C.1 Project Description

PROJECT TITLE

Live parameter-space exploration of simulations with uncertainty

AIMS AND BACKGROUND

When environmental systems collapse during disasters, for example flood, fire, wind, earthquake and warfare, information from computational simulation is typically needed urgently and in an acceptable format to help decision makers. Meeting such a modelling challenge is not simply a matter of compute power but the application of ideas from software engineering, computational mathematics, optimised distributed systems and data visualisation. Furthermore, all of these disciplines need to be tightly coupled and benchmarked to realistic human-in-the-loop scenarios. This project addresses this challenge through a live-programming approach to system development and usability, with a focus on understanding and quantifying uncertainty at all stages of the simulation.

To do this, we will build immersive virtual "control room" environments backed by live (run-time modifiable) distributed computing infrastructure for *interactively* exploring model input-output relationships. Using these environments, decisionmakers will be able to examine many different scenarios in a short space of time, even with computationally expensive models. This will enable them to develop an intuitive understanding of the uncertainty relationships in the system—from uncertainties in input data through to representations of uncertainty in model outputs. The importance of this human-in-the-loop workflow is summarised by **pike˙science˙2009** "it is through the interactive manipulation of a visual interface—the analytic discourse—that knowledge is constructed, tested, refined and shared."

Although the mathematical and computational tools we will employ in this project are generalisable to any modelling workflow, we will focus on application domains where (a) the model is (computationally) complex, (b) uncertainty is significant and (c) decisions are time-sensitive. Specifically, we will build interactive interfaces for the following scenarios: storm surge modelling, bushfire modelling and flood modelling. By working with domain experts in these fields and building new computational tools and immersive visualisation environments, we will gather concrete qualitative data on the usefulness of a dynamic, interactive, exploratory modelling workflow.

Computational decision support systems (DSS) for disaster management have existed for many decades (**wallace˙decision˙1985**), and recent advances have been made both in incorporating uncertainty (**thompson˙social˙2014-1**; **neale˙navigating˙2015**) and providing real-time output (**yu˙support˙2006**). In these (and other) contexts, immersive environments, with multiple large screens, rich audiovisual feedback and multi-modal interaction, have been used to aid exploration and understanding of simulation results (**colella˙participatory˙2000**; **lui˙supporting˙2014**).

However, existing systems generally only handle "known unknowns", relying on specific apriori knowledge of the nature and characteristics of the uncertainty, and a way to encode and propagate this uncertainty through the model. This approach has some merit, but often falls short in the real world, where the occurrence of "unknown unknowns" may require unforeseen constraints or new knowledge to be incorporated into the model in real-time.

We contend that this challenge can be effectively tackled by deploying recent developments in programming language systems, known as "live programming" (LP), into the context of scientific modelling and simulation. Live programming is a term which has been used to denote systems which support the direct intervention of the programmer in a program's run-time state. It can be thought of as an extreme version of just-in-time (JIT) programming where there is a direct correlation between a program's representation and its execution. An example of this is the visual language Self (**ungar˙self˙1987**) where each visual element in the language is a directly programmable object. As the ambition of LP systems has grown, these systems and languages themselves have evolved to have access to clock-time and other computer-system properties at a language level in order to, for example, create, modify and interact with computer music and devices in real time. Examples of the latter include the Impromptu (**sorensen˙impromptu˙2005**) and Extempore (**sorensen˙extempore**) languages which one of the authors of this proposal has been working on and off with over several years.

In addition to this live programming approach, we will use sparse grid and reduced basis methods to allow sophisticated approximations to complex models with high-dimensional parameter spaces. The use of such methods will allow us to perform parameter space exploration more efficiently, resulting in an interactive feedback loop even with complex and computationally expensive models. The live programming tools provide the *capability* to modify the simulation on-the-fly, whilst sparse grids and reduced basis models shorten the feedback latency to make this capability *useful*.

This project will provide the following discoveries and benefits to the international research community:

1. a suite of interactive, real-time modelling tools which combine full simulations with coarse-grid, rough simulations to quantify uncertainty, optimised for human exploration
2. interactive information visualisations of simulation predictions together with uncertainties
3. an agile software engineering approach to the further development and optimisation of these software systems

This research is significant because it aims to unlock the power of large scale computational simulation for interactive use. Although we concentrate our research on case studies in disaster response, the ultimate potential of this work is to eventually empower domain experts from a broad range of areas to use the high-performance computing power which is now available to them. We envision a future where performing a complex flood model or disaster simulation is as interactive and *alive* as flicking through photos on an iPad, rather than a tedious "queue up the job, come back a week later to pore over a dump of the results" process. This benefits not only experts already working in that field, but also allows new stakeholders (or even schoolchildren) to understand the behaviour of such systems through interactive play rather than dry textbooks.

TODO mention FOR

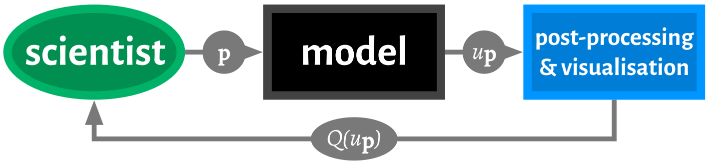

Figure 1: The human-in-the-loop modelling workflow. A scientist selects an initial parameter $\mathbf{p}_0$ for their model, examines the model output $Q(u_{\mathbf{p}})$, and either accepts the output of the model or re-runs the model with a different choice of the parameter $\mathbf{p}_1$.

RESEARCH PROJECT

The January 2011 Brisbane River floods in south-east Queensland cost 32 lives and caused 2.5 billion dollars worth of damage (**vandenhonert˙2011˙2011**). In the days leading up to these events, a key issue facing authorities was incorporating their **uncertainty** about the preceding fortnight's rainfall into the modelling. In their report on the causes, impacts and implications of the floods, **vandenhonert˙2011˙2011** conclude:

> whilst the dam operators were acting in accordance with the operations manual for the dam, their modelling did not take account of forecast rainfall in determining the predicted dam water level, and this resulted in a sub-optimal water release strategy. Employing tools for decision making under uncertainty would have resulted in a different water release strategy.

At the other end of the environmental spectrum, bushfire model predictions are similarly "fraught with uncertainty" (**alexander˙limitations˙2013** p375). As Australia's climate changes and extreme weather events become more common, there is a significant need for better ways to glean timely insights from modelling in the presence of uncertainty.

Computational modelling/simulation is an invaluable tool in these scenarios, allowing both domain scientists and other domain experts such as emergency services personnel to explore what might happen under various scenarios. Some of the inputs to the model may be well known, while others are known only approximately, and others still may only be guessed at.

More formally, we have a mathematical model $M_{\mathbf{p}}$ parameterised by $\mathbf{p} \in \mathcal{P}$. For example, $M_{\mathbf{p}}$ may be a parameterised partial differential equation (PPDE) in a storm surge model. For each $\mathbf{p}$ we suppose the model is well-defined and there exists a unique function

$$u_{\mathbf{p}}(\mathbf{x}) \quad \mathbf{x} \in \Omega \tag{1}$$

which is a solution to the model problem, that is $M_{\mathbf{p}}(u_{\mathbf{p}}) = 0$.

The goal of the scientist is to better understand the relationship between $\mathbf{p}$ and some lower-dimensional quantity of interest $Q(u_{\mathbf{p}})$, or to find the parameter choice $\mathbf{p}$ which optimises Q. This high-level description of the "model selection/optimisation" problem glosses over some nuances, but this general workflow (shown graphically in Figure 1) lies at the heart of a great deal of modern science.

There are many ways of finding the optimal $\mathbf{p}$, from trial and error or expert judgements through to fully automated algorithmic optimization procedures. Often there are ways to optimise $\mathbf{p}$ algorithmically, although this usually introduces new parameters (the arguments of

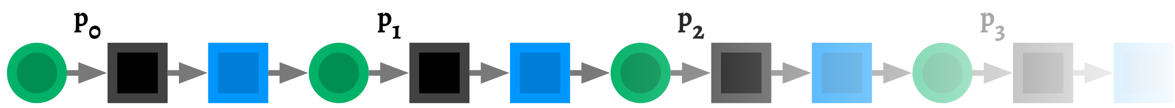

Figure 2: If the modelling & post-processing/visualisation steps can be performed sufficiently quickly, then the scientist can explore the $\mathbf{p} \to Q(u_\mathbf{p})$ relationship *interactively*, with the all the associated benefits for exploratory analysis.

the function being optimised) which must be selected by the scientist. As a result, this feedback loop will often require many iterations, with a human scientist in-the-loop, evaluating the results of the model (possibly through visualising the model output) and choosing a parameter update $\Delta\mathbf{p}$ at each step (see Figure 2). Each step through this loop provides feedback to the scientist about the response of the system to a particular value of $\mathbf{p}$, for example the maximum storm surge level under a particular rainfall scenario. Through the process of trying different model parameterisations, the scientist is able to build up an understanding of the *general* relationship between $\mathbf{p}$ and $Q(u_\mathbf{p})$, including the areas of the parameter space $\mathcal{P}$ which have the greatest influence on the result. This is especially important if the model inputs are not known with certainty—since the scientist is better able to know which types of uncertainty have the greatest impact on the certainty of the results.

From a workflow perspective, the productivity of the scientist is proportional to the rate at which they can explore the $\mathbf{p} \to Q(u_\mathbf{p})$ relationship. Any latency improvements in this feedback loop will translate into productivity gains (**liu˙effects˙2014**).

If the model parameters $\mathbf{p}$ and inputs $\mathbf{x}$ are known precisely and the quantity $Q(u_\mathbf{p})$ is cheap to calculate and easy to interpret, then the task is simple: provide the scientist with an interface for manipulating $\mathbf{p}$ and set them loose. However, for real-world models (such as those used in flood/storm surge/bushfire modelling) this is often not the case, there are **three primary challenges**:

1. *The model may not provide a way to express uncertainty in the inputs.* Many models do not provide methods for including uncertainty information in their inputs, as was the case in the Lockyer valley example.

2. *The quantity $Q(u_\mathbf{p})$ may not be cheap to calculate* (as shown in Figure 3). Many sophisticated models require non-trivial computing resources (e.g. supercomputers) to evaluate. These compute resources may be difficult to secure, with jobs having to wait in a queue, and may take a long time to compute even when the resources are available. This is especially problematic in a disaster response scenario, where an approximately correct answer provided in a short time is significantly more useful than a perfect answer provided after it is too late to act on.

3. *The quantity $Q(u_\mathbf{p})$ may not be easy to interpret.* This may be because of technical reasons, such as a complex problem domain where coming up with a meaningful loss function is difficult, or may be due to ethical reasons—how to balance the predicted cost to private property vs damage to the natural environment. Finally, this is a visualisation

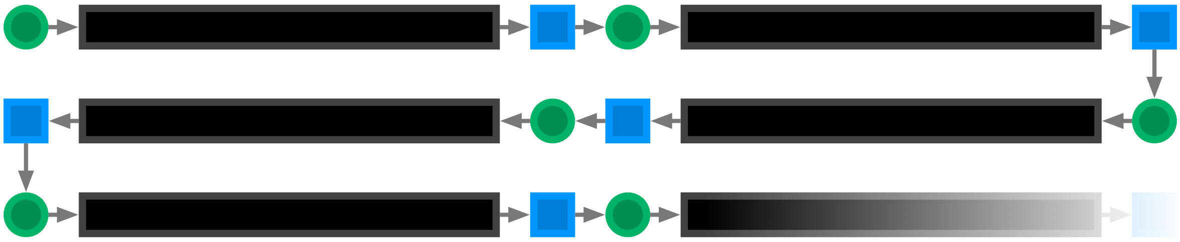

Figure 3: If the model is computationally expensive to run, then the workflow is dominated by waiting for the model to finish. This results in lower productivity—not only due to the time spent waiting for the model, but also because of the temporal separation between the selection of a new parameter $\mathbf{p}$, and seeing its impact on the results of the model.

problem— the mapping $\mathbf{p} \to Q(u_{\mathbf{p}})$ may be high-dimensional, and presenting that to the scientist (especially with uncertainty information) may not be straightforward.

By accelerating the feedback loop between $\mathbf{p}$ and $Q(u_{\mathbf{p}})$, including estimates of uncertainty, we will give the scientist the ability to interactively *explore* the connection (and the associated uncertainty) between the different dimensions of $\mathbf{p}$ and the overall response of the system. Ultimately, the scientist needs an **interactive interface** for gleaning insights from their models in the presence of these challenges.

Our conceptual framework for dealing with these three challenges involves using sparse grids and reduced basis models combined with live modification of simulation parameters, dynamic computational resources and visualisation. This approach allows us to deal with the three primary challenges mentioned above.

1. *The model may not provide a way to express uncertainty in the inputs*. Using sparse grids and reduced basis models to precompute a surrogate model, existing models (which may not provide a way to encode uncertainty information) can be used as "black boxes", and expectation integrals over subsets of the parameter space $\mathcal{P}$ (which are important for quantifying uncertainty) can be efficiently be estimated from the surrogate model $\tilde{u}_{\mathbf{p}}(\mathbf{x})$. This has significant benefits over classical Monte Carlo methods when the integrand is sufficiently smooth (**JakemanRoberts2013**; **FranzelinDiehlPfluger2014**).

2. *The quantity $Q(u_{\mathbf{p}})$ may not be cheap to calculate*. For many problems the computation of solutions $u_{\mathbf{p}}(\mathbf{x})$ to the model $M_{\mathbf{p}}$ can be done cheaply using the combination technique over the domain $\mathbf{x} \in \Omega$, as shown in Figure 4. One may also develop reduced basis models to achieve the same outcome.

3. *The quantity $Q(u_{\mathbf{p}})$ may not be easy to interpret*. Since a sparse grid sampling of $\mathbf{p} \in \mathcal{P}$ enables a fast and efficient exploration of parameter space, there is more time for visualisation and post-processing in an interactive interface, which allowing richer ensembles of visualisation techniques to assist the scientist in interpreting the results of the model.

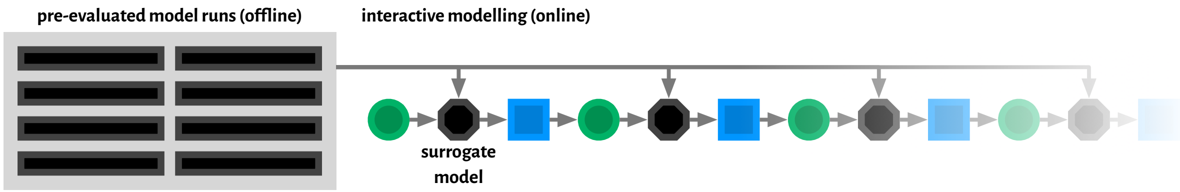

Figure 4: Using the sparse grids and reduced basis models, the computationally expensive model calculations can be done ahead-of-time and used to construct a surrogate model which can be used to re-claim the interactive workflow of Figure 2
.

Impact: new algorithms, software, high performance computer systems and visualisation techniques for time-bound environmental simulation with uncertainty. New software tools for the simulation of flood surges and tsunami. New methodologies for rapid and agile software development and usability using live programming. New knowledge of human-in-the-loop requirements for support systems in the context of environmental disaster management. New algorithms, software, high performance computer systems and visualisation techniques for time-bound environmental simulation with uncertainty. New software tools for the simulation of flood surges and tsunami. New methodologies for rapid and agile software development and usability using live programming. New knowledge of human-in-the-loop requirements for support systems in the context of environmental disaster management.

The mathematical component of this project will be based on new mathematical developments combining sparse grids and uncertainty quantification. One of the main difficulties in current models is the high dimensional spaces involved, often both in the parameter and domain spaces. Sparse grids (**BungartzGriebel2004**) are well-known to reduce the effects of the so called 'curse of dimensionality', whereby the cost of computation increases exponentially with the dimension of the model. For example, discretising $[0, 1]^d$ such that there are m points in each dimension leads to m^d points whereas a sparse grid approximation uses only $\mathcal{O}(m \log(m)^{d-1})$ points whilst only increasing error by a factor $\mathcal{O}(\log(m^{-1})^{d-1})$.

A popular way of estimating sparse grid solutions is via the "combination technique" (**Griebel1990**). This technique is unintrusive as it allows one to avoid working directly with the hierarchical basis. Instead, approximations are computed on several anisotropic grids and then combined additively. Figure 5 depicts the combination technique, the equivalent sparse grid and the corresponding full grid. When we refer to computing with sparse grids in what follows we typically mean via the combination technique.

Much recent work with sparse grids has shown that the methods and ideas from the classical work can be applied in other ways. For example, sparse grids and related ideas have been used for

- fault tolerant computing (**HardingHLS2015**; **AliEtal2015**)
- development of highly scalable algorithms (**StrazdinsEtal2015**)
- gradient-enhanced approximation methods (**deBaarHarding2015**; **Jakeman2015**)

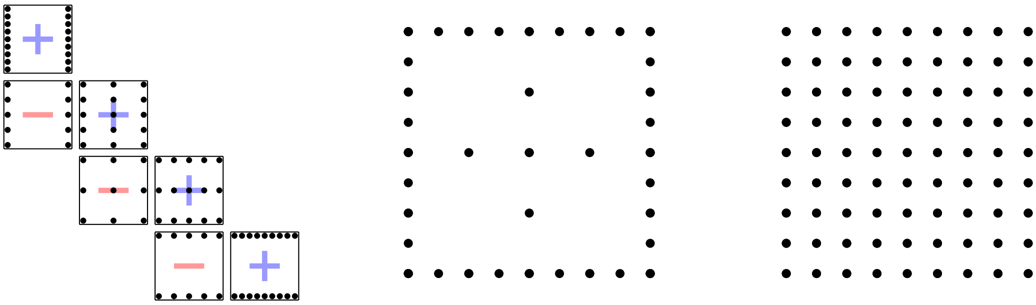

Figure 5: Combination grids on the left (with coefficients, marked with a blue plus for $+1$ and red minus for -1), sparse grid in the middle, full grid on the right.

- multi-fidelity approximation (similar to (**deBaarRDM2015**))
- solution of large scale inverse problems (**Zabaras2010**)
- uncertainty quantification (**JakemanRoberts2013**; **FranzelinDiehlPfluger2014**)
- reduced basis methods (**Peherstorfer2013**; **ChenSchwab2015**)

We envisage this project making full use of these advances in addition to the classical approaches with the end goal of providing fast, accurate and robust approximations of model solutions to scientists whilst incorporating all of the uncertainties in the underlying model and its inputs. Achieving this goal will require further research and development. We expand on some of these points in the remainder of the section.

We have also recently found new ways to incorporate gradient information and multi-fidelity models into sparse grid approximations (**deBaarHarding2015**; **Jakeman2015**; **deBaarRDM2015**).

Another challenge faced by computational science is the sheer size of the computations required to be carried out to evaluate models. This poses two problems in general, the first being the even distribution of workload across a large parallel machine, and the second being resilience to failures that may occur. Having highly scalable software for the offline phase that is resilient to errors is crucial in matinaining confidence in the discoveries and decisions that are established from the model output. Much work has been carried out on the development of highly scalable algorithms for the combination technique (**StrazdinsEtal2015**) which we will utilise and continue to develop. Further, there are known techniques that we will employ to ensure the computations are robust to hard faults (**HardingHLS2015**; **AliEtal2015**). Research on the resilience to soft faults is currently underway [cite Alfredo et.al.] and is also something we intend to integrate into the software framework as it develops. Through the usage and evaluation of these techniques we hope to continue to stimulate research in these areas.

In this project we will leverage on-demand compute resources, such as the Amazon AWS cloud (**amazon˙aws**) and the National Compute Infrastructure NCI Cloud (**nci˙cloud**). Using these cloud services will further improve the project's ability to deliver timely results in high-pressure and time-critical decisionmaking scenarios.

ROLE OF PERSONNEL

The personnel involved in this project will be the CIs from the two ANU colleges, two post-doctoral fellows and four PhD students.

CI Gardner is responsible for the overall project and will lead the interactive and visual interface components. The distributed computing and mathematical component is lead by CI Roberts. One post-doctoral fellow, to be situated in the Research School of Computer Science (RSCS), will develop large portions of the live programming tools and software interface which will interact with the scientific models. The second post-doctoral fellow, to be situated in the Mathematical Sciences Institute (MSI), will develop the numerical methods for computing sparse grid surrogates and reduced basis models which are able to efficiently propagate uncertainties.

The four PhD students will be split between the two departments—two in RSCS and two in MSI, focusing on specific aspects of the project. In RSCS, these will be the dynamic distributed computing infrastructure and the development of live interfaces for data visualisation. In the MSI, these will be the uncertainty quantification and the development of numerical methods for the sparse grid technique.

RESEARCH ENVIRONMENT

As the home institution of the Extempore live programming environment,

It seems we need to address three issues:

1. the environment within the department/school/research group
2. the environment between the groups—perhaps between the MSI group and the CS group, (and also CSIRO?)
3. how does it align with the ANU's research plans and strategies

Ideas…

Numerical methods and applied mathematics—we're kicking goals.

Collaboration across maths and compsci, have track record through e.g. the Fujitsu grant

CSIRO/ADFA/BOM(? although we haven't mentioned them thus far) for more fire and flood stuff

COMMUNICATION OF RESULTS

We will communicate the results of this project by publishing in top-ranked journals and conferences. Advances in numerical methods will be communicated in scholarly journals in numerical analysis and high-performance computing including the journals “SIAM Journal of Scientific Computing” and “Parallel Computing”, and national and international conferences like the “SIAM Conference on Computational Science and Engineering”, the “Computational Techniques and Applications Conference (CTAC)”, “Supercomputing” and various international workshops. We will share our results on the human-in-the-loop workflow in the context of simulations with uncertainty in human factors conferences and journals and conferences such as CHI and VL/HCC.

In addition, the source code contributions of this project will be released to the public. Both the Extempore live programming system (`https://github.com/digego/extempore`) and the AnuGA shallow water simulation package (`https://github.com/GeoscienceAustralia/anuga_core`) are available on GitHub under MIT and GPLv2 licences respectively. We are committed to accessible and reproducible computational science, and support these goals by using free software licences and developing our code in the open on GitHub.

MANAGEMENT OF DATA

REFERENCES

E.1 Justification of funding requested from the ARC for the duration of the Project

E.2 Details of non-ARC contributions

F.12 Research Opportunity and Performance Evidence (ROPE)

Recent significant research outputs and ARC grants (since 2005)

F.13 Research Opportunity and Performance Evidence (ROPE)
Ten career-best research outputs

G.1 Research support from sources other than the ARC

Ok, from here on it's the wild west. Caveat lector.

Ben notes: uncertainty and risk in decisionmaking - it's a real problem that bites

BNE floods were in part a human factors problem - they decided to make decision based on ignorance rather than uncertainty

fire: The myriad of information collected from diverse sources had a broad range of reliability—(**cruz˙anatomy˙2012**)

> Given the nonlinear dynamics of free-burning wildland fires (Sullivan 2009d), model output may be highly sensitive to a particular parameter over one range of values and quite insensitive to that same parameter over a different value range (Albini 1976a, Cruz et al. 2006) ((**alexander˙limitations˙2013**))

but one other limitation is the modelling is just provided - limited ability to assess different scenarios other than the ones presented, and therefore unable to do the helpful "relative/anchoring" natural decisionmaking approach which is what actually happens in these scenarios

> In the aftermath of the January 2011 flood, a senior Wivenhoe engineer has stated before the Commission of Inquiry [17] that uncertainty in BoM rainfall forecasts and the paucity of rain gauges in the catchment immediately above the Wivenhoe Dam led them to conclude that the precipitation forecasts were not sufficiently reliable to form the basis for operational decision making. ((**vandenhonert˙2011˙2011**) p1158)

naturalistic decisionmaking(**lipshitz˙taking˙2001**)

Can we make the case that the better approach to the uncertainty thing is the constructivist one? need some refs I reckon

uncertainty and complexity are present and irreducible. human factors!

human in the loop (or something similar) is generally accepted

but all the papers making a big deal of that gloss over the technical challenges of this + complex models

Notes from Henry: The proposal needs to document the methodology to be followed. I think that this could include:

1. construction of a virtual command room
2. construction of a prototype suite of modelling software
3. interviews with domain experts from Geosciences, Flood management, RFT
    a) from these interviews, develop principal distraction scenarios to be tested in the virtual command room
    b) build a form of cooperative game that would be played by a group of test participants
4. development and testing of interactive visualistions

What this proposal does NOT include is advice from AI agents. That can be left to others in collaboration with us.

At present, the version of the proposal that I have seen has a focus on flood modelling. I wonder if we could better position it in the realms of storm surges. Important for the future. Links to bathymetry and tsunamis etc. What do people think?

zack˙role˙2007 describes four classes (well, 2×2 really) of uncertainty:

- Uncertainty: not having enough information;
- Complexity: having more information than one can
- Ambiguity: not having a conceptual framework for interpreting information;
- Equivocality: having several competing or contradictory conceptual frameworks.

In addition, recent developments in sensor networks and pervasive monitoring technologies have increased our ability to gather data from the world. Streaming data analysis is an active area of research, and in an uncertainty quantification context streaming data provides the opportunity to be continuously estimating model uncertainties and updating our models in response to new data coming in.

This project asks the question: **can streaming data be used to update and model uncertainties and provide intelligible insights in time-critical decisionmaking scenarios?**

One example scenario: sensors detect an earthquake in the Pacific Ocean, and the countdown timer begins—what is the likelihood of a destructive tsunami, where will it impact, and does an evacuation order need to be issued? Some initial modelling has already been done, but new readings are coming in regularly from sensor buoys throughout the ocean. How can this new (and potentially noisy) data be used to update the model, and how does this affect the predicted impact? How does uncertainty in the buoy readings affect our confidence in the model?

What follows is from a draft which Henry and Ben started to put together last year, and there is some material here which could be used to fill out the "building the interfaces" part of the "Research Project" section of this application. But it won't necessarily make sense as-is.

AIMS AND BACKGROUND

Aims

The modern world appears to be saturated with computing power. Contemporary smart phones outperform the original Cray 1 "supercomputer" by three orders of magnitude. Accessing contemporary high-performance computing (HPC) resources has never been easier and, when combined with machine learning(**Hastie2009**), and big data(**Manyika2011**), modern supercomputing performance has the potential to provide answers to deep questions in science, business and the humanities.

Modern comodity computing devices are spectacularly **visual** and **interactive**, enabling users to slide, tap, watch, speak and listen with real-time feedback. It is this combination of interactivity with power which makes mobile and personal computing useful, usable and an agent of change.

But the closer one comes to true high-performance computing, the less interactive computing systems become. Due to a combination of technical and social factors, high-performance scientific computing (HPSC) is still often performed in a batch *compile-run-analyse* style: simulation codes are complex both conceptually (i.e. the underlying mathematics and physics) and in implementation (distributed computation, large codebases). Supercomputing resources are expensive and need to be kept secure.

It is worthwhile imagining some example scenarios that might be possible if better interactive **access to** and **steering of** scientific supercomputing applications were to be available:

1. A computational physicist submits a job to a supercomputer which takes 3 weeks to execute. But subsequent monitoring of incremental visualisations indicates that something about the simulation does not make sense.A check of the, almost 200, input parameters reveals that two of these have been incorrectly set. The phyicist cancels this run and submits another.

2. That same physicist has a sudden insight into the physical system being modelled as a result of watching some of the incremental visualisations of the running code. She initiaties a new simulation with several changes to the parameter deck before the first simulation is complete. Both simulations turn out to be useful in understanding the phenomenon under study.

3. Our physicist decides to improve the numerical accuracy of the model itself. She quickly modifies one of the functions being called by that simulation, one that implements flux-conservation across computational elements of the magneto-fluid being modelled. She then tests her new function on the active program by branching the simulation and swapping the function call on active data. Through a comparison of the two active simulations, she is able to estimate what the effects of her new numerical method will be on the accumulation of errors in the simulation data. She lets both simulations run to completion, and in her subsequent data analysis, she is able to use her error estimates to adjust and compare sumulations using both methods.

4. Performance diagnostics reveal that a multicore simulation is very unbalanced in terms of processor allocation. Due to the complexity of this simulation, the only way to rebalance resources is to rewrite part of the code which maps the geometric grid to the processor model. Our physicist is able to rewrite this function and to swap the new version in, in real-time. The simulation continues seemlessly with enhanced use of resources and finishes faster.

5. A computer scientist is developing a custom visualisation of a scientific simulation. Because of the size of the dataset being rendered, and because of its distribution across hundreds of processors it is impossible to move all of the data. Instead each data nodes

needs to be "visited" and a sample of that data transferred for visualisation. This visualisation is planned, tuned and debugged on running code and is complete by the time actual the simulation is finished.

6. A team of emergency-services personnel are monitoring a real-time satellite feed of a bushfire which is hours away from threatening homes and lives. A computational simulation of the bushfire is being monitored and compared with data but it appears to be giving unreliable results. The simulation experts quickly initiate several companion simulations having different values of fuel loads. The fire-front trajectories of these companion simulations neatly bracket those of the main simulation and provide an estimate of the confidence that can be ascribed to those predicted fire-fronts.

All of these scenarios benefit from a tight human-in-the-loop (HIL) coupling between run-time analysis, visualisation and high-performance simulation software. The benefits include greater *efficiency* and greater **potential for discovery** in scientific workflows. **The aim of this project proposal is thus to build human-in-the-loop interactivity into high performance scientific computing**.

Background

In 2007, a National Science Foundation (NSF) workshop(**Gil2007**) highlighted the need for dynamic interactivity as a looming "grand challenge" in scientific computation and data analysis. In spite of many advances in the field of computational steering it is clear that challenges remain. A recent (2014) review, "Dynamic steering of HPC scientific workflows: A survey"(**Mattoso**), identified six key areas of interaction/steering:

1. **monitoring**: the ability to monitor intermediate quantities of interest (e.g. total energy)
2. **analysis**: the ability to perform analysis of simulation quantities (e.g. viewing a live histogram of the energy distribution)
3. **adaptation**: the ability to adapt/modify the running simulation
4. **notification**: the ability of the system to alert users of the need/opportunity to interfere
5. **interface for interaction**: the interface (either command-based or GUI) presented to the user to monitor and adapt the computation
6. **computing model**: the ability to tailor the types of interaction depending on the compute model in use (e.g. providing full monitoring when running locally on a workstation vs only statistical sampling when the job is distributed over a cluster)

At the end of their survey, Mattoso et al. particularly identify the *analysis* and *adaptation* branches of their taxonomy as not being well addressed by the current state of the art.

Elaborating further, in terms of *real-time analysis* the open challenges are:

- **in-situ analysis**: modern terascale compute infrastructure is often IO-bound, and HPC applications must minimise inter-node communication to achieve efficient use of the hardware. As a result, it is often unfeasible to move the data of interest back to a central node for processing and analysis, so any such analysis must be performed in-situ(**Bennett2012**) on the compute node itself.

- **decision-support tools**: giving a user the *ability* to interfere with a running computational process is one thing—knowing *when and how* to interfere is another thing altogether. Dynamic SWfMS must provide as useful real-time feedback to the user (a domain expert) to assist them in deciding when and how to interfere. This problem is an example of a more general end-user programming (EUP)(**Myers2006**) problem—giving domain experts the appropriate tools to transfer their domain expertise into the computational domain.

In the area of *adaptation* the open challenges are:

- **dynamic workflow engines**: most of the surveyed systems allowed the user to adapt the computation at a high level (e.g. add/delete subtasks). However, there was much less support for modifying the evolution of specific tasks (e.g. swapping out low-level algorithms and tight-loops in computation). This level of adaptivity would allow for even more steering possibilities, but may exacerbate the problem of deciding when/how to interfere (although support for rolling back interventions would be helpful here).

- **parameter slice exploration**: many scientific problems involve searching high-dimensional parameter spaces for optimal parameter configurations, and many algorithms exist for this task. However, there are some problems (e.g. buisness informatics) where cost/loss functions are hard to articulate, and more traditional purely numerical optimization approaches tend to overfit or get stuck in local maxima. In these situations an expert domain user, if presented with appropriate feedback in real-time, may be able to recognise these type of convergence problems early on, and perform appropriate algorithmic and parameter adaptations to avoid the problem.

RESEARCH PROJECT

This project will allow domain experts to better harness the power of computational simulation the in a safe and effective way by creating tools and interfaces for *interactive* computational simulation and anaylsis. This involves integrating research and tools from the domains where such computational analyses are used with insights from human-computer interaction and software workflows.

Through a series of exemplar case studies in different domains, we will unlock new ways to perform complex computation analyses through three steps:

1. **explore**: first, by exploring the problem domain (and existing tools) of interest in conjunction with domain experts, do determine the dimensions of the domain which could most benefit from interactive, human-in-the-loop analysis

2. **enliven**: taking these modelling and simulation codes and providing new interfaces for real-time interaction—this is not just a matter of superficially wrapping these systems in a slick GUI, but requires a restructuring of codes which are designed with batch processing in mind to allow for meaningful interaction and feedback

3. **empower**: using a user-centred design approach, we will create appropriate interfaces (e.g. graphical user interfaces) to empower these domain experts to unlock their expertise in interactive computational simulation

Research significance

Computational simulation and modelling is ubiquitous in both academia and industry, and its importance is growing. Large-scale basic physics relies on computational simulation to probe the subatomic landscape; the auto industry uses simulation extensively for calculating the efficiency of new designs; and algorithmic pattern recognition and recommendation has transformed the way we search (Google), shop (Amazon), watch (Netflix).

However, performing these analyses increasingly requires specialised computational skills such as high-performance and distributed computing. There is a risk that the power of simulation is available only to a HPC-literate elite, and to those rare experts who assimilate both the domain knowledge and computing skills to access it, and to interpret it's results.

Advancing the knowledge base with innovation

As highlighted by Mattoso et al.(**Mattoso**), there is a desire for more dynamic interactivity in scientific workflows, but progress in that area has been limited. This project will contribute to the knowledge base by providing these dynamic workflows, allowing us to answer open questions such as:

- how can interactivity most effectively be "added" to existing scientific workflows?
- how can automated assistance/analysis be provided to the user to best guide their interaction?
- how do other outstanding issues in scientific computation, such as data provenance, reproducibility, etc. impact the interactive workflow?