

Fractional factorial design for parameter sweep experiments using Nimrod/E

T.C. Peachey^a, N.T. Diamond^a, D.A. Abramson^{a,*}, W. Sudholt^b, A. Michailova^c and S. Amirriazi^c

^a *Monash University, Clayton, 3800, VIC, Australia*

^b *Institute of Organic Chemistry, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland*

^c *Department of Bioengineering, University of California San Diego, and San Diego Supercomputer Centre, 9500 Gilman Drive, La Jolla, CA, USA*

Abstract. The techniques of formal experimental design and analysis are powerful tools for scientists and engineers. However, these techniques are currently underused for experiments conducted with computer models. This has motivated the incorporation of experimental design functionality into the Nimrod tool chain. Nimrod has been extensively used for exploration of the response of models to their input parameters; the addition of experimental design tools will combine the efficiency of carefully designed experiments with the power of distributed execution. This paper describes the incorporation of one type of design, the fractional factorial design, and associated analysis tools, into the Nimrod framework. The result provides a convenient environment that automates the design of an experiment, the execution of the jobs on a computational grid and the return of results, and which assists in the interpretation of those results. Several case studies are included which demonstrate various aspects of this approach.

Keywords: Experimental design, computer models, distributed execution

1. Introduction

Scientific experimentation underpins just about every aspect of modern life, from applied disciplines with practical outcomes through to our theoretical understanding of the natural world. The idea of exploring the world by controlling inputs and observing outputs is so much part of modern culture that it escapes close scrutiny in our education systems. In fact, many of us don't think about how to set up experiments in any formal way.

Early last century researchers used ad hoc approaches to varying the controllable parameters of an experiment. However, this changed in the 1920s when Fisher devised a formal and structured approach called Experimental Design [5,12,13]. Fisher showed how to select parameters in order to obtain the maximum information for a given number of experimental runs. Since then, Experimental Design has been used in a wide range of activities, and is now a large and mature theory, widely applied to scientific research and industrial processes.

With the growing power and availability of computers, physical experiments are increasingly being

augmented or replaced by experiments with computer models [19]. Such models typically have many input parameters. Researchers use their knowledge of the underlying science to select the parameters that will be varied. Use of wide area computation “grids” allows many thousands of runs, even for computationally intensive models [31]. Even this, however, may be insufficient to explore all the input parameters of interest.

We perceive a need for more efficient experimental designs in the field of computer modelling. Factorial and fractional factorial designs are the most commonly used experimental designs [5], but have been underused in computer experiments. To facilitate these techniques, we have developed tools that automatically generate fractional factorial designs, perform the experiment, and then provide an analysis of the results. The tools interface with an existing tool called Nimrod/G [2], which is used to organise the execution of the model using the parallelism offered by a computational grid. The new system, known as Nimrod/E, augments the existing Nimrod tool chain [2,3,25].

The behaviour and applicability of fractional factorial designs is explored here for four case studies: two simple mathematical functions where output values and relative importance of all parameter combina-

*Corresponding author. E-mail: davida@csse.monash.edu.au.

tions can be easily determined, and two more complex realistic scientific examples.

In this paper, Section 2 expands on the use of computer models in research. The theory and application of fractional factorial design is detailed in Section 3. Nimrod is described from a user's point of view in Sections 4 and 5; the former gives a general overview of the Nimrod toolset and the latter concentrates on the provision of fractional factorial experiments using Nimrod/E. Section 6 describes the use of Nimrod/E in four case studies, chosen to demonstrate different aspect of such experiments.

2. Experiments using computer models

Computer models are now ubiquitous in many areas of research, in both physical sciences and social sciences [19]. A model that can accurately mimic a real world situation offers many advantages. It may enable experiments in situations where the real system cannot be used, where it may be physically impossible [27], too expensive [1] or unethical [34] to manipulate. The model may allow many more internal variables to be monitored and so a more detailed understanding of the behaviour. Availability of computer models has extended the possibility of experimentation to disciplines that were formally descriptive. For example, an epidemiologist may explore different drug control strategies (for example, providing public injecting rooms for injecting drug users) and may evaluate metrics such as the spread of communicable diseases like HIV/AIDS and Hepatitis [20].

A similar trend is occurring in engineering design [16]. Mathematical models are now sufficiently complex to make realistic predictions of the performance of machines and structures. This is typically cheaper than traditional prototyping, and allows a more thorough exploration of design parameters.

Computer models often have a very large number of input parameters and internal settings. Current practice with these models typically involves the experimenter identifying which parameters to vary, selecting input values for these parameters, and then generating all possible combinations of these values. In some cases, such experimentation may be required just to determine the domain over which the function operates. Each combination requires a run of the computer model. So varying even a small number of parameters can generate a huge number of runs (jobs).

The task of generating and running these jobs may be automated by software such as the Nimrod family of tools [2,3,25]. When the models are computationally intensive, then the experiment will benefit by concurrent execution of jobs on a cluster or on the distributed resources of a computational grid. Nimrod automates the distribution of jobs over available resources and the collation of results. The power of this approach is illustrated, for example, by an experiment performed at the IEEE Supercomputing Conference in 2003 involving a quantum chemistry code [31,33]. A single job with an individual parameter combination took typically about 20 minutes to run on a fast Unix machine, and the parameter sweep generated 60000 such jobs, an 800 day task for a serial machine. By using a global testbed of multi-processor clusters as a computational grid, the full experiment was completed within three days.

However, combinatorial explosion ensures that increasing the number of parameters will quickly overwhelm any computational resource. For example, just 10 parameters with four values each generate a million jobs. In the language of experimental design [5], this approach of taking the direct product of parameter values is known as a full factorial experiment. Experimental design shows how a suitably chosen subset of the full factorial experiment will be sufficient to obtain reliable results, given certain assumptions that usually apply in practice. These techniques can be dramatically parsimonious; in the 10 parameter situation above, results for a full sweep can be reconstructed from the results of just 1024 jobs, with some confidence in the results.

One difference between computer experiments and physical experiments on real world systems is the repeatability of results. Results for physical experiments have an unpredictable component, due to uncontrolled inputs, or even quantum uncertainty. Statistical analysis helps the experimenter to distinguish the signal from the noise.

For computer experiments however, unless the model is designed with random number elements, the results it produces for a given set of inputs are always the same. This gives the impression that the results are exact and have no noise component, but close inspection of most computer models refutes this. For example, many models are subject to floating point errors which may accumulate to significant values. And most models of real world systems use some discretization of the continuum and the exact choice of this effects the results. Further, models often use constant values for settings that are in fact experimental values and

hence subject to uncertainty. So the results of a computer model do contain noise and can be treated as single observations from a distribution. Thus the methods of analysis used for physical experiments apply also to computer experiments.

There is a considerable literature on the application of experimental design to computer experiments [11, 24, 28]. However, these techniques have not found wide acceptance within many areas where computer experiments are practised. We believe that a significant reason for this is that they are not embedded in the design tools of choice. To remedy this deficiency, we propose to incorporate the techniques of experimental design within the Nimrod job execution framework. The first part of this project, the implementation of fractional factorial design within Nimrod, has been completed. We demonstrate the usefulness of the approach with a variety of case studies.

3. Fractional factorial experimental design

3.1. Theory

A full factorial experiment consists of a number of factors (parameters¹), each with a number of levels (values), with all combinations of the levels being performed. The effects of each of the variables on the response, called main effects, can be determined, as well as the two-way interaction effects of each of the pairs of factors, and higher-order interaction terms.

The simplest situation is where each factor takes only two levels, a low value and a high one. Suppose the factors are A, B, C, \dots and the output of interest is ϕ . Then ϕ may always be written as

$$\begin{aligned} \phi = & k + \{k_1a + k_2b + k_3c + \dots\} \\ & + \{k_{12}ab + k_{13}ac + k_{23}bc + \dots\} \\ & + \{k_{123}abc + \dots\} + \dots \end{aligned} \quad (3.1)$$

Here a is -1 when A takes the low value and $+1$ for the high. Similarly b, c, \dots are -1 or $+1$ corresponding to whether B, C, \dots are low or high. The other symbols are constant for a given output function and are called “effects”, k is the mean effect, the k_i are the main effects, k_{ij} are the two-way interaction effects, and k_{ijk} etc. are higher-order interaction effects. So,

for example, if $k_{13} = -3.0$ then the interaction of a and c makes a contribution of -3.0 to the output when a and c are both high or both low, and a contribution of $+3.0$ otherwise.

If there are n factors A, B, \dots , then there will be 2^n effects in formula (3.1). A full factorial experiment consists of 2^n runs. Substituting the results into (3.1) provides 2^n linear equations for the effects, sufficient to evaluate them. (In the presence of noise the values will be estimates rather than exact determinations.) Assuming that the higher-order interactions contribute negligible effects, then (3.1) can be approximated by a formula with fewer terms, so a smaller set of runs may be enough to determine estimates of the remaining effects. The runs however must be appropriately chosen since different combinations of inputs can give the same linear equation for a truncated version of (3.1).

Suppose, for example, we only perform runs for which $abcde = +1$, these constitute one half of a full factorial experiment. Then the a term in (3.1) will always have the same sign as the bcd term, so the two terms are effectively combined as one, namely $(k_1 + k_{2345})a$. The estimate obtained for the primary effect of A will then include the estimate for the interaction of B, C, D and E . We say that A is “biased” by $BCDE$. Further, the estimate for AB will be biased by CDE , AC will be biased by BDE , and so on. Assuming that three-way interaction effects and higher are negligible, these biases will not be a problem. There are now half as many simultaneous linear equations but half as many effects to be determined. For this example, $ABCDE$ is called a “word in the defining contrast”; each additional word in the defining contrast halves the number of runs, resulting in a “fractional factorial design”.

To create a fractional factorial experiment the designer must first decide which effects are required to be estimated and which can be considered negligible. Defining contrasts are then chosen that allow the former to be biased by the latter. More fully, the effects are divided into three classes:

- Primary effects: effects for which estimates are required.
- Secondary effects: effects for which estimates are not necessarily required, but are not allowed to bias the estimates of the primary effects.
- Other effects: effects for which estimates are not necessarily required, and are allowed to bias the estimates of the primary effects.

¹We adopt Fisher’s nomenclature from this point on, referring to function “parameters” as “factors”.

For a full factorial design, the primary effects would include all effects, main effects, two-factor interactions, three-factor interactions, and so on. By restricting the primary effects, the size of the design (i.e., the number of runs) can be reduced.

The most important fractional factorial designs are described using a special shorthand terminology developed by Box [5]. These are referred to as “resolution III”, “resolution IV” and “resolution V”. In resolution III designs, the primary effects set is the set of all main effects, while the secondary effects set is empty. Hence, main effects might be biased by two-factor interaction effects. In resolution IV designs, the primary effects set is the set of all main effects, and the secondary effects set is the set of all two-factor interaction effects. Main effects may only be biased by three and higher-order interactions, but not two-factor interactions, and the two-factor interaction effects may not be estimable. In resolution V designs, the primary effects set is the set of all main effects and the set of all two-factor interaction effects, and the secondary effects set is empty. With these designs all main effects and two-factor interaction effects can be estimated, biased by three-factor and higher order interaction effects. Designs with higher resolution than V can be produced by extending this pattern, but these are less commonly used since three-factor interactions are usually small.

Fractional factorial designs are easier to design and analyse for two-level factors than for factors with more than two levels. However, a 2^n level factor is equivalent to n two-level “pseudo-factors”, and hence the theory and practice for two-level designs can be easily extended to designs with factors with more than two levels. For direct use of pseudo-factors, the number of levels needs to be a power of two. For example if a factor a has four levels, then these are indexed by two pseudo-factors, say $A1$ and $A2$, each corresponding to one bit in the index. While it is convenient to design the fractional factorial experiment in terms of the pseudo-factors, the analysis is best conducted by using orthogonal polynomials for the main effects, and for the interaction effects for designs of resolution V. See Draper and Smith [10] for more details.

3.2. Analysis

An experimental design will provide a list of runs to be performed; each run assigns a value for each input factor. (In computer modelling the runs are usually called “jobs”.) Once these jobs have run and each produced a numerical result, then the estimates of ef-

fects, the coefficients in Eq. (3.1), can be obtained as described above. These results can be used in various ways.

1. The relative sizes of the effects may be of scientific interest. For example, very small values for the main effect and interactions of a particular factor suggest that the factor may be irrelevant in the model. If two factors have significant main effects but a small interaction, then the factors are making independent contributions to the result. The experimenter with knowledge of the science is best placed to interpret and make use of such results. In particular, disagreement between the effect estimates and what would be expected from the science may indicate problems with the computational model.
2. Results for effects may suggest larger experiments. For example if, in a resolution V experiment, the two-way interactions AB and AC are found to be large, it suggests that the three-way interaction ABC may not be negligible. The experimenter may then add this to the primary effects and design a new experiment that will estimate this effect.
3. An experiment with a small number of levels per factor may be performed as a preliminary to one with more levels. The factors found most significant in the preliminary experiment might warrant more levels in the later one.
4. When many levels are taken for a factor, the effect corresponding to the least significant bit gives an indication of the smoothness of the response surface.
5. Knowledge of the relative sizes of effects can assist in subsequent optimization [8] procedures. For example, if factors have negligible effect then it may pay to first perform a search that ignores those factors, as reduced dimensionality assists some search algorithms. Again, if the factors fall into two groups with no interaction between groups, the output will be the sum of two independent parts. Then optimizing each part separately is possible and may be advantaged by the reduced dimensions.
6. Estimates of the effects can be used in Eq. (3.1) to compute the results of a full sweep. Where there are many factors, this may be the only practical way to produce such results.
7. The results produced by predictions based on estimates of effects are a form of data smoothing as higher order interaction effects are removed. The

results may be used to reduce noise or to detect errors in the results.

It follows from our discussion in Section 2 that the estimates of effects should be treated as containing some random noise. How do we decide whether the values obtained are significant or just noise? The standard methods of analysis assume that the noise components are normally distributed with a zero mean. On this basis, two graphical methods have been devised that display all the estimated effects and facilitate judgement of their significance: the Daniel plot [9] and the Lenth plot [17].

A Daniel plot shows the effects in numerical order on probability paper. Under the hypothesis that none of the factors affects the result, the plotted points will fall in a straight line. Typically, most higher-order effects are negligible, and so many points do fall on a line. Those points that clearly do not lie in the line are probably significant. An example is shown in Fig. 7. It clearly shows that single effects A , B and the interaction AB are significant. However, the assessment of these plots is somewhat subjective.

An alternative display is the Lenth plot. Here the estimates of effects are plotted vertically as in Fig. 6, in order from largest to smallest in absolute size. Horizontal lines indicate the significance of the effects. The inner dashed lines are known as ME (margin of error) lines. Effects that are in fact zero have a probability of 95% of falling within these lines. The outer solid lines, SME (simultaneous margin of error), apply to all effect estimates as a group. If all effects are zero there is a probability of 5% of one estimate falling outside these lines. So if a point falls outside the SME lines then the effect is probably significant, while if the point falls within the ME lines then there is no strong evidence of significance, non-zero values may just be noise.

Most computational models of course produce many numerical results. Once a fractional factorial experiment has been designed and run, a separate analysis may be produced for each result. The effect estimates may be very different in the separate analyses. For example, one output may be strongly affected by factors A , B and C while another by factors D , E and F . Such information may be significant in understanding the computational model.

The fractional factorial methodology may give the impression of a simple linear approach: pose the question, design the experiment, take the measurements, and interpret the results. However, real problems are rarely as simple as that, as Box [6] has observed. Box describes two categories of research: the initial “dis-

covery” phase that generates hypotheses, and the later “testing” phase where they are confirmed or rejected. As Box explains, there are no recipes for the discovery phase of research. It depends on both implicit and explicit knowledge of the researchers, on judgement and on lateral thinking. It is typically a messy iterative procedure. We see fractional factorial experiments as a useful component in this iterative process.

4. The Nimrod toolset

Nimrod is a family of tools that facilitate parametric studies with computational models, that is, the exploration of the effect of varying the input parameters on the results produced. Nimrod has been extensively used for this purpose [7].

Nimrod/G [2] was designed to assist engineers and scientists in performing such studies, using concurrent execution on a cluster of processors or the resources of a computational grid. The user prepares a “plan file” which specifies the experiment; Fig. 1 shows an example. The first section specifies the parameters (factors) and the values assigned (levels). The “task” section specifies how the computational model is to be executed using distributed execution. Here the executable “compModel” is copied to a remote node together with a generic input file, the input file is “substituted” to insert current values for the parameters and the model then executed to produce results in a file called “results”. This is then copied back to the root node and given an extension “\$jobname”, which is a unique Nimrod identifier.

A plan file is expanded into a “run file” which lists the appropriate parameter combinations required for execution. The Nimrod/G core will run on a processor called the “root node” whereas individual jobs are executed on “remote nodes”. Nimrod/G handles the file transfers required to the remote nodes, execution of computational tasks and transfer of results back to the root node. The number of concurrent jobs is limited only by the number of processors available. Thus the user may achieve high concurrency without modifying the executables and without concern for grid specific details. A commercial version, enFuzion [4], has been used in financial modelling, bio-informatics, graphics rendering and modelling of power grids.

Nimrod/O [25] optimizes the numerical outputs of computational models. It provides a range of optimization algorithms and leverages Nimrod/G or enFuzion to perform batches of concurrent evaluations. The user

```

parameter temp float range from 200 to 300 points 2
parameter pressure float range from 5000 to 6000 points 4
parameter concent float range from 0.002 to 0.005 points 2
parameter material text select anyof "Fe" "Al"
task main
  copy compModel node:compModel
  copy inputFile.skel node:inputFile.skel
  node:substitute inputFile.skel inputFile
  node:execute ./compModel < inputFile > results
  copy node:results results.$jobname
endtask

```

Fig. 1. Plan file for a Nimrod/G experiment.

Fig. 2. Nimrod portal planning page.

prepares a “schedule file”, which, like the Nimrod/G plan file, specifies the parameters, their ranges and the tasks required for execution of the model. But it also specifies the optimization algorithms to be used and the settings for the algorithms. Nimrod/O performs multiple searches for an algorithm and multiple algorithms, all in parallel.

Although all of the Nimrod tools can be used by experienced users in a “command line” mode, most users prefer the interactive Nimrod Portal [22]. The Nimrod Portal uses drop down menus to design and run an experiment and to select computational resources. Such resources may be added or removed as the experiment proceeds. The Portal allows a user to manage their computational testbed without resorting to the detailed and complex command line tools used by the underlying grid middleware.

Figure 2 shows a page from the Portal where the user plans the experiment. When the user has selected parameters and tasks, the “save” button will create a Nimrod plan file.

Whilst a Nimrod user could apply the experimental design techniques discussed in Section 3 by consult-

ing published tables of designs, or experimental design programs, to prepare a run file, we have integrated the techniques into the Nimrod tool chain. The result is convenient and less prone to error, and is the motivation for the development of Nimrod/E.

5. The design and architecture of Nimrod/E

A fractional factorial experiment within the Nimrod framework is a four-stage process, as shown in Fig. 3. Given the factors to be controlled and the levels used for those factors, the first stage is the production of an experimental design. For this purpose, we have implemented the RSSEF algorithm (discussed in Section 5.2) in a program called *nimrodFracDes*. The program produces a Nimrod/G run file, which specifies which jobs are required. When Nimrod/G has completed these jobs and assembled the corresponding result files, a third application called *nimrodFracAn* analyses the results, producing results in a graphical form known as Daniel plots and Lenth plots. The

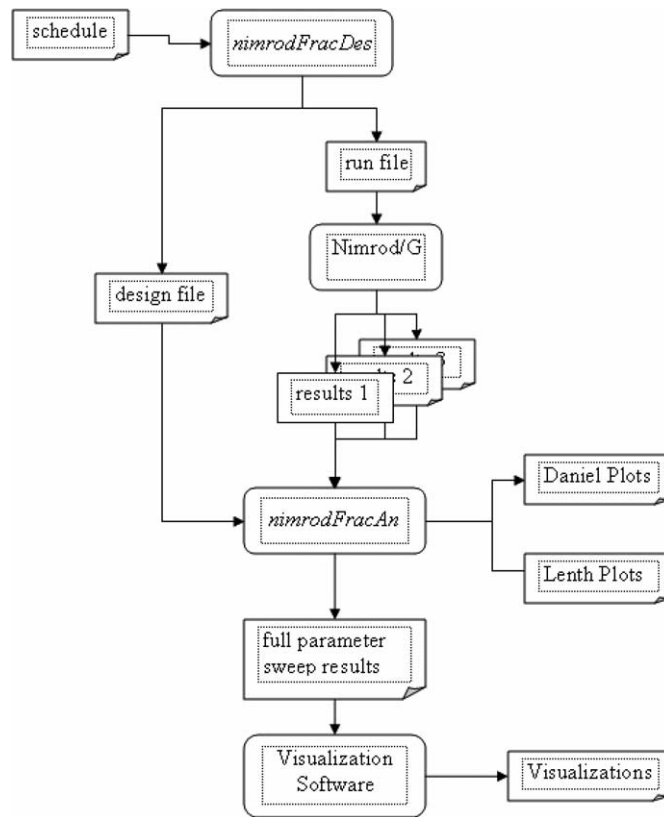


Fig. 3. Workflow for a Nimrod/E fractional factorial experiment.

application also uses these effect values to generate the results of a full factorial experiment. It thus fills in the results for the runs that have not been performed. The results are then in a form suitable for visualization software such as OpenDX [23]. However, we have not incorporated a visualization component in the present system.

5.1. The experiment schedule

The experimenter's input may be confined to the preparation of a "schedule" file. Such a file contains four sections, as shown in the example of Fig. 4. The first two sections are the standard format used in Nimrod plan files as described above. The following "design" section specifies either the resolution of the design, or else directly specifies the primary and secondary effects. Here the experiment will have resolution V.

The final (optional) section controls the analysis to be done after the fractional factorial runs are complete. For this experiment there will be three separate analyses performed, corresponding to three different output

values found in the output file. Nimrod/E has a record of which values for \$jobname correspond to which runs and hence can compute the estimates of the effects for each of the three results.

5.2. Creation of the factorial design

Computer experiments often involve many factors, more than traditionally handled in physical experiments. Published tables of experimental designs do not extend to more than 20 factors for resolution V. There are several published algorithms for fractional factorial design [14,15,18]. Of these, the only *general purpose* one that can handle many factors is the SEF algorithm of Liao and Iyer [18], which can cope with 80 factors at two levels each, for a resolution V design. This may seem sufficient for practical purposes. However, if more than two levels are required the number of factors is significantly reduced. For example, with eight levels per factor only 14 factors can be handled. Consequently, we have developed a variant of that algorithm, called RSSEF [26], which can handle 130 two-level factors or 18 eight-level factors.

```

parameter a float range from -1 to 1 points 2
parameter b float range from -1 to 1 points 2
parameter c float range from -1 to 1 points 2
parameter d float range from -1 to 1 points 2
parameter e float range from -1 to 1 points 2

task main
  copy testFunction node:testFunction
  node:execute ./testFunction $a $b $c $d $e > results
  copy node:results results.$jobname
endtask

design
  resolution 5
enddesign

results "NoNoise" in file "results" line 1 field 1
results "Noise30%" in file "results" line 2 field 1
results "Noise100%" in file "results" line 3 field 1

```

Fig. 4. Schedule file for Experiment 1.

Fig. 5. Screenshot of Nimrod portal experimental design planning page.

5.3. The Nimrod/E portal interface

Access to Nimrod/E has been incorporated into the Nimrod Portal. If an experiment is specified as Experimental Design then the page for editing a schedule contains extra functionality for specifying the design and the results for analysis. Figure 5 shows this page, with the four sections corresponding to the four parts of an experiment schedule. In the “Design Spec-

ification” section the user may select the resolution required for the experiment. Alternatively each main effect and each two-way effect may be individually specified as either primary (P) secondary (S) or other (blank) thus allowing finer control of the design. There is an alternative text mode interface that permits specification where a list of primary and secondary effects is typed. This mode permits three-way and higher interaction effects.

The final section “Results for Analysis” is used to specify where, among the results of the runs, the numbers for analysis will be found. In this case each run is producing a file called “output” where these numbers are written. Two analyses will be performed; one on the first item in the first line, and another on the first item in the second line.

When this design is saved, the Portal will use *nimrodFracDes* to compute the fractional factorial design and the Nimrod/G run file. If the user has also selected computational resources then no further user input is required. The Portal will launch Nimrod/G to perform the runs and, when the runs are complete, will apply *nimrodFracAn* to produce the analyses.

6. Case studies

This section illustrates the use of Nimrod/E and reports the results of four example experiments. The first two use known output functions and are included to demonstrate the principles of experimental design and analysis. The third experiment uses a computational model of heart muscle cells; the analysis was used to gain insight into the nature of the objective calculated. The final experiment is a re-analysis of data obtained from an earlier large computational chemistry experiment and demonstrates another use for Nimrod/E.

6.1. Experiment 1: A known quadratic response

This section illustrates the use of fractional factorial design using a simple model where the response function is known. This function is

$$\begin{aligned} \phi(A, B, C, D, E) \\ = 2A - B + 0.1D + AB \\ - 0.2AE + \sin \pi AC + kN, \end{aligned} \quad (6.1)$$

where N is a random number selected from a rectangular distribution over an interval $[0, 0.01]$, and k will be chosen later. We use two levels, -1 and $+1$, for each parameter A, B, C, D, E .

In many computational models the input parameters vary in importance, with some making negligible contributions to the results. The response function (6.1) reflects this with D having only a small effect; E has a small effect also, but only as an interaction with A . Although C is significant within the function, it appears only in the sine term which is zero for the levels used,

so it has no effect in this experiment. This term is chosen to illustrate the influence of the chosen levels.

The random number term, kN , has been included to illustrate this effect of noise on estimates of the effects. We perform experiments with various values of k , namely 0, 0.3 and 1.0, to vary the noise component. If $k = 0$, then the function is a linear combination of single parameters and two-combination products, so we should expect that a resolution V experiment will give an exact reconstruction of the full factorial experiment.

Figure 4 shows the schedule used for this experiment, prepared by interaction with the Nimrod Portal. Nimrod/E used this schedule to compute a resolution V design, which generated 16 runs as compared to $2^5 = 32$ for a full parameter sweep. The results are presented in graphical form in the Lenth plots of Fig. 6.

With no noise the effect estimates exactly equal the coefficients of formula (6.1), so that the results for the 16 jobs are sufficient to reconstruct all 32 results for these levels. With 30% noise the small effect of the 0.1D term is obscured; the D effect is in the zone of uncertainty between the ME and SME lines. With 100% noise both the AE and D effects are within the ME lines and so cannot be deemed as active.

The alternative presentation of effects using Daniel plots is shown in Fig. 7. To save space only the 100% noise case is shown. The points representing the effects A, B , and AB clearly deviate from the straight line and may thus be deemed significant.

This experiment illustrates item 1 in Section 3.2, how fractional factorial design can identify the relative importance of factors and factor interactions, and how these become obscured as the amount of noise increases.

6.2. Experiment 2: A more complex known response

The next experiment explores the effect of the choice of resolution, again using a model with a known output function. In this experiment we concentrate on the reconstruction of the full factorial experiment from the results of fractional factorial ones, item 3.2.6 in our list of applications.

In this example, the output function $\phi(A_1, A_2, A_3, A_4, A_5, A_6)$ is defined recursively by $\phi_1 = A_1$, $\phi_i = \sqrt{100.0 + \phi_{i-1} + iA_i}$ for $i = 2, 3, \dots, 6$ and $\phi = \phi_6$. We use two levels, 0 and 1, for each input factor. Nimrod/E was used to design three fractional-factorial experiments, with Resolution III, IV, and V, respectively. In each case the estimates of the effects were

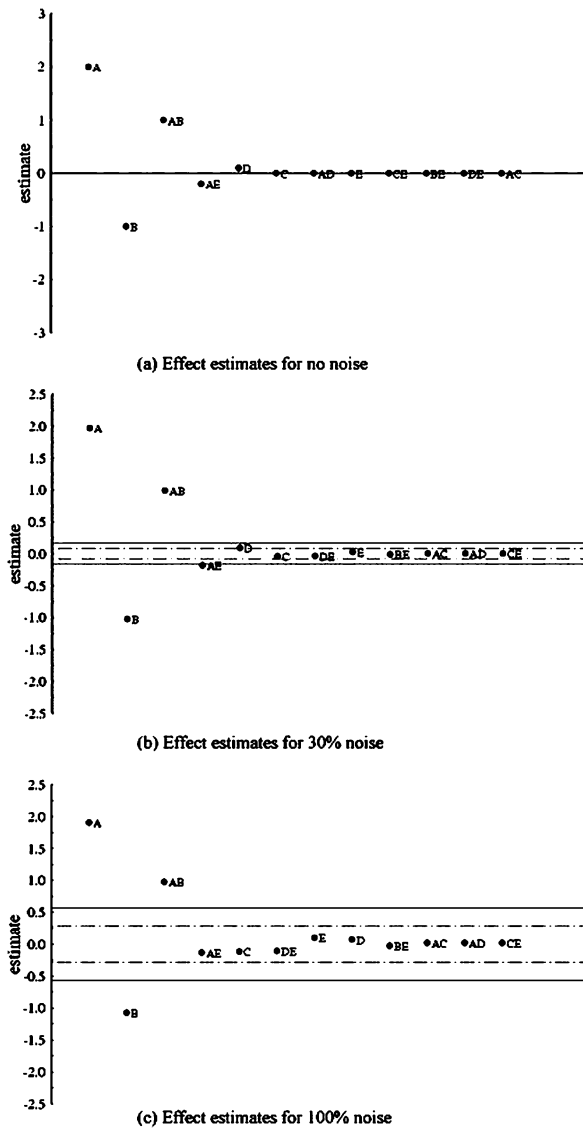


Fig. 6. Lenth plots for Experiment 1.

used to construct results for a full parameter sweep. To assess the discrepancy between these estimated values E_i and the genuine results of a full parameter sweep, R_i , the usual statistic $SSD = \sum(E_i - R_i)^2$ was computed. As this statistic fails to indicate the error relative to the true values, a normalized statistic $SSND = \sum[(E_i - R_i)/R_i]^2$ was also computed. Table 1 shows the results for all of these experiments.

Resolution V produced results very close to the true values, indicating that the effects of interactions of more than two factors are small. The results illustrate the trade-off between the number of runs and the accuracy of the predictions; as the resolution is decreased,

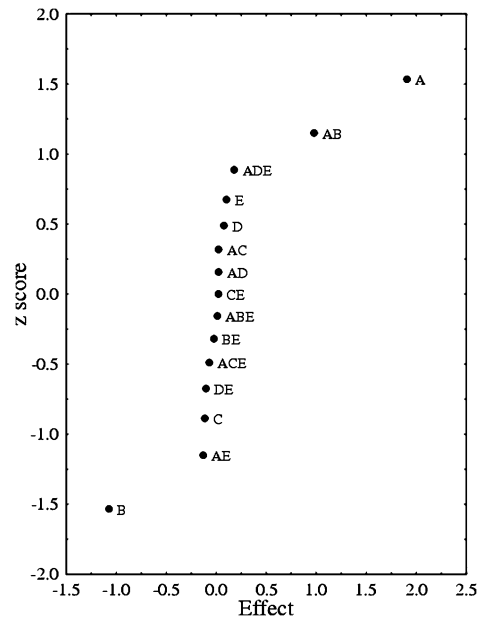


Fig. 7. Daniel plot for Experiment 1 for 100% noise data.

Table 1

Reconstruction of a full sweep from fractional-factorial Experiment 2

Resolution	No. runs	SSD	SSND
III	8	1.26939	0.0111786
IV	16	3.89334e-3	3.43098e-05
V	32	9.46657e-12	8.31543e-14

the number of runs reduces but the accuracy of the predictions also falls, but is still high. The close agreement between predictions and true values demonstrates a major justification for the fractional factorial approach.

6.3. Experiment 3: Ionic flux in a cardiac model

This experiment concerns a mathematical model of excitation-contraction coupling in the rabbit ventricular muscle cells [30]. In the model a system of ordinary differential equations is used to describe the intracellular ionic fluxes (for Ca^{2+} and Na^+). The model incorporates many features of cell structure and is able to reproduce experimental data using parameters based on measurements in rabbit cells. In this experiment we extended the model to incorporate Mg^{2+} -nucleotide regulation [21]. The experiment involves validation of the updated model by varying nine input parameters (see legend in Fig. 8) so that the predicted I_{NaK} , I_{Ca} , I_{pCa} and I_{SERCA} currents are as close as possible to

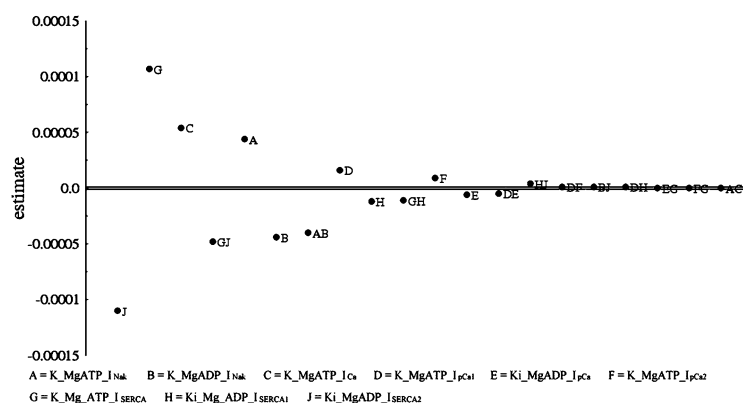


Fig. 8. Lenth plot for Experiment 3.

the experimentally measured. This was phrased as an optimization problem using Nimrod/O, minimizing the sum of the moduli of the differences between computed metabolic factors (I_{NaK} , I_{Ca} , I_{pCa} , I_{SERCA}) and experimental values.

The code is implemented in Matlab, and each computation takes about 10 minutes on a high end workstation. Prior to running Nimrod/O, we decided that a preliminary parameter sweep with Nimrod/G would indicate the broad features of the objective landscape and this might lead to more efficient searches in Nimrod/O. However, with nine parameters a full sweep becomes a large undertaking, so we performed a Nimrod/E experiment instead.

Here we denote the metabolic factor inputs as A, B, \dots, G, H and J (by convention I is not used to denote factors in experimental design). Two values were chosen for each of these factors which were judged from experience to span the realistic range. A resolution V experiment was designed and executed.

Figure 8 shows the Lenth plot for the most significant effects on the basis of the resolution V experiment. Clearly, all nine factors had a significant effect. Of the two-way interactions, the significant ones are AB, DE, DF, GJ, GH and HJ . Other combinations cannot be distinguished from noise. This prompted a closer look at the model which explained this result. The factors fall into four groups $\{A, B\}$, $\{C\}$, $\{D, E, F\}$ and $\{G, H, J\}$. It turned out that only the first group affects the first term in the objective, only the second affects the second term, and so on. Thus we would expect no interactions between factors in different groups. Whilst the result was obvious with hindsight, the work demonstrated the power of a tool like Nimrod/E – we had not expected this result, and were treating the computational model as a black box. No-

tably, Nimrod/E’s analysis discovered the lack of coupling between certain factors without detailed knowledge of the code. The reasons for this lack of coupling only became obvious to us once we looked *inside* the computation.

The result also had significant implications for the Nimrod/O run that was to follow. As discussed in item 3.2.5, a more efficient search was possible by separately optimizing the four components of the output.

6.4. Experiment 4: A quantum model for complex molecules

This experiment concerns quantum chemical models – based on the Schrödinger equation – of assemblies of atoms as found in large biomolecules. The problem with using a pure quantum mechanical approach is that these methods are extremely computationally expensive, and thus can normally only be applied to relatively small molecules. Hybrid quantum mechanics–molecular mechanics (QM/MM) models, on the other hand, just describe a small, “active” region by accurate quantum techniques, while the surrounding larger, “inactive” region is treated with more approximate classical force fields. As a result, they tend to be faster than a full quantum chemistry computation. Unfortunately, the two involved physical concepts are so different that they cannot be easily combined. One approach to solve this combination is to saturate each of the quantum chemical bonds between both regions with a single parameterized “capping atom”, instead of with the bulk of the large molecule. This capping atom is not real, but can be adjusted by a hypothetical pseudopotential function [35], as shown in Fig. 9.

This experiment explores such a capping potential for a carbon–carbon single bond, using the “group dif-

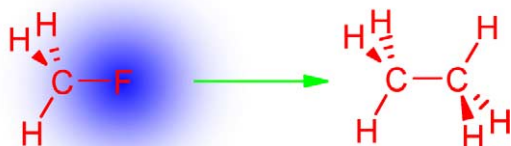


Fig. 9. Replacement of part of the structure with a capping atom (F) and a pseudopotential.

ference potential” (GDP) approach of Sudholt et al. [31–33]. In this model, ethane ($\text{CH}_3\text{--CH}_3$) is used for the parameterization. One of its methyl groups (CH_3) is substituted by an isoelectronic fluorine atom (F) and a pseudopotential of the form

$$U_{\text{eff}}(r) = A \exp(-Cr^2) + B \exp(-Dr^2), \quad (6.2)$$

where r is the radius. Here parameters A, B, C, D are to be determined that will provide the best fit between the model and target properties of ethane and the methyl radical. This is assessed by a normalized least squares expression. The quantum chemistry code GAMESS [29] was used to compute the model properties used in this comparison, each task taking about 20 minutes of computation time.

The experiment used the results of the very large sweep over the A, B, C, D parameter space described in [31]. The results showed a huge variation in the size of the output results, from ~ 500 to over 10^{11} , so we decided to analyse the logarithms of these values rather than the direct values.

With four factors, a resolution V fractional factorial design is just a full sweep. A resolution IV design does give a reduction of 50% in the number of runs but biases the estimates of effects. We used Nimrod/E to perform a resolution IV experiment with two levels per parameter. Figure 10 is a Daniel plot for the results. It shows that the effect largest in absolute size is the four-factor interaction $ABCD$, which can be understood from the mathematical form of the pseudopotential. This indicates that the usual result, that higher order effects are smaller than lower order ones, fails in this case. So a reconstruction of the full sweep using results of the resolution IV experiment cannot be expected to give accurate results here. This case study illustrates the robust nature of the methodology which indicates when a full sweep is required. Nothing is lost by performing a fractional factorial experiment first as the full sweep can be designed to perform only the missing runs.

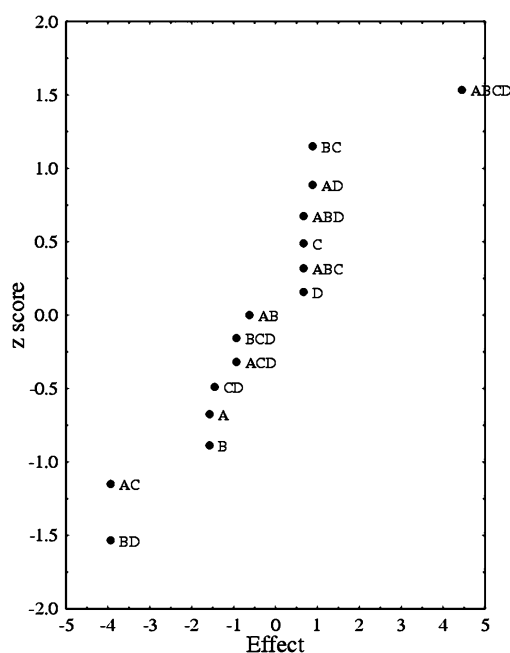


Fig. 10. Daniel plot for Experiment 4.

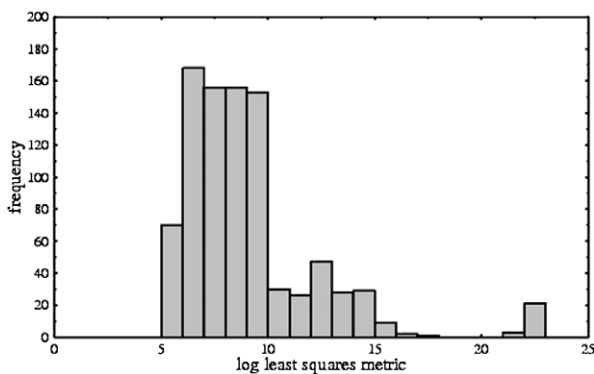


Fig. 11. Histogram of results for Experiment 4.

There was however one application of fractional factorial analysis that was useful with this data. We decided to use a design with more than two levels per factor. In the original data the number of levels were not powers of two, so we extracted a subset using 8 levels for each factor A and B , and 4 levels for each of C and D . This produced a data set of 1024 values. A histogram of these values, Fig. 11, revealed a gap in the distribution. A small proportion of results were very large, over 21, with nothing in the range 18–21. This gap was not evident in the full dataset which showed a contiguous distribution. This naturally raised the question as to whether the outlying results were valid, or a symptom of a failure of the model. To explore this we

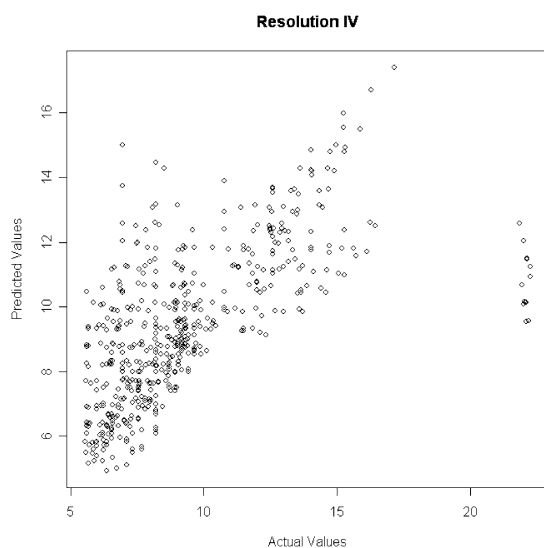


Fig. 12. Correlation between predicted and actual values for Experiment 4.

used reconstruction from a fractional experiment as a smoothing filter on the data, as discussed in Section 3.2 item 7.

Nimrod/E was used to design a resolution IV experiment, run the experiment, and produce predicted values. In Fig. 12 the predicted values are graphed against the actual values on a scatter plot. The correlation is not high, but sufficient to indicate that all the outliers, the points on the right side, are aberrant, they reveal a discontinuity in the response function. A likely explanation is failures in the GAMESS calculations that were not flagged to the output. This example illustrates the application of fractional factorial design to assess the validity of data produced by computational experiments. Here we see the cyclical nature of experimentation, results feeding back into the model which may then be adjusted and produce the next round of results.

7. Conclusions and further work

This paper illustrates the use of fractional factorial methods in design of computer experiments. Fractional factorial designs typically offer a large saving in the number of jobs required to explore a parameter space, but also provide a robust way of indicating when a full sweep might be required. Furthermore they provide tools for the experimenter to assess the relative importance of the various parameters and their

interactions and offer insights into the computational model.

To facilitate these methods we have developed a tool, Nimrod/E, which automates the design of the experiment, interfaces with Nimrod/G for the execution of the experiment, and then provides tools for the final analysis. A complete project may be controlled from the Nimrod Portal. This offers a considerable advantage over previous approaches that require operator intervention to supply a design appropriate to the situation.

We have illustrated the effectiveness of Nimrod/E with four case studies. Two of these were hypothetical examples used to demonstrate the basic functions, but two were real world scientific experiments. Importantly, Nimrod/E produced useful results in both of the real world studies. In the cardiac modelling work, Nimrod/E detected the structure of the underlying computational model without any specific details, and allowed us to make an improvement in the overall performance of a subsequent optimization run. In the quantum chemistry work we showed the necessity of a complete parameter sweep and that it was not possible to perform a simpler and cheaper experiment than the one we did. Further, a comparison of the fractional factorial predictions with results of the full sweep was used to investigate questions raised by some of the model outputs.

We see the implementation of fractional factorial designs as the first stage of a continuing project. Other experimental designs offer advantages, for example: empirical kriging methodology, Latin hypercube designs and maximum entropy designs [11]. We hope to include them in the final suite of Nimrod/E tools.

Acknowledgements

Nimrod/E was developed under funding provided by the Australian Research Council as part of the Australian Centre for Complex Systems (ACCS). The cardiac modelling work was supported in part by the U. S. National Science Foundation, the California Institute for Telecommunication and Information Technology, and the National Biomedical Computation Resource. The quantum chemistry application was supported in part by the German Academic Exchange Service (DAAD) and the University of California, San Diego.

References

- [1] D.A. Abramson, M. Cope and R. McKenzie, Modelling photochemical pollution using parallel and distributed computing platforms, in: *Proceedings of PARLE-94*, Athens, Greece, 1994, pp. 478–489.
- [2] D. Abramson, J. Giddy and L. Kotler, High performance parametric modeling with Nimrod/G: Killer application for the global grid?, in: *International Parallel and Distributed Processing Symposium (IPDPS)*, May 2000.
- [3] D. Abramson, P. Roe, L. Kotler and D. Mather, ActiveSheets: Super-Computing with Spreadsheets, *2001 High Performance Computing Symposium (HPC'01), Advanced Simulation Technologies Conference*, Seattle, USA, April 22–26, 2001, pp. 110–115.
- [4] Axceleon, <http://www.axceleon.com/>, accessed 20 August 2007.
- [5] G.E. Box, W.G. Hunter and J.S. Hunter, *Statistics for Experimenters*, Wiley, 1978.
- [6] G.E. Box, Statistics and quality improvement, *Journal of the Royal Statistical Society A (Statistics in Society)* **157** (1994) 209–229.
- [7] Collaborative Projects using Nimrod Tools, <http://message.infotech.monash.edu.au/Nimrod/Projects>, accessed 20 August 2007.
- [8] R.P. Brent, *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, 1973.
- [9] C. Daniel, *Applications of Statistics to Industrial Experimentation*, Wiley, 1976.
- [10] N. Draper and H. Smith, *Applied Regression Analysis*, 2nd edition, Wiley, 1981.
- [11] K.-T. Fang, R. Li and A. Sudjianto, *Design and Modelling for Computer Experiments*, Chapman and Hall/CRC, 2006.
- [12] R.A. Fisher, *Statistical Methods for Research Workers*, Oliver and Boyd, 1925.
- [13] R.A. Fisher, *The Design of Experiments*, Oliver and Boyd, 1935.
- [14] M.F. Franklin and R.A. Bailey, Selection of defining contrasts and confounding effects in two-level experiments, *Applied Statistics* **26** (1977), 321–326.
- [15] A. Fries and W. G. Hunter, Minimum aberration 2^{k-p} designs, *Technometrics* **22** (1980), 601–608.
- [16] R. Jones, D. Peng, P. Chaperon, M. Tan, D. Abramson and T. Peachey, Structural optimization with damage tolerance constraints, *Journal of Theoretical and Applied Fracture Mechanics* **43** (2005), 133–155.
- [17] R.V. Lenth, Quick and easy analysis of unreplicated factorials, *Technometrics* **31** (1989), 469–473.
- [18] C.T. Liao and H.K. Iyer, A stochastic algorithm for selecting [sic] of defining contrasts in two-level experiments, *Biometrical Journal* **6** (1999), 671–678.
- [19] The London e-science centre, <http://www.lesc.ic.ac.uk/support/projects.html>, accessed 26 July 2007.
- [20] D. Mather and N. Crofts, A computer model of the spread of hepatitis C virus among injecting drug users, *European Journal of Epidemiology* **15** (1999), 5–10.
- [21] A. Michailova, W. Lorentz and A. McCulloch, Modelling transmural heterogeneity of KATP current in rabbit ventricular myocytes, *AJP-Cell Physiology* **293** (2007), C542–C557.
- [22] Nimrod Portal, <http://www.csse.monash.edu.au/~davidan/nimrod/portal.htm>, accessed 20 August 2007.
- [23] OpenDX, <http://www.opendx.org/about.html>, accessed 26 July 2007.
- [24] A.B. Owen, Latin supercube sampling for very high-dimensional simulations, *ACM Transactions on Modeling and Computer Simulation* **8** (1998), 71–102.
- [25] T. Peachey, D. Abramson and A. Lewis, Model optimization and parameter estimation with Nimrod/O, in: *International Conference on Computational Science*, University of Reading, UK, May 28–31, 2006.
- [26] T.C. Peachey, N.T. Diamond, D. Abramson and C.M. Enticott, Computing factorial designs for large numbers of factors with 2^m levels, submitted for publication.
- [27] P. Saha and L.L.R. Williams, Gravitational lensing model degeneracies: Is steepness all-important?, *The Astrophysical Journal* **653** (2006), 936–941.
- [28] T. Santner, B. Williams and W. Notz, *The Design and Analysis of Computer Experiments*, Springer-Verlag, New York, 2003.
- [29] M.W. Schmidt, K.K. Baldrige, J.A. Boatz, S.T. Elbert, M.S. Gordon, J.H. Jensen, S. Koseki, N. Matsunaga, K.A. Nguyen, S. Su, T.L. Windus, M. Dupuis and J.A. Montgomery, General atomic and molecular electronic structure system, *J. Comput. Chem.* **14** (1993), 1347–1363; <http://www.msg.ameslab.gov/GAMESS/>, accessed September 20, 2007.
- [30] T.R. Shannon, F. Wang, J. Puglisi, C. Weber and D.M. Bers, A mathematical treatment of integrated Ca dynamics within the ventricular myocyte, *Biophysical Journal* **87** (2004), 351–371.
- [31] W. Sudholt, K.K. Baldrige, D. Abramson, C. Enticott and S. Garic, Applying grid computing to the parameter sweep of a group difference pseudopotential, in: *Computational Science – ICCS 2004: 4th International Conference*, Kraków, Poland, June 6–9, 2004, Lecture Notes in Computer Science, Vol. 3036, 2004, pp. 148–155.
- [32] W. Sudholt, K.K. Baldrige, D. Abramson, C. Enticott and S. Garic, Parameter scan of an effective group difference pseudopotential using grid computing, *New Generation Computing* **22** (2004), 125–136.
- [33] W. Sudholt, K. Baldrige, D. Abramson, C. Enticott and S. Garic, Application of grid computing to parameter sweeps and optimizations in molecular modeling, *Future Generation Computer Systems* **21** (2005), 27–35.
- [34] B. Vanderstraeten, P.W. Chin, M. Fix, A. Leal, G. Mora, N. Reynaert, J. Seco, M. Soukup, E. Spezi, W. De Neve and H. Thierens, Conversion of CT numbers into tissue parameters for Monte Carlo dose calculations: a multi-centre study, *Physics in Medicine and Biology* **52** (2007), 539–562.
- [35] Y. Zhang, T.-S. Lee and W. Yang, A pseudobond approach to combining quantum mechanical and molecular mechanical methods, *J. Chem. Phys.* **110** (1999), 46–54.




Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

