CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2006; (in press)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1046

Optimization of integrated Earth System Model components using Grid-enabled data management and computation

A. R. Price[1,*,†], G. Xue[1], A. Yool[2], D. J. Lunt[3], P. J. Valdes[3], T. M. Lenton[4], J. L. Wason[1], G. E. Pound[1], S. J. Cox[1] and the GENIE team[5]

[1]*Southampton Regional e-Science Centre, University of Southampton, Southampton SO17 1BJ, U.K.*
[2]*National Oceanography Centre, University of Southampton, Southampton SO14 3ZH, U.K.*
[3]*School of Geographical Sciences, University of Bristol, University Road, Bristol BS8 1SS, U.K.*
[4]*School of Environmental Sciences, University of East Anglia, Norwich NR4 7TJ, U.K.*
[5]*http://www.genie.ac.uk/*

SUMMARY

In this paper, we present the Grid enabled data management system that has been deployed for the Grid ENabled Integrated Earth system model (GENIE) project. The database system is an augmented version of the Geodise Database Toolbox and provides a repository for scripts, binaries and output data in the GENIE framework. By exploiting the functionality available in the Geodise toolboxes we demonstrate how the database can be employed to tune parameters of coupled GENIE Earth System Model components to improve their match with observational data. A Matlab client provides a common environment for the project Virtual Organization and allows the scripting of bespoke tuning studies that can exploit multiple heterogeneous computational resources. We present the results of a number of tuning exercises performed on GENIE model components using multi-dimensional optimization methods. In particular, we find that it is possible to successfully tune models with up to 30 free parameters using Kriging and Genetic Algorithm methods. Copyright © 2006 John Wiley & Sons, Ltd.

Received 12 January 2005; Revised 1 June 2005; Accepted 24 August 2005

KEY WORDS: optimization; Grid; data management; Earth System Model; Kriging; Genetic Algorithm

*Correspondence to: A. R. Price, School of Engineering Sciences, University of Southampton, Southampton SO17 1BJ, U.K.
†E-mail: a.r.price@soton.ac.uk

Contract/grant sponsor: Natural Environment Research Council (U.K. e-Science Programme); contract/grant number: NER/T/S/2002/00217

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 (ESMs). The GENIE code base consists of mature models of Earth system components (ocean, atmosphere, land surface, sea-ice, ice-sheets, biogeochemistry, etc.) that can be flexibly coupled together and run over multi-millennial timescales, primarily for glacial–interglacial simulations. An important part of such simulations is the parameterization of many of the physical processes of the Earth system that occur on relatively short 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 optimizing a parameterized model over a multi-dimensional state space (see, e.g., [2] and references therein). Choosing an appropriate methodology depends on 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 optimization methods to new models often requires additional code development to implement a suitable algorithm, integrate with an optimization package or link with numerical library routines. For example, the Climate*Prediction*.net project [3] 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 tuning studies of two GENIE implementations (the c-GOLDSTEIN climate model [4] and the Intermediate complexity General Circulation Model (IGCM) atmosphere component [5]). This system provides an interface to the computational Grid, integration with a sophisticated optimization and design package OPTIONS [6] and access to our file and metadata repository. We show how the Grid enabled tools provided by the Geodise project [7] 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 optimization 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 toolboxes and the OPTIONS design package. We discuss how bespoke tuning studies can be scripted in the Matlab environment in Section 3. The results from a number of multi-dimensional optimization studies are presented in Section 4. We discuss the data management system and summarize in Section 5.

2. DATA MANAGEMENT SYSTEM

The implementation of large-scale data and compute Grids can be seen in NASA's Information Power Grid and the Department of Energy's Science Grids [8, Chapter 5]. The marriage of data and compute Grids is particularly exemplified by the numerous projects [9] set up to manage and analyze the large volumes of data that will be produced by the Large Hadron Collider from 2007. Exploiting Grid technology to analyze expensive data sets is a common paradigm. For example, in the social sciences the FINGRID project [10] illustrates how both quantitative and qualitative data from multiple sources can be processed, analyzed and fused using a Grid-based application. The experiments performed using the GENIE framework generate data that is of value to the community. The challenge for the project is to exploit compute and data Grids to feed and recycle data back into model development.

DOI: 10.1002/cpe

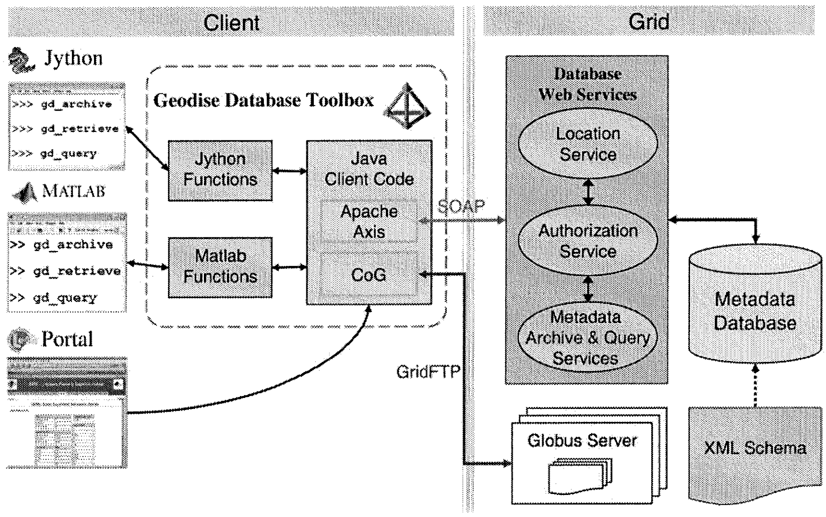

Figure 1. Architectural design of the GENIE data management system.

We have adopted an augmented version of the Geodise Database Toolbox 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 [11]; 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. Interactions with the system are transactional and client tools are provided in Matlab [12] and Jython [13] to allow the user to upload, query and retrieve data in the repository. The XML Toolbox [14] is used to convert Matlab data structures into XML for storage in the database and subsequent querying. Access to the system is controlled by authenticating the user through their X.509 certificate. The system therefore provides an open and transparent facility that members of the project Virtual Organization can use to share data.

The Geodise system has been designed to provide a flexible data management solution for the engineering design process and must be able to handle any user-defined metadata. 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 schema is exploited by the DBMS to create relational tables for the storage of XML documents. This allows the database to rewrite subsequent queries to an internal representation and derive an execution plan based on conventional relational algebra. The execution time of queries is reduced compared to raw searches of the XML data. We maintain the flexibility of the system by managing the XML schema mapped into the database. The data management system thus provides a resource for storing metadata, files and data structures.

DOI: 10.1002/cpe

Table I. Description of a subset of Geodise functions used in tuning studies of GENIE codes. Further details of Geodise functionality are available in [18].

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

The Geodise Computational Toolbox [15] provides an interface to the Grid through functions written in the Matlab scripting environment that invokes classes in the Java CoG 1.1 [16] and Condor [17]. 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 tools we therefore have a set of functions that enable powerful use of the Grid. A subset of the Matlab functions is described in Table I.

In addition to the tools described above, the system also interfaces to OPTIONS [6], a design exploration and optimization package that has been developed in the Computational Engineering and Design Centre at the University of Southampton. This software provides a suite of sophisticated multi-dimensional optimization algorithms developed primarily for engineering design optimization. The package has been made available to Matlab via the OptionsMatlab interface.

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) A unique datagroup [18] is created in the database with accompanying metadata to describe the experiment. All further interactions with the database use this logical identifier.
(2) The code is compiled on the target platform and the resulting binary archived to the database.
(3) 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 available Grid-enabled computational resources (e.g. Globus, Condor, batch processing, etc.).
(4) The output data from each component run is uploaded to the database repository.
(5) The data is post-processed to obtain a 'skill score', or 'objective function' that gives a measure of how accurately the simulation agrees with experimental observations.
(6) The input parameter set is tuned using numerical tools available in Matlab, such as *fminsearch*, invoking external optimization tools (e.g. OPTIONS) that have been exposed as Grid/Web services or applying a user-defined algorithm such as the Ensemble Kalman Filter (EnKF) [19].
(7) The results are available to the user from the database and can be viewed at any time.

DOI: 10.1002/cpe

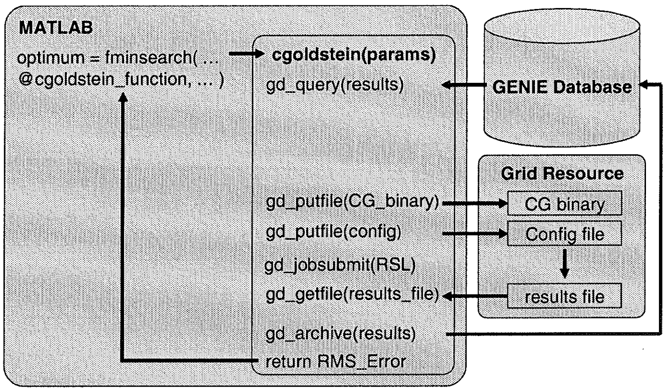

Figure 2. Schematic of the Matlab tuning process.

We have performed a number of studies using this methodology to optimize two new composite climate models from the GENIE framework. For each model a binary executable is created, wrapped in the Matlab scripting language and presented as a function that 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 optimization tools available in Matlab. Figure 2 illustrates the optimization process in Matlab using the *fminsearch* method with a wrapped implementation of the c-GOLDSTEIN model.

Upon each invocation, the *cgoldstein* function performs a query on the database to establish whether the data point specified by the parameters has been analyzed in a previous experiment. If the parameters match (to a specified accuracy) entries in the database then the RMS error value is simply retrieved and returned by the function. 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 Resource Specification Language (RSL) string is created and submitted to the resource job manager (e.g. fork, Portable Batch System) on the target platform. The component code is executed and a Globus 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 model run.

To perform an optimization of the wrapped binary the Matlab optimization tools can be passed a reference to the Matlab function, a starting point in the parameter space and limits or constraints on valid parameter values. In Figure 2, the *fminsearch* method minimizes the function by invoking it with parameters determined by the algorithm being employed. At any point the optimization can be monitored by retrieving the function evaluations from the database. 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 optimization scripts to study a component of their choice, execute on available Grid resources and share results from other experiments through the database.

We have studied two of the core GENIE ESM components; the c-GOLDSTEIN climate model and the IGCM atmosphere model. A number of optimization methods including methods from the Matlab optimization toolbox and algorithms in the OptionsMatlab suite of tools have been employed to tune and study these codes. In the next section we present the results of simple tuning in one and two dimensions for the c-GOLDSTEIN model. We then present two separate tuning methods for high-dimensional problems. The first employs a Kriging metamodel [20] to investigate 12 free parameters in the c-GOLDSTEIN model. The second uses a Genetic Algorithm (GA) [21] to study 30 free parameters of the IGCM atmosphere component. Both methods are well suited to exploit shared Grid-enabled resources.

4. RESULTS

The c-GOLDSTEIN ESM is a composite of three of the initial GENIE component codes and consists of a three-dimensional (3D) frictional geostrophic ocean model coupled to a sea-ice model and a two-dimensional (2D) energy–moisture balance atmosphere model. The resulting ESM is computationally efficient with a 4000-year integration being possible in approximately two hours on a standard 1 GHz P III desktop system. This is sufficient for the slowest component of the model to reach equilibrium.

The model has 12 tuneable parameters that affect various properties of the ocean, atmosphere and sea-ice. The parameters that we study are detailed in [4] and summarized in Table II. The objective function for the optimization problem is calculated as the discrepancy between model and observational fields evaluated by calculating a root-mean-squared (RMS) error for the model state variables (ocean temperature and salinity, atmospheric temperature and specific humidity). The observational data are taken from the National Centers for Environmental Prediction [22]. The 3D ocean error fields and 2D atmosphere error fields are weighted and averaged together to determine a single measure of model–observation mismatch. For the studies of c-GOLDSTEIN performed in this paper, each objective function evaluation is made after a 2000-year integration of the model.

4.1. 1D and 2D optimization

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-dimensional (1D) and 2D minimizations that examine the atmospheric parameterization of the model. The first experiment (Figure 1(a)) 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 of Figure 1(a)) point, the THC 'collapses', radically altering the distribution of heat and salt in the ocean and causing the extreme RMS error.

In the second experiment (Figure 1(b)), two parameters are co-varied: atmospheric CO_2 concentration and atmospheric heat diffusivity. The results here suggest that the default heat diffusivity

DOI: 10.1002/cpe

Table II. Parameters varied in the c-GOLDSTEIN climate model. The Min and Max columns define the valid ranges of the parameter values. The EnKF, Proximal-ACCPM and Krig columns show the optimal values obtained using the Ensemble Kalman Filter, Proximal Analytic Center Cutting Plane Method and the Kriging method, respectively. See [4] for further details.

Parameter	Min	Max	EnKF	Proximal-ACCPM	Krig	Units
Wind-scale	1.0	3.0	1.6674	1.1841	1.0026	—
Ocean horizontal diffusion	3.0×10^{2}	1.0×10^{4}	4.1264×10^{3}	5.5321×10^{3}	3.4336×10^{3}	$m^2\ s^{-1}$
Ocean vertical diffusion	2.0×10^{-6}	2.0×10^{-4}	1.8134×10^{-5}	3.8818×10^{-5}	4.5038×10^{-5}	$m^2\ s^{-1}$
Inverse drag	5.0×10^{-1}	5.0	3.4331	4.9959	2.7201	days
T diffusion amplitude	1.0×10^{6}	1.0×10^{7}	3.7548×10^{6}	2.5839×10^{6}	5.0535×10^{6}	$m^2\ s^{-1}$
Q diffusion amplitude	5.0×10^{4}	5.0×10^{6}	1.7447×10^{6}	1.9337×10^{6}	1.3088×10^{6}	$m^2\ s^{-1}$
T advection coefficient	0.0	1.0	6.0357×10^{-2}	8.9163×10^{-2}	2.4843×10^{-3}	—
Q advection coefficient	0.0	1.0	1.3674×10^{-1}	1.4885×10^{-2}	2.6769×10^{-2}	—
Sea-ice diffusion	5.0×10^{2}	8.0×10^{3}	6.2494×10^{3}	7.9913×10^{3}	2.1769×10^{3}	$m^2\ s^{-1}$
Freshwater flux factor	0.0	2.0	8.9796×10^{-1}	1.0406	9.5476×10^{-1}	(0.32Sv)
T diffusion width	5.0×10^{-1}	2.0	1.3071	1.9920	7.6329×10^{-1}	radians
T diffusion slope	0.0	2.5×10^{-1}	6.8597×10^{-2}	2.3644×10^{-1}	1.8427×10^{-1}	—
RMS error function	—	—	0.4986	0.4891	0.5145	—

DOI: 10.1002/cpe

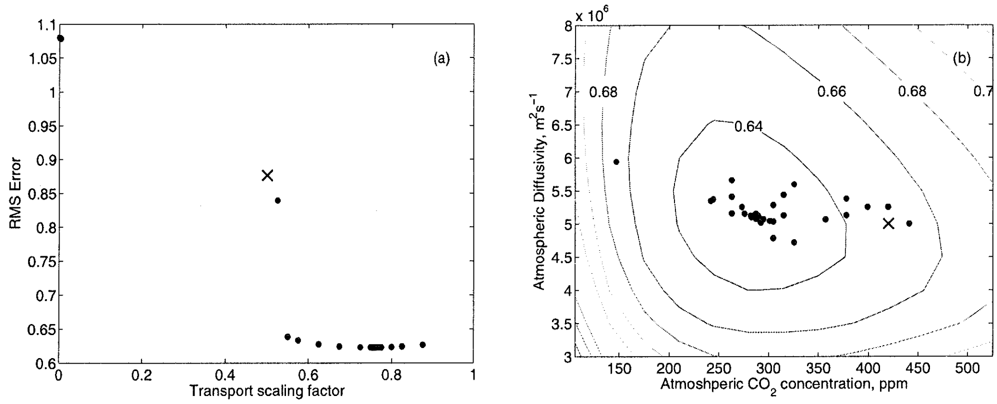

Figure 3. (a) The atmospheric transport scaling factor is tuned by minimizing the RMS error and (b) the RMS error function is (contour) mapped as a function of atmospheric diffusivity and the CO_2 concentration. A minimization (dots) is performed using the Matlab *fminsearch* function. The crosses indicate the starting points of the searches.

of 5×10^6 m^2 s^{-1} is close to optimal but that the tuned setting for the atmospheric CO_2 level is a pre-industrial value of ~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. Optimization of 12 free parameters in the c-GOLDSTEIN model

We now optimize all 12 of the tunable parameters of the c-GOLDSTEIN model in order to minimize the objective function described above. However, a direct search on the code using downhill methods is impractical with so many free parameters because the individual objective function evaluations take approximately two hours. Edwards and Marsh [4] first studied this problem using a Latin hypercube Monte Carlo method, performing a total of 2 000 000 integration years but failing to reduce the error below 0.6000. In recent studies of this problem, Annan *et al.* [23] applied an EnKF, an efficient data assimilation technique, to integrate an ensemble of 54 models over 10 000 years to tune the parameters. Beltran *et al.* [24] described the application of the proximal-ACCPM technique, a plane-cutting method that employed cheap calculations of local gradients in the state space and required just 47 000 years of integration to locate a minimum. The EnKF and proximal-ACCPM methods find optimal parameters with values for the objective function of 0.4986 and 0.4891, respectively. These values represent our best knowledge of the underlying function.

With our Grid-enabled function wrapper and the OPTIONS design exploration system we have a multitude of methods available to investigate the problem. The Grid provides an environment where concurrent objective function evaluations can be easily performed. We report the results of a Response Surface Modeling (RSM) approach that is commonly applied to this type of problem in the aircraft design discipline [20]. The RSM process involves sampling the expensive objective function over

DOI: 10.1002/cpe

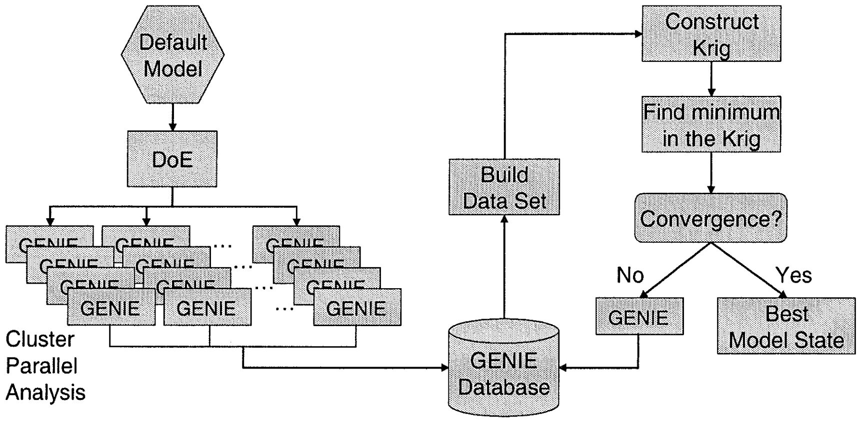

Figure 4. RSM procedure used to study GENIE models. Following a DOE sampling of the parameter space a Krig is performed to model the response surface. Iterative updates to the Krig improve the quality of the RSM until convergence is achieved at an optimum in the state space.

a limited number of points in the state space and then generating a metamodel of the response of the underlying function using curve-fitting techniques. The initial points are usually generated using formal Design of Experiment (DOE) methods. The modeled surface provides a representation of the objective function that can be searched very quickly. The surface is refined by evaluating the true objective function values at points of interest and feeding them back into the data set to generate a new surface. Once convergence is achieved, the surface can be used to locate minima in the objective function.

For these studies the Matlab function wrapper to the GENIE binary was configured to use both local and U.K. national Grid resources. These included a local IBM eServer BladeCentre consisting of 12 dual-node Intel Xeon blades and the four nodes of the U.K. National Grid Service [25]. The function randomly selected a compute resource on which to run the binary so that the load was distributed evenly over the available systems. For these studies we allocated jobs to these resources in the ratio 10:16:8:8:8 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, in the absence of any computational brokering infrastructure in the U.K.

Using the modified function wrapper we evaluated a DOE over the first 100 points of a LPτ sequence [26]. The Kriging then proceeded as in Figure 4 employing a Dynamic Hill Climbing (DHC) algorithm to search the modeled surface. At each iteration, the minimum found by the DHC was evaluated and the point fed back into the data set for the Krig. For the c-GOLDSTEIN study the algorithm converged after 23 function updates. A measure of the quality of the final Krig was obtained by performing an additional random sampling of 200 points in the parameter space and querying the RSM for its estimate at each. The correlation between the two gives a measure of the accuracy of the Krig and is plotted in Figure 5. The correlation coefficient for our data is 0.9052.

The optimum point found by the Kriging method is reported in Table II and has the value 0.5145, which is approximately 5% worse than the optimum found by the proximal-ACCPM method [24].

DOI: 10.1002/cpe

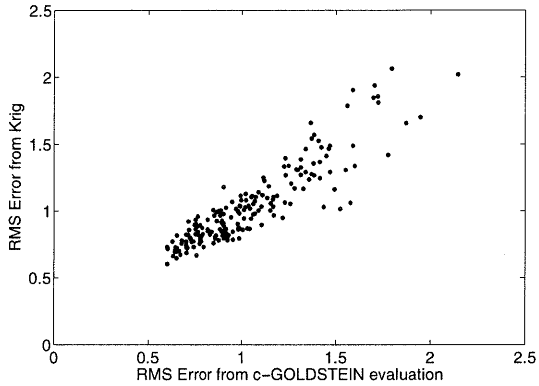

Figure 5. Correlation between 200 random c-GOLDSTEIN RMS error evaluations and those predicted by the Kriging trained on a separate set of 100 LPτ evaluations plus 23 updates.

This is consistent with reported performance for this type of study in the aircraft design discipline over direct search methods. Although the Krig has only located a local minimum in the state space the method makes no assumptions about the underlying function.

4.3. Optimization of 30 free parameters in the IGCM atmosphere component

In order to produce a stable coupling of components in the GENIE framework it is imperative that the fluxes between the models are compatible. Our second study investigated the 3D atmosphere component, coupling it to a 2D slab ocean and a sea-ice model. The atmosphere component is a realistic intermediate complexity atmospheric IGCM which is essentially that of de Forster *et al.* [5].

The IGCM has a large number of parameters, of which at least 30 are ill-constrained by observations or theory. To address the problem of tuning a model with so many degrees of freedom we have employed a GA to search the state space. Direct search methods (e.g. Simplex) are simply too expensive to perform over the large number of parameters. The Kriging method is also found to be too computationally demanding once more than about 10–20 variables are involved [20].

The objective function for the IGCM optimization problem is calculated as the discrepancy between the model state variables and equivalent observational data (net Solar radiation, net longwave radiation, latent heat and sensible heat fluxes plus wind stresses and precipitation rate). The observational data is again taken from the National Centers for Environmental Prediction [22]. For each time-averaged state variable, the RMS error at each gridpoint is calculated and averaged to obtain a global-mean error. The errors for each state variable are then combined to obtain a single measure of model–observation mismatch. Each objective function evaluation is made after a six-year integration of the model, taking the average of the final five years of output data (the model reaches equilibrium within one year).

The GA uses a population size of 200 members and is performed over 30 generations of the algorithm. The 200 function evaluations for each generation are called concurrently by the OPTIONS package. For this problem, the scripts were configured to make use of a large

DOI: 10.1002/cpe

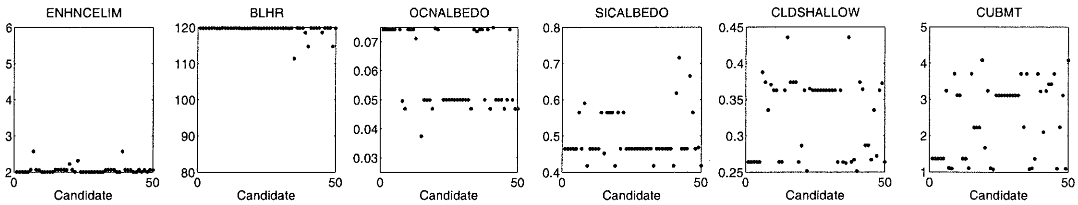

Figure 6. The 50 best candidate points found by the GA for a subset of the free parameters: boundary layer gust limit (ENHNCLIM); boundary layer relative humidity (BLHR); ocean albedo (OCNALBEDO); sea-ice albedo (SICALBEDO); shallow convection cloud fraction (CLDSHALLOW); and convection timescale in hours (CUBMT).

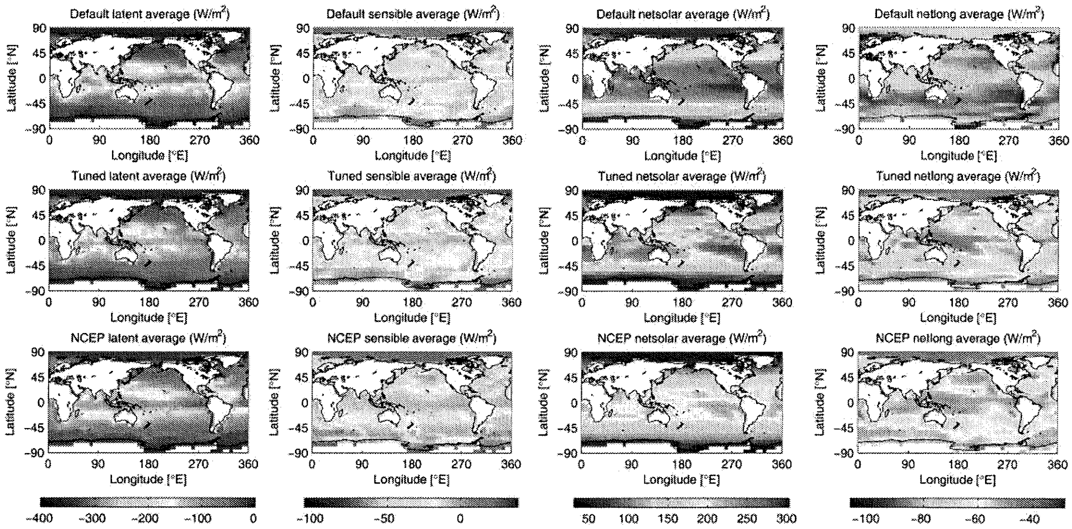

Figure 7. A GA has been employed to optimize the model from its default state (top row) to a tuned state (middle row) that is closer to the target observational data (bottom row). The columns correspond to the model fields for latent heat, sensible heat, net Solar and net longwave radiation fluxes respectively.

University Condor pool at Southampton. This facility provides more than 1800 compute nodes and enabled the completion of a generation of the GA in approximately three hours, even during periods of high activity on campus.

The GA search reduces the objective function by approximately 36% compared with its value at the default point in parameter space. A subset of the 50 best candidate points returned by the GA are plotted in Figure 6. The plots help indicate convex parameters where the preferred value lies at the upper or lower bounds of the variable (e.g. BLHR). We note that the limits on these variables, which in some cases were chosen fairly arbitrarily, could be re-assessed in light of these results. We can also identify variables that have little correlation with the objective function (e.g. CUBMT) suggesting that they have little influence in the local regions of the parameter space that these plots represent.

DOI: 10.1002/cpe

In Figure 7 we present the results of the optimization for four of the individual tuning targets. The tuned surface energy fluxes all show marked improvement over the default state of the model when compared with the observational data. In order to produce a stable coupling of the IGCM atmosphere model with the GOLDSTEIN ocean component of the GENIE framework it is imperative that the fluxes between the models are compatible. The methods we have demonstrated for studying the individual components will enable us to achieve this coupling in future GENIE Earth system modelling.

5. CONCLUSION

We have presented the GENIE data management solution and demonstrated its use in the tuning of simulation codes from the GENIE framework. By using tools available in the Geodise toolboxes and OptionsMatlab we have scripted the execution of tuning studies, used local and national Grid resources and exploited our database repository to facilitate the sharing of data as the optimization progressed.

The study of the GENIE models has demonstrated the flexibility of the Grid-enabled Matlab scripting approach. The model binaries are wrapped as functions that are configured to select an available resource on which to perform the Earth system simulation. The optimization tools can be quickly and easily configured to run a variety of optimizers or design search algorithms to tune the simulation. Finally, it is straightforward to modify the objective of the tuning process.

The framework therefore provides all of the tools required for tuning the GENIE models: a scripting environment; a database repository; a computational Grid interface; and optimization 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. We have applied a Kriging method to optimize 12 free parameters in the c-GOLDSTEIN model and employed a GA to study 30 free parameters of the IGCM atmosphere component. These techniques will ultimately enable us to tune the more demanding coupled models that the framework will produce.

ACKNOWLEDGEMENTS

The GENIE project is funded by the Natural Environment Research Council (NER/T/S/2002/00217) through the U.K. e-Science Programme. We are grateful to A. J. Keane and I. I. Voutchkov for their advice and guidance.

REFERENCES

1. Grid ENabled Integrated Earth system model. http://www.genie.ac.uk/ [November 2004].
2. Keane A, Nair P. *Computational Methods in Aerospace Design: The Pursuit of Excellence*. Wiley: New York, 2005.
3. Stainforth D, Kettleborough J, Allen M, Collins M, Heaps A. Distributed computing for public interest modelling research. *Computing in Science and Engineering* 2002; **4**(3):82–89.
4. Edwards NR, Marsh R. Uncertainties due to transport-parameter sensitivity in an efficient 3-d ocean–climate model. *Climate Dynamics* 2005; **24**(4):415–433.
5. de Forster PM, Blackburn M, Glover R, Shine KP. An examination of climate sensitivity for idealised climate change experiments in an intermediate general circulation model. *Climate Dynamics* 2000; **16**:833–849.
6. Keane AJ. The OPTIONS design exploration system user guide and reference manual. http://www.soton.ac.uk/~ajk/options.ps [November 2004].
7. Grid Enabled Optimization and Design Search for Engineering. http://www.geodise.org/ [November 2004].
8. Berman F, Fox GC, Hey AJG. *Grid Computing: Making the Global Infrastructure a Reality*. Wiley: New York, 2003.

9. Avery P. Data Grids: A new computational infrastructure for data-intensive science. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* 2002; **360**(1795):1191–1209.
10. Ahmad K, Taskaya-Temizel T, Cheng D, Gillam L, Ahmad S, Traboulsi H, Nankervis J. Financial Information Grid—an ESRC e-social science pilot. *Proceedings of the 3rd U.K. e-Science All Hands Meeting*, Nottingham, U.K., September 2004. EPSRC, 2004; 355–362.
11. The GridFTP protocol and software. http://www.globus.org/datagrid/gridftp.html [November 2004].
12. Matlab 6.5. http://www.mathworks.com/ [November 2004].
13. Jython. http://www.jython.org/ [November 2004].
14. The Geodise XML Toolbox. http://www.geodise.org/toolboxes/generic/ [November 2004].
15. Eres MH, Pound GE, Jiao Z, Wason JL, Xu F, Keane AJ, Cox SJ. Implementation and utilisation of a Grid-enabled problem solving environment in Matlab. *Future Generation Computer Systems* 2005; **21**(6):920–929.
16. The Java Commodity Grid Kit. http://www.globus.org/cog/java/ [November 2004].
17. Condor: High throughput computing. http://www.cs.wisc.edu/condor/ [January 2005].
18. The Geodise toolboxes: A user's guide. http://www.geodise.org/documentation/html/index.htm [May 2005].
19. Evensen G. The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics* 2003; **53**:343–367.
20. Keane AJ. Wing optimization using design of experiment, response surface, and data fusion methods. *Journal of Aircraft* 2003; **40**(4):741–750.
21. Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer: Berlin, 1992.
22. The NCEP/NCAR reanalysis project. http://www.cdc.noaa.gov/cdc/reanalysis/ [November 2004].
23. Annan JD, Hargreaves JC, Edwards NR, Marsh R. Parameter estimation in an intermediate complexity Earth system model using an Ensemble Kalman Filter. *Ocean Modelling* 2005; **8**(1–2):135–154.
24. Beltran C, Edwards NR, Haurie AB, Vial J-P, Zachary DS. Oracle-based optimization applied to climate model calibration. *Environment Modelling and Assessment* 2005. DOI: 10.1007/s10666-005-9024-4.
25. The U.K. National Grid Service. http://www.ngs.ac.uk/ [November 2004].
26. Statnikov RB, Matuzov JB. *Multicriteria Optimization and Engineering*. Chapman and Hall: New York, 1995.