

Electrochemical Parameter Optimization using Scientific Workflows

Colin Enticott, Thomas Peachey, David Abramson
Faculty of Information Technology
Monash University
Clayton, 3800, Victoria, Australia
{colin.enticott,tom.peachey,david.abramson}@monash.edu

Elena Mashkina, Chong-Yong Lee, Alan Bond,
Gareth Kennedy
School of Chemistry
Monash University
Clayton, 3800, Victoria, Australia
{elena.mashkina,chongyong.lee,alan.bond}@monash.edu,
gareth.f.kennedy@gmail.com

David Gavaghan
Oxford Computing Laboratory
Oxford University
Oxford, UK
david.gavaghan@comlab.ox.ac.uk

Darrell Elton
Department of Electronic Engineering
La Trobe University
Bundoora, Victoria, Australia
D.Elton@latrobe.edu.au

Abstract— Modern chemistry involves a mix of experimentation and computer-supported theory. Historically, these skills have been provided by different groups, and range from traditional “wet” laboratory science to advanced numerical simulation. This paper discusses the application of advanced e-Science software tools to electrochemistry research and involves collaboration between laboratories at Monash and Oxford Universities. In particular, we show how the Nimrod/OK tool can be used to automate the estimation of electrochemical parameters in Fourier transformed voltammetry. Replacing an ad-hoc manual process with e-Science tools both accelerates the process and produces more accurate solutions. In this work much attention was given to the role of the scientist user. During the research, Nimrod/K (on which Nimrod/OK is built) was extended to shield that user from technical details of the grid infrastructure; a new system of file management and grid abstraction has been incorporated.

e-Science; Fourier Transformed Voltammetry; Parameter tools; Optimization; Grid Workflows; Grid computing; Kepler

I. INTRODUCTION

Modern science increasingly employs disparate resources: laboratory equipment, computer clusters and data storage devices, to name a few. Many software tools have been created to simplify access to such resources, but are limited in the services they provide. For example, the execution of a computational model on remote resources may be automated, but the processing, visualization and storage of the results may require manual intervention. In another example, modern microscope control software allows a user to control the zoom, focus and exposure settings remotely, capture an image and store it in the local file system. But, even though such powerful software makes the microscope easy to control, it does not usually automate the complex processing pipeline that follows image capture. Thus, steps involving image analysis, archive

and display are often performed manually on other computer systems.

e-Science infrastructure offers the potential to streamline such experiments. In particular, scientific workflow engines provide mechanisms for sequencing, executing and monitoring, the individual experimental steps, and for transferring information between these steps as required. This paper focuses on how such workflows can be used to combine theoretical models with real world experiments in, robust, novel and powerful ways.

Modern chemistry makes extensive use of both computational models [24] and wet-lab experiments. The application here involves fitting the parameters of a computational electrochemical simulation model so it matches experimental work. Previously, this has been solved in an ad-hoc manual fashion, in which an experiment is performed, and then the simulation parameters adjusted in turn until the output achieves a reasonable fit with the experimental results. Such experiments are used to determine fundamental chemical properties of the reaction, in this case the electron transfer-rate constant.

In previous work we developed a system called Nimrod/K [5] for designing and executing Grid workflow experiments. Recently it has been extended to support file management and abstraction of Grid-resources, this provides a flexible, repeatable and extensible environment for e-Science experiments. Nimrod/OK is an extension of Nimrod/K that uses workflows to solve complex optimization problems [14][15], and this provides a suitable framework for solving the electrochemical problem discussed above. Nimrod/OK applies optimization algorithms, that can be used to automate the parameter fitting by minimizing some measures of the difference between simulated and experimental results. The

current application has different output measures that are sensitive to different input parameters. So the optimization is performed in two stages. One measure is used to determine a subset of the parameters and these are then fixed in the second stage where the other parameters are fitted. This is implemented as a two-stage optimization workflow with the results of the first driving the second.

The case study presented gave a large variation in results for the optimizations. We demonstrate the flexibility of Nimrod/OK by showing how the workflow can be modified quickly to a new experiment that explained these results and yields significant new information.

II. WORKFLOWS, KEPLER AND NIMROD/K

e-Science experiments frequently involves a workflow, a sequence of tasks each building on the results of any earlier ones. A medical research example may involve capturing a three-dimensional image of a patient, cleaning and processing the image to detect tumours, use of computational software to determine a regime of X-ray therapy that destroys the tumours with minimal collateral damage [13], computer controlled delivery of this therapy, further cycles of these operations and archiving the data. As in this case, different tasks may be performed on different machines, to access hardware, software, licenses, processor farms, or storage capacity. Many e-Science workflows are intrinsically Grid applications.

Many tools have been developed to facilitate such scientific workflows [19]. These encapsulate the tasks and handle the flow of information between them. In particular, graphical workflow engines, such as Kepler [6], represent the tasks as screen glyphs and the information flow by connecting arrows. The user may thus create novel workflows just by selecting the tasks (called **actors** in Kepler) from a pallet and connecting them. Figure 1 shows such a Kepler workflow canvas. Groups of tasks can be combined into their own sub-workflow (called **composite actors** in Kepler) to create a new task definition. Customising the actor's parameters is done by double clicking the actor.

With e-Science experiments, it is not uncommon to execute the same software application several times with different data, to process different datasets [18] or to vary input parameters to simulation software to explore ideas [16]. Clearly, it is desirable to perform such executions concurrently if dependencies allow. All workflows environments can achieve this by executing multiple copies of the workflow with the varying input data. In previous work, we introduced Nimrod/K [5], which used a tagged data system to allow a workflow to dynamically set the number of data sets to process in the Kepler environment. This dynamic processing allows datasets to be processed concurrently within the same environment as efficiently as possible. We have added common parameter actions that create threads that can be executed concurrently. Each thread can execute independently, working at the problem assigned by the parameters. In this environment, we can also parameterise a list of files to be processed concurrently.

Nimrod/K is one tool of the Nimrod toolset developed at the Faculty of Information Technology, Monash University. Development began with Nimrod/G [7] that controls repeated

execution of computational models over distributed Grid-resources. It generates the required parameter set for each job, negotiates with various middleware and controls the appropriate transfer of files to and from the computational nodes. Nimrod/G has found widespread application in e-Science [3]. Nimrod/K provides the current parameter set generation tools available in Nimrod/G.

As well as parameter tools, Nimrod/K provides the Grid execution features of Nimrod/G. Orchestrating a workflow action on a Grid machine can often be frustrating for users who want to focus only on the business logic of their e-Science experiment and not the finer details of the Grid. Current workflows typically involve file transfers, execution requests, and then more file transfers to move the data onto the next machine. Exposing these actions on the workflow using low level actors can be time-consuming, error prone and can lack the flexibility of easily reconfiguring the workflow for another Grid machine. We have recently extended Nimrod/K to address these issues within the Kepler environment, hiding these technicalities from the user by providing support for multiple Grid middleware tools, for file management and job processing requests. In Nimrod/K only one actor is needed for the whole action called a *GridJob* actor, which will pre-stage files to the machine, request a processing slot and transfer output files directly to the next machine.

With Nimrod/K's Grid middleware abstraction, the workflow can be reconfigured to use different Grid middleware without needing to recreate the whole workflow. Also, it allows a combination of various Grid middleware on the same workflow. In previous work we used customised Nimrod/G actors as the Grid execution mechanism for computational intensive tasks, but they were developed before the new file management features. File management provided by Nimrod/K records the location of all files associated with the workflow including any replications. It is also Grid middleware independent and supports files being copied by different middleware on the same workflow. The advantage here is the workflow user does not need to explicitly specify how the file is to be copied. The user can focus on what actions need to be performed on files, instead of all the moving and scheduling of the intermittent files. This presents a system where only the business logic of the e-Science experiment needs to be represented on a workflow regardless of the varieties of middleware it uses. These two new features of Nimrod/K remove the restriction of being limited to one Grid middleware layer. It allows the user to combine different middleware solutions into the one workflow and allows the workflow to be easily updated to support new middleware solutions as the technology changes.

Nimrod/K is primarily designed around the idea that multiple users from different disciplines contribute to the whole workflow environment. The four main user types are; the workflow user, the Grid middleware implementer, the actor developer and the scheduler developer. The workflow user assembles all the actors together into a workflow. This user does not need to know what Grid-resource the actor uses, but is more focused on the business logic of the workflow. The Grid middleware implementer provides their services into the Nimrod/K API to allow an actor to use that middleware.

Another user, the actor developer, is responsible for creating a task definition. This could involve creating a Kepler composite actor from existing actors or using the Grid tools provided by Nimrod/K to create a new *GridJob* actor. Like the workflow user, the actor developer is aware of the Grid but they do not need to know explicitly the implementation. This allows the actor developer to focus on providing a remote service as a new action in Nimrod/K without becoming an expert in everything that is Grid related. The scheduler developer is responsible for resource discovery, determining when to invoke actors, copying files and requesting job execution. A discussion of scheduling parameter sweeps in Nimrod/K is reported in [17]. These four users contribute their expertise into one cohesive workflow environment to create a system for running e-Science experiments on the Grid.

III. PARAMETER ESTIMATION AND NIMROD/O

Much e-Science involves the exploration of computational models, models designed to capture the behaviour of the science under investigation. Usually such models require various input parameters; the outputs may be numbers, images or even animations. One methodology for understanding the science is to execute the model for a representative sample of points within the parameter space. If the parameters are not too numerous, this can be done by a parameter sweep where each parameter is allowed certain values and the parameter set sweeps through all combinations. Middleware tools, such as Nimrod/G can assist in the scheduling and execution of such sweeps, using parallel computations on multi-core or Grid/Cloud resources.

Another common technique is determination of those input parameters that produce outputs that are extreme in some sense. For example, a model of a composite laminate may predict the damage under impact loading. In [8] the orientation of the ply layers are design parameters and the aim is to determine the orientations that yield optimally robust laminates. That research used a genetic algorithm; many alternate algorithms are available for model optimization. These optimization algorithms are not guaranteed to yield the best possible parameter sets. Searches are normally iterative and terminate in a local optimum. Often the search space has multiple local optima, and noise introduced by approximations in the modelling process may add extra local optima. The use of multiple independent searches from varying starting points may give some indication of the quality of the optima found. If many searches all give the same result, then the user may have some confidence that the values found constitute a global optimum. If not, then further investigation may be indicated.

Similar optimization problems occur with “inverse problems” where the investigator seeks input parameter values that produce a desired output. A large literature, including complete journals [11][12], has grown around these “inverse problems”. For example, [10] describes extensions to the Shannon-Bers model for ionic flows in rabbit cardiac cells. The model requires settings for nine unknown metabolic constants. Here the inverse problem is to find model values that yield electrical activity matching experimental results. Such inverse problems may be approached as an optimization, minimizing the difference between the model output and real-world

experimental results. This requires evaluation of some metric to measure the difference between these outputs.

With inverse problems, the variables that are directly measured in an experiment are not those variables for which values are required. Craig and Brown [9] argue that inverse problems are intrinsically difficult because information is degraded in the flow from the parameters required to measurable variables and again in the measurement of those variables. Suppose that two different inputs f_1 and f_2 produce respective outputs g_1 and g_2 then there is a danger that “... radically different model functions f_1 and f_2 correspond to data functions g_1 and g_2 which differ very little...” [9;p 11]. Consequently, if a difference metric is being minimized as described above, then the objective function for that difference may be very flat and noise may introduce multiple local minima.

Nimrod/O [4] is a generic tool for computational model optimization. It offers a range of standard algorithms. Multiple algorithms and multiple instances of an algorithm from different starting points may all be executed in parallel. For each iteration of an algorithm, a batch of jobs is required; these may be executed locally on a forked process or passed to Nimrod/G to access Grid computational resources. All jobs are funnelled through the cache Nimcache to avoid duplication and to record results for fast recapitulation if an optimization is aborted. It has been widely used for e-Science inverse problems and for optimal design [3].

IV. NIMROD/OK

Optimization algorithms may themselves be viewed as workflows, usually involving repetitive looping so that results are passed from one iteration to the next. Nimrod/OK extends Nimrod/K to include the functionality of Nimrod/O, exposing the component tasks of an optimization loop and allowing the user to assemble novel arrangements of those components. Thus the user may perform a sweep within an optimization, or an optimization within a sweep, or use the results of one optimization as the starting approximation for another. Nimrod/OK has previously [14] been applied to the Shannon-Bers model mentioned previously. Here we report on an application to electrochemistry which utilises the new Grid functionality of Nimrod/K.

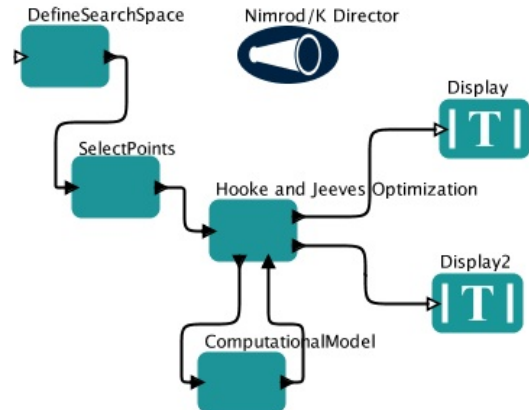


Figure 1: A simple Nimrod/OK workflow

Figure 1 shows a simple example of an optimization canvas under Nimrod/OK where the result of a computational model is optimized by the Hooke and Jeeves algorithm [23]. The domain for the search is specified in *DefineSearchSpace*. In *SelectPoints* a set of “points” is chosen. Here a “point” is a set of values, one for each parameter, within the allowed domain for that parameter. The *Hooke and Jeeves Optimization* actor is triggered by the arrival of these starting points at the left port. One optimization search is initiated for each such point. The Hooke and Jeeves algorithm is used to generate sets of points which are sent to the *ComputationalModel* actor as a batch of jobs for execution. The resulting objective values determine the selection of the next set of points in the optimization. This iterates until some convergence conditions are met. The best point found for each search is forwarded from the top right port of the optimization actor. The lower right port gives a statistical summary of the search effort.

Prior to execution the user needs to customise some actors. In *DefineSearchSpace* the user enters details of the parameters, their names, data types and domain. The *SelectPoints* actor requires the number of such points required and the method of generating them, they may be randomly selected, or evenly spaced throughout the search space, or specifically defined. In optimization actors, specifications of convergence criteria maybe required.

V. THE FOURIER TRANSFORMED ALTERNATING CURRENT (AC) VOLTAMMETRY INVERSE PROBLEM

Dynamic electrochemical techniques are based on the experimental measurements of the response of a chemical system when an electrical input signal is applied. In the case of voltammetry, a time-dependent potential V is applied and the response, a current I , is measured as a function of time [1][2]. To date, voltammetry has been widely applied in the laboratory to quantify the kinetics and thermodynamics associated with electron transfer processes. These are of interest because of their ubiquity and essential role in many physical, chemical, and biological processes.

More specifically, in Fourier transformed AC voltammetry, a sinusoidal potential, frequency f , is superimposed on a more slowly varying saw-tooth function. As the electron transfer component of the system is non-linear, the output I contains a component with the same frequency f , and harmonics with multiples of that frequency.

This work investigated the one-electron transfer process $\text{Red} \rightleftharpoons \text{Ox} + e^-$ where the electroactive species are alternately oxidised (Ox) and reduced (Red). The theoretical model uses the Butler-Volmer equation (assuming the charge transfer coefficient α is 0.50) for this transfer and Fick’s laws of diffusion to calculate the details of the accompanying mass transport of Ox and Red to and from the electrode surface, respectively. It predicts the net current flowing at the working electrode in terms of the electron transfer-rate constant, k ; determination of that constant is the prime aim of this work. Other terms in the model are either known constants or can be directly measured during an experiment, with the exception of the uncompensated resistance, R , and the electrochemical double layer capacitance function $C(V)$. Thus this is a standard inverse problem where the aim is to determine k , R and $C(V)$

that produce model outputs agreeing with those obtained from physical experiments.

The current procedure for the determination of the parameters uses a computer controlled physical experiment to obtain voltammetry data for a given analyte. Then, the equivalent computational model is executed repeatedly with various parameters, and results evaluated by observing the differences between waveform envelopes for the physical and simulated systems in the, so called, ‘heuristic approach’ [2]. Analysis of harmonics indicates that the fundamental harmonic provides the basic information on the $C(V)$. In the case of second and higher harmonic, effectively shorter time scales are being probed than for the fundamental harmonic, so that the dependence of the higher harmonics on k and R is even greater than for the fundamental harmonic. The aim of the new work is to mimic this procedure within Nimrod/OK where parameter fitting will be expedited by the use of standard optimization algorithms and distributed execution on Grid-resources.

VI. TEST CASE

As a test case, a standard three-electrode cell (with glassy carbon, Pt-wire and Ag/AgCl as working, counter and reference electrodes respectively) was used to study the heterogeneous electron transfer process for oxidation of ferrocene. The electroactive analyte, 1mM ferrocene was dissolved in acetonitrile in the presence of 100mM tetrabutylammonium hexafluorophosphate as a supporting electrolyte. The experimental parameters are as follows: ramp potential - 0.3V to 1.0V to -0.3V, scan rate 0.9686s^{-1} , frequency 9.02Hz and temperature 295K. The experiment was conducted using a computer based instrumentation system developed by Elton [1].

The simulation for this experiment used MECSim, a FORTRAN code based on the matrix formulation presented in [20]. MECSim was developed in-house at Monash and has been applied in previous work [21]. All input parameters for the model were fixed except: the electron transfer rate constant k , the uncompensated resistance, R , and four coefficients determining the capacitance function. The output is a table of current values corresponding to that produced by the physical experiment.

A Matlab program is used to compute metrics for the difference between experimental and simulated outputs. First the output data is separated into components using simple band-pass filters in the Fourier domain; if the input sinusoid has frequency f then the i^{th} component contains frequencies in the range $[(i - 1/2)f, (i + 1/2)f]$. Preliminary experiments failed using a metric based on the sum of squares of differences of these components; optimal fit was achieved with $k = 0$, for which the simulated current was zero. This was found to be caused by a phase difference between the simulated and physical sinusoidal components of applied signal. The simplest way to obviate this problem was to fit a spline envelope to the harmonics in the time domain, and to compute a difference metric on those envelopes. The same method was used previously when the optimization was performed informally.

Note that the computation described gives a sequence of metrics, one for each harmonic. We adopted the methodology

described in the previous section, using the fundamental harmonic to estimate the capacitance parameters and a higher harmonic for R and k .

VII. EXPERIMENT 1

Figure 2 shows the Nimrod/OK workflow canvas for this experiment. The left and right sections of the workflow correspond to two separate optimizations using the simplex algorithm [22]. On the left the *DefineSearchSpace* actor gives ranges for k , R and the four capacitance parameters, and the *SelectPoints* actor picks random values for these six parameters within their allowed ranges, three sets are chosen to seed three optimizations. For each of the three starting points the *SimplexOptim* actor performs a simplex search sending batches of jobs to *Voltammetry* for simulation. Note that the *Voltammetry* actor has four outputs; these give the respective metrics for the first four harmonics. In this case we are using the first harmonic as the capacitance parameters are known to be sensitive to that value. The *SimplexOptim* actor is configured with the convergence criterion. In this experiment iterations cease when the proportional variation throughout the simplex vertices is less than the tolerance $\varepsilon = 0.0001$. When the algorithm has converged the optimal point is forwarded to the *ChooseBest* actor which selects the best of the three optimization results.

Action moves to the next stage with new *DefineSearchSpace* and *SelectPoints* actors. These are configured to define only values for k and R , so the other parameters, the capacitance values found in the first stage, will be preserved and are constants during the next optimization. Here the *SelectPoints2* actor generates ten starting points for k and R , so the second simplex actor performs ten optimizations. In this case the third harmonic is optimized as it is known to be more sensitive to these parameters. Final results go to the *Display* actor.

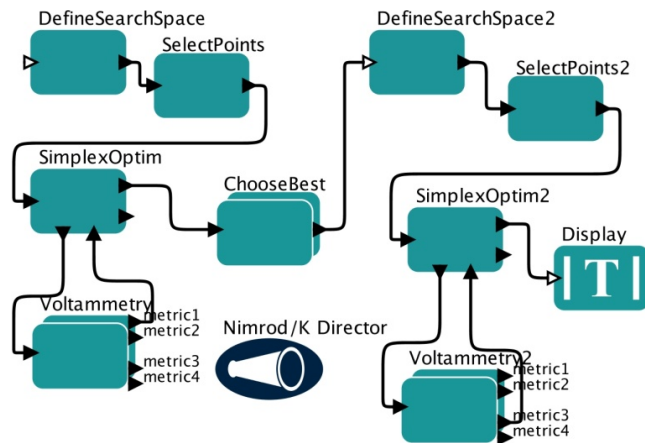


Figure 2: Workflow for Experiment 1

Note that the *Voltammetry* actors are composite actors, as signified by the doubling of their icons. Figure 3 shows their components. The input port takes values for the six parameters, which are then used by the *MECSim* actor which encapsulates the MECSim FORTRAN code. Results in the form of a Grid file are forwarded to the *EnvelopeMetric* actor for comparison

with experimental results. Those experimental results are currently supplied by another Grid file; firing of the *MECSim* actor will trigger that file also to be forwarded. These Grid files allow the *MECSim* actor and the *EnvelopeMetric* actor to run on different Grid-resources without the user needing to explicitly specify file copying. The *MECSim* and the *EnvelopeMetric* actors are extended from the *GridJob* actor to access the Grid job and file management features of Nimrod/K. For this first experiment, we ran the MECSim simulation and the metric computation on an 8 core Intel Xeon (E5310) 1.6GHz machine. In this experiment Nimrod/K is not using any Grid middleware, but utilises the machine as a Grid compute node.

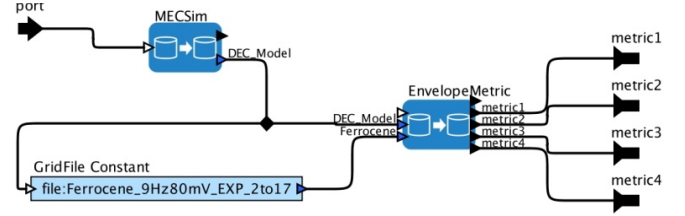


Figure 3: Expansion of the Voltammetry sub-workflow

The experiment ran for 297 minutes with the first optimization stage running for 232 minutes, which produced 571 batches and a total of 2,329 executions of both codes. The second stage of the workflow ran for 65 minutes with 184 batches and 725 executions. Even though the second optimization had more starts, they converged earlier due to the lower dimensionality of the search space. The MECSim code ran for an average of 7.3 seconds and the Envelope Metric code had an average of 31.2 seconds. The experiment would take 32.5 hours on a single processor giving us a speed up factor of 6.6. The theoretical speed up of 8 cannot be reached as only four processors are used at the end of each optimization hence under-utilizing the machine.

Opt	k (cm s ⁻¹)	R (ohm)	Metric (10 ⁻⁹ amp ²)
1	0.509	144.9	1.2678
2	0.688	147.6	1.2673
3	0.874	149.5	1.2719
4	1.000	151.4	1.2760
5	0.488	144.8	1.2694
6	0.590	146.2	1.2643
7	0.337	139.1	1.2859
8	1.000	151.0	1.2740
9	0.406	141.5	1.2698
10	0.396	141.4	1.2719

Table 1: Results for k and R of ten optimizations.

Results of the optimizations are shown in Table 1. They show consistent results for the uncompensated resistance R , but a great variation in the electron transfer rate constant k , with a least value of 0.377 and a greatest of 1.000 where two of the searches collided with the upper bound for k . Since the

determination of k is the prime aim of the experiment, these results show that the aim is not met. It was decided to investigate the search space further by performing a sweep.

VIII. EXPERIMENT 2

To explain the results of the previous experiment a sweep was performed with 20 by 31 evaluations equally spaced through the search space, and the results used to create a contour map for the objective metric. Little change was required in the workflow; the second optimization was replaced by a sweep actor and the results piped to a *Contour* visualization actor. The Nimrod/K *Parameter Sweep* actor has been described in earlier work [5]. The *SweepToArray* actor waits for all results from the sweep and sends the results as one array to the next actor. The *Contour* actor uses a Matlab process. Figure 4 shows the new workflow.

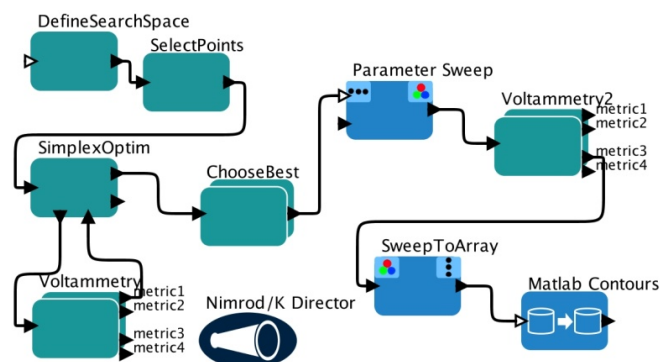


Figure 4: Parameter Sweep workflow

The resulting contour map is presented as Figure 5. Optimal fit with experiment corresponds to points in the central white section. This is a long valley with little variation of the metric along that valley. Hence any optimization procedure will be unable to yield a localized estimate of k . The only reasonable conclusion is some lower bound, say $k > 0.2$.

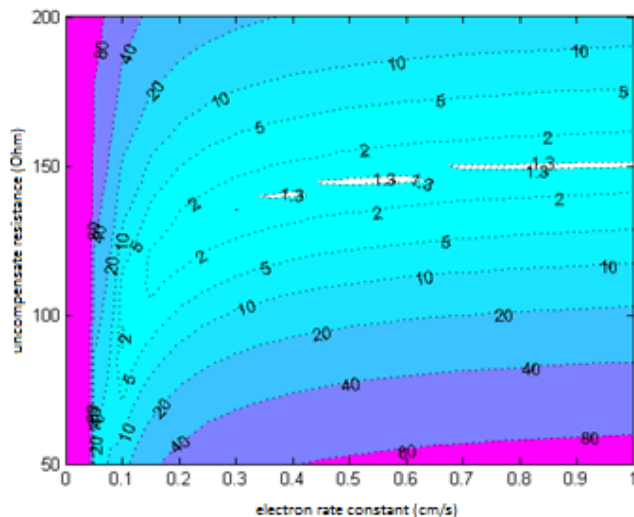


Figure 5: Contour map of the results from the parameter sweep. Contour values are 10^{-9} amp^2

To run this experiment we used our Grid compute cluster to execute the code. This was done by changing an option in the *GridJob* actors, from “Local machine” to “Nimrod/G middleware”. As a comparison with the previous experiment, we requested 8 processing slots using the same processors as the first experiment. A total of 620 executions of both codes would equate to 6.6 hours on one processor, however while running on the cluster, both codes performed file transfers to run. The MECSim code produced a 5MB file and the Envelope metric code required two 5MB files and other configuration files. This added a considerable amount of time to the experiment as MECSim ran for an average of 10.9 seconds and the Envelope metric code took 46.6 seconds. These new statistics now increases the theoretical best to 9.9 hours. The experiment ran for 75 minutes that gave us a speed up factor of 7.92. The rest of the time can be attributed to the queuing system on the cluster.

IX. CONCLUSIONS AND FURTHER WORK

This work is the result of collaboration between chemists and computer scientists. From a chemistry perspective the results demonstrate that the voltammetry experiments used here require further development (use of square, triangular, sawtooth wave forms and multiple sin waves of different amplitudes and frequencies, etc.), perhaps with extra criteria needed to achieve accurate determinations of the rate constant. The computer science interest centres on the development of a workflow tool, Nimrod/OK, that is well-adapted to e-Science experiments, in particular to parameter estimation inverse problems. The experiments presented show the power and flexibility of this approach with the technicalities of Grid computing hidden from the user. The chemists have an environment where they can focus on putting together chemistry applications and not worry about adding finer grained Grid steps in between. The workflows are repeatable and extensible and can be easily configured for newer experiments when Grid tools change.

The workflow framework facilitates further experiments. Future work will investigate variation of the signal frequency, implementation of extra parameters as announced (e.g. mid point and reversible potentials and charge transfer coefficient) into the optimization search, more complicated mechanisms where chemical reactions are coupled with electron transfer and multi electron transfer reactions. Testing of the ability to use Marcus-Hush theory in the simulation routine, as well as the use of other search algorithms and the metrics will be undertaken.

In this work, incorporation of the experimental results from the physical experiment required user intervention to transfer the appropriate file. To complete the system, we intend to see if it is feasible to add an actor for the experimental control so that this becomes part of the workflow, making the full experiment seamless and less susceptible to error.

ACKNOWLEDGMENT

This work has been supported by the Australian Research Council under the Discovery grant scheme. We thank our colleagues from the MeSSAGE Lab for their support in this work.

REFERENCES

- [1] Bond, A. M., Duffy, N. W., Guo, S.-X., Zhang, J., and Elton, D. M. "Changing the Look of Voltammetry", 2005 Anal. Chem. 77, pp 186A-195A.
- [2] Sher A. A., Bond A. M. et al., "Transformed Large Amplitude Square-Wave Voltammetry as an Alternative to Impedance Spectroscopy", Electroanalysis 2005, 17, pp 1450-1462.
- [3] <http://messagelab.monash.edu.au/EScienceApplications>
- [4] Abramson, D. Peachey T. and Lewis A., "Model Optimization and Parameter Estimation with Nimrod/O", The International Conference on Computational Science, May 28-31, 2006, University of Reading, UK.
- [5] Abramson D, Enticott C, Altintas I., "Nimrod/K: Towards Massively Parallel Dynamic Grid Workflows", IEEE SuperComputing 2008, Austin, Texas November 2008.
- [6] Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao J. and Zhao, Y. "Scientific Workflow Management and the Kepler System", Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows, 2005.
- [7] Abramson, D., Giddy, J. and Kotler, L. "High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?", International Parallel and Distributed Processing Symposium (IPDPS), pp 520- 528, Cancun, Mexico, May 2000.
- [8] M. Yong, B.G. Falzon, L. Iannucci, "On the application of genetic algorithms for optimising composites against impact loading", International Journal of Impact Engineering, Volume 35, Issue 11, November 2008, Pages 1293-1302.
- [9] Craig I. J. D. and Brown J. C., Inverse Problems in Astronomy, Adam Hilger, 1986.
- [10] Amirrazi S., Chang S., Peachey T., Abramson D. and Michailova A., Optimizing Cardiac Excitation-Metabolic Model By Using Parallel Grid Computing, Biophysics 2008, Long Beach, California, February 2008.
- [11] Inverse Problems in Science and Engineering, <http://www.tandf.co.uk/journals/titles/17415977.asp>
- [12] Inverse Problems, <http://iopscience.iop.org/0266-5611>
- [13] Chin P.W., Lewis D. G. and Giddy J. P., "A Monte Carlo solution for external beam photon radiotherapy verification", The Monte Carlo Method: Versatility Unbounded in a Dynamic Computing World, Chattanooga, Tennessee, April 2005 pp 17-21.
- [14] Abramson, D., Bethwaite, B., Enticott, C., Garic, S., Peachey, T., Michailova, A., Amirrazi, S., "Embedding optimization in computational science workflows", Elsevier, Journal of Computational Science 1, 2010, pp 41-47
- [15] Abramson, D., Bethwaite, B., Enticott, C., Garic, S., Peachey, T., Michailova, A., Amirrazi, S., Chitters, R., "Robust Workflows for Science and Engineering". 2nd Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS 2009), Portland, USA, 16 November 2009
- [16] Abramson D., Bethwaite B., Enticott C., Garic S., Peachey T., "Parameter Space Exploration using Scientific Workflows", ICCS 2009, Baton Rouge, Louisiana, May 2009
- [17] Smachat, S., Indrawan, M., Ling S., Enticott, C. and Abramson, D. "Scheduling Multiple Parameter Sweep Workflow Instances on the Grid", IEEE e-Science 2009, Dec 9 - 11th, Oxford, UK.
- [18] Abramson, D, Bethwaite, B, Dinh, M, Enticott, C, Firth, S, Garic, S, Harper, I, Lackmann, M, Nguyen, H, Ramdas, T, Russel, A.B.M, Schek, S and Vail, M., "Virtual Microscopy and Analysis using Scientific Workflows", IEEE e-Science 2009, Dec 9 - 11th, Oxford, UK.
- [19] Yu J., Buyya R., A Taxonomy of Workflow Management Systems for Grid Computing. J. Grid Comput. 3(3-4): 171-200 (2005).
- [20] Rudolph M., Reddy, D. P. and Feldberg, S. W. "A Simulator for Cyclic Voltammetric Responses", 1994, *Analytical Chemistry*, 66, 589A-600A.
- [21] Shiddiky M. J. A., Torriero A. A. J., Chuan Zhao, Bugar I., Kennedy G. and Bond A. M., "Nonadditivity of Faradaic Currents and Modification of Capacitance Currents in the Voltammetry of Mixtures of Ferrocene and the Cobaltocenium Cation in Protic and Aprotic Ionic Liquids", 2009, Journal of the American Chemical Society, 131(23), 7976-89.
- [22] Nelder, J.A. and Mead, R., "A simplex method for function minimization", Computer Journal, 1965, 7, 308-313,
- [23] Hooke R. and Jeeves T. A., "Direct search solution of numerical and statistical problems", Journal of the Association for Computing Machinery, 1961, 8, 212 - 229.
- [24] Cramer C. J., Essentials of Computational Chemistry, Theories and Models, Wiley, 2002.