

---

# BAYESIAN MODELLING OF THE WAIRAKEI GEOTHERMAL SURFACE NETWORK

---

LOGAN WU

*UPI: LWU308*

*ID: 254620375*

PARTNER: VIDA FOX

SUPERVISORS: DR. DAVID DEMPSEY AND DR. CAMERON WALKER

THE UNIVERSITY OF AUCKLAND

*Department of Engineering Science*

*September 17, 2018*

## Abstract

The Wairakei geothermal field is one of the oldest geothermal electricity producers in the world, and is unique in its utilisation of lower enthalpy fluids. The current operator, Contact Energy Ltd., uses a system of spreadsheets, test results and real-time measurements to update forecasts of the network and make decisions. Their model has no formal measure of uncertainty.

We implement a Bayesian network model that performs a statistical simulation of the Wairakei surface network, simultaneously incorporating data from both sources and over a long period of time to estimate the network parameters.

Our simulation incorporates time into production curves using a linear model, and propagates uncertainty from the wells further down the network to probabilistically verify flows where operational constraints may need to be adhered to. We also find that manual well test data and automatically logged data differ in utility, with test data being better for estimating well production parameters, and logs from the PI system being more effective for forecasting.

Feels like I'm describing what I've done, not summarising. Let me know.

# Contents

<b>1 Introduction</b>	<b>1</b>	7.1 Model Specification . . . . .	11
<b>2 Advantage of Bayesian Estimation</b>	<b>2</b>	7.2 Monitoring . . . . .	15
<b>3 Wairakei Network Structure</b>	<b>3</b>	7.3 Diagnostics . . . . .	16
3.1 Well Nodes . . . . .	4	<b>8 Results</b>	<b>18</b>
3.2 Flash Plant Nodes . . . . .	4	8.1 Well Production Curves . . . . .	18
3.3 Generator Nodes . . . . .	5	8.2 Well Mass Flows . . . . .	18
<b>4 Data Sources</b>	<b>5</b>	8.3 Individual Well Declines . . . . .	20
4.1 Network Structure . . . . .	5	8.4 Down-flow Results . . . . .	21
4.2 Well Test Data . . . . .	5	<b>9 Conclusions</b>	<b>23</b>
4.3 PI Flow Meters . . . . .	5	<b>10 Further Development</b>	<b>23</b>
4.4 Uncertainty . . . . .	6	10.1 Direct Data Integration . . . . .	23
4.5 Constraint Limits . . . . .	7	10.2 Data Integration With Other Models . . . . .	24
<b>5 Data Integration</b>	<b>7</b>	10.3 Time Series . . . . .	24
5.1 Data Extraction . . . . .	7	10.4 Prior Specification . . . . .	24
5.2 Pre-Processing . . . . .	7	10.5 Hierarchical Model Resolution	24
5.3 Automatic and Manual Configuration . . . . .	8	<b>Appendices</b>	<b>25</b>
<b>6 Simulation Methods</b>	<b>8</b>	<b>A Time Series</b>	<b>25</b>
6.1 JAGS . . . . .	8	<b>B Verification</b>	<b>26</b>
6.2 Sampling Algorithms . . . . .	9	<b>C All Well Declines</b>	<b>27</b>
6.3 Gibbs Sampling . . . . .	10	<b>11 Flash Plant Data Sources</b>	<b>28</b>
<b>7 Simulation Implementation</b>	<b>11</b>	<b>12 JAGS Model</b>	<b>29</b>

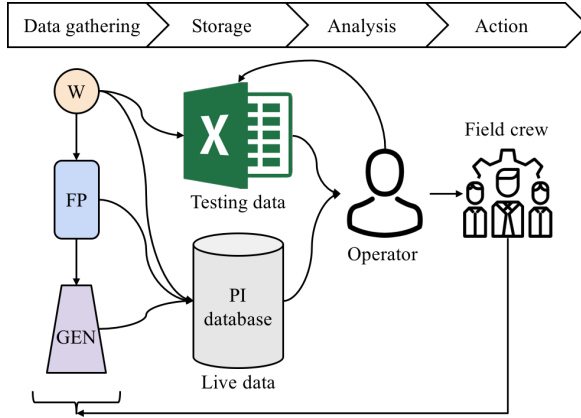


Figure 1: Current system diagram. Data on the surface network components (wells, flash plants and generators) is stored in two systems, which an operator reconciles to build a model.

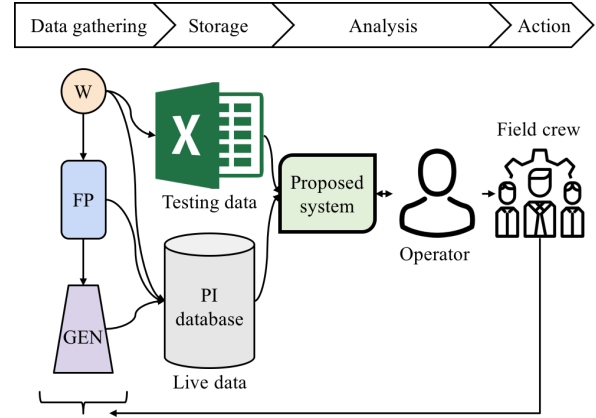


Figure 2: Proposed system, which ingests data from both sources, incorporates it into a model and interacts with the operator to test scenarios and make predictions.

## 1 Introduction

Contact Energy Ltd. (CEL), the operator of the Wairakei geothermal field, records a combination of field tests and live data to monitor the state of the geothermal surface network. Staff use the data to make operational decisions about well maintenance, valve positions and long-term sustainability. Currently, the flow of information in this system looks like Figure 1. Data from flow meters and well tests is stored for analysis by an operator, who manages the maintenance and operation of the field.

One of the major pain points is between the storage and analysis stages. To access the PI database, the operator exports data into an Excel spreadsheet. They then compare it with the well test data stored in another spreadsheet.

After calibrating the test data, running regressions and making forecasts, they obtain metrics about each well's condition and make recommendations such as whether to perform a work-over to remove deposits inside the well bore. Accessing and processing the data is a highly manual task, and the sheets have become cumbersome and slow to open. Calibration requires experience to know what the values should look like, adding dependence on a single operator.

In this paper, we develop the intermediate system shown in Figure 2 between the data storage and operator, that integrates the two datasets and assists in the operator's tasks by automating the statistical analysis.

There are two advantages to our proposed system:

1. Data from multiple sources are combined into a statistical model that includes uncertainty using Bayesian statistics.
2. The operator can interact with the internal model through Excel to conduct scenario analysis and automatically visualise the results.

## 2 Advantage of Bayesian Estimation

Estimation of uncertainty is an essential part of making informative, yet realistic models. The current geothermal model used by engineers at CEL 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. 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.

Historically, the main barrier to Bayesian statistics was 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. Many problems with Normal distributions have analytic solutions and are therefore cheap. An example is the Central Limit Theorem for large data sets, where the average of  $n$  independent samples converges to a Normal distribution.

With Bayesian statistics, no such assumptions are made. The key components of Bayesian statistics are three functions: a prior distribution, a likelihood function and a posterior distribution. The prior is an expert/modeller's initial belief of what the model parameters could be, before seeing any data. The likelihood function is the statistical likelihood of observing the data given any parameter value drawn from the prior, and this is used to update the prior to create the posterior, the probability of the parameter values given both the expert knowledge and the observed data. Computing the posterior is the goal, and it can either be done analytically or through simulation.

In our network model, deterministic and/or stochastic operations occur at nodal facilities such as wells, flash plants and generators.

The proportional Bayes formula is:

$$f(\theta|x) \propto f(x|\theta)f(\theta) \quad (1)$$

where  $f(\theta|x)$  is the posterior distribution of the parameters  $\theta$  given the data  $x$ ,  $f(x|\theta)$  is the likelihood function of the observed data  $x$  given  $\theta$ , and  $f(\theta)$  is the prior distribution of  $\theta$ . The modeller selects a prior distribution based on their prior beliefs about the parameters. For example:

1. If it is known that  $\theta_i \in [0, 1]$ , a Beta prior would be appropriate as it is non-zero on the interval  $[0, 1]$ .
2. If a parameter resides in some ballpark due to expert knowledge, a Normal distribution may be chosen, with a variance determined by the expert's knowledge.
3. Or, if there is no prior knowledge, this is often represented by a uniform prior  $f(\theta) \propto 1$ , where 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  $\sigma$  of 5 units [REF]. Therefore, we set  $f(\sigma^2)$  as an Inverse-Gamma prior on measurement error

	Frequentist	Bayesian
Parameter $\theta$	Fixed by a null hypothesis $h_0$	Unknown and probabilistic
Data $\vec{x}$	Random; we observe a sample	Fixed; from an unknown distribution
95% CI	Under repeated sampling, 95% of CIs will contain the true $\theta$	$\theta$ has a 95% chance of lying within the CI

Table 1: Differences in interpretation between the two main statistical paradigms. The Bayesian interpretation is usually more intuitive, but often the end results are similar.

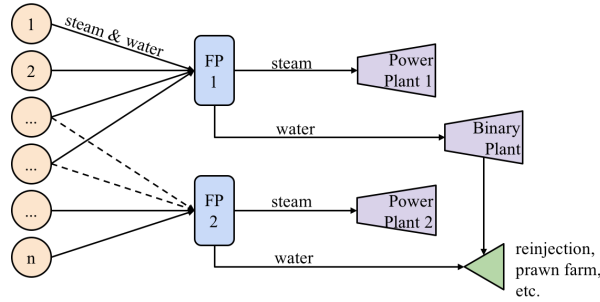


Figure 3: Simplified structure of the Wairakei geothermal surface network. Some wells can send steam to a selection of flash plants, but only one at a time (indicated by the solid/dashed lines).

with a mode of 5<sup>2</sup>. This is the preferred method because it satisfies the Bayes formula, where the prior is selected before observing the data.

The advantage of Bayesian statistics is that the posterior distribution represents the belief of the true network parameters after both specifying any expert information and observing the data. Bayesian credible intervals also have a more intuitive interpretation than the Frequentist confidence interval. We use the term ‘credible interval’ whenever the Bayesian interpretation is used (e.g. parameter estimates), and ‘confidence interval’ for Frequentist interpretations (e.g. the T-test). Differences in terms are detailed in Table 1.

By applying computational Bayesian techniques to the Wairakei geothermal surface network, we will create an algorithmic method to calibrate our model  $y = f(\theta|\vec{x})$  using recorded data, simulate probabilistic flows in the network and incorporate uncertainty in our predictions.

### 3 Wairakei Network Structure

At a single point in time, the Wairakei surface network can be represented as a directed, acyclic graph shown in Figure 3. As some wells have multiple flash plants they can be routed to, the active arcs are pre-determined according to the configuration we wish to simulate.

There are three types of nodes: wells, flash plants and generators. Each node has inputs, transformations, outputs and associated parameters we might like to know.

### 3.1 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  $h_i$  and extensive properties of mass flow rate  $\dot{m}_i$ . Their well-head pressure is set by the operator – note that well-head pressure is not the pressure differential between the wellbore and atmospheric. It is the pressure measured at the gauge, so zero indicates the well is at maximum flow venting into the atmosphere.

The mass flow can be predicted using wellbore simulators such as TOUGH2, but these take a long time to run and it is easier for CEL to evaluate an statistical approximation as  $\dot{m} = f(\vec{x})$ . CEL approximates  $f$  using well test data, using three data points from a given day to fit two degrees of freedom. Often the full data set includes many sets of tests over several years.

CEL fits multiple models using a small subset of the data for each set of well tests. We use their data set to calibrate a regression model with time effects and mass flow variance by using data over different dates and from multiple sources. By incorporating more data, our regression model can also estimate uncertainty in its parameters.

Introduc  
equa-  
tions

### 3.2 Flash Plant Nodes

Flash plants take flow inputs from a partition of the set of wells, such that every (active) well maps to one flash plant in a many-to-one relationship. Configurations can come from historical records, external optimisation routines or any hypothetical setup we wish to test.

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

$$\dot{m}_j = \sum_{i \in I(j)} \dot{m}_i \quad \forall j \quad (2)$$

$$h_j = \frac{\sum_{i \in I(j)} \dot{m}_i h_i}{\sum_{i \in I(j)} \dot{m}_i} \quad (3)$$

These formulae assume conservation of mass and enthalpy, which holds if network components are sufficiently sealed and insulated. Enthalpy loss in the Wairakei pipes is estimated at 0.6% in the pipes by [REF], who concludes that it is negligible.

Impure steam from the wells causes pitting and corrosion in the generator turbines. Flash plants convert the fluid into steam with one or two pressure drops to boil the liquid component of the fluid [REF]. The resulting steam mass flow from a drop to pressure  $P$  can be calculated by:

$$\dot{m}_{\text{steam},j} = \chi \dot{m}_j, \quad \chi = \min \left\{ \max \left\{ \frac{h_j - h_{f@P}}{h_{fg@P}}, 0 \right\}, 1 \right\} \quad (4)$$

where  $h_{f@P}$  is the specific enthalpy of saturated water at pressure  $P$  and  $h_{fg@P}$  is the latent heat of evaporation. Vapour quality  $\chi$  is bounded between zero and one, and the remaining mass flow is the liquid fraction.

The outflowing steam and water are not constrained to go to the same generator. For example, several flash plants send steam and water to the Poihipi and binary plants respectively.

### 3.3 Generator Nodes

Generators accept intermediate and low pressure steam or water from flash plants, where the subset of flash plants supplying steam to generator  $k$  is  $I_{\text{steam}}(k)$  and the subset of flash plants supplying water is  $I_{\text{water}}(k)$ . Currently, the power output  $\dot{W}_k$  from a generator  $k$  with efficiency  $\eta_k$  is proportional to the mass flow of steam feeding it:

$$\dot{W}_k = \eta_k \sum_{i \in I_{\text{steam}}(k)} \dot{m}_{\text{steam},i} \quad \text{for steam generators} \quad (5)$$

or

$$\dot{W}_k = \eta_k \sum_{i \in I_{\text{water}}(k)} \dot{m}_{\text{water},i} \quad \text{for pentane (binary) generators} \quad (6)$$

We are interested in the posterior distribution of total power output,  $\sum_{k \in K} \dot{W}_k$ , 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.

## 4 Data Sources

We use numerical data supplied by Contact in several forms, including network schematics, raw data and knowledge about uncertainty and limits in the field. We wish to build a production curve model and use this to make forecasts, given the operating pressures for each well. A component of this project is integrating multiple data sources to predict mass flow because neither of the raw data sources covers all the wells on their own.

### 4.1 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, the configuration is pre-determined by any operational decision.

### 4.2 Well Test Data

Well tests from as far back as 2002 are recorded in an Excel spreadsheet. Wellbore tests are performed at multiple operating pressures to fit a production curve, as discussed in the Literature Review. The spreadsheet also contains results from tracer flow tests (TFTs), which are easier and cheaper to run because the well can remain connected to the network. We use regression techniques to replicate the production curves with more data.

Tests are only performed on the liquid wells. Well-head pressure, mass flow and enthalpy are recorded manually. The liquid wells are wells with a mixed-phase fluid output, and are the main focus of this model because of their readily available data and relationship with the flash plants that separate their wet steam. ‘Dry’ wells (wells with integrated separators that only output steam) do not have well test data, and are covered using exported flow meter data from CEL’s automatic loggers.

### 4.3 PI Flow Meters

Real-time data is supplied using flow meters. The benefit of live data is that it is stored once a day in a PI system (treated here as a generic database) in a regular time-series containing

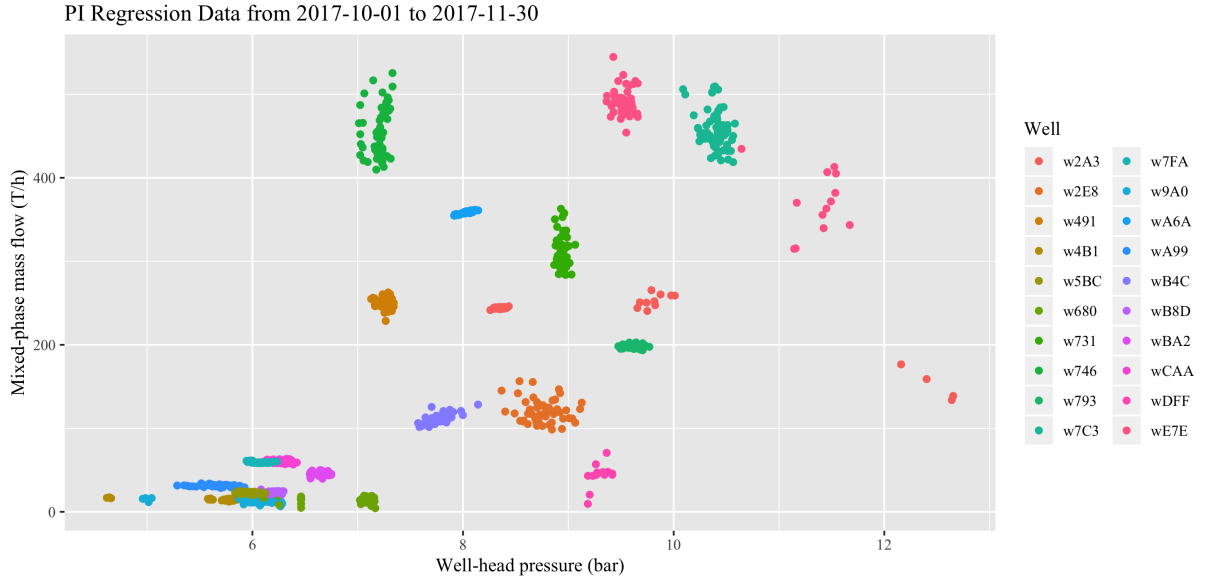


Figure 4: One month of the most recent regression data from the PI database. We use a combination of the wellbore test data (not shown) to estimate the regression parameters, and the PI data to increase the weight and precision for short-term forecasts.

every meter. It is therefore much more regular than the well test data. The parameters recorded include well-head pressure, separator pressure and mass flows for some facilities.

Twenty liquid wells also have pressure gauges and flow meters logged in PI, and seventeen dry wells are not included in the liquid wells regression data. The data from the twenty wells shown in Figure 4 is included in the production curve regression, and we include one month (thirty instances) of data to increase the weight in the regression on the most up-to-date points. We also use them to forecast the operating pressures. However, they cannot regress the production curve on their own because they tend to all be of the same well-head pressure and the regression parameter estimation would be ill-conditioned.

PI data is used for the ten wells without well-head pressure in a time-series analysis without a production curve. This allows us to ‘fill in’ the gaps from the wellbore test data even though we cannot model the relationship between well-head pressure and mass flow.

#### 4.4 Uncertainty

To take full advantage of the Bayesian framework, we want to specify well-informed parameters. We have obtained prior estimates for some of the measurement uncertainties by correspondence with Contact Energy:

1. Well test mass flow measurements are  $\pm 5\%$ - $10\%$
2. Flash plant mass flow measurements are  $\pm 10\%$
3. Steam to power conversion factors are up to  $\pm 5\%$

Turning these statements into prior specifications is to the modeller’s discretion and will be discussed for the model. However, in practice most sensible priors work if there is enough data available.





Figure 5: Our project includes a Python script to process the sample data files, and an R script which runs a model and processes the results.

## 4.5 Constraint Limits

Flash plants have a flow limit on the fluid components. This data is not a component of the Bayesian model, but can be used when analysing the outputs to check the probability of a constraint violation given a certain network configuration.

1.  $\text{fpE2} < 525 \text{ T/h}$  of IP and LP steam
2.  $\text{fp62} < 775 \text{ T/h}$
3.  $\text{fp19 IP} < 420 \text{ T/h}$
4.  $\text{fp19 IP+} < 450 \text{ T/h}$

## 5 Data Integration

Direct integration of our routine with Contact’s PI systems was not possible for this study because of the sensitivity of live asset data. Exported Excel worksheets therefore provide the main source of data input for our implementation. This section details how we extract the raw data from provided Excel samples, how we process it and what processed data our statistical analysis requires.

### 5.1 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 much longer for the actual operational spreadsheet. Therefore, we use a Python script that accepts our unmodified data spreadsheets and immediately converts the data into efficient data frame objects. Without the overhead of Excel and the myriad of formulae within the spreadsheet, loading the data into memory from storage takes seconds.

Our Python routine implements 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. Data cleaning takes negligible time.

### 5.2 Pre-Processing

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

The second half of the Python script maps facility names to unique integer IDs and converts time formats into the number of days since an arbitrary baseline. These allow the data to be ingested by R. We hash all the well and flash plant names before generating outputs. Wells begin with “W” and flash plants begin with “FP”.

Explain  
FP16  
(or  
ignore  
if it  
isn't  
important

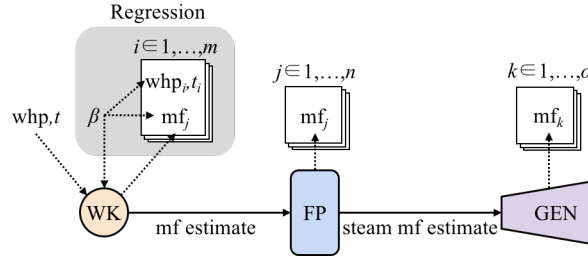


Figure 6: One iteration of slice sampling, where a univariate density is sampled as a bivariate uniform density.

We found at least one data point for 62 wells. Data was missing the following:

	Wells
Assigned to FP but no data	w6D0, wDF4, w04E, w38B, w709
Data available but no FP	wAAB, w1EB, w35E, w424, w8AF, wFAB

Table 2: Potential data quality issues. wFAB is known to be not connected, and wB98 has an A/B pairing with w2F7.

The impact of missing well data includes under-estimation of flows feeding flash plants and generators. Mass proportions at the separators may also be affected if the missing wells have extremely high or extremely low enthalpies.

### 5.3 Automatic and Manual Configuration

The rest of the workflow is carried out in R. To make the program usable to non-programmers, R reads in configuration options from a separate Excel spreadsheet. Here, the user configures the well mappings and the pressures at which they intend to operate the wells. The entire network structure can also be changed to test scenarios with different facilities.

An R script reads in the processed data and the configuration file. It uses these to construct instructions for our simulation, specifying the stochastic network's structure and parameters. R also acts as our simulation interface, performing post-processing and visualisation of the outputs.

## 6 Simulation Methods

When calculating a posterior density, often there is a tradeoff between flexibility and efficiency, with Markov Chain Monte-Carlo (MCMC) methods being the most flexible and analytic evaluation being the most efficient but sometimes impossible. We use a specific implementation of MCMC called JAGS through RJAGS, a package for the R language. This section details the components of JAGS used by our model.

### 6.1 JAGS

JAGS (Just Another Gibbs Sampler) is a GNU-licensed program that implements a Markov Chain Monte-Carlo method called Gibbs sampling. The exact sub-components of its Gibbs routine are abstracted as a black-box to the user through an R-like syntax.

The statistical model we input into JAGS is a directed, acyclic graph. We then supply JAGS with priors for parameter nodes and data for any nodes with observations.

Figure 6 shows the information flow between prior nodes, intermediate values and observed values. It is based on a graphical interface from a similar program, WinBUGS, where solid lines are deterministic relationships and dotted lines are stochastic relationships. Root nodes are prior distributions specified by the modeller, and leaf nodes are observations.

To computationally compute the posterior, JAGS samples from the priors  $f(\theta)$  at the root nodes (here, the well-head pressures  $whp$  and regression parameters  $\beta$ ) and propagates forward through the arcs. When a parameter value reaches a node we observe through measurements such as mass flow, the likelihood function  $f(\vec{x}|\theta)$  is computed. The product of prior and likelihood  $f(\vec{x}|\theta)f(\theta)$  becomes the posterior probability  $f(\theta|\vec{x})$  for the set of parameter values  $\theta$  in that iteration.

By computing the posterior distributions for all parameters, we extract the values of the regression coefficients and estimates for flows at any point in the network.

## 6.2 Sampling Algorithms

At run-time, JAGS automatically chooses the most appropriate algorithm for each node, so different nodes can use different methods. These methods come from separate modules – *base* JAGS and *BUGS*.

Both of the following sampling methods are used in iterations of a Gibbs Sampler, which will be discussed afterwards.

### 6.2.1 BUGS::Conjugate

The conjugate sampler in the BUGS (Bayesian inference Using Gibbs Sampling) module is used when a parameter's posterior is a conjugate distribution to the prior. Conjugate distributions are where the posterior  $f(\theta|x)$  and the prior  $f(\theta)$  come from the same family of distributions. This holds when the prior is the conjugate prior to the likelihood  $f(x|\theta)$ . The conjugate distributions (priors and posteriors) used in this model are the Normal and Gamma distributions, when the likelihood is a Normal distribution.

Conjugate priors and likelihoods are used wherever possible because the resulting posterior can be calculated analytically. For example, if we have the likelihood of a parameter as  $f(\vec{x}|\mu, \sigma^2) \sim N(\mu, \sigma^2)$  where the conjugate prior distributions on mean and variance are  $N(\mu_0, \sigma_0^2)$  and  $\text{Inv-}\gamma(\alpha, \beta)$  respectively, we can calculate the analytic posterior for the individual parameter assuming the other is fixed:

$$f(\mu|\vec{x}, \sigma^2) \sim N\left(\frac{1}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}}\left(\frac{\mu_0}{\sigma_0^2} + \frac{\sum \vec{x}}{\sigma^2}\right), \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right)^{-1}\right) \quad (7)$$

$$f(\sigma^2|\vec{x}, \mu) \sim \text{Inv-}\gamma\left(\alpha + \frac{n}{2}, \beta + \frac{\sum (\vec{x} - \mu)^2}{2}\right) \quad (8)$$

Note that in JAGS, the Normal distribution is often parameterised by precision  $\tau = 1/\sigma^2$  rather than variance, leading to a Gamma conjugate prior instead of Inverse-Gamma.

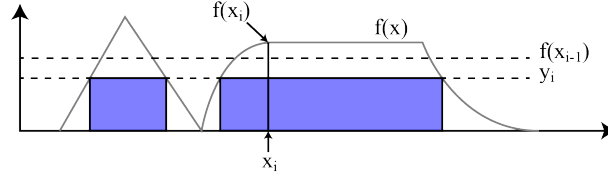


Figure 7: A single iteration of slice sampling where a bivariate MCMC state space is constructed from a univariate density.

Once the posterior parameters are obtained, samples are obtained by a range of methods often specific to one distribution family such as the Box-Muller method, an efficient method for transforming independent uniform (pseudo) random variates into standard Normal samples.

### 6.2.2 base::Slice

In our model, a Slice Sampler is used for all other parameters in the model that do not use conjugate distributions. The principle of slice sampling treats a univariate density as a uniform bivariate density, with one of the variates giving the same steady-state posterior as the original univariate.

The steps to take a slice sample are demonstrated in Figure 7: [REF]

1. Given  $x_{i-1}$ , sample  $y_i$  uniformly from  $[0, f(x_{i-1})]$ .
2. Given  $y_i$ , sample  $x_i$  uniformly from  $\{x | f(x) > y_i\}$ .

The long-run distribution of  $x_i$  will converge on  $f(x)$ . Slice sampling can also be used for discrete variables, but our model only uses continuous parameters.

Slice sampling is a special form of random walker (Metropolis-Hastings) that is relatively simple to implement. By solving the inverse problem for the set  $\{x | f(x) > y_i\}$ , we allow it to jump to any  $x \in \{x | f(x) > y_i\}$  and it avoids issues faced by other Metropolis algorithms where the random walker can become trapped by zero-valued regions.

## 6.3 Gibbs Sampling

The previous methods are all implementable as part of a Gibbs sampling algorithm. A Gibbs sampler is a MCMC (Markov Chain Monte-Carlo) method, where each iteration depends on the state of the chain in the previous iteration in a stochastic function determined by the individual samplers.

Our goal is to determine the full joint distribution of all the parameters in our model, which is much more difficult than univariate or bivariate distributions. MCMC allows us to construct an ergodic Markov chain whose equilibrium or stationary distribution converges on the full, multivariate joint distribution. If it is ergodic, it will satisfy two conditions:

1. The chain can explore all possible parameter states from any starting state.
2. When run to infinity, its expected state distribution is equal to the joint distribution.

In the Gibbs sampler, each iteration consists of sampling one parameter (or a block of pa-

rameters when possible, e.g. multivariate Gaussian) given the most recent values of the other parameters. One iteration proceeds as follows, where  $i$  is the iteration for  $p$  variables:

$$\theta_1^{(i)} \sim f\left(\theta_k|\vec{x}, \theta_2^{(i-1)}, \theta_3^{(i-1)}, \dots, \theta_p^{(i-1)}\right) \quad (9)$$

$$\theta_2^{(i)} \sim f\left(\theta_k|\vec{x}, \theta_1^{(i)}, \theta_3^{(i-1)}, \dots, \theta_p^{(i-1)}\right) \quad (10)$$

$$\vdots$$

$$\theta_p^{(i)} \sim f\left(\theta_k|\vec{x}, \theta_1^{(i)}, \theta_2^{(i)}, \dots, \theta_{p-1}^{(i)}\right) \quad (11)$$

This specific MCMC scheme is useful because instead of sampling from a  $p$ -dimensional posterior, we sample from  $p$  one-dimensional posteriors. Although this means each sample is not independent as would be expected from direct sampling routines, its equilibrium state still converges to the correct distribution. The time taken to reach equilibrium depends on how well the random walkers mix within their distributions, which is why conjugate and slice samplers are effective as they do not use a fixed step-size.

## 7 Simulation Implementation

JAGS code is language-agnostic and defined using a text string, Appendix [REF]. We use R instead of similar Python packages because the R interface is better supported. The RJAGS package also includes extra model and convergence diagnostics that are not available in Python, but the model declaration is still the same in any language.

There are three main steps to running a JAGS model: model specification, post-processing and model diagnostics.

### 7.1 Model Specification

JAGS code is declarative and interpreted simultaneously. However, our code can still be interpreted as a set of steps where each block leads into the next. We begin at the wells and progress through the network to the generators.

#### 7.1.1 Covariate Centering

Our code contains a generalised linear model (GLM) in Equation 15 that is incompatible with JAGS' *GLM* module, a specialised sampler for GLMs that is efficient when there is covariance between the parameters. Since we cannot use the module, we center the covariates. For example:

$$x_{whp} \leftarrow x_{whp} - \overline{x_{whp}} \quad (12)$$

When there is high covariance, univariate samplers mix poorly because the step of one parameter is highly dependent on the value of another parameter rather than being mostly random. Centering the covariates makes the expectation of covariance zero, and we observe better mixing afterwards.

#### 7.1.2 Well Production Curve Regression

One of CEL's current tasks is to fit a model to well production curves, where mass flow is a function of well-head pressure. Production curves change over time as the well and reservoir conditions change. Grant and Bixley propose a shifted elliptic form because it has interpretable

real-world parameters of a maximum pressure and a maximum mass flow.

$$\frac{(\dot{m} - \beta_1)^2}{\beta_2^2} + \frac{P_{\text{wh}}^2}{\beta_3^2} = 1 \quad (13)$$

Contact Energy's existing spreadsheets use a centered ellipse, which only requires two data points to fit rather than three for the extra axis shift parameter:

$$\frac{\dot{m}^2}{\beta_1^2} + \frac{P_{\text{wh}}^2}{\beta_2^2} = 1 \quad (14)$$

In practice, our model is not trying to estimate maximum pressures and mass flows, since these are theoretical and are not obtained in the field [REF Grant & Bixley]. Therefore, we are not restricted to this form of equation, and we can use a linear regression, which is accurate in the vicinity of the data when the production curve can be approximated as a first-order multivariable Taylor series. The Taylor series assumption holds because well tests and PI data are taken at the exact or similar well-head pressures as our predictions, and we use recent data when possible.

Elliptic models have convergence issues in Bayesian regression. A potential cause is the square root operation, which is undefined for negative arguments. If we want to use a linear, non-curved model, we must check whether curvature is present in the data. Since we cannot fit an elliptic model, we add a quadratic  $P_{\text{wh}}^2$  term to the linear model and compare the deviance information criterion (DIC), a goodness of fit measure for Bayesian models. Raw DIC results are shown in Table 3.

	Quadratic	Linear
Mean deviance	20805	20883
Penalty	144.3	122.1
Penalised deviance	20949	21005

Table 3: DIC comparison of two GLM candidates

The quadratic model has a lower mean deviance because more the extra parameter will always fit the data better. However, the penalty on the number of effective parameters is higher. The penalised deviances of both models are on the same order of magnitude, so the difference in fitting ability is very small. We therefore prefer the linear model for the interpretability of its parameters.

One of our extensions on the current Contact model is to incorporate time as a covariate. This allows for estimation of the production decline over time and can forecast into the future. The equation we fit is:

$$\dot{m} = \beta_0 + \beta_{\text{whp}} P_{\text{whp}} + \beta_{\text{date}} t + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2) \quad (15)$$

and the corresponding Normal likelihood function per data point (the full likelihood is the product of its components):

$$L(\dot{m} | \vec{\beta}, \sigma^2, P_{\text{whp}}, t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{-\dot{m} + \beta_0 + \beta_{\text{whp}} P_{\text{whp}} + \beta_{\text{date}} t}{2\sigma^2}} \quad (16)$$

where  $\dot{m}$  is the mass flow,  $P_{\text{who}}$  is a specified well-head pressure,  $t$  is a specified number of days after a baseline, and  $\varepsilon$  is a Normally distributed error of unknown variance. This form has several benefits over the other models considered:

1. Coefficients are interpretable as rates of change, rather than theoretical maximum limits in the elliptical model.
2. A simpler model with fewer parameters is less likely to over-fit to the data.
3. There are no root or power operations. Sampling is significantly faster (5x).

This model assumes the trend in the relationship is linear with time, the same assumption currently made by CEL in their spreadsheets. We also assume a stationary distribution of independent errors  $\sigma^2$ , which are a product of measurement errors and flow variance. We derive a prior for the measurement error from CEL, and sample from flow variance when we make predictions. In cases where  $P_{\text{whp}}$  is unavailable, such as from the PI database, we drop the  $\beta_{\text{whp}} P_{\text{whp}}$  term, leading to the assumption that  $P_{\text{whp}}$  is constant. This assumption is true as long as there is no change in back-pressure in the network from a configuration change, and the condition of the well is steady between the measurements and the time of prediction.

We sample the regression parameters to estimate mass flow decline over time  $\beta_{\text{date}}$  and flow variance.

We use a hierarchical Bayes structure to set priors on the regression coefficients  $\vec{\beta}$ ,  $\tau = 1/\sigma^2$ , where  $\tau$  is precision and is often used because it makes the analytic calculations simpler. Rather than making every individual parameter non-informative, we use the assumption that parameters between wells come from the same (unknown) distribution:

$$\beta_i \sim N(\mu_{\beta_i}, \tau_{\beta_i}) \quad (17)$$

$$\tau \sim \gamma(\alpha_\tau, \beta_\tau) \quad (18)$$

where the hyper-parameters have an non-informative prior,  $f(\mu_{\beta_i}, \tau_{\beta_i}, \alpha_\tau, \beta_\tau) \propto 1$ .

This is physically motivated because:

1. For  $\beta_{\text{date}}$ , pressure loss over time affects multiple wells.
2. For  $\beta_0$  and  $\beta_{\text{whp}}$ , wells in a field may have similar production curves.

This is not the same as saying the parameters between the wells are identical. Instead, it fits a distribution of well production curves from which each well is observed. Adding this bias gives a reduction in variance for wells with insufficient or no data. We instead make an educated imputation that their production curve is similar to the wells we have data for, rather than having absolutely no imputation at all.

Verification of our regression model includes plots of observed mass flows against fitted mass flows, and a standardised residual plot using residuals:

$$\varepsilon_{\text{std}} = \frac{\hat{\dot{m}} - \dot{m}}{\text{sd}(\hat{\dot{m}})} \quad (19)$$

where  $\hat{\dot{m}}$  is sampled from the Bayesian regression.

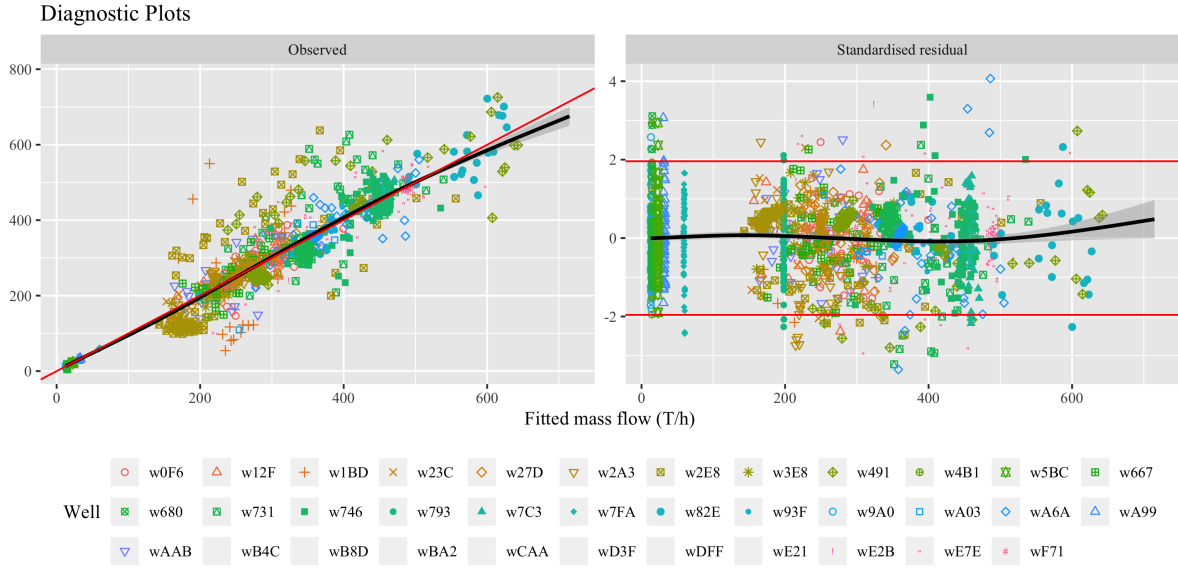


Figure 8: Diagnostic plots with the observed response against fitted values, and residuals against fitted response. These diagnostics suggest a good fit to the data.

A well-fitting regression shows a linear correlation between the fitted and observed values. If we believe the residuals are Normally distributed, the standardised residual plot will have a standard Normal distribution for all the fitted values. We present diagnostic plots in Figure 8. There is good correlation in the Observed  $\sim$  Fitted plot and standardised residuals are roughly symmetrically distributed.

### 7.1.3 Prediction

With posterior production curves fitted for most wells, e.g. Figure 10, we can estimate the mass flows at a given well-head pressure and date, specified in the configuration file. We also assign a measured enthalpy to the well flows, or apply a hierarchical posterior to any missing enthalpy values. Wells without production curves are imputed using a time-series on the PI data – see Figure 21 for examples of the time series regression.

Figure 9 shows the modelled connectivity between wells, flash plants and generators. Dummy generators have been added so that a flash plants can send one type of flow (e.g. intermediate pressure steam) to one generator, and another (e.g. water) to a second generator. This diagram is useful for showing the overall structure of the conceptual model, but can also be used to verify that the model configuration has been specified correctly in the Excel interface.

### 7.1.4 Flash Plant Flows

Well-to-flash-plant assignment is specified in the configuration spreadsheet. Although most well/flash-plant relationships are fixed by the physical pipes, there is a subset of wells that can swing between three flash plants, which have steam flow limits. The decisions for which wells flow to which flash plants can be optimised with a mixed-integer linear program [REF], chosen manually, or chosen by any arbitrary method.

To calculate mass flows and flow enthalpies entering a facility (both flash plants and generators), we use equations 2 and 3. During preprocessing, we add a dummy node to all  $I$  to



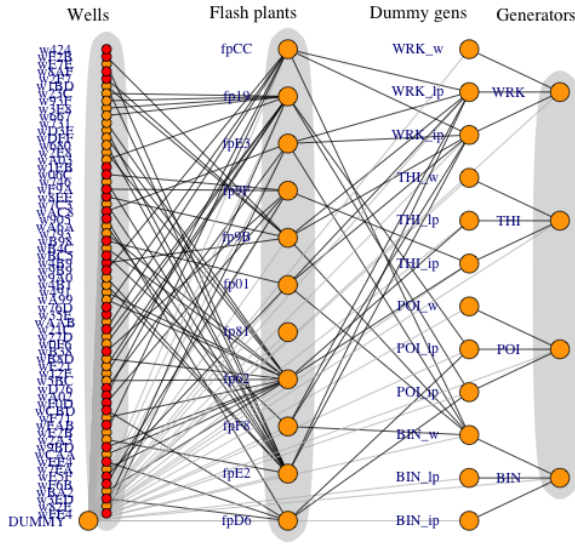


Figure 9: Full network diagram, with flows from left to right. Red wells indicate forecasts have been filled in from the PI data without a production curve, and dummy arcs are in grey.

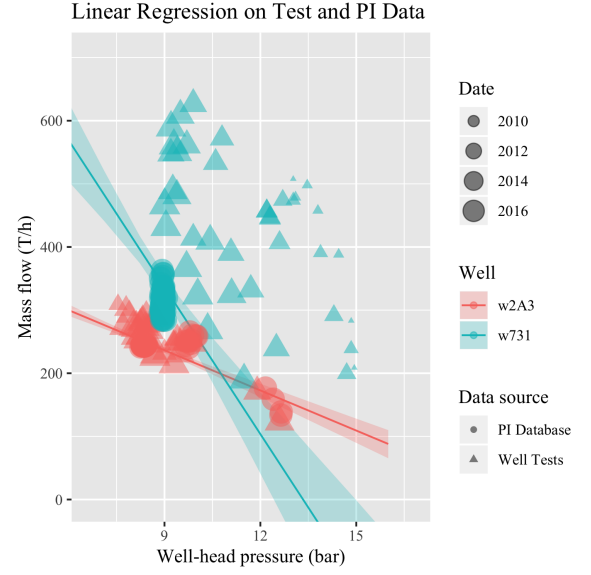


Figure 10: Two fitted production curves, forecasted one day in the future. We expect it to fit closely to the most recent data. The shaded region shows the 95% credible intervals.

ensure that  $I(j)$  is not an empty set; otherwise, JAGS throws an index error. Despite using an R-like syntax, JAGS only accepts integer indices, so generating  $I(j)$  is why facilities need to be re-identified by a unique integer rather than a string.

### 7.1.5 Generator Flows and Power Conversions

During pre-processing, three dummies per generator placed in the network before the actual aggregated power is calculated. In Figure 9, there is a dummy for each type of fluid flow: intermediate pressure (IP), low pressure (LP), or water (W). Whether these are actually used depends on the configuration inputs. For instance, the binary *BIN* plant will only ever use the *BIN\_w* dummy because in reality, it is a waste to send steam to the low-efficiency binary generator.

Dummy generators flow into their respective generator. Contact Energy calculates power output as a function of the bulk mass flow by Equation 6. These efficiencies are given to us in units of T/h/MW. Our uncertainty in the conversion factor is  $\pm 5\%$ , which we interpret as  $\eta \sim \text{Unif}(0.95\bar{\eta}, 1.05\bar{\eta})$ , which holds for small ( $< 10\%$ ) percentages.

## 7.2 Monitoring

Monitors are how JAGS returns posterior samples to the user. There are three types of monitors:

1. Mean/variance, showing the current mean/variance of a parameter up to and including a sample. We are not interested in mean and variance because the same information can be extracted from the trace.
2. DIC (deviance information criterion), to evaluate goodness of fit of the model to the data. Although DIC can play a role in validating the model structure during development, DIC

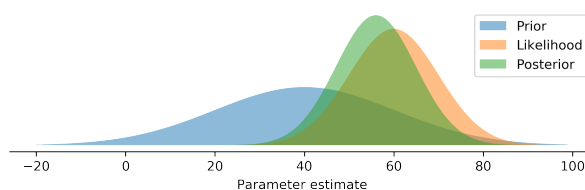


Figure 11: The prior and likelihood functions multiply to give a posterior parameter distribution. The posterior is what we are interested in after observing the data. The uncertainty in the parameter is represented by the width of its distribution; conversely, the precision can be interpreted as the height of its peak because densities integrate to one.

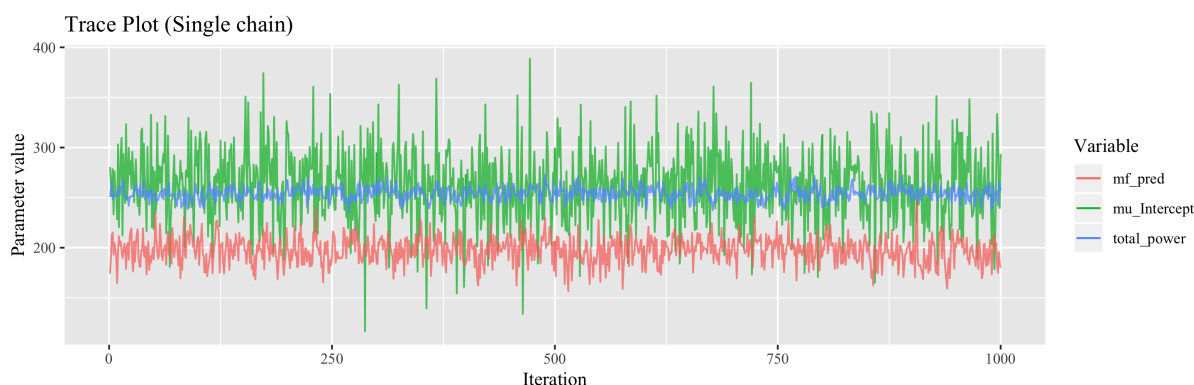


Figure 12: Example trace plots displaying normal behaviour. The sampler appears to have reached its equilibrium distribution with no trend.

is also not a permanent part of our model once it has been validated. Also, graphical goodness-of-fit techniques are preferred because an informed user can identify specific issues such as erroneous data that would not be possible with a single value.

3. Trace, showing every sample of a parameter. We use these to sample from the stationary distribution.

The first monitor we set is on the well regression, predicting for a range of covariates to build a picture of the production curves. Examples of these curves are shown in Figure 10.

Next, we monitor mass flows at each facility and their probability densities. An example of its interpretation is given in Figure 11. These estimates for uncertainty are the strength of Bayesian inference; we can sample from the posteriors at the wells and propagate our uncertainty through the network to find the uncertainties at other locations.

### 7.3 Diagnostics

One of the difficulties with MCMC approximations is they often require a burn-in (warm-up) period before settling into the stationary distribution of the Markov chain. Only the stationary distribution corresponds to the joint distribution we are interested in. 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.

Poor convergence and mixing is represented by a strong trend at the beginning of the trace

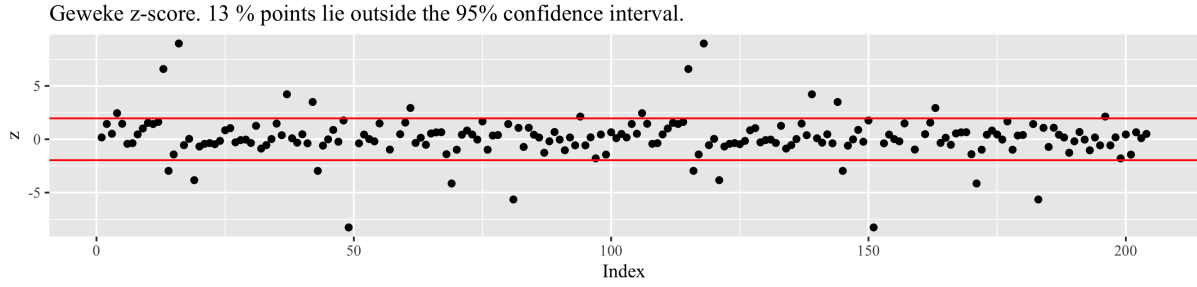


Figure 13: More than 5% of z-scores outside the confidence interval indicates the chains have not converged and are not long enough or contain burn-in.

plot. This is not present in Figure 12, but visual analysis of trace plots for every parameter. JAGS provides other diagnostic tests in the CODA package. There are two main tests to confirm this:

### 7.3.1 Geweke's Convergence

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 if the sample is drawn from a stationary distribution, indicating the burn-in period has been successfully excluded. Geweke's statistic has a T-distribution using the following T-test statistic:

$$T = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n} + \frac{s_2^2}{m}}} \quad (20)$$

where  $\bar{X}_1$  and  $\bar{X}_2$  are the sample means of the first 10% and last 50% of the samples,  $s^2$  are their corresponding standard variances, and  $n$  and  $m$  are the number of samples in the two groups. The results of Geweke's test for 100 randomly selected parameters are shown in Figure 13.

Most of the parameters pass Geweke's test with a z-score (a Normal approximation of the T-statistic) less than 1.96 for a 95% confidence interval. However, the changes in the power output traces were significant, so a short burn-in of 200 iterations was introduced.

### 7.3.2 Gelman's Potential Scale Reduction Factor

Gelman's test gives the potential scale reduction factor for each parameter. This requires at least two parallel chains using independent random variates (JAGS uses the Wichmann-Hill, Marsaglia-Multicarry, Super-Duper and Mersenne-Twister pseudorandom generators for the first four chains to ensure they are independent), 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 variance-inflated and their credible intervals may be too large. [REF]

Executing Gelman's test on all monitored parameters runs into issues with an internal Cholesky matrix factorisation of an ill-conditioned matrix. Testing a smaller selection yields Table 4.

Some of the upper CIs are slightly greater than one, but not significantly. We can expect their credible intervals and standard errors are approximately the correct width. Running the simulation for longer will shrink the CI, but for how long is a balance between computational

Increase  
sam-  
ple  
size  
for  
final  
run

	Point est.	Upper C.I.
mf_pred[8]	1.00	1.00
mf_pred[9]	1.00	1.02
beta_date[9]	1.01	1.02
mu_beta_whp	1.00	1.02
mu_beta_date	1.01	1.01
mu_Intercept	1.00	1.00
total_power	1.00	1.00

Table 4: Select potential scale reduction factors from Gelman’s diagnostic test.

resources and the need for precision – large PSRFs are acceptable if they are in components of the network that do not affect parameters of interest.

## 8 Results

Contact Energy has provided wellbore test data up to early 2018 and PI logs up to the end of December 31. They also supplied an example calculation for December 1st. To test the forecasting ability of our model, we discarded all data after November 30th, 2017, and compared predicted mass flows with measured flows for December 1st. The traces were analysed in R. Because there are so many facilities and parameters to monitor, the plots in this section may only include a subset of facilities that represent the different outcomes present in the simulation.

### 8.1 Well Production Curves

We compare production curves before and after including PI data in figures 14 and 15. PI data has changed the regression for w2A3 slightly, but in the proximity of the data (where we might want to make predictions), the regression is almost exactly the same.

The regression for w731 has adjusted to fit the PI data more closely in the well-head pressures of the PI data. Therefore, we know that the PI data does not agree perfectly with the well test data – however, we trust the PI data for our predictions because it is the most recent. A consequence of this disagreement is the parameters estimates become more uncertain – in regions away from the PI data, the width of the 95% credible interval has roughly doubled. We conclude that although PI data is up-to-date, it can decrease the precision of the production curve parameters. Since PI data is taken from  $n$  consecutive days, it is similar to weighting one point  $n$  times a single well test instance and it gains disproportionate influence over the regression parameters.

### 8.2 Well Mass Flows

Figure 16 presents the posterior densities for mass flow at selected wells. We can see that some wells have more variation than others. Variance can either come from the regression model or the data, with issues including:

1. Lack of data to fit the regression.
2. Data is for pressures and times far away from the values used for prediction; e.g. very old data that is no longer relevant.
3. Mass flow is non-linear with pressure and time (i.e. bad fit).

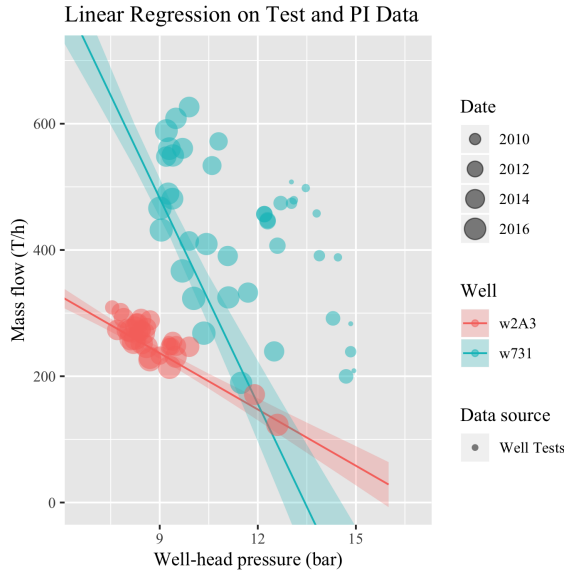


Figure 14: Fitted production curves with just well test data. Shaded regions are 95% credible intervals.

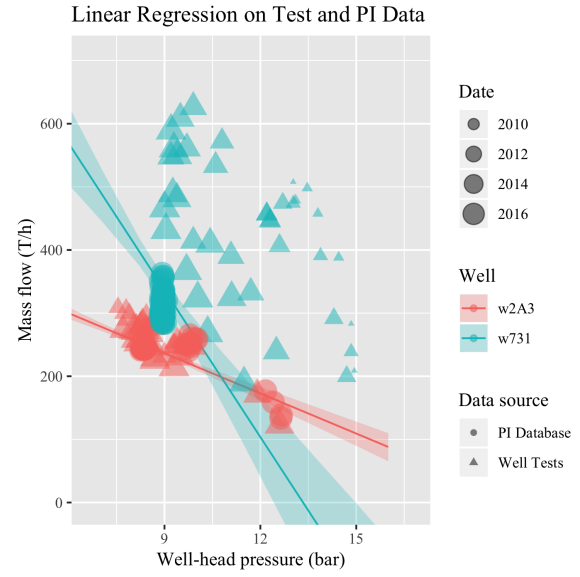


Figure 15: Production curves after including PI data. We expect this to make better predictions near the most recent (PI) data.

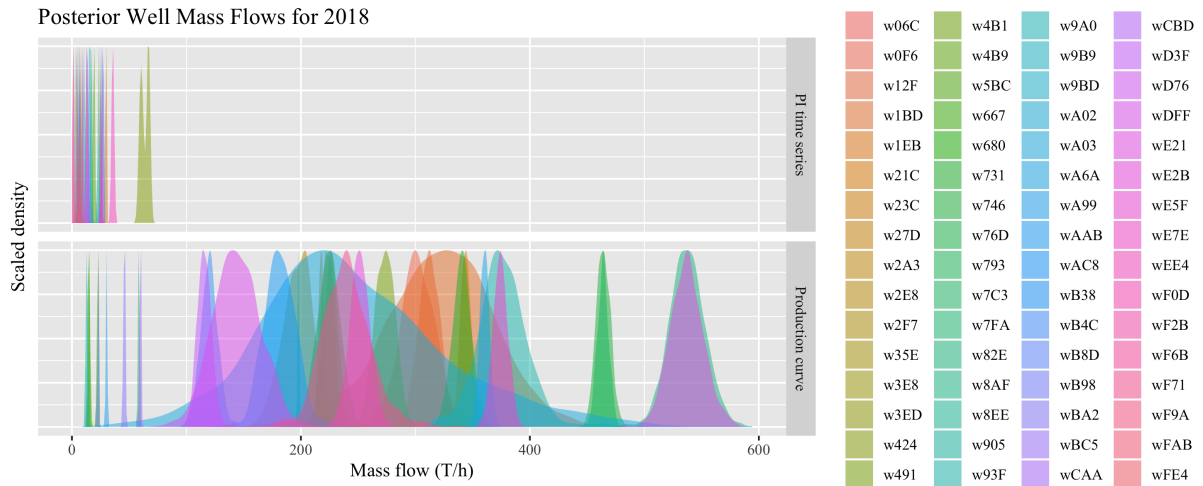


Figure 16: Posterior mass flows, divided into wells with production curves and wells with a simple time-series (see Figure 21 for time-series examples)

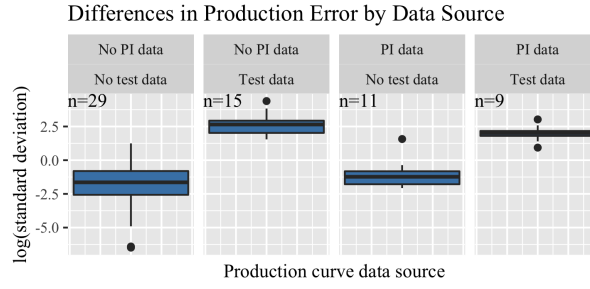


Figure 17: Facets of the data sources for the production curve. No data (far left) means there was no curve – these mass flows were predicted by time series on PI mass-flow data.  $n$  is the number of wells, and standard deviation is calculated as  $\text{sd}(\hat{m}_j)$  for each well.

	Well tests	PI data
Variance	Low	High
Period	Long-term, 2002-2017	Very recent, since the last change in configuration
Effective for	Short-term time-series prediction	Production curve and parameter estimation

Table 5: Differences in the two data sources

We do not see any issues with non-linearity in the diagnostic plots (Figure 8), and we know from previous DIC comparisons that higher order terms do not significantly aid the regression. Therefore, we will next look at the data. We can compare the performance of well predictions by comparing well standard deviations by their data sources.

Figure 17 shows that the most precise wells are predicted by time series, rather than production curve. This is not surprising – by predicting from a production curve of mass flow by well-head pressure and time, we introduce intermediate errors. The other three facets with production curves suggest that test data is the source of variance in the regression. We class the data sources in Table 5.

### 8.3 Individual Well Declines

well	Mean	Lower 2.5%	Upper 97.5%	n
w1BD	0.021	-0.008	0.049	28
w2A3	-0.016	-0.021	-0.011	95
w5BC	-0.004	-0.017	0.009	61
w93F	-0.116	-0.339	0.056	5
wA03	-0.068	-0.236	0.083	6
wE2B	-0.062	-0.086	-0.039	31

Table 6: Credible intervals for  $\beta_{\text{date}}$  in units T/h/d.  $n$  is the number of test data points rather than the total points include PI data, because all PI data is from a single month, so cannot estimate the  $\beta_{\text{date}}$  parameter on its own. Full table in Table 7.

Table 6 presents credible intervals for decline at selected wells. The rates are mostly negative (in decline), but some rates are not statistically significant, such as w1BD and w5BC.

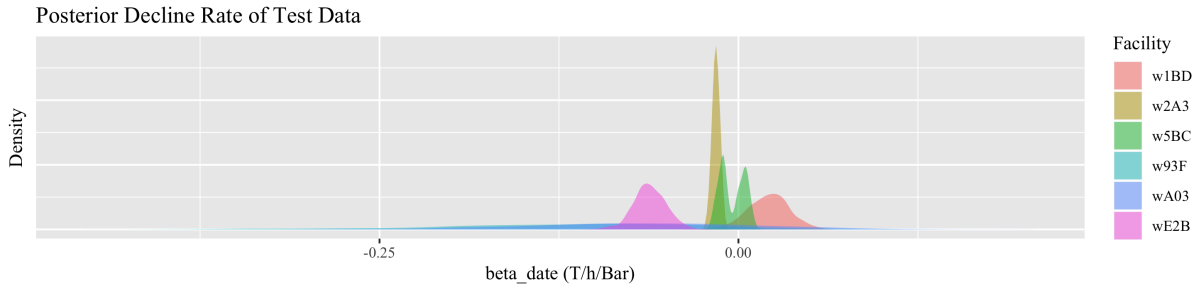


Figure 18: Posterior decline rate of selected liquid wells.

There is varying precision in the decline – w2A3 is an example of a well with good precision in the decline rate. This is because there have been 64 recorded values with a range of covariate values, giving the parameter estimates good support.

Inspection of wells with poor precision in the regression parameters show that there is insufficient well test data available. For instance, w93F has five points; this is not enough to estimate variance in a model with three parameters. When precision is low, we cannot make strong claims about the decline rate for that well. However, we can say that at a fixed well-head pressure, mass output for well w2A3 is declining between -0.011 and -0.016 T/h per day.

## 8.4 Down-flow Results

The results from the previous section propagate through the network to the flash plants and generators. We are most interested in the flash plants, where we have an example spreadsheet of flows for a particular day and constraints on steam flows .

### 8.4.1 Flash Plant Mass Flow Verification

After making predictions for the wells, we obtain mass flows entering the flash plants. This is split up into intermediate pressure, low pressure and water flows. We verify our posterior beliefs for these flows against an example calculation from CEL. The full comparison is in the appendices, Figure 22. There are three main observations:

1. Our mass flows tend to under-estimate.
2. Four flash plants have uncertain enthalpies but very precise mass flows (precision being unrelated to accuracy). The wells feeding these plants do not have enthalpy, which is why the estimate for  $h$  is disperse. They also use time-series estimates on the mass flow without a production curve, leading to high precision.
3. fpE2, fp62 and fp19, are fed by wells with production curves. However, fp62 seems to under-estimate total  $\dot{m}$  and therefore all the other components of flow.

To investigate the difference between fp62 and fpE2/fp19, we look at the data methods shown in Appendix 10.5. Over half of the wells feeding fp62 have no test data, whereas most wells in fpE2 and fp19 have substantial amounts of test data data ( $n_{\text{test}} > 10$ ). However, this is not strong evidence and would need to be investigated further.



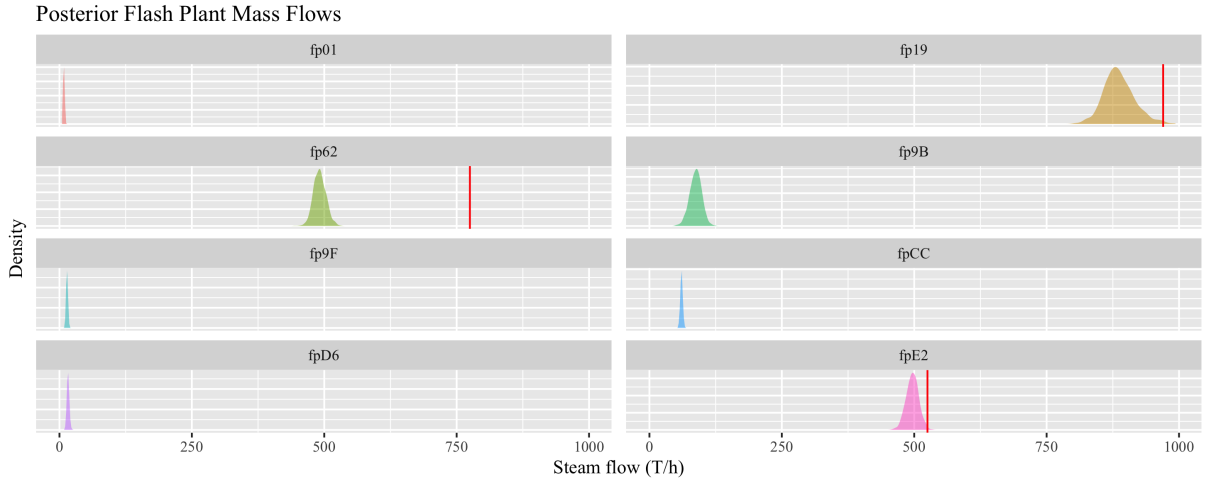


Figure 19: Posterior density plot of the steam flows through the flash plants. Known steam limits are shown by a vertical red line. Two flash plants with significant missing well data are not shown.

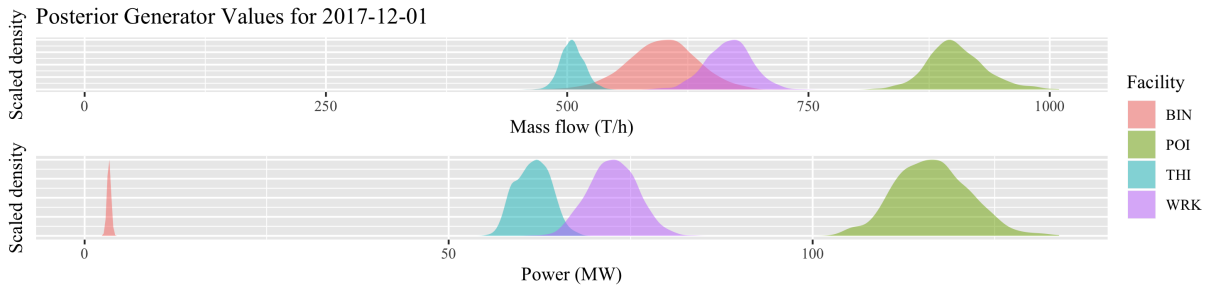


Figure 20: Estimates for mass flow and power generation, including supplied uncertainties in the steam/power conversion.

#### 8.4.2 Flash Plant Constraints

Contact Energy defines constraints on the flows in the network. We can estimate the probabilities of each constraint being violated by the proportion of the trace exceeding the constraint value. However, more nuanced comparisons can be made with the density plot in Figure 19.

Some of the flows are under-estimated because data is not available for all wells. However, in Figure 19 we can see that all of the flash plants are likely operating below their limits in the December 1st configuration. However, one flash plant (fpE2, one of the flash plants where wells can be redirected) is more likely to violate its constraints than the others, suggesting that one or more wells should be redirected to another flash plant.

#### 8.4.3 Generator Flows and Power

We now reach the generators. Again, we only receive mass flows from wells we have data for, so the sums at the generators presented in Figure 20 are incomplete. Nonetheless, we show that our model can propagate uncertainty through to the generators' power estimates, and that they are relatively disperse.



## 9 Conclusions

In this report, we show a hierarchical regression model and a directed network with uncertainty can be implemented in a single Bayesian model to estimate the true state of a geothermal surface network. Our model includes estimates of errors in both the steady-state properties of the network such as flows, and in the regression parameters such as production decline in the wells.

Our model addresses two needs:

1. Produce production curves with uncertainty so the operator can track well declines and see how production changes with well-head pressure.
2. Make forecasts

We also compare our posterior beliefs of the network against constraints set by the field's operators, and find that our simulation is able to probabilistically verify whether those constraints are violated in a given network configuration, as long as we have sufficient test data or PI data about the wells feeding a facility.

Despite CEL and Grant & Bixley [REF] using an elliptic or otherwise curved model in their regression, we do not find significant evidence that the data would be better suited by a non-linear model. We are also able to incorporate all historical test data into a regression model for the production curves, giving good parameter estimates for a temporal decline model and better robustness compared to CEL's current methods. Incorporating PI data into the regression improves forecasting accuracy because the PI data is recent, but PI data sometimes decreases precision in the regression parameters representing change in mass flow with change in pressure or time and may be undesirable if the parameters are more important than forecasts.

When the goal is to make mass flow forecasts, a production curve can give similar results to time-series analysis. However, using the production curve also creates extra error if the well-head pressure is incorrect, or if there is insufficient data recorded at the operational well-head pressure. We recommend that time-series analyses include PI database flow data whenever possible.

To operate the model, we provide a configuration spreadsheet where the well/flash-plant and flash-plant/generator connectivity is specified, either as the output of an external third-party optimisation algorithm or as part of scenario analysis. This spreadsheet is also where constants such as enthalpy are specified.

## 10 Further Development

There are many opportunities to expand on the utility of this work.

### 10.1 Direct Data Integration

In this work, all data was obtained through Excel workbooks, even though some of it was originally stored in a PI system. Direct integration with the PI database would have benefits:

1. Automatic updates with daily data can track changes in trend as they happen within 24

should I provide a screenshot?

hours. We aimed to create a system that could deliver results within minutes instead of the considerable it takes to operate the Excel sheet, but this is only relevant if it is not delayed by the data source.

2. When available, data from the PI system is superior because it is automatically recorded once a day, and is more recent. We had to remove some data because of inconsistencies with how variables were recorded, and issues with the Excel output formatting. We also did not understand some of the variable codings. Access to a consistent and documented PI database would allow us to incorporate more variables.

## 10.2 Data Integration With Other Models

Enthalpies are important because they affect the mass proportions exiting the flash plant. We use the most recent recorded enthalpy for each well, and do a probabilistic imputation for wells with unknown enthalpies. Taking simulation results from a wellbore simulator such as TOUGH2 would avoid the uncertainty associated with imputation, and may also give us more up-to-date enthalpies.

## 10.3 Time Series

In our implementation of well declines, we treat measurements independently with respect to time. However, measurements are almost never truly independent and are often auto-correlated with previous measurements. Extra pre-processing to include auto-regression and differencing in the JAGS model is a common technique although it is difficult with irregularly spaced multivariate time series. Implementing non-independent time-series analysis will result in reduced standard errors.

We would also be interested to compare the performance of a standalone AR(1) forecast without any production curves. The downsides include having no estimate for the parameterisation of mass flow, and not having flow meters for all the wells. A proof of concept may provide motivation to install more flow meters.

## 10.4 Prior Specification

More prior specifications for parameters will add extra educated bias to our model, helping with variance reduction. Priors can be added to the hierarchical parameters on the regression coefficients, which are currently non-informative. For instance, if we knew an overall production decline rate and variance in decline across the field.

## 10.5 Hierarchical Model Resolution

Partitioning of components by manufacturer's specifications or age/generation will improve the hierarchical model. We currently treat all wells or flash plants as coming from the same family of facilities. However, we know the Wairakei geothermal field was built in a series of stages and that some facilities are more similar than others. Each family of facility should have its own hierarchical model, reducing variance in the hyper-parameters.

# Appendices

## A Time Series

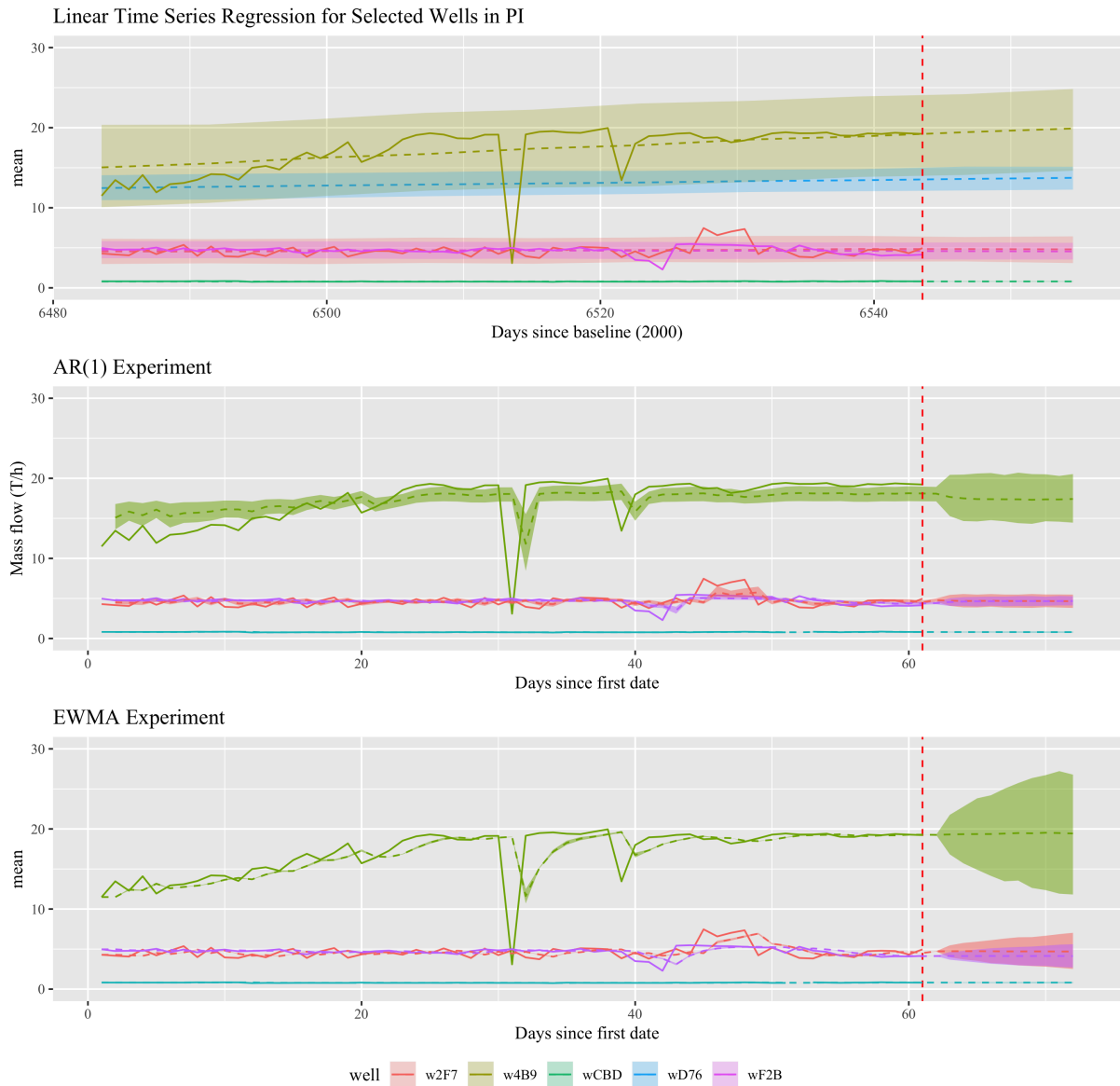


Figure 21: Time-series techniques on data from the PI database. We use the linear time series regression (top) for its robustness to systematic changes in operation, which cause the other regressions to become unstable. Auto-regression appears to be preferable to exponential-weighted moving averages.

## B Verification

Comparison Between Predicted FP Flows and Sample Data

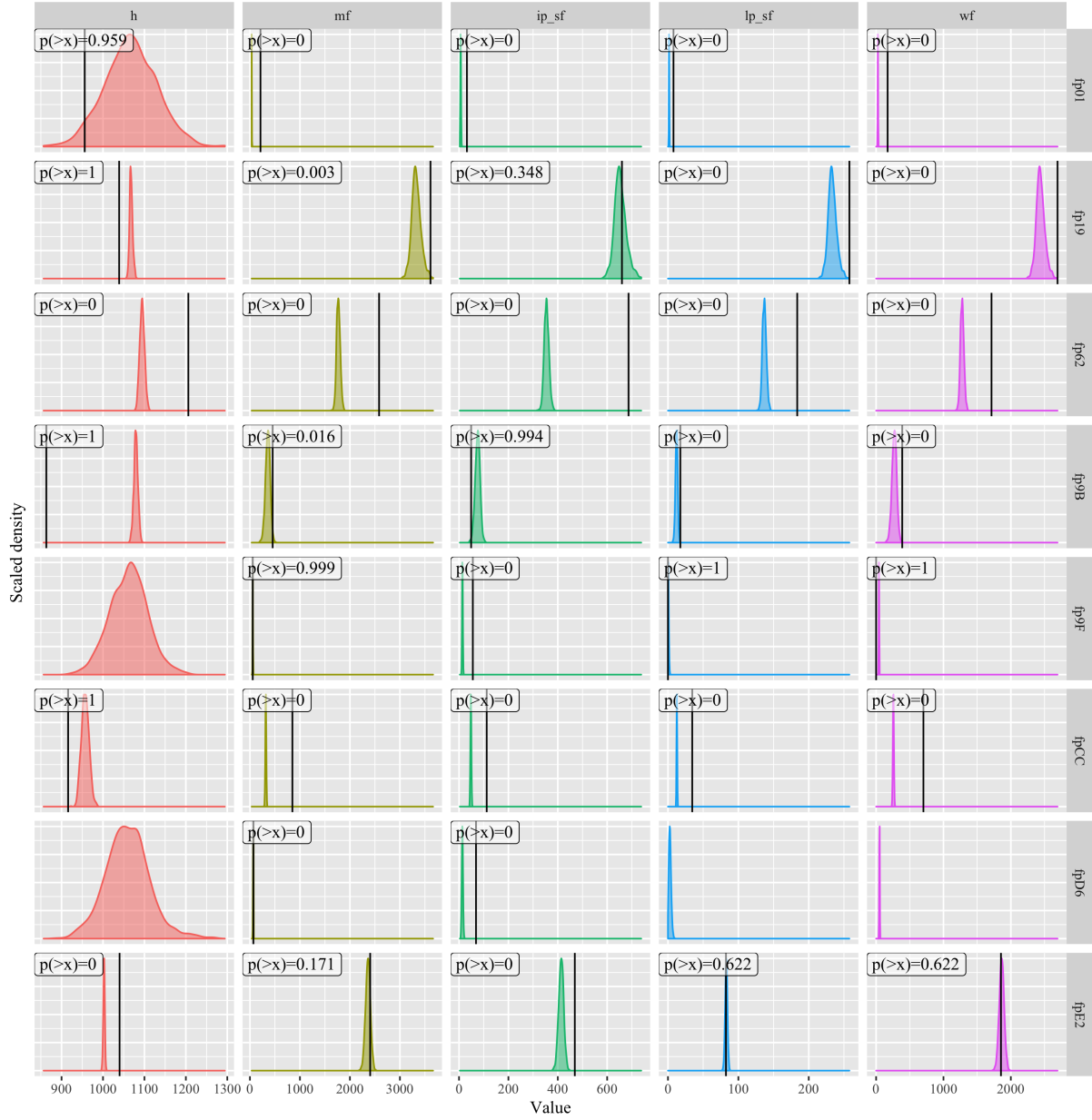


Figure 22: Verification of predicted flows with supplied calculations shows a systematic underestimation for most quantities, if we assume CEL's figures as the ground truth. Densities are the model predictions and black lines are the given figures from CEL (estimated by CEL, not direct from the PI loggers). Two flash plants with unavailable well data have been removed from the comparison.

## C All Well Declines

well	Mean	Lower 2.5%	Upper 97.5%	n
w0F6	-0.046	-0.067	-0.025	33
w12F	-0.019	-0.036	-0.001	32
w1BD	0.021	-0.008	0.049	28
w23C	-0.024	-0.032	-0.015	39
w27D	-0.024	-0.031	-0.017	37
w2A3	-0.016	-0.021	-0.011	95
w2E8	-0.202	-0.254	-0.150	82
w3E8	-0.067	-0.081	-0.054	38
w491	-0.300	-0.352	-0.246	92
w4B1	0.003	-0.002	0.007	61
w5BC	-0.004	-0.017	0.009	61
w667	-0.063	-0.082	-0.045	37
w680	-0.007	-0.037	0.037	58
w731	-0.158	-0.193	-0.120	102
w746	0.006	-0.013	0.025	91
w793	-0.008	-0.018	0.006	61
w7C3	-0.016	-0.152	0.087	61
w7FA	-0.029	-0.036	-0.022	61
w82E	-0.145	-0.183	-0.110	36
w93F	-0.116	-0.339	0.056	5
w9A0	-0.030	-0.049	-0.012	61
wA03	-0.068	-0.236	0.083	6
wA6A	-0.152	-0.174	-0.131	91
wA99	-0.047	-0.058	-0.039	61
wAAB	-0.054	-0.072	-0.035	37
wB4C	-0.142	-0.195	-0.091	86
wB8D	-0.026	-0.031	-0.022	61
wBA2	0.002	-0.020	0.015	61
wCAA	-0.012	-0.023	-0.001	61
wD3F	0.001	-0.005	0.006	44
wDFF	-0.209	-0.287	-0.131	42
wE21	-0.125	-0.157	-0.093	29
wE2B	-0.062	-0.086	-0.039	31
wE7E	-0.061	-0.072	-0.049	113
wF71	-0.147	-0.308	0.001	7

Table 7: Credible intervals for  $\beta_{\text{date}}$  in units T/h/d.

## 11 Flash Plant Data Sources

	facility	mean	sd	n_test	n_pi	use.test	use.pi	production.curve
1	w1BD	321.70	46.16	28	0	Test data	No PI data	Production curve
2	wE7E	374.24	6.39	52	61	Test data	PI data	Production curve
3	w746	464.16	6.07	34	57	Test data	PI data	Production curve
4	wD3F	115.56	5.94	44	0	Test data	No PI data	Production curve
5	w7C3	463.84	4.81	0	61	No test data	PI data	Production curve
6	wA6A	360.34	4.04	34	57	Test data	PI data	Production curve
7	w2A3	217.84	2.54	34	61	Test data	PI data	Production curve
8	w680	14.77	0.69	0	58	No test data	PI data	Production curve
9	w9A0	12.95	0.42	0	61	No test data	PI data	Production curve
10	w4B1	14.93	0.13	0	61	No test data	PI data	Production curve

Table 8: Data methods feeding flash plant fpE2

	facility	mean	sd	n_test	n_pi	use.test	use.pi	production.curve
1	wF71	239.82	22.51	7	0	Test data	No PI data	Production curve
2	wDFF	144.01	20.60	27	15	Test data	PI data	Production curve
3	w2E8	197.52	13.25	32	50	Test data	PI data	Production curve
4	w491	273.42	8.75	31	61	Test data	PI data	Production curve
5	w731	340.68	8.13	41	61	Test data	PI data	Production curve
6	wB4C	119.93	7.52	25	61	Test data	PI data	Production curve
7	w9BD	15.61	0.61	0	0	No test data	No PI data	Time series
8	wBA2	45.83	0.46	0	61	No test data	PI data	Production curve
9	w793	197.89	0.35	0	61	No test data	PI data	Production curve
10	wCAA	59.86	0.29	0	61	No test data	PI data	Production curve
11	w5BC	22.67	0.29	0	61	No test data	PI data	Production curve
12	wA99	30.25	0.17	0	61	No test data	PI data	Production curve
13	w7FA	58.46	0.17	0	61	No test data	PI data	Production curve
14	wB8D	22.96	0.14	0	61	No test data	PI data	Production curve

Table 9: Data methods feeding flash plant fp62

	facility	mean	sd	n_test	n_pi	use.test	use.pi	production.curve
1	wA03	246.79	80.35	6	0	Test data	No PI data	Production curve
2	w93F	375.95	25.01	5	0	Test data	No PI data	Production curve
3	w82E	537.40	15.67	36	0	Test data	No PI data	Production curve
4	w0F6	301.68	14.92	33	0	Test data	No PI data	Production curve
5	wE21	537.05	14.53	29	0	Test data	No PI data	Production curve
6	w667	224.18	10.47	37	0	Test data	No PI data	Production curve
7	w12F	311.74	9.68	32	0	Test data	No PI data	Production curve
8	w3E8	225.85	6.45	38	0	Test data	No PI data	Production curve
9	w23C	223.51	5.54	39	0	Test data	No PI data	Production curve
10	w27D	343.96	4.69	37	0	Test data	No PI data	Production curve

Table 10: Data methods feeding flash plant fp19

## 12 JAGS Model

The following model is loaded as a text string into the RJAGS package. Not listed is the data, pre-processing code and post-processing of the results. This code also includes two extra time-series models that were not used for outputs. (Pre-/post-processing source code available on the author's Github)

```

1
2 data {
3   D <- dim(ts)
4 }
5 model {
6   #####
7   # fit individual regressions to liquid wells #
8   #####
9   for (i in 1:length(mf)) {
10    mu[i] <- Intercept[well_id[i]] + beta_whp[well_id[i]] * whp_c[i] + beta
      _date[well_id[i]] * date_numeric_c[i]
11    mf[i] ~ dnorm(mu[i], tau[well_id[i]])
12    mf_fit[i] ~ dnorm(mu[i], tau[well_id[i]])
13    # mf_fit[i] ~ dnorm(mu[i]*measurement_error_factor[i], tau[well_id[i]])
14    # measurement_error_factor[i] ~ dunif(0.9, 1.1)
15  }
16  # fit regression to dry wells
17  for (i in 1:length(mf_dry)) {
18    mu_dry[i] <- Intercept[well_id_dry[i]] + beta_date[well_id_dry[i]] *
      date_numeric_dry_c[i]
19    mf_dry[i] ~ dnorm(mu_dry[i], tau[well_id_dry[i]])
20    mf_dry_fit[i] ~ dnorm(mu_dry[i], tau[well_id_dry[i]])
21    # measurement_error_factor_dry[i] ~ dunif(0.9, 1.1)
22  }
23  for (j in dry_well_ids) {
24    Intercept[j] ~ dnorm(0, 1e-12)
25    beta_date[j] ~ dnorm(0, 1e-12)
26    tau[j] ~ dgamma(1e-12, 1e-12)
27  }
28  # experimental AR1 model for dry wells
29  for (j in ar_well_ids) {
30    for (t in 2:D[1]) {
31      mu_ar[t,j] <- c_ar[j] + theta_ar[j]*ts_ar[t-1,j]
32      ts_ar[t,j] ~ dnorm(mu_ar[t,j], tau_ar[j]) T(0,)
33    }
34    theta_ar[j] ~ dnorm(0, 1e-12)
35    c_ar[j] ~ dnorm(0, 1e-12)
36    tau_ar[j] ~ dgamma(1e-12, 1e-12)
37  }
38  # experimental EMA model (don't use)
39  for (j in ar_well_ids) {
40    for (t in 2:D[1]) {
41      mu_ema[t,j] <- alpha*mu_ema[t-1,j] + (1-alpha)*ts_ema[t-1,j]
42      ts_ema[t,j] ~ dnorm(mu_ema[t,j], tau_ema[j]) T(0,)
43    }
44    mu_ema[1,j] <- ts_ema[1,j]
45    theta_ema[j] ~ dnorm(0, 1e-12)
46    c_ema[j] ~ dnorm(0, 1e-12)
47    tau_ema[j] ~ dgamma(1e-12, 1e-12)
48  }

```

```

49  alpha ~ dbeta(0.5, 0.5)
50
51  # HIERARCHICAL
52  # fills in for any missing wells
53  for (j in liq_well_ids) {
54    Intercept[j] ~ dnorm(mu_Intercept, tau_Intercept)
55    beta_whp[j] ~ dnorm(mu_beta_whp, tau_beta_whp)
56    beta_date[j] ~ dnorm(mu_beta_date, tau_beta_date)
57    tau[j] ~ dgamma(1e-12, 1e-12)
58    sd[j] <- 1/max(sqrt(tau[j]), 1e-12)
59  }
60
61  # fill in any missing data
62  for (i in 1:length(mf)) {
63    date_numeric_c[i] ~ dnorm(mu_date_numeric, tau_date_numeric)
64  }
65  mu_date_numeric ~ dnorm(0, 1e-12)
66  tau_date_numeric ~ dnorm(1e-12, 1e-12)
67
68  # set hyperparameters
69  mu_Intercept ~ dnorm(0, 1e-12)
70  mu_beta_whp ~ dnorm(0, 1e-12)
71  mu_beta_date ~ dnorm(0, 1e-12)
72  tau_Intercept ~ dgamma(1e-12, 1e-12)
73  tau_beta_whp ~ dgamma(1e-12, 1e-12)
74  tau_beta_date ~ dgamma(1e-12, 1e-12)
75
76  #####
77  # production curve for verification #
78  #####
79  for (i in 1:length(whp_prod)) {
80    mu_prod[i] <- Intercept[well_id_prod[i]] + beta_whp[well_id_prod[i]] *
      whp_prod_c[i] + beta_date[well_id_prod[i]] * today_numeric_c
81    # mf_prod[i] ~ dnorm(mu_prod[i], tau[well_id_prod[i]])
82    mf_prod[i] <- mu_prod[i]
83  }
84  for (i in 1:length(date_numeric_ts)) {
85    mu_ts[i] <- Intercept[well_id_ts[i]] + beta_date[well_id_ts[i]] * date_
      numeric_ts[i]
86    mf_ts[i] ~ dnorm(mu_ts[i], tau[well_id_ts[i]])
87  }
88
89  #####
90  # simple model to fill in missing FP enthalpy constants #
91  #####
92  for (i in fp_ids) {
93    # missing fp constants
94    hf_ip[i] ~ dgamma(param[1], param[7])
95    hg_ip[i] ~ dgamma(param[2], param[8])
96    hfg_ip[i] ~ dgamma(param[3], param[9])
97    hf_lp[i] ~ dgamma(param[4], param[10])
98    hg_lp[i] ~ dgamma(param[5], param[11])
99    hfg_lp[i] ~ dgamma(param[6], param[12])
100  }
101  for (i in c(1, well_ids)) {
102    h[i] ~ dgamma(param[13], param[14])
103    whp_pred_c[i] ~ dnorm(param[15], param[16])

```



```

104   } # missing well constants
105   for (i in 1:16) { param[i] ~ dgamma(1e-12, 1e-12) } #
      uniform priors
106
107   #####
108   # make predictions (the stuff we want) #
109   #####
110   mf_pred[dummy] <- 0 # dummy well
111   ip_sf[dummy] <- 0
112   lp_sf[dummy] <- 0
113   wf[dummy] <- 0
114
115   # use production curve
116   for (j in liq_well_ids) {
117     mf_pred[j] <- max(Intercept[j] + beta_whp[j] * whp_pred_c[j] + beta_
      date[j] * today_numeric_c, 0)
118   }
119   # use naive TS reg
120   for (j in dry_well_ids) { #dry_no_ar_well_ids) {
121     mf_pred[j] <- max(Intercept[j] + beta_date[j] * today_numeric_c, 0)
122   }
123   # use AR(1)
124   # for (j in ar_well_ids) {
125   #   mf_pred[j] <- mu_ar[D[1], j]
126   # }
127
128   for (i in fp_ids) {
129     mf_pred[i] <- sum(mf_pred[m[i,1:n_inflows[i]]])
130     h[i] <- sum(mf_pred[m[i, 1:n_inflows[i]]] * h[m[i, 1:n_inflows[i]]]) /
      ifelse(mf_pred[i]!=0, mf_pred[i], 1)
131
132     ip_sf[i] <- min(max((h[i] - hf_ip[i]), 0) / hfg_ip[i], 1) * mf_pred[i]
133     lp_sf[i] <- min(max((min(hf_ip[i], h[i]) - hf_lp[i]), 0) / hfg_lp[i],
      1) * (mf_pred[i] - ip_sf[i])
134
135     total_sf[i] <- ip_sf[i] + lp_sf[i]
136     wf[i] <- mf_pred[i] - total_sf[i]
137   }
138   # dummy gens and actual gens
139   for (i in ip_gen_ids) { mf_pred[i] <- sum(ip_sf[m[i, 1:n_inflows[i]]]) }
140   for (i in lp_gen_ids) { mf_pred[i] <- sum(lp_sf[m[i, 1:n_inflows[i]]]) }
141   for (i in w_gen_ids) { mf_pred[i] <- sum(wf[m[i, 1:n_inflows[i]]]) }
142   for (i in gen_ids) {
143     mf_pred[i] <- sum(mf_pred[m[i,1:n_inflows[i]]])
144     power[i] <- mf_pred[i] / mu_factor[i]
145     mu_factor[i] ~ dunif(0.95*factor[i], 1.05*factor[i]) # uncertainty
      from email
146   }
147   total_power <- sum(power[gen_ids])
148 }

```