Tuning GENIE Earth System Model Components using a Grid Enabled Data Management System

A. R. Price[1], G. Xue[1], A. Yool[2], D. J. Lunt[3], T. M. Lenton[4], J. L. Wason[1],

G. E. Pound[1], S. J. Cox[1] and the GENIE team

[1]School of Engineering Sciences, University of Southampton, Southampton, UK
[2]Southampton Oceanography Centre, Southampton, UK
[3]School of Geographical Sciences, University of Bristol, Bristol, UK
[4]School of Environmental Sciences, University of East Anglia, Norwich, UK

`http://www.genie.ac.uk/`

Abstract

We present the Grid enabled data management system that has been deployed for the GENIE project and demonstrate its use in tuning studies of an Earth system model. A Matlab client to the system provides a common environment for the project Virtual Organization to share scripts, binaries and output data. By using tools available in the Geodise toolkits we have scripted the execution of tuning studies which exploit multiple heterogeneous computational resources and use the database repository to steer computation using multi-dimensional optimisation methods.

1   Introduction

The GENIE project (Grid ENabled Integrated Earth system model [1]) is creating a Grid enabled component framework for the composition, execution and management of Earth system models. The GENIE code base consists of mature models of Earth system components (ocean, atmosphere, land surface, sea-ice, ice-sheets, biogeochemistry, etc.) which can be flexibly coupled together and run over multi-millennial timescales, primarily for glacial-interglacial simulations. An important part of such simulations is the parameterisation of many of the physical processes of the Earth System that occur on relatively small timescales. In order to make meaningful predictions it is vital that these parameters are tuned to appropriate values and that the effects of uncertainties in these parameters are quantified.

There are many methods that may be adopted for the general problem of optimising a parameterised model over a multi-dimensional state space. Choosing an appropriate methodology depends upon many factors including the nature of the problem, the size of the state space and the cost involved in evaluating data points. The application of optimisation methods to new models often requires additional code development to implement a suitable algorithm, integrate with an optimisation package or link with numerical library routines. E.g. The Climate*Prediction*.net project has developed an entire distributed client application in order to perform an exhaustive study of the state space of the Hadley climate model. In this paper we present the design of the data management system we have deployed for the GENIE project and demonstrate its use in a tuning study of an example GENIE implementation (the c-GOLDSTEIN climate model [2]). This system provides an interface to the computational Grid, integration with a sophisticated optimisation and design package OPTIONS [3] and access to our file and metadata repository. We show how the Grid enabled tools provided by the Geodise project [4] enable bespoke tuning studies to be quickly configured and executed and how the data management system provides a resource that can be exploited for computational steering of the optimisation study. This provides the environmental scientist with a common toolset with which to investigate and tune their models.

In section 2 we present the data management system, the tools available in the Geodise toolkits and the OPTIONS design package. We then discuss how bespoke tuning studies can be scripted in the Matlab environment in section 3. The results from a number of multi-dimensional optimisation studies are presented in section 4. We discuss the data management system and future intentions in section 5.

2   Data Management System

We have adopted an augmented version of the Geodise Database Toolkit to provide a generic data management solution for the GENIE project. The Geodise system exploits database technology to enable metadata to be associated with any file submitted to the repository for archiving (Figure 1). The database interface is exposed as Web services and files are archived in the system through a two step process: a) the file is transferred to a user specified file server using the GridFTP protocol [5] and b) information is recorded in the database about the file including its location, its unique system generated identifier, access rights and any user-defined metadata. Client tools are provided in Matlab [6] and Jython [7] to allow the user to upload, query and retrieve data in the repository. The XML Toolbox [8] is used to convert Matlab data structures into XML for communication with the Web service interface to the database. Access to the system is controlled by authenticating the user through their X.509 certificate. The system therefore provides an open and transparent facility through which members of the project Virtual Organisation can share data.
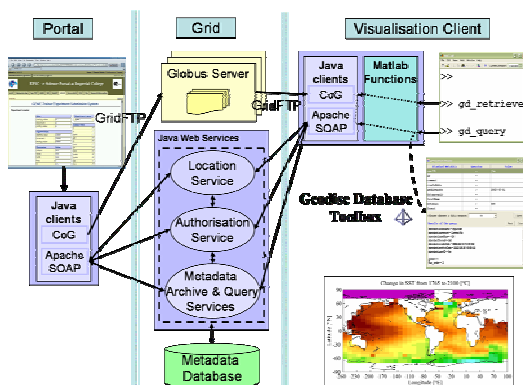
Figure 1: Architectural design of the GENIE data management system.

The Geodise system has been designed to provide a flexible management solution for the engineering design process and must be able to handle any user-defined metadata that the engineer wishes to record. However, the data generated by the GENIE framework is produced by well-defined component codes and the metadata is thus more tightly constrained. This enables us to significantly improve the efficiency of the system by mapping an XML Schema to the underlying Oracle9i database. The metadata is therefore handled by the database using relational tables but we maintain the flexibility of the system by managing the XML Schema.

The data management system thus provides a resource for storing metadata, files and data structures. The system is flexible enough that the developing needs of the framework can be supported while maintaining the efficiency of the database.

The Geodise Computational toolkit [9] provides an interface to the Grid through functions written in the Matlab scripting environment which invoke classes in the Java CoG 1.1 [10]. These functions allow a user to submit compute jobs to the Grid, transfer files using GridFTP and monitor jobs and resources. In addition to the database toolset we therefore have a set of functions that enable powerful use of the computational grid. A subset of the Matlab functions is described in Table 1.

In addition to the tools described above the system also interfaces to OPTIONS, a design exploration and optimisation package that has been developed by the Evolutionary Optimization Research Group at the University of Southampton. This software provides a suite of sophisticated multi-dimensional optimisation algorithms developed primarily for engineering design optimisation. The package has been made available to the Matlab environment by the Geodise project via the OptionsMatlab interface.

| Command	Description
gd_putfile	Transfer a file to a specified Grid system using GridFTP.
gd_getfile	Retrieve a file from a specified Grid system using GridFTP.
gd_jobsubmit	Submit a Globus RSL job specification string to a specified job manager.
gd_jobstatus	Returns the status of a Globus GRAM job.
gd_archive	Archive a file or data structure to the GENIE database.
gd_query	Query the database for data matching specified criteria.
gd_retrieve	Retrieve files from the data management system.

Table 1: Description of the subset of Geodise functions used in tuning studies of GENIE codes. See [9] for further details.

3 Scripting Bespoke Tuning Studies

We have exploited the tools described above in conjunction with our database system to perform bespoke tuning studies of a GENIE simulation code:

1) The GENIE code is statically compiled on the target platform and the resulting binary is archived to the database.
2) The Matlab scripting environment is used to orchestrate the submission and execution of the GENIE binary with an accompanying set of parameters. The user is free to tailor the experiment to their local or national Grid-enabled computational resources (e.g. Globus, Condor, batch processing).
3) The output data from each component run is uploaded to the database repository.
4) The data is post-processed to obtain a "skill score", or "objective function", that gives a measure of the realism of each simulation.
5) The input parameter set is tuned using numerical tools available in Matlab, such as fminsearch, invoking external optimisation tools (e.g. OPTIONS [3]) that have been exposed as Grid/Web services or by applying a user defined algorithm such as the Ensemble Kalman filter [11]. The tool seeks to minimise or maximise the "objective function" by exploring the provided parameter space.
6) The results are available to the user from the database and can be viewed at any point in the optimisation.

We have performed a number of studies using such a design to optimise the c-GOLDSTEIN climate model, a composite of three initial GENIE components. The binary executable has been wrapped in the Matlab scripting language and presented as a function which accepts as input the variables to be tuned. The function returns, after simulation, the model's RMS error value determined by comparing the model climate and ocean state for that parameter set to present day observations. Having written such a function wrapper to the component code it is trivial to exploit the optimisation tools available in Matlab. Figure 2 illustrates the optimisation process in Matlab using the *fminsearch* method with a wrapped executable.
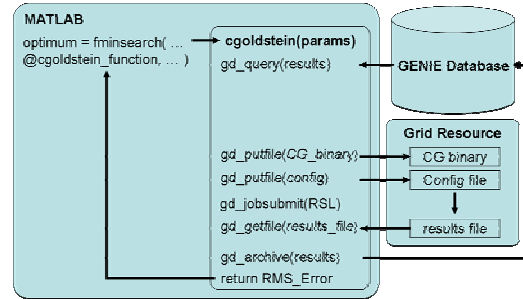
Figure 2: Schematic of the Matlab tuning process.

Upon each invocation, the *cgoldstein* function performs a query on the database to establish whether the data point specified by the parameters has been analysed in a previous experiment. If the result is available from the database then the RMS error value is simply retrieved from the database and returned by the function. This saves time by eliminating unnecessary duplication of effort. If the result is not available then the function proceeds by creating a parameter file, retrieving the binary executable from the database and transferring both to a grid-enabled resource using GridFTP. A Globus RSL string is created and submitted to the resource job manager (e.g. Fork, PBS) on the target platform. The component code is executed and a job handle is monitored from Matlab within the function call. Once the code has completed, the function uploads the parameters and results as a data structure to the database, archives the output data files and returns the RMS error function value of the component run.

To perform an optimisation of the wrapped binary the Matlab optimisation tools can be passed the function handle, a starting point in the parameter space, and limits or constraints on valid parameter values. In Figure 2, the *fminsearch* method minimises the function by invoking it with parameters determined by the algorithm being employed. At any point the optimisation can be monitored by retrieving the function evaluations from the database.

An important aspect of working with the emerging Grid software is dealing with occasional failures that inevitably occur. The tuning studies we describe in this paper submit hundreds of Grid jobs to a nationally distributed set of resources, and have had to be designed to cope with these occasional failures. The Matlab and Jython scripting environments provide native support for sophisticated exception handling. We therefore ensure that every interaction with the Grid is wrapped in a try-catch block and that any failures are handled

Label	Min	Init	Max	Truth	Final	Meaning	Units
SCLTAU	1.3	1.4	2.1	1.7	1.702	Wind-scale	-
INVDRAG	2.0	2.1	4.8	3.4	3.7	Inverse drag	Days
OCNHORZDIFF	2500	2600	5700	4100	4511	Ocean horizontal diffusion	m^2s^{-1}
OCNVERTDIFF	0.5261	0.6	6.325	1.824	1.931	Ocean vertical diffusion	$10^{-5}\ m^2s^{-1}$
ATMDIFFAMPT	2.8	2.9	4.8	3.8	3.545	T diffusion amplitude	$10^6\ m^2s^{-1}$
ATMDIFFAMPQ	1.3	1.4	1.9	1.6	1.686	Q diffusion amplitude	$10^6\ m^2s^{-1}$
TDIFFWIDTH	0.9	1.0	1.7	1.3	1.438	T diffusion width	radians
TADVCTCOEFF	0.0	0.01	0.12	0.06	0.065	T advection coefficient	-
QADVCTCOEFF	0.06	0.08	0.22	0.14	0.151	Q advection coefficient	-
SEAICEDIFF	3200	3400	9200	6200	7157	Sea-ice diffusion	m^2s^{-1}
SCALECO2	0.95	0.96	1.05	1.0	1.011	CO_2 concentration factor	(350 ppm)
SCALEFWF	0.8	0.81	1.2	1.0	1.053	Fresh Water Flux factor	(0.29 Sv)

Table 2: Parameters varied in the c-GOLDSTEIN climate model. The Min and Max columns define the valid ranges of the parameter values. The Truth and Final columns are discussed in the text. See [2] for further details.

such that the optimisation may continue without being adversely affected. For example, if the function evaluation cannot be completed then the script will return a NULL RMS error value that is ignored by the optimisation algorithm.

Allowing the user to wrap their binaries in this way provides a flexible means for the project to study the new models being developed for the framework. Users are free to adapt the optimisation scripts to study a component of their choice, execute on the local or national resources available to them and share results from other experiments through the database.

We have studied the c-GOLDSTEIN model using a number of optimisation methods including methods from the Matlab optimisation toolbox and the OptionsMatlab suite of tools. In the next section we present the results of some simple optimisations of the model in one- and two-dimensions. We then present a twin-study to demonstrate the use of a Genetic Algorithm [12] to attempt to recover an optimal set of model parameters in twelve dimensions using OptionsMatlab. The results of applying the same method to tune the model towards observational data are then also presented.

4   Results

The c-GOLDSTEIN Earth System Model (ESM) is a composite of three of the initial GENIE component codes and consists of a three-dimensional frictional geostrophic ocean model coupled to a sea-ice model and a two-dimensional energy-moisture balance atmosphere model. The resulting ESM is computationally efficient with a 4000 year integration being possible in ~2 hours on a standard 1GHz PIII desktop system. This is

sufficient time for the slowest component of the model to reach equilibrium.

The model has twelve tuneable parameters which affect various properties of the ocean, atmosphere and sea-ice. The parameters that we study are detailed in [2] and are summarised in Table 2. The objective function for the optimisation problem is calculated as the discrepancy between model and observational fields evaluated by calculating an RMS error for the model state variables (ocean temperature and salinity, atmospheric temperature and specific humidity). The observational data is taken from the National Centers for Environmental Prediction [13]. The 3D ocean error fields and 2D atmosphere error fields are weighted and averaged together to determine a single measure of model-observation mismatch.

4.1   1D and 2D optimisation

Initial experiments using the grid tools applied the Matlab function *fminsearch* to find local minima in a small subset of the parameter space for the c-GOLDSTEIN code.

Figure 3 presents the results of one- and two-dimensional minimisations that examine the atmospheric parameterisation of the model. The first experiment varies a scaling factor that modifies the strength of freshwater anomalies applied to the model to transport moisture between the Atlantic and Pacific basins. The result suggests that, given standard values of other parameters, the default freshwater correction (1.0) is slightly stronger than that required to reproduce the current climate. The discontinuity in the function is caused by a phase change in the model where the characteristics of the ocean's thermohaline circulation (THC) alter dramatically. In the most extreme (top left) point, the THC

"collapses", radically altering the distribution of heat and salt in the ocean, and causing the extreme RMS error.
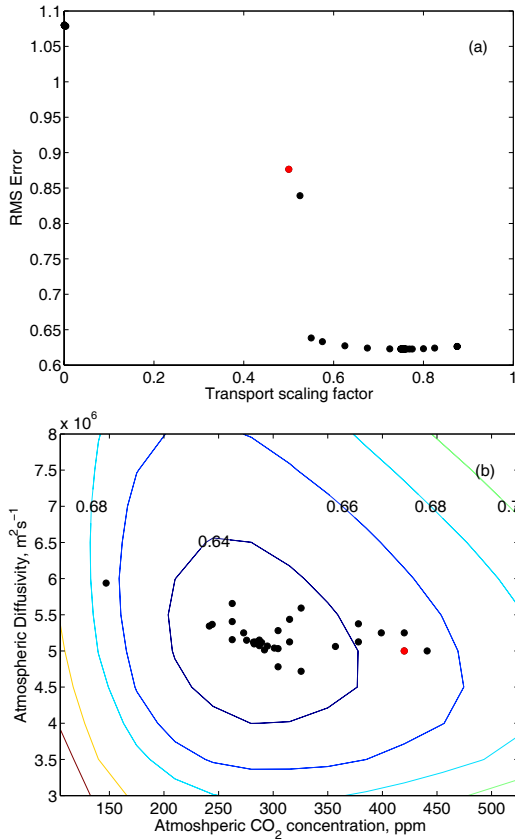
Figure 3: a) The atmospheric transport scaling factor is tuned by minimising the RMS error and **b)** the RMS error function is (contour) mapped as a function of atmospheric diffusivity and the CO_2 concentration. A minimisation (dots) is performed using the Matlab fminsearch function. Red dots indicate the starting points of the searches.

In the second experiment, two parameters are co-varied: atmospheric CO_2 concentration and atmospheric heat diffusivity. The results here suggest that the default heat diffusivity of $5 \times 10^6 \ m^2 s^{-1}$ is close to optimal but that the tuned model prefers a pre-industrial value of atmospheric CO_2 (~280 ppm). This is consistent with the fact that the present day climate has yet to respond significantly to increased levels of CO_2 (the model default for the present day is 350 ppm). While these results only probe a small subset of the model's parameter space, we have demonstrated that such a study can be easily configured, executed and managed from within Matlab.

4.2   c-GOLDSTEIN optimisation

We have explored twelve of the tuneable parameters of the c-GOLDSTEIN model using the OPTIONS package. As an initial test of the Grid enabled OptionsMatlab scripts we set up a twin study following the methodology applied by Annan *et al.* [14] when validating their Ensemble Kalman Filter using the same model. For this validation we attempt to tune the parameters to a known state of the model using a Genetic Algorithm followed by a local search around the best candidate point.

For the 'truth' state of the model we have run the simulation for 2000 years with a specific set of parameters. The resulting state fields have been processed to provide a data set from which the RMS error statistic is subsequently derived. For the optimum 'truth' parameters the RMS error therefore evaluates to exactly zero.

In a similar process to that described for the *fminsearch* optimisation the OptionsMatlab library is invoked so that it performs an optimisation over the function wrapper of the model binary. The Matlab interface to the OPTIONS suite of optimisation tools allows a user to apply an algorithm of choice with minimal reworking of the underlying scripts.

The Genetic Algorithm uses a population size of 100 members and is performed over 10 iterations of the algorithm. The 100 function evaluations are performed concurrently by the OPTIONS package. The Matlab function wrapper to the c-GOLDSTEIN binary has been configured to use both local and national Grid resources. These include a local IBM ©serverBladeCentre consisting of 12 dual node Intel Xeon blades and the four clusters of the National Grid Service [15] hosted at Oxford, Leeds, RAL and Manchester. The function randomly selects a compute resource on which to run the binary so that the load is distributed evenly over the available systems. For these studies we have allocated jobs to these resources in the ratio 10:16:8:8:36 to reflect the different configurations of the 'short' job queues on these systems. This is one of the ways in which the scripting environment allows us to flexibly adjust the study. With the overhead of file transfers we typically complete a generation of the Genetic Algorithm in approximately 90 minutes.
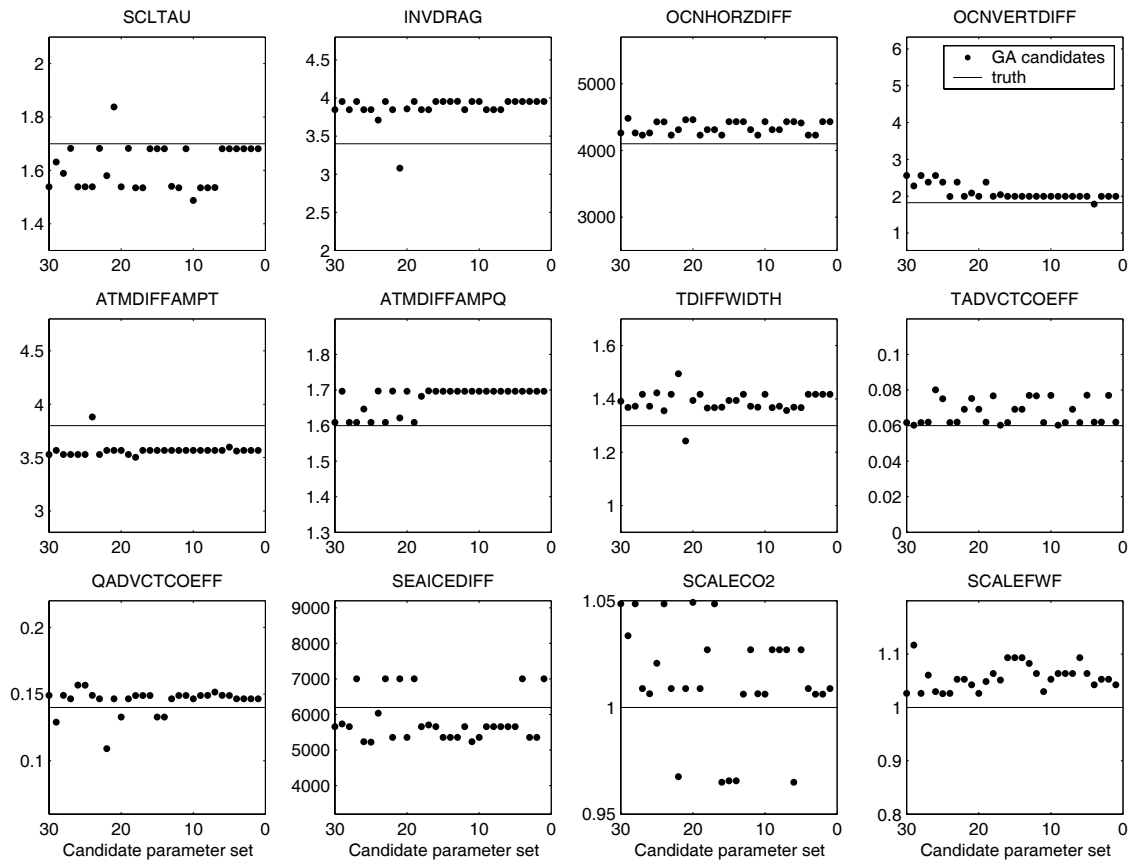
Figure 4: Parameter values for the 30 best points found by the Genetic Algorithm. The true optimal values for the function are indicated by the truth lines.

The parameter values for the 30 best candidate points returned by the Genetic Algorithm are plotted in Figure 4 along with the optimal values for the 'truth' data. While the method appears to have recovered the correct values for some of the parameters it has clearly not found the global optimum. This is not unexpected since time constraints have meant that we have not really provided the Genetic Algorithm with a sufficient population size to find the global minimum in a twelve parameter state space.

However, having found a candidate point we perform a local search of the parameter space using a Nelder-Mead Simplex algorithm. This is easily achieved by simply passing the candidate point back to OptionsMatlab and configuring it to run the local search. In Figure 5 we plot the trace history of the RMS error value of both the Genetic Algorithm and the Simplex search. The first 885 function evaluations were generated by the Genetic Algorithm; in this experiment we 'lost' 115 evaluations due to Grid job failures. The remaining points are evaluations from the Simplex search. The final optimum parameters

are presented in Table 2 and can be compared with the 'truth' values.
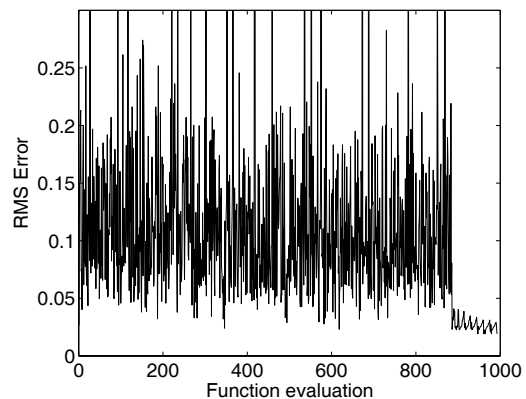
Figure 5: Optimisation trace history of the twin-test study.

During a study of this size it is inevitable that failures will occur in the execution of the model on the Grid. Job submission failure, GridFTP failure, numerical instability and network problems are all potential sources of error. However, as all valid function evaluations

are recorded in the database any subsequent re-analysis of the problem will benefit from these data points being readily available.

We have applied the tuning process again using the original observational data to generate the RMS error statistic.
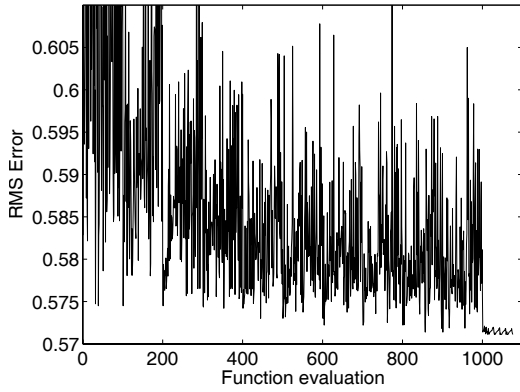

Figure 6: cGOLDSTEIN optimisation trace history. RMS Error statistic derived by comparing the model state with NCEP data.

The optimisation trace history is presented in Figure 6 for the Genetic Algorithm and the Simplex search around the best candidate point from the GA.
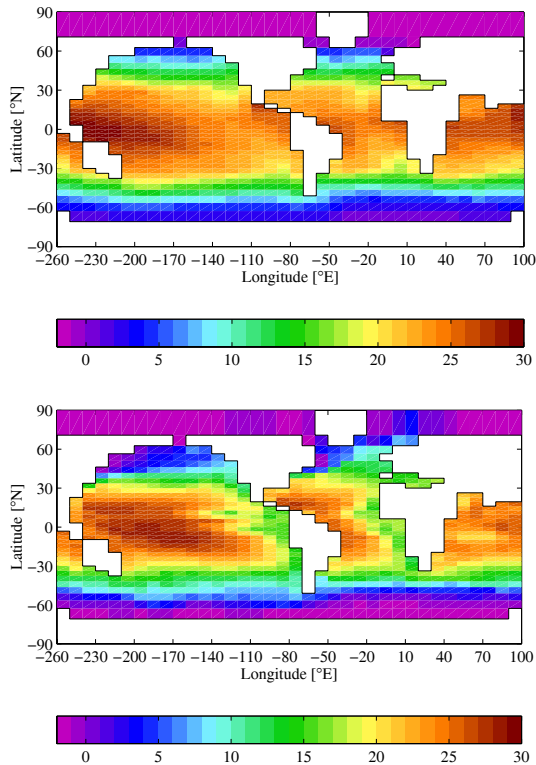
Figure 7: a) Sea surface temperatures of the model evaluated at the optimum parameters. **b)** NCEP observed sea-surface temperature data.

In Figure 7 we present the ocean temperature patterns generated by the model at the optimum parameter set and the equivalent observational data. The patterns generally compare well, although the model has a tendency to higher west Pacific temperatures because there is no Indonesian through-flow in the model (the channel connecting the Pacific and Indian Oceans). The temperatures around Antarctica are also higher than those of the dataset because of deficiencies in the land surface scheme in this area of the model (there is no ice on the land so the albedo in this region is therefore too low). Unfortunately, there are no tuning parameters in this study that can affect these deficiencies in the model.
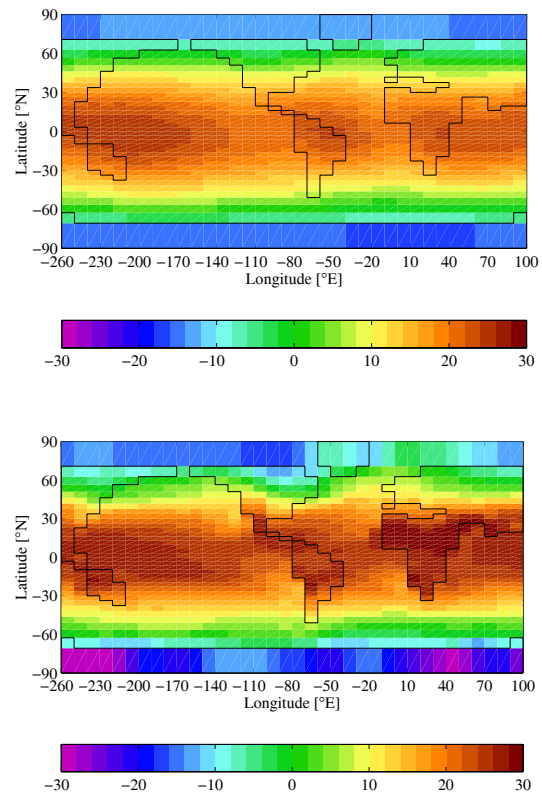
Figure 8: a) Air temperatures for the model evaluated at the optimum parameters. **b)** NCEP observational air temperatures.

In Figure 8 the air temperature patterns of the model are compared to the observational data. Again the data compare well, although the lack of a detailed land surface scheme means that land temperatures cannot reach the extremes of the data (i.e. the model is too warm at the poles, and too cold in the deserts). This is, again, a deficiency that the parameters of the model cannot fix. Future iterations of the GENIE model will include a proper land surface scheme to remedy these deficiencies.

5 Discussion and Future Work

We have presented the GENIE data management solution and demonstrated its use in the tuning of a simulation code from the GENIE framework. By using tools available in the Geodise toolkits and OptionsMatlab we have been able to script the execution of a tuning study which uses local and national Grid resources and exploits our database repository to steer the computation and facilitate the sharing of data in a transparent way.

The study of the c-GOLDSTEIN model has demonstrated the flexibility of the Matlab scripting approach. The model binary is wrapped as a function which is configured to select an available resource on which to execute the model. The optimisation tools can be quickly and easily configured to run a variety of optimisers or design search algorithms on the function. It is also possible to change the nature of the analysis by modifying the logic that determines the function's return value.

The framework therefore provides all of the tools required for tuning the GENIE models; a scripting environment, database repository, computational Grid interface and a suite of design optimisation algorithms. A global minimum can reliably be found in low dimensional problem space. For higher dimensional problems we have demonstrated that the tools are appropriate for locating local minima in the state space. We have applied a Genetic Algorithm to the c-GOLDSTEIN model but could easily apply other methods such as simulated annealing or an evolutionary program. To improve our understanding of the GENIE models we will provide an implementation of the Ensemble Kalman Filter. The EnKF is a highly efficient method that can execute $O(50)$ runs to tune the parameters of a model and has already been successfully applied to optimise the c-GOLDSTEIN model [14].

The occasional failures we encounter when using the Grid are currently handled by letting the function fall through and return a null value to the optimiser. In most cases the failures are intermittent and a subsequent attempt at the Grid interaction would prove successful. While the sophistication of the error handling could be improved to deal with this it would be detrimental to the ease of the scripting approach we wish to adopt. To address this issue we will liaise with system administrators to investigate improving the quality of service of the Grid resources. We will work with the Geodise project to investigate the implementation of additional logic in the toolset to improve robustness. A merit of our framework is that it also facilitates the integration of middleware such as ICENI currently being developed by members of the project at the London e-Science Centre [16]. This would provide alternative Grid job management and resource allocation.

6 References

[1] The GENIE project. http://www.genie.ac.uk

[2] Edwards, N. R. and Marsh, R., 2003. An efficient climate model with three-dimensional ocean dynamics. Clim. Dyn., submitted.

[3] Keane, A. J., OPTIONS: Design Exploration System, http://www.soton.ac.uk/~ajk/options.ps

[4] The GEODISE project. http://www.geodise.org

[5] The GridFTP Protocol and Software. http://www.globus.org/datagrid/gridftp.html

[6] Matlab 6.5. http://www.mathworks.com

[7] Jython. http://www.jython.org

[8] The XML Toolbox. http://www.geodise.org/toolboxes/generic

[9] Eres, M. H., Pound, G.E., Jiao, Z., Wason, J. L., Xu, F., Keane, A. J. and Cox, S. J., 2004. Implementation and utilisation of a Grid-enabled problem solving environment in Matlab. Future Generation Computer Systems, In Press.

[10] The Java Commodity Grid Kit. http://www.globus.org/cog/java

[11] Evensen, G., 2003. The Ensemble Kalman Filter: theoretical formulation and practical implementation. Ocean Dynamics **53**, 343-367.

[12] Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin, 1992.

[13] The NCEP/NCAR reanalysis project. http://www.cdc.noaa.gov/cdc/reanalysis/

[14] J. D. Annan, J. C. Hargreaves, N. R. Edwards and R. Marsh, Parameter estimation in an intermediate complexity earth system model using an ensemble Kalman filter, Ocean Modelling, In Press, Corrected Proof, Available online 23 January 2004.

[15] The National Grid Service. http://www.ngs.ac.uk/

[16] M.Y.Gulamali, A.S. McGough, S.J. Newhouse, and J. Darlington, 2004. Using ICENI to run parameter sweep applications across multiple Grid resources. In *Global Grid Forum 10, Case Studies on Grid Applications Workshop*, Berlin, Germany.