Using Gaussian Processes for the Calibration and Exploration of Complex Computer Models

by

C.E.Coleman-Smith

Department of Physics
Duke University

Date: ______________________

Approved:

____
Berndt Müller, Supervisor

____
Robert Wolpert

____
Steffen A. Bass

____
Shailesh Chandrasekharan

____
Mark Kruse

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Physics
in the Graduate School of Duke University
2014

Abstract

Using Gaussian Processes for the Calibration and Exploration of Complex Computer Models

by

C.E.Coleman-Smith

Department of Physics
Duke University

Date: __________________________
Approved:

____
Berndt Müller, Supervisor

____
Robert Wolpert

____
Steffen A. Bass

____
Shailesh Chandrasekharan

____
Mark Kruse

An abstract of a dissertation submitted in partial fulfillment of the requirements
for
the degree of Doctor of Philosophy in the Department of Physics
in the Graduate School of Duke University
2014

Revision: e5588b3 (2014-04-23)

Abstract

Cutting edge research problems require the use of complicated and computationally expensive computer models. I will present a practical overview of the design and analysis of computer experiments in high energy nuclear and astro phsyics. The aim of these experiments is to infer credible ranges for certain fundamental parameters of the underlying physical processes through the analysis of model output and experimental data.

To be truly useful computer models must be calibrated against experimental data. Gaining an understanding of the response of expensive models across the full range of inputs can be a slow and painful process. Gaussian Process emulators can be an efficient and informative surrogate for expensive computer models and prove to be an ideal mechanism for exploring the response of these models to variations in their inputs.

A sensitivity analysis can be performed on these model emulators to characterize and quantify the relationship between model input parameters and predicted observable properties. The result of this analysis provides the user with information about which parameters are most important and most likely to affect the prediction of a given observable. Sensitivity analysis allow us to identify what model parameters can be most efficiently constrained by the given observational data set.

In this thesis I describe a range of techniques for the calibration and exploration of the complex and expensive computer models so common in modern physics research. These statistical methods are illustrated with examples drawn from the fields of high energy nuclear physics and galaxy formation.

Now with 47 delightful figures.

A green oak tree's by a cove curving;
A gold chain on that oak is found,
And night and day a cat most learned
Walks by that chain, around, around,
When he walks right, sweet songs intoning,
When leftwards, tells a fairy tale.

I dedicate this thesis to Cassie, and Cup-a-Joe coffee shop. I couldn't have done it without you.

Contents

List of Tables

List of Figures

1
Introduction

And I am dumb to tell a weather's wind
How time has ticked a heaven round the stars.

Reality is messy and complicated, science is a continual attempt to unfold and understand some of this complexity. Frequently when trying to attack hard problems one is forced to turn to numerical models. Big questions usually require big models, with many inputs and outputs and a broad swathe of control parameters and settings. These big models usually require a lot of computer resources, making them expensive to run and often rendering the prospect of a serious exploration of their behaviour infeasible or at least unappealing to the pragmatic researcher.

If our model had no adjustable components and its counterpart process in reality could be reasonably measured we could directly compare the model output to the set of experimental measurements. For computer codes of any complexity this is no longer the case, there are usually a host of adjustable quantities present both experimentally and within the model itself. If we are modeling a very complex process it is unlikely that we are simply solving a well defined equation (such as heat diffusion $\nabla^2\phi = 0$, with some simple boundary conditions) instead we are dealing with the potentially stochastic interactions of many complex sub-processes such as the propagation of interacting particles via an approximation to the Boltzmann equation or the intricate web of interactions within a climate model.

The models I am primarily interested in are those that are designed as exploratory tools as opposed to precision calculators. Research scientists often do not know what is the appropriate way to model a novel phenomenon, the models they do create are always a best (simplest, fastest, easiest to implement,. . .) guess at the underlying processes taking place. With this in mind, our goal is to use the available experimental measurements to simultaneously poke holes in and shore up these models as best we can. This is a deviation from the bulk of the literature in the statistical model-analysis community. This usually focusses on making precise estimates of the deviations of a well understood model from reality. This approach will become important as the models themselves crystallize from exploratory to precision tools. In the initial stages of scientific exploration partial rapid feedback is much preferred over a more complete long term analysis.

Experimental information about complex processes such as galaxy formation or particle collisions can be expensive to obtain and it is usually difficult to provide observations which span the full parameter space. As data is collected gradually an approach which can readily include new results is to be preferred.

It is usually the case that the domain scientists have explored their model output through variations of a single input parameter. Fixing a given value that seems to give good results and then varying the next parameter. The high computational cost of these models usually makes systematic exploration in terms of multiple parameter variation prohibitively expensive. For models with even moderately sized parameter spaces the volume sampled in this fashion will rapidly become minute.

It is important to acknowledge the tension between devoting time and resources to understanding a model and the need to actually “get things done” meaning applying the model to apposite domain-science questions. In an ideal world one would make no siege upon the latter without a detailed and extensive effort towards the former. The methods, techniques, and rules of thumb presented herein are intended to give a harassed scientist a good grasp on their model, it's workings, and applicability without requiring a lifetime of effort.

1.1 The central questions – An Outline

Faced with this kind of situation we may find ourselves asking:

- Can we effectively approximate our model in some fashion? (Chapter: 3)
- How well does our model actually reproduce physical reality, as we understand it? (Chapter: 9)
  - What is the best value of a given setting? Can we sensibly talk about a *true* value of a parameter?
  - What can we conclude about our understanding of physical reality given a set of experimental data and our model?
- What are the most important inputs and parameters for this model? (Chapter: 8)
- What are the uncertainties in our understandings of our model? (Chapter: 7)

In this thesis I will attempt to draw together the wide literature on the statistical analysis of computer experiments and present it in a format that should be accessible to physicists. In so doing I will address the above central issues with illustrations drawn from my own practical experiences in the analysis of transport models of relativistic heavy ion collisions [1, 2], and galaxy formation models [3, 4]. The results and methods collected and expounded upon herein should be sufficient to carry out a complete analysis of a typical computer model.

1.2 What is a computer model?

Suppose that we have some computer model which we will represent as a function $f(x, u)$, a function of two sets of numbers x and u the observation and calibration parameters respectively. The observation parameters are those which can be systematically varied in both physical and computer experiments. The calibration parameters usually will be quantities which are not directly accessible experimentally. These could be parameters which control some purely numerical aspect of the model which are of no great *physical* significance but of course great *computational* significance. We wish to learn about best, in the sense of most compatible with experimental observations, values of these quantities so that we can run our model most effectively.

There may also be calibration parameters which encode some unknown quantity that has a real physical significance, such as the mass of a certain particle or a given coefficient in some model. Of course these "physical" calibration quantities may not actually have a direct corresponding quantity in reality, since they are the product of the long chain of approximations, and conceptual models that makes up the complex game that we call science. Philosophy aside, we are certainly interested in learning as much as we can about these physical calibration variables as they represent a powerful tool for the *falsification* of the theoretical ideas our model itself is built upon.

For simplicity we can begin by restricting our attention to computer models which produce only scalar output, i.e. f is a function

$$f : \mathbb{R}^{p_x} \times \mathbb{R}^{p_u} \to \mathbb{R} \tag{1.1}$$

where p_x is the number of observation parameters (not to be confused with any finite number of actual observations of something) and p_u is the number of calibration parameters. Throughout the course of this thesis I will use the term *sim-*

ulator to stand in for computer-model whenever there might be the possibility of confusion between computational models and the statistical models that we hope to make of them. The diagram Fig: 1.1 gives a quick overview of where I am going with these layers of models.

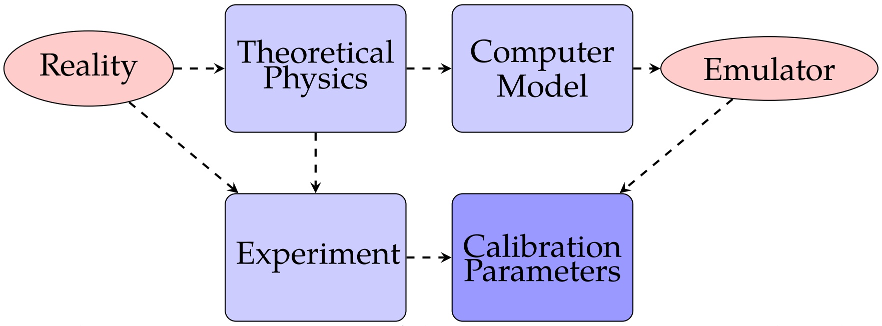


FIGURE 1.1: A schematic representation of the connections between reality, theory, experiment, our computer model or simulator, and the statistical emulator or surrogate we will create of it.

To address how well the model reflects reality, we should adjust the calibration parameters to their "true" values $u_\star$ and then make a set of observations of the model output over the range of x which could then be systematically compared to experimental data. We'll denote the set of experimental observations as $Y_f(x, u_\star)$, these will unavoidably have some observation error ϵ_f associated with them. Let's denote the real output as $Y_f(x, u_\star)$, this is what we would measure if we could make observations without error and what we believe that our simulator is reproducing.

Proceeding in this way we can now develop a model of the difference. Writing the simulator output as $Y_m(x, u) = f(x, u)$, reality as $Y_r(x, u_\star)$ and our field observations as $Y_f(x, u_\star)$ then

$$\begin{aligned} Y_f(x, u_\star) &= Y_r(x, u_\star) + \epsilon_f(x) \\ Y_r(x, u_\star) &= Y_m(x, u_\star) + b(x, u_\star), \end{aligned} \tag{1.2}$$

where $\epsilon_f(x)$ represents the error in the experimental observations and $b(x, u)$ is some unknown function representing the discrepancy between our model and reality. This is all well and fine however we generally have no idea what $u_\star$ should be and so we have to evaluate Y_m over a range of values of u. Furthermore the functional form of b is strongly confounded with u, for differing values of u the model will produce varying output changing the form of b. The form given in (1.2) was first promulgated by Kennedy and O'Hagan [5]. Though this is by no means the *only* possible formulation it is a reasonable place to begin for most simulators.

With the information we have it is impossible to uniquely determine both $u_\star$ and the correct form of b. Imagine two people with weights θ_1, θ_2 standing on a scales at the same time, the measured weight would be

$$y = \theta_1 + \theta_2.$$

No matter how we repeat the process or the values of the two weights we will not be able to make a sensible estimate of either one with only observations of y. In this case the quantities θ_1, θ_2 are not statistically *identifiable*. Of course if were to able to hold one weight fixed (θ_1 say) while systematically varying the other we would be able estimate θ_1. However this is a rather different situation since the systematic variation of θ_2 promotes it from a random quantity to a certain one.

Returning to our definition (1.2) we can make certain choices of prior distribution for the discrepancy which attempt to balance the functional form of b so that its influence is "small" relative to that of the computer model Y_m. This is reasonable since we typically have a fairly large number of observations of the computer model output across the x, u space, although this is typically biased towards the u side of things, and a far smaller number of experimental observations since these are typically drastically more expensive to obtain than most computer models.

The bias term is important to fully and fairly understanding the computer model. If we do not explicitly include it then we are artificially creating some residual x space structure either in the error in the experimental data or into whatever error structure we create for our statistical model of the simulator output. As we will see, the more bias is needed to square the model with experimental data the more uncertain our estimates of $u_\star$

Suppose we carry out a model calibration procedure and obtain some estimates of the *true* values of the calibration parameters $\hat{u}_\star$, we must be very cautious as to how we interpret these estimates. There is no iron-clad guarantee that the platonicaly true values $u_\star$ would actually give a *better* fit to the observed model and simulation data than whatever estimated values $\hat{u}_\star$ we obtain. The estimates we will obtain are the best set of values we could find for the calibration parameters given not only all of the particular details of our sampling of the model (Y_m) and reality (Y_r), the approximation procedures we use to represent the model and discrepancy at untried input locations, and more subtly all of the assumptions that went into the construction of the model and of course into the interpretation of whatever raw information was processed to give the field data.

Nevertheless we should not give up before we even get started, while we cannot hope to exactly pin down the true values $u_\star$ in any practical situation we can reasonably expect to obtain credible ranges for their values. Hopefully these credible ranges obtained after carrying out the analysis of the computer model and running our experiments will be tighter than our prior ranges. Of course the case where they are substantially *wider* may actually be more exciting since then we may have evidence that our computer model and the theoretical framework it is based upon is incompatible with these field observations.

1.3 The statistical analysis of computer experiments: a microscopic review

The systematic investigation of the Fermi-Pasta-Ulam model [6], which models the dynamics of a lattice of non-linearly coupled oscillators, represents perhaps the first formal computer experiment where the goal was to explore the variability in the model's behaviour as a function of various calibration parameters.

The first serious statistical treatment of the analysis of computer experiments can be found in the papers of Sacks et al [7] and Currin et al [8]. Which present frequentist and Bayesian approaches respectively to using Gaussian Processes (GPs), a kind of stochastic process which can be adjusted to produce a wide range of functional forms, to model the relatively smooth output of computer codes and make predictions of the output at untried locations in the parameter space. The left hand panel of Fig: 1.2 shows some random samples from a one dimensional Gaussian Process.

The process of using a GP to make predictions about a smoothly varying field at unmeasured locations given a set of inputs has its roots in spatial statistics, beginning with applications to mineral exploration, and is known as Kriging [9] or Gaussian Process regression. A central reference for spatial statistics is Cressie's book [10], a serious discussion of the mathematical details of Kriging or GP regression can be found in [11]. An introduction to all aspects of GP regression from a machine learning perspective can be found in the excellent book of Rasmussen and Williams [12]. The right hand panel of Fig: 1.2 shows random samples of a Gaussian Process after carrying out GP regression, the process has been conditioned to pass through a set of observations of a toy model (solid black points). In this panel the confidence interval is no longer a uniform band, it shrinks to zero near the training points since here the model values are *known* with certainty and grows in the gaps where the model output is unknown.

The use of GP's and GP regression as part of a larger analysis of a computer model, with a view to obtaining calibrated model predictions is introduced in the seminal papers of Kennedy and O'Hagan [13, 5], where a Gaussian Process is used as a *surrogate* or *emulator* for the model output at untried locations in the parameters space. Here the focus is on making the most accurate predictions rather than on understanding the calibration parameters themselves, the papers by Bayarri et al [14] and Higdon et al [15] develop the Kennedy and O'Hagan framework with an emphasis on calibration itself with the Higdon article espousing a fully Bayesian procedure in contrast to the partial-Bayesian procedure found in Bayarri [1]. The calibration of models which produce multivariate (or functional) output requires some careful considerations, the fully Bayesian framework of Higdon et al is extended to multivariate data in [16], the procedure of Bayarri et al is extended in [17]. Detailed discussions about the optimal design of computer experiments, i.e. how best to layout a finite set of model evaluations through the parameter space, can be found in the book of Santner et al [18] and in [19, 20, 21].

Besides calibration GP emulators have been used to develop an understanding of the variability in a model's output when some (perhaps all) of the parameters are unknown and are allowed to vary according to a given joint probability distribution, this is known as *uncertainty analysis* (sometimes uncertainty quantificiation in engineering applications). The tutorial by O'Hagan [22] (based on the detailed article about uncertainty analysis [23]) is a fine not so technical introduction to the uncertainty analysis of computer codes via GP emulators and also to the use of Gaussian processes as emulators for computer models. So called *polynomial chaos* methods, based upon expanding the simulator as a series of stochastic polynomials, have recently arisen as an alternative method to GP's for understanding variability due to uncertain parameters in dynamical systems [24], while powerful these methods require a complete mathematical formulation of the model and are rather outside the scope of this work.

We can naturally extend the concept of uncertainty analysis to building an un-

[1] The general consensus appears to be that the results from these approaches are roughly equivalent for practical purposes, while the fully Bayesian procedure may be more elegant it also requires rather more Monte-Carlo simulation effort than the MLE drop-in procedure.

derstanding of the relative influences of the various input parameters, and their combinations, upon the model output. This *sensitivity analysis* requires a great deal of computational effort if directly applied to the simulator, however it becomes relatively straightforward when carried out on a GP model surrogate [25, 26, 27, 28]. The resulting information about which parameters (or their combinations) have the most influence on the simulator output provide focal points for the detailed investigation and calibration of the model.

All of these ideas have been directly applied to complex cutting edge computer models in the physical sciences, in cosmology [29], galaxy formation [30, 31, 4, 3], modeling risks from extreme events [32], and increasingly in Heavy Ion physics [2, 1]. The article by Soltz et al [33] deserves mention as a serious attempt at using model calibration to make inference about the true values of unknown physical parameters arising in Heavy Ion physics, although no GP surrogate is used in this particular analysis.

1.4 Sources of uncertainty in computer models

In [5] the authors include a an extensive list of the possible sources of uncertainty arising in the analysis of computer experiments. I have reproduce the essential details of this list here as these definitions provide the basis of a common language for the discussion of computer experiments.

- **Parameter Uncertainty**: Our uncertainty about the values of the calibration parameters. The best values for a given set of observational data may not be the same as the true values.

- **Model Inadequacy**: Even if there was no parameter uncertainty, so that we knew the *true* settings for the calibration parameters there will still be some discrepancy between the predicted value and the true value of the process we are modelling, *"All models are wrong"*.

This discrepancy, specifically the difference between the mean value of the observed process and the model prediction at the true value of the calibration parameters is the model inadequacy.

- **Residual Variability**: We believe that our parameterization of the model is sufficient so that repeated observations at the same settings will always take the same value. In practice this may not be the case, the variability that arises may be due to stochastic elements in the model or it may be that we have failed to fully specify all the conditions or parameters needed. A good example is a simulator which relies to some extent upon Monte-Carlo (MC) methods which introduce residual variability into the final output. This variability may be reduced by increasing the number MC samples used. In some sense the variability can totally eliminated by including the seed supplied to the random-number-generator as a parameter in the analysis, however this is unlikely to be of much practical use.

- **Parametric Variability**: Sometimes it may be useful to obtain predictions of the model output where some subset of the model inputs are allowed to vary according to some joint probability distribution, for instance when one wishes to understand the influence of nuisance parameters or systematic errors on some model.

- **Observation Error**: Model calibration requires a set of field observations, these will necessarily have some uncertainty associated with their collection. Typically these observation errors will be smaller than the computer model uncertainties.

- **Code Uncertainty**: Practically the output of our computer model at some as yet untried location in the parameter space is unknown. Strictly we know that this output must be a function of the inputs and so the output is not truly uncertain. However it may take a great deal of effort to obtain this information for non trivial codes and so it is reasonable to treat this as an additional source of uncertainty.

1.5 Simple and Complex models

Now that we have an idea of what I mean by a computer model, let us introduce an important classification between those models which are fundamentally simple and those which are fundamentally complex or challenging.

Simple models are: models of situations where physical observations and measurements can be readily made, deterministic, almost certainly the right solution to the problem, typically solving engineering problems where the underlying physical process is well understood and one seeks to fully understand a given particular application.

Complex models are: situations where physical observations and measurements are very expensive and difficult to make, not necessarily the right formal description of the problem i.e. phenomenological, perhaps somewhat stochastic, typically these models represent research problems where the underlying physical process is not well understood and a general understanding of this process is the primary goal.

An example of a simple model is a code that would model the evolution of the temperature $T(r,t)$ of a metal block for a given initial distribution of temperature T_0 and a given heat conductance k (and block, i.e. boundary conditions). We could numerically solve

$$\frac{\partial T}{\partial t} = -k\nabla T \quad T(\vec{r}, 0) = T_0. \tag{1.3}$$

Suppose that a suitable experiment could be setup to verify our results. Where could uncertainty arise with this model? First let us suppose that with a simple model we are always *certain* that we are simulating the correct process although perhaps with incorrect model parameters. Given that we believe that we are solving the correct equation we could be uncertain about the initial condition, the heat constant in our block k and our temperature measurement. In this case we are fairly sure that there must be some good set of these parameters which will agree with our experimental data. With a simple model our goal is to develop the best possible treatment of our uncertainty in our implementation of the process and to build up a complete understanding of the models deviations from reality, if indeed there are any.

An example complex model might be a simulation of the interaction and transport of some set of hadrons using the Boltzmann equation to describe the behavior of nuclei during a very high energy nuclear collision

$$p^{\mu}\frac{\partial}{\partial x^{\mu}}F_k(x,p) = \sum_i \mathcal{C}_i F_k(x,p), \tag{1.4}$$

where $F_k(x,p)$ is the one particle distribution function of the k'th species and $\mathcal{C}_i$ represents some complicated collision functional. The complexity in the model arises from the collision term, the interaction of pairs of particles makes this equation rather analytically intractable. Relatively slow numerical simulation of this equation can be carried out.

In this extreme case we are uncertain of a great deal: we do not know the configuration of particles within each nucleus before the collision; we can only approximate the nature of their interactions and we cannot even be certain that we are using the correct model to describe the system. In fact we actually know that using a model which includes a hydrodynamical evolution sandwiched between initial and final periods of microscopic transport gives a better description of the experimental data. To make matters worse we cannot directly observe the system, we can only make measurements of the resulting particles a long time after the collision and try and relate them to the processes taking place in our model.

With a complex model we are not sure that we are actually modeling reality at all. Our goal here is to determine to what extent our complex model could be reproducing reality if indeed it does this at all.

1.6 Settings and parameters

We have defined our model in terms of two sets of parameters u and x. Parameters in set x exist in reality and importantly we can make controlled observations at differing values, hence the name observation parameters. Parameters in set u either do not have a well defined counterpart in reality, or the counterpart is a (known or unknown) fixed value such as fundamental constants. These are often referred to as calibration or tuning parameters.

Experimental observations will be made with the fundamental constants at their true values, the values of the calibration constants do not matter in terms of experimental measurements. We denote the location in u space where everything either takes its true value (in terms of fundamental constants) or the best possible value given the structure of the model as $u_{\star}$.

There are also parameters that will change the experimental observations which have no counterpart within the model, for instance the resolving power of detectors in a heavy-ion experiment is not (directly) included in a transport model like UrQMD [34, 35, 36]. We will not directly deal with the latter. These missing processes / parameters are important as they are likely to contribute to the discrepancy between our model and reality.

For clarity here let us take UrQMD as our model, this is a hadronic transport approach including an ideal (3+1) dimensional hydrodynamic evolution for the hot and dense stage of the evolution. This approach represents a class of state-of-the-art models which decribe the dynamical evolution of heavy ion collision based on combining hadronic transport approaches that are well suited to deal with the non-equilibrium initial and final state and a hydrodynamic evolution where the equation of state is an explicit input and phase transitions can be treated properly. Examples of parameters in the x set would be $\sqrt{S_{NN}}$, centrality of collisions, what kind of nuclei are involved. Examples of calibration parameters include the grid spacing in the hydro code, the smoothing width σ used to convert from the micro to the hydro stages, the freeze-out scale and the equation of state for hot nuclear matter. The latter is an example of a calibration parameter which does exist in reality, learning about these is often of great interest to the domain scientists.

With complex models we often have a large set of internal calibration parameters u and a relatively smaller set of externally variable parameters x. The limited information in the x space further restricts our ability to approximate the systematic bias between the model and reality.

The models that we are typically dealing with day-by-day are complex, their behavior as their inputs are jointly varied is not always well understood. Experimental data is usually limited to small discrete sets of values in the x space. For instance in high-energy nuclear experiments the mechanical details of the particle accelerator only allow a small discrete set of possible colliding particles and energies. In this situation we prioritize exploring the model output space in terms of varying the calibration parameters u and making comparisons with available data over attempting to build up a representation of the model discrepancy.

1.7 Getting started, in media res

To close this introduction, let's consider a tiny example of the use of GP emulators as part of the analysis of a real computer model. I will introduce all of the relevant mathematical details later (mostly in chapters 3 & 4), for now we can simply consider the GP emulator as a black-box interpolation scheme. Importantly this particular interpolation scheme provides a direct measure of the uncertainty in its predictions at untried locations.

As mentioned in the previous section UrQMD [34, 35, 36] is a transport code which describes the evolution of relativistic heavy ion collisions. It simulates the collision from end to end, starting with the initial scattering of the two nucleii, including the hydrodynamic evolution of the bulk of the system and finishing with a Boltzmann transport of the hot hadronic matter formed in the aftermath. This is a complex model with a wide range of potential parameters. Currently the largest uncertainty in the overall description of heavy-ion collisions lies in the specification of the initial states of the colliding nucleii and their early time interactions.

In [2] we explored the influence of two parameters in UrQMD associated with the early stages of the collision, we began with a very simple experimental design, using only two parameters t_{start} and σ. The parameter t_{start} controls the time (in units of fm/c) after initial collision at which the code switches the evolution from the microscopic transport of the hadrons arising from the collision to the hydrodynamic treatment. This is an important parameter as it sets the amount of almost free-streaming that takes place before the strongly interacting hydrodynamical processes take over. It's reasonable to expect that this parameter has some kind of counterpart in physical reality, although the process is likely not a sharp transition. The parameter σ controls the kernel-width of a smoothing function used to convert the hadronic degrees of freedom into the initial energy density for the hydrodynamical evolution, this effectively sets the lumpiness of the initial conditions of a very important stage of the evolution. As hydrodynamics is a fundamentally effective theory of the strong interactions of system this parameter does not have an obvious counterpart in physical reality.

To explore the dependence of the model on these parameters we constructed a GP emulator of the total number of pions produced at mid rapidity in central Au–Au collisions at $\sqrt{S} = 200$ AGeV. The set of training data collected at various values of our parameters of interest t_{start} and σ is shown in the left hand panel of Fig: 1.3 along with the STAR data [37].

Using a simple measure of deviation from the experimental measurement we found a wide valley structure in the σ–t_{start} plane where comparable pion multiplicities could be produced, see Fig: 1.3 Here we defined the implausibility or feasibility of a given location in the calibration parameter space as a measure of the distance between our interpolated predictions and experimental data

$$I^2(u) = \frac{\left(\mathrm{E}[y_{\mathrm{emu}}(u)] - \mathrm{E}[y_{\mathrm{field}}]\right)^2}{\mathrm{V}[y_{\mathrm{emu}}(u)] + \mathrm{V}[y_{\mathrm{field}}] + \mathrm{V}[y_{\mathrm{model}}]} \tag{1.5}$$

where y_{emu} represents a scalar GP emulator and $\mathrm{V}[y_{\mathrm{model}}]$ represents an informed estimate in the overall model error. Throughout this thesis I shall use the notation $\mathrm{E}[X]$ and $\mathrm{V}[X]$ which respectively represent the expectation and variance of the random variable X. This simple measure, the implausibility, includes the information we have about the uncertainty in the interpolation scheme in the term $\mathrm{V}[y_{\mathrm{emu}}]$. This previously unknown structure, suggesting that changes in one parameter can be traded off against changes in the other, would have required a very large set of model runs to uncover without a suitable interpolation scheme.

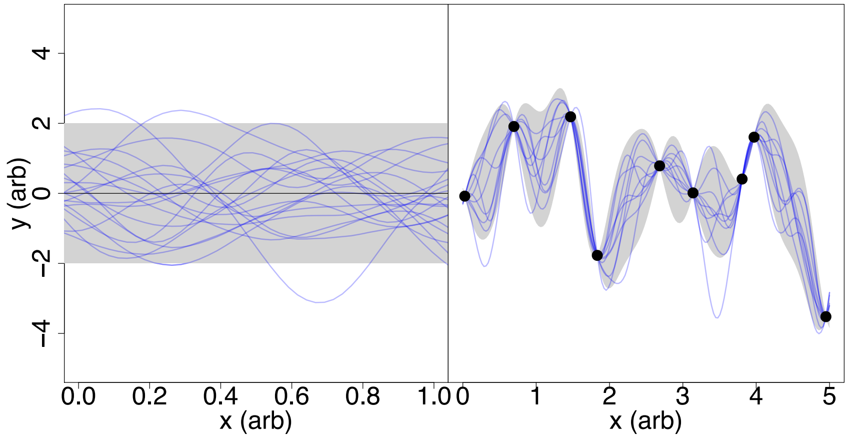


FIGURE 1.2: Left: random samples from an unconditioned Gaussian Process, smooth functions with a predetermined correlation structure in the x dimension. Right: random samples from a Gaussian Process after regression carried out against observations of a toy model (solid points). The random samples now interpolate the training data, passing through the observed points and deviating more from the true function value the further they get from these points . In both panels the gray bands show approximate 95% confidence intervals around the mean.

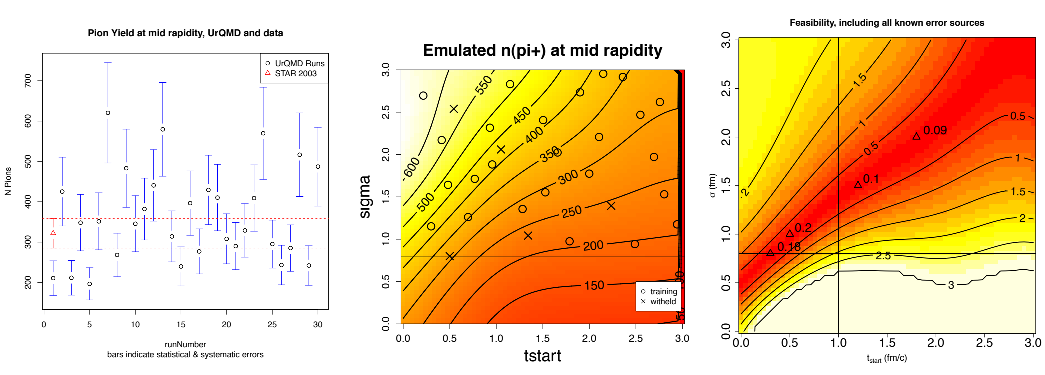


FIGURE 1.3: Left panel: the training data for the simple analysis. The mean number of pions at mid rapidity for each design point is plotted along with 95% confidence intervals, the STAR experimental result is also plotted in red. Center panel: the expected number of pions as predicted by our GP surrogate model plotted across the $2d$ design space, the symbols show locations of the training points. Right panel: the feasibility (1.5) of the various regions in the design space.

2

Heavy Ion Physics – A biased survey

All hadronic matter is made up of a tightly bound quarks which interact through the exchange of gluons. The theory of quark and gluon interactions Quantum ChromoDynamics (QCD) is characterized by the running of the coupling constant α_s with momentum. At the soft momentum scale typical of the interactions between confined quarks and gluons the coupling constant becomes large and the theory is non-perturbative. At very high energy densities this confinement of the hadronic constituents can be broken. There is a transition from individual separate hadrons into a sea of strongly interacting matter. The production and study of this new deconfined phase, the quark gluon plasma (QGP) is the goal of heavy-ion collision experiments like those carried out at RHIC and the LHC. For reviews on the physics of the QGP see [38, 39, 40, 41, 42]. Direct observations of the QGP are not possible due to the incredibly short lifetime of this system ~ 10 fm/c. Once the medium cools past the transition temperature strongly bound hadrons form which are then measured in the experimental detectors. This non-perturbative process forms a screen around the QGP.

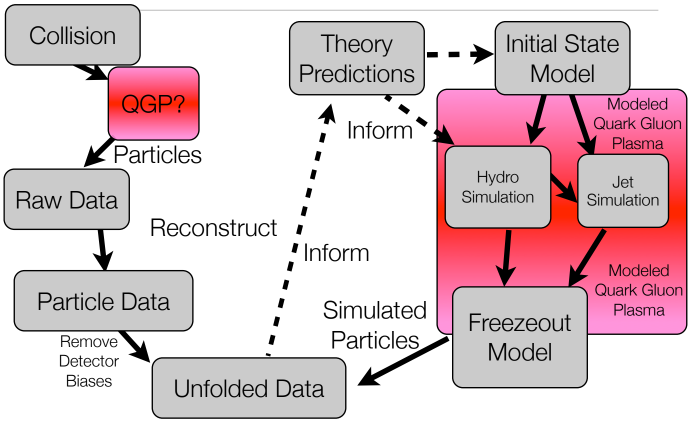
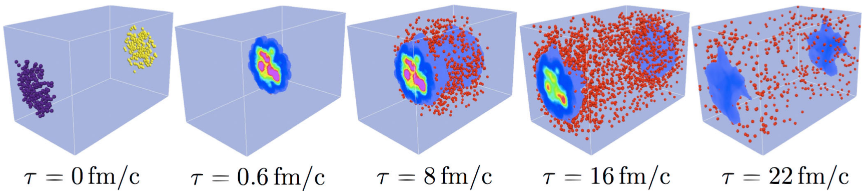


FIGURE 2.1: A schematic representation of the relationships between experiment and theory in relativistic heavy ion physics.

At high energies, i.e. center of mass scales $\sqrt{s_{NN}} > 10$ GeV, the collision of nuclei results in quantitatively different behaviour compared to the collision of single nucleons. There is strong evidence for large scale collective behaviour amongst the products of the initial collision which cannot be descrbed by the superposition of pQCD processes. This is in contrast to proton-proton events at similar scales which can be well described as a combination of a single, perhaps quite complex, hard QCD (and therefore perturbative) process with universal non-perturbative objects. The latter representing the probabilty of finding a parton within a nucleon (a parton distribution function PDF) and the probability of a given hadron being produced by the color-confinment of a particular parton (a fragmentation function).



$\tau = 0\,\mathrm{fm/c}$ $\tau = 0.6\,\mathrm{fm/c}$ $\tau = 8\,\mathrm{fm/c}$ $\tau = 16\,\mathrm{fm/c}$ $\tau = 22\,\mathrm{fm/c}$

FIGURE 2.2: A cartoon of the evolution of a relativistic heavy ion collision. The points in the far left frame represent the nucleons in the inbound nucleii. The colored volumes represent the QGP matter as simulated by a hydrodynamic model. The red points in the later frames are the hadrons produced by the cooling of the QGP. Reproduced from [43].

The colliding matter is believed to form a hot deconfined state called a Quark-Gluon-Plasma (QGP) with a transition temperature at $T_c \sim 170$ MeV. The colliding nucleons are rapidly heated which leads to the deconfinement of their constituent degrees of freedom. The constituents of each nucleon, quarks and gluons, are liberated. The high temperature and pressure forces the nucleons themselves to melt, alternatively one can think of this as a local melting of the QCD vacuum which enforces confinement. The QCD interactions of the now free colored partons give rise to a rich set of interesting observable phenomnena, such as collective flow and the suppression of hard partons and heavy hadronic states.

Heavy ion collisions provide a window for studying the novel properties of the quark gluon plasma and the mechanisms of its creation and evolution. However, only experimental observations of the momenta of particles which comprise the remains of the collision are possible. The process of using theoretical predictions and experimental models to learn about the nature of these hidden processes is schematically presented in Fig: 2.1. This situation with a chain of experimental observations, of the remnants of the true processes of interest, feeding into simulations developed from theoretical models and then feeding back into the experimental process itself is an almost ideal one for the application of the statistical methods contained in this thesis.

To address the fundamental questions concerning the properties of QGP matter and understand its evolution requires the application of large and complex transport models. These models typically combine a viscous hydrodynamic treatments of the evolution of the thermalized quark-gluon plasma ($\sim$1-7 fm/c) with microscopic hadronic transport simulations which describe the propagation and breakup of the produced hadrons ($\sim$7-20 fm/c). During the first fm/c of the collision, when the system is too far from equilibrium for even a viscous hydrodynamic treatment, quantitative modeling carries large uncertainties.

The data sets from the Relativistic Heavy Ion Collider (RHIC) and from the heavy ion programs at the Large Hadron Collider (LHC) are immense. The mixed nature of this data, along with the strong interdependence of disparate observables with respect to basic model parameters, makes a unified interpretation of this data rather challenging. The field has progressed by identifying the principal connections between model parameters and observables through phenomenological and theoretical modelling.

This situation – of non-trivial computer models built from phenomenological treatments of very complex underlying processes and also of widely held *qualitative* beliefs about the influence of such and such upon such and such – is typical across the various sub-fields of QGP physics. Understanding how to use these models along with the wide range of field data to most effectively turn the many fascinating and hard won qualitative results into strong quantitative statements about the properties of QGP matter is a top priority for ensuring the future relevance of the field. This requires a conscious effort on the part of heavy-ion scientists. This transition to precision measurements needed to mature the field will not, infact surely cannot, come about from a business as usual approach to QGP phenomenology.

2.1 Bulk Properties

We can separate the observed behaviour of the QGP into bulk evolution and hard probes. The vast majority of the interactions in the initial instants of the collision are relatively soft ~ 1 GeV with a power law distribution of interactions at higher momentum scales. This soft matter is strongly interacting and appears to give rise to most of the observed phenomena (flow, particle spectra and yields in the final state), its evolution can be well modelled by ideal (inviscid) ultra-relativistic hydrodynamics [44, 45, 46, 47, 48, 49]. In this picture the evolving deconfined material is modelled as a strongly-interacting liquid. Hydrodynamics explains bulk properties, it is not a microscopic theory which can describe the evolution of a particular gluon or quark any more than the Navier-Stokes equations can tell you about the transport of a particular water molecule. All of the detailed microscopic information about the QCD matter is absorbed into the equation of state.

Hydrodynamics is simply a statement of energy and momentum conservation:

$$\partial_\mu T^{\mu\nu} = 0, \tag{2.1}$$

where T is the stress-energy tensor. It is a reasonable first approximation to use the stress energy tensor of an ideal fluid

$$T^{\mu\nu} = (\mathscr{E} + \mathscr{P})u^\mu u^\nu - \mathscr{P} g^{\mu\nu}, \tag{2.2}$$

where $\mathscr{E}$ is th energy density, $\mathscr{P}$ is the presure and u^μ is the four velocity of the fluid. We need to introduce conservation equations for baryon density n ($\partial_\mu N^\mu = 0$) and finally we need an equation of state which relates the pressure $\mathscr{P}$ to the energy and baryon densities. This is typically obtained from lattice QCD. For hydro to be applicable we need the mean free path of particles to be much smaller than the typical size of the system, so that we can describe the system interms of its bulk flow instead of interms of particulate properties. Also the material in question needs to be in local thermal equilibrium, the transition from collision to TE is apparently extremely rapid $\tau \sim 0.6 - 0.8$ fm/c. We can therefore only apply hydro to evolution on scales where $p_t < 2$ GeV (from uncertainty), further we cannot use hydro to describe the initial or final (particulate) stages of the collision. The baryon number conservation equation can be expanded to

$$\begin{aligned} \partial_\mu N^\mu &= u^\mu \partial_\mu n + n \partial_\mu u^\mu, \\ &= Dn + n\theta \end{aligned} \tag{2.3}$$

where I have introduced the convective derivative $D = u^\mu \partial_\mu$ and the four divergence $\theta = \partial_\mu u^\mu$. Explicitly writing out the derivative of the stress-tensor gives

$$\partial_\mu T^{\mu\nu} = \partial_\mu(\mathscr{E} + \mathscr{P})u^\mu u^\nu + (\mathscr{E} + \mathscr{P})\left(u^\nu \partial_\mu u^\mu + u^\mu \partial_\mu u^\nu\right) - g^{\mu\nu}\partial_\mu \mathscr{P}, \tag{2.4}$$

we can simplify these four equations (for $\nu = 0, \ldots, 3$) by projecting along and perpendicular to u^ν, this leads to final set of five ideal hydrodynamics equations

$$Dn + n\theta = 0 \tag{2.5}$$

$$D\mathscr{E} + (\mathscr{E} + \mathscr{P})\theta = 0 \tag{2.6}$$

$$(\mathscr{E} + \mathscr{P})Du^i - \nabla^i \mathscr{P} = 0 \tag{2.7}$$

where ∇^α is the spatial gradient. Recall that the system is closed by the equation of state. A solution can be obtained in the so called Bjöken model [50], where the system is assumed to be boost invariant and homogeneous in the x, y spatial directions. Here we switch coordinates to $t = \tau \cosh \eta, z = \tau \sinh \eta$. In this special (and greatly simplified) case the pressure and energy density are purely functions of the proper time τ, using $D = u^\mu \partial_\mu = \partial_\tau$ and $\theta = \partial_\mu u^\mu = \frac{1}{\tau}$ then the only contribution from the stress-tensor equations is (2.6) which simplifies to

$$\partial_\tau \mathscr{E} + \frac{\mathscr{E} + \mathscr{P}}{\tau} = 0, \tag{2.8}$$

using the equation of state of a relativistic ideal gas $\mathscr{E} = 3\mathscr{P}$ we obtain

$$\partial_\tau \mathscr{E} = -\frac{4}{3}\frac{\mathscr{E}}{\tau}$$

which gives a simple power law solution for the evolution of the energy density as the system expands and cools $\mathscr{E}(\tau) = \mathscr{E}_0 \left(\frac{\tau_0}{\tau}\right)^{4/3}$.

In general one cannot solve the ideal hydrodynamics equations of motion without recourse to numerical methods. Treatment of viscous corrections to the ideal

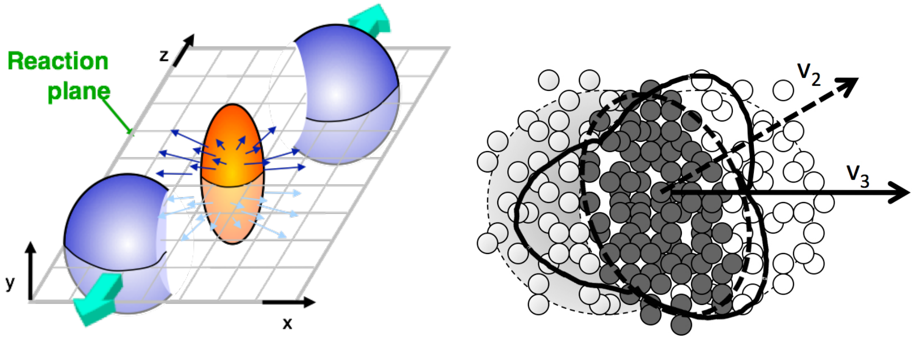


Figure 2.3: Left: A cartoon of the geometry in a heavy ion collision. The two nucleii are very unlikely to ever collide head on, instead the finite impact parameters typically lead to an elliptical overlap region. Right: A cartoon showing the contributions of the finite overlap region to the elliptic v_2 and triangular v_3 flow.

motion introduces additional complexities and transport coefficients. This is usually carried out in the form of a gradient expansion of the equations of motion, originally developed by Israel and Stewart [51, 52, 53] and subsequently adapted and adopted in the work of Song & Heinz [54, 55], Romatschke [56, 57] and more. Inference about the transport coefficients introduced by this process such as the shear and bulk viscosity, derived from the interaction of computer simulations and experimental observations is highly desirable as there are often no direct ways to obtain information about these quantities.

As an example, the shear viscosity of the quark-gluon plasma is known to strongly influence the observed anisotropic flow coefficients v_n. These are the coefficients of an azimuthal Fourier decomposition of the momentum distribution of final particles which provide information about collective flow during the collision.

Typically collisions do not occur at zero impact parameter. The resulting rugby ball shaped overlap region leads to differential expansion rates in the plane of the collision versus out of the collision plane (see Fig: 2.3), as a result of the difference in the initial pressure gradients. An elliptic initial state energy distribution leads to an elliptic final state momentum distribution (albeit in a rotated plane), this elliptic (and higher order) flow is quantified by the moments v_n of a Fourier expansion of the azimuthal angular distribution of the momenta of the observed hadrons

$$\frac{dN}{d\phi} = \frac{N}{2\pi}\left[1 + 2\sum_{n=1}^{\infty} v_n \cos\left(n(\phi - \Psi)\right)\right], \tag{2.9}$$

where Ψ is the event-plane angle which acts as a reference angle for the expansion. Extensive efforts have been put into investigating the correlations between these final state quantities and a similar decomposition of the experimentally unobservable initial state of the nucleii just before they interact [58, 59].

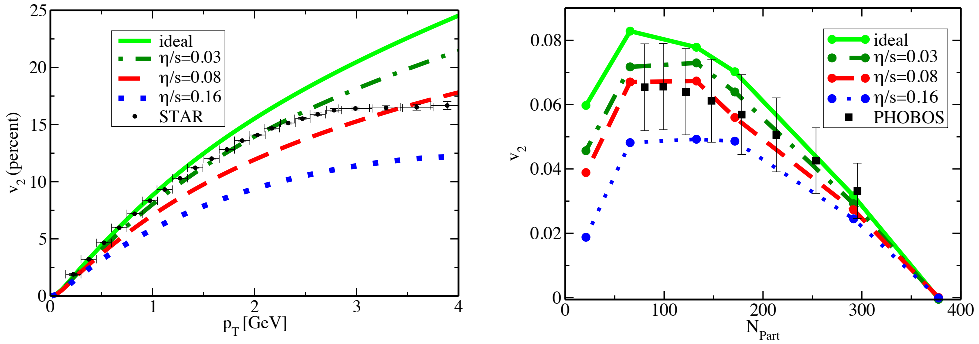


FIGURE 2.4: Left: variation of v_2 as a function of the number of participating nucleons (a measure of centrality) as a function of the shear viscosity to entropy ratio η/s. Right: variation of v_2 as a function of particle transverse momentum p_T for minimum bias events. Au+Au collisions were simulated at $\sqrt{s} = 200$ GeV, both figures are reproduced from [57].

In an early analysis [57], this viscosity was adjusted in a hydrodynamical model until a satisfactory fit with the observed anisotropic flow coefficient v_2 was obtained. The shortcoming of such one at a time an approach is that it leaves untouched the other unknown parameters, such as the spatial anisotropy of the initial state [60], which also are known to influence the flow v_2. To make matters worse each of these parameters also influences numerous other observables. Similar approaches with more advanced models [33, 61, 62, 63, 64, 65, 44, 66, 67] have considered the variation of several parameters at the same time, and also the effects of such parameters on the momentum spectra of the produced hadrons. However these approaches have so far largely been unable to consider the simultaneous variation of more than two or three parameters, or to consider a wider range of experimental observables.

Finally it is interesting to note that the success of these straightforward methods of extracting information about bulk transport coefficients can in part be attributed to the nature of hydrodynamics itself. Hydrodynamics is an *effective theory* of the strongly coupled and thus non-perturbative interaction of the hot QCD matter that makes up the QGP. One consequence of this is that all of the fine details of these interactions have been effectively integrated out of the dynamics and now only enter through the equation of state and the specification of the initial conditions. All of the remaining detail of relativistic hydrodynamics is essentially generic. This is perhaps something of an overstatement, there are many nuanced ways to approach the viscous corrections and deal with numeric instabilities. Nevertheless this genericity is an enormous boon to this particular sub-field since models and theoretical descriptions can all be readily couched in the same language and so productively compared and developed. Sadly this is not the case in all aspects of relativistic heavy ion physics.

2.2 Initial conditions and fluctuations

The major uncertainty in determining transport properties of the QGP, such as the ratio of shear viscosity to entropy, lies in the specification of the initial conditions of the collision. The initial conditions have been mainly assumed to be smooth distributions that are parametrized implementations of certain physical assumptions. Recently the importance of including fluctuations in these distributions has been recognized, leading to a whole new set of experimental observations of higher flow coefficients and their correlations [68, 69, 70, 71]. On the theoretical side there has been a lot of effort to refine the previously schematic models with fluctuation inducing corrections and to employ dynamical descriptions of the early non-equilibrium evolution [72, 59, 73, 74].

Hydrodynamical simulations can take these fluctuations into account by generating an ensemble of runs each with a unique initial condition, so-called *event by event* simulations. This is in contrast to *event averaged* simulations where an ensemble of fluctuating initial conditions is generated, and then a single initial condition corresponding to this set's ensemble average is subject to evolution. Event by event modeling has proven to be essential for correctly describing many details of the bulk behavior of heavy ion collisions [75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85].

The two main models for the generation of hydrodynamic initial conditions are the Glauber [86, 87, 88, 59] and color glass condensate (CGC) models [89, 90, 91, 92, 93, 94]. The Glauber model samples a Woods-Saxon nuclear density distribution for each nucleus. Color glass condensate models are *ab initio* calculations motivated by the idea of gluon saturation of parton distribution functions at small momentum scales x. In CGC models the gluon distribution for each nucleon is computed and the nuclear collision is modeled as interactions between these coherent color fields. Each of these models generates spatial fluctuations whose details depend on the assumptions made in the specific implementation. Glauber fluctuations come from Monte-Carlo (MC) sampling the nuclear density distribution. CGC fluctuations arise similarly with additional contributions from the self interaction of the color fields.

These event by event fluctuations can be seen as intrinsic to the initial energy distribution and lead to very different flow profiles during the collision. The influence of these fluctuations are to be thought of as *additional* to the fluctuations induced by the randomly distributed impact parameters present. As discussed above these impact parameter (or centrality) fluctuations lead primarily to elliptic flow, as measured by the coefficient v_2. The intrinsic or event-by-event fluctuations arise from quantum mechanics rather than from the geometry of the collision region. There is no reason for the nuclear (or even nucleon) ground state to be a position eigenstate and so at the instant of collision their constituents will have some random distribution described by the details of the wavefunction. These fluctuations contribute strongly to the higher modes v_3, v_4 in the flow expansion see e.g. [95, 58, 96]. In Fig: 2.5 the ATLAS event-by-event measured flow coefficients v_n [97] are shown for a variety of centrality parameters (zero centrality means a head on collision). Even in the very central (most head on) collisions there are non trivial amounts of flow, which we can largely attribute to initial state fluctuations since the collision geometry here can hardly have an influence.

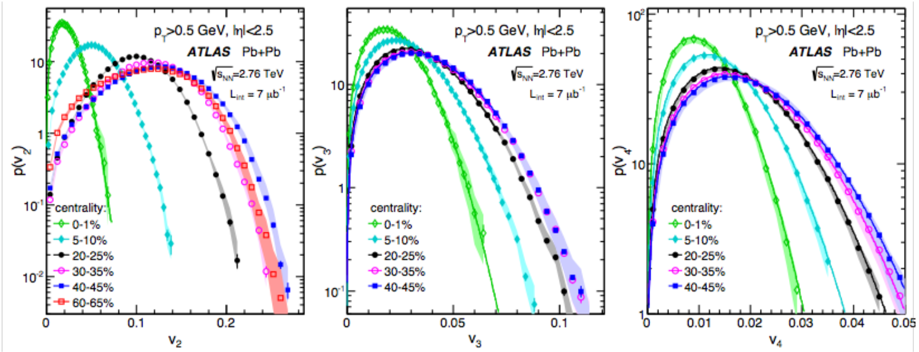


FIGURE 2.5: Event by event flow v_n as reported by the ATLAS collaboration [97]. From left to right the panels show v_2, v_3 and v_4.

2.3 Jet Suppression in QCD matter

In heavy ion physics a jet is a cone of high momentum particles with highly correlated momenta. In a high energy nucleon–nucleon collision the valence quarks, which carry the majority of the momentum, may undergo a hard scattering. This results in the production of a back to back pair of outgoing partons which have some large time-like virtual mass. This virtuality is reduced by the emission of collinear gluons. These gluons can themselves split further into more gluons and pairs of quarks. This repeated emission processes leads to the formation of cones of high momentum partons which will eventually hadronize, a jet.

During a heavy ion collision the majority of the hadronic matter produced comes from the soft interactions of the sea quarks and gluons in the colliding nuclei, eventually leading to the formation of the deconfined medium. The infrequent hard interactions of the valence quarks will lead to the production of jets. While the jets propagate through the medium their constituent partons will interact with the medium to some extent.

Experimental observations of the momentum spectra of hadrons have shown suppression at high p_T at both RHIC [98, 99, 100, 101] ,and the LHC [102, 103, 104, 105, 106, 107, 108, 109, 110]. These high momentum particles represent jet-final states and the reduction in their yield suggests that something is modifying jets compared to those produced in nucleon-nucleon collisions. This suppression seems to come from the quenching of jets within the deconfined medium. For recent reviews of experimental jet quenching observables and the various theoretical approaches to modelling this process see [42, 111].

These results are experimental signals that something is happening to jets as they propagate. This has to be a final state effect, photons which do not couple strongly to the colored medium are not suppressed. Furthermore this suppression is a partonic process, hadron formation times are long and their interactions will take place far outside of the medium. Jet production and evolution in vacuum is perturbative. By measuring the modifications to these processes we may learn some features of the deconfined medium.

The suppression of high p_T hadrons in heavy ion collisions is measured through the nuclear modification parameter R_{AA} (Fig: 2.6), this is the ratio of the measured hadronic spectrum integrated over some range of impact-parameters (binned in rapidity and transverse momentum) to that from $p-p$ collisions scaled by the expected number of binary interactions $\langle N_{bin}(b)\rangle$ which is a strong function of the collisional impact parameter b,

$$R_{AA} = \frac{\frac{dN^{AA}}{dyd^2p_T}}{\langle N_{bin}(b)\rangle \frac{dN^{pp}}{dyd^2p_T}}. \tag{2.10}$$

We would expect $R_{AA} = 1$, i.e. no modification to the pp process, if a heavy ion collision was merely the superposition of $\langle N_{bin}\rangle$ nucleon-nucleon collisions with no further interactions. The deviation from unity implies that some further physics must take place and that during these new processes partons with high p_T tend to lose energy. Collected results from the CMS collaboration at the LHC are shown in Fig: 2.8 [112], photons are clearly not suppressed while charged particles and identified b-quarks exhibit a strong suppression. Similar behaviours were also observed at RHIC collision and p_T scales, in Fig: 2.6 results from PHENIX [113] are shown.

Di-hadron correlations [114] provide another window on this phenomenon. The azimuthal angle distribution of hadrons ϕ is measured relative to trigger particles with high transverse momentum. A correlation structure emerges with a strong peak around $\Delta\phi = 0$ corresponding to particles in the jet, the far side recoil jet at $\Delta\phi = \pi$ is strongly suppressed in central Au+Au collisions compared to d+Au (Fig: 2.7).

These observables (R_{AA}, correlations etc) do not require explicit jet reconstruction. The QGP induced quenching effects can be observed by applying appropriate selection cuts &c to events without ever having to identify which set of tracks and calorimeter tower hits actually make up the jets in the event, and indeed if how many jets are present in that event. This reconstruction process is *de-rigeur* in vac-

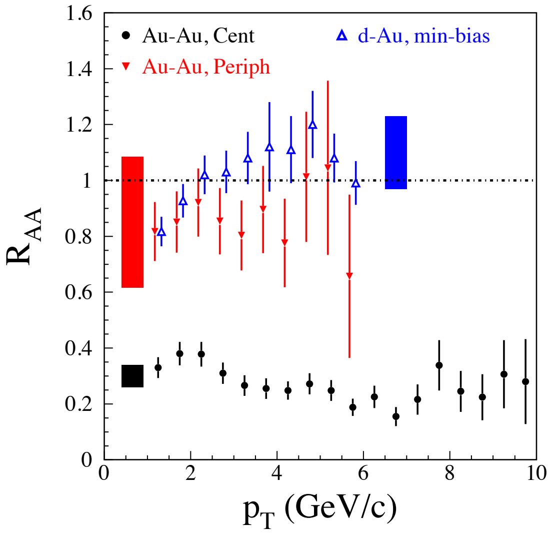


FIGURE 2.6: R_{AA} for Au+Au at $\sqrt{S_{NN}} = 200$ GeV measured at PHENIX relative to a reference NN spectrum. The peripheral collisions show far less suppression, there is insufficient medium formed during these events. Filled bars represent systematic error. Reproduced from [113].

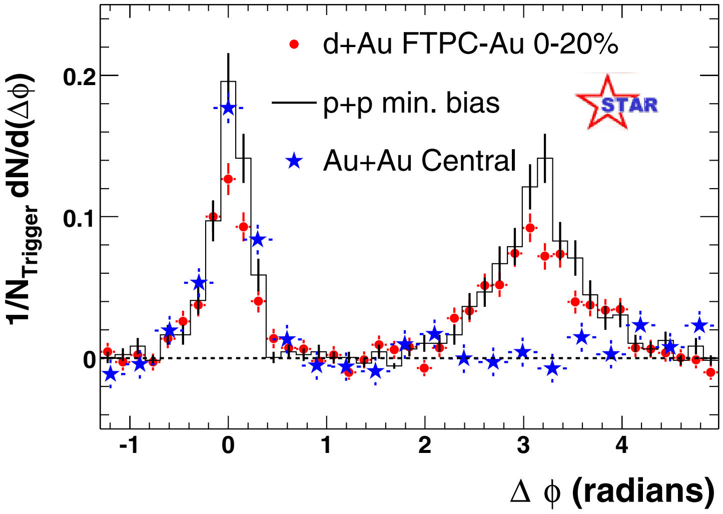


FIGURE 2.7: Azimuthal angle distribution of associated particles with $p_T > 2$GeV. Au+Au collisions (stars) show clear suppression relative to the reference p+p and d+Au data. Reproduced from [115].

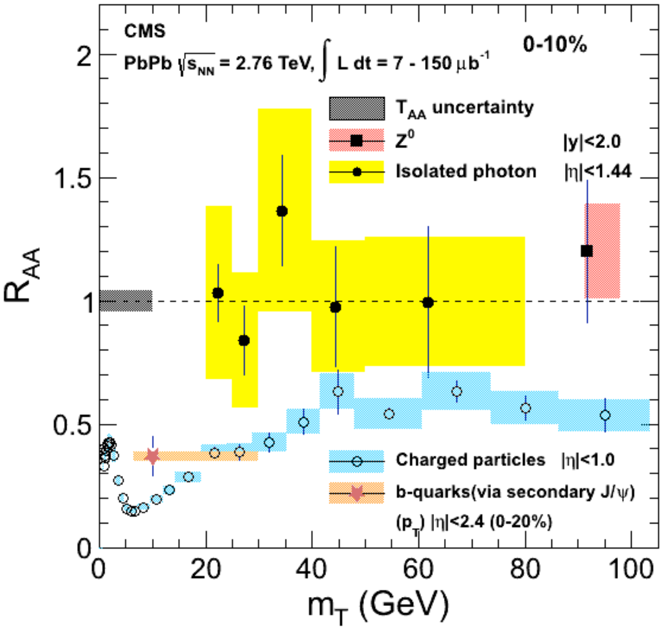


FIGURE 2.8: A summary of the hadronic suppression factor R_{AA} as collected at the LHC by the CMS collaboration [112].

uum jet physics, i.e. in p+p collisions. This also reduces the complexity required by theoretical treatments of these observables.

From the earliest discussions of jet quenching [116] by Bjorken far before any data was collected a wide variety of phenomenological and theoretical models arose. The early treatments of Gyulassy, Wang et al were based upon directly

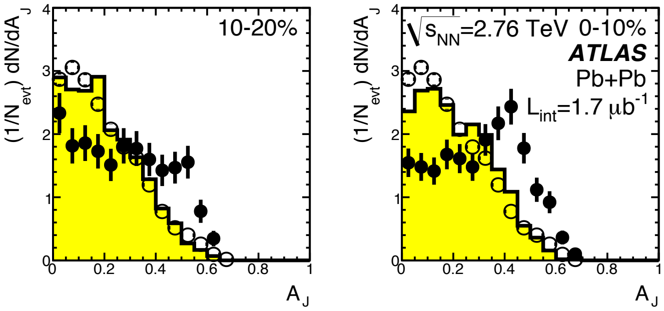


FIGURE 2.9: The ATLAS observed dijet asymmetry for most central collisions reproduced from [102].

enumerating the various possible jet-medium interactions systematically within a simple static medium [117, 118, 119, 120], the Schrödinger equation based treatment of finite size radiation interference effects due to Zakharov [121, 122, 123] and the related BDMPS prescription [124, 125, 126], the strict pQCD calculation inspired Higher-Twist formalism [127, 128, 129], and *many more* [1]. As mentioned above see [111, 131] for fuller reviews.

These mostly phenomenological models were focused on describing just enough of the underlying jet transport phenomena to obtain predictions of R_{AA} and similar, typically these revolve around treatments of the propagation of single hard partons rather than of an entire jet. Notable computer models which include the physics from this era of jet quenching are the venerable HIJING [132] which includes some aspects of the Gyulassy-Wang model, Q-PYTHIA [133], JEWEL [134, 135] and PYQUEN [136] which are all based around the BDMPS formalism.

Jet reconstruction is possible in heavy-ion collisions, however the large background signal from the hadrons produced after the bulk phase of the collision freezes-out makes this process technically challenging. The heavy ion program at the LHC has opened up the study of the modification of entire jets by greatly extending the available kinematic region for jet production. At the LHC jets can be produced at energy scales far separated from the dominant background scale. The study of full jets and their correlations, as opposed to leading hadrons, appears to afford more information about the medium modification, as their shape and fragmentation functions can now be studied. Indeed, the suppression of high energy $E_t \sim 100 - 200$ GeV dijets (a pair of back to back jets formed from the same hard scattering) in heavy ion collisions has now been observed at the LHC [102, 107, 106]. These results have shown the feasibility of using dijets as correlated

[1] In contrast to the theoretical modelling of the bulk evolution, the development of the field of jet quenching has often seemed like a real bike-shed situation [130]

probes of jet modification in hot QCD matter. The dijet asymmetry was initially motivated as a measurement of jet quenching and is defined as

$$A_j = \frac{E_{t,\ell} - E_{t,s}}{E_{t,\ell} + E_{t,s}}, \tag{2.11}$$

where $E_{t,\ell}$ is the transverse energy of the leading jet and $E_{t,s}$ is that of the subleading jet, ATLAS results for the dijet asymmetry in the most central event classes are shown in Fig: 2.9. In this figure the open circles show the results for p-p collisions, where no medium suppression is expected, the yellow filled histogram is the result of running Pythia a Monte-Carlo p-p jet simulation, the black filled circles show the observed results for Pb-Pb collisions. The finite width of the p-p and Monte-Carlo results arises from higher order quantum corrections to the simple back to back jet production. The Pb-Pb results show a clear deviation from the expected distribution. The interpretation is that often one jet of the pair has a shorter distance in the medium to travel than the other and thus is less suppressed, leading to enhanced asymmetry.

With the shift from single hadron observables to full jet reconstructions a new set of simulators arose. The earlier simpler theoretical models and simulators were either not able to make useful statements about the modification of full jets or in some cases were rejected since they were unable reasonably to reproduce these newer observations consistently with the single hadron results. Currently the two most broadly successful models are MARTINI [137, 138, 139] which is a full jet transport based upon the AMY formalism [140, 141, 142] and YaJEM [143, 144, 145] which is a rather simple quenching scheme motivated by BDMPS. Aspects of these full jet results have been successfully reproduced by various authors with a variety of models [146, 147, 148].

It is not obvious how to extract details of the jet suppression mechanism such as the energy loss rate from experiment. Jet modification in QED can be measured since the final state particles from a QED jet, electrons and photons, can be measured in a detector. In a QCD jet the constituents are necessarily unmeasurable, confinement sits in between any detector and the jet-physics. Theoretical results have tended to be expressed in the form of intuitive quantities which express the rate of energy loss and similar quantities which allows for ready model comparison, however efforts are made to generate realistic predictions for hadronic final states to allow comparison to experiment (see for example [149, 150]).

The determination of the transport coefficients of the Quark-Gluon-Plasma (QGP) is a major goal of the LHC and RHIC heavy ion programs. Partons moving through the QGP lose energy and gain momentum perpendicular to their trajectory. The interaction of a hard probe with the QGP medium is traditionally divided into elastic scattering and medium induced radiation. Although this separation may be artificial it is convenient to view the two processes as being independent. The strength of the probe's interaction with the medium is quantified in-terms of the transport coefficients $\hat{q}$ and $\hat{e}$ which represent the average transverse momentum gained and the average energy lost by a hard probe passing through a QGP medium. These can be schematically defined in terms of the differential elastic scattering cross section $d\sigma/dt$,

$$\hat{q} = \langle \int t \frac{d\sigma}{dt} dt \rangle, \tag{2.12}$$

$$\hat{e} = \langle \int (E_f - E_i) \frac{d\sigma}{dt} dt \rangle, \tag{2.13}$$

where the angular brackets denote a medium average. More formal definitions can be given in terms of gauge field correlators in QCD [127, 151, 152]. The radiative process and the role of its transport coefficient $\hat{q}$ in jet quenching have been discussed extensively [153, 154, 155], aspects of the practical definition of these coefficients along with their extraction are discussed in [156, 157]. The relative importance of elastic energy loss is typically less well understood.

There have been several notable attempts to collate and compare these models and their predictions. Bass et al [155], manually adjusted these coefficients in several models in an attempt to reproduce RHIC data and produced credible ranges on $\hat{q}$ for each model. The details of the medium model that the simulated jets pass through add another layer of complexity to the calibration process, in [131] the outputs of mostly single hard probe models were compared for propagation through a "brick" of QGP matter at a fixed temperature, this effort has been recently updated by the recent Jet-Collaboration article [158] where systematic attempts to extract these transport coefficients from experimental data are made. To progress our understanding of the mechanisms of jet quenching the field needs an objective measure of how these models perform in comparison with the data. A systematic comparison of the predictions of several models using the methods outlined in this thesis could be extremely fruitful.

2.4 Summary

Above I have outlined some of the most interesting physics of the QGP in my personal opinion. However the bulk evolution, initial state specification and jet quenching are by no means the only topics of interest. Looking further one can consider: the production and modification of heavy quark jets; the propagation, production and destruction of QCD bound states in the medium; the deconfinement and chiral phase transitions and the equation of state of QCD matter; the final state correlation structure and what it can tell us about the initial interactions; statistical-mechanics based models of the thermal production of hadrons at freeze out; and understanding the recently observed QGP like phenomena found in p-A and high multiplicity p-p collisions.

Already some progress has been made towards the application of statistical techniques to the calibration and exploration of computer models of the evolution of the bulk of the hot QCD matter. It is my hope that the methods and results contained in the rest of this thesis go some way towards promoting their adoption in the Heavy Ion community. Especially so since the field is so fundamentally based around determining the values of calibration parameters.

3

Gaussian Processes

I'm very well acquainted, too, with matters mathematical,
I understand equations, both the simple and quadratical,
About binomial theorem I'm teeming with a lot o' news,
With many cheerful facts about the square of the hypotenuse.

In order to explore, understand, and calibrate a complex computer model we need a lot of information about the output of the model. At the same time we always need to balance this against the realities of finite computing and time resources. If our model is relatively well behaved, intuitively this means producing rather smooth output as a function of its parameters, we might turn to an interpolation scheme. Then with a sensible choice of sample locations one would hope to get a good understanding of the model's response from a fairly small number of actual model runs.

In this chapter I will introduce a method that uses Gaussian Processes (GPs) for interpolation of computer model outputs. A seminal reference for GPs and their applications is Rasmussen & William's book [12], along with [11, 7, 23].

There are many possible interpolation methods, many of which like splines and polynomial interpolation seem appealing through their simplicity. In practice this simplicity can often be a facade thrown over a mass of pitfalls.

Making the (moderate) effort needed to use GPs for this purpose has several distinct advantages. A GP interpolation scheme is a statistical model of our computer simulation, often called an emulator. These emulators are defined in terms of *probability distributions* for the output of the computer model. As a consequence of this predictions for the model output at new locations naturally come with a measure of how reliable those predictions are, making them a powerful tool for computer experiments.

Throughout this chapter I shall illustrate various concepts with application to the $1d$ toy model

$$Y_m(x) = \sin(x) + 2\sin(2x) - 2\sin(4x). \tag{3.1}$$

This toy model was picked for its simplicity and several characteristic length scales.

3.1 An introduction to Gaussian Processes

A Gaussian Process (GP)

$$\mathrm{GP}\left(\mu(\cdot), \mathcal{C}(\cdot,\cdot)\right) : \mathbb{R}^n \to \mathbb{R}, \tag{3.2}$$

is a stochastic process over some n dimensional space specified via a mean function $\mu(\cdot)$ and a covariance function $\mathcal{C}(\cdot,\cdot)$. A stochastic process is a parameterized collection of random variables $\{x_t\}_{t\in T}$ defined on a probability space $(\Omega, \mathcal{F}, P)$, and taking values in $\mathbb{R}^n$. The indexing space T is usualy $[0,\infty)$ but this is quite general and can be extended to subsets of $\mathbb{R}^n$. E.g. A random walk, gamblers ruin, arrival times of cars at traffic lights. See [159] for many examples.

A GP is defined by the fundamental property that any finite marginalization of the process to some set of points $X = \{x_1, \ldots, x_k\}$ will be multivariate-normal (MVN) with mean and covariance given by μ and $\mathcal{C}$. Thus, restricting the process to a single point x would give $P(x) \sim \mathrm{N}(\mu(x), \mathcal{C}(x,x))$, for a set of three points

$$\begin{aligned} x_1, x_2, x_3 \sim \mathrm{MVN}(\vec{\mu}, K), & \\ \vec{\mu} &= (\mu(x_1), \mu(x_2), \mu(x_3))^{\intercal}, \\ K_{i,j} &= \mathcal{C}(x_i, x_j). \end{aligned} \tag{3.3}$$

where $K_{i,j}$ is the i, j'th element of the covariance matrix.

A Gaussian process is translation invariant or *stationary* if

$$\mu(s) = \mu(s+h), \quad \mathcal{C}(s+h, t+h) = \mathcal{C}(s,t) \tag{3.4}$$

for all h [1]. If this is the case then the mean must be constant and the covariance function can only depend on the distance between two locations $\mathcal{C}(s,t) = \mathcal{C}(s-t, 0)$. In this case $\mathcal{C}(s,t) = \mathcal{C}_0(s-t)$, for some $\mathcal{C}_0(h) = \mathcal{C}(h,0) : \mathbb{R}^n \to \mathbb{R}$. We shall typically be concerned with stationary GP's, in this case all the interesting information about the process is contained in the covariance function. Note that a stationary GP can be used to model a simulator with some overall trend in its output. This is accomplished by treating the overall trend first typically with a linear model and then using the stationary GP to model the residuals. Real world random fields, such as the distribution of oil or gold across a given geographic region, may only be approximately stationary. However stationarity is a typically a reasonable assumption for "smooth-ish" computer models which don't undergo some dramatic change across their parameter space. Models with "jumpy" phenomena such as phase transitions or regime changes can also be treated but additional care is needed[2] .

Not every function can be a covariance function. For starters it must be an even function, this arises neatly from the symmetry of the covariance, writing $Z(\cdot)$ as

[1] Strictly this is the definition for a general stochastic process to be *weakly stationary* but the two concepts are equivalent for a GP

[2] In this case one might consider dividing the model output space up into distinct regions and training different GP's in each region, see [160]

the GP evaluated at a given point

$$\mathcal{C}(s-t) = \operatorname{cov}(s,t) = \mathrm{E}\left[(Z(s)-\mu(s))(Z(t)-\mu(t)\right] = \operatorname{cov}(t,s) = \mathcal{C}(t-s).$$

From the marginalization property we obtain the restriction that the covariance function $\mathcal{C}$ must be positive definite, since this is arises from the form of the multivariate normal density,

$$\int \mathcal{C}(x,y)f(x)f(y)\; d\mu(x)d\mu(y) \geqslant 0. \tag{3.5}$$

Bochner's theorem states that, all positive semidefinite functions can be written

$$\mathcal{C}_0(h) = \int_{\mathbb{R}^n} \exp(ih\omega)G(\omega)d^n\omega \tag{3.6}$$

where $G(\omega)$ is a positive function on $\mathbb{R}^n$. All positive semidefinite fns can be written in this form for some positive measure $G(\omega)$, the *spectral measure*. Since the Gaussian process itself is real, the spectral density must also be an even fn and we can write

$$C_0(h) = \int_{\mathbb{R}^n} \cos(h\omega)G(\omega)d^n\omega G(\omega) = \frac{1}{2\pi^n}\int_{\mathbb{R}^n} \cos(h\omega)C_0(h)d^n\omega. \tag{3.7}$$

This spectral representation of stationary GP's provides a tool to gain some powerful insights into their behaviour and particularly into their asymptotic properties, see for instance Stein [11] .

A Gaussian process is an extension of a normal distribution to a stochastic process that generates functions with a controllable amount of correlation across the indexing space. This makes them a very suitable choice for a prior on a distribution of unknown functions, such as the output of a computer model.

3.2 Drawing samples from a GP

A GP is a probability distribution for functions with some mean and spatial correlation structure. This seems rather abstract, given particular mean and covariance functions how can we generate realizations of the process.Let's suppose that our GP is defined on a one dimensional space, i.e. $n = 1$. Using the fundamental marginalization property if we pick some finite set of say k points at which to evaluate the process the problem becomes one of drawing samples from a k-variate MVN. Given the mean and covariance functions $\mu(\cdot), \mathcal{C}(\cdot, \cdot)$ we can generate a set of samples at these points as follows.

1. Compute the covariance matrix $\mathcal{C}$ for the k points, where
$$\mathcal{C}_{ij} = \mathcal{C}(x_i, x_j)$$

2. Compute the Cholesky decomposition S of the covariance matrix $\mathcal{C} = SS^\intercal$ [161].

3. The vector
$$z = \mu + Su \tag{3.8}$$
where u is a vector of k standard normal samples, i.e. $u_i \sim N(0, 1)$, is the desired sample from the GP.

We directly see that the vector z has the correct expectation $\mathrm{E}[z] = \mu$. The covariance of z is also correct
$$\begin{aligned} \mathrm{cov}[z] &= \mathrm{E}[zz^\intercal] = \mathrm{E}[Su(Su)^\intercal], \\ &= S\mathrm{E}[uu^\intercal]S^\intercal = SS^\intercal = \mathcal{C}. \end{aligned} \tag{3.9}$$

Some illustrations of samples drawn from GP's can be see below in Fig: 3.8 and Fig: 3.9, a more complex example is shown in Fig: 3.1. Here a GP is used as a model for the space-time fluctuations of a massless relativistic scalar field in $1 + 1$ dimensions. The GP has mean zero and covariance function
$$\mathcal{C}(x, y) = \frac{1}{2}\left(iG^+(x, y) + iG^-(x, y)\right), \tag{3.10}$$
$$G^\pm(x, x') = \frac{-i}{4\pi^2}\frac{1}{\Delta x_\pm^2}, \quad \Delta x_\pm^2 = (t - t' \pm i\epsilon)^2 - \|x - x'\|^2$$

where $G^{\pm}(x,y)$ are positive and negative frequency massless free-field Green functions. In Wightman's axiomatic construction of QFT [162] one can show that the two point function of the field itself $W(x,y) = \langle 0|\phi(x)\phi(y)|0\rangle$ is itself positive definite, which could also be interesting to simulate.

Practically one may need to add a vector of random noise w with $w \sim N(0,\epsilon)$ and $\epsilon \ll 1$ to the diagonal of the covariance matrix $\mathcal{C}$. The eigenvalues of covariance matrices usually fall off very rapidly which can make the Cholesky decomposition numerically unstable. This adds noise with variance ϵ^2 to the generated samples, however one can usually select a sufficiently small value of ϵ such that the linear algebra converges without appreciably changing the samples.

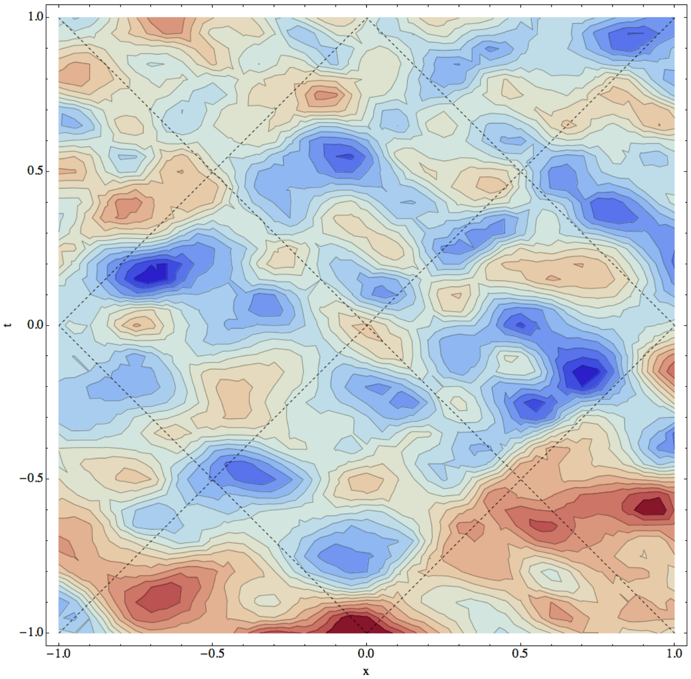


FIGURE 3.1: A realization from a Gaussian Process model of a massless relativistic scalar field in $1+1$ dimensions. The GP has mean zero and covariance function given by (3.10). The dashed lines are drawn along null (light-like) directions, it is interesting to note that the fluctuation structure is fairly well correlated with these directions.

This Cholesky decomposition based method is $\mathcal{O}(k^3)$, for very large values of k the linear algebra may become unstable and computationally impractical. There are several other more mathematically complex methods for simulating (drawing samples from) Gaussian processes which are more computationally efficient see [10, 163, 164, 165].

3.3 GPs for Interpolation (or regression)

We can use a GP as a method for interpolating the output of a computer model, for the purposes of this section we will not distinguish between the parameter sets u and x. Let us denote the model output at a point x in the combined parameter space as

$$Y_m(x) = f(x), \quad \mathbb{R}^n \to \mathbb{R} \tag{3.11}$$

where for now we have assumed that we can make observations of the computer model output without any noise or uncertainty and that the model output is univariate and real. We will use a Bayesian approach to develop a statistical model of the output of the code, an emulator. This is done by taking a Gaussian process, with a given covariance function $\mathcal{C}$ and mean μ, as the prior distribution for the simulator output and conditioning it a set of observations of the simulator.

Let us denote the design, the set of d points in the parameter space where the model has been evaluated, as

$$\mathcal{D} = \{x_1, x_2, \ldots, x_d\}, \quad x_i \in \mathbb{R}^n. \tag{3.12}$$

The vector of d outputs evaluated at these points is

$$\mathcal{Y} = (Y_m(x_1), Y_m(x_2), \ldots, Y_m(x_d)) \,. \tag{3.13}$$

Our GP prior amounts to $Y_m \mid \mathcal{C}, \mu \sim \mathrm{GP}(\mu, \mathcal{C})$, we can update this prior with our set of observations $(\mathcal{D}, \mathcal{Y})$ to obtain a posterior distribution for the simulator output $Y_\star$ at some yet untried set of k points $X_\star = \{x_{\star,1}, \ldots, x_{\star,k}\}$. Our observations of the model output represent a finite marginalization of our GP prior, as such they are distributed

$$
\begin{aligned}
\mathcal{Y} \mid \mathcal{D}, \mathcal{C}, \mu &\sim \mathrm{MVN}(\mu_\bullet, K_{\bullet,\bullet}), \qquad (3.14)\\
\mu_\bullet &= \left(\mu(x_1), \mu(x_2), \ldots, \mu(x_d)\right),\\
K_{i,j} &= \mathcal{C}(x_i, x_j), \quad x_i \;\&\; x_j \in \mathcal{D},
\end{aligned}
$$

where $\mu_\bullet$ (d) is the vector of the prior mean evaluated at each point in the design space and $K_{\bullet,\bullet}$ $(d \times d)$ is the covariance matrix arising from evaluating the prior covariance between each point in the design. We can write the joint distribution for our set of known observations $\mathcal{Y}$ and the as yet unknown $y_\star$ as

$$
\begin{pmatrix} Y_\star \\ \mathcal{Y} \end{pmatrix} \sim \mathrm{MVN}\left\{ \begin{pmatrix} \mu_\star \\ \mu_\bullet \end{pmatrix}, \begin{pmatrix} K_{\star,\star} & K_{\star,\bullet} \\ K^\intercal_{\star,\bullet} & K_{\bullet,\bullet} \end{pmatrix} \right\} \qquad (3.15)
$$

where $K_{\star,\star}$ $(k \times k)$ and $\mu_\star$ (k) represent the prior covariance function and mean evaluated at the unknown locations and $(K_{\star,\bullet})_{ij} = \mathcal{C}(x_{\star,i}, x_j)$ $(k \times d)$ is the matrix of covariances between each new point $x_{\star,i}$ and the current design set. This matrix plays an essential role in the rest of the formulation, we will find that our predictions for the new points are weighted averages of the training data with the weighting given by this set of covariances.

We can write the conditional distribution for our untried locations $Y_\star$ given our set of observations $\mathcal{Y}$ as another multivariate normal

$$
Y_\star \mid X_\star, \mathcal{Y}, \mathcal{D}, \mathcal{C}, \mu \sim \mathrm{MVN}\left(\bar{\mu}(X_\star), \bar{K}\right) \qquad (3.16)
$$

following the derivation in § A.2 from (A.11) we obtain the posterior mean

$$
\bar{\mu}(X_\star) = \mu_\star + K_{\star,\bullet}\, K^{-1}_{\bullet,\bullet}(\mathcal{Y} - \mu_\bullet) \qquad (3.17)
$$

this will serve as our prediction for value of the model output at the untried locations. From (A.12) we have

$$
\bar{K} = K_{\star,\star} - K_{\star,\bullet}\, K^{-1}_{\bullet,\bullet} K^\intercal_{\star,\bullet} \qquad (3.18)
$$

this gives the posterior covariance at the set of untried locations. The actual simulator observations $\mathcal{Y}$ enter only linearly in the posterior mean and are entirely absent from not the posterior variance. Our ability to make accurate predictions/interpolations of our computer model is apparently only a function of the choices we make when designing our experiment.

3.4 Developing an understanding

Let's take a moment to examine these results, for simplicity let's consider the case where we only want to make predictions at a single unknown point. A simple example of GP regression is shown in Fig: 3.4, the left panel shows several draws, the light blue lines, from a GP prior with zero mean and a power-exponential covariance function. For more details on the covariance structure itself see § 3.10.1. It is important to note that the draws from the prior are smooth functions, this reflects our prior belief that the output of any computer model we are hoping to emulate is also reasonably smooth.

In the right panel a set of 9 observations of the example model (3.1) have been made, these are plotted as the solid points. Draws from the posterior distribution, with mean given by (3.17) and variance given by (3.18) are plotted. These posterior draws all pass through the training points, they are still smooth functions and their variability increases away from the training locations. The gray band shows an approximate 95% confidence interval around the process mean in both panels. In the trained case these bubbles grow away from locations where observations have been made and shrink to zero at the trained locations.

The posterior mean (3.17) is a linear combination of the prior mean at the unknown location $\mu_\star$ and the term $K_{\star,\bullet}K^{-1}_{\bullet,\bullet}(\mathcal{Y}-\mu_\bullet)$ which is a linear transform $\mathcal{A}(x_\star) = K_{\star,\bullet}K^{-1}_{\bullet,\bullet}$ applied to the residuals of the observed data $\mathcal{Y}$ under the prior mean $\mu_\bullet$.

For clarity let's drop the prior mean, i.e. $\mu = 0$.

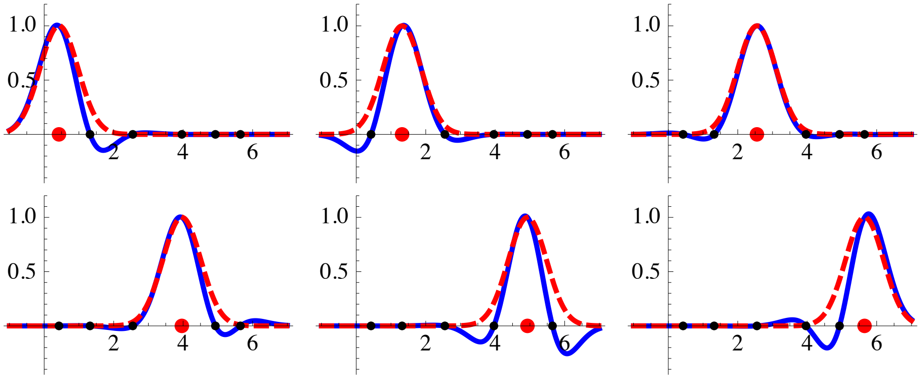


FIGURE 3.2: In red dashed lines elements of the prior covariance vector $K_{\star,\bullet}$ are plotted for a simple one dimensional example. In blue solid lines the equivalent elements of the vector $\mathcal{A}(x_\star) = K_{\star,\bullet}K_{\bullet,\bullet}^{-1}$ are plotted. The points show the design locations, the corresponding element for each panel is enlarged and plotted in red.

We can view $K_{\star,\bullet}$ as a vector of functions of $x_\star$, each of these functions is the prior covariance function centered on one of the design points. In Fig: 3.2 the red dashed traces show plots of the elements of $K_{\star,\bullet}$ for a simple one dimensional example. This represents our prior knowledge of the correlation structure in the design space. The equivalent elements of $\mathcal{A}(x_\star)$ are plotted in blue. The more complicated structure here shows how our choice of the whole design influences the shape of the correlations between points in the space. The extremities which reach outside the design in the first (top left) and final (bottom right) panels are relatively unchanged while the other panels some modification due to the influence of the other points.

In Fig: 3.3 the panels show partial interpolation function

$$B_k(x_\star) = \sum_{i=1}^{k} \mathcal{A}(x_\star)_i \mathcal{Y}_i, \tag{3.19}$$

which includes the first k observations. Although the resulting interpolation is of a very poor quality, when compared with the underlying function (dashed red curve) this figure makes it clear how successive observations points influence the shape of the posterior mean. With this in hand we can understand how $\mathcal{A}(x_\star)\mathcal{Y}$

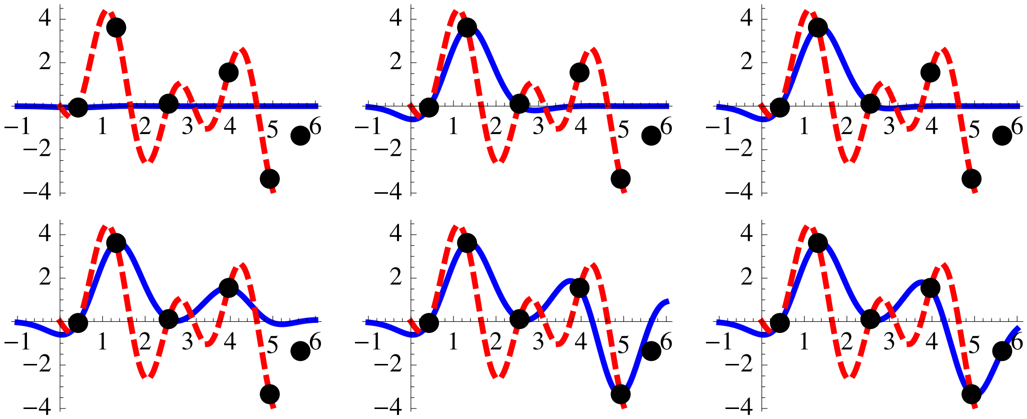


FIGURE 3.3: From top left to bottom right the panels show B_k plotted in blue, as given in (3.19). The toy model is shown in dashed red and the training observations are plotted as the solid points. At each panel an additional training point is included into the resulting partial interpolation function.

can be viewed as a weighted dot product between the modified covariance kernels and the observations. The posterior variance at the unknown location (3.18) is also a linear combination of the prior variance and another term $K_{\star,\bullet}K_{\bullet,\bullet}^{-1}K_{\star,\bullet}^{\intercal}$ which can be interpreted as another weighted inner product, however this time it is a weighted norm of the vector $K_{\star,\bullet}$.

Since $K_{\bullet,\bullet}$ is a positive definite matrix, its inverse is also positive definite therefore the posterior variance $\bar{K}$ of our prediction at the untried location is always smaller than our prior $K_{\star,\star}$. Following a similar line of argument as used above to derive the form of this posterior variance we can conclude that every time we add an additional observation to this GP model our posterior variance at the untried location will decrease relative to the previous value. This is an interesting consequence of our assumption of stationarity. The amount that our posterior variance will decrease by is not entirely trivial to obtain.

Let us return again to our inspection of the posterior mean and variance. For this to be a sensible interpolation scheme we require that when $x_\star$ is one of the

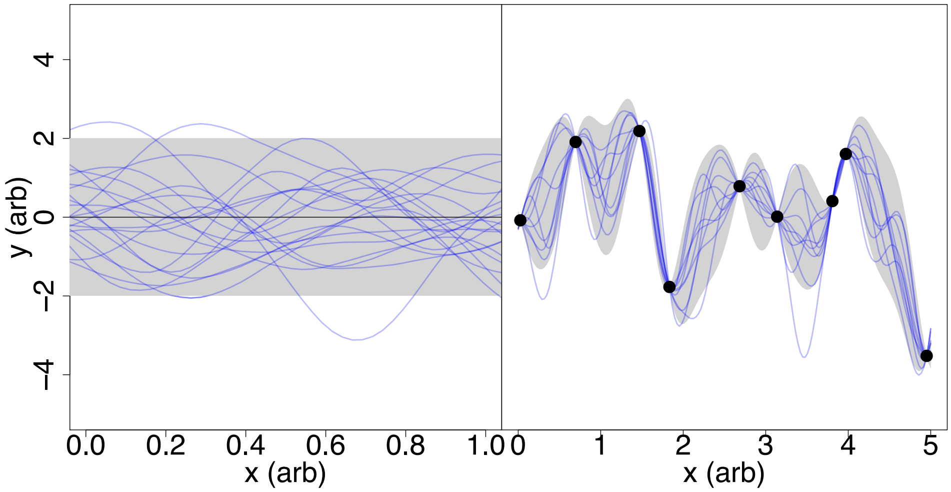


Figure 3.4: Left: draws from a mean zero GP prior with a power exponential covariance function. Right: draws from the posterior distribution after observation of a toy model (solid points). In both panels the gray bands show approximate 95% confidence intervals around the mean. The model function is given by (3.1).

points in the design $\mathcal{D}$ the posterior mean should be the appropriate training value and the posterior variance ought to be zero, since we know the output of the model *with certainty* at this location. Suppose we pick our test point to be the p'th point in our design, then writing

$$\bar{\mu}(x_p) = \mu_p + \sum_{j=1}^{d} \mathcal{A}_j (\mathcal{Y} - \mu_{\bullet})_j, \tag{3.20}$$

$$\mathcal{A}_j = (K_{p,\bullet})_i (K^{-1}_{\bullet,\bullet})_{ij}.$$

We are free to order our basis in the X space any way we like, in this case it is convenient to pick an ordering where p is the final element in the basis, in which case the covariance matrix $K_{\bullet,\bullet}$ has the block form

$$K_{\bullet,\bullet} = \begin{pmatrix} K_{\circ,\circ} & K_{p,\circ} \\ K^{\intercal}_{p,\circ} & K_{p,p} \end{pmatrix} \tag{3.21}$$

where $K_{\circ,\circ}$ $(d-1 \times d-1)$ is the covariance matrix of all the design points apart from the p'th point, &c for $K_{p,\circ}$ $(1 \times d-1)$. Using the Sherman-Morrison-Woodbury inversion formula given in § A.1

$$K_{\bullet,\bullet}^{-1} = \begin{pmatrix} K_{\circ,\circ}^{-1} + \frac{1}{k} K_{\circ,\circ}^{-1} K_{p,\circ} K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1} & -\frac{1}{k} K_{\circ,\circ}^{-1} K_{p,\circ} \\ -\frac{1}{k} K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1} & \frac{1}{k} \end{pmatrix} \tag{3.22}$$

where $k = K_{p,p} - K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1} K_{p,\circ}$. Now we can evaluate $\mathcal{A}_j$, when $j = p$

$$\mathcal{A}_p = \sum_{i=1}^{d-1} \left\{ \mathcal{C}(x_i, x_p) \left(-\frac{1}{k} K_{\circ,\circ}^{-1} K_{p,\circ} \right) \right\} + \frac{1}{k} \mathcal{C}(x_p, x_p),$$

$$= \frac{1}{k} \left(K_{p,p} - K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1} K_{p,\circ} \right) = 1, \quad j = p \tag{3.23}$$

for the other terms $j \neq p$

$$\mathcal{A}_p = \sum_{i=1}^{d-1} \mathcal{C}(x_i, x_p) \left(K_{\circ,\circ}^{-1} + \frac{1}{k} K_{\circ,\circ}^{-1} K_{p,\circ} K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1} \right),$$

$$- \frac{\mathcal{C}(x_p, x_p)}{k} K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1}$$

$$= K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1} + \frac{1}{k} \left(K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1} K_{p,\circ} - K_{p,p} \right) K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1}$$

$$= K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1} - K_{p,\circ}^{\intercal} K_{\circ,\circ}^{-1} = 0. \tag{3.24}$$

This is sufficient to conclude that the posterior mean reverts to the values of the input data set when evaluated at the design values. In a similar manner one can show that the posterior variance vanishes when evaluated at points in the design.

3.5 Observations with noise

We can readily expand the GP regression procedure introduced above to the case where we can only make observations of our model with random noise,

$$Y_m(x) = f(x) + z, \quad \mathbb{R}^n \to \mathbb{R}, \quad z \sim \mathrm{N}(0, \sigma^2). \tag{3.25}$$

This noise is assumed to be constant over the space of model inputs or *homoscedastic* [3]. We evaluate the computer model at design set of d points $\mathcal{D}$ in $\mathbb{R}^n$ obtaining a

[3] There has been significant effort put into developing GP's which can handle observations coming from a varying noise process (called *heteroscedastic* input) for more details see eg [166, 167].

vector of model observations $\mathcal{Y}$. The observation noise process is *a-priori* not spatially correlated $\mathrm{E}[z(x_i)z(x_j)] = \delta_{ij}\sigma^2$, as such we can again write the conditional distribution of our observations $\mathcal{Y}$ given the choice of covariance function, prior mean and design as

$$\mathcal{Y} \mid \mathcal{D}, \mathcal{C}, \mu, \sigma^2 \sim \mathrm{MVN}\left(\mu_\bullet, K_{\bullet,\bullet} + \sigma^2 I_d\right), \tag{3.26}$$

where $\mu_\bullet$ (d) and $K_{\bullet,\bullet}$ $(d \times d)$ have the same definitions as above and the observation error enters only along the diagonal of the covariance matrix. Proceeding as before the posterior mean $\bar{\mu}$ and covariance $\bar{K}$ at some set of k untried locations $X_\star$ given the current set of training observations are

$$\bar{\mu}(X_\star) = \mu_\star + K_{\star,\bullet} \left(K_{\bullet,\bullet} + \sigma^2 I_d\right)^{-1} (\mathcal{Y} - \mu_\bullet), \tag{3.27}$$

$$\bar{K} = \left(K_{\star,\star} + \sigma^2 I_k\right) - K_{\star,\bullet} \left(K_{\bullet,\bullet} + \sigma^2 I_d\right)^{-1} K_{\star,\bullet}^{\intercal}. \tag{3.28}$$

Evaluating the posterior mean at a point x_p in the design will no longer return precisely y_p, considering again the linear mixing term

$$\tilde{\mathcal{A}}_j = K_{\star,\bullet} \left(K_{\bullet,\bullet} + \sigma^2 I_d\right)^{-1},$$

picking our basis so that the point p is the final element, the block form of the covariance matrix is

$$(K_{\bullet,\bullet} + \sigma^2 I_d) = \begin{pmatrix} K_{\circ,\circ} + \sigma^2 I_{d-1} & K_{p,\circ} \\ K_{p,\circ}^{\intercal} & K_{p,p} + \sigma^2 \end{pmatrix}. \tag{3.29}$$

Where we use the same notation $K_{\circ,\circ}$ for the covariance evaluated over the $(d-1)$ element reduced design. The inverse is

$$\left(K_{\bullet,\bullet} + \sigma^2 I_d\right)^{-1} = \begin{pmatrix} \tilde{K}^{-1} + \frac{1}{k'}\tilde{K}^{-1} K_{p,\circ} K_{p,\circ}^{\intercal} \tilde{K}^{-1} & -\frac{1}{k'}\tilde{K}^{-1} K_{p,\circ} \\ -\frac{1}{k'} K_{p,\circ}^{\intercal} \tilde{K}^{-1} & \frac{1}{k'} \end{pmatrix}, \tag{3.30}$$

where $\tilde{K} = (K_{\circ,\circ} + \sigma^2 I_{d-1})$ and $k' = (K_{p,p} + \sigma^2) - K_{p,\circ}^{\intercal} \tilde{K}^{-1} K_{p,\circ}$. For $j = p$ the linear mixing term is

$$\begin{aligned}
\tilde{\mathcal{A}}_{j=p} &= \sum_{i=1}^{d-1} \left\{ \mathcal{C}(x_i, x_p) \left(-\frac{1}{k'} \tilde{K}^{-1} K_{p,\circ} \right) \right\} + \frac{1}{k'} \mathcal{C}(x_p, x_p), \\
&= \frac{1}{k'} \left(K_{p,p} - K_{p,\circ}^{\intercal} \tilde{K}^{-1} K_{p,\circ} \right) = \frac{k' - \sigma^2}{k'}, \\
&= 1 - \frac{\sigma^2}{k'} \qquad (3.31)
\end{aligned}$$

for the other terms $j \neq p$

$$\begin{aligned}
\tilde{\mathcal{A}}_{j \neq p} &= \sum_{i=1}^{d-1} \left\{ \mathcal{C}(x_i, x_p) \left(\tilde{k}^{-1} + \frac{1}{k'} \tilde{K}^{-1} K_{p,\circ} K_{p,\circ}^{\intercal} \tilde{K}^{-1} \right) \right\} - \frac{\mathcal{C}(x_p, x_p)}{k'} K_{p,\circ}^{\intercal} \tilde{K}^{-1}, \\
&= K_{p,\circ}^{\intercal} \tilde{K}^{-1} + \frac{1}{k'} \left(K_{p,\circ}^{\intercal} \tilde{K}^{-1} K_{p,\circ} - K_{p,p} \right) K_{p,\circ}^{\intercal} \tilde{K}^{-1}, \\
&= K_{p,\circ}^{\intercal} \tilde{K}^{-1} \left(1 - \frac{k' - \sigma^2}{k'} \right) = \frac{\sigma^2}{k'} K_{p,\circ}^{\intercal} \tilde{K}^{-1}. \qquad (3.32)
\end{aligned}$$

The resulting deviation from the training value y_p is proportional to σ^2 the prior observation error. A similar analysis shows that the posterior variance at training points is no longer zero. These results are illustrated in Fig: 3.5. Observe how noise in the measurements not only pushes the posterior mean and its confidence intervals away from the training data but also adds an overall local roughness to the draws from the posterior.

3.6 Incorporating an explicit set of basis functions

Suppose that we want to model the mean of the computer model output with some basis of functions $h(x)$, for instance if we were interested in polynomial regression of order r then $h(x) = \{1, x, x^2, \ldots, x^r\}$. We can write our statistical model for the simulator as

$$Y_m(x) = h^{\intercal}(x)\beta + f(x), \quad Y_m : \mathbb{R}^n \to \mathbb{R}, \quad f \sim \text{GP}\left(0, \mathcal{C}\right) \qquad (3.33)$$

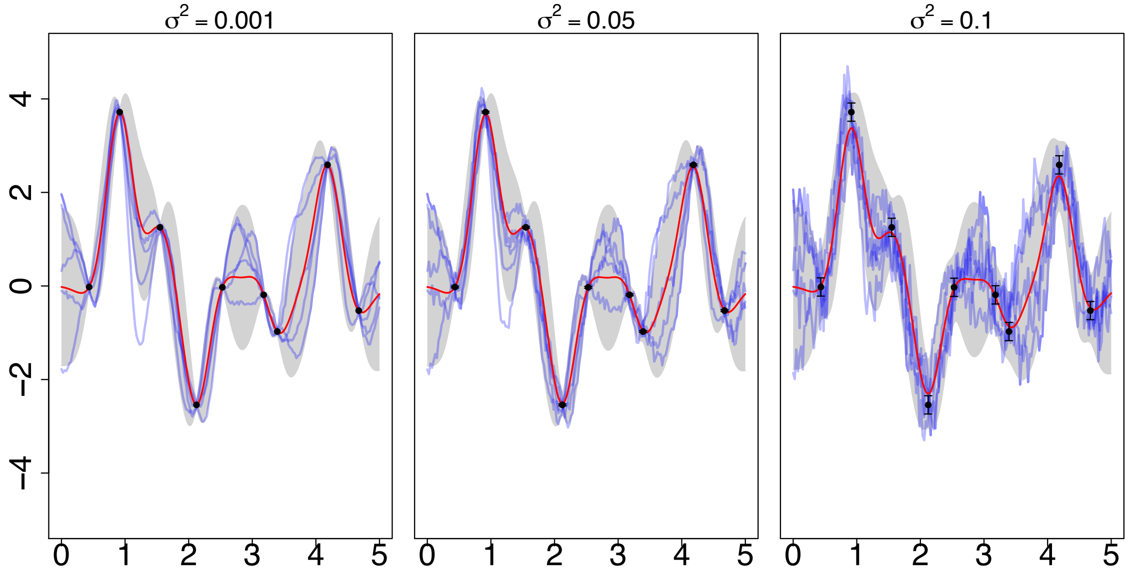


FIGURE 3.5: Draws from the posterior density (blue) after observation of a toy model (3.1) (solid black circles) with varying amounts of observation noise σ. The posterior mean is shown in red, note how the mean along with the draws no longer passes exactly through the training points. The gray region shows a 95% confidence interval around the mean, note how as the observation noise increases the posterior variance at the training points is pushed away from zero.

where we are now modelling the mean with our basis of r functions and some vector of unknown constants β and then modelling the residuals with a Gaussian Process with covariance function $\mathcal{C}$. Taking a normal prior on the parameters $\beta \sim \mathrm{N}(b, B)$ $(r \times 1)$ along with a design over some d points $\mathcal{D}$ and the associated vector of observations $\mathcal{Y}$ then by integrating out β (see (A.17))

$$\mathcal{Y} \mid \mathcal{D}, \mathcal{C}, b, B \sim \mathrm{MVN}\left(H_{\bullet}^{\intercal} b, K_{\bullet,\bullet} + H_{\bullet}^{\intercal} B H_{\bullet}\right). \tag{3.34}$$

where $H_{\bullet}$ $(r \times d)$ is the matrix of the r regression functions evaluated at each of the d design locations. Following the same procedures as above we can obtain the probability distribution for $y_{\star}(x_{\star})$ the computer model output at some unknown location $x_\star$. Plugging into (3.17) and (3.18) we find

$$\begin{aligned}\bar{\mu}(x_\star) &= H_\star^\intercal b + \left(K_{\star,\bullet} + H_\star^\intercal B H_\bullet\right)\left(K_{\bullet,\bullet} + H_\bullet^\intercal B H_\bullet\right)^{-1}\left(\mathcal{Y} - H_\bullet b\right), &(3.35)\\ \bar{K} &= \left(K_{\star,\star} + H_\star^\intercal B H_\star\right)\\ &\quad - \left(K_{\star,\bullet} + H_\star^\intercal B H_\bullet\right)\left(K_{\bullet,\bullet} + H_\bullet^\intercal B H_\bullet\right)^{-1}\left(K_{\star,\bullet} + H_\star^\intercal B H_\bullet\right)^\intercal, &(3.36)\end{aligned}$$

where the convention of stars and bullets is the same as the previous sections. After some algebra and more applications of the SMW matrix inverse formula we obtain

$$\begin{aligned}\bar{\mu}(x_\star) &= H_\star^\intercal \bar{\beta} + K_{\star,\bullet}^\intercal K_{\bullet,\bullet}^{-1}\left(Y - H_\bullet^\intercal \bar{\beta}\right) &(3.37)\\ \bar{K} &= K_{\star,\star} - K_{\star,\bullet}\, K_{\bullet,\bullet}^{-1} K_{\star,\bullet}^\intercal + R^\intercal\left(B^{-1} + H_\bullet K_{\bullet,\bullet}^{-1} H_\bullet^\intercal\right)^{-1} R. &(3.38)\end{aligned}$$

Where the posterior regression coefficient is

$$\bar{\beta} = \left(B^{-1} + H_\bullet K_{\bullet,\bullet} H_\bullet^\intercal\right)^{-1}\left(H_\bullet K_{\bullet,\bullet}^{-1}\mathcal{Y} + B^{-1} b\right), \tag{3.39}$$

and $R = \left(H_\star + H_\bullet K_{\bullet,\bullet}^{-1} K_\star\right)$. Consider taking the limit of $B \to \infty$ in the prior, this corresponds to an infinite prior variance for the fit coefficients. In this limit the posterior regression coefficient becomes

$$\lim_{B\to\infty} \bar{\beta} = \left(H_\bullet K_{\bullet,\bullet} H_\bullet^\intercal\right)^{-1}\left(H_\bullet K_{\bullet,\bullet}^{-1}\mathcal{Y}\right), \tag{3.40}$$

which is the usual Ordinary Least Squares (OLS) form for the fit coefficients in a linear model.

3.7 A confession

I've slightly pulled the wool over your eyes in the preceding sections, all of the above discussion is predicated on knowing the prior mean and covariance functions which correctly describe the underlying Gaussian Process. Where the GP itself is being used to describe the output of our computer model. It's very unlikely that we would actually know this *a priori*. We need to use the output from the simulator in two simultaneous roles, to estimate the parameters of the GP as well as providing the actual data for interpolation. This double dipping actually makes the whole procedure rather non-linear as the choice of covariance structure will now depend upon the observations of the model in some complicated fashion. The extent to which one can really hope to perfectly reproduce the mean and covariance functions of a GP from some finite set of samples is carefully explored by Stein [11].

To ameliorate this problem one typically selects a given functional form or family of functions for the prior covariance and prior mean. This shifts the burden of estimation onto the set of hyper-parameters which describe these functions. In the Bayesian community this is referred to as a hierarchical model.

At this point one can use a maximum-likelihood process to estimate the parameters of the prior mean and covariance given the observations and then take these estimates as certain for the remainder of the analysis, this is commonly referred to as a “drop in” process. Alternatively one can place distributions on these unknown parameters and fold these into the rest of the analysis obtaining a posterior distribution for the GP which fully accounts for the uncertainty in the hyperparameters. This approach is more consistent with a Bayesian philosophy. The GP likelihood surface itself is typically fairly sharply peaked, as such the drop-in approach usually ends up providing a satisfactory treatment of the GP parameters with less complexity than the fully Bayesian approach. The drop in procedure will systematically under represent the amount of uncertainty in the emulator parameters.

3.8 Estimation

In the above sections I have outlined several approaches to "training" a Gaussian Process on a set of observed data $\{\mathcal{Y}, \mathcal{D}\}$ from a computer model $y_m(x)$ so that the resulting posterior mean functions as an interpolating function for the computer model with a concomitant measure of its own uncertainty. As discussed in § 3.7 these methods are predicated upon knowing the right prior mean and covariance functions. By picking certain parameterized functional forms for the prior covariance and mean this problem can be split into two separate issues:

- **Model Selection**: which family of covariance functions, or set of linear model (regression) basis functions is most suitable for describing the data set?
- **Estimation**: given a family of covariance functions and a mean model parameterized by some set of values Θ, which particular values Θ° result in a posterior distribution which best reproduces the true model output $y_m(x)$?

In this section I will discuss the second of these questions. The model selection question is a tricky one as it is highly dependent upon the situation one is trying to model, some of the discussion in § 3.10 is relevant to this question, the practical examples in later chapters will hopefully provide some illumination. Some general advice for model selection follows naturally from linear modelling:

- Plot the data in as many ways as you can. Although typically too complex for publication scatterplot matrices can make a world of difference interms of understanding the co-variation of your data.
- Try and motivate modelling choices by an understanding of the underlying processes leading to the observations,

- Favor moderately good, parsimonious (simple), models over highly specialized ones.

The final point is aimed at avoiding *over-fitting* , an over fitted model leaves very little room for further variation in the sample. This will certainly perform beautifully on the initial set of observations and then most likely totally fail to match any further observations as there is essentially no "slack" left. This can be a serious issue for GP emulators and is a well known problem in machine learning.

Returning to the estimation problem, let us first consider a simple example. In Fig: 3.6 GP regression on a toy model is shown. Here the prior mean was taken to be zero and no additional linear model was enforced, the prior covariance function is of the power-exponential form (3.67) with fixed roughness (for some discussion of this family of covariance functions see § 3.10.1),

$$\mathcal{C}(x_i, x_j; \theta) = \theta_0 \, \exp\left(-\frac{(x_i - x_j)^2}{\delta^2} \right) + \delta_{ij}\sigma^2. \tag{3.41}$$

This covariance function is parameterized by an overall variance θ_0 and a length scale δ (which has dimensions of length), note that I have also included a term σ^2 which only contributes along the diagonal, this so called "nugget" term serves the same role as the observation noise discussed in § 3.5.

In Fig: 3.6 three choices of δ are shown for the same training set. By inspection of the covariance function and through our intuition built up in § 3.4 it's evident that this parameter, which sets the length over which pairs of points in the parameter space have a strong influence upon each other, is going to be pivotal in determining how the GP reproduces the underlying function.

The left panel shows the result of specifying a very short length scale, the resulting posterior distribution is very variable with large uncertainty bands around the mean. Note that a large value of σ is needed to ensure that the resulting covariance matrix is non singular. The right panel shows the result of specifying a longer length scale, again a large amount of measurement error is needed to make this mathematically feasible. However this figure is certainly more intuitively reasonable than the left panel.

The central panel shows the result of specifying a length scale and noise level σ which maximize the likelihood of the posterior, essentially this is a numerically optimal choice of parameters given the data set $\{\mathcal{Y}, \mathcal{D}\}$. Ignoring the mathematical details for a moment it's readily apparent that this reproduces the underlying function rather well. The uncertainty is essentially zero between the sample points and begins to grow at each of the ends of the data range. For a toy model this is quite reasonable, however this kind of overly-confident fit would probably be an underestimate of the true variation of any "interesting" computer model, this illustrates over-fitting fairly well.

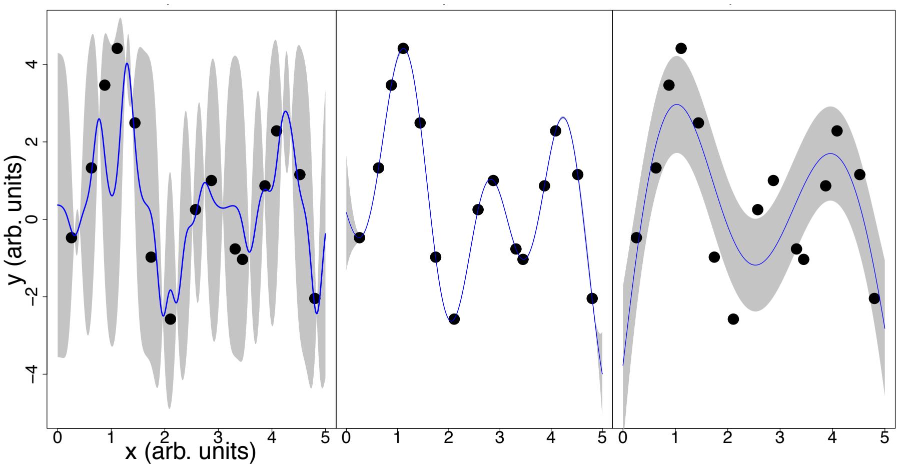


FIGURE 3.6: Varying the characteristic length scale δ in a power-exponential covariance function gives very different posterior distributions. Note that a substantial amount of observation noise σ^2 is present in the right and left hand panels, without this the covariance matrix would be singular.

Now let us return to the concept of maximum likelihood. Given a zero-mean GP prior with some parameterized covariance function $\mathcal{C}(\cdot, \cdot; \Theta)$ then our set of n observations and design $\{\mathcal{Y}, \mathcal{D}\}$ have a joint multivariate-normal distribution, as discussed in detail in § 3.3. The joint distribution of the observations conditioned on the design and the choice and parameterization of the covariance function is

$$\mathcal{Y} \mid \mathcal{D}, \mathcal{C}, \Theta \sim \mathrm{MVN}\left(0, K_{\bullet,\bullet}(\Theta)\right), \qquad (3.42)$$
$$K_{i,j}(\Theta) = \mathcal{C}(x_i, x_j; \Theta), \quad x_i \& x_j \in \mathcal{D}.$$

Given some prior on this set of parameters $P(\Theta)$ the posterior distribution for Θ is

$$P(\Theta \mid \mathcal{Y}, \mathcal{D}, \mathcal{C}) = \frac{P(\mathcal{Y} \mid \mathcal{D}, \mathcal{C}, \Theta) P(\Theta)}{P(\mathcal{Y} \mid \mathcal{D}, \mathcal{C})}, \qquad (3.43)$$

a fully Bayesian approach to GP regression would be to write the probability distribution for the simulator output $y_\star(x_\star)$, evaluated at some new point $x_\star$ as the integral over all possible values of these parameters

$$P\left(y_\star \mid x_\star, \mathcal{Y}, \mathcal{D}, \mathcal{C}\right) = \int P\left(y_\star \mid x_\star, \mathcal{Y}, \mathcal{D}, \mathcal{C}, \Theta\right) P\left(\Theta \mid \mathcal{Y}, \mathcal{D}, \mathcal{C}\right) \, d\Theta. \qquad (3.44)$$

This integral can be approximated relatively efficiently with modern Markov-chain Monte-Carlo (MCMC) methods [4]. We can avoid this added complication for now by making some reasonable approximations. If the posterior distribution of $y_\star$ is sharply peaked around the most probable values of the parameters Θ° then

$$P\left(y_\star \mid x_\star, \mathcal{Y}, \mathcal{D}, \mathcal{C}\right) \propto P\left(y_\star \mid x_\star, \mathcal{Y}, \mathcal{D}, \mathcal{C}, \Theta^\circ\right) P\left(\Theta^\circ \mid \mathcal{Y}, \mathcal{D}, \mathcal{C}\right). \qquad (3.45)$$

This formulation is compatible with the results for the posterior mean and variance given above (i.e. (3.17) & (3.18)) if we evaluate the covariance function with the extremal Θ° as long as the distribution for Θ is sufficiently sharply peaked that $P(\Theta^\circ \mid \mathcal{Y}, \mathcal{D}, \mathcal{C}) \simeq 1$.

To find the most probable parameters Θ° we should find the set of values Θ which maximize the posterior $P(\Theta \mid \mathcal{Y}, \mathcal{D}, \mathcal{C})$ as given in (3.43). This is still a tricky

[4] see [169, 170, 171] for introductions to MCMC procedures

proposition since the term $P(\mathcal{Y} \mid \mathcal{D}, \mathcal{C})$ also requires integrating out all possible values of Θ. However as we are only interested in finding the set of parameters which maximize (3.43) it is sufficient to find values of Θ which maximize a function proportional to the numerator,

$$\mathcal{L}(\Theta) = P(\mathcal{Y} \mid \mathcal{D}, \mathcal{C}, \Theta)P(\Theta) \propto P(\Theta \mid \mathcal{Y}, \mathcal{D}, \mathcal{C}), \tag{3.46}$$

this is the likelihood function. For simplicity we can drop the prior on Θ, explicitly writing out the multivariate normal density for our observations Y we have

$$\mathcal{L}(\Theta) = \frac{1}{2\pi^{n/2}|K(\Theta)|} \exp\left(-\frac{1}{2}\mathcal{Y}^{\intercal}K(\Theta)^{-1}\mathcal{Y}\right). \tag{3.47}$$

This likelihood is to be interpreted as a "score" for a given value of Θ, larger values are better. We will need to numerically maximize this scoring function, i.e. find values of Θ where $\frac{\partial\mathcal{L}}{\partial\Theta_i} = 0$ and $|\frac{\partial\mathcal{L}}{\partial\Theta_i\partial\Theta_j}| < 0$. For the purpose of a numerical treatment it's far easier to consider maximizing the log likelihood

$$\log\mathcal{L}(\Theta) = -\frac{1}{2}\log\det K(\Theta) - \frac{1}{2}\mathcal{Y}^{\intercal}K(\Theta)^{-1}\mathcal{Y} - \frac{n}{2}\log 2\pi, \tag{3.48}$$

as the values of $\mathcal{L}$ are often rather small. The partial derivative of the log likelihood (3.48) with respect to the j'th parameter is readily obtained

$$\begin{aligned}\frac{\partial}{\partial\Theta_j}\log\mathcal{L}(\Theta) &= \frac{1}{2}\mathcal{Y}^{\intercal}\frac{\partial K^{-1}}{\partial\Theta_j}\mathcal{Y} - \frac{1}{2}\mathrm{tr}\left(K^{-1}\frac{\partial K}{\partial\Theta_j}\right), \\ &= \frac{1}{2}\mathrm{tr}\left((\gamma\gamma^{\intercal} - K^{-1})\frac{\partial K}{\partial\Theta_j}\right), \quad \gamma = K^{-1}\mathcal{Y}. \end{aligned} \tag{3.49}$$

In the general case where one has some non trivial prior mean structure as in § 3.6 one has to estimate parameters for the covariance function Θ and the fit coefficients β. Indeed typically one can get away with simply inserting the OLS estimates $\bar{\beta}$ as given in (3.40).

Any numerical scheme for obtaining Θ°, such as conjugate gradients or similar Newton-like methods [172], will necessarily involve evaluating (3.48) and (3.49) several times. This is a computationally costly procedure, dominated by computing the inverse of $K(\Theta)$ which is $\mathcal{O}(n^3)$. Note that in (3.48) and (3.49) the covariance matrix inverse appears as part of a vector matrix product ($\gamma = K^{-1}\mathcal{Y}$). Instead of explicitly computing K^{-1} and then the matrix-vector multiplication for a cost of $\mathcal{O}(n^3) + \mathcal{O}(n^2)$ one should directly solve for γ via a Cholesky or QR decomposition at a cost of $\mathcal{O}(n^3)$ [161, 173].

A practical computational strategy for this maximization process is to first run a Nelder-Mead or similar gradient free routine to obtain a rough candidate local maximum and then use a gradient based method such as BFGS to obtain a precise result [172]. There is little reason to assume that $\log \mathcal{L}$ is globally convex as such this procedure should be run from as many initial conditions as computationally feasible, a process which is a good candidate for a multi-threaded approach.

A significant amount of effort has been put into alleviating the numerical problems arising from large n, typically by attempting to find lower rank approximations to the covariance matrix K see chapter 8 in [12] for a relatively recent review.

Finally a word of warning, as the above derivation should suggest, the actual value of the likelihood function evaluated at the maximum, $\mathcal{L}(\Theta^{\circ})$, is *meaningless* outside of finding the maximum. We threw away the denominator in (3.43) and so there is nothing to sensibly compare with this number.

3.9 Full GP Emulator Specification

In the coming chapters it will sometimes prove useful to have a full description of the probability distribution of the simulator given an emulator with some specified prior mean and covariance function given priors on their parameters. If we want to use the emulator as part of a larger statistical analysis of the behaviour of the simulator we will need the posterior distribution of the model conditioned on our observations, choice of GP prior mean and covariance and the hyper parameters which determine them.

Taking our design $\mathcal{D} = \{x_1, \ldots, x_d\}$ as a set of d points in an n dimensional subset of $\mathbb{R}^n$, denoting our simulator as $Y_m(x)$, and the training set as $\mathcal{Y} = \{Y_m(x_1), \ldots, Y_m(x_d)\}$ then our prior on the model $Y_m(\cdot)$ is a function of the prior mean and variance $m_0(\cdot), V_0(\cdot, \cdot)$

$$Y_m(\cdot) \mid \beta, \sigma^2, \Theta \sim \mathrm{GP}\left(m_0(\cdot), V_0(\cdot, \cdot)\right). \tag{3.50}$$

The prior mean is a linear model $m_0(x) = h(x)^\intercal \beta$ with $h(\cdot) : \mathbb{R}^n \to \mathbb{R}^q$, where q is the number of components in the linear model and β is some set of q unknown fit coefficients. The prior covariance $V_0(x, x') = \sigma^2 \mathcal{C}(x, x'; \Theta)$ has the total scale σ^2 factored out and the covariance function is described by some set of parameters Θ.

According to the prior (3.50) the simulation output vector $\mathcal{Y}$ has conditional distribution

$$\mathcal{Y} \mid \beta, \sigma^2 \sim \mathrm{MVN}_d \left(H_\bullet\, \beta, \sigma^2 K_{\bullet,\bullet}\right) \tag{3.51}$$

where $H_\bullet = (h(x_1), \ldots, h(x_d))^\intercal$ $(q \times d)$ is the matrix of the regression model functions evaluated at each point in the design, and $K_{\bullet,\bullet}$ $(d \times d)$ has elements $K_{i,j} = \mathcal{C}(x_i, x_j; \Theta)$ as before. As above using (3.51) and the results for conditional multivariate normal distributions we obtain the conditional distribution for the simulator $Y_m(\cdot)$ given our observations and the parameterization

$$Y_m(\cdot) \mid \beta, \sigma^2, \Theta, \mathcal{Y} \sim \mathrm{GP}\left(\bar{m_0}(\cdot), \bar{V_0}(\cdot, \cdot)\right) \tag{3.52}$$

where the GP mean and variance are similar to those above

$$\bar{m_0}(x^\star) = h(x^\star)^\intercal \beta + K_{\star,\bullet}^\intercal K_{\bullet,\bullet}^{-1} \left(\mathcal{Y} - H_\bullet \beta\right) \tag{3.53}$$

$$\bar{V_0}(x^\star, x^{\star\prime}) = \sigma^2 \left\{ \mathcal{C}(x^\star, x^{\star\prime}; \Theta) - K_{\star,\bullet}^\intercal K_{\bullet,\bullet}^{-1} K_{\star',\bullet} \right\}, \tag{3.54}$$

where $(K_{\star,\bullet})_{ij} = \mathcal{C}(x_{\star,i}, x_j; \Theta)$. Under the weak prior $p(\beta, \sigma^2) \propto \sigma^{-2}$ using (3.51) we can obtain the posterior for (β, σ^2) which is a normal inverse-gamma distribution. From Bayes theorem

$$f_\beta(\beta \mid \mathcal{Y}, \sigma^2, \Theta) \propto f_{\mathcal{Y}}(\mathcal{Y} \mid \beta, \sigma^2, \Theta) f_\beta(\beta),$$

the PDF for the model output is MVN,

$$f_{\mathcal{Y}}(y \mid \beta, \sigma^2, \Theta) \propto \exp\left\{ -\frac{1}{2} \left(\mathcal{Y} - H_\bullet \beta\right)^\intercal \left(\sigma^2 K_{\bullet,\bullet}\right)^{-1} \left(\mathcal{Y} - H_\bullet \beta\right) \right\},$$

with a little algebra we can re-arrange this and obtain the conditional distribution for β (recall this is a q length vector)

$$\beta \mid \mathcal{Y}, \sigma^2, \Theta \sim \mathrm{N}\left(\hat{\beta}, \sigma^2 (H_\bullet^\intercal K_{\bullet,\bullet} H_\bullet)^{-1}\right), \tag{3.55}$$

$$\hat{\beta} = \left(H_\bullet^\intercal K_{\bullet,\bullet}^{-1} H_\bullet\right)^{-1} H_\bullet^\intercal K_{\bullet,\bullet}^{-1} \mathcal{Y}. \tag{3.56}$$

Note that $\hat{\beta}$ is structurally very similar to the usual least squares estimator for the linear model $\mathcal{Y} = H\beta + \epsilon$. Similarly for the scale σ^2,

$$P(\sigma^2 \mid \mathcal{Y}, \Theta) \propto \int P(\beta \mid \mathcal{Y}, \sigma^2, \Theta) P(\sigma^2) d\beta$$

so the conditional density for σ^2 is

$$\begin{aligned} f_{\sigma^2}(\sigma^2 \mid \mathcal{Y}, \Theta) &\propto (\sigma^2)^{-(1+\frac{d}{2})} |K_{\bullet,\bullet}|^{-1/2} \int \exp\left\{ -\frac{1}{2} (\sigma^2)^{-1} \left(\beta - \hat{\beta}\right)^\intercal \left(H_\bullet^\intercal K_{\bullet,\bullet}^{-1} H_\bullet\right) \left(\beta - \hat{\beta}\right) \right\} \, d\beta \\ &= (\sigma^2)^{-(1+\frac{d}{2})} |K_{\bullet,\bullet}|^{-1/2} \exp(-\frac{2}{\sigma^2}) \exp(-\frac{1}{2} \mathcal{Y}^\intercal K_{\bullet,\bullet}^{-1} \mathcal{Y}) \times \\ &\exp\left(-\frac{1}{2} \hat{\beta} (H_\bullet^\intercal K_{\bullet,\bullet}^{-1} H_\bullet) \hat{\beta} \right) |H^\intercal K_{\bullet,\bullet}^{-1} H^\intercal|^{1/2} (\sigma^2)^{q/2} \end{aligned} \tag{3.57}$$

which is proportional to the PDF of an inverse gamma distribution $f(x, \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp(-\frac{\beta}{x})$. Reading the terms off from (3.57) in this case we have

$$\sigma^2 \mid \mathcal{Y}, \Theta \sim \mathrm{InvGamma}\left(\frac{m-q}{2}, \frac{(m-q-2)}{2} \hat{\sigma}^2 \right) \tag{3.58}$$

where $\hat{\sigma}^2 = \frac{1}{m-q-2}\mathcal{Y}^\intercal \left(K_{\bullet,\bullet}^{-1} - K_{\bullet,\bullet}^{-1} H_\bullet \left(H_\bullet^\intercal K_{\bullet,\bullet}^{-1} H_\bullet \right)^{-1} H_\bullet^\intercal K_{\bullet,\bullet}^{-1} \right) \mathcal{Y}$.

To obtain the final form for our distribution for the model output we would like to eliminate the dependence on the hyper-parameters σ^2, β, Θ. First lets 'average' the conditional posterior for the simulator output over all values of mean coefficients β,

$$P(Y_m(\cdot) \mid \mathcal{Y}, \sigma^2, \Theta) = \int P(Y_m(\cdot) \mid \beta, \sigma^2, \Theta \mathcal{Y}) P(\beta \mid \mathcal{Y}, \sigma^2, \Theta)\, d\beta$$

this is yet another set of Gaussian integrals, after the dust settles we find

$$Y_m(\cdot) \mid \mathcal{Y}, \sigma^2, \Theta \sim \mathrm{GP}\left(\bar{m}_1(\cdot), \bar{V}_1(\cdot, \cdot) \right), \tag{3.59}$$

where the posterior mean $\bar{m}_1$ is structurally unchanged but the coefficients β have been replaced by the estimates $\hat{\beta}$, the posterior variance $\bar{V}_1$ gains terms which account for the variance explained by the linear model on the mean

$$\bar{m}_1(x_\star) = h(x_\star)\hat{\beta} + K_{\star,\bullet}^\intercal(\mathcal{Y} - H_\bullet \hat{\beta}), \tag{3.60}$$

$$\begin{aligned} &\bar{V}_1(x_{\star,\star'}) = \bar{V}_0(x^\star, x^{\star'}) + \\ &\sigma^2 \left[(h(x^\star) - K_{\star,\bullet}^\intercal K_{\bullet,\bullet}^{-1} H_\bullet)^\intercal (H_\bullet^\intercal K_{\bullet,\bullet}^{-1} H_\bullet)^{-1} (h(x^\star) - K_{\star,\bullet}^\intercal K_{\bullet,\bullet}^{-1} H_\bullet) \right]. \end{aligned} \tag{3.61}$$

To obtain the final form we must average over all values of σ^2 using (3.58) and (3.59)

$$P(Y_m(\cdot) \mid \mathcal{Y}, \Theta) = \int P(Y_m(\cdot) \mid \mathcal{Y}, \sigma^2, \Theta) P(\sigma^2 \mid \mathcal{Y}, \Theta)\, d\sigma^2,$$

to this end we obtain the form for the predictive distribution of the model output given our set of observations and the choice of covariance function parameters Θ

$$\eta(\cdot) = Y_m(\cdot) \mid \mathcal{Y}, \Theta \sim \mathrm{StudentProcess}\left(m - q, \bar{m}_1(\cdot), V_1(\cdot, \cdot) \right), \quad V_1 = \frac{\hat{\sigma}^2}{\sigma^2}\bar{V}_1, \tag{3.62}$$

this is how our emulator $\eta(\cdot)$ is defined. Similar to a Gaussian Process a Student-Process is a stochastic process where any finite marginalization of points are jointly distributed with a noncentral Student-t distribution with $m-q$ degrees of freedom.

It turns out that going further to integrate out Θ is typically intractable. We can take some prior $p(\Theta)$ then we can construct the likelihood

$$\begin{aligned} p(\Theta \mid \mathcal{Y}) &\propto p(\Theta) \int p(\mathcal{Y} \mid \beta, \sigma^2, \Theta) p(\beta, \sigma^2) \; d\beta \; d\sigma^2 \\ &\propto p(\Theta) |K_{\bullet,\bullet}|^{-1/2} |H_{\bullet}^{\intercal} K_{\bullet,\bullet}^{-1} H_{\bullet}|^{-1/2} (\hat{\sigma}^2)^{-(m-q)/2}, \end{aligned} \tag{3.63}$$

in theory we would like to compute the posterior distribution for $Y_m(\cdot)$ conditioned only on the training data

$$p(Y_m(\cdot) \mid \mathcal{Y}) \propto \int P(Y_m(\cdot) \mid \mathcal{Y}, \Theta) P(\Theta \mid \mathcal{Y}) \; d\Theta.$$

This integral can be approximated by Monte-Carlo methods, however it is usually sufficient to use maximum-likelihood methods on the log of (3.63) to obtain a set of estimated values $\hat{\Theta}$ (as discussed in § 3.8),

$$\log \mathscr{L}(\Theta) = \log P(\Theta) - \frac{1}{2} \log \det K_{\bullet,\bullet} - \frac{1}{2} \log \det \left(H_{\bullet}^{\intercal} K_{\bullet,\bullet}^{-1} H_{\bullet} \right) - \frac{(m-q)}{2} \log \hat{\sigma}^2 \tag{3.64}$$

The fit coefficients $\hat{\beta}$ and the overall variance $\hat{\sigma}^2$ are functions of the estimated length scales which needs to be taken into account when computing gradients of $\log \mathscr{L}$. Once we have a set of estimated lengths Θ° we can then readily obtain an estimated set of coefficients. One then proceeds by dropping the estimate Θ° into (3.62) (for all practical purposes this means (3.60), (3.61)) and treating this as the full emulator. As far as simulating draws from the emulator we are typically in the limit $n \gg q \simeq 1$ and so we can usually use the methods for making draws from GP's as a reasonable approximation (as developed in § 3.2).

3.10 Covariance Functions

As discussed above the choice of GP covariance function is not entirely free, we are required to select from positive semi-definite functions. It is useful to note that both the sum and product of pairs of positive definite functions are also positive definite, in this way complicated covariance structures with multiple characteristic length scales per dimension can be constructed.

It is sometimes helpful to represent a covariance function as an infinite sum of eigenfunctions ϕ and eigenvalues λ,

$$\mathcal{C}(x_i, x_j) = \sum_{l=1}^{\infty} \lambda_l \phi_l(x_i) \phi_l(x_j), \tag{3.65}$$

where the eigenfunctions are orthogonal $\int \phi_i(x)\phi_j(x)\ dx = \delta_{ij}$. From the positive definite requirement, and symmetry, it's clear that this spectral decomposition will always exist and be real for any finite covariance matrix. In d dimensions the spectral density for an isotropic covariance function $\mathcal{C}(\cdot)$ is

$$g(r) = \int_0^{\infty} r \left(\frac{\rho}{2\pi r}\right)^{d/2} J_{d/2-1}(r\rho)\mathcal{C}(\rho) d\rho. \tag{3.66}$$

3.10.1 Power Exponential

The power-exponential form is the the most commonly used covariance form, this relatively simple form is flexible enough to handle most practical applications.

$$\mathcal{C}(\vec{x}, \vec{y}; \theta, \vec{\delta}, \alpha) = \theta \exp\left(-\frac{1}{2}\sum_{i=1}^{L} \frac{|x_i - y_i|^{\alpha}}{\delta_i^{\alpha}}\right), \quad 1 \leqslant \alpha \leqslant 2. \tag{3.67}$$

The overall variance for the process is set by θ, the scalars δ set the characteristic correlation length scale in each of the dimensions spanned by the parameter space, finally the power α sets the smoothness of draws from the process, for rather technical reasons relating to the spectral properties of the resulting Gaussian process a value just less than 2 is preferred [11].

The spectral density for the limiting cases $\alpha = 1$ and $\alpha = 2$, in d dimensions, can be found to be

$$g(r)_{\alpha=1} = \pi^{-\frac{d}{2}-\frac{1}{2}}\theta\delta^d \left(\frac{1}{r}\right)^{d/2} r^{d/2}\Gamma\left(\frac{d+1}{2}\right)\left(\delta^2 r^2 + 1\right)^{-\frac{d}{2}-\frac{1}{2}} \tag{3.68}$$

$$g(r)_{\alpha=2} = 2^{-d}\pi^{-\frac{d}{2}}\theta\left(\frac{1}{\delta^2}\right)^{-\frac{d}{2}} e^{-\frac{1}{4}\delta^2 r^2} \tag{3.69}$$

where we have isotropized the covariance i.e. $r(x, y) = \|x - y\|$.

The general shape of the power-exponential covariance function and its spectral density are plotted in Fig: 3.7. In Fig: 3.8 draws with two different characteristic length scales δ are shown, this scale is typically set by estimation from the data. The dependence of the process upon the roughness scale is shown in Fig: 3.9. The overall length scale δ is fixed in each panel of this figure, it's clear that this roughness scale introduces variability at a much smaller spatial scale.

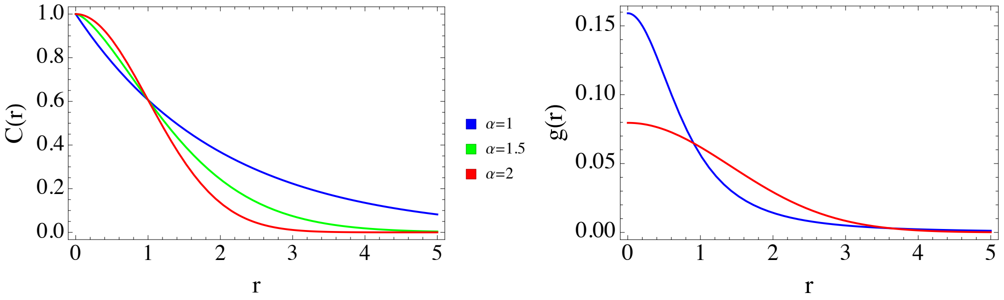


FIGURE 3.7: Left: The shape of the power-exponential covariance function (3.67) for various values of the roughness scale α and fixed values of the length and overall scales δ and θ. Right: The spectral densities for the $\alpha = 1$ (blue) and $\alpha = 2$ (red) limits (3.68) for $d = 2$ with $\theta = 1$ and $\delta = 1$.

FIGURE 3.8: Draws from a mean-zero GP with a power exponential covariance function (3.67), the length scale on the left (blue, $\delta = 0.05$) is significantly shorter than that on the right (red, $\delta = 0.3$) note the increased number of zero crossings. In both panels the overall scale and roughness paramters are fixed to $\theta = 1$ and $\alpha = 1.999$.

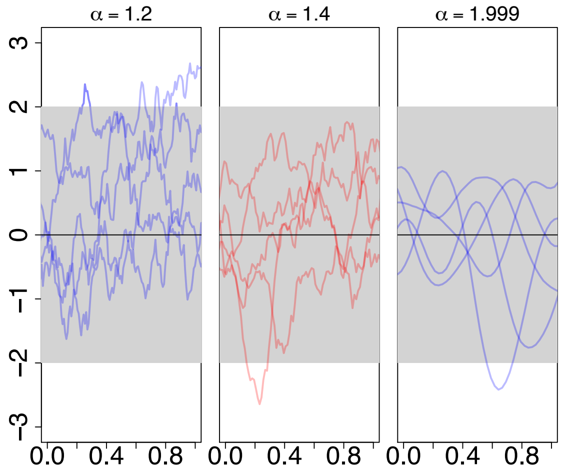


FIGURE 3.9: Draws from a mean zero GP with a power exponential covariance function, the roughness parameter α is varied from left to right. The length scale is fixed $\delta = 0.2$ as is the overall scale $\theta = 1$.

3.10.2 Matern Class

The Matern class is another important form for the prior covariance function, most commonly used in geo-spatial statistical applications. this covariance function is parameterized by a length scale ℓ and a parameter ν which sets the degree of differentiability of the underlying Gaussian Process, again θ parameterizes the overall variance of the process. This is an isotropic function, it only depends on the distance $r = \|x - y\|$ between the two points.

$$\mathcal{C}(r; \ell, \nu) = \frac{2^{1-\nu}\theta}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu r}}{\ell} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu r}}{\ell} \right), \tag{3.70}$$

where K_ν is the modified bessel function of the second kind [174], for half integer values of the order parameter ν we obtain the following simple forms

$$\mathcal{C}\left(r;\ell,\frac{3}{2}\right) = \theta\left(1+\frac{\sqrt{3}r}{\ell}\right)\exp\left(-\frac{\sqrt{3}r}{\ell}\right),$$

$$\mathcal{C}\left(r;\ell,\frac{5}{2}\right) = \theta\left(1+\frac{\sqrt{5}r}{\ell}+\frac{5r^2}{3\ell^2}\right)\exp\left(-\frac{\sqrt{5}r}{\ell}\right).$$

The Matern spectral density in d dimensions is

$$g(r) = \frac{\pi^{-\frac{d}{2}}2^{\nu\theta}\nu^\nu\ell^d}{\Gamma(\nu)}\Gamma\left(\frac{d}{2}+\nu\right)\frac{1}{(\ell^2r^2+2\nu)^{\frac{d}{2}+\nu}}. \tag{3.71}$$

The shape of this covariance function and its spectral representation are plotted in Fig: 3.10. Realizations of a mean zero GP with the Matern class covariance function are shown in Fig: 3.11, for differing values of the roughness/differentiability parameter ν. In the limit that $\nu \to \infty$ the Matern covariance will converge to the squared exponential form (3.67).

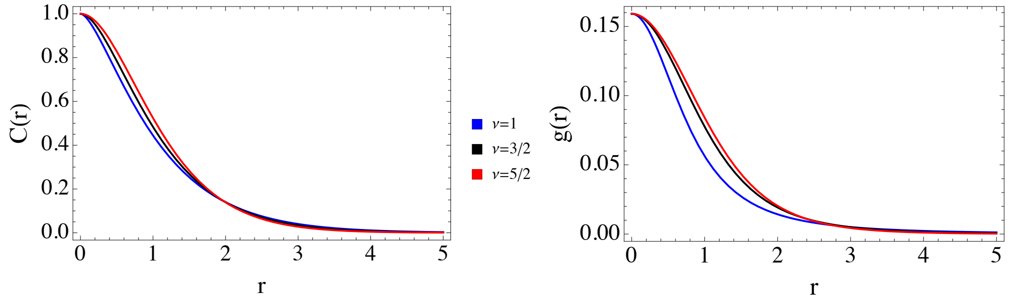


FIGURE 3.10: Left, the shape of the Matern covariance function (3.70) for various values of the differentiability scale ν and fixed $\ell = 1.0$, $\theta = 1$.

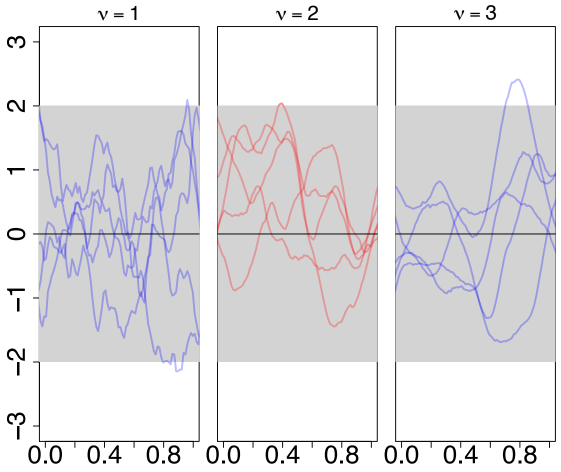


FIGURE 3.11: Draws from a mean zero GP with a Matern covariance function (3.70) with differing values of the parameter ν and fixed $\ell = 0.2$, $\theta = 1$.

4

Practical details for Emulator building

In the previous chapter I introduced the essential concepts and mathematical tools needed to build a Gaussian-Process emulator, which will serve as a statistical approximation to our simulator. In this rather shorter chapter I will cover a few practical details for constructing and using an emulator.

Of primary concern is testing how well the emulator represents the underlying simulator structure. Assessing the validity of any simulator, or a representation of it, in the very literal sense of "*does this particular computer model actually do what I think it should be doing?*" is an important step towards being able to use that simulator to talk about reality. If we do not understand the validity of our emulator formulation we can hardly expect to be able to draw strong conclusions further along the road. The article by O'Hagan et al [175] provides a primary reference for this section, although there are many small discussions of emulator validation scattered throughout the literature it is here that most attention has been paid to the detailed testing and validation of GP emulators.

Also in this chapter I will discuss some of the practical issues related to actually setting up a computer experiment: how many samples of the simulator output are needed and how should these samples be distributed in the parameter space. In the statistical literature these questions fall under the concept "experimental design". In many situations involving the real-life collection and analysis of data it is often possible to propose an experimental design – a scheme for how many conditions to test and how many replicates to make and so forth – which is optimal. This optimality is typically in the sense of requiring the fewest resources while providing the most robust set of answers to whatever questions the designers wish, such as which field and fertilizer combination gives the best yield of crops or which barrel size and chemical combination gives the best pickles [176].

There has been a substantial amount of work put into addressing the design of computer experiments with pioneering work done by Sacks et al [7], this is exhaustively treated in the recent book of Santner et al [18]. Due to the rather complex nature of a GP emulator's posterior distribution (§ 3.9) and the intractability of the estimation process (§ 3.8) these results are typically heuristic rather than provably optimal.

Finally, it is useful to note that we do not have to construct an emulator $\eta(\cdot)$ from the direct model output $\mathcal{Y}$. We can equally well use a transformed set $\mathcal{Y}_g = \{g(Y_m(x_1)), \ldots, g(Y_m(x_d))\}$ where $g(Y)$ is any strictly monotone function. In [32] the authors use $g(Y) = \log(Y + 1)$, this disperses small positive values of the output and reduces the influence of very large values. In this application this log transformation was substantially more stable than an emulator developed from the raw model output.

4.1 "Goodnesss of Fit"

Suppose we have some simulator which produces output $Y_m(x)$ where $x \in \mathbb{R}^n$ is a point in n dimensional parameter space. As discussed in the previous chapter we can construct a GP emulator η for $Y_m(x)$ given some set of d observations $\mathcal{Y} = \{Y_m(x_1), \ldots, Y_m(x_d)\}$ of the model output evaluated according to some design $\mathcal{D} = \{x_1, \ldots, x_d\}$. Let's take the prior mean to be a linear model with some set of q regression functions $h(x) = \{1, x, \ldots\}$ and chose a suitable covariance function parameterized by length scales Θ. Following the methods outlined in the previous chapter we can construct maximum-likelihood estimates Θ° for the correlation lengths and use these to obtain MLE values for the fit coefficients $\hat{\beta}$ and $\hat{\sigma}^2$ overall variance.

At this point we want to try and understand how well our estimated parameters, and in fact our choice of prior mean and covariance functions, work as a description of our simulator. There are two related concepts here:

- **Verification:** does the simulator, and in this case our statistical model of it, do what the designer intended.
- **Validation:** is the simulator a sufficiently accurate representation of the real world.

In this section I will address the question of emulator verification, we will be concerned with quantifying how well our emulator represents the true model output. The question of validation is rather more difficult to address, see the calibration chapter for a detailed discussion.

By construction our GP emulator is modelling a deterministic function, the model output at a given point in the design space is *certain* [1]. A natural conse-

[1] Of course there is the possibility for treating models which produce output with some uncer-

quence of this is that there is no simple concept of residuals, the GP output at a training point is exactly what was put in, as discussed in § 3.4. In the theory of general linear models, of which our GP emulator is a certain limit, there are two kinds of residuals

- **Marginal:** the errors between the observed data and that predicted by the model. This is the typical definition of residuals $y - \hat{\beta}x$.
- **Conditional:** errors between predicted data and observed values *not* used to build the model.

We will naturally be concerned with conditional residuals when analyzing the performance of our GP emulator. If we consider the form for the posterior variance of our emulator predictions at some unknown points (eg (3.18), (3.53)) it's clear that any set of predictions are not going to be statistically independent. This non independence will have to be taken seriously when we are considering these conditional errors.

To construct a set of conditional errors we need an additional set of p points $\mathscr{D}' = \{x'_1, \ldots, x'_p\}$ in the parameter space. These locations should be chosen to span the space with particular emphasis on regions where we are very interested in using the emulator to learn about the model. At each of the locations in $\mathscr{D}'$ we need to observe the simulator $\mathcal{Y}' = \{Y_m(x'_1), \ldots, Y_m(x'_p)\}$ and the emulator $\eta' = \{\eta(x'_1), \ldots, \eta(x'_p)\}$. With these in hand we can construct a series of useful diagnostic quantities.

Practically it is reasonable to take a validation set of perhaps $p \simeq (0.2)d$. After constructing an emulator with the original output data and using this validation set to gauge its performance one may be in one of several situations. If the original

tainty this is still essentially certain data. We have already observed the output and its uncertainty and have constructed our GP emulator accordingly.

emulator performance is generally seen as satisfactory then the validation set can be included with the original training data to build a slightly improved final emulator. The Mahalanobis distance (see § 4.1.2) can be used to ascertain the extent of this improvement. Alternatively if the original emulator is not found to be satisfactory it may be the case that including the validation set into the original training data is sufficient to alleviate this, otherwise more runs are needed. However care is needed here as it is often the case that particular regions in parameter space are the root cause of a bad emulator fit, this can be found by graphical inspection of the various metrics below. In this case it is important to ensure that the locations of the new set of training points are focused in the regions of parameter space where the current emulator performs most poorly.

4.1.1 Individual Prediction Errors

These are the simplest errors we can create,

$$D_i^I(\mathcal{Y}') = \frac{y_i' - \mathrm{E}[\eta(x_i') \mid \mathcal{Y}, \Theta^\circ]}{\sqrt{\mathrm{V}[\eta(x_i') \mid \mathcal{Y}, \Theta^\circ]}}. \tag{4.1}$$

Since the emulator posterior is a StudentProcess (3.62) we expect these errors to be t distributed with $m - q$ degrees of freedom. Further since we've hopefully made a sensible design $m \gg 1$ we can approximate these errors having a standard normal distribution, in this case values of $|D_i^I(y')| \geqslant 2$ indicate a discrepancy between our simulator and the emulator description.

Plotting these against the input parameters x is likely to be useful, for example see Fig: 4.1. We would expect to see all the errors randomly distributed within a horizontal band, significant deviations or structure here may indicate issues with stationarity. Plotting these individual errors against the emulator predictions $E[\eta(x') \mid \mathcal{Y}, \Theta^\circ]$ is likely to be very useful. If for some ranges of the emulator output the errors are consistently of the same sign then there is likely a problem with the mean functions $h(x)$. If any particular errors seem particularly large then it is likely that the overall variance $\hat{\sigma}^2$ has been underestimated. If points which are particularly close to those in the training set exhibit large errors then this indicates a problem with the estimation of the covariance length scales Θ.

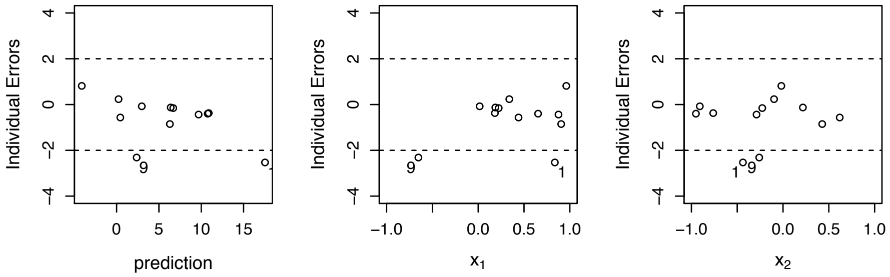


FIGURE 4.1: Individual prediction errors D_i^I plotted for a toy model with $Y_M(x_1, x_2) = 5\exp(-3x_1x_2)\sin(10x_1) + 4$, with $m = 64$, $q = 1$ ($h = \{1\}$) and $p = 12$. From left to right the errors are plotted against the predicted values, the coordinate x_1 and the coordinate x_2. The outstanding points $1, 9$ appear to be located at fairly extreme values of x_1 and this may well be the reason why they perform badly.

4.1.2 Mahalanobis Distance

To obtain a single error statistic we can consider the sum of the squares of the individual prediction errors

$$D_{\chi^2}(\mathcal{Y}') = \sum_{i=1}^{p} D_i^I(\mathcal{Y}')^2 \tag{4.2}$$

this is suggestive of a χ^2 quantity (i.e. $D_{\chi^2}(y') \sim \chi^2_p$), however the errors D_i^I are not actually independent and we shouldn't throw away our knowledge of their correlation structure. Instead we can introduce the "Mahalanobis Distance"

$$D_{MD}(\mathcal{Y}') = (\mathcal{Y}' - \mathrm{E}[\eta(x') \mid \mathcal{Y}, \Theta^\circ])^\intercal \left(\mathrm{V}[\eta(x') \mid \mathcal{Y}, \Theta^\circ]\right)^{-1} (\mathcal{Y}' - \mathrm{E}[\eta(x') \mid \mathcal{Y}, \Theta^\circ]) \,. \tag{4.3}$$

If we write $D_{MD} = \frac{Z}{W}$ where $Z = (\mathcal{Y}' - \bar{m}_1')^\intercal \, (\bar{V}_1')^{-1} \, (\mathcal{Y}' - \bar{m}_1')^\intercal$ and $W = \frac{\hat{\sigma}^2}{\sigma^2}$ (see (3.53)), then $Z \mid \mathcal{Y}, \sigma^2, \Theta \sim \chi^2_p$ and $(m-q-2)w \mid \mathcal{Y}, \Theta \sim \chi^2_{m-q}$ and so since these two variables are independent χ^2 their ratio is F distributed

$$\frac{(m-q)}{p(m-q-2)} D_{MD}(\eta(x')) \mid \mathcal{Y}', \Theta \sim F_{p,m-q}. \tag{4.4}$$

This quantity correctly takes into account the non-independence of our verification data set, large values of (4.3) compared with those expected from (4.4) indicate that there is certainly a discrepancy between the simulator and the emulator. However unlike the individual errors D_i^I we can't really learn much about *where* this problem lies.

4.1.3 Variance Decomposition

Let's define a standard deviation matrix G such that $\mathrm{V}[\eta(x') \mid \mathcal{Y}, \Theta^\circ] = GG^\intercal$, then we can introduce the vector of p transformed errors

$$D_G(\mathcal{Y}') = G^{-1} \left(\mathcal{Y} - \mathrm{E}[\eta(x') \mid \mathcal{Y}, \Theta^\circ]\right), \tag{4.5}$$

the elements of D_G are uncorrelated and are student-t distributed with $m-q$ degrees of freedom, furthermore the Mahalanobis distance can be recovered $D_{MD}(\mathcal{Y}') = D_G(\mathcal{Y}')^\intercal D_G(\mathcal{Y}')$. We can obtain G either by carrying out an eigendecomposition of $\mathrm{V}[\eta(x') \mid \mathcal{Y}, \Theta^\circ]$ or through a regular or pivoted cholesky decomposition. The choice of decomposition which leads to G gives the errors (4.5) subtly different interpretations. The eigendecomposition and pivoted cholesky are most useful, the eigendecomposition errors give linear combinations of validation locations in descending order of predictive error. The pivoted cholesky decomposition permutes the basis of validation points x' into descending order of conditional predictive variance, such that the first point has the largest predictive variance the second point has the largest variance conditioned on the first and so forth. Plots of the individual elements of D_G against both the emulator predictions and the index in the vector are most likely to be useful.

4.2 How many model runs is enough?

The glib answer to the question of "how many model runs to use" is as many as you can afford, which will turn out to be fairly good advice. There are currently no general results for the optimal design of computer experiments, however we can at least heuristically motivate some reasonable lower bounds on how many points to use.

As discussed above one should budget for a set of validation runs which are well dispersed through the parameter space, although this may seems tedious there really is very little one can do with a tool if one has no idea of how well it actually works.

Realistic estimates of how long the simulator takes to run at a given point in the parameter space, along with any additional pre and post processing of data, are essential for developing an appropriate *experimental design*. Here an experimental design is the actual set of points $\mathcal{D}$, however the choices of input parameters and which output quantities are of interest are acutely relevant to this process.

The computational complexity (time and space) of training and using a GP emulator scales with the total number d of design points. The dimension of the input space n itself is really important, the so called *curse of dimensionality* applies here. Consider a hypercube with side length $2r$ and a hypersphere with radius r, the cube's volume in $\mathbb{R}^n$ is $V_{\text{cube}}(n) = 2^n r^n$ while the volume enclosed by the sphere is $V_{\text{sphere}}(n) = \frac{2r^n \pi^{n/2}}{n\Gamma(n/2)}$. The fraction of the volume of our n dimensional hyper-cube which is within the hyper-sphere is

$$\alpha = \frac{V_{\text{sphere}}(n)}{V_{\text{cube}}(n)} = \frac{\pi^{n/2}}{n 2^{n-1}\Gamma(n/2)},$$

and this ratio really doesn't fare well in large dimensions, $\lim_{n \to \infty} \alpha = 0$. As the dimension of a space grows most of the volume ends up in the corners, in four dimensions only a third of the total volume is contained within the unit hyper-sphere.

As a consequence, constructing a GP emulator is likely to become impractical for high dimensional parameter spaces (large n) if there is *significant structure* in all dimensions. However all is not lost, it is frequently the case that the output of computer models with high dimensional parameter spaces is dominated by a small number of parameters which determine most of the structure with the remaining parameters playing minor roles. A sensitivity analysis (see chapter 8) is a good way to begin approaching these kinds of models.

The choice of a relatively smooth prior covariance function reflects our belief that the simulator output itself is fairly smooth across the parameter space. This is a strong assumption and the extent to which it holds is largely responsible for the success or failure of GP emulators of computer models. When this assumption is justified and the prior mean is reasonably well modelled only a relatively small number of model observations are required to establish the characteristic structures, usually far fewer than would be needed to obtain an interpolation of a similar quality using more traditional methods.

The rule of thumb (due to Sacks) which should be thought of as providing a reasonable lower bound for the total number of points is to allocate at least ten points per spatial dimension

$$\min d = 10n \tag{4.6}$$

this "rule" is seriously explored in [21] and found to be fairly reasonable. In Fig: 4.2 the number of design points d for a toy one dimensional model is varied between 6 and 12. For a quantifiable approach to understanding how many training points are sufficient one can consult the statistical and machine-learning literature about "learning curves" for GP regression (see [12, 177, 178]). The primary object of concern here is the generalization error, this is the average over all possible designs of a loss-function typically the L_2 distance between the true function and the GP mean computed over the whole parameter space. Results obtained for these are typically of fairly limited practical use and so not presented here.

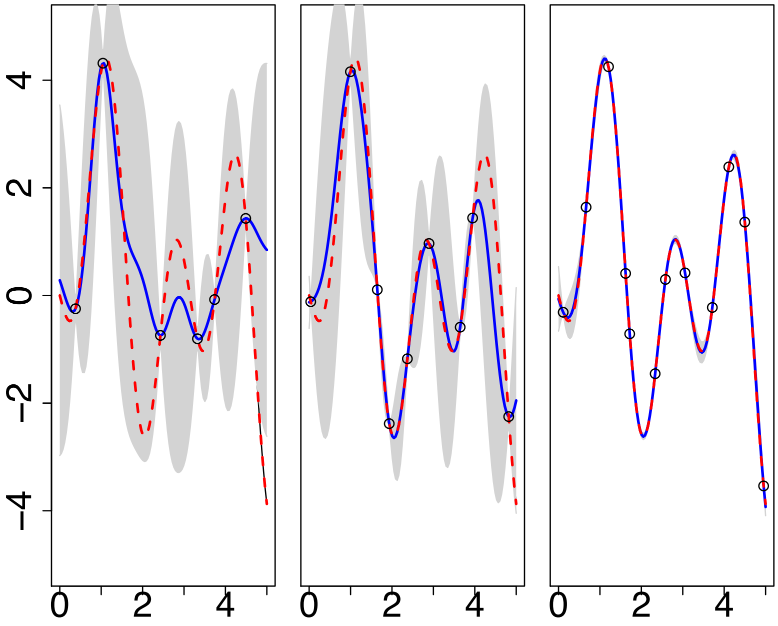


FIGURE 4.2: The panels show the effects of varying the number of design points (model samples), between 6 (left) which is clearly too few, the center panel has 9 design points and the right has 12. Note the reduction in the 95% confidence intervals as the number of design points increases. The red dashed line shows the true model curve given by (3.1), the blue solid line shows the posterior mean of the GP and the open circles show the points where the model function was evaluated.

4.3 Design

Supposing we have selected the n most interesting input parameters to explore and we decide on some number of points d that will be sufficient for at least a first pass, we are then left with the question of how to distribute these d points through our n dimensional space.

Typically the nominal ranges of the model parameters will form some irregular volume $[a_1, b_1] \otimes [a_2, b_2] \otimes \cdots \otimes [a_n, b_n] \subset \mathbb{R}^n$. Naturally the inputs fed to the simulator when making the training set $\mathcal{Y}$ must be within these ranges. However the resulting analysis and emulator construction will be rather easier if we transform these ranges onto the unit hyper cube $[0, 1]^n$. The transformations used to achieve this may be linear in the case of finite $[a, b]$ and will be nonlinear for infinite initial ranges. Standardizing the parameter space in this way is helpful as it places all the parameters on an equal footing with respect to typical length scales. This allows for a ready comparison of the relative sensitivity of the output to variations in each input dimension, which is useful feedback for understanding the model.

As mentioned in the introduction to this chapter the optimal design of experiments has a long and illustrious history. There are formal results for optimal experimental designs for GP emulators in certain limiting cases, these are however somewhat academic [18]. In practice any suitably dispersed simple pattern that spans the design space will probably work fairly well. I will heuristically discuss and illustrate a scheme which is almost always sufficient in practical applications.

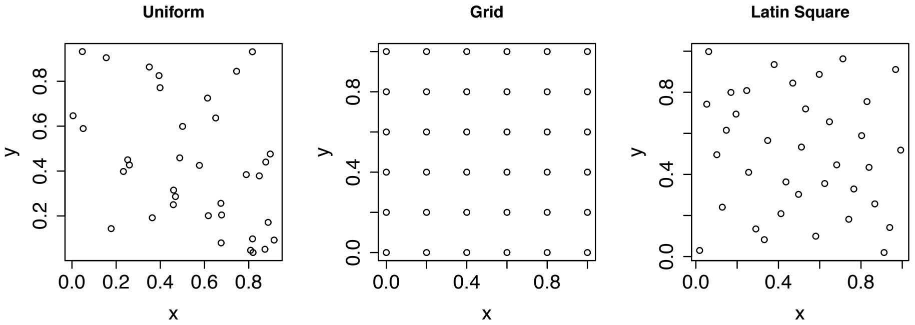


FIGURE 4.3: Examples of various $d = 36, n = 2$ designs. Left: points are iid draws from a two dimensional uniform distribution. Center: the sample points are arranged on a uniform square lattice. Right: a maximin Latin square design.

We can identify two opposing limiting procedures for distributing the d points into our n dimensional space. We could distribute the points completely randomly throughout the space, say by taking the points as a set of d independent uniform samples. This will result in a relatively clumpy distribution with some rather large distance between the points, this is the limit of the least *intentional* structure in the choice of points. The other limit, the scheme with the most intentional structure, would be to arrange the points on some kind of uniform lattice that spans the space. Each of these schemes has strengths and weaknesses and neither is strictly practical, the commonly used schemes represent a compromise between these two limits.

Uniformly and independently distributed sets of random points are surprisingly clumpy, as a result some parts of the parameter space are likely to be under sampled and some will be over sampled. However this clumpiness does have the advantage that a wide variety of length scales of the model will be sampled. A uniform grid with some spacing $a \propto \left(\frac{1}{d}\right)^{1/n}$ will guarantee to fill the space as fairly as possible, no region will be especially over or under sampled. However as a result of the fixed grid this design will only be sensitive to structures in the model output which appear at spatial scales greater than a, since there is no data available to inform about shorter length scales. Furthermore if we believe that the simulator output is well modelled by a *stationary* GP then it is really a huge waste of effort to attempt to uniformly span the space since in this case we only need to gather enough information about the simulator's output to reasonably estimate the characteristic length scales.

To be truly effective the grid design requires a huge number of points which given our strong prior on the smoothness of the model output seems rather wasteful. The uniform random design is appealing as it gives access to a wide range of length scales which will be important when estimating the correlation structure of our simulator. However this comes at the price of wasting effort by creating too many clusters, which may well not even be in interesting parts of the parameter

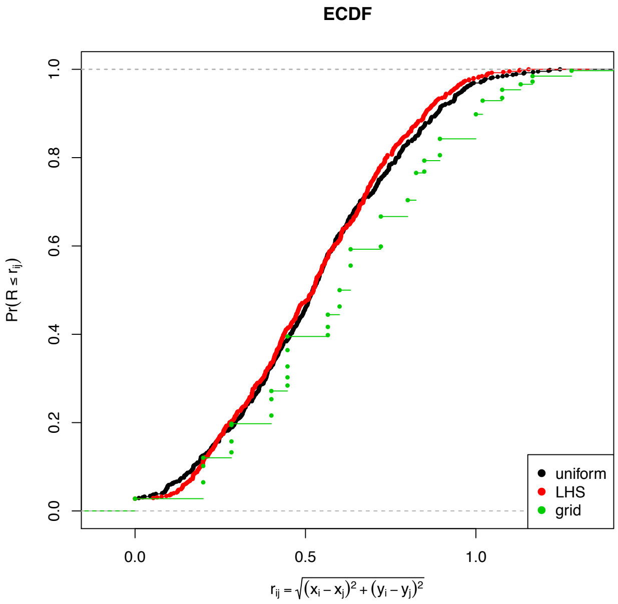


FIGURE 4.4: Empirical CDFs of the euclidean distance between all pairs of points in each of the example designs ($d = 36, n = 2$) shown in Fig: 4.3. The uniform and LHS designs are generally similar, although the propensity for tighter clusters in the uniform design is observable in the differences in the tails of the CDF's.

space, and of simply failing to sample the model at all in others.

To effectively compromise we want a design that has something like the guaranteed coverage of a grid (although it doesn't have to be quite so uniform), with some degree of randomness mixed in so that we are able to effectively sample the simulator's dominant length scales without having to resort to a sampling so dense as to be computationally impractical.

The dominant method for generating designs for computer experiments is *Latin Hypercube Sampling*(LHS) [19, 179, 180]. A Latin square is an $n \times n$ array filled with n symbols such that each symbol occurs exactly n times and exactly once in each row and column, an example with $n = 4$ is

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix}. \tag{4.7}$$

Latin squares are rather interesting beasts in their own right, with many symmetry and invariance properties. For the purposes of LHS we will relax the mathematical definition of a Latin square to being an $n \times n$ grid with n non zero entries arranged such that there is only one non zero entry in each row and column, like this

$$\left(\begin{array}{c|c|c|c} & 1 & & \\ \hline & & 1 & \\ \hline 1 & & & \\ \hline & & & 1 \end{array}\right). \tag{4.8}$$

In Latin Hypercube Sampling we distribute our d points over an n dimensional uniformly spaced grid ($d \times d \times \cdots_{n-3} \times d$) such that every possible two dimensional marginalization of our grid has the relaxed Latin square property, i.e. looks like (4.8). Since each occupied cell in this grid corresponds to some fraction of the total volume of the parameter space the location of the corresponding design point is typically uniformly sampled within this volume. There are several strategies for selecting an optimal (or nearly optimal) LHS design from some ensemble of candidates, again for more detail see [18]. A robust strategy is to generate a moderately large ensemble of candidate LHS designs and select the element which has the largest minimal interpoint distance, i.e. $\max_{\text{designs}}(\min_{\text{points}} r_{ij})$, this is known as a maximin LHS design. An example maximin LHS design is plotted in the right panel of Fig: 4.3.

The array of points produced by an LHS design is much sparser than a full grid design, which would require d^n points, and yet provides a good coverage of the parameter space. This design is particularly good for stationary processes as it incorporates samples at a variety of inter point distances in each dimension. However if there are strong interactions between different parameters, i.e. the model output is dominated by nonlinear terms coupling different dimensions in the parameter space, then this design is likely to be so sparse that it will not be possible to accurately assess the extent of these effects. This would be a *bad thing*.

If more training points are needed, for instance due to poor diagonistics of the current emulator, or if more computational resources become available then there are several methods for efficiently extending these designs [181, 182, 18].

4.4 Step-By-Step Analysis Procedure

This is a sort of checklist for proceeding with an analysis of a model.

1. Learn as much as you can from model developers (if you're not one) about the model, what it aims to achieve and at what level of physical detail/realism.

2. Identify which inputs and outputs are interesting for the problem domain. Make a table of these. Obtain expected ranges for output paramters and sensible ranges for the input parameters. If possible consider splitting input parameters into calibration u and tuning x parameters.

3. Identify which inputs and outputs have a correspondance with reality and obtain appropriate field data if possible.

4. Identify how deterministic the model output is, how many replicates will be needed at each design point to obtain a sensible estimate of the quantity of interest. From this estimate how computationally expensive it will be to obtain the necessary number of replicates at a design point.

5. Plan on carrying out the analysis on a unit hypercube in the parameter space. This makes the interpretation of correlation length scales a lot easier. Typically a linear scaling from the natural ranges of the parameters to the unit cube is sufficient, although it may not be the only way.

6. A minimal rule of thumb is to start with $m = 10d$ points per input dimension [21]. Create an m point d dimensional LHS design $\mathcal{D}$ on $[0, 1]^d$ with as many

points as the computational budget allows. Create a second LHS design with a smaller set of points for use in validation.

7. Run the code, collate the raw output $\mathcal{Y}_r$.

8. Graphically investigate the model output. Make scatter plots of the output against the various parameters along with histograms/boxplots and QQ plots. These will help identify what kind of transformation (if any) is needed so that the GP training set $\mathcal{Y}$ is sufficiently normal (see § A.4). It is often useful to see how much structure there is in the model output and how this structure depends on the parameters. These plots should help inform the choice of prior covariance and mean functions.

9. Center the model output, compute the sample mean $\bar{\mu} = \frac{1}{d}\sum_{i=1}^{d} Y_m(x_i)$ and subtract it from all the elements of the raw output $\mathcal{Y}_r$.

10. It may be appropriate to scale the centered raw output $\mathcal{Y}_r - \bar{\mu}$ so that the final training set is $\mathcal{Y} = \frac{\mathcal{Y}_r - \bar{\mu}}{s}$ (where $s^2 = \frac{1}{d}\sum_{i=1}^{d} \left(Y_m(x_i) - \bar{\mu}\right)^2$ is the sample variance). For simulators which produce univariate data this is typically fine, although extreme outliers may adversely skew this transformation. This scaling makes the specification of priors for the variance, such as the marginal precision λ_m rather simple. For simulators which produce multivariate output scaling each output to unit variance needs careful consideration. If the sample variances between each output variable are rather different and this difference is believed to be physically significant then it may not be sensible to hide this away from the rest of the analysis.

11. Go forth and emulate! Construct a GP and draw samples from it using the methods outlined in chapter 3.

5

Dealing with Multivariate Output

In chapter 3 I outlined how to create a surrogate model, or emulator, which smoothly interpolates the output of a simulator. The results and methods presented are applicable to computer models with a scalar output. In practice interesting computer models typically produce many outputs and we are naturally interested in how each of these outputs varies across the parameter space, furthermore we are typically also very interested in the extent to which these outputs vary together.

In theory one could construct individual GP emulators for each component of the output vector. While conceptually simple this has the distinct disadvantage of potentially requiring a lot of computational work if the number of output components is high, this approach throws away the correlations between the output components across the training set.

It is theoretically possible to define an explicitly multivariate Gaussian Process, however specifying the prior correlation between the outputs becomes tricky as does estimating all the parameters needed to determine this correlation structure. Instead the typical process for dealing with a model with multivariate output is to construct a lower dimensional approximation to the observed data. A popular method for achieving this is the method of *principal component analysis* (or PCA), which constructs a set of orthogonal (and approximately statistically independent) basis functions which describe the observed data and its variability. The training data set is projected into this basis and a set of GP emulators are trained on the resulting weights for each basis component.

5.1 Principal Components

Suppose our computer model produces a set of k different outputs at each point in the parameter space

$$Y_m(x,u) = \left\{Y_m^1(x,u), \ldots, Y_m^k(x,u)\right\},$$

in general these outputs will be somewhat correlated across the parameter space. If we have observations for all k of these outputs we can easily modify the GP emulator framework to treat this multivariate model.

We go about this by constructing a principal component decomposition for our set of model outputs. This defines a projection which rotates the data onto the directions of maximal variation, which are by construction orthogonal. We can then construct GP emulators for the data projected onto each of these directions. Each of these projected directions are approximately independent and so the posterior covariance matrix between our emulators is diagonal. Finally can we rotate the predictions from the PCA basis back into the original or physical space. In this fashion we can compute the mean vector $\hat{\mu}(x,u)$ (k) of our observables at some untested location and also the covariance between them C_Y $(k \times k)$.

To compute the P.C decomposition suppose that we have a set of k dimensional observations obtained by running the multivariate model at some set of d locations in the n dimensional parameter space, i.e. as before our design set is

$$\mathscr{D} = \{(x_1, u_1), \ldots, (x_d, u_d)\},$$

each observation is a k length vector

$$Y_m(x_i, u_i) = \left\{Y_m^1(x_i, u_i), \ldots, Y_m^k(x_i, u_i)\right\}$$

and our training set is $\mathcal{Y} = \{Y_m(x_1, u_1), \ldots, Y_m(x_d, u_d)\}$. From the training set $\mathcal{Y}$ we compute the sample mean vector $\hat{\mu}$ (length k) and the sample covariance matrix $\hat{\Sigma}$ $(k \times k)$,

$$\hat{\mu}_\alpha = \frac{1}{d} \sum_{i=1}^{d} Y_m^\alpha(x_i, u_i), \tag{5.1}$$

$$\hat{\Sigma}_{\alpha,\beta} = \frac{1}{d} \sum_{i=1}^{d} \left(Y_m^\alpha(x_i, u_i) - \hat{\mu}_\alpha\right) \left(Y_m^\beta(x_i, u_i) - \hat{\mu}_\beta\right)^\intercal. \tag{5.2}$$

An eigendecomposition of the matrix $\hat{\Sigma}$ defines our principal component decomposition

$$\hat{\Sigma} = U \Lambda U^\intercal, \tag{5.3}$$

here U $(k \times k)$ is a matrix whose columns are the eigenvectors of $\hat{\Sigma}$ and $\Lambda_{ij} = \delta_{ij}\lambda_i$ is a diagonal matrix of eigenvalues sorted in descending order. The trace of Λ corresponds to the total sample covariance of our observations $\mathcal{Y}$. As such each eigenvalue represents the covariance contribution of its associated eigenfunction to the observed total covariance. This decomposition identifies the direction in the space spanned by our data $\mathcal{Y}$ which corresponds to the largest observed variation and the remaining eigenvectors correspond to orthogonal directions with successively smaller amounts of variation. Each additional eigen-component describes a lesser amount of the sample variation which is orthogonal to all the others. The eigenvectors U describe the rotation from our observations into the P.C space. The set of k $d-$dimensional vectors

$$Z_k(x_i, u_i) = \frac{1}{\sqrt{\hat{\lambda}_k}} \hat{u}_k^{\intercal}[Y_m(x_i, u_i) - \hat{\mu}], \tag{5.4}$$

where $\hat{u}_k$ is the k'th eigenvector and λ_k is the k'th eigenvalue, represent the projection of our original set of d $k-$dimensional model observation vectors into our orthogonal P.C space. Each of the k vectors Z_k is then used as an input for a single GP emulator, which is otherwise constructed exactly as described in chapter 3. Each of these emulators now interpolates the weights $Z_k(x, u)$ and can be made to give predictions at untrained locations as before.

The PC rotated weights Z_k are statistically independent if the original data $\mathcal{Y}$ has a multivariate normal distribution, in this case the covariance $\mathrm{Cov}[\mathrm{Z_i}, \mathrm{Z_j}] = \lambda_{\mathrm{i}}\delta_{\mathrm{ij}}$ is diagonal by construction. This independence underlies our ability to construct individual scalar GP emulators for each of the Z_k.

It's important to realize that the sample outputs of a real computer model generated by some design that spans the parameter space *may not be particularly normally distributed*. This needs to be explicitly tested and addressed on a case by case basis. To test for normality one can construct so called "QQ" plots and compute the squared distances which should be χ^2 with degrees of freedom equal to the rank of $\hat{\Sigma}$ see § A.4. Often the sample data can be transformed to improve normality, for instance a square-root transformation in the case of count data. For more information see [183]. In the case of sample non-normality the potentially non trivial higher moments of the sample distribution will not be removed by the PC transformation, leading to some amount of residual correlation between the variables which is neglected in the remainder of the analysis leading to a less faithful multivariate emulator.

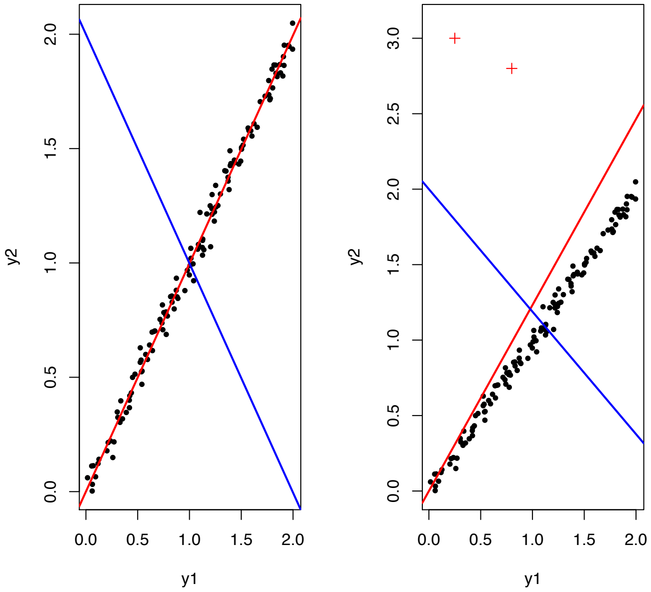


FIGURE 5.1: A toy example of a principal component decomposition, showing the potentially skewing influence of outliers. The solid black circles show 128 samples from a toy model with $y_2 = y_1 + \delta$ where is a mean zero normally distributed random variable with standard deviation 0.05. The red and blue lines show the two principal directions (eigenvectors). In the left panel $\lambda_1 = 0.998$ and $\lambda_2 = 0.0018$, the first principal component (red) explains essentially all of the variation in the sample, as we would expect. In the right panel outlying data points (red) have been added to the data set. As a result the two principal directions which are now skewed. Also now $\lambda_1 = 0.904$ and $\lambda_2 = 0.095$, the contribution of the second direction to the variance decomposition has erroneously become enlarged.

We can rotate our predictions back into the 'physical' space, in general

$$Y_M(x_i, u_i) = \hat{\mu} + U\sqrt{\lambda}Z(x_i, u_i), \tag{5.5}$$

where $Z(x_i, u_i) = \{Z_1(x_i, u_i), \ldots, Z_k(x_i, u_i)\}$ is a vector of the P.C rotated quantities, the expectation of the emulator in the P.C space is the same as before and so

$$\mathrm{E}[Y_M(x_i, u_i)] = \hat{\mu} + U\sqrt{\lambda}\mathrm{E}[Z(x_i, u_i)], \tag{5.6}$$

where

$$\mathrm{E}[Z(x_i, u_i)] = \{\mathrm{E}[Z_1(x_i, u_i)], \ldots, \mathrm{E}[Z_k(x_i, u_i)]\},$$

$$\mathrm{E}[Y_M(x_i, u_i)] = \left\{\mathrm{E}[Y_M^1(x_i, u_i)], \ldots, \mathrm{E}[Y_M^k(x_i, u_i)]\right\}.$$

The estimated covariance between the components of $Y_m(x_i, u_i)$ can also be obtained at a fixed location x_i, u_i

$$\mathrm{Cov}[\mathrm{Y^l_m(x_i, u_i), Y^j_m(x_i, u_i)}] = \sum_{\alpha,\beta,\gamma=1}^{\mathrm{k}} \mathrm{U_{l\alpha}}\Lambda^{1/2}_{\alpha\beta}\mathrm{U_{j\gamma}}\Lambda^{1/2}_{\gamma\beta}\mathrm{Var}[\mathrm{Z}_\beta(\mathrm{x_i, u_i})], \tag{5.7}$$

where $\mathrm{Var}[Z_\beta(x_i, u_i)]$ is the variance of the β'th principal component weight GP emulator at x_i, u_i as given by (3.18). We can also estimate the covariance between two locations in the parameter space $(x_i, u_i), (x_j, u_j)$ and between two different observables Y^α_m, Y^β_m as

$$\begin{aligned}
\mathrm{Cov}[Y^\alpha(x_i, u_i), Y^\beta(x_j, u_j)] &= \mathrm{E}[Y^\alpha(x_i, u_i)Y^\beta(x_j, u_j)] - \mathrm{E}[Y^\alpha(x_i, u_i)]\mathrm{E}[Y^\beta(x_j, u_j)], \\
&= U_{\alpha\delta}\Lambda^{1/2}_{\delta\gamma}U_{\beta\epsilon}\Lambda^{1/2}_{\epsilon\chi}\mathrm{Cov}[Z_\gamma(x_i, u_i), Z_\chi(x_j, u_j)], \\
&= U_{\alpha\delta}\Lambda^{1/2}_{\delta\gamma}U_{\beta\epsilon}\Lambda^{1/2}_{\epsilon\chi}\delta_{\gamma\chi}\mathrm{Cov}[Z_\gamma(x_i, u_i), Z_\chi(x_j, u_j)].
\end{aligned}$$

Where we have used the independence of the P.C space to set $\mathrm{Cov}[Z_\gamma(x_i, u_i), Z_\chi(x_j, u_j)] = \delta_{\gamma\chi}\mathrm{Cov}[Z_\gamma(x_i, u_i), Z_\chi(x_j, u_j)]$.

5.2 Dimensional Reduction

We have described how to use a principal component decomposition to construct an orthogonal basis for a set of potentially correlated data. If k is very large, for instance if our vector of model outputs corresponds to some discrete sampling of a continuous process such as a time series, then it becomes painful to construct and sample all k emulators. In this case it is usual to retain only the first r components of the P.C decomposition, by construction these are the largest contributors to the observed variation in the input data. This process is often referred to as "Dimensional Reduction" or as "finding a low rank approximation" to $\hat{\Sigma}$.

Typically r is selected to reproduce some large fraction of the sample variance usually around 95%, this value of r can be approximated by solving

$$V(r) = \sum_{i=1}^{r} \frac{\lambda_i}{\mathrm{Tr}\Lambda} = 0.95, \tag{5.8}$$

for r. Naturally by selecting $r < k$ we have lost some of the original information about our sample set $\mathcal{Y}$, however with judicious choice of r this is usually not a serious issue. It is often very useful to plot $V(r)$ a so called scree plot. Inspection of such plots gives a good visual indication of how well a PCA dimensional reduction approach will work. If the plot saturates quickly then only a small number of components will be needed to reproduce the most important parts of the variation of the model output across the training set.

A set of training data from a simple multivariate model is shown in the left panel of Fig: 5.2, the toy model here is

$$y_m(x, u_1, u_2) = 5 \exp\left(-3 u_1 x\right) \sin(10x) + 2u_2, \tag{5.9}$$

where u_1, u_2 are interpreted as calibration parameters and x is an index that picks out the different elements of the model output. I have discretized x into $k = 128$ uniformly spaced sample locations on $[0, 2]$, the values of the calibration parameters are sampled from a maximin LHS with $d = 64$ (see § 4.3). This kind of high dimensional functional model output might represent a time-series or the bins of a histogram. This training data set clearly has a lot of structure in the functional dimension x, by inspection we would expect that we should be able to pick an $r \ll k = 128$. The right hand panel in Fig: 5.2 shows the first few P.C basis functions (eigenvectors of the sample covariance matrix), the legend gives the associated standard deviations (square roots of the eigenvalues) for these components. Examining the scree plot and the eigenvalues makes it clear that taking $r = 3$ would give a fairly faithful reproduction of the input data, one could perhaps make a case

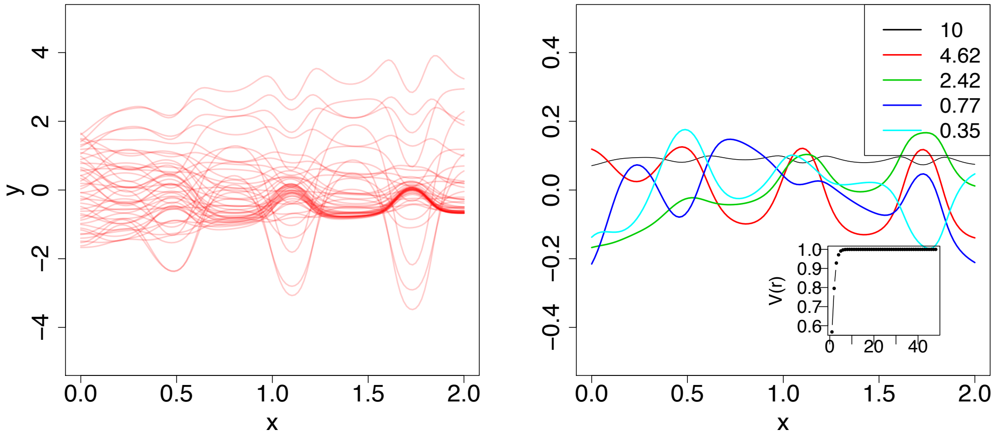


FIGURE 5.2: Left: Training data $\mathcal{Y}$ for the toy functional model (5.9) with $k = 128$ and $d = 64$. Right: The five most significant eigenvectors of the sample covariance matrix of $\mathcal{Y}$, the legend gives the standard deviation associated with each component. The inset figure shows the cumulative variance explained by the eigenvectors $V(r)$ (5.8).

for including up $r = 5$ but any additional components are likely to add no further information. This is a substantial reduction from the naïve case of constructing $k = 128$ GP emulators.

The observed correlations in the data provide a low-rank approximation to the full sample covariance matrix. However if the scree plot saturates very slowly then there may not be a suitable lower dimensional representation. For a nice treatment of the analysis of scree-plots and other PCA related diagnostics consult [184, 183].

After determining r one proceeds as above, but with the eigendecomposition matrices truncated $U = U_r$ $(k \times r)$ and $\Lambda = \Lambda_r$ $(r \times r)$ as such one obtains a truncated vector $Z(u_i) = \{Z_1(u_i), \ldots, Z_r(u_i)\}$. There are other methods of dimensional reduction (such as wavelets etc), however it can be shown that the truncated P.C decomposition is the highest fidelity *linear* transformation, we lose the least information by making this rotation.

5.3 Principal Pitfalls

Finally, it is important to note that the presence of outlying data points in a sample set can have a very strong influence on the resulting P.C basis and weights, see Fig: 5.1 for a toy example. Here a two dimensional data set with a strong linear correlation between the two variables is decomposed (left panel). In this case the two principal directions could be deduced by inspection, the first direction (red) is responsible for the vast majority of the variation observed in the sample $\mathcal{Y}$. The right hand panel shows the same data set with the addition of two rather exaggerated outliers, plotted as red crosses. These two outlying points strongly skew the two principal directions and push the variance explained by the second direction up to almost 10%. This sensitivity makes a blind application of these multivariate methods somewhat unadvisable.

To make this more precise let's consider the change in the decomposition induced by adding a new (and outlying) observation $Y_{\text{out}} = \left\{Y^1_{\text{out}}, \ldots, Y^k_{\text{out}}\right\}$, we can compute the new sample mean and covariance as updates to (5.1),

$$\hat{\mu}'_\alpha = \frac{1}{d+1}\left(\sum_{i=1}^{d} Y^\alpha(x_i, u_i) + Y^\alpha_{\text{out}}\right) = \frac{d}{d+1}\hat{\mu}_\alpha + \frac{1}{d+1}Y^\alpha_{\text{out}}, \tag{5.10}$$

$$\begin{aligned}
\hat{\Sigma}'_{\alpha,\beta} &= \frac{1}{d+1}\left\{\sum_{i=1}^{d}\left(Y^\alpha_m(x_i, u_i) - \hat{\mu}'_\alpha\right)\left(Y^\beta_m(x_i, u_i) - \hat{\mu}'_\beta\right) + \left(Y^\alpha_{\text{out}} - \hat{\mu}'_\alpha\right)\left(Y^\beta_{\text{out}} - \hat{\mu}'_\beta\right)\right\}, \\
&= \frac{d}{d+1}\hat{\Sigma}_{\alpha,\beta} + \frac{1}{(1+d)^3}\left(Y^\alpha_{\text{out}}\left(\left(d^2+1\right)Y^\beta_{\text{out}} - 2d^2\hat{\mu}_\beta\right)\right. \\
&\left.+\hat{\mu}_\alpha\left(\left(3d^2-1\right)\hat{\mu}_\beta - 2d^2 Y^\beta_{\text{out}}\right)\right)
\end{aligned} \tag{5.11}$$

Recalling results from elementary linear algebra [161, 185], we could compute the compute the corrections to our original eigendecomposition of $\hat{\Sigma}$ perturbatively (in $\epsilon = \frac{1}{d}$ and taking the limit that d is large). Writing

$$\hat{\Sigma}' \approx \hat{\Sigma} + \epsilon V + \mathcal{O}(\epsilon^3)$$

where the elements of the perturbing matrix V are

$$V_{\alpha,\beta} = \left(-2\hat{\mu}_\beta Y^\alpha_{\text{out}} - 2\hat{\mu}_\alpha Y^\beta_{\text{out}} + Y^\alpha_{\text{out}} Y^\beta_{\text{out}} + 3\hat{\mu}_\alpha \hat{\mu}_\beta\right). \tag{5.12}$$

Writing the eigenvalues and eigenvectors of $\hat{\Sigma}'$ $(k \times k)$ as a power series $\lambda'_i = \lambda^0 + \epsilon\lambda^1 + \ldots$, $u'_i = u^0_i + \epsilon u^1_i + \ldots$, with the unperturbed values as the lowest order terms we obtain the usual results for the first order corrections in to the eigenvalues and eigenvectors

$$\lambda^1_j = u^{0\intercal}_j V u^0_j, \quad u^1_j = \sum_{\ell=1,\ell\neq j}^{k} \frac{u^{0\intercal}_\ell V u^0_j}{\lambda^0_j - \lambda^0_\ell}, \tag{5.13}$$

after inserting the series expansions into the definition of the eigendecomposition and matching terms order by order. The shift in the eigenvalues and vectors is linear in the matrix elements $V_{\alpha,\beta}$ (as it must be at first order in the expansion). We can easily see the skewing influence of the outlying points by slightly re-writing (5.12)

$$V_{\alpha,\beta} = \left\{2\hat{\mu}_\beta \left(\hat{\mu}_\alpha - Y^\alpha_{\text{out}}\right) + 2\hat{\mu}_\alpha \left(\hat{\mu}_\beta - Y^\beta_{\text{out}}\right) + \left(Y^\alpha_{\text{out}} Y^\beta_{\text{out}} - \hat{\mu}_\alpha \hat{\mu}_\beta\right)\right\}. \tag{5.14}$$

It's clear that these matrix elements are directly proportional the difference between the sample mean values of a given output component $\hat{\mu}_\alpha$ and the associated component of the additional data point Y^α_{out}. If the point truly is outlying then it will exert a strong pull on the PCA decomposition that is proportional to the extent to which it outlies the main trend in the data.

6

An Example Analysis: ChemTreeN

What are the stars but points in the body of God where we
insert the healing needles of our terror and longing?

To illustrate some of these ideas let's look at a detailed example of an analysis of a model with a significantly multivariate output. In [4] the authors (myself included) investigate the hybrid galaxy formation model chemtreeN, for a detailed description of the physics included in the model see [186, 187]. Fiducial points are selected in the calibration parameter space, the model output at these points is used as artificial field data. We construct a series of GP emulators based upon different subsets of the model outputs and use these to explore how well the fiducial locations in the parameter space can be reconstructed.

Galaxies like the Milky-Way have complicated formation histories, the underlying dark matter mergers between evolving galaxies and the capture of smaller galaxies play a rôle alongside the stellar chemistry which determines the stellar content of the galaxies as we observe them. Because we are embedded in it, information about the physical properties of the Milky Way can be measured at an exquisite level of detail. Recent studies seem to indicate that our Galaxy may not be a typical galaxy after all. For example, observations of a large sample of the Sloan Digital Sky Survey (SDSS) galaxies have shown that the Milky Way has significantly more satellites than a typical galaxy of its luminosity

The model ChemTreeN layers stellar chemistry over a given set of dark matter merger dynamics. We selected several reasonable candidate dark matter histories for the Milky Way and systematically calibrated the model to best reproduce observable features of the Milky Way such as its population of satellite galaxies. The computational overhead for such an analysis would have been prohibative without surrogate models of the computer codes.

Surprisingly the resulting sets of best fit parameters, which determine the evolution of the baryonic components of our Milky Way-like galaxy, obtained from each candidate merger history were found to be strikingly inconsistent. The details of the dark matter history must play a important rôle in galaxy formation. This exercise provided an interesting and new insight into how the dark-matter merger history of different candidate galaxies influences the full galaxy formation process.

6.1 An Introduction to the Problem Domain

Understanding the formation and evolution of galaxies is a central and long-standing problem in astrophysics. Over the past century, and particularly in the past decade, a tremendous amount of information has been gleaned about populations of galaxies and their temporal evolution, and data have been collected on galaxies spanning more than six orders of magnitude in stellar mass and over thirteen billion years in the age of the Universe. These observations show that the galaxies that we can see have undergone radical changes in size, appearance, and content over the last thirteen billion years [188, 189, 190, 191]. Complementary observations have provided a rich data-set on the kinematics and elemental abundances of stars in our own Milky Way, including large numbers of metal-poor stars in the halo of our own galaxy and in local dwarf galaxies. In principle, this 'galactic fossil record' can probe the entire merger and star formation history of the Milky Way and its satellites, and complement direct observations at higher redshifts.

The quantity and quality of observational data on galaxy formation, which is already staggering, is going to increase exponentially over the next decade. Surveys such as LAMOST [192], SkyMapper [193], Gaia [194], and, ultimately, the Large Synoptic Survey Telescope [195] will produce petabytes of data on billions of individual objects, both galactic and extra galactic, that will strongly inform our understanding of galaxy behavior.

Despite this wealth of observational information, we currently lack the detailed and self-consistent theoretical models necessary to adequately interpret such observational data sets. Purely analytic (i.e., "pencil-and-paper") theoretical models are insufficient to address the questions that are currently being asked about galaxy formation, due in no small part to the range of physical components that must be simultaneously modeled (e.g., gravity, dark matter, gas dynamics, radiative cooling, star formation and feedback), and the complex and nonlinear coupling of these components. As a result of these complications, two separate theoretical methods are commonly used to study galaxy formation: multiphysics hydrodynamical simulations and semi-analytic models.

Multi-physics numerical simulations are typically used to model galaxy formation by implementing all of the relevant physical processes in as realistic a manner as is technically and computationally feasible. These calculations are typically based on N-body dark matter dynamics simulations of cosmological structure formation, and include gas dynamics, the radiative cooling and heating of gas, models for star formation and feedback, and possibly more complex physics such as magneto-hydrodynamics, radiation transport, and the formation of, and feedback from, super massive black holes. Commonly-used codes of this type include Enzo

[196, 197], Gadget [198], Gasoline [199], RAMSES [200], and more recently AREPO [201]. These codes produce broadly similar results, although some important differences remain to be resolved [202, 203, 204, 205].

The main advantage of such calculations is that they attempt to faithfully reproduce the relevant physical processes in as accurate of a manner as possible, and by virtue of their construction automatically include any complex, nonlinear interaction between important physical processes. The main disadvantage of this sort of simulation lies in its cost: current-generation calculations of a single Milky Way-like galaxy performed at high (~ 100 pc) spatial resolution [e.g. 206] can easily consume hundreds of thousands of CPU hours and months of time to complete, making it challenging to model statistically-significant numbers of galaxies or to perform a meaningful study of variations in free parameters within the models, even with the methods discussed in this thesis.

A second approach is often referred to as "semi-analytic" or "phenomenological" modeling of galaxy formation. This type of model typically is based upon either the extended Press-Schechter formalism or N-body cosmological simulations, which provide the evolutionary histories for a population of galaxies. Prescriptions are then applied on top of these evolutionary histories to describe the behavior of the gas and stellar populations contained within, and surrounding, the dark matter halos that drive dynamics on large scales, as well as the observational properties of the resulting galaxies. These models are then calibrated by comparison to some set of observations. Some examples of this sort of model include GALFORM [207, 208, 209], Galacticus [210], and ChemTreeN [186, 187].

Two important strengths of this type of model are flexibility and speed: one can easily implement variations on a model (gas ejection from galaxies as a function of halo mass and redshift versus a constant value) and then see within minutes how this affects the modeled population of galaxies. The disadvantages of this modeling technique include the large number of free parameters and the extent to which the observable properties of simulated galaxies depend on the models of specific physical phenomena, such as the behavior of galaxies during mergers. Even with these substantial downsides, however, semi-analytic models are incredibly useful for exploring the consequences of various physical phenomena on the observable properties of galaxies.

We combine semi-analytic models of the formation of the Milky Way (including several different N- body simulation-based merger histories) with modern statistical techniques to explore how one might meaningfully constrain the formation of the Milky Way's stellar halo and population of satellite galaxies both from a theoretical standpoint and in terms of guiding future observations.

6.2 The Model – Input and Output

ChemTreeN belongs to the class of semi-analytic galaxy formation models mentioned above. The chemical processes of galaxy formation, here chemical can typically be read as nuclear-astrophysical, are described phenomenologicaly by the model through a series of differential equations [186, 187]. The model takes as primary input a (cosmological) series of snapshots of the state of an N-body simulation of the purely gravitational interaction of a primordial distribution of dark matter. This dark-matter history, where fluctuations in the initial dark matter distribution evolve to form gravitationally bound clumps and eventually merge into galactic scale objects, forms the backbone of the simulation. These bound clumps are referred to as halos, the halos are individually tracked throughout the evolution along with their merger into larger halos or their consumption of smaller ones. These merger histories are computationally very expensive to obtain as they currently require super-computer level resources to obtain a reasonable resolution of the simulated cosmology, a typical simulation takes 5–10 days of runtime on 3000 cores.

The chemical evolution of the galaxy as described by the model is coupled to the details of this merger history. The model describes the evolution of *populations of stars* associated with each of the dark matter halos. These populations evolve through star formation, interactions with stellar winds and stellar decays. Each halo is modelled as having some initial gas that is accreted from the interstellar medium (ISM), this gas collapses into stars and eventually these stars decay and return energy and metals [1] back to the halo and into the larger environment. This process is iterated and new generations of stars form from the now metal enriched gas in the halo.

We will begin by concerning ourselves with building an understanding of the influence of an important subset of the calibration parameters controlling the chemical evolution of our candidate milky way by varying them under a single fixed merger history. Later we will turn to examining the influence of a small set of candidate dark-matter merger histories.

6.2.1 Parameters

Table 6.1: ChemTreeN calibration parameters.

Parameter	Fiducial Value	Range	Description	Explored
z_{r}	10	5 - 19	Epoch of re-ionization	Yes
f_{bary}	0.05	0 - 0.2	Baryonic mass fraction	Yes
f_{esc}	50	0 - 110	Escape factor of metals	Yes
ϵ_{*}	1×10^{-10}	0.2 - 1.8	Star formation efficiency (10^{-10} yr^{-1})	Yes
$m_{\mathrm{Fe}}^{\mathrm{II}}$	0.07	0.04 - 0.2	SN II iron yield ($M_{\odot}$)	Yes
f_{Ia}	0.015	$\cdots$	SN Ia probability	No
ϵ_{SN}	0.0015	$\cdots$	SNe energy coupling	No
$m_{\mathrm{Fe}}^{\mathrm{Ia}}$	0.5	$\cdots$	SN Ia iron yield ($M_{\odot}$)	No

[1] Here metals refers to baryonic matter other than hydrogen

The calibration parameters identified as being potentially interesting by the domain scientists are listed in Table: 6.1. They represent a series of potentially subjective parameterizations of very complex physical processes. Learning about the values of these calibration parameters which are compatible with observational data would be very interesting, as it would help constrain the scale and nature of many of the modelled physical processes. However it should not be thought of as a measurement of a fundamental physical constant, such as measuring the charge of an electron or the mass of the Higgs boson, since these calibration parameters may not have a directly corresponding physical constant.

Some brief description of the explored parameters is given below.

z_r: The red-shift z at which reionization begins. Recall that for a FRW cosmology with scale parameter $a(t)$ the cosmological redshift between two times is

$$1 + z = \frac{a(t_{\text{now}})}{a(t_{\text{then}})}$$

After recombination the early universe was initially populated with hot hydrogen. Reionization is the cosmological process where radiation from (and perhaps their explosive decay) the initial population of very bright pure hydrogen stars ionizes and induces a large velocity dispersion in the interstellar medium. This is an important process as it imposes a lower bound on the mass of a dark-matter halo which is sufficient to collect enough baryonic matter to form a substantial population of stars. Halos which are not heavy enough by this 'time' will remain effectively barren.

f_{bary}: The baryonic mass fraction, the proportion of baryonic matter to dark matter assigned to each halo. This is substantially larger than say the WMAP3 cosmological baryon density as it is a local quantity.

$f_{\rm esc}$: The escape factor of metals. This sets the level of metallicitiy enrichment of galactic winds relative to the interstellar medium metallicity. This effectively sets up a flux of metallicity out from the simulation, without this the simulated stellar populations would be far too metal rich.

$\epsilon_{\star}$: Sets the rate of star formation.

$m_{\rm Fe}^{\rm II}$: The amount of iron in solar masses produced by core collapse (type II) Super Novae (SN).

To create the training set a $d = 200$ point design was created using a Latin Squares method (see § 4.3). This number of points gives an acceptable balance between covering the available space and run time. The input parameters are allowed to vary within the ranges specified in Table: 6.1.

6.2.2 Output

The model output for each of the $d = 200$ locations in the design is plotted In Fig: 6.1. The output is in the form of the cumulative distribution of the number of satellite galaxies below a certain absolute visual magnitude (left panel) and the satellite metallicity ratio as a function of absolute visual magnitude (right panel). Both of these outputs are functional, in the sense of being naturally given by some curve, over M_v the absolute visual magnitude.

From a superficial inspection of these figures it's clear that the ranges of the calibration parameters covered by our design lead to substantial variations in the model performance. Furthermore one should note that the fiducial curves (black) have roughly central positions in both figures. These are both positive signs that we have constructed a design which samples the model output in a wide, and roughly symmetric, variety of conditions around our case of interest. Had we observed a very reduced variation this might suggest that the design is not wide enough, or that the model doesn't actually respond very strongly to the parameters in our design. If our field data (in this case the fiducial curves) was not roughly spanned by all the runs in the training set this would suggest that our model may have some systematic deviations from true physical process which would need to explicitly treated as part of the analysis and calibration process (see § 9.3). We will use these fiducial curves to stand in for field data to give a focal point for building an understanding of the model response.

The absolute visual magnitude is a standardized measure of brightness for a astronomical object, with the standard being the brightness that would be observed if the object was at a distance of 10 parsec. The scale is inverse and logarithmic, a difference of five magnitudes corresponds to a factor of 100 in brightness. More negative values are brighter and more positive values are dimmer.

The Luminosity Functions (LF) (left panel Fig: 6.1) show the cumulative number of satellite galaxies at a given luminosity. Satellite galaxies are gravitationally bound clumps of stars which are themselves bound to our milky way candidate. Typically we see that there are rather few very bright satellites, recall that absolute magnitude is a inverse logarithmic scale so points towards the left end of the spectrum are brighter. The cumulative distributions grow fairly slowly and relatively uniformly with decreasing magnitude. The black curve shows the result of running the simulation at the given set of fiducial values. To reduce this data to a more manageable form we slice the full spectra into a set of five bins spaced at increasing intervals in magnitude, these are indicated by the dashed vertical lines. The choice of bins was made so as to be most sensitive to the shape of the luminosity function at the bright end of the spectrum as the bright satellites are typically the most influential.

The metallicity ratio (L-Z) (right panel Fig: 6.1) is a logarithmic scaled measure of how average amount of iron present in a satellite galaxy relative to the amount of hydrogen. Since iron is produced by type two super novae this is a measure of the maturity of the stellar populations in the satellites. The data plotted in the figure is a result of applying a linear fit to the metallicity and luminosity for each simulation. The fit coefficients were used to represent this data in the further analysis. The results of the fiducial run are plotted in black. We can see that the brighter satellites (left end of the plot) are typically much more metal rich than the dimmer ones.

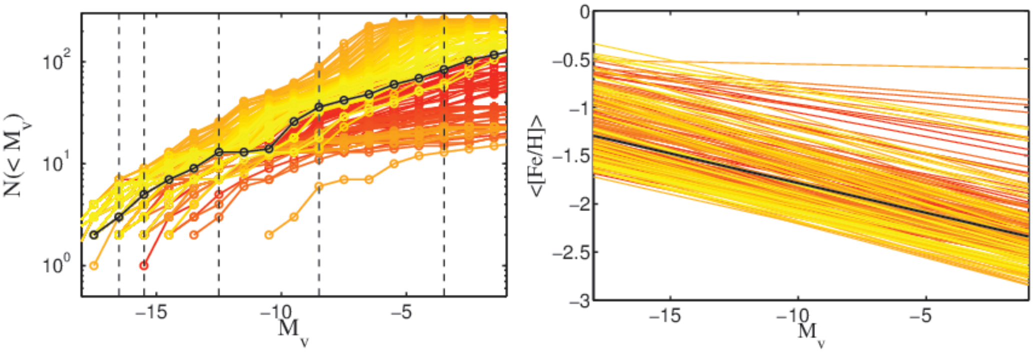


FIGURE 6.1: Satellite galaxy luminosity functions (left panel) and satellite galaxy luminosity-metallicity relations (right panel). The result of a linear fit to each luminosity-metallicity relation is shown. The models were obtained after coupling ChemTreeN with the N-body simulation MW1. The vertical dashed lines on the left panel indicate the five values of M_{v} chosen to sample the LFs. The black solid line on both panels indicate the model considered to be the galaxy's "true" observational quantities, obtained after running ChemTreeN with the input parameter vector u_{obs}.

6.3 Emulator Specification

The first step in constructing a model emulator is to obtain a finite set of model outputs at the design points. These outputs are obtained by running ChemtreeN using different sets of input parameters drawn from an experimental design $\mathcal{D} = \{(x_1, u_1), \ldots, (x_d, u_d)\}$. From here on out we will only deal with calibration parameters u, we will take u_i as a three-component vector, $u_i = (z_{\mathrm{r}}^i,\ f_{\mathrm{esc}}^i,\ f_{\mathrm{bary}}^i)$. These variable were identified as the three most interesting parameters for a primary analysis. It is trivial to increase the dimensionality of u_i, however interpretation and visualization of the final results become progressively more complicated with increasing dimensionality.

Once the models have been run, the next step is to choose the set of outputs, $Y = \{y_1, \ldots, y_k\}$ of interest. Initially let us construct individual emulators for each of these outputs. Motivated by the discussion in § 6.2, we chose to emulate five values of the satellite galaxy luminosity function, each one at a different absolutely magnitude, in addition to the slope and the intercept of the satellite galaxy luminosity-metallicity (*L*-*Z*) relation. This gives us a total of $k = 7$ outputs to be extracted from the model runs. Each of these outputs strongly constrains different model parameters.

The model parameters u and output $\mathcal{Y}$ are scaled and centered prior to emulator analysis, Centering the model output $\mathcal{Y}$ is usually a good idea as it removes any trends which are common to all of the design points, allowing the GP emulator to deal with more interesting residual variation across $\mathcal{D}$. Scaling should be approached with a little more caution as it puts the residual variation in *all outputs* on the same footing. This is quite reasonable if one has a strong prior belief that all the outptus are equally important. However if for some reason the variance in the model output varies across the k dimensional output space this may not be a good idea. For instance if the model output was a spectrum built from a finite number of observations of an underlying power law (e.g. the p_T distribution of jets in a high energy P–P collision) then the bins at higher values are naturally going to be more uncertain as relatively fewer observations will have been made. Scaling the bins of such histogram to all have the same variance would be a mistake, this would override the variation which naturally arose from the sampling scheme.

After training the seven model emulators by computing the maximum likelihood estimates for the GP covariance parameters, we compare the model (via the emulators) to the observable data by calculating surfaces of implausibility $I(x, u)$

for each observable (see § 6.4). The values of these three-dimensional surfaces provide an indication of which parts of the input space u are more likely to reproduce the desired observational data set Y_f.

The observable data should be obtained from the luminosity function and *L*-*Z* relation of the Milky Way satellite galaxies. However, to test the constraining power of this approach, a particular run of the ChemTreeN model will be used as a mock observable data set. This type of controlled experiment can be very helpful in model performance assessment, since we know exactly what values of the input parameters were used to obtain the artificial "field data." The black solid lines in Fig: 6.1 show the luminosity function and *L*-*Z* relation of the model used as the mock observations. The values of the input parameters used are $u_{\rm obs} = (z_{\rm r},\ f_{\rm esc},\ f_{\rm bary}) = (10,\ 50,\ 0.05)$. It is important to note that this input parameter vector is not included among the design points $\mathscr{D}$.

6.4 Comparison to Fiducial Data

To make a simple comparison of the simulator (via the GP emulator) to experimental data, it is convenient to introduce the notion of *implausibility* [31]. Let's define an implausibility measure $I(x,u)$ as follows. Consider a model with a single output for which we have generated an emulator with posterior mean $\bar{\mu}(x,u)$ and variance $\bar{K}((x,u),(x,u))$. The implausibility of the emulated model output at a point (x,u) in the parameter space is given by

$$I^2(x,u) = \frac{(\bar{\mu}(x,u) - \mathrm{E}[Y_f])^2}{\bar{K}((x,u),(x,u)) + \mathrm{V}[\mathrm{Y_f}]}, \tag{6.1}$$

where Y_f represents the distribution of field data that we seek to compare our model against. Here we have only accounted for the variation from the emulator itself and the field data. In the following work we carry out comparisons of the model output with idealized field data generated from the model itself. We will compare the model output at different locations in the parameter space against certain selected default values, as such we are free to neglect model bias or discrepancy terms.

The output of ChemTreeN is multivariate – the code produces predictions for many observables, such as the distributions of stellar populations in stellar halos of Milky Way-like galaxies or its satellite galaxy luminosity function and metallicity-luminosity relation. It is possible to separately compare each observable with a model emulator generated from the corresponding model output. Considerably more powerful conclusions can be drawn by examining the joint properties of the observables and model outputs, as discussed in chapter 5. Consider a k-dimensional vector of model outputs $y(x, u) = \{y_1, \ldots, y_k\}$ with a corresponding vector of field data Y_f. We extend our training set to be the $d \times k$ matrix $\mathcal{Y} = \{y(x_1, u_1), \ldots, y(x_n, u_n)\}$.

We apply a principal component decomposition to our training data set $\mathcal{Y}$ to obtain a set of approximately independent and numerically orthogonal basis vectors spanning the k dimensional output space, see § 5.1 discarding terms in the eigen-decomposition which contribute less than 5% of the total variation. We construct individual independent emulators from the training values projected onto each basis. When we wish to evaluate the model output at a new location we invert this transformation to obtain predictive distributions for each of the t model outputs at a given location in the parameter space.

The implausibility (6.1) can be naturally extended to the multivariate case. From the emulator we obtain a k-dimensional vector of predictions for the model output with means $\bar{\mu}(x, u)$. The emulated $k \times k$ dimensional covariance matrix $\bar{\mathbf{K}}(x, u)$ between the model outputs at the point x, u in the design space can also be constructed from the PCA decomposition. With these two quantities, we define the joint implausibility $J(x,u)$ for observables Y_f with measurement variance $\mathrm{V}[\mathrm{Y_f}]$ and mean values $\mathrm{E}[Y_f]$.

$$J^2(x,u) = (\mathrm{E}[Y_f] - \bar{\mu}(x,u))^{\intercal} \left(\bar{\mathbf{K}}(x,u) + I \cdot \mathrm{V}[\mathrm{Y_f}]\right)^{-1} (\mathrm{E}[Y_f] - \bar{\mu}(x,u)) \,, \tag{6.2}$$

this construction provides a covariance weighted combination of the multiple observables which gives a reasonable indication of which regions in x, u are predicted by the emulator to be close to the observed values Y_f. This implausibility score $J(x,u)$ is a normally distributed variable with zero mean and unit standard deviation, confidence intervals for values of $J(x,u)$ can then be established in the usual way. In this section we consider approximate 95% (2σ) confidence intervals as representative of the true values of x, u. Locations in the parameter space with $J(x,u) < 2$ are viewed as being regions which are very likely to give model outputs closely reproducing the observational data, given the experimental and interpolation uncertainties.

6.5 Parameter space exploration

6.5.1 Independent Emulators

Let's first explore the extent of the parameter space that we can constrain using our fiducial data with a set of $k = 7$ independent scalar GP emulators. Each of these GP emulators was trained on only a single component of the multivariate output.

Fig: 6.2 shows three different sections of each of the implausibility surfaces obtained from the five *independent* model emulators constructed for the LF's outputs. The 3-dimensional implausibility surfaces are sliced with three orthogonal planes as defined by the components of u_{obs}. The top row panels show the $f_{\text{esc}} = 50$ section of the $I(u)$ surfaces. The black dashed lines indicate the values of the remaining two components of u_{obs}. Given an input parameter vector u_t, the larger the value of the $I(u_t)$ the less likely it is that a good fit to the observed (fiducial) data could be obtained, given the uncertainty arising from the emulator and with the fiducial data itself.

From the left-most panel (i.e. $M_{\rm v} = -16.5$) it becomes clear that the parameter controlling the amount of available gas to form stars, $f_{\rm bary}$, is strongly constrained by the number of satellite galaxies at the bright end of the satellite galaxy luminosity function. Furthermore, within the range of values considered here, the number of satellites at this $M_{\rm v}$ is independent of the redshift of the epoch re-ionization, $z_{\rm r}$. The most plausible parameter values are near the true value of $f_{\rm bary} = 0.05$.

As we move towards the faint end of the luminosity function the model parameter $z_{\rm r}$ becomes progressively more constrained and the total number of satellite galaxies becomes less dependent on $f_{\rm bary}$. For $M_{\rm v} = -3.5$ (top right-most panel). The corresponding model emulator strongly constrains the input parameter space around values of $z_{\rm r} \approx 10$, but it gives equally good fits for nearly all possible values of $f_{\rm bary}$. The second row of panels show sections of the $I(\boldsymbol{u})$ surfaces at $f_{\rm bary} = 0.05$. The satellite galaxy luminosity function appears to be completely independent of the value adopted for the escape factor of metals, $f_{\rm esc}$. At the bright end of the luminosity function, any combination of $z_{\rm r}$ and $f_{\rm esc}$ would yield an equally good fit to the mock observable data. However at the faint end values of $z_{\rm r} \approx 10$ are required to fit the mock data. A similar result can be obtained for the third row of panels showing the remaining sections, i.e., $z_{\rm r} = 10$. Again, a good fit to the "observable" data can be obtained with values of $f_{\rm bary} \approx 0.05$ for any possible value of $f_{\rm esc}$.

It is possible to put constraints on the parameter $f_{\rm esc}$ by exploring cross-sections of the implausibility surfaces constructed from the satellite galaxy luminosity-metallicity relation's slope and intercept model emulators. The middle and bottom panels of Fig: 6.3 show the sections defined by $f_{\rm bary} = 0.05$ and $z_{\rm r} = 10$, respectively. Comparison with Fig: 6.2 reveals implausibility surfaces with a more complex geography. Although both emulators present regions of low implausibility for a wide range of $f_{\rm esc}$ values, these regions are strongly correlated with $z_{\rm r}$ and $f_{\rm bary}$. These two parameters are also strongly constrained by the slope and intercept of the *L*-*Z* relation, as shown in the top row panels.

6.5.2 Joint Emulators

Individually, none of the previously explored implausibility surfaces constrain the full parameter space. This is not the case with the joint implausibility measure $J(u)$, which combines the information obtained from the seven model emulators into one quantity. Following the PC methods in § 5.1 we construct a multivariate emulator using all $k = 7$ resulting principal components.

Figure 6.4 shows different iso-implausibility surfaces of the resulting $J(u)$. Notice that as the value of $J(u)$ decreases the volume enclosed by each iso-surface becomes smaller, converging towards the values associated with $u_{\rm obs}$, as shown by the red solid lines. This can be more clearly appreciated in Fig: 6.5. Each row of panels shows different sections $J(u)$ as we traverse one of the three possible dimensions in u.

The black solid line on the color bars show the 2σ cutoff applied to the joint implausibility. A value of $J(u) > 2$ indicates that it is very implausible to obtain a good fit to the observed data with the corresponding values of the model parameters. Thus, regions of the parameter space lying above this threshold can be disregarded. We find that $J(u)$ strongly constrains the full parameter space under study. Furthermore, the values of the components of $u_{\rm obs}$ are located in the most plausible regions of the space, as indicated by the star symbols in the corresponding panels.

6.6 The rôle of merger histories

In the previous section we showed that it was possible to recover the set of input parameter chosen to create a mock Milky Way-like observational data set using a suite of model emulators. In this "controlled" experiment, both training and mock observable data were obtained by coupling ChemTreeN to a merger tree obtained from a single simulation. We have implicitly assumed that the exact merger history of our Milky Way-like galaxy is a known quantity. In reality, this merger history is poorly known, and should be regarded as an extra input parameter of the model. It is thus important to study how different merger histories can compromise our ability to meaningfully constrain the input parameter space.

For this purpose, we perform the following set of controlled experiments. Using the merger trees extracted from the four available N-body simulations we generate four different training sets (each training set containing $n = 200$ design points) and construct, for each set, the suite of model emulators discussed previously. Hereafter, we will refer to these emulators' as "MWi-emulators", with $i =$ 1, 2, 3 and 4. The input parameter vector $\boldsymbol{u}_{\rm obs} = (z_{\rm r},\ f_{\rm esc},\ f_{\rm bary}) = (10, 50, 0.05)$ is used to obtain a mock observational data set from each merger tree. We will refer to these mock observables as "MWi-observables". The cumulative mass of these merger trees as a function of redshift, is shown in Fig: 6.6. While there is clearly some variation between the candidates, the extent to which this variation matters is not clear. We then ask the following question: *is it possible to recover the input parameter vector, $\boldsymbol{u}_{\rm obs}$, if we use training data obtained from a merger tree different than that used to obtain the mock observables?*

In Fig: 6.7 we show the outcome of this experiment. Each block of four panels shows joint implausibility surface's sections obtained after comparing a given MWi-observables with the four MWi-emulators. The merger tree used to generate the MWi-observables in each block is indicated with the green label, MWi. For example, on the top left corner we show the result of such comparison using MW1-observables. As previously shown in Fig: 6.5, when the model emulators are trained on the same merger tree that was used to generate the mock observables, we can successfully constrain the input parameter space and recover the components of $u_{\rm obs}$. However, when model emulators constructed on different merger trees are considered, the most plausible regions are located around values of $f_{\rm bary}$ much larger than those used to obtain the mock observables. This is not surprising since, as shown in Fig: 6.6, MW1 is the Milky Way-like halo that contains the largest number of satellites at all $M_{\rm v}$. To achieve a good fit to MW1-observables in the remaining simulations, it a larger amount of gas to form stars is required. Note as well that the joint implausibility surface obtained with the MW3-emulators has no values below the chosen threshold. Thus MW1-observables cannot be reproduced using the merger history extracted from halo MW3. Another interesting example is shown on the lower right panels of Fig: 6.7. Here MW4-observables are considered. Very good fits to these observables can be obtained for either larger (MW3-emulators) or smaller (MW2-emulators) values of $f_{\rm bary}$ than that used to generate the mock observables. A similar situation is observed for the input parameter $z_{\rm r}$. Note that we have only considered the $f_{\rm esc} = 50$ section of each joint implausibility surface.

The previous analysis clearly shows how a particular merger history can influence the model parameter selection: similarly good fits to a given set of observables can be obtained with different model parameter values simply by modifying the host's merger history. In our experiments these values may differ from those used to generate the mock observables. When comparing with real observational data, a given set of best fitting parameter's values may be significantly off from the values that could best parametrize the desired underlying physical processes. This in turn may have important implications on other observable quantities that we would like to study and which have not been used for model parameter selection.

6.7 Conclusions

We successfully used the joint implausibility to constrain the possible parameter space to a small region around the point u_{obs} selected to generate the fiducial data. By exploring the simple implausibility surfaces generated for each observation we gain some useful insights into how sensitive the respective observables are to the calibration parameters.

By expanding the scope of the analysis to considering a small set of superficially similar dark matter merger histories we were able to show that the calibration parameters needed to reproduced the various fiducial runs are quite different. This result suggests that details of the merger histories have a more important impact on the chemical model then previously appreciated.

This analysis has provided not only a good first step towards understanding how ChemTreeN responds to its most significant calibration parameters, but also has provided the first evidence for a relatively novel scientific result namely that superficially similar merger histories may not be entirely generic.

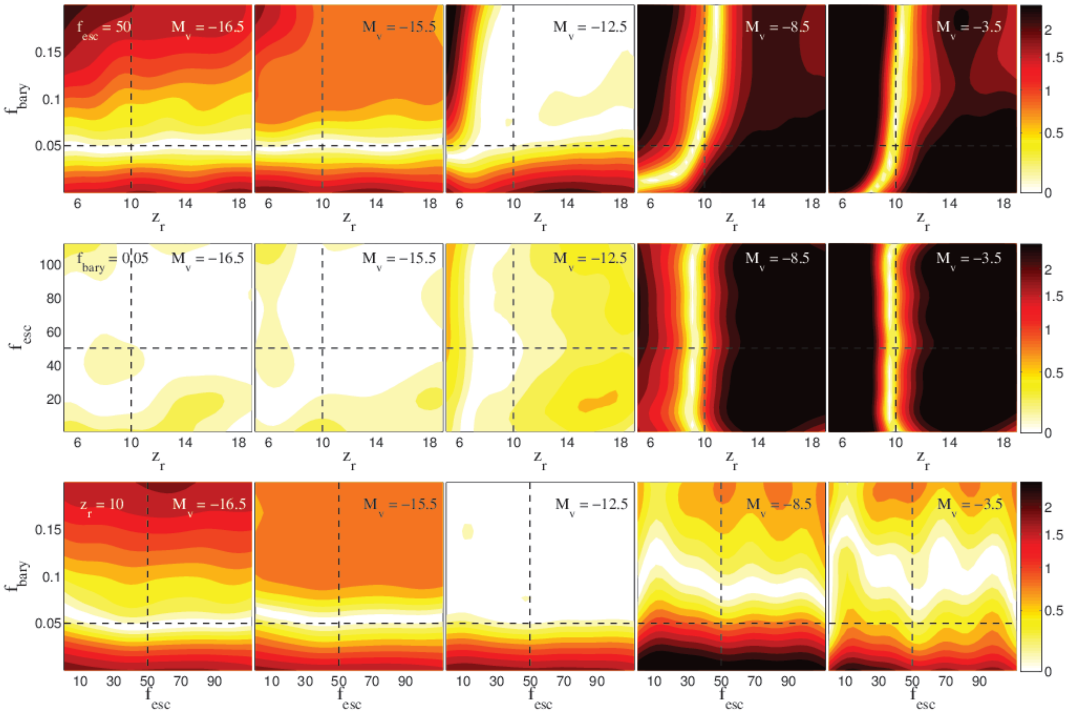


FIGURE 6.2: Sections through the Implausibility surfaces, $I(u)$, obtained from the five model emulators constructed for the LF's outputs. The output being emulated is indicated on the top right corner of each panel. Columns correspond to different observables. The $3d$ implausibility surfaces are sliced with three orthogonal planes as defined by the components of u_{obs}. The top, middle and bottom row panels show the $f_{\mathrm{esc}} = 50$, $f_{\mathrm{bary}} = 0.05$ and $z_{\mathrm{r}} = 10$ sections of the $I(u)$ surfaces, respectively. The black dashed lines indicate the values of the remaining components of u_{obs}. Given an input parameter vector u_t, the larger the value of the $I(u_t)$, the less likely a good fit to the observable data can be obtained. It is possible to strongly constrain the parameters f_{bary} and z_{r}, but not f_{esc}.

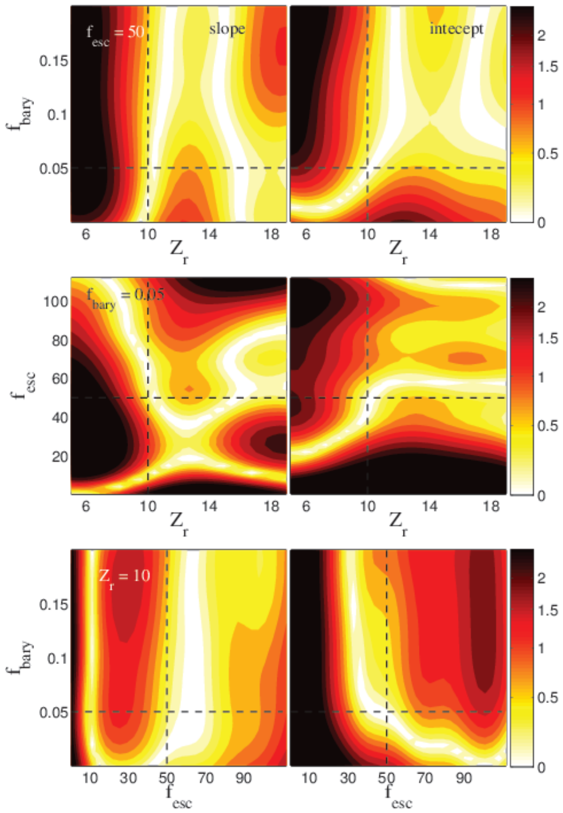


FIGURE 6.3: As in Fig: 6.2 for the two model emulators constructed for the L-Z relation. The left and right panels show sections of the implausibility surfaces associated with the slope and the intercept, respectively. These model emulators provide strong constraints on the model parameter $f_{\rm esc}$.

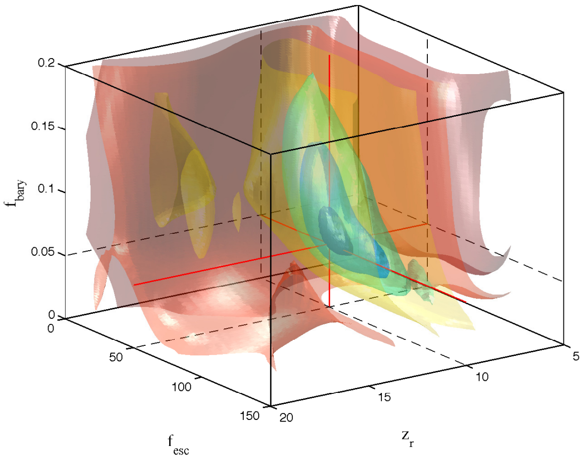


FIGURE 6.4: Iso-implausibility surfaces extracted from the joint implausibility measure $J(u)$. Redder colors indicate larger values of $J(u)$. The region of lowest implausibility (and thus highest plausibility) is shown by the opaque blue volume at the intersection of the red lines.

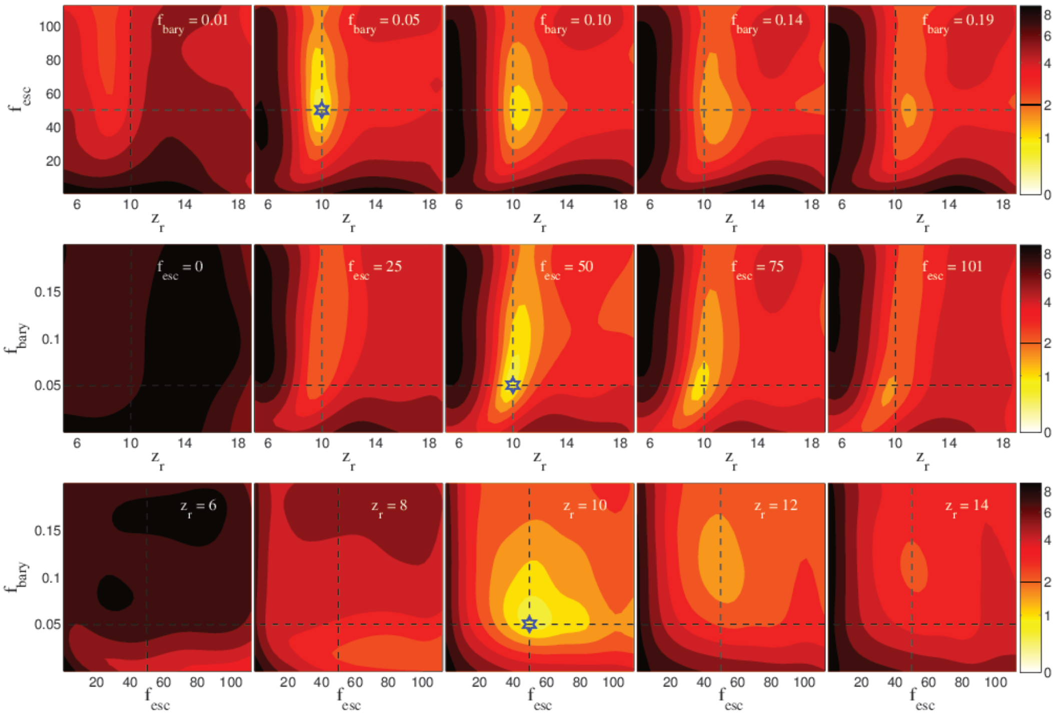


FIGURE 6.5: Sections of the Joint implausibility surface, $J(u)$, obtained by combining information provided by the seven model emulators shown in Figures 6.2 and 6.3. The top, middle and bottom row panels show different sections of constant f_{esc}, f_{bary} and z_{r}. On each row, the black dashed lines indicate the values of two of the components of u_{obs}. If the three components are simultaneously present in a section, the location of u_{obs} is indicated with a blue star. The horizontal black solid line on the color bars indicate the imposed two-sigma threshold.

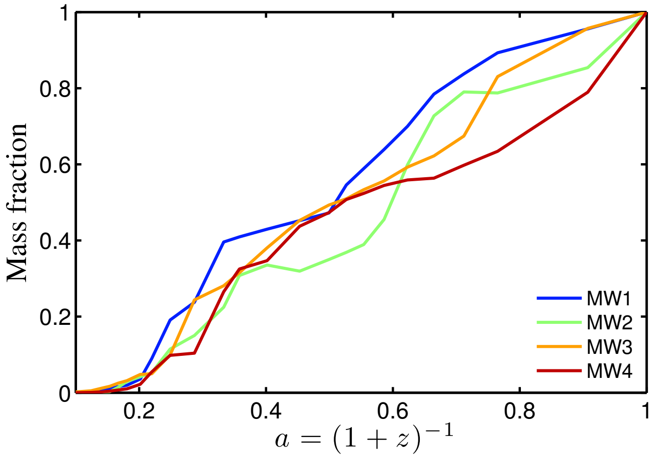


FIGURE 6.6: Galaxy formation history as shown by the virial mass of the most massive progenitor of the four candidate Milky Way-like dark matter halos as a function of the expansion factor. In all cases, the mass is normalized to the $z = 0$ mass of the galaxy.

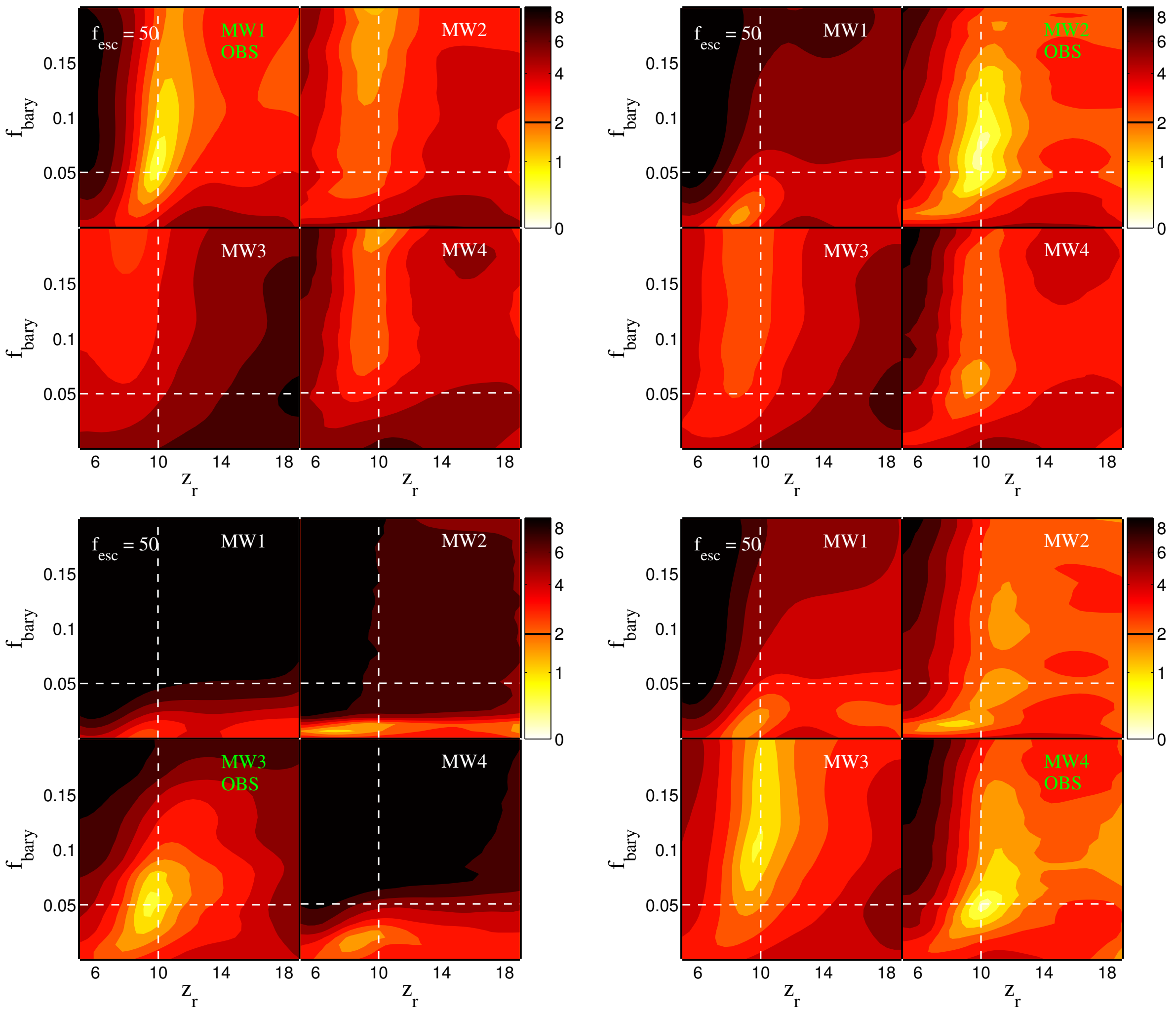


FIGURE 6.7: Sections of Joint Implausibility surfaces at constant $f_{\rm esc}$, obtained after comparing different models and mock observables. Each block of panels shows the results of comparing a given MWi-observables to the four sets of MWk-emulators (see text), where $k,\ i = 1,\ 2,\ 3$ and 4. MWi-observables are obtained by running ChemTreeN on the merger tree extracted from simulation MWi, using the input parameter vector $u_{\rm obs}$. The labels on the top left corner of each panel indicates the MWk-emulators being considered. In green we indicate the MWi-observables associated with each block. The white dashed lines indicate the values of two of the components. Note that, in many cases, similarly good fits to a given set of observables can be obtained with different parameter's values simply by modifying the host's merger history.

7

Uncertainty Analysis

In this chapter I will discuss a simple practical application of the GP emulator results developed in § 3. Consider a computer model $Y_m(x, u)$ with n observation parameters x and p calibration parameters u. As discussed in the introduction in § 1.6 the calibration parameters are typically unknowns that we wish to learn about but which we cannot explicitly vary when making experimental observations. They might be a fundamental physical constant such as a particle's mass or a certain transport coefficient or perhaps a parameter in a model of how a detector responds to a certain input. Calibration parameters might be further classified into those parameters which have some physically interesting meaning or they may be purely a residue of the computer modelling process, such as a certain choice of a numerical cut-off or a grid spacing in some finite difference scheme.

Later on in § 9 I will outline the steps needed to use a set of experimental observations along with a set of observations of a computer model to try and learn about the *true* values of these calibration parameters. Where as always true really means the set of values which best agree with our choice of computational and statistical models. Let us suppose that we have the above computer model $Y_m(x, u)$ with some calibration parameters and experimental data is not available. The calibration parameters are assumed to be weakly known. By weakly known I am referring to the fairly usual case, when although a model developer doesn't *a-priori* know the true values of these u they can be pushed into giving some kind of plausible range or bounds for their values.

Given such a set of weak prior knowledge about these calibration parameters a natural question is: "how does the uncertainty on these parameters pass into our uncertainty about the model output?". Suppose that we are interested in the model output at a particular set, or range, of observation parameter values $x_\star$ then given some plausible ranges on the unknown calibration parameters u we would like to know about the distribution of model outputs we should expect at these observation locations. If we can motivate some prior distribution for the unknown parameters u then schematically we're interested in obtaining a conditional distribution for $Y_m(x_\star)$ given the prior $P(u)$. In the literature this kind of process is referred to as uncertainty analysis, [22, 23].

An uncertainty analysis like this certainly does not tell us anything about what the true or *best* values of these calibration parameters should be. However it does offer a substantial amount of insight into the behaviour of the simulator. Understanding the amount of parametric variability in a simulator is a key step in learning where to most carefully focus attention in the collection of additional data.

An example of this kind of situation is the estimation of systematic errors in physics experiments. Here the calibration parameters u represent unknown aspects of a detector response or similar and the computer model could represent the entire data analysis.

After ascertaining nominal values for a set of unknown calibration parameters $\bar{u}$ a typical process for estimating their influence upon some observable of interest is to repeat the analysis with a given parameter set to values representing the extremes of its plausible range and then quantifying the influence of this procedure on the observables of interest. This can be thought of as approximately linearizing the model's response around the nominal parameter values.

$$Y_m(x_\star, u) \simeq Y_m(x_\star, \bar{u}) + (u - \bar{u}) \left. \frac{\partial Y_m(x_\star, u)}{\partial u} \right|_{u=\bar{u}} . \tag{7.1}$$

If the other calibration parameters are also held fixed during this procedure, instead of somehow jointly varying them, then this procedure essentially diagonalizes the covariance structure of the model. This is sometimes referred to as the "one at a time" or OAT process. Depending on the details of model and the number of calibration parameters this varies between a bad and an awful way to approach the problem. As the number of model parameters grows the volume of the parameter space explored by such naïve procedures becomes very small. For further discussion see the definitive text by Saltelli et al [27].

7.1 A Contrived Model

To help illustrate these ideas let's consider a simple model inspired by particle physics. The Breit-Wigner distribution is a probability distribution for observing an unstable particle at a given centre of mass energy E given the particle has a decay width (or inverse lifetime for the state) Γ and mass M

$$f_{BW}(E, M, \Gamma) = \frac{k}{\left(E^2 - M^2\right)^2 + M^2\Gamma^2}, \tag{7.2}$$

$$k = \frac{2\sqrt{2} M \Gamma \gamma}{\pi \sqrt{M^2 + \gamma}}, \quad \gamma = \sqrt{M^2(M^2 + \Gamma^2)}.$$

Here we can think of E as the observation parameter, this represents an energy that an otherwise perfect particle detector is tuned to. The particle mass M and the decay width Γ can serve as our calibration parameters. Now let's suppose that we are in the unfortunate situation of only being able to build a detector which measures the mean energy over some width ΔE centered around whatever energy E we tune it to, i.e.

$$y_{\mathrm{BW}}(E, \Delta E, M, \Gamma) = \frac{\int_{E-\Delta E/2}^{E+\Delta E/2} E' f_{BW}(E', M, \Gamma)\, dE'}{\int_{E-\Delta E/2}^{E+\Delta E/2} f_{BW}(E', M, \Gamma)\, dE'}. \tag{7.3}$$

We are interested in understanding how predictions from y_m at some fixed E and ΔE vary given our uncertainty in the decay width Γ and the particle's mass M. Putting some numbers in let's suppose we're trying to measure the energy of the Z boson, for reference the Particle Data Group (PDG) reported mass and decay width are $m_z = 91.1876 \pm 0.0021$ GeV/c^2, $\Gamma_z = 2.4952 \pm 0.0023$ GeV/c^2 [211]. We will center our detector energy at $E_d = m_z$ and set the energy width to $\Delta E_d = 20$ GeV/c^2, evaluated at these "nominal" values our model gives

$$y_{\mathrm{BW}}(E_d, \Delta E_d, m_z, \Gamma_z) = 91.1102 \tag{7.4}$$

for brevity from now on I will drop the units on the model output. The integrals in (7.3) can be evaluated analytically although the result is a little messy, this provides a neat form for comparison with our statistical methods.

7.2 Uncertainty Analysis

We aim to quantify the uncertainty in our model outputs induced by uncertainty in the inputs, for now lets suppose our uncertainty is confined to the calibration parameters. We can consider the uncertain calibration vector u to be a random vector U, now our model output is promoted to a random variable $\eta = Y_m(x, U)$. Given a probability distribution G for the uncertain calibration vector U we want to learn about the probability distribution for the model output η.

Lets consider the simplest useful things we can learn about η, the mean $\mathrm{E}[\eta]$ and variance $\mathrm{V}[\eta]$. These quantities along with some form of credible interval for η, i.e. a region bounding the mean that we expect an observed value to fall within with a certain probability, should give sufficient information about how our uncertainty in the parameters passes through to the model output.

The simplest approach to uncertainty analysis is to sample the probability distribution G (using typical Monte-Carlo methods [171]) to obtain some set of n_{MC} input configurations $u = \{u_1, u_2, \ldots, u_{n_{\mathrm{MC}}}\}$, where the distribution of u approximates G as $n_{\mathrm{MC}} \to \infty$. The model can then be run at each of these points, giving a set of outputs $y = \{Y_m(x, u_1), Y_m(x, u_2), \ldots, Y_m(x, u_{n_{\mathrm{MC}}})\}$. Sample estimates of y, such as the mean and variance, are naturally sample estimates of the same quantities of η. This conceptually very simple Monte-Carlo method is certainly superior to the naïve range sampling method. However it will most likely require a fairly large number of model evaluations to obtain a posterior which is a good approximation to η. For non trivial models this may require a substantial investment of effort and resources.

Let's examine what we can learn about our toy model using this simple method. To keep things interesting let's ignore the quoted (and small) uncertainties from the PDG data and instead we'll take 10% uncertainties on the measured values, promoting M and Γ to independent random variables we have

$$M \sim \mathrm{N}(\mathrm{m_z}, (0.1\mathrm{m_z})^2), \quad \Gamma \sim \mathrm{N}(\Gamma_{\mathrm{z}}, (0.1\Gamma_{\mathrm{z}})^2),$$

so that in this case the distribution on our inputs is

$$G = \begin{pmatrix} M \\ N \end{pmatrix} \sim \mathrm{MVN}\left\{ \begin{pmatrix} m_z \\ \Gamma_z \end{pmatrix}, \begin{pmatrix} (0.1m_z)^2 & 0 \\ 0 & (0.1\Gamma_z)^2 \end{pmatrix} \right\}.$$

A histogram of the set of Monte-Carlo outputs y, generated after sampling a set of $n_{\mathrm{mc}} = 4000$ input configurations from G is shown in Fig: 7.1. The uncertainty in

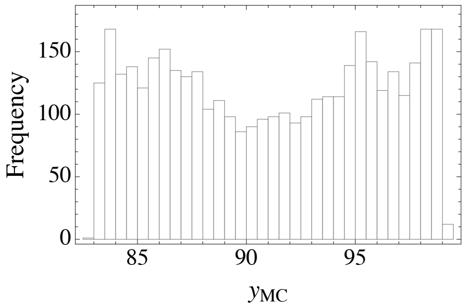


FIGURE 7.1: A histogram of model output (defined in (7.3)) generated from a set of 4000 sample input configurations.

our inputs has resulted in a large spread of measured values, note that the bulk of this distribution is not centered around our expected mean value (7.4). Lets focus our attentions on the posterior mean value of the MC distribution,

$$\bar{y}_{\mathrm{MC}} = E[y] = \frac{1}{n_{\mathrm{MC}}} \sum_{i=1}^{n_{\mathrm{MC}}} Y_m(x, u_i), \tag{7.5}$$

this is the average energy that our detector would measure as predicted by our model given the uncertain inputs. The choice of the posterior mean is purely for simplicity, practically one would certainly also be interested in the width of the posterior distribution as quantified by estimates of the variance or estimates of quantiles. For completeness the sample variance is defined as

$$s^2_{y,\mathrm{MC}} = \frac{1}{n_{\mathrm{MC}} - 1} \sum_{i=1}^{n_{\mathrm{mc}}} \left(Y_m(x, u_i) - \bar{y}_{\mathrm{MC}}\right)^2, \tag{7.6}$$

the expected variance of $s^2_{y,\mathrm{MC}}$ can be obtained but pursuing this is an unnecessary complication for the purposes of this analysis.

From the data used to make Fig: 7.1 we obtain $\bar{y}_{\mathrm{MC}} = 91.0952$ the Monte-Carlo sample variance is $s^2_{y,\mathrm{MC}} = 24.0458$. The standard error associated with our Monte-Carlo sample mean is $\sqrt{24.0458/4000}$ and so a 95% interval for $\bar{y}_{\mathrm{MC}}$ is $[90.9432, 91.2472]$. Since our model can be expressed algebraically we can directly

Table 7.1: Estimates for the mean of the model output, given the uncertainty distribution G computed by various methods. The naïve bounds are computed by evaluating y_m with M, Γ set at two standard deviations below and above the nominal values.

	$\bar{y}$	conf lower	conf upper
exact	91.0635		
naïve	91.1102	86.8729	94.7689
$n_{\mathrm{MC}} = 4000$	91.0952	90.9432	91.2472
$n_{\mathrm{MC}} = 40000$	91.0662	91.0176	91.1147

compute the exact mean of the model output given the uncertainties,

$$\begin{aligned} \bar{y}_{\mathrm{exact}} &= \int_{-\infty}^{\infty} y_m(E, \Delta E, M, \Gamma) f(M, m_z, 0.1 m_z) f(\Gamma, \Gamma_z, 0.1\Gamma_z)\, dM\, d\Gamma \\ &= 91.0635, \end{aligned} \tag{7.7}$$

where $f(x, \mu_x, \sigma_x) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_x)^2}{2\sigma_x^2}\right)$ represents the normal density with mean μ_x and standard deviation σ_x. With 4000 model evaluations we appear to have obtained a moderately accurate Monte-Carlo estimate of the posterior mean, given the distribution G on our uncertain parameters. These results are summarized in Table: 7.1.

7.3 Uncertainty Analysis with an Emulator

In the above section we used some large-ish number of model evaluations to get an apparently reasonable estimate of the posterior mean. For many computer models this kind of brute-force Monte-Carlo sampling may represent an unreachable amount of computing. This high barrier to entry may discourage many computer-model builders from even beginning to think seriously about estimating the influence of parameter uncertainty on model predictions.

We can use the Gaussian-Process regression methods developed in chapter 3, particularly the results from § 3.3, to build a statistical surrogate model of the slow computer model. This surrogate will be computationally cheap to evaluate and typically will require far fewer samples of the underlying computer model to generate than the number of model evaluations needed by the direct MC method.

To proceed we should generate a design of m points (where $m \ll n_{\mathrm{MC}}$) which span the parameter space $\mathcal{D} = \{u_1, u_2, \ldots, u_m\}$ for details on the design see § 4.3. We then construct the training set $\mathcal{Y} = \{Y_m(x, u_1), \ldots, Y_m(x, u_m)\}$ by evaluating the computer model at each of these points. Making reasonable choices for the parametric forms of the prior mean and covariance function for the GP and estimating their hyper-parameters, as discussed in § 3.8, we fully specify our surrogate model.

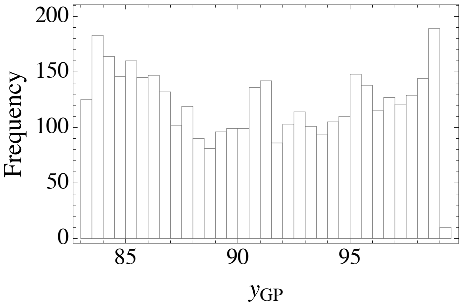


FIGURE 7.2: A histogram of the mean of the GP emulated model output $\bar{\mu}(u)$ this output was generated from a set of 4000 sample input conditions. The GP emulator was trained from a set of $m = 128$ observations of the model.

With the emulator constructed we can proceed to carry out the same simple Monte-Carlo procedure as given in the previous section, generating a set of n_{MC} sample configurations drawn u from G and instead of directly evaluating the model at each of these locations we evaluate the emulator mean $\bar{\mu}(u)$ (3.17). A histogram of these samples is shown in Fig: 7.2, these are generated using a GP surrogate with a training set with $m = 128$. Estimates for the posterior mean $\bar{y}$ generated using this method are shown in Table: 7.2 for $n_{\mathrm{MC}} = 4000$ as a function of the number of training points m. The agreement with the exact result is rather good especially given that these results require a substantially reduced set of full model evaluations.

Table 7.2: Estimates for the mean of the model output, given the uncertainty distribution G computed using the Monte-Carlo method on top of a GP emulator with $n_{\mathrm{MC}} = 4000$.

m	$\bar{y}$	conf lower	conf upper
exact	91.0635		
60	91.1072	90.9550	91.2594
128	91.0444	90.8910	91.1978
256	91.0196	90.8667	91.1724

7.4 Direct Calculation

Since the GP posterior defining our surrogate is such a simple function we can directly compute the mean of the model output given the uncertainty on the parameters G,

$$\bar{y}_{\mathrm{GP}} = \int \bar{\mu}(u) dG(u), \tag{7.8}$$

we can also compute the expected point-wise GP variance

$$s^2_{y,\mathrm{GP}} = \int \bar{K}(u,u) dG(u), \tag{7.9}$$

where $\bar{K}$ is given by (3.18), and $\bar{\mu}$ is given by (3.17). For simple distributions G these integrals can typically be computer analytically, for full results see [23]. In the next chapter I will discuss the process of sensitivity analysis which is closely related to that of UA and indeed we will directly compute analogues of (7.8) and (7.9).

8

Sensitivity Analysis

Given the complexity of typical simulators, model builders are often interested in understanding how the simulator outputs interms of individual inputs. As discussed in the previous chapter building an understanding of this can be a tricky proposition if the model is computationally expensive. Naïve approaches to this problem might be to generate a set of Monte-Carlo samples of the simulator output across the parameter space and use these to attempt to reconstruct the model output or response surface about points of interest. Sophisticated Monte-Carlo sampling procedures have been developed for this purpose which attempt to minimize the total number of model evaluations, (see [27]), however they are often still very expensive.

Using a model emulator we can do somewhat better than this. Following along from the ideas developed in the previous section we can construct a surrogate model and then consider the simulator inputs as random variables. The GP posterior mean and variance which represent our simulator are sufficiently straightforward that one can directly evaluate many useful quantities [25].

To gain an understanding of the shape of the model's output, or response surface, as a function of the various input parameters we can decompose the output into a series of functions of increasingly complex combinations of the input parameters. These functions, known as the main effects and interactions, provide a measure of how each individual input parameter and each combination of parameters contribute to the response surface. We construct these functions by directly integrating out all but the subset of parameters that we are interested in, without a surrogate model this would be a very challenging procedure. This decomposition can be a very effective way of understanding which parameters are more influential.

These techniques were recently used by myself and collaborators in a follow up analysis of ChemtreeN [3] (see chapter 6 for a detailed discussion and analysis) to screen out the most important parameters from a larger set and to compare how the sensitivity of the model to these parameters varied as a function of the Dark Matter merger history.

8.1 Inference for main effects and interactions

Suppose we have a computer model $Y_m(x)$ which produces scalar output and some number d of input parameters. Given a suitable design $\mathcal{D}$, some set of m points in the parameter space $\mathcal{D} = \{x_1, \ldots, x_m\}$, our training set $\mathcal{Y}$ is the set of model outputs $\mathcal{Y} = \{Y_m(x_1), \ldots, Y_m(x_d)\}$. Supposing that we specify a suitable prior covariance function and a prior mean, setting aside issues of estimating their parametric forms, we obtain a GP surrogate

$$\eta(\cdot) \sim \mathrm{GP}\left(\bar{\mu}(\cdot), \mathcal{C}(\cdot, \cdot)\right), \tag{8.1}$$

with mean and variance given by (3.17) and (3.18) for instance. If we integrate out the coefficients β of a prior mean model along with the total scale then the resulting emulator is strictly speaking Student-t distributed (see § 3.9)

We can decompose the emulated model $\eta(\cdot)$ into its main effects and interactions

$$\eta(\mathbf{x}) = \mathrm{E}[Y] + \sum_{i=1}^{d} z_i(x_i) + \sum_{i<j} z_{i,j}(\mathbf{x}_{i,j}) + \sum_{i<j<k} z_{i,j,k}(\mathbf{x}_{i,j,k}) + \ldots + z_{1,2,\ldots,d}(\mathbf{x}) \quad (8.2)$$

where we have

$$z_i(x_i) = \mathrm{E}[Y \mid x_i] - \mathrm{E}[Y], \quad (8.3)$$

$$z_{i,j}(\mathbf{x}_{i,j}) = \mathrm{E}[Y \mid \mathbf{x}_{i,j}] - z_i(x_i) - z_j(x_j) - \mathrm{E}[Y], \quad (8.4)$$

$$z_{i,j,k}(\mathbf{x}_{i,j,k}) = \mathrm{E}[Y \mid \mathbf{x}_{i,j,k}] - z_{i,k}(\mathbf{x}_{i,k}) - z_{j,k}(\mathbf{x}_{j,k}) - z_i(x_i) - z_j(x_j) - z_k(x_k) - \mathrm{E}[Y]. \quad (8.5)$$

and the higher order terms follow naturally. The *main effect* to x_i is $z_i(x_i)$, the *first order interaction* is $z_{i.j}(\mathbf{x}_{i,j})$. This is a decomposition of the model in terms of its mean over the whole space $\mathrm{E}[Y]$ and a series of progressively more complex terms which isolate the influence on the model output of a given set of parameters.

These terms depend upon the distribution G of the uncertain inputs. Computing and plotting the main effects and first order interactions is my main goal, this should provide a strong indication of how the model output depends upon each input and how "tangled" these influences become.

8.1.1 Notation

Above we introduced $\mathscr{D} = \{x_1, \ldots, x_m\}$ the design, this is not to be confused with the vector $\mathbf{x}^{\mathsf{T}} = (x_1, \ldots, x_d)$ of the model inputs, I shall try and make it clear by denoting elements of the design set as $\tilde{x}_a$ and elements of the input vector x_i. A sub vector (x_i, x_j) is given by $\mathbf{x}_{i,j}$ and for a general set of indicies $q = \{x_i, \ldots, x_k\}$ then $\mathbf{x}_q$ is the sub vector of $\mathbf{x}$ whose elements have those indices. Further $\mathbf{x}_{-q}$ is the sub vector of $\mathbf{x}$ which excludes the set of indices q.

8.1.2 Inference for effects

We want to know about:

$$\mathrm{E}[Y \mid \mathbf{x}_p] = \int_{\chi_{-p}} \eta(\mathbf{x}) dG_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_p) \tag{8.6}$$

where χ_{-p} is the space of possible values of $\mathbf{x}_{-p}$. Since this is a linear functional of a Gaussian Process we can derive the posterior mean as follows.

Recalling the results from § 3.9 the posterior mean and variance of our GP emulator after integrating out our priors are

$$\bar{m}_1(x_\star) = h(x_\star)\hat{\beta} + K^\intercal_{\star,\bullet}(\mathcal{Y} - H_\bullet\hat{\beta}), \tag{8.7}$$

$$\begin{aligned}&\bar{V}_1(x_{\star,\star'}) = \hat{\sigma}^2\bar{V}_0(x^\star, x^{\star'})+\\ &\hat{\sigma}^2\left[(h(x^\star) - K^\intercal_{\star,\bullet}K^{-1}_{\bullet,\bullet}H_\bullet)^\intercal(H^\intercal_\bullet K^{-1}_{\bullet,\bullet}H_\bullet)^{-1}(h(x^\star) - K^\intercal_{\star,\bullet}K^{-1}_{\bullet,\bullet}H_\bullet)\right]\end{aligned} \tag{8.8}$$

with

$$\bar{V}_0(x^\star, x^{\star'}) = \left\{\mathcal{C}(x^\star, x^{\star'}; \Theta) - K^\intercal_{\star,\bullet}K^{-1}_{\bullet,\bullet}K_{\star',\bullet}\right\}$$

$$\hat{\sigma}^2 = \frac{1}{m-q-2}\mathcal{Y}^\intercal\left(K^{-1}_{\bullet,\bullet} - K^{-1}_{\bullet,\bullet}H_\bullet\left(H^\intercal_\bullet K^{-1}_{\bullet,\bullet}H_\bullet\right)^{-1}H^\intercal_\bullet K^{-1}_{\bullet,\bullet}\right)\mathcal{Y}$$

$$\hat{\beta} = \left(H^\intercal_\bullet K^{-1}_{\bullet,\bullet}H_\bullet\right)^{-1}H^\intercal_\bullet K^{-1}_{\bullet,\bullet}\mathcal{Y}$$

Quantities defined with respect to the posterior distribution of $\eta(\cdot)$ are denoted by $F^\star$ etc.

$$\mathrm{E}^\star\{\mathrm{E}[Y \mid \mathbf{x}_p]\} = R_p(\mathbf{x}_p)\hat{\beta} + T_p(\mathbf{x}_p)\mathbf{e} \tag{8.9}$$

where

$$R_p(\mathbf{x}_p) = \int_{\chi_{-p}} \mathbf{h}(\mathbf{x})^T dG_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_p), \tag{8.10}$$

$$T_p(\mathbf{x}_p) = \int_{\chi_{-p}} K^\intercal_{\star,\bullet}(\mathbf{x}) dG_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_p), \tag{8.11}$$

$$\mathbf{e} = \left(\mathcal{Y} - H_\bullet\hat{\beta}\right) \tag{8.12}$$

Just to be clear R_p is a vector of length q where q is the number of regression functions, T_p is a vector of length d where d is the number of design points, as is e. It is then just a matter of writing down the posterior mean of the main effects and first order interactions

$$\mathrm{E}^{\star}\{z_i(x_i)\} = \{R_i(x_i) - R\}\hat{\beta} + \{T_i x_i - T\}\mathbf{e}, \tag{8.13}$$

$$\begin{aligned} \mathrm{E}^{\star}\{z_{i,j}(\mathbf{x}_{i,j})\} = &\{R_{i,j}(\mathbf{x}_{i,j}) - R_i(x_i) - R_j(x_j) - R\}\hat{\beta} + \\ &\{T_{i,j}(\mathbf{x}_{i,j}) - T_i(x_i) - T_j(x_j) - T\}\mathbf{e}. \end{aligned} \tag{8.14}$$

Finally we would certainly like to the variances of our estimated posterior means, we can derive them from the general result

$$\begin{aligned} &\mathrm{cov}^{\star}\{\mathrm{E}[Y \mid \mathbf{x}_p], \mathrm{E}[Y \mid \mathbf{x}'_q]\} \\ &= \hat{\sigma}^2 \int_{\mathcal{X}_{-p}} \int_{\mathcal{X}_{-q}} \bar{V}_1(\mathbf{x}, \mathbf{x}') dG_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_{-q}) dG_{-q|q}(\mathbf{x}'_{-q} \mid \mathbf{x}'_q) \\ , &= \hat{\sigma}^2 \left[U_{p;q}(\mathbf{x}_p, \mathbf{x}'_q) - T_p(\mathbf{x}_p) A^{-1} T_q(\mathbf{x}'_q)^T \right. \\ &+ \left. \left\{ R_p(\mathbf{x}_p) - T_p(\mathbf{x}_p) A^{-1} H_{\bullet} \right\} W \left\{ R_q(\mathbf{x}'_q) - T_q(\mathbf{x}'_q) A^{-1} H_{\bullet} \right\}^T \right], \end{aligned} \tag{8.15}$$

where we have defined

$$U_{p;q}(\mathbf{x}_p, \mathbf{x}'_q) = \int_{\mathcal{X}_{-p}} \int_{\mathcal{X}_{-q}} c(\mathbf{x}, \mathbf{x}') dG_{-p|p}(\mathbf{x}_{-\mathbf{p}} \mid \mathbf{x}_{\mathbf{p}}) dG_{-q|q}(\mathbf{x}'_{-\mathbf{q}} \mid \mathbf{x}'_{\mathbf{q}}), \tag{8.16}$$

$$W = \left(H_{\bullet}^T A^{-1} H_{\bullet} \right)^{-1}. \tag{8.17}$$

While the above looks a bit daunting, if we make some reasonable assumptions about G, h and c we can readily obtain analytic expressions floor R_p, T_p and so forth, however I won't take up any more space by reproducing these here. Furthermore all of the integrals we have encountered here can be easily done numerically.

We can plot our posterior mean main-effects (and interactions) constructed from (8.9) using (8.15) to obtain say 2σ confidence intervals on either side. This gives a reasonably intuitive graphical representation of the sensitivity of the model output to a given input however we need to consider a decomposition of the total variance in the model output Y if we want to get a full understanding. Our posterior inference for the expected values of these effects and their posterior variances can never give us a really full understanding of the actual variability of the model output explained by the various inputs, since by construction we're starting from inference about the mean.

8.2 Inference for Variances

We can also consider the sensitivity of the output Y in terms of the reduction of the total variance which would be observed if we knew the value of one of the inputs x_i with certainty. This reduction of variance can be written schematically as

$$\Delta \mathrm{V}[Y] = \mathrm{V}[Y] - \mathrm{V}[Y \mid x_i],$$

since we don't actually know the true value of x_i we will compute the average reduction of the total variance over all values of x_i

$$\mathrm{E}\{\Delta \mathrm{V}[Y]\} = \mathrm{V}[Y] - \mathrm{E}\{\mathrm{V}[Y \mid x_i]\}. \tag{8.18}$$

Recalling the so called Adam and Eve formulas for conditional expectations and variances

$$\mathrm{E}[X \mid A] = \mathrm{E}\left\{\mathrm{E}[X \mid A] \mid B\right\}, \tag{8.19}$$

$$\mathrm{V}[X] = \mathrm{E}\left\{\mathrm{V}[X \mid A]\right\} + \mathrm{V}\left\{\mathrm{E}[X \mid A]\right\}, \tag{8.20}$$

using these we can simplify (8.18) to obtain a simple form for the variance reduction

$$V_i = \mathrm{E}\{\Delta \mathrm{V}[Y]\} = \mathrm{V}\left\{E[Y \mid x_i]\right\}, \quad S_i = \frac{V_i}{\mathrm{V}[Y]} \tag{8.21}$$

where S_i is the standardized value. Another construction that may be useful is V_{T_i} the remaining uncertainty in Y after all inputs but x_i are known with certainty,

$$V_{T_i} = \mathrm{V}[Y] - \mathrm{V}\left\{\mathrm{E}[Y \mid \mathbf{x}_i]\right\}, \tag{8.22}$$

$$S_{T_i} = \frac{V_{T_i}}{\mathrm{V}[Y]} = 1 - S_{-i}. \tag{8.23}$$

Oakley and O'Hagan refer to S_i as the *main effect index* of x_i and S_{T_i} as the *total effect index* of x_i. The main effect indices can be interpreted as giving the relative importance of the various inputs to the total uncertainty in the output. If we want to consider the influence of learning the true values of pairs (or more complex combinations) of parameters we must consider their joint contribution along with their individual contributions, i.e. we should compute the variance due to their joint effect

$$V_{i,j} = \mathrm{V}\left\{\mathrm{E}[Y \mid \mathbf{x}_{i,j}]\right\} = \mathrm{V}\left\{z_i(x_i) + z_j(x_j) + z_{i,j}(\mathbf{x}_{i,j})\right\}. \tag{8.24}$$

8.2.1 Variance Decomposition

If the distribution G on the inputs is such that the elements of x are independent then the total variance of the output Y can be decomposed into another series of terms relating to the main effects and interactions

$$\mathrm{V}[Y] = \sum_{i=1}^{d} W_i + \sum_{i<j} W_{i,j} + \sum_{i<j<k} W_{i,j,k} + \ldots + W_{1,2,\ldots,d}, \tag{8.25}$$

where $W_p = \mathrm{V}\left\{z_p(\mathbf{x}_p)\right\}$. In fact we can see that $W_i = V_i$ is the reduction in the total variance of Y obtained when we learn the true value of the i'th input. Further (8.24) can be written as $V_{i,j} = W_i + W_j + W_{i,j}$, so we can interpret $W_{i,j}$ as the extra variance removed after learning the true value of both the i'th and j'th parameters.

When we make such a decomposition of the total variance then we see that V_{-i} is the sum of all the W_p terms appearing in (8.25) which do not include the i'th point. As such the total effect index $S_{T_i} = 1 - S_{-i}$ is the proportion of the total variance accounted for by all the effect and interaction terms in (8.25) which do involve the i'th parameter.

8.2.2 *Inference for the Variance Decomposition*

We would like to carry out inference for the posterior means and variances, after constructing a GP emulator of the model output, of the various terms in (8.25), as we did in § 8.1.2 for the decomposition of the mean. The complexity of the integrals relative to how instructive they are rapidly gets out of hand here. I will attempt to illustrate the calculation of the posterior mean of $V_p = \mathrm{V}\left\{\mathrm{E}(Y \mid \mathbf{x}_p)\right\}$ which is part of the variance contribution of a sub-vector of p inputs, the posterior variance of V_p can be obtained but its complexity to information ratio is sufficient to prohibit reproduction. Invoking the Eden formulae (8.19) and the definition of variance

$$\begin{aligned} V_p = \mathrm{V}\left\{\mathrm{E}(Y \mid \mathbf{x}_p)\right\} &= \mathrm{E}\left\{\mathrm{E}(Y \mid \mathbf{x}_p)^2\right\} - \mathrm{E}\left\{\mathrm{E}(Y \mid \mathbf{x}_p)\right\}^2, \\ &= \mathrm{E}\left\{\mathrm{E}(Y \mid \mathbf{x}_p)^2\right\} - \mathrm{E}(Y)^2. \end{aligned}$$

Above in § 8.1.2 we computed $\mathrm{E}^\star\{\mathrm{E}[Y]\}$ in (8.9) and $\mathrm{V}^\star\{\mathrm{E}(Y)\}$ in (8.15), these are all we need to obtain $\mathrm{E}^\star\{\mathrm{E}(Y)^2\}$. The remaining term we need for the posterior mean of V_p is

$$\begin{aligned} &\mathrm{E}^\star[E\{E(Y \mid \mathbf{x}_p)^2\}] \\ &= \int_{\chi_p}\int_{\chi_{-p}}\int_{\chi_{-p}} E^\star\{\eta(x)\eta(x^\circ)\} dG_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_p) dG_{-p|p}(\mathbf{x}'_{-p} \mid \mathbf{x}_p) dG(\mathbf{x}_p) \\ &= \int_{\chi_p}\int_{\chi_{-p}}\int_{\chi_{-p}} \left[\bar{V}_1(\mathbf{x}, \mathbf{x}^\circ) + \bar{m}_1(\mathbf{x})\bar{m}_1(\mathbf{x}^\circ)\right] dG_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_p) dG_{-p|p}(\mathbf{x}'_{-p} \mid \mathbf{x}_p) dG(\mathbf{x}_p) \end{aligned}$$

where here $\mathbf{x}^\circ$ is the vector made from the elements $\mathbf{x}_p$ and $\mathbf{x}'_{-p}$ just as $\mathbf{x}$ is the vector constructed from the elements $\mathbf{x}_p$ and $\mathbf{x}_{-p}$, in general the set p may not be a trivial partition of the d possible elements there isn't really a neater way to represent this.

Writing the measure as $d\Gamma_p(\mathbf{x}_{-p}, \mathbf{x}'_{-p}, \mathbf{x}) = dG_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_p) dG_{-p|p}(\mathbf{x}'_{-p} \mid \mathbf{x}_p) dG(\mathbf{x}_p)$, the integrals over the posterior variance and mean are

$$\int_{\chi_p} \int_{\chi_{-p}} \int_{\chi_{-p}} \left[\bar{V}_1(\mathbf{x}, \mathbf{x}^\circ)\right] \, d\Gamma_p(\mathbf{x}_{-p}, \mathbf{x}'_{-p}, \mathbf{x})$$

$$= \hat{\sigma^2} \left\{ U_p - \mathrm{tr}(K^{-1}_{\bullet,\bullet} P_p) + \mathrm{tr}\left(W(Q_p - S_p K^{-1}_{\bullet,\bullet} H_\bullet - H^\intercal_\bullet K^{-1}_{\bullet,\bullet} S^\intercal_p + H^\intercal_\bullet K^{-1}_{\bullet,\bullet} P_p K^{-1}_{\bullet,\bullet} H_\bullet \right) \right\}$$

$$\int_{\chi_p} \int_{\chi_{-p}} \int_{\chi_{-p}} \left[\bar{m}_1(\mathbf{x}) \bar{m}_1(\mathbf{x}^\circ)\right] \, d\Gamma_p(\mathbf{x}_{-p}, \mathbf{x}'_{-p}, \mathbf{x}) = \mathrm{tr}(\mathbf{e}^\intercal P_p \mathbf{e}) + 2\mathrm{tr}(\hat{\beta} S_p \mathbf{e}) + \mathrm{tr}(\hat{\beta} Q_p \hat{\beta})$$

These are given interms of the integrals

$$U_p = \int_{\chi_p} \int_{\chi_{-p}} \int_{\chi_{-p}} \bar{V}_1(\mathbf{x}, \mathbf{x}^\circ) \, d\Gamma_p(\mathbf{x}_{-p}, \mathbf{x}'_{-p}, \mathbf{x}),$$

$$P_p = \int_{\chi_p} \int_{\chi_{-p}} \int_{\chi_{-p}} K_{\star,\bullet}(\mathbf{x}) K^\intercal_{\star,\bullet}(\mathbf{x}^\circ) \, d\Gamma_p(\mathbf{x}_{-p}, \mathbf{x}'_{-p}, \mathbf{x}),$$

$$Q_p = \int_{\chi_p} \int_{\chi_{-p}} \int_{\chi_{-p}} h(\mathbf{x}) h(\mathbf{x}^\circ) \, d\Gamma_p(\mathbf{x}_{-p}, \mathbf{x}'_{-p}, \mathbf{x}),$$

$$S_p = \int_{\chi_p} \int_{\chi_{-p}} \int_{\chi_{-p}} h(\mathbf{x}) K_{\star,\bullet}(\mathbf{x}^\circ) \, d\Gamma_p(\mathbf{x}_{-p}, \mathbf{x}'_{-p}, \mathbf{x}).$$

From this we can obtain the posterior means of the main effect variances V_i and their complementary quantities V_{T_i}, ideally we want to look at the standardized quantities S_i and S_{T_i} however

$$\mathrm{E}^\star\{S_i\} = \mathrm{E}^\star\left[\frac{V_i}{\mathrm{V}(Y)}\right] \neq \frac{\mathrm{E}^\star\{V_i\}}{\mathrm{E}^\star\{\mathrm{V}(Y)\}}.$$

Regardless of this the quantities on the far right hand side above are still useful to examine.

8.3 A toy example

Let's consider the following simple three parameter model

$$Y_m(x) = \exp(-0.1x_1^2 + 0.2x_2) + 0.4\sin(x_3) + 0.1x_1x_2 + 0.4x_2x_3, \tag{8.26}$$

after training a GP emulator on a 64 point LHS design we can estimate the main effects and interactions (§ 8.1.2) and their contributions to the total variance. In the top left panel Fig: 8.1 the main effects clearly show the sinusoidal term in x_3 and vaguely hint at the Gaussian term in x_1, however the role of x_2 is not very clear. If we were to purely judge by this figure we might expect x_3 to be the dominant variable. However after examining the pairwise terms we see that the range of values in the plot of the joint effect $z_{1,2}(\mathbf{x}_{1,2})$ is significantly larger than any of the others, suggesting that this interaction is very important. Finally we might conclude that x_1 and x_3 are relatively independent of each other.

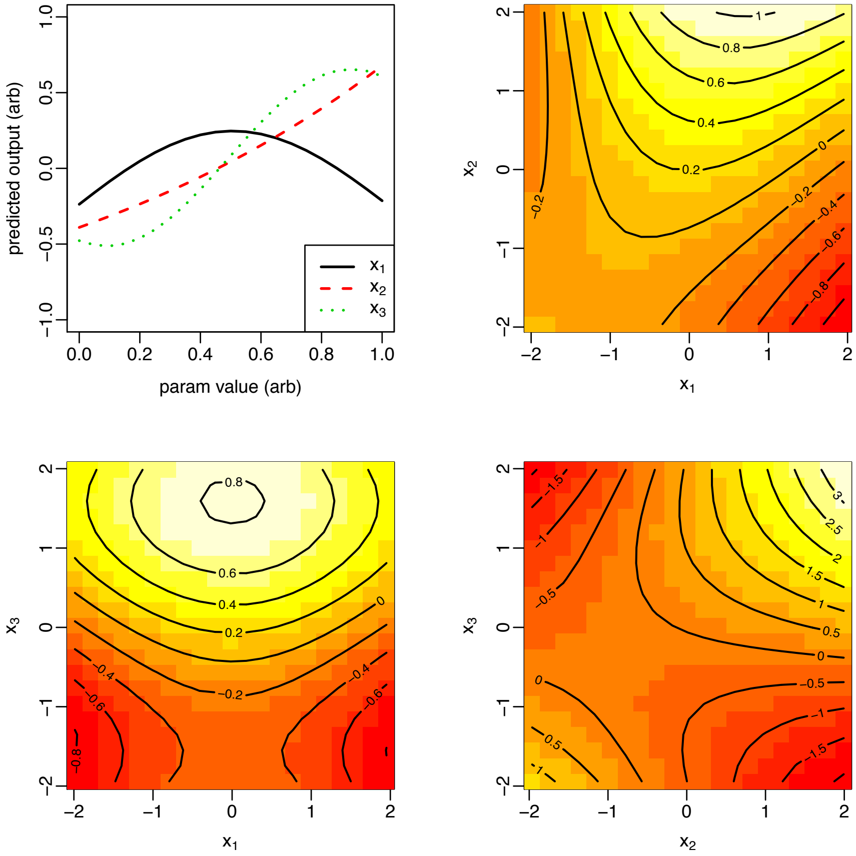


FIGURE 8.1: Posterior mean main effects and interactions for the toy model (8.26). Top left panel: the posterior mean main effects $E^{\star}\{z_i(\mathbf{x}_i)\}$ for each parameter, the remaining panels show the pairwise interactions $E^{\star}\{z_{i,j}(\mathbf{x}_{i,j})\}$.

Table 8.1: Posterior mean contributions to the total observed variance for the parameters and their first interactions, the interaction between x_2 and x_3 dominates.

param	$100\frac{\mathrm{E}^\star\{V_i\}}{\mathrm{E}^\star\{\mathrm{V}(Y)\}}$
p1	2.34
p2	10.50
p3	22.36
p1:p2	3.98
p1:p3	0.00
p2:p3	60.80
total	99.8

After consulting Table: 8.1 the table of estimated effect indexes S_i it is immediately clear that the interaction of x_2 and x_3 dominate, with the factor of three smaller x_3 contribution as the next most important term. This is in line with what we would expect from the form of the model (8.26). Finally we can conclude from the table that at least up to the accuracy of our GP emulator the x_1 and x_3 variables are indeed independent.

8.4 Application to ChemTreeN

In chapter 6 I outlined an analysis of the hybrid galaxy formation model ChemtreeN. In the recent article [3] we continued to develop our understanding of the model and applied the GP emulator based sensitivity analysis techniques described in this chapter to examine the influence of an extended set of model inputs. The training data is essentially the same as that described in § 6.2. The main changes are as follows: we increased the number of slices through the luminosity function to eight these now span $M_V = [-3.5, \ldots, -17.5]$; we switched from the linear fit to the average metallicity-luminosity function (as shown in the right hand panel of Fig: 6.1) to the cumulative distribution of average metallicity which was then summarized by slicing it at four relatively equally spaced values; finally we increased the number of calibration parameters set to seven (see Table: 6.1), f_{1a} and ϵ_{SN} were also explored. For more details on the model and data see § 2 of [3].

In Fig: 8.2 the main effects for the eight slicings of the luminosity function are plotted, the seven calibration parameters are all plotted on the same standardized scale. From this figure it is possible to infer what parameters are most important to explaining the variability observed on each observable. The main effects for all the luminosity bins are dominated by f_{bary} (solid black) and Z_r (solid red), it is interesting to note that the dominant effect switches between f_{bary} for the luminosity bins $M_v = [-17.5, \ldots, -11.5]$ to Z_r. Note as well that some parameters, such as $m_{\mathrm{Fe}}^{\mathrm{II}}$ and f_{Ia}, do not show a strong influence on the values of the selected observables. In Fig: 8.3 the main effects for the four metallicity bins are plotted, careful examination of these reveals that they are dominated by an entirely different set of parameters than the luminosity results.

Fig: 8.4 shows the results of applying the variance decomposition methods described above to the expanded ChemTreeN model, for eight bins through the luminosity function. This graphical representation of the variance decomposition allow us to quickly identify what input parameters are more important on explaining the variability observed on each observable. The results largely confirm our intuition developed from the plots of the main effects. The total variance in the dimmer bins $-17.5, -15.5, -13.5$ is dominated by contributions from the baryon fraction f_{bary}, at $M_v = -11.5$ and brighter however the epoch of reionization becomes an increasingly important factor. Further in this figure we can see that the interaction between these two variables is non trivial, this information was not at all obvious from the plots of the mean decomposition. A similar figure can be made for the metallicity results. These results proved very useful in the selection of observables and inputs needed for further analysis of the model, see § 4–5 of [3].

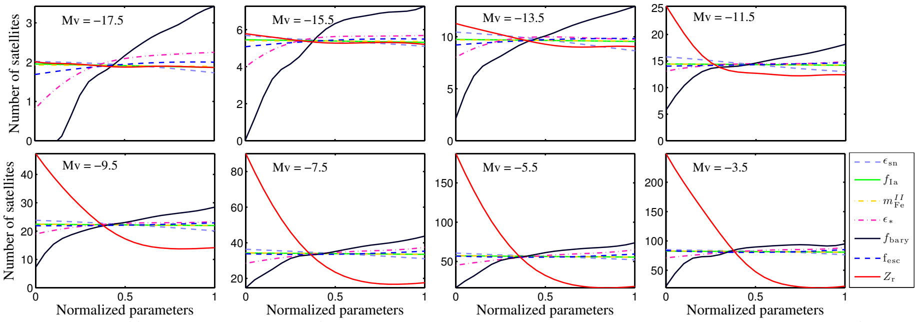


FIGURE 8.2: Main effects obtained from a Gaussian process model emulator of ChemTreeN, with *seven* different input variables. The results were obtained using the simulation labeled MW1. The panels show the results for different bins in the luminosity function, as indicated in the top left corner of each panel. The plotted lines show the main effect associated with a different input variable, as indicated in the legend located at the bottom right corner.

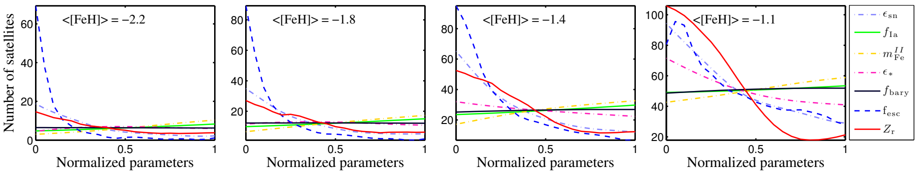


FIGURE 8.3: As in Fig: 8.2, now for mock observables obtained from the cumulative number of satellite galaxies as a function of mean metallicity, $\langle$[Fe/H]$\rangle$.

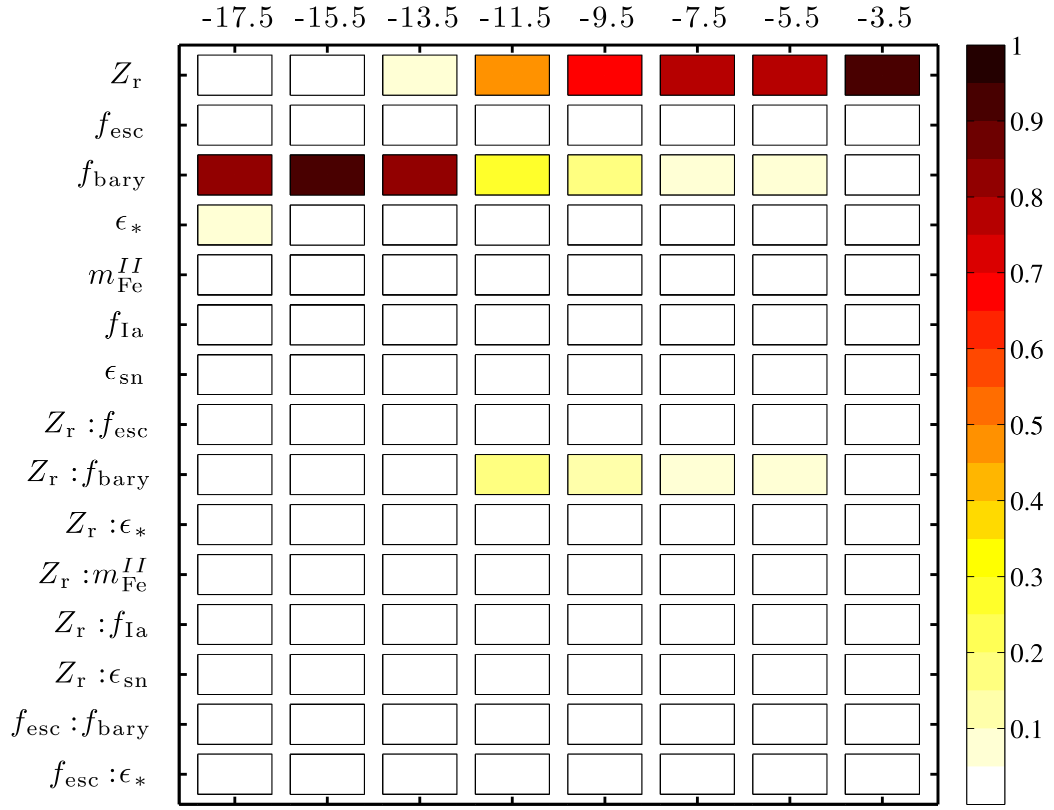


FIGURE 8.4: Variance decomposition (see 8.2.1) obtained a Gaussian process model emulator of ChemTreeN with *seven* input variables. The results were obtained using the simulation labeled MW1. The different columns correspond to different observables, rows are associated to variance contribution V_p for either main effects or interactions. The columns correspond to different bins of the luminosity function. We only consider up to two-variable interaction effects. Note that, for simplicity, not all interaction effects are shown. The different colors indicate the percentage of the total variance that can be explained by the corresponding effect.

9

Calibration

In this chapter we turn our attention back to the primary concern, that of finding a set of true or best values for our unknown model parameters, model calibration. Excellent references for calibration of computer models using GP emulators are the papers by Kennedy and O'Hagan [5, 13] and Higdon et al [15, 16]. In this chapter I will follow the general lines of the concise methodology outlined in [15] as this is readily generalized to treat simulators with multivariate output as discussed in [16].

Suppose we have a simulator $Y_m(x, u)$ which has observation parameters x and calibration parameters u and we are interested in using field observations $Y_f(x)$ to learn about the 'true' values $u_\star$ of the calibration parameters. In the introduction I laid out the following relations between measured field data $Y_f(x)$ which is measured with some observation error $\epsilon_f(x)$, the true physical process $Y_r(x, u_\star)$ and the simulator output $Y_m(x, u)$

$$\begin{aligned} Y_f(x) &= Y_r(x, u_\star) + \epsilon_f(x), \\ Y_r(x, u_\star) &= Y_m(x, u_\star) + b(x, u_\star). \end{aligned} \tag{9.1}$$

Let's suppose that we obtain a set of n observations of the field process Y_f at locations $\mathscr{D}_f = \{x_1, \ldots, x_n\}$. To simplify the analysis here lets further suppose that we understand the experimental data collection process enough to be able to accurately characterize the observational errors with some distribution, typically this will be a multivariate normal with some correlation structure.

Let's restrict ourselves to the case of simulators that produce only a single output, the processes outlined below can be generalized to work with multivariate output using a suitable decomposition (see [16]), such as those discussed in chapter 5.

9.1 Fast Faithful Model

The simplest calibration case we can address is one where the computer model Y_m is sufficiently fast that we can effectively make an unlimited number of observations of the model at any location in the x, u parameter space that we wish.

Furthermore lets suppose that the simulator $Y_m(x, u)$ *faithfully* simulates the true physical $Y_r(x, u^\star)$ system when evaluated at the true, but currently unknown, values $u = u^\star$. Under this assumption we can simplify our model (9.1) to

$$Y_f(x_i) = Y_m(x_i, u_\star) + \epsilon(x_i), \quad i = 1 \ldots n, \tag{9.2}$$

where the n values $x_i \in \mathscr{D}_f$ are the settings where the field observations are made. It's important to note that at this point we don't actually *know* the values $u_\star$, we will model these as a random variable and use the field data and the model to make inference about their values. Taking the field observation errors as independent normal with some known standard deviation σ_f, we can write the likelihood of the vector of n observations $y_f = (Y_f(x_1), \ldots, Y_f(x_n))^\intercal$ as

$$L\left(y_f \mid Y_m(u_\star)\right) \propto \exp\left\{-\frac{1}{2}(y_f - Y_m(u_\star))^\intercal \Sigma_f^{-1}(y_f - Y_m(u_\star))\right\}, \tag{9.3}$$

where the n element vector $Y_m(u_\star) = (Y_m(x_1, u_\star), \ldots, Y_m(x_n, u_\star))^\intercal$ and $\Sigma_f = \sigma_f^2 \mathbb{I}_n$. This is to be interpreted as the probability for observing the field data y_f given the set of model outputs $Y_m(u_\star)$. Treating $u_\star$ as a random variable we introduce a prior distribution $\pi(u_\star)$ which captures our prior uncertainty about the true calibration values. Note that we are only sampling our model at the x parameter values that we have field data for, this is reasonable since we believe the model is faithful at this stage. The posterior distribution for $u_\star$ given our prior and the observations y_f is then

$$\pi\left(u_\star \mid y_f\right) \propto L\big(y_f \mid Y_m(u_\star)\big)\pi(u_\star). \tag{9.4}$$

Typically the full form of this posterior is intractable, unless our model is a very simple function we will not be able to proceed much further algebraically. However we can use Markov Chain Monte Carlo (MCMC) [171, 169, 170] to generate a series of samples $u_\star^1, \ldots, us^{N_{MC}}$, if we generate enough samples then their empirical distribution will (eventually) converge to the distribution of the posterior $\pi\left(u_\star \mid y_f\right)$.

9.1.1 *Metropolis MCMC Algorithm*

The Metropolis algorithm [212] is a simple but effective implementation of MCMC, it may be may well be familiar as it is the typical process introduced to numerically explore the Ising ferromagnetic model [213]. A common feature of MCMC algorithms is that they typically scale very well with the dimensionality of the distribution being sampled.

I will outline the algorithm with the variables $u_\star$ introduced above. The procedure itself is quite general and can be easily adapted to many situations where one wants samples of some posterior distribution whose full form would be prohibitively difficult to obtain.

1. Pick some initial value for the calibration parameters $u_\star^1$. The particular value

is theoretically not important since the sampling procedure will "thermalize" to the target distribution $\pi\left(u_{\star} \mid y_{f}\right)$ effectively forgetting the particular choice $u_{\star}^{1}$.

2. At step t the current sample is $u_{\star}^{t}$, generate a new proposed sample $u_{\star}^{\prime}$ from some symmetric distribution, i.e. the proposal distribution must satisfy $P(u_{\star}^{t} \mid u_{\star}^{\prime})=P(u_{\star}^{\prime} \mid u_{\star}^{t})$.

3. Compute the Metropolis acceptance ratio

$$\alpha=\min \left\{1, \frac{\pi\left(u_{\star}^{\prime} \mid y_{f}\right)}{\pi\left(u_{\star}^{t} \mid y_{f}\right)}\right\} . \tag{9.5}$$

4. Accept the proposed step, $\theta^{t+1}=\theta^{\star}$ with probability α, otherwise reject the proposal and set $\theta^{t+1}=\theta^{t}$. This can be done by generating a standard uniformly distributed random number r and accepting the proposed move if $r \leqslant \alpha$.

5. Iterate steps 2-4.

One of the great advantages of this procedure is that the posterior density only enters in a ratio with itself, as such we only need to specify the terms which do not cancel. In this case we can directly insert the product of $\pi(u_{\star})$ and (9.3).

Given a chain of draws $u_{\star}^{1}, \ldots, us^{N_{MC}}$ obtained from the MCMC procedure we can histogram them to obtain an estimate of the posterior $\pi(u_{\star} \mid y_{f})$. A good first order of business is to compare this histogram with that of the prior distribution, if the posterior histogram is more concentrated in the parameter space than the prior then our observations have reduced our uncertainty in the true values $u_{\star}$[1]. Sample estimates of moments of the chain (such as the mean and variance etc) are also estimates for the corresponding moments of the posterior distribution.

[1] which is usually the whole point of the exercise

Table 9.1: A comparison of the prior $\pi(u_\star)$ and MCMC posterior $\pi(u_\star \mid y_f)$, the prior ranges are simply the appropriate normal quantiles.

	HPD lower (95%)	mean	HPD upper (95%)
prior	0.089	0.5	0.911
posterior	0.556	0.629	0.707

If the posterior is sufficiently peaked we might report the posterior mean and variance of $u_\star$ as a summary of the calibration procedure, credible intervals may also be useful here, see [170].

9.1.2 A toy example

Lets consider the following toy model,

$$Y_m(x,u) = 5x^2\exp(-3x^2)\sin(x-u)+2, \quad x \in [0,2] \tag{9.6}$$

given $n = 4$ observations equally spaced in x can we infer the true calibration parameter u_{star}? Taking the observation errors as i.i.d normal with standard deviation $\sigma_f = 0.025$ and a normal prior distribution $\pi(u_\star) \sim \mathrm{N}(0.5, 0.25^2)$. Results of using Metropolis MCMC procedure to sample (9.4) are shown in Fig: 9.1 and summarized in Table: 9.1. Here 10000 Metropolis steps were used with a normal proposal distribution centered on the current value $u'_\star \mid u^t_\star \sim \mathrm{N}(u^t_\star, 0.3^2)$. Given the uncertainty in the field observations we should be rather satisfied with the results of this procedure, we have strongly reduced the variability in our model function so that posterior draws typically fall within the 95% confidence intervals associated with our field data.

9.2 Slow Faithful Model

Now lets consider the slightly more realistic situation where our simulator is sufficiently complex that we can only obtain a finite number d of runs $\mathcal{Y} = \{Y_m(x_1,u_1),\ldots,Y_m(x_d,u_d)\}$ generated from running the simulator at some design

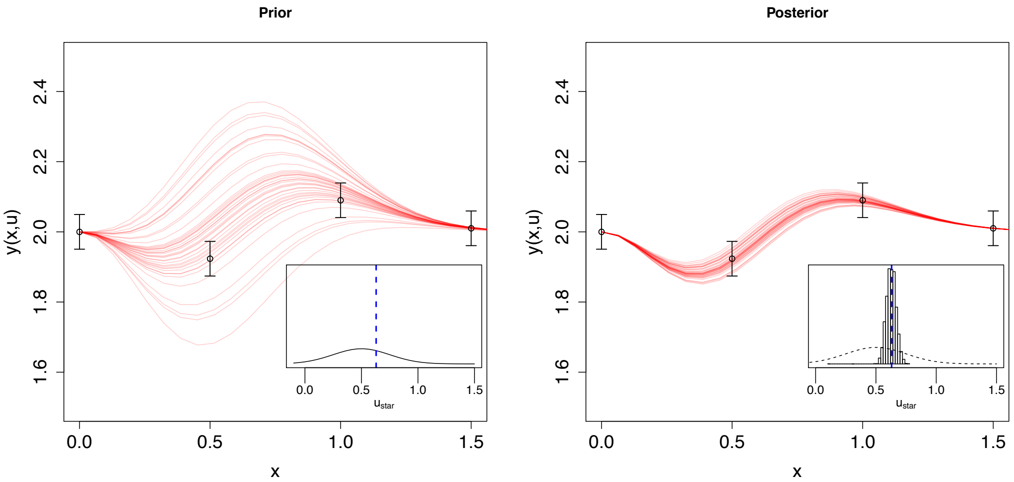


FIGURE 9.1: Left: the four field observations making up y_F are plotted as open circle with 95% confidence intervals, samples of $Y_m(x, u_\star)$ with $u_\star$ drawn from the prior $\pi(u_\star)$ are shown in light red. The inset figure shows the prior density, the true value is shown as the dashed blue line. Right: the light red curves are plots of $Y_m(x, u_\star)$ with $u_\star$ drawn from the MCMC posterior which approximates (9.4). The inset figure shows the prior density $\pi(u_\star)$ (dashed) and the MCMC posterior density $\propto \pi\left(u_\star \mid y_f\right)$.

$\mathcal{D} = \{(x_1, u_1), \ldots, (x_d, u_d)\}$. Now we have to treat the simulator output $Y_m(x, u)$ as being unknown when evaluated at locations not in the design $\mathcal{D}$. Let's take the total dimension of the parameter space as $p = p_x + p_u$ where p_u is the number of calibration parameters and p_x is the number of observation parameters. Placing a GP prior on the simulator with a constant mean μ and a power exponential prior covariance function

$$\mathcal{C}((x, u), (x', u')) = \frac{1}{\lambda_m} \exp\left\{ -\sum_{k=1}^{p_x} \frac{(x_k - x'_k)^\alpha}{(\beta_k^m)^\alpha} - \sum_{k=1}^{p_u} \frac{(u_k - u'_k)^\alpha}{(\beta_{p_x+k}^m)^\alpha} \right\} \tag{9.7}$$

where the p_x quantities β_k^m are the length scales for the observation and calibration parameters and λ_m is the marginal precision.

The model (9.2) is again appropriate here. We have n field observations y_f with $x_i \in \mathcal{D}_f$ and a set of d observations of our simulator $\mathcal{Y}$ with $(x_i, u_i) \in \mathcal{D}$. We can introduce the $n+d$ length vector $z = (y_f^\intercal, \mathcal{Y}^\intercal)^\intercal$ which corresponds to input settings $\mathcal{D}_z = \{(x_1, u_\star), \ldots, (x_n, u_\star), (x_1, u_1), \ldots (x_d, u_d)\}$. The first n observation parameter settings in z are from $\mathcal{D}_f$ with the calibration parameters set to their *unknown* true values. The remaining d sets of observation and calibration parameters are set by the simulator design $\mathcal{D}$.

Taking the same model for the field observations as above, we can write the likelihood of our vector of samples and observations z given a value of the 'true' parameters $u_\star$ along with values of λ_m, η which specify the length scales in our GP,

$$L(z \mid u_\star, \mu, \lambda_m, \beta^m, \Sigma_f) \propto |\Sigma_z|^{-1/2} \exp\left\{ -\frac{1}{2} (z - \mu \mathbb{I}_{n+d})^\intercal \Sigma_z^{-1} (z - \mu \mathbb{I}_{n+d}) \right\}, \tag{9.8}$$

where

$$\Sigma_z = \Sigma_m + \begin{pmatrix} \Sigma_f & 0 \\ 0 & 0 \end{pmatrix}, \quad \Sigma_m = \begin{pmatrix} \Sigma_{y_f y_f} & \Sigma_{y_f \mathcal{Y}} \\ \Sigma_{y_f \mathcal{Y}}^\intercal & \Sigma_{\mathcal{Y}\mathcal{Y}} \end{pmatrix}$$

and Σ_m is the $(n+d \times n+d)$ matrix obtained by applying (9.7) to every pair of inputs in the augmented set $\mathcal{D}_z$. When we sample the posterior associated with this likelihood and appropriate priors for the GP parameters and $u_\star$ we will be effectively estimating the distribution for $u_\star$ as well as the distribution for the parameters controlling GP covariance structure . While this is elegant one could always insert the maximum likelihood estimates for the GP parameters λ_m, μ and β obtained using the methods outlined in § 3.8 treating them as fixed quantities and then carry out MCMC sampling for the unknown calibration parameters $u_\star$.

Scaling the input parameter space onto the unit hyper cube $[0, 1]^{p_x + p_u}$ and centering and scaling the model output data so that $\mathcal{Y}$ has unit sample variance simplifies the prior specification process. With the parameter space mapped onto the unit cube we can identify unimportant parameters as those whose estimated length scale is approximately 1. Taking a gamma prior for the marginal precision λ_m and beta priors on the length scales

$$\begin{aligned} \pi(\lambda_m) &\propto \lambda_m^{a_m - 1} e^{-b_m \lambda_m} \\ \pi(\beta_k^m) &\propto (\beta_k^m)^{a_\beta - 1} (1 - \rho_k)^{b_\beta - 1}, k = 1, \ldots, p_x + p_u. \end{aligned}$$

we can take $a_m = b_m = 5$ which pushes λ_m towards 1. For the correlation lengths we take $a_\beta = 1$ and $b_\beta = 0.1$, this makes the prior probability of a length scale being somewhat significant $P(\beta_k^m < 0.98) \approx \frac{1}{3}$. Centering the observations z allows us to simplify things by taking $\mu = 0$, if this is somehow not appropriate we can of course specify some prior form for the GP mean.

After conditioning on our vector of observations $z = (y_f^\intercal, \mathcal{Y}^\intercal)^\intercal$ we obtain the posterior

$$\pi(u_\star, \mu, \lambda_m, \beta^m \mid z) \propto L(z \mid u_\star, \mu, \lambda_m, \beta^m, \Sigma_f)\pi(u_\star)\pi(\mu)\pi(\lambda_m)\pi(\beta^m), \tag{9.9}$$

which we can again sample using MCMC methods. Given one such sample $(u_\star, \mu, \lambda_m, \beta^m)$ we can sample our GP emulator at any given point in the parameter space $Y_m(x', u')$ just as we would using the drop-in emulators discussed in previous chapters. Essentially we obtain the conditional distribution of the emulator at the new location given the simulator observations from their joint distribution using (A.10).

9.2.1 A toy model

Lets consider the previous toy model,

$$Y_m(x, u) = 5x^2 \exp(-3x^2)\sin(x - u) + 2, \quad x \in [0, 2] \tag{9.10}$$

given $n = 4$ observations equally spaced in x and a set of $d = 32$ observations of the simulator distributed in the $p = 2$ dimensional parameter space with a LHS design. Again we will take the observation errors as i.i.d normal with standard deviation $\sigma_f = 0.25$ and a normal prior distribution $\pi(u_\star) \sim \mathrm{N}(0.5, 0.25^2)$. The posterior mean of the resulting GP emulator $\bar{m}_1(x, u)$ is shown in Fig: 9.2, the values of $\beta^m = (0.0787, 0.142)$ and $\lambda_m = 1.138$ were randomly drawn from the $N_{\mathrm{MC}} = 30,000$ MCMC samples.

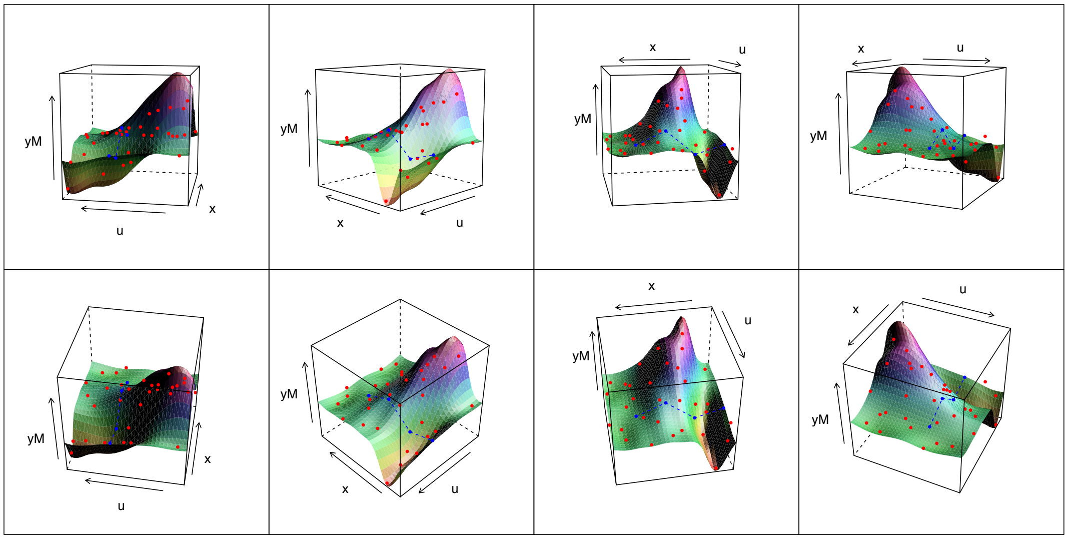


FIGURE 9.2: Several views of the posterior mean $\bar{m}_1(x, u)$ of a GP emulator developed from the 32 observations $\mathcal{Y}$ of (9.6). The red points are the training data set $\mathcal{Y}$, the blue points and line show the field observations. The emulator parameters, β^m, λ_m were drawn at random from the MCMC chain.

Table 9.2: A comparison of the prior $\pi(u_\star)$ and MCMC posterior $\pi(u_\star \mid z)$, the prior ranges are simply the appropriate normal quantiles.

	HPD lower (95%)	mean	HPD upper (95%)
prior	0.089	0.5	0.911
posterior	0.462	0.629	0.868

As in the previous example the left panel of Fig: 9.3 shows draws from the prior distribution for $u_\star$ as fine red lines. In addition the training data of 32 sample points (projected into the x dimension) are plotted as the solid points, these correspond with solid red points shown in Fig: 9.2. The right hand panel shows draws from the posterior density $\pi(u_\star \mid z)$, although somewhat noisy these are mostly well grouped around the true value of the model function (plotted as the blue solid line). The inset panel shows the posterior distribution for $u_\star$ as the histogram with solid bins, the prior density is drawn as the dashed line and the true value is drawn as the vertical dashed blue line.

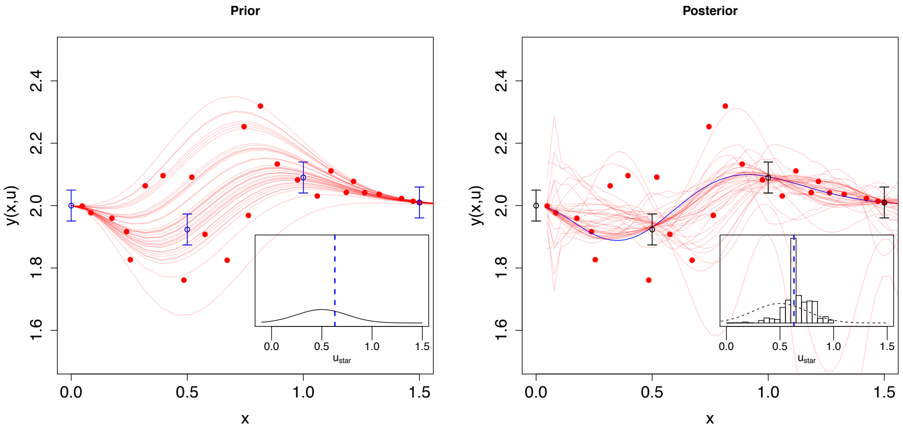


FIGURE 9.3: Left: the field observations y_f are plotted as blue open circles of (9.10), the training data $\mathcal{Y}$ projected into the x direction are plotted as red closed circles, the curves are draws from $Y_m(x, u_\star)$ with $u_\star$ drawn from the prior density. Right: the light red curves are plots of $Y_m(x, u_\star)$ with $u_\star$ drawn from the MCMC posterior which approximates (9.9). The inset figure shows the prior density $\pi(u_\star)$ (dashed) and the MCMC posterior density $\pi(u_\star, \mid z)$

Admittedly the performance is not quite so beautiful as in the case with the fast model in terms of the posterior draws. This is still a very good result given the relatively small number of training points. The posterior distribution for $u_\star$ is significantly constrained as shown in Table: 9.2. The performance could likely be improved, in the sense of posterior draws more perfectly approximating the true output, by increasing the number of MCMC samples and optimizing the proposal distributions and perhaps by considering alternative forms for the prior.

9.3 Slow Unfaithful Model

If we have reason to believe that there is a systematic difference between the output of our simulator and the observational data, i.e. that our model is no longer a faithful representation of reality, we may still be able to obtain some interesting information about the true values of the calibration parameters $u_\star$. Typically the "smaller" the discrepancy is the more we can learn about $u_\star$. We now adopt the model

$$Y_f(x_i) = Y_m(x_i, u_\star) + \delta(x_i) + \epsilon(x_i), \quad i = 1 \dots n, \tag{9.11}$$

where $\delta(x_i)$ is a function which represents the systematic deviation between our simulator and reality. We model the discrepancy with a mean zero Gaussian Process with covariance function

$$C_\delta(x, x') = \frac{1}{\lambda_\delta} \exp\left\{ -\frac{1}{2} \sum_{k=1}^{p_x} \frac{(x_k - x'_k)^\alpha}{(\beta_k^\delta)^\alpha} \right\}, \tag{9.12}$$

and take similar priors to those used above for the model GP

$$\pi(\lambda_\delta) \propto \lambda_\delta^{a_\delta - 1} e^{-b_\delta \lambda_m}, \tag{9.13}$$

$$\pi(\beta_k^\delta) \propto (\beta_k^\delta)^{a_{\beta\delta} - 1} (1 - \rho_k)^{b_{\beta\delta} - 1}, \quad k = 1, \dots, p_x. \tag{9.14}$$

Suggested values are given by Higdon et al as $a_\delta = 1$, $b_\delta = 0.11$ and $a_\beta^\delta = 1$, $b_\beta^\delta = 0.1$ in [16]. The likelihood for our augmented vector z is structurally the same as (9.8) with the modified covariance matrix

$$\Sigma_z = \Sigma_m + \begin{pmatrix} \Sigma_f + \Sigma_\delta & 0 \\ 0 & 0 \end{pmatrix}$$

where Σ_m is the $(n + d \times n + d)$ matrix obtained by applying (9.7) to every pair of inputs in the augmented set $\mathcal{D}_z$, Σ_f is the $(n \times n)$ covariance matrix of the field data, and Σ_δ is the $(n \times n)$ matrix obtained by evaluating (9.12) at every pair of points in the observation design $\mathcal{D}_f$. After conditioning on our vector of observations z the posterior is now

$$\pi(u_\star, \mu, \lambda_m, \beta^m, \lambda_\delta, \beta^\delta \mid z) \propto L(z \mid u_\star, \mu, \lambda_m, \beta^m, \lambda_\delta, \beta^\delta, \Sigma_f) \times \\ \pi(u_\star)\pi(\mu)\pi(\lambda_m)\pi(\beta^m)\pi(\beta^\delta)\pi(\lambda_\delta), \tag{9.15}$$

which can be sampled to obtain realizations of the vector $(u_\star, \mu, \lambda_m, \beta^m, \lambda_\delta, \beta^\delta)$. These realizations can be used to obtain posterior predictions for the model output at any point in the untried space $Y_m(x', u')$, the discrepancy function at any location $\delta(x')$ and the 'real' physical process at any point of interest $Y_r(x', u_\star) = Y_m(x', u_\star) + \delta(x')$.

9.4 A Heavy Ion Analysis

In [1] we considered a $3+1$d viscous hydrodynamic+microscopic transport model of Au+Au collisions at $\sqrt{S} = 200$ AGeV at RHIC, the model was developed by S.Pratt et al and is described in detail in the article.

Six calibration parameters were identified (see Table: 9.3), with four of these describing various aspects of the initial state and the two η/s and α describing viscous aspects of the hydrodynamics flow. Inference about all of these parameters is highly desirable. The initial state of heavy ion collisions is widely believed to currently be the largest source of uncertainty in most calculations of bulk evolution. The shear viscosity to entropy ratio η/s and its temperature dependence α are extremely interesting as these are fundamental properties of the strongly-interacting QGP.

A novel feature of this analysis is that the initial state is described with prescriptions for the initial energy density and flow profiles that can be adjusted parametrically. The initial energy density $\epsilon(x, y)$ is constructed as a balance between wounded nucleon (Glauber) and saturation (CGC) based profiles, this balance is controlled by f_{wn}. The parameter σ_{sat} controls the cross-section scale for changing the behavior of the saturation model from the binary collision limit where $\epsilon \sim T_A T_B$ (where T_X is the nuclear thickness function, essentially the density of the nucleus projected into the plane transverse to the beam axis) to the saturated limit when ϵT_{min}. The change occurs for $T_{max} \approx 1/\sigma_{\text{sat}}$.

The initial transverse flow profile is approximated as being proportional to T_{0i}/T_{00} where $T_{\mu\nu}$ is the stress energy tensor (see § 2.1), the extent of this propor-

Table 9.3: Summary of model parameters. Six model parameters were varied. The first four describe the initial state being fed into the hydrodynamic module, and the last two describe the viscosity and its energy dependence.

parameter	description	range
$(dE/dy)_{pp}$	The initial energy per rapidity in the diffuse limit compared to measured value in pp collision	0.85–1.2
$\sigma_{\rm sat}$	This controls how saturation sets in as function of areal density of the target or projectile. In the wounded nucleon model it is assumed to be the free nucleon-nucleon cross section of 42 mb	30 mb–50 mb
f_{wn}	Determines the relative weight of the wounded-nucleon and saturation formulas for the initial energy density	0–1
$F_{\rm flow}$	Describes the strength of the initial flow as a fraction of T_{0i}/T_{00}	0.25–1.25
$\eta/s\vert_{T_c}$	Viscosity to entropy ratio for T = 170 MeV	0 – 0.5
α	Temperature dependence of η/s for temperatures above 170 MeV$/c$, i.e., $\eta/s = \eta/s\vert_{T_c} + \alpha\ln(T/T_c)$	0 - 5

tionality is set by the parameter $F_{\rm flow}$. The shear-viscosity η arises as a constant in the Israel-Stewart gradient expansion of the hydrodynamical equations of motion, a temperature dependence for the shear viscosity to entropy density ratio in the QGP phase η/s was taken as

$$\frac{\eta}{s} = \frac{\eta}{s}\Big|_{T_c} + \alpha \ln\left(\frac{T}{T_c}\right).$$

A wide range of observables were initially collected and considered for use in a calibration procedure with runs being made in two centrality bins $0 - 5\%$ and $20 - 30\%$. A 729 point LHS design was used for each centrality class. The outputs selected (see Table: 9.4) include average particle multiplicities and transverse momenta, the average elliptic flow and two-particle correlations in the form of Hanbury-Brown-Twiss (HBT) source radii [214, 215]. These observations were reducing using principle components and then a scheme roughly similar to that outlined in § 9.2 was used to obtain posterior distributions for the calibration parameters, with initially flat priors $\pi(u_\star)\propto\Theta(1 - u_\star)$.

The marginal and joint posterior distributions for the calibration parameters are

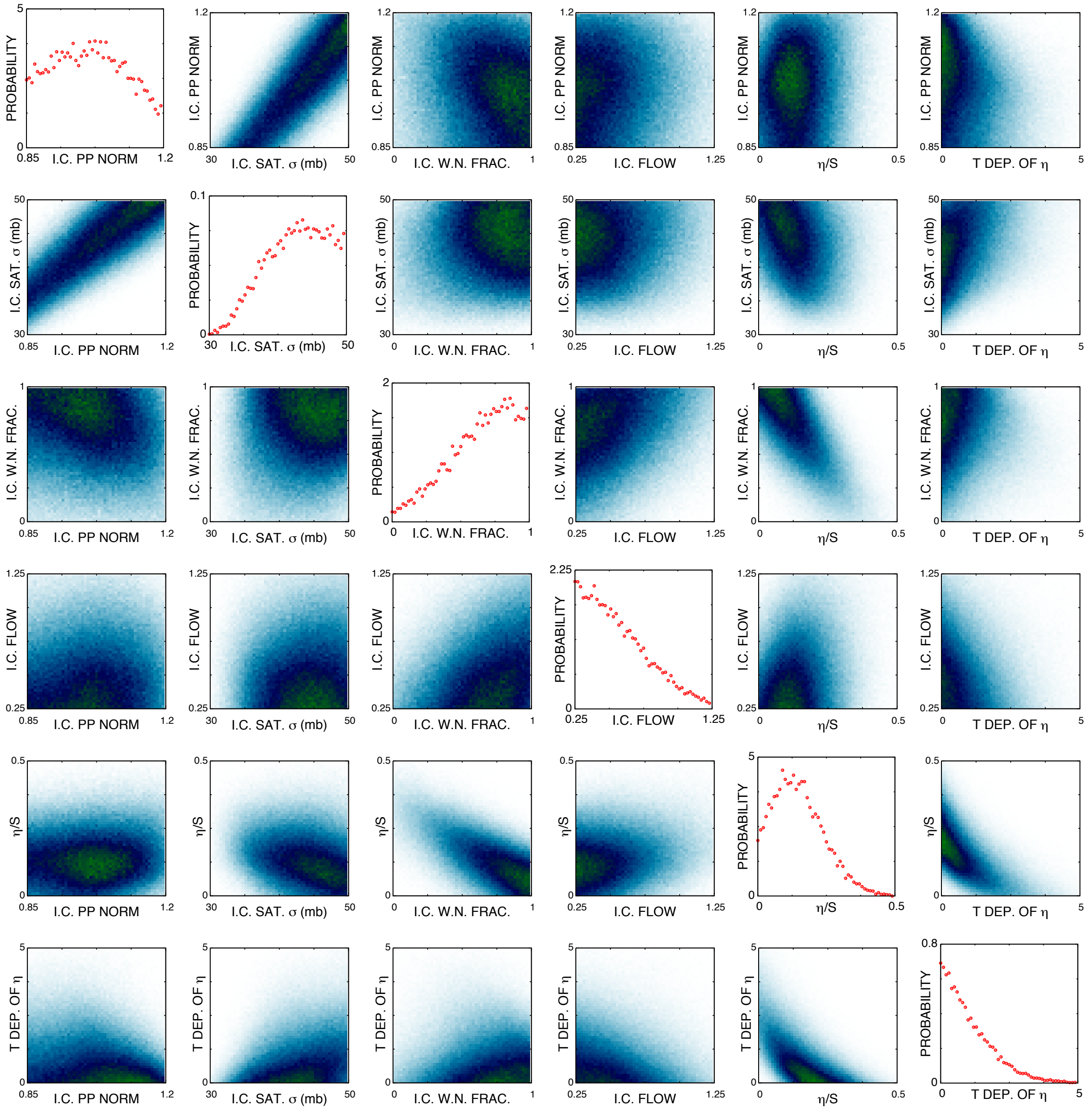


FIGURE 9.4: The marginal posterior distributions of the six calibration parameters are shown along the diagonal. The off-diagonal plots display the joint distributions of the calibration parameters. Four of the six parameters refer to the initial state (hence the "I.C." in their name) and the last two describe the shear viscosity.

Table 9.4: Observables used to compare models to data. *To account for non-flow correlations, the value of v_2 was reduced by 10% from the value reported in [216].

observable	p_t weighting	centrality	ref	err
$v_{2,\pi^+\pi^-}$	ave. over 11 p_t bins from 160 MeV/c to 1 GeV/c	20-30%	[216]*	12%
R_{out}	ave. over 4 p_t bins from 150-500 MeV/c	0-5%	[217]	6%
R_{side}	ave. over 4 p_t bins from 150-500 MeV/c	0-5%	[217]	6%
R_{long}	ave. over 4 p_t bins from 150-500 MeV/c	0-5%	[217]	6%
R_{out}	ave. over 4 p_t bins from 150-500 MeV/c	20-30%	[217]	6%
R_{side}	ave. over 4 p_t bins from 150-500 MeV/c	20-30%	[217]	6%
R_{long}	ave. over 4 p_t bins from 150-500 MeV/c	20-30%	[217]	6%
$\langle p_t \rangle_{\pi^+\pi^-}$	200 MeV/$c < p_t <$ 1.0 GeV/c	0-5%	[218]	3%
$\langle p_t \rangle_{K^+K^-}$	400 MeV/$c < p_t <$ 1.3 GeV/c	0-5%	[218]	3%
$\langle p_t \rangle_{p\bar{p}}$	600 MeV/$c < p_t <$ 1.6 GeV/c	0-5%	[218]	3%
$\langle p_t \rangle_{\pi^+\pi^-}$	200 MeV/$c < p_t <$ 1.0 GeV/c	20-30%	[218]	3%
$\langle p_t \rangle_{K^+K^-}$	400 MeV/$c < p_t <$ 1.3 GeV/c	20-30%	[218]	3%
$\langle p_t \rangle_{p\bar{p}}$	600 MeV/$c < p_t <$ 1.6 GeV/c	20-30%	[218]	3%
$\pi^+\pi^-$ yield	200 MeV/$c < p_t <$ 1.0 GeV/c	0-5%	[218]	6%
$\pi^+\pi^-$ yield	200 MeV/$c < p_t <$ 1.0 GeV/c	20-30%	[218]	6%

shown in Fig: 9.4. Although over 90% of the six-dimensional parameter space is eliminated at the one-sigma level, the individual parameters are rarely constrained to less than half their initial range when other parameters are allowed to vary.

The first four parameters ("I.C. PP NORM", "I.C. SAT σ", "I.C. W.N. FRAC" and "I.C. FLOW") determine the initial state fed into the hydro. The first parameter "I.C. PP NORM" sets the constant of proportionality between the product of the areal densities of the incoming nuclei, and the initial energy density fed into the hydro. In the limit of low aerial densities this should be consistent with pp collisions. Thus, the range of the prior distribution was quite small, and the statistical analysis did little to further constrain it. The parameter "I.C. SAT σ" refers to σ_{sat} and parameterizes the saturation of the energy density with multiple collisions.

The preferred value appears rather close to the value of 42 mb typically used in the wounded nucleon model, though there is a fairly wide range of accepted values. The parameter "I.C. W.N. FRAC" sets the weights between the wounded nucleon and the saturation parameterizations. This shows a preference for the wounded nucleon prescription which gives a smaller initial anisotropy than the saturation parameterization. The final initial-condition parameterization, "I.C. FLOW" sets the fraction of initial transverse flow in the hydrodynamic calculation. The posterior points to a rather small fraction of this flow, though like all of the initial-condition parameters has a fairly broad range of possible values.

The last two parameters refer to the viscosity. The viscosity at $T = 170$ MeV is referred to as "η/s" in Fig: 9.4, and the temperature dependence is labelled by "T DEP. of η". Both are significantly constrained as a fraction of the original parameter space. The range of η/s is consistent with similar, but less complete, searches through parameter space using similar models [33, 61]. In [219], the authors found little sensitivity to the viscosity at higher temperatures, but considered a smaller variation of the viscosity with temperature than was considered here.

Figure 9.4 also shows the pairwise joint posteriors of the calibration parameters . Several parameters are strongly correlated. For instance, the energy normalization "I.C. PP NORM" and "I.C. SAT σ" are strongly correlated in that one can have less saturation of the cross section if the energy normalization is turned down. There is also a strong correlation between "I.C. FLOW" and "I.C. W.N. FRAC". One can compensate for less initial flow if the saturation prescription is more heavily used than the wounded nucleon. Again, this is expected because the wounded nucleon parameterization leads to less spatial anisotropy and a somewhat more diffuse initial state.

The inferred viscosity is clearly correlated with the weighting between the wounded nucleon and saturation parameterizations, as expected from the arguments in [60]. The two viscous parameters are also correlated with one another as expected. One can compensate for a very low viscosity at $T = 170$ MeV by having the viscosity rise quickly with temperature. Higher values of the temperature dependence α are increasingly unlikely for higher values of $\eta/s|_{T_c}$.

The procedures applied here represent a significant improvement to the state-of-the-art for comparisons of data and models in the field of relativistic heavy ion physics. Previously, parameters were varied either individually, or in small groups.

10

Afterword

I was the smudge of ashen fluff–and I
Lived on, flew on, in the reflected sky

Computer models are an essential tool in the study of complicated physical systems, in fact they often seem to be unavoidable. In this thesis I have introduced the concept of the computer experiment, the systematic analysis of a computer model and it's inputs and outputs as a means for not just understanding the model but also the potential rolê it can play in making strong statements about observable and un-observable physical quantities.

While these ideas are not entirely cutting edge in and of themselves, their practical application/adoption is presently rather confined to experts or at least to those projects which can afford to devote a graduate student to become an approximate expert. The results and experiences that I have collected here should hopefully serve to ameliorate this situation. As illustrated in chapter 2 there are a great many ripe opportunities for the careful application of computer experiments in the field of Heavy-Ion physics, where almost all the quantities of interest are not directly observable. The techniques themselves are general and can and should be applied widely in the physical sciences.

The direct calibration of Heavy Ion bulk evolution simulators will lead to precision estimates of the shear viscosity and other transport coefficients, has already begun in [1, 33] (§ 9.4). Furthermore with a sufficiently well calibrated model the loop between experimental and computational data might be closed enough to allow direct inference about the plausible initial conditions which lead to a given set of experimental observables [220]. There is great potential for studying how experimental data and a calibrated model constrain "un-observable observables" such as the simulated initial energy and flow distributions of the system in a given model, this is a very intriguing prospect.

A similarly bright future exists in the application of these techniques to the study of jet quenching in heavy ion physics, values of (or really distributions for) $\hat{q}$ and $\hat{e}$ can be extracted from the various advanced transport models and these should be subject to careful scrutiny and happily these efforts are already underway although perhaps in a slightly ad-hoc fashion [157, 221, 158, 131, 155]. Finally it might be very interesting to turn this calibration process on its head and undertake a top down (primarily experimentally driven) approach to understanding jet quenching. Alongside the more traditional focus on accepting or rejecting a particular microscopic theoretical model of in-medium jet transport described above I suggest a new approach based on attempting to estimate the scale and nature of the family of general jet modification kernels that are compatible with a given set of observations. Taking advantage of the relatively low theoretical uncertainty in the treatment of the bulk evolution of the QGP, even lower once one is using a really well calibrated model, along with the well understood vacuum jet production process it may be possible to invert the experimental data to give a range of acceptable quenching forms, using a Gaussian Process as a prior on our family of modification kernels.

Appendix A

Some Useful Results

A.1 Block Matrix Inverse

If A, C and $C^{-1} + DA^{-1}B$ are nonsingular square matrices then the following is true

$$(A + BCD)^{-1} = A^{-1} + A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}. \tag{A.1}$$

Further if we write

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X & Y \\ Z & U \end{pmatrix} = \begin{pmatrix} I_m & 0 \\ 0 & I_n \end{pmatrix}. \tag{A.2}$$

Then by the definition of the matrix inverse

$$\begin{pmatrix} X & Y \\ Z & U \end{pmatrix} = \frac{1}{(AD - BC)} \begin{pmatrix} D & -B \\ -C & A \end{pmatrix}. \tag{A.3}$$

Which we can re-arrange to obtain some common factors

$$\begin{pmatrix} X & Y \\ Z & U \end{pmatrix} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}. \tag{A.4}$$

If we multiply the X and A matrices the other way around we find that

$$\left(\begin{array}{cc} X & Y \\ Z & U \end{array} \right) = \left[\begin{array}{cc} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{array} \right]. \tag{A.5}$$

Which are equivalent. Now we can use these results to find the inverse of an $(m + 1) \times (m + 1)$ matrix M in block form

$$M = \left(\begin{array}{cc} A & B \\ B^T & C \end{array} \right). \tag{A.6}$$

After a little algebra we obtain

$$M^{-1} = \left(\begin{array}{cc} A^{-1} + \frac{1}{k}A^{-1}BB^TA^{-1} & -\frac{1}{k}A^{-1}B \\ -\frac{1}{k}B^TA^{-1} & \frac{1}{k} \end{array} \right) \tag{A.7}$$

Where $k = C - B^TA^{-1}B$. This is the Sherman-Morrison-Woodbury inversion formula, it's fun to think of A, B, C as representing each of the authors.

A.2 Gaussian Identities

The probability density for a p dimensional multivariate normal variable, with mean vector μ and covariance matrix Σ is

$$f(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^p|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^{\intercal}\Sigma^{-1}(x-\mu) \right) \tag{A.8}$$

Let x and y be jointly distributed Gaussian random vectors, their joint distribution is

$$\left(\begin{array}{c} x \\ y \end{array} \right) \sim \mathrm{MVN}\left\{ \left(\begin{array}{c} \mu_x \\ \mu_y \end{array} \right), \left(\begin{array}{cc} A & C \\ C^{\intercal} & A \end{array} \right) \right\}. \tag{A.9}$$

Where the block matrix C can be thought of as setting the degree of statistical dependence, or lack of independence between the two vectors. The conditional distribution of x given a particular value of $y = \tilde{y}$ is

$$f(x \mid y = \tilde{y}) \sim \mathrm{MVN}\left(\bar{\mu}, \bar{\Sigma}\right), \tag{A.10}$$

where the conditional mean depends on the value taken by y

$$\bar{\mu} = \mu_x + CB^{-1}(\tilde{y} - mu_y) \tag{A.11}$$

and the conditional covariance is independent of y

$$\bar{\Sigma} = A - CB^{-1}C^{\intercal}. \tag{A.12}$$

To see why this is so let us first note the following fact, if $X \sim \mathrm{MVN}(\mu, \Sigma)$ then any linear combination $a'X = a_1X_1 + \ldots$ has distribution: $a'X \sim \mathrm{MVN}(a'\mu, a'\Sigma a)$, [183]. Let us define the block matrix

$$\Gamma = \begin{pmatrix} I & -CB^{-1} \\ 0 & I \end{pmatrix},$$

if we label the joint distribution of x and y given in (A.9) as X then

$$\begin{aligned} \Gamma(X - \mu) &= \begin{pmatrix} x - \mu_x \\ y - \mu_y \end{pmatrix}, \\ &= \begin{pmatrix} x - \mu_x - (y - \mu_y)CB^{-1} \\ y - \mu_y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}, \end{aligned} \tag{A.13}$$

is jointly normal with covariance matrix $\Gamma\Sigma\Gamma^{\intercal}$

$$\Gamma\Sigma\Gamma^{\intercal} = \begin{pmatrix} A - CB^{-1}C^{\intercal} & 0 \\ 0 & B \end{pmatrix}. \tag{A.14}$$

Note that we can immediately conclude that under this transform x' is independent of y'. Again if we are given the value $y = \tilde{y}$ then

$$x' \sim \mathrm{MVN}\left(0, A - CB^{-1}C^{\intercal}\right) \tag{A.15}$$

and $\mu_x + CB^{-1}(\tilde{y} - \mu_y)$ is a constant. By the independence of x' and y', the conditional distribution of x' given $y = \tilde{y}$ is the same as its unconditional distribution, then we can write

$$x|y \sim \mathrm{MVN}\left(\mu_x + CB^{-1}(\tilde{y} - \mu_y), A - CB^{-1}C^{\intercal}\right) \tag{A.16}$$

which is the desired result.

The product of two Gaussians is another Gaussian

$$\begin{gathered} \mathrm{N}(x \mid a, A)\mathrm{N}(x \mid b, B) = Z^{-1}\, \mathrm{N}(x \mid c, C), \\ c = C(A^{-1}a + B^{-1}b), \quad C = (A^{-1} + B^{-1}), \end{gathered} \tag{A.17}$$

where the normalization constant is itself another Gaussian

$$Z^{-1} = (2\pi)^{p/2}|A + B|^{-1/2} \exp\left(-\frac{1}{2}(a - b)^\intercal (A + B)^{-1}(a - b)\right). \tag{A.18}$$

A.3 Some Probability Things

For lots of fascinating reading about probability, Bayes and otherwise consult [222, 223, 170, 169, 224]. The particle data group reviews on statistics and probability give an experimental physics perspective on some of these issues [225].

A.3.1 Bayes Theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{A.19}$$

A.3.2 Poisson distribution

$$f(x; \lambda) = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{A.20}$$

$$F(x; \lambda) = \Pr(X \leqslant k) = e^{-\lambda} \sum_{i=0}^{\lfloor k \rfloor} \frac{\lambda^i}{i!} \tag{A.21}$$

The mean and variance $\mathrm{E}[X] = \mathrm{V}[X] = \lambda$ are both equal to the rate.

A.3.3 Student-t distribution

The Student-t or just t distribution with ν arises from considering the distribution of the sample mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} X_i$ of a set of $n = \nu + 1$ samples from some population, i.e. the distribution of sample means that would be obtained after making a large number of repeated observations from the same population. If we introduce the sample variance $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{\mu}\right)^2$ then the centralized and standardized quantity

$$\frac{\bar{x} - \mu}{s/\sqrt{n}} \sim t_{\nu=n-1} \tag{A.22}$$

is t distributed with $n-1$ degrees of freedom, where μ is the population mean.

$$f(x;\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}}\Gamma\left(\frac{\nu}{2}\right)\left(1 + \frac{x^2}{\nu}\right)^{-\nu+\frac{1}{2}} \tag{A.23}$$

$$F(x;\nu) = \frac{1}{2} + x\Gamma\left(\frac{\nu+1}{2}\right){}_2F_1\left(\frac{1}{2}, \frac{\nu+1}{2}; \frac{3}{2}; -\frac{x^2}{\nu}\right)\frac{1}{\sqrt{\pi\nu}\Gamma\left(\frac{\nu}{2}\right)} \tag{A.24}$$

The mean is zero and the variance $\mathrm{V}[X] = \frac{\nu}{\nu-2}$, examining the density it's clear that in the limit $\nu \to \infty$ the distribution will become normal using the well known result $\lim_{n\to\infty}\left(1 + \dfrac{x}{n}\right)^n = \exp(x)$.

The noncentral Student-t distribution is a generaliztion, if $Z \sim \mathrm{N}(0,1)$ and $V \sim \chi^2_k$ and V and Z are statistically independent then the variable T,

$$T = \frac{Z + \mu}{\sqrt{V/k}} \sim t_{k,\mu} \tag{A.25}$$

has a noncentral t distribution with k degrees of freedom and noncentrality parameter μ. The CDF is

$$F_{k,\mu}(x) = \begin{cases} \frac{1}{2}\sum_{j=0}^{\infty}\frac{1}{j!}(-\mu\sqrt{2})^j e^{\frac{-\mu^2}{2}}\frac{\Gamma(\frac{j+1}{2})}{\Gamma(1/2)}I\left(\frac{k}{k+x^2}; \frac{k}{2}, \frac{j+1}{2}\right), & x \geqslant 0 \\ 1 - \frac{1}{2}\sum_{j=0}^{\infty}\frac{1}{j!}(-\mu\sqrt{2})^j e^{\frac{-\mu^2}{2}}\frac{\Gamma(\frac{j+1}{2})}{\Gamma(1/2)}I\left(\frac{k}{k+x^2}; \frac{k}{2}, \frac{j+1}{2}\right), & x < 0 \end{cases} \tag{A.26}$$

where $I(x,a,b) = \frac{B(x;a,b)}{B(a,b)}$ is the regularized incomplete beta function, where

$$B(x,a,b) = \int_0^x t^{a-1}(1-t)^{b-1}\,dt, \quad B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}.$$

The density is

$$f(x) = \begin{cases} \frac{k}{x}\left\{F_{k+2,\mu}\left(x\sqrt{1+\frac{2}{k}}\right) - F_{k,\mu}(x)\right\}, & \text{if } x \neq 0, \\ \frac{\Gamma(\frac{k+1}{2})}{\sqrt{\pi k}\Gamma(\frac{k}{2})}\exp\left(-\frac{\mu^2}{2}\right), & \text{if } x = 0. \end{cases} \tag{A.27}$$

The mean and variance exist as long as k is large enough,

$$\mathrm{E}\left[T\right] = \mu\sqrt{\frac{k}{2}}\frac{\Gamma((k-1)/2)}{\Gamma(k/2)}, \qquad \text{if } k > 1, \tag{A.28}$$

$$\mathrm{Var}\left[T\right] = \frac{k(1+\mu^2)}{k-2} - \frac{\mu^2 k}{2}\left(\frac{\Gamma((k-1)/2)}{\Gamma(k/2)}\right)^2, \qquad \text{if } k > 2. \tag{A.29}$$

A.3.4 Chi-Squared distribution

If $X_1, \dots X_k$ are independent standard normal random variables then the sum of their squares is chi-squared distributed with k degrees of freedom

$$Q = \sum_{i=1}^{k} X_i^2 \implies Q \sim \chi_k^2, \tag{A.30}$$

the density and distribution functions are

$$f(x;k) = \frac{1}{2^{k/2}\Gamma\left(\frac{k}{2}\right)} x^{\frac{k}{2}-1} e^{-\frac{x}{2}}, \quad x \geqslant 0 \tag{A.31}$$

$$F(x;k) = \frac{1}{\Gamma\left(\frac{k}{2}\right)}\gamma\left(\frac{k}{2}, \frac{x}{2}\right), \tag{A.32}$$

$$\gamma(x,s) = \int_0^x t^{s-1}e^{-t}\,dt.$$

where $\gamma(x,s)$ is known as the *lower incomplete gamma function*. The mean of a χ_k^2 distribution is k and the variance is $2k$. It may be useful to note that the sum of independent chi-squared variables is also chi-squared, i.e. if $X_i \sim \chi_{k_i}^2$ then if we define $Z = \sum_{i=1}^{n} X_i$, $Z \sim \chi^2_{\sum_{i=1}^{n} k_i}$. Asymptotically in the number of degrees of freedom k, a standardized χ_k^2 variable converges in distribution to a standard normal, i.e. as $k \to \infty$ $\frac{\left(\chi_k^2 - k\right)}{\sqrt{2k}} \xrightarrow{d} N(0,1)$. If Z is a n dimensional Gaussian random vector with mean μ and rank k covariance matrix C then the sum of squared distances

$$X = (Z - \mu)^{\intercal} C^{-1} (Z - \mu), \quad X \sim \chi_k^2,$$

this result is clear if we imagine diagonalizing C first. If $X_1 \sim \chi_{k_1}^2$ and $X_2 \sim \chi_{k_2}^2$ then the ratio $Y = \frac{k_2 X_1}{k_1 X_2}$ is F distributed, $Y \sim F(k_1, k_2)$. The F distribution comes up fairly often in the context of linear modelling, the details of the distribution are fairly tedious and best found by consulting a standard references [174, 226].

A.4 Assessing Normality

A.4.1 Univariate Data

Given a set of d samples $\mathcal{Y} = \{y_1, \ldots, y_d\}$ where we believe that the samples are roughly normally distributed, we can compute the sample mean $\bar{\mu} = \frac{1}{d}\sum_{i=1}^{d} y_i$ and sample standard variance $s^2 = \frac{1}{d}\sum_{i=1}^{d}(y_i - \bar{\mu})^2$ in the usual way and then we want to assess if

$$\frac{\mathcal{Y} - \bar{\mu}}{s} \sim N(0,1).$$

A nice visual way to do this is by making a so called quantile-quantile (QQ) plot, here one plots a set of empirical quantiles generated from the sample data set against theoretical quantiles from the distribution of interst. Essentially one is plotting a set of points from the CDF of the sample against the same set of points from the CDF of the test distribution. In this case if the samples are well described by a normal distribution the graph should be a relatively straight line, the major advantage of a QQ plot is that it allows one to rapidly assess the location of any deviations.

A contrived example using a QQ plot for some diagnostics is shown in Fig: A.1. A set of 256 samples were drawn from a χ^2 distribution, in the left panel empirical quantiles for these samples are plotted against those of a standard normal distribution, the red-line shows what one would expect if the samples were normally distributed. Although the central part of the sample QQ curve looks roughly linear its clear that there are serious deviations at both tails of the sample distribution, confronted with this sort of plot one would not be convinvced of the normality of ones samples. In the right panel I have plotted the sample quantiles against theoretical quantiles from a χ^2 distribution with 4 degrees of freedom. The agreement here is far better at the left tail although there is still some deviation at the right tail. This remaining deviation is a result of drawing a finite number of samples from a distribution with long tail.

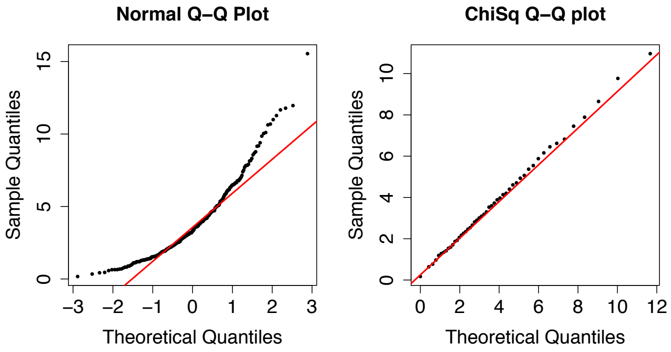


Figure A.1: Two QQ plots for a set of 256 samples drawn from a χ^2_4 distribution. In the left panel the sample quantiles are plotted against theoretical quantiles from a standard normal distribution. In the right panel the sample quantiles are plotted against theoretical quantiles from the *population distribution* namely χ^2_4. In both panels the red curve shows the expected result if the sample and theoretical distributions were identical.

A.4.2 Multivariate Data

Given a set of d sample vectors $\mathcal{Y} = \{y_1, \dots, y_d\}$ where each sample is a k-length vector $y_1^\intercal = (y_1^1, \dots, y_1^k)^\intercal$ we can construct the sample mean (a k vector) in the usual way

$$\hat{\mu}^\alpha = \frac{1}{d}\sum_{i=1}^{d} y_i^\alpha,$$

we can also construct the elements of the sample covariance matrix $\hat{\Sigma}$ $(k \times k)$

$$\hat{\Sigma}^{\alpha\beta} = \frac{1}{d}\sum_{i=1}^{d}\left(y_i^{\alpha} - \hat{\mu}^{\alpha}\right)\left(y_i^{\beta} - \hat{\mu}^{\beta}\right).$$

We can then introduce the set of squared distances d_j^2

$$d_j^2 = \left(y_j - \hat{\mu}\right)\hat{\Sigma}^{-1}\left(y_j - \hat{\mu}\right)^{\intercal}. \tag{A.33}$$

If the data $\mathcal{Y}$ is multivariate normally distributed then we would expect that these distances to have a χ^2 distribution with degrees of freedom given by the rank $r \leqslant k$ of $\hat{\Sigma}$ [183]

$$d_j^2 \sim \chi_r^2. \tag{A.34}$$

Now we have obtained a set of quantities d_j^2 which can be easily visually examined with QQ plots, but here the theoretical quantiles we're plotting our data against are those of a χ_r^2 distribution.

Bibliography

[1] J. Novak, K. Novak, S. Pratt, C. E. Coleman-Smith, and R. Wolpert, (2013), arXiv:1303.5769.

[2] H. Petersen, C. E. Coleman-Smith, S. A. Bass, and R. Wolpert, J.Phys. **G38**, 045102 (2011), arXiv:1012.4629.

[3] F. A. Gómez, C. E. Coleman-Smith, B. W. O'Shea, J. Tumlinson, and R. L. Wolpert, (2013), arXiv:1311.2587.

[4] F. A. Gómez, C. E. Coleman-Smith, B. W. O'Shea, J. Tumlinson, and R. Wolpert, Astrophys.J. **760**, 112 (2012), arXiv:1209.2142.

[5] M. C. Kennedy and A. O'Hagan, Journal of the Royal Statistical Society **B63**, 425 (2001).

[6] E. Fermi, J. Pasta, and S. Ulam, "studies of non-linear problems", in *"The Collected papers of Enrico Fermi"* Vol. 2, pp. 492–502, The University of Chicago Press, 1965.

[7] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, Stat. Sci. **4**, 409 (1989).

[8] C. Currin, T. J. Mitchell, M. D. Morris, and D. Ylvisaker, Journal of the American Statistical Association **86**, 953 (1991).

[9] D. G. Krige., Journal of the Chemical, Metallurgical and Mining Society of South Africa **52**, 119 (1951).

[10] N. A. C. Cressie, *Statistics for Spatial Data* (John Wiley & Sons, New York, NY, 1993).

[11] M. Stein, *Interpolation of Spatial Data: Some Theory for Kriging* (Springer, 1999).

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* (The MIT Press, 2005).

[13] M. C. Kennedy and A. O'Hagan, Biometrika , 1 (2000).

[14] M. J. Bayarri *et al.*, *A framework for validation of computer models*, 2002.

[15] D. Higdon, M. Kennedy, J. Cavendish, J. Cafeo, and R. Ryne, SIAM Journal on Scientific Computing **26**, 448 (2004).

[16] D. Higdon, J. Gattiker, B. Williams, and M. Rightley, Journal of the American Statistical Association **103**, 570 (2008).

[17] M. J. Bayarri *et al.*, The Annals of Statistics **35**, 1874 (2007).

[18] T. J. Santner, B. J. Williams, and W. Notz, *The Design and Analysis of Computer Experiments* (Springer Verlag, New York, NY, 2003).

[19] M. D. McKay, R. J. Beckman, and W. J. Conover, Technometrics **21**, pp. 239 (1979).

[20] Kai-Tai Fang, Runze Li, and Agus Sudjianto, *Design and Modeling for Computer Experiments* (Chapman and Hall/CRC 2005, 2005).

[21] J. L. Loeppky, J. Sacks, and W. J. Welch, Technometrics **51**, 366 (2009).

[22] A. O'Hagan, Reliability Engineering & System Safety **91**, 1290 (2006), The Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004) - SAMO 2004.

[23] J. Oakley and A. O'Hagan, Biometrika **89**, 769 (2002).

[24] D. Xiu, *Numerical methods for stochastic computations : a spectral method approach* (Princeton University Press, Princeton, NJ, 2010).

[25] J. E. Oakley and A. O'Hagan, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **66** (2004).

[26] E. D. Smith, F. Szidarovszky, W. J. Karnavas, and A. T. Bahill, The Open Cybernetics & Systemics Journal **2**, 39 (2008).

[27] A. Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, Stefano Tarantola, *Global Sensitivity Analysis: The Primer* (Wiley, 2008).

[28] M. Schonlau and W. J. Welch, *Screening the Input Variables to a Computer Code Via Analysis of Variance and Visualization* (Springer New York, New York, 2006), chap. Screening the Input Variables to a Computer Code Via Analysis of Variance and Visualization.

[29] S. Habib, K. Heitmann, D. Higdon, C. Nakhleh, and B. Williams, Phys.Rev. **D76**, 083503 (2007), arXiv:astro-ph/0702348.

[30] R. G. Bower *et al.*, MNRAS**407**, 2017 (2010), arXiv:1004.0711.

[31] I. Vernon, M. Goldstein, and R. G. Bower, Bayesian Analysis **5**, 619 (2010), (with discussion).

[32] M. J. Bayarri *et al.*, Technometrics **51**, 402 (2009).

[33] R. Soltz *et al.*, Phys.Rev. **C87**, 044901 (2013), arXiv:1208.0897.

[34] S. Bass *et al.*, Prog.Part.Nucl.Phys. **41**, 255 (1998), arXiv:nucl-th/9803035.

[35] M. Bleicher *et al.*, J.Phys.G **G25**, 1859 (1999), arXiv:hep-ph/9909407.

[36] H. Petersen, J. Steinheimer, G. Burau, M. Bleicher, and H. Stocker, Phys.Rev. **C78**, 044901 (2008), arXiv:0806.1695.

[37] STAR Collaboration, J. Adams *et al.*, Phys.Rev.Lett. **92**, 112301 (2004), arXiv:nucl-ex/0310004.

[38] *Lectures on quark matter.*, 2002, Prepared for 40th Internationale Universitatswochen fuer Theoretische Physik: Dense Matter (International University School of Theoretical Physics) (IUKT 40), Schladming, Styria, Austria, 3-10 Mar 2001.

[39] J. Letessier and J. Rafelski, Camb.Monogr.Part.Phys.Nucl.Phys.Cosmol. **18**, 1 (2002).

[40] K. Yagi, T. Hatsuda, and Y. Miake, Camb.Monogr.Part.Phys.Nucl.Phys.Cosmol. **23**, 1 (2005).

[41] B. V. Jacak and B. Muller, Science **337**, 310 (2012).

[42] B. Muller, (2012), arXiv:1207.7302.

[43] Z. Qiu, *Event-by-event Hydrodynamic Simulations for Relativistic Heavy-ion Collisions*, PhD thesis, The Ohio State University, 2013, arXiv:1308.2182.

[44] H. Song, S. A. Bass, U. Heinz, T. Hirano, and C. Shen, Phys.Rev.Lett. **106**, 192301 (2011), arXiv:1011.2783.

[45] C. Nonaka and S. A. Bass, Phys.Rev. **C75**, 014902 (2007), arXiv:nucl-th/0607018.

[46] P. Huovinen, P. Kolb, U. W. Heinz, P. Ruuskanen, and S. Voloshin, Phys.Lett. **B503**, 58 (2001).

[47] B. Schenke, S. Jeon, and C. Gale, Phys.Rev. **C82**, 014903 (2010), arXiv:1004.1408.

[48] P. Huovinen, Nucl.Phys. **A761**, 296 (2005).

[49] T. Hirano and K. Tsuda, Phys.Rev. **C66**, 054905 (2002).

[50] J. Bjorken, Phys.Rev. **D27**, 140 (1983).

[51] W. Israel, Ann. Phys. **100**, 310 (1976).

[52] J. M. Stewart, Proceedings of the Royal Society **A 357**, 59 (1977).

[53] W. Israel and J. M. Stewart, Ann. Phys. **188**, 341 (1979).

[54] H. Song and U. W. Heinz, Phys.Lett. **B658**, 279 (2008), arXiv:0709.0742.

[55] H. Song and U. W. Heinz, Phys.Rev. **C78**, 024902 (2008), arXiv:0805.1756.

[56] M. Luzum and P. Romatschke, Phys.Rev. **C78**, 034915 (2008), arXiv:0804.4015.

[57] P. Romatschke and U. Romatschke, Phys.Rev.Lett. **99**, 172301 (2007), arXiv:0706.1522.

[58] H. Petersen, R. La Placa, and S. A. Bass, J.Phys.G **G39**, 055102 (2012), arXiv:1201.1881.

[59] G.-Y. Qin, H. Petersen, S. A. Bass, and B. Muller, Phys.Rev. **C82**, 064903 (2010), arXiv:1009.1847.

[60] H.-J. Drescher, A. Dumitru, C. Gombeaud, and J.-Y. Ollitrault, Phys.Rev. **C76**, 024905 (2007), arXiv:0704.3553.

[61] U. Heinz, C. Shen, and H. Song, AIP Conf.Proc. **1441**, 766 (2012), arXiv:1108.5323.

[62] C. Shen *et al.*, J.Phys. **G38**, 124045 (2011), arXiv:1106.6350.

[63] C. Shen, U. Heinz, P. Huovinen, and H. Song, Phys.Rev. **C84**, 044903 (2011), arXiv:1105.3226.

[64] H. Song, S. A. Bass, and U. Heinz, Phys.Rev. **C83**, 054912 (2011), arXiv:1103.2380.

[65] H. Song, S. A. Bass, and U. Heinz, Phys.Rev. **C83**, 024912 (2011), arXiv:1012.0555.

[66] C. Shen, U. Heinz, P. Huovinen, and H. Song, Phys.Rev. **C82**, 054904 (2010), arXiv:1010.1856.

[67] H. Song and U. W. Heinz, Phys.Rev. **C81**, 024905 (2010), arXiv:0909.1549.

[68] Z. Qiu, C. Shen, and U. Heinz, Phys.Lett. **B707**, 151 (2012), arXiv:1110.3033.

[69] B. Schenke, S. Jeon, and C. Gale, Phys.Rev.Lett. **106**, 042301 (2011).

[70] B. H. Alver, C. Gombeaud, M. Luzum, and J.-Y. Ollitrault, Phys.Rev. **C82**, 034913 (2010).

[71] D. Teaney and L. Yan, Phys.Rev. **C83**, 064904 (2011).

[72] B. Schenke, P. Tribedy, and R. Venugopalan, (2012), arXiv:1202.6646.

[73] A. Dumitru and Y. Nara, Phys.Rev. **C85**, 034907 (2012).

[74] M. Alvioli, H. Holopainen, K. Eskola, and M. Strikman, Phys.Rev. **C85**, 034902 (2012).

[75] Z. Qiu and U. W. Heinz, Phys.Rev. **C84**, 024911 (2011), arXiv:1104.0650.

[76] P. Staig and E. Shuryak, Phys.Rev. **C84**, 044912 (2011), arXiv:1105.0676.

[77] M. Bleicher *et al.*, Nucl.Phys. **A638**, 391 (1998).

[78] F. Grassi, Y. Hama, O. Socolowski, and T. Kodama, J.Phys.G **G31**, S1041 (2005).

[79] B. Tavares, H.-J. Drescher, and T. Kodama, Braz.J.Phys. **37**, 41 (2007).

[80] R. Andrade, F. Grassi, Y. Hama, T. Kodama, and J. Socolowski, O., Phys.Rev.Lett. **97**, 202302 (2006).

[81] R. Andrade, F. Grassi, Y. Hama, T. Kodama, and W. Qian, Phys.Rev.Lett. **101**, 112301 (2008).

[82] R. P. G. Andrade, F. Grassi, Y. Hama, and W.-L. Qian, Phys.Lett. **B712**, 226 (2012), arXiv:1008.4612.

[83] H. Holopainen, H. Niemi, and K. J. Eskola, Phys.Rev. **C83**, 034901 (2011).

[84] K. Werner, I. Karpenko, T. Pierog, M. Bleicher, and K. Mikhailov, Phys.Rev. **C82**, 044904 (2010).

[85] H. Petersen, G.-Y. Qin, S. A. Bass, and B. Muller, Phys.Rev. **C82**, 041901 (2010), arXiv:1008.0625.

[86] R. Glauber and G. Mattaie, Nucl.Phys.B **B21**, 135 (1970).

[87] M. L. Miller, K. Reygers, S. J. Sanders, and P. Steinberg, Ann.Rev.Nucl.Part.Sci. **57**, 205 (2007).

[88] PHENIX Collaboration, S. Esumi, J.Phys.G **G38**, 124010 (2011), arXiv:1110.3223.

[89] D. Kharzeev and M. Nardi, Phys Lett B **B507**, 121 (2001).

[90] D. Kharzeev and E. Levin, Phys Lett B **B523**, 79 (2001).

[91] F. Gelis, E. Iancu, J. Jalilian-Marian, and R. Venugopalan, Ann.Rev.Nucl.Part.Sci. **60**, 463 (2010).

[92] A. Kovner, L. D. McLerran, and H. Weigert, Phys.Rev. **D52**, 3809 (1995).

[93] A. Krasnitz and R. Venugopalan, Phys.Rev.Lett. **84**, 4309 (2000).

[94] L. D. McLerran and R. Venugopalan, Phys.Rev. **D50**, 2225 (1994), arXiv:hep-ph/9402335.

[95] C. Gale, S. Jeon, B. Schenke, P. Tribedy, and R. Venugopalan, Phys.Rev.Lett. **110**, 012302 (2013), arXiv:1209.6330.

[96] T. Hirano, P. Huovinen, and Y. Nara, Phys.Rev. **C83**, 021902 (2011).

[97] ATLAS Collaboration, J. Jia, (2012), arXiv:1209.4232.

[98] STAR Collaboration, C. Adler, Phys. Rev. Lett. **89**, 202301 (2002).

[99] PHENIX Collaboration, K. Adcox, Phys. Rev. Lett. **88**, 022301 (2001).

[100] STAR Collaboration, J. Adams *et al.*, Phys.Rev.Lett. **91**, 172302 (2003), arXiv:nucl-ex/0305015.

[101] STAR Collaboration, C. Adler *et al.*, Phys.Rev.Lett. **90**, 082302 (2003), arXiv:nucl-ex/0210033.

[102] ATLAS Collaboration, G. Aad *et al.*, Phys.Rev.Lett. **105**, 252303 (2010), arXiv:1011.6182.

[103] ATLAS Collaboration, G. Aad *et al.*, Phys.Rev.Lett. **110**, 182302 (2013), arXiv:1212.5198.

[104] ATLAS Collaboration, G. Aad *et al.*, Phys.Rev.Lett. **111**, 152301 (2013), arXiv:1306.6469.

[105] P. Steinberg, J.Phys. **G38**, 124004 (2011), arXiv:1107.2182.

[106] CMS Collaboration, S. Chatrchyan *et al.*, Phys.Lett. **B712**, 176 (2012), arXiv:1202.5022.

[107] CMS Collaboration, S. Chatrchyan *et al.*, Phys.Rev. **C84**, 024906 (2011), arXiv:1102.1957.

[108] CMS Collaboration, S. Chatrchyan *et al.*, Phys.Lett. **B718**, 773 (2013), arXiv:1205.0206.

[109] ALICE Collaboration, K. Aamodt *et al.*, Phys.Rev.Lett. **108**, 092301 (2012), arXiv:1110.0121.

[110] B. Muller, J. Schukraft, and B. Wyslouch, Ann.Rev.Nucl.Part.Sci. **62**, 361 (2012), arXiv:1202.3233.

[111] A. Majumder and M. Van Leeuwen, (2010), arXiv:hep-ph/1002.2206.

[112] CMS Collaboration, S. Chatrchyan *et al.*, Eur.Phys.J. **C72**, 1945 (2012), arXiv:1202.2554.

[113] PHENIX, K. Adcox *et al.*, Nucl. Phys. **A757**, 184 (2005), arXiv:nucl-ex/0410003.

[114] STAR Collaboration, C. Adler, Phys. Rev. Lett. **90**, 082302 (2003).

[115] STAR Collaboration, J. Adams *et al.*, Phys. Rev. Lett. **91**, 072304 (2003).

[116] J. Bjorken, *Energy Loss of Energetic Partons in Quark - Gluon Plasma: Possible Extinction of High p(t) Jets in Hadron - Hadron Collisions*, 1982.

[117] M. Gyulassy, P. Levai, and I. Vitev, Nucl. Phys. **B571**, 197 (2000), arXiv:hep-ph/9907461.

[118] M. H. Thomas and M. Gyulassy, Nuclear Physics A **544**, 573 (1992).

[119] I. Vitev, M. Gyulassy, and P. Levai, Heavy Ion Phys. **17**, 237 (2003), arXiv:nucl-th/0204019.

[120] M. Gyulassy, P. Levai, and I. Vitev, Phys. Rev. **D66**, 014005 (2002), arXiv:nucl-th/0201078.

[121] B. G. Zakharov, JETP Lett. **63**, 952 (1996), arXiv:hep-ph/9607440.

[122] B. G. Zakharov, JETP Lett. **65**, 615 (1997), arXiv:hep-ph/9704255.

[123] S. Caron-Huot and C. Gale, (2010), arXiv:hep-ph/1006.2379.

[124] R. Baier, Y. L. Dokshitzer, A. H. Mueller, S. Peigne, and D. Schiff, Nucl. Phys. **B483**, 291 (1997), arXiv:hep-ph/9607355.

[125] R. Baier, Y. L. Dokshitzer, A. H. Mueller, S. Peigne, and D. Schiff, Nucl. Phys. **B478**, 577 (1996), arXiv:hep-ph/9604327.

[126] R. Baier, Y. L. Dokshitzer, S. Peigne, and D. Schiff, Phys. Lett. **B345**, 277 (1995), arXiv:hep-ph/9411409.

[127] A. Majumder, Phys.Rev. **C80**, 031902 (2009), arXiv:0810.4967.

[128] A. Majumder, (2009), arXiv:0912.2987.

[129] A. Majumder and B. Muller, Phys. Rev. **C77**, 054903 (2008), arXiv:nucl-th/0705.1147.

[130] C. N. Parkinson, *Parkinson's Law, or the Pursuit of Progress* (John Murray, 1958).

[131] N. Armesto *et al.*, Phys.Rev. **C86**, 064904 (2012), arXiv:1106.1106.

[132] X.-N. Wang and M. Gyulassy, Phys.Rev. **D44**, 3501 (1991).

[133] N. Armesto, L. Cunqueiro, and C. A. Salgado, Eur.Phys.J. **C63**, 679 (2009), arXiv:0907.1014.

[134] K. C. Zapp, F. Krauss, and U. A. Wiedemann, JHEP **1303**, 080 (2013), arXiv:1212.1599.

[135] K. C. Zapp, Eur.Phys.J. **C74**, 2762 (2014), arXiv:1311.0048.

[136] I. Lokhtin and A. Snigirev, Eur.Phys.J. **C45**, 211 (2006), arXiv:hep-ph/0506189.

[137] C. Young, B. Schenke, S. Jeon, and C. Gale, Phys.Rev. **C86**, 034905 (2012), arXiv:1111.0647.

[138] B. Schenke, C. Gale, and S. Jeon, Acta Phys.Polon.Supp. **3**, 765 (2010), arXiv:0911.4470.

[139] B. Schenke, C. Gale, and S. Jeon, Phys.Rev. **C80**, 054913 (2009), arXiv:0909.2037.

[140] P. B. Arnold, G. D. Moore, and L. G. Yaffe, JHEP **0112**, 009 (2001), arXiv:hep-ph/0111107.

[141] P. B. Arnold, G. D. Moore, and L. G. Yaffe, JHEP **0111**, 057 (2001), arXiv:hep-ph/0109064.

[142] S. Jeon and G. D. Moore, Phys.Rev. **C71**, 034901 (2005), arXiv:hep-ph/0309332.

[143] T. Renk, Int.J.Mod.Phys. **E20**, 1594 (2011), arXiv:1009.3740.

[144] T. Renk, Phys.Rev. **C83**, 024908 (2011), arXiv:1010.4116.

[145] T. Renk, H. Holopainen, J. Auvinen, and K. J. Eskola, Phys.Rev. **C85**, 044915 (2012), arXiv:1105.2647.

[146] G.-Y. Qin and B. Muller, Phys.Rev.Lett. **106**, 162302 (2011), arXiv:1012.5280.

[147] C. E. Coleman-Smith, S. Bass, and D. Srivastava, Nucl.Phys. **A862-863**, 275 (2011), arXiv:1101.4895.

[148] Y. He, I. Vitev, and B.-W. Zhang, Phys.Lett. **B713**, 224 (2012), arXiv:1105.2566.

[149] G.-Y. Qin and A. Majumder, (2009), arXiv:hep-ph/0910.3016.

[150] C. E. Coleman-Smith and B. Müller, Phys.Rev. **C86**, 054901 (2012), arXiv:1205.6781.

[151] A. Majumder, Phys.Rev. **C87**, 034905 (2013), arXiv:1202.5295.

[152] M. Benzke, N. Brambilla, M. A. Escobedo, and A. Vairo, JHEP **1302**, 129 (2013), arXiv:1208.4253.

[153] R. Baier, Nucl.Phys. **A715**, 209 (2003), arXiv:hep-ph/0209038.

[154] P. B. Arnold and W. Xiao, Phys.Rev. **D78**, 125008 (2008), arXiv:0810.1026.

[155] S. A. Bass *et al.*, Phys.Rev. **C79**, 024901 (2009), arXiv:0808.0908.

[156] T. Renk, (2010), arXiv:1004.0809.

[157] T. Renk, Phys.Rev. **C85**, 044903 (2012), arXiv:1112.2503.

[158] K. M. Burke *et al.*, (2013), arXiv:1312.5003.

[159] B. Øksendal, *Stochastic Differential Equations*Universitext (Springer, 2013).

[160] R. B. Gramacy, H. K. H. Lee, and W. MacReady, 21st International Conference on Machine Learning , 353 (2004).

[161] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. (Johns Hopkins Studies in Mathematical Sciences, 1993).

[162] R. Steater and A. Wightman, *PCT, Spin and statistics and all that* (W.A.Benjamin, New York, 1963).

[163] J.-P. Chilès and P. Delfiner, *Geostatistics: Modeling Spatial Uncertainty* (John Wiley & Sons, New York, NY, 1999).

[164] A. T. A. Wood and G. Chan, Journal of Computational and Graphical Statistics **3**, pp. 409 (1994).

[165] P. W. G. Søren Asmussen, *Stochastic simulation: algorithms and analysis* (Springer, 2007).

[166] Q. V. L. Quoc, A. J. Smola, and S. Canu, Heteroscedastic gaussian process regression, in *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[167] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, Most likely heteroscedastic gaussian process regression, in *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[168] H. Wickham, *ggplot2: elegant graphics for data analysis* (Springer New York, 2009).

[169] P. Hoff, *A First Course in Bayesian Statistical Methods* (Springer, 2009).

[170] A. Gelman *et al.*, *Bayesian Data Analysis*, 3 ed. (Chapman & Hall, CRC, 2013).

[171] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov chain Monte Carlo in practice* (Chapman & Hall, London, 1998).

[172] J. Norcedal and S. J. Wright, *Numerical Optimization*Springer Series in Operations Research and Financial Engineering, 2nd ed. (Springer, 2006).

[173] B. N. Datta, *Numerical linear algebra and application* (Pacific Grove, 1995).

[174] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions* (Dover, New York, 1972).

[175] L. S. Bastos and A. O'Hagan, Technometrics **51**, 425 (2009).

[176] F. Ronald, *The design of experiments*, 8th ed. (Hafner, New-York, NY, 1971).

[177] C. K. I. Williams and F. Vivarelli, Machine Learning **40**, 70 (2000).

[178] P. Sollich and A. Halees, Neural Computation **14**, 1393 (2002).

[179] B. Tang, Journal of the American Statistical Association **88**, 1392 (1993).

[180] J. R. Koehler and A. B. Owen*Handbook of Statistics* Vol. 13 (Amsterdam: ElsevierScience, 1996), chap. "Computer Experiments", pp. 261–308.

[181] A. J. Booker *et al.*, Structural Optimization **17**, 1 (1999).

[182] P. Ranjan, D. R. Bingham, and G. Michailidis, Technometrics **50**, 521 (2008).

[183] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis*, 6th ed. (Pearson, 2007).

[184] I. T. Jollife, *Principal Component Analysis* (Springer Series in Statistics, Springer, New York, 2002).

[185] J. Sakurai, *Advanced Quantum Mechanics* (Addison-Wesley, 1967).

[186] J. Tumlinson, ApJ**641**, 1 (2006), arXiv:astro-ph/0507442.

[187] J. Tumlinson, ApJ**708**, 1398 (2010), arXiv:0911.1786.

[188] H.-W. Rix *et al.*, ApJS**152**, 163 (2004), arXiv:astro-ph/0401427.

[189] J. Brinchmann *et al.*, MNRAS**351**, 1151 (2004), arXiv:astro-ph/0311060.

[190] C. Papovich, M. Dickinson, M. Giavalisco, C. J. Conselice, and H. C. Ferguson, ApJ**631**, 101 (2005), arXiv:astro-ph/0501088.

[191] A. E. Shapley, ARA&A**49**, 525 (2011), arXiv:1107.5060.

[192] H. J. Newberg, L. p. o. China, P. in LAMOST, and U. (PLUS), The LAMOST Spectroscopic Survey of Milky Way Stars (LEGUE), in *American Astronomical Society Meeting Abstracts 213*, , Bulletin of the American Astronomical Society Vol. 41, p. 416.14, 2009.

[193] S. C. Keller *et al.*, PASA **24**, 1 (2007), arXiv:astro-ph/0702511.

[194] M. A. C. Perryman *et al.*, A&A**369**, 339 (2001), arXiv:astro-ph/0101235.

[195] LSST Science Collaborations *et al.*, ArXiv e-prints (2009), arXiv:0912.0201.

[196] B. W. O'Shea *et al.*, ArXiv Astrophysics e-prints (2004), arXiv:astro-ph/0403044.

[197] M. L. Norman *et al.*, ArXiv e-prints (2007), arXiv:0705.1556.

[198] V. Springel, MNRAS**364**, 1105 (2005), arXiv:astro-ph/0505010.

[199] J. W. Wadsley, J. Stadel, and T. Quinn, New Astronomy **9**, 137 (2004), arXiv:astro-ph/0303521.

[200] R. Teyssier, A&A**385**, 337 (2002), arXiv:astro-ph/0111367.

[201] V. Springel, MNRAS**401**, 791 (2010), arXiv:0901.4107.

[202] B. W. O'Shea, K. Nagamine, V. Springel, L. Hernquist, and M. L. Norman, ApJS**160**, 1 (2005), arXiv:astro-ph/0312651.

[203] O. Agertz *et al.*, MNRAS**380**, 963 (2007), arXiv:astro-ph/0610051.

[204] E. J. Tasker *et al.*, MNRAS**390**, 1267 (2008), arXiv:0808.1844.

[205] D. Sijacki, M. Vogelsberger, D. Keres, V. Springel, and L. Hernquist, ArXiv e-prints (2011), arXiv:1109.3468.

[206] O. Agertz, R. Teyssier, and B. Moore, MNRAS**410**, 1391 (2011), arXiv:1004.0005.

[207] A. J. Benson *et al.*, ApJ**599**, 38 (2003), arXiv:astro-ph/0302450.

[208] R. G. Bower *et al.*, MNRAS**370**, 645 (2006), arXiv:astro-ph/0511338.

[209] A. J. Benson and R. Bower, MNRAS**405**, 1573 (2010), arXiv:1003.0011.

[210] A. J. Benson, NewA **17**, 175 (2012), arXiv:1008.1786.

[211] Particle Data Group, J. Beringer *et al.*, Phys. Rev. D **86**, 010001 (2012).

[212] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, J.Chem.Phys **21**, 1087 (1953).

[213] M. E. J. Newmann, *Monte Carlo Methods in Statistical Physics* (Oxford University Press, USA, 1999).

[214] S. Koonin, Phys.Lett. **B70**, 43 (1977).

[215] M. A. Lisa, S. Pratt, R. Soltz, and U. Wiedemann, Ann.Rev.Nucl.Part.Sci. **55**, 357 (2005), arXiv:nucl-ex/0505014.

[216] STAR Collaboration, J. Adams *et al.*, Phys.Rev. **C72**, 014904 (2005), arXiv:nucl-ex/0409033.

[217] STAR Collaboration, B. Abelev *et al.*, Phys.Rev. **C80**, 024905 (2009), arXiv:0903.1296.

[218] PHENIX Collaboration, S. Adler *et al.*, Phys.Rev. **C69**, 034909 (2004), arXiv:nucl-ex/0307022.

[219] H. Niemi, G. S. Denicol, P. Huovinen, E. Molnar, and D. H. Rischke, Phys.Rev.Lett. **106**, 212302 (2011), arXiv:1101.2442.

[220] H. Petersen and B. Muller, Phys.Rev. **C88**, 044918 (2013), arXiv:1305.2735.

[221] T. Renk, Phys.Rev. **C85**, 064908 (2012), arXiv:1202.4579.

[222] J.Jacod and P.Protter, *Probability essentials*, 2nd ed. (Springer, 2003).

[223] H. Jeffreys, *Theory of Probability* (Oxford University Press, 1998).

[224] E. Jaynes, *Papers on probability, statistics, and statistical physics* (Kluwer Academic, 1989).

[225] Particle Data Group, J. Beringer *et al.*, Phys.Rev. **D86**, 010001 (2012).

[226] G. Casella and R. L. Berger, *Statistical Inference* (Cengage Learning, 2001).

[227] C. E. Coleman-Smith, J.Phys.Conf.Ser. **446**, 012008 (2013).

[228] C. E. Coleman-Smith, B. Müller, and S. Bass, Nucl.Phys. **A904-905**, 759c (2013).

[229] C. E. Coleman-Smith and B. Müller, (2012), arXiv:1210.3377.

[230] C. E. Coleman-Smith, G.-Y. Qin, S. Bass, and B. Müller, AIP Conf.Proc. **1441**, 892 (2012), arXiv:1108.5662.

[231] C. E. Coleman-Smith, H. Petersen, and R. L. Wolpert, J.Phys. **G40**, 095103 (2013), arXiv:1204.5774.

[232] H. Petersen, C. Coleman-Smith, and R. Wolpert, Acta Phys.Polon.Supp. **6**, 797 (2013).

[233] C. E. Coleman-Smith and B. Müller, (2012), arXiv:1209.3328.

[234] M. Younus, C. E. Coleman-Smith, S. A. Bass, and D. K. Srivastava, (2013), arXiv:1309.1276.

[235] C. E. Coleman-Smith and B. Müller, Phys.Rev. **D89**, 025019 (2014), arXiv:1307.5911.

[236] C. E. Coleman-Smith and B. Müller, Phys.Rev. **D87**, 044047 (2013), arXiv:1212.1930.

Biography

1. Who wrote this document: Christopher Edward Coleman-Smith.

2. When was he born: The 18th of February, 1984.

3. What degrees has he been granted: MPHYS Physics with Theory University of Manchester (2006), MA Duke University (2011).

4. What has he been awarded: A poster award at Quark Matter 2010.

5. Where has the author been a fellow: SAMSI graduate fellow, during the "Big Data" program (2012–2013). The VECC in Kolkata, India as a recipient of the 2011 IUSSTF, U.S.-India Exchange Grant.

6. What are his research interests and contributions?

   - The analysis of computer experiments in Heavy Ion physics and galaxy formation [3, 4, 2, 1].
   - Jet quenching by hot QCD matter. He has extensively developed the VNI/BMS partonic Boltzmann transport model to include the LPM effect, a quantum interference effect for radiation in dense systems, [227, 228, 229, 150, 230, 147]
   - The author has made contributions to the phenomenology of the QGP including, a new method to classify the granularity of the initial state [231, 232] (code available at `https://github.com/jackdawjackdaw/2dfourier`), discussions of relative roles of elastic and radiative energy loss in jet propagation [233], studies of charm quark energy loss [234], and the rolê of fluctuations in the inelastic nuclear cross-section as a means of understanding recent p-Au results [235].
   - Loop quantum gravity phenomenology. Loop quantum gravity predicts a granularity of space with each grain having a quantum behavior. The volume of each grain is quantized and this volume spectrum has a rich structure. These grains of space can be treated as polyhedra with faces of fixed area. It is possible to impose a fascinating Hamiltonian dynamics onto these polyhedra [236].