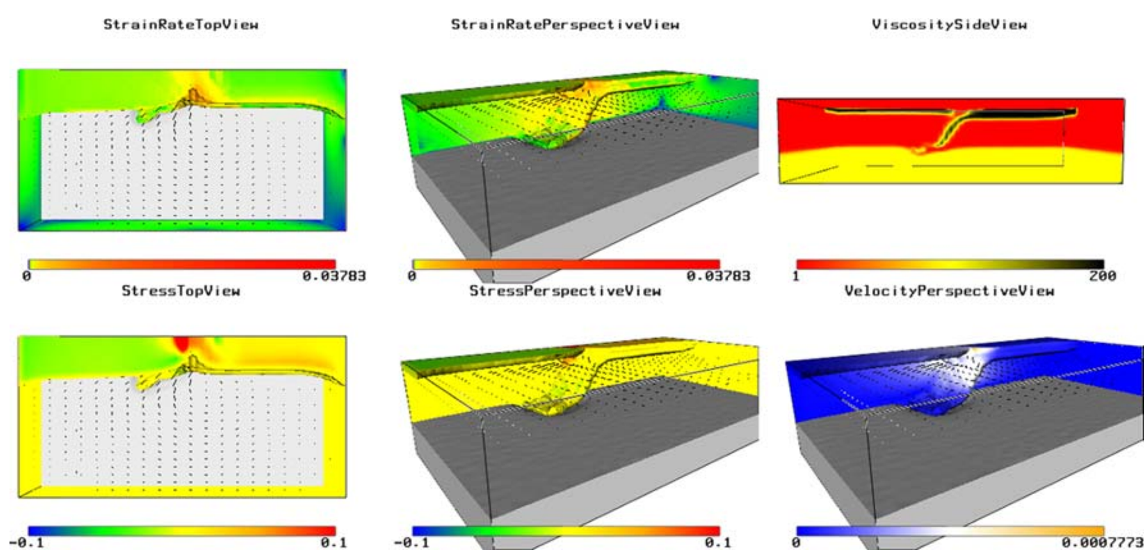


Fig. 12 Model C (Hellenic): strain rate, stress, viscosity and velocity (amplitude) and velocity (vectors) shown for the Hellenic trench model at approximately 30 Ma from initiation. The overriding plate (Eurasian) is under rapid extension as the trench rolls back

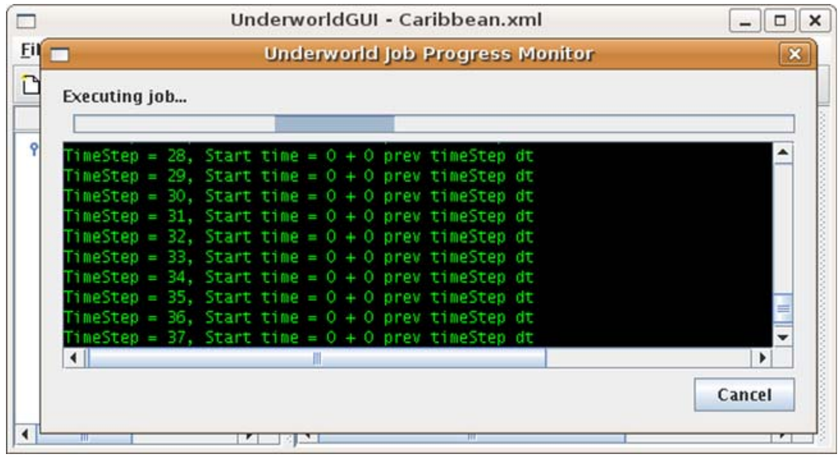


Fig. 13 Dialog for monitoring the status of a job submission in progress

exported to the library, including the original input file for the model, and a number of image and video files to assist the user in identifying the model later (Fig. 14). The model could then be imported from the library by selecting it from the list of models in the Underworld Library Importer dialog (Fig. 15). By scrolling through the images associated with this model, the user could gain an overview of its important features.

Status and future directions

In the current phase of development, establishing a connection between the UnderworldGUI and the Underworld software stack has been given the highest priority. Ultimately, two levels of communication are envisioned

1. A LOCAL connection, where the Underworld software is running on the same machine as the GUI; and
2. A REMOTE connection, where the Underworld software is running on a remote machine, and the GUI performs job control and file transfers via existing protocols, for example, Secure Shell (SSH) for submitting job requests, and Secure Copy Protocol (SCP) or Secure FTP (SFTP) for uploading the input file(s) and downloading the simulation output.

The immediate focus of development is on the first point above. The GUI must be able to fully process a job request on the local machine, executing the appropriate commands and collecting the output files upon completion, before development moves on to the additional complexities of remote file transfer. Hence, the second point is a medium- to

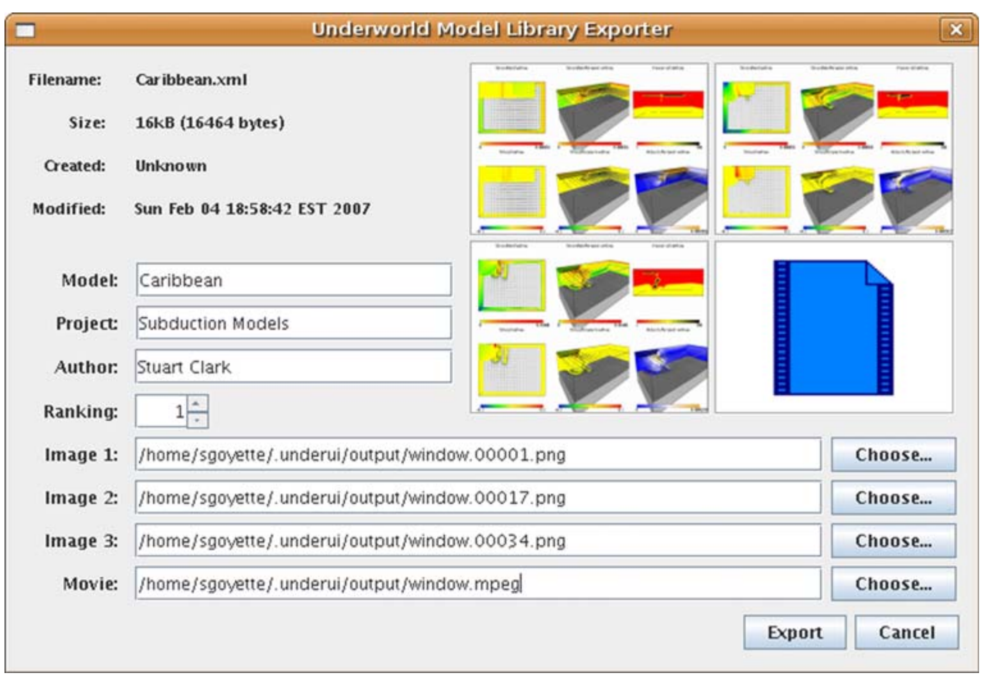


Fig. 14 Exporting model B (Caribbean) to the Underworld model library

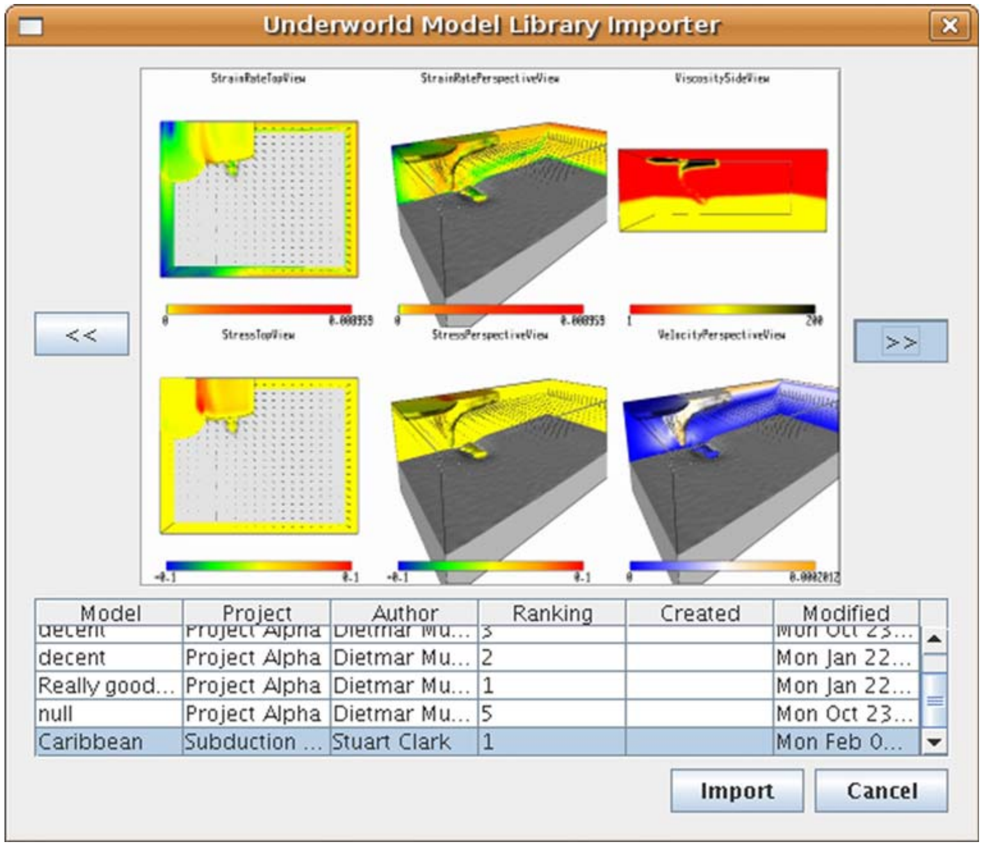


Fig. 15 Importing model B (Caribbean) from the Underworld model library. The second of three preview images is shown

long-term development goal. All that remains to fulfil the first goal is to automate the retrieval and display of the simulation results. Furthermore, as the UnderworldGUI is being developed as a platform-independent Java application, it is well suited to an applet or Web Start application deployment. Establishing this kind of web portal could potentially simplify access to, for instance, a particular deployment of Underworld on a cluster, since the user would not be required to access the cluster to submit the job request themselves. This would broaden the UnderworldGUI's potential user base, and will be an area for future development.

There are also a number of enhancements and fixes to be carried out on the GUI's design and layout.[4] A palette sidebar has been incorporated into the design, which, once fully implemented, will allow the user to drag-and-drop components and parameters into the model via the currently open editor. More sophisticated text editing tools, such as a search function, are planned for the Plain Text View editor. The final implementation of the View port Editor will include the tools for assigning customised visual representations to variable data values in the simulation. Furthermore, all of these enhancements and fixes will be guided by the requirements and preferences of the user community.

Initially, the UnderworldGUI was envisioned as an interface for creating and modifying Underworld input parameters, similar to the EllipsisGUI software (Dyksterhuis et al. 2007). The current vision of the project now includes a model library for storing and sharing previously run models, and a view port editor for managing the visualisation parameters in a more user-friendly way. Furthermore, the UnderworldGUI is designed to communicate with the Underworld software, thus allowing job control in addition to job input manipulation.

Acknowledgments This project was supported by the Australian Partnership for Advanced Computing (APAC). Underworld example models were run on APAC resources. Our humble thanks go to J. D. Clemens and D. Yuen for their editorial handling, and to the anonymous reviewers for improving this paper with their comments.

References

Bonneau M (1984) Correlation of the Hellenic nappes in the South-East Aegean and their tectonic reconstructions. In: Dixon JE, Robertson AHF (eds) Geological evolution of the eastern Mediterranean. Geol Soc London Spec Publ 17:517–527

Boschetti F, Moresi L (2001) Interactive inversion in geosciences. Geophysics 66(4):1226–1234

[4] See the UnderworldGUI wiki at http://www.wiki.vislab.usyd.edu.au/moinwiki/UnderworldGUI, last accessed 25 January 2007.

Dyksterhuis S, Müller RD, Rey P, Moresi L (2007) A graphical user interface for particle-in-cell finite element analysis of lithospheric deformation and mantle convection in two dimensions. Comp Geosci, in press

Giunta G, Beccaluva L, Siena F (2006) Caribbean Plate margin evolution: constraints and current problems. Geol Acta 4(1–2):265–277

Moresi L, Dufour F, Mühlhaus H–B (2003) A Lagrangian integration point finite element method for large deformation modelling of viscoelastic geomaterials. J Comp Phys 184:476–497

O'Neill C, Moresi L, Müller D, Albert R, Dufour F (2006) Ellipsis 3D: a particle-in-cell finite-element hybrid code for modelling mantle convection and lithospheric deformation. Comp Geosci 32:1769–1779

Peyton V, Levin V, Park J, Brandon M, Lees J, Gordeev E, Ozerov A (2001) Mantle flow at a slab edge: seismic anisotropy in the Kamchatka region. Geophys Res Lett 28(2):379–382

Sdrolias M, Müller RD (2006) Controls on back-arc basin formation. Geochem Geophys Geosyst 7(4):Q04016 doi:10.1029/2005GC001090

Stegman DR, Moresi L, Turnbull R, Lo A, Quenette S (2005) gLucifer: Next-generation visualization framework for high-perfomance computational geodynamic models. Eos Trans AGU Fall Meet Suppl.:Abstract IN42A-03

Stegman DR, Freeman J, Schellart WP, Moresi L, May D (2006) Influence of trench width on subduction hinge retreat rates in 3D models of slab rollback. Geochem Geophys Geosyst 7(3):Q03012 doi:10.1029/2005GC001056

Wortel MJR, Spakman W (2000) Subduction and slab detachment in the Mediterranean–Carpathian region. Science 290(50):1910–1917