Bayesian Modelling of the Wairakei Geothermal Surface Network

ENGSCI 700

Logan Wu

Department of Engineering Science

The University of Auckland

Auckland, New Zealand

lwu308@aucklanduni.ac.nz

*Abstract*—The Wairakei geothermal field is one of the oldest geothermal electricity producers in the world, and it has been instrumental in advancing the utilisation of lower enthalpy fluids. Contact Energy Ltd. is the current operator, and they wish to find ways to increase the productivity of their assets and staff.

Contact’s staff use a system of spreadsheets, test results and real-time measurements to update forecasts of the network and make decisions. Their model has no uncertainty measure.

We implement a Bayesian network model that performs a statistical simulation of the Wairakei surface network.

…

# Introduction

Estimation of uncertainty is an essential part to making informative, yet realistic models. The current geothermal model used by engineers at Contact Energy Ltd. in Wairakei, north of Taupo, is deterministic. It does not take into account factors such as measurement uncertainty and parameter uncertainty when modelling the surface network of wells, pipes, flash plants and power plants. Therefore, there is a lack of understanding around how reliable the forecasts generated by the model are, and how this reliability might change in different parts of the surface network.

The desire for a new model that takes into account different sources of uncertainty, from errors in model parameters to errors in measurements, lends itself to Bayesian techniques. Bayesian statistics treats each parameter in a model as a random variable sampled from a distribution. Although its namesake Thomas Bayes lived in the 18th century, Bayesian techniques only became practical in the 1960s to the 1980s with the invention of new Markov Chain Monte Carlo algorithms such as Metropolis-Hastings and sufficient computing power. These algorithms generate stochastic parameter samples from a prior distribution, propagating from the leaves of a directed acyclic graph. Deterministic and/or stochastic operations occur at nodes, which indicate facilities such as wells, flash plants and generators in a geothermal surface network.

The proportional Bayes’ formula is:  
where is the posterior distribution of the parameters *θ* given the data *x*, is the likelihood function of the observed data *x* given *θ*, and is the prior distribution of *θ.* The modeler selects a prior distribution based on their prior beliefs about the parameters. For example:

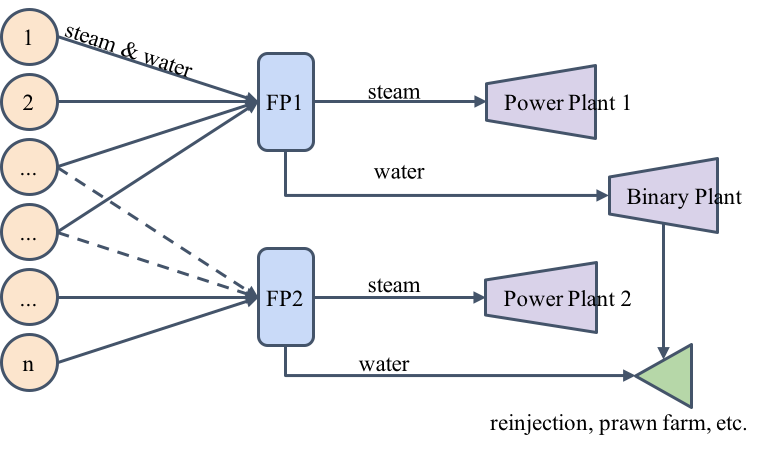
* If it is known that , a Beta prior would be appropriate as it is non-zero on the interval [0, 1].
* If a parameter resides in some ballpark due to expert knowledge, a Normal distribution may be chosen, with an uncertainty determined by the expert’s uncertainty.
* Or, if there is no prior knowledge, this is often represented by a uniform prior, ; any real value is equally likely.

In some cases, these distributions can be chosen using true prior knowledge. Measurement uncertainty is often known to some degree, with some meters rated as having a standard error of 5 units [REF]. Therefore, we set a Gamma prior on measurement error with a mode of 5. This is the preferred method.

The advantage of Bayesian statistics is that the posterior distribution represents the posterior belief of the true network parameters after specifying both the prior beliefs and the data. Frequentist analysis often makes approximations such as the central limit theorem, and does not allow for specification of prior beliefs.

By applying computational Bayesian techniques to the Wairakei geothermal surface network, we will create an algorithmic method to calibrate our model using recorded data, simulate flows in the network and incorporate uncertainty in our predictions.

# Wairakei Network Structure



: CHANGE THIS and add details

At a single point in time, the Wairakei surface network can be represented as a directed, acyclic graph shown in Figure 1. As some wells have multiple flash plants they can be routed to, the arcs that are enabled are pre-determined according to the configuration we wish to simulate.

The main nodes are of three times: wells, flash plants and generators. Each node has inputs, transformations and outputs, and associated parameters.

## Well Nodes

In the model, each well begins with an operating well-head pressure. All active wells generate mixed-phase fluid with the intensive properties of enthalpy *hi* and extensive properties of mass flow rate *ṁi*. Mass flow at each well is determined as a stochastic function of well-head pressure according to the data:

*Haven’t decided how I’ll do the indices. There are a lot*

where:

* are unknown parameters with uniform priors,
* *t* is the number of days since an arbitrary baseline date (set to January 1st, 2000),
* *εi* are normally distributed errors with zero mean and unknown variance *σ2*, which has a uniform positive prior.

An independent model is set for each well, but measurement error variance is the same because each well has a similar flow meter. *will test this assumption*

By using uninformative priors, the posterior distribution for the parameters of one well is equal to the likelihood function:

This regression is approximately equivalent to a frequentist ordinary-least-squares regression with errors in the parameters given by a covariance matrix.

## Flash Plant Nodes

Each flash plant takes mass flow inputs from a partition of the set of wells, such that every (active) well maps to one flash plant, but more than one well can map to the same flash plant. This analysis uses the results from a set partitioning optimisation when assigning a subset of wells to each flash plant.

At a flash plant *j*, the extensive properties are summed over its subset of wells *I*(*j*) and intensive properties are mass flow-weighted averages.

These formulae do not include any drop in enthalpy. Wairakei surface pipes are insulated and heat loss is estimated at 0.6% in the pipes by [REF], who concludes that it is negligible. *Will need to investigate why the flash plants don’t have enthalpy loss*

Flash plants are critical features in the Wairakei geothermal system because they convert the wet, mixed-phase fluid into pure steam [REF]. One of the unique technical difficulties faced by Wairakei engineers is impure steam causes pitting and maintenance issues in the turbines. The flash plants use one or two pressure drops to boil the liquid component of the fluid. The resulting steam mass flow from a drop to pressure *P* can be calculated by:

where *hf@P* is the specific enthalpy of saturated water at pressure *P* and *hfg@P* is the latent heat of evaporation. The remaining mass flow is the liquid fraction.

Although these are deterministic formulae, the mass fractions can become stochastic if there is uncertainty in the flash plant pressure *P*.

## Generator Nodes

Each flash plant inserts into a single generator, again in a many-to-one set partitioning relationship where the subset of flash plants flowing into generator *k* is *J*(*k*). Currently, the power output *Ẇk* from a generator *k* with efficiency *ηk* is proportional to the mass flow of steam feeding it.

We are interested in the posterior distribution of total power output, , as this is what we wish to maximise. Analysis of intermediate variables within the network gives insight into where sources of variation or uncertainty in total power output arise.

# Data

We use data supplied by Contact in several forms.

## Network Structure

Contact has provided a schematic indicating the connectivity of wells, to flash plants, to generators. In some cases, wells have in-built flash plants. These are treated the same as any other flash plant, except they only have one well feeding them. When a well has the option to feed to several flash plants, we use an optimisation algorithm to choose which one it is mapped to.

## Well Data

To make a prediction for the mass flow at each well under a given operating pressure, we fit a model to historical wellbore tests.

*Following copy/pasted from literature review. Can I?*

*Bore tests* involve taking a well offline, applying a testing apparatus to the well-head, and running the disconnected well at three different pressures to generate an elliptical estimate of the function , where is the mass extraction rate and is the well-head pressure. *Tracer flow tests* (TFTs) inject a tracer dye at the well-head under normal operating conditions. The well can continue production during the TFT and mass flow is measured under realistic conditions, but this only provides one data point which cannot describe a curve. [REF]

The operating well-head pressures and enthalpies we use to make predictions on are taken from the most recent data. *TODO: fit a model + uncertainty to WHP & h rather than taking the most recent reading*.

## Flash Plant and Generator Data

While flow readings from wells are only taken during tests, flash plants have flow meters that report in real-time and are logged once a day. To fit a model that predicts flow

*TODO: I haven’t done this yet*

# Simulation Methods

The main historical barrier to Bayesian statistics is the computational cost of calculating posterior distributions. In frequentist statistics, the use of approximations allows standard distributions (e.g. the exponential family, notably the Normal distribution) to be used. Operations are therefore cheap; an example is the Central Limit Theorem – the average of *n* independent and identical distributions converges to a Normal distribution.

With Bayesian statistics, no such assumptions are made. If conjugate distributions are used for the prior and likelihood, the posterior can be calculated analytically. However, this is often not the case and sampling methods are used.

Sampling methods rely on the law of large numbers. Generally, we assume that if we take enough samples from a distribution, we will converge to that distribution. This is often treated as a black box by the programmatic abstraction, but this section will discuss the background and justifications for some of the coding choices made in our high-level implementation.

## Sampling Algorithms

*I don’t know if I should have a section on this. Would mention Markov Chain Monte Carlo (MCMC) methods and JAGS/PyJAGS. May be more appropriate for the literature review? Brief explanation of MCMC would explain why burn-in is necessary.*

## Diagnostics

One of the difficulties with sampling simulations that use Markov chains is they often include a burn-in (warm-up) period before settling into the stationary distribution of the chain. In most practical uses, there is no way to predict convergence, so it must be done by monitoring the sample trace and running diagnostic tests.

The implementation of Gibbs Sampling we use, JAGS (Just Another Gibbs Sampler), also provides diagnostic tests in the CODA package. Diagnostic tests for stochastic simulation are important because of the MCMC methods used – in general, convergence of the trace to a steady state distribution is required before any inference can be drawn from the results. There are two main tests to confirm this:

### Geweke

Geweke’s convergence diagnostic for MCMC samples tests for equality of the means in the first 10% and last 50% of the trace (the samples in iteration order). The means will be equal f the sample is drawn from a stationary distribution, indicating the burn-in period has been successfully excluded. Geweke’s statistic has a T-distribution so the following T-test statistic can be used:

where and are the sample means of the first 10% and last 50% of the samples, *s2* is their corresponding standard variance, and *n* and *m* are the number of samples in the two groups. Spectral densities are used to estimate the sample variances. [REF]

### Gelman [REF]

Gelman’s test gives the potential scale reduction factor for each parameter. This requires at least two parallel chains, and tests whether the chains have converged to identical distributions. If the chains have not converged, the scale reduction factors will have upper confidence limits greater than one and the samples obtained are likely to be over-dispersed.[REF]

*TODO: Confirm simulation passes both tests and others (Raftery, autocorrelation).*

# Data Integration and Simulation

Direct integration with Contact’s PI systems was not possible for this study because of the commercial sensitivity of live asset data. Exported Excel worksheets therefore provide the main source of data input for our implementation.

## Data Extraction

Accessing the data using the Microsoft Excel desktop application is slow, on the order of ten minutes for the sample data supplied, but hours for the actual operational spreadsheet. Instead, we use a Python script that accepts our unmodified data spreadsheets and immediately converts the data into Pandas DataFrame objects. It has sufficient automatic data-cleaning capabilities that fix known inconsistencies such as capital letters, reject incomplete or erroneous lines, and discriminate between data and meta-data such as comments. Reading from file takes around five seconds, and data cleaning is instantaneous.

## Pre-Processing

The original Excel spreadsheets are in human-readable formats. These include well names rather than well IDs, and lacks certain helpful metadata such as the quantities of each facility or the mappings of wells to flash plants.

Adding variables such as unique well IDs ranging from 1 to *N*, flash plant IDs from 1 to *M* and generator IDs from 1 to *O* prepares the data to be accepted by JAGS.

To make predictions, JAGS can take a missing value (masked array for PyJAGS, *NA* for rjags) and makes samples just as it would for an unknown parameter. These missing values are appended to the input value.

## Simulation

*Include DAG diagram showing variable relationships*

We initialise four parallel Markov chains with a burn-in sample of 500 iterations until convergence to a stationary distribution. *Will increase all these numbers.* We record the next 2000 samples per chain, for a total of 8000 samples from the stationary posterior distribution.

## Verification

*Shows results from convergence diagnostics.*

# Results

##### References