



DEPARTMENT OF COMPUTER SCIENCE

A Bayesian Inference Engine for UMIS Structured Data

Tom Jager

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree
of Master of Engineering in the Faculty of Engineering.

Monday 6th May, 2019

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of MEng in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Tom Jager, Monday 6th May, 2019

Contents

Executive Summary

A compulsory section, of at most 1 page

Supporting Technologies

Notation and Acronyms

An optional section, of roughly 1 or 2 pages

Any well written document will introduce notation and acronyms before their use, *even if* they are standard in some way: this ensures any reader can understand the resulting self-contained content.

Said introduction can exist within the dissertation itself, wherever that is appropriate. For an acronym, this is typically achieved at the first point of use via “Advanced Encryption Standard (AES)” or similar, noting the capitalisation of relevant letters. However, it can be useful to include an additional, dedicated list at the start of the dissertation; the advantage of doing so is that you cannot mistakenly use an acronym before defining it. A limited example is as follows:

CC	:	Concentration Coefficient
ERN	:	Entity Relationship Diagram
IE	:	Industrial Ecology
IOA	:	Input Output Assessment
LCA	:	Life Cycle Assessment
MC	:	Monte Carlo
MCMC	:	Markov Chain Monte Carlo
MFA	:	Material Flow Assessment
pdf	:	Probability density function
STAF	:	Stocks and Flows
STAFDB	:	Stocks and Flows Database
TC	:	Transfer Coefficient
UMIS	:	Unified Materials Information System
YSTAFDB	:	Yale Stocks and Flows Database

Acknowledgements

Chapter 1

Contextual Background

1.1 Industrial Ecology

Industrial ecology (IE) is an area of research focused around the flow of material, energy and money through a system. It can also be described as socio-economic metabolism (SEM) as it models the production, consumption and storage of resources by society. It primarily focuses on the relationship between human originating sectors (anthroposphere) and the natural ecosystem by modelling industrial infrastructures as their own subsystems that interact with their environment [?]. The International Society for Industrial Ecology uses the following definition by White, "the study of the flows of materials and energy in industrial and consumer activities, of the effects of these flows on the environment, and of the influences of economic, political, regulatory and social factors on the flow, use and transformation of resources" [?]. As such it is a multi-disciplinary field which seeks to account for material and energy data and use it to inform social and economic policy as well as business strategy. The primary motivator for Industrial Ecology is to encourage and ensure sustainable development. By most agreed definitions, this involves ensuring that the economical and societal growth occurs without hampering the ability of development in the future [?]. Therefore research in Industrial Ecology focuses on decoupling the relationship industrial development has on natural ecosystems. To do this, the entire life cycle of products and materials are analyzed in order to find areas for greater efficiency and reduced reliance on natural resources. Studies in Industrial Ecology can focus on identifying the amount of flow of specific materials from the anthroposphere into the environment [?], to assessing where new stocks of materials are accumulating [?], or finding energy and material "loops" which can be closed in order to reuse waste material and energy[?].

1.1.1 Industrial Ecology Methodology

There exist a variety of different methodologies to conduct studies in Industrial Ecology. The data resulting from IE studies therefore is usually in a format only suited for that methodology. The three most prevalent methods are Life Cycle Assessments (LCAs), Input-Output Analysis (IOA) and Material

Flow Analysis (MFA) [?].

Life Cycle Assessments follow the environmental impact of a product system throughout its life cycle [?]. It often involves compiling an inventory analysis where the life cycle is modelled as a system of processes with material and energy flowing between them. Inputs and outputs from each process are specified with special interest paid to flows into and from the environment. This is used to assess the impact of a product and provide information for corporate decision making in order to improve the efficiency of a product. Other motivators are in reducing a product's environmental impact. LCAs have been known to have sector wide impacts through industry collaboration [?]. Open data formats such as EcoSpold and ILCD as well as shared databases such as Ecoinvent [?], make this possible as different industry partners can combine research and also easily recreate results to ensure accuracy.

Input-Output Analyses takes an economic approach to IE and tracks the flow of money between entities in a geographic region. These can then be used to allocate environmental impact to these entities. As economic data on inter-sector flows are generally more granular than data on the movement of physical material, it can be useful to apply data from IOAs in studies that use other methodologies [?].

Material and Energy Flow Analyses maps the presence of a specific material(s) in a system, paying particular attention to where it accumulates in the form of stocks. It does this by modelling a system as a collection of processes and then accounting for flows between and the build up of stocks in them. This is useful for identifying where cycles can be created and enhanced in these systems in order to increase recycling and therefore reduce a system's dependence on its environment [?]. Static MFAs are where the scope of the analyses falls over a single specified timeframe, whilst dynamic MFAs use data about past and present quantities to estimate future impacts. This can highlight which resources may become scarce in the future and provide warnings about future environmental impacts.

1.2 Unifying Industrial Ecology

Whilst some studies have incorporated data from one methodology into another [?], there is need to provide a common framework for all varieties of industrial ecological data. In [?], Pauliuk et al performed a comprehensive analysis on the myriad of methodologies present in IE. They discovered that each methodology described the system using a shared structure, that of a bipartite directed graph. The shared common property between each system is that material was *transformed* in one process and then *distributed* in a subsequent process. Edges in this bipartite graph denote a flow of material from one a *distribution* node to a *transformation* node or vice versa.

1.2.1 STAFDB and UMIS

Over the past 20 years, the Graedal research group at Yale University have compiled Industrial Ecology data on over 100 materials on a variety of spacial and temporal scales. The data from these studies has been extracted and used to create the Yale Stocks and Flows Database (YSTAFDB) [?]. Ongoing work by Myers, Hoekman and Petard (to be published), is in developing a community driven database for this data called the Stocks and Flows Database (STAFDB). This is an improvement on YSTAFDB as it is

designed to be more user friendly and deals with the problem of divergent disaggregation.

Disaggregation is where a property of data (e.g a process that data is coming from or the material being described) is divided into components. In a broad example, data about cars could be disaggregated into electric and non-electric vehicles or large and small cars. If data from two studies with different techniques of disaggregation (or divergent disaggregation) on the same data were structured into the same system, data could be counted twice when performing analysis [?]. To prevent this, process and material data in STAFDB contains a parent field and an `is_separator` flag which serves to keep track of methods of disaggregation and prevent double counting.

Myers et al. has created UMIS, the Unified Materials Information System [?] as a data format to structure data contained in STAFDB. UMIS structures stocks and flows data that comprise IE studies into a format that is agnostic to IOA, MFA and LCA methodologies as a UMIS diagram. This diagram can displayed visually and is also machine readable which allows for greater automation when dealing with stocks and flows data.

UMIS is a step in shifting industrial ecology towards a more virtual platform where industrial ecological data can be stored in a centralized knowledge base, with an ecosystem of tools and routines to develop models and analyze the data for further studies. This would reduce the time taken to perform further studies, allow of greater collaboration in the field and provide greater transparency on the results of studies [?]. An example of such a platform is the Metabolism Of Cities project, a digital research lab created by Paul Hoekman [?]. The platform's primary aim is to encourage research and allow collaboration in studying the metabolism of resources and energy surrounding specific regions. It contains a store of publications, IE data and an online material flow analysis tool to allow users to easily conduct MFA studies and interface with a common online database. Work is currently being done to integrate metabolism of cities with STAFDB to allow for future research to be directly inserted into the database.

1.2.2 UMIS Terminology

The components and design of a UMIS diagram are drawn from MFA concepts but can still be reconciled with IOA and LCA data. A definition list for the components of a UMIS diagram can be found in table ??.

A UMIS diagram can be thought to consist of three layers, a processes and flows layer, a virtual reservoir layer and a metadata layer. The processes and flows layer builds of the work of Pauliuk in [?], and structures the system as flows between transformation and distribution processes in the form of a bipartite directed graph. The virtual reservoir lies on top of this graph and contains information relating to stock. When material is moved in or out of storage in a system, it is represented by moving into or out of the virtual reservoir. As processes represent physical locations in a system, stock are only positioned on top of existing processes. The metadata layer is for storing additional information about stocks flows and processes. This is information concerning the reference space or time frame of a component, source of the data, uncertainty around a quantity's value, unit of the value, calculation details e.t.c. A visualization of the key aspects of a UMIS diagram can be seen in figure ?. In order to ensure that UMIS diagrams are able to be computer generated, transformation processes are enforced to have only one outflow to a distribution process. If a system's design indicates a flow from a transformation process to multiple distribution processes, the processes must be further disaggregated to separate them.

UMIS Component	Definition
System Boundary	Definition of the boundary of the data. This is defined by the reference space, time frame and material. Provides a limit of what parts of the real world we are interested in and modelling in this system
Reference Space	The geographical space within which all processes, stocks and flows internal to the UMIS diagram reside.
Reference Material	The material whose stocks and flows are described in the diagram. Can be at a high level of disaggregation (e.g a diagram with a reference material of Steel can contain stocks and flows of Carbon and Iron)
Reference Time frame	The time frame over which material is flowing or is stocked.
Process	An event involving a material
Transformation Process	A process where an object is transformed into another object
Distribution Process	A process where an object is transferred to another process or location
Storage Process	A process where material is moved into or from storage
Stock	Movement of material between a process and storage
Flow	Movement of material between a transformation and distribution process inside the diagram
Cross boundary flow	Movement from material outside the diagram to an internal transformation or distribution process.

Table 1.1: Components of a UMIS diagram

Divergent Disaggregation

UMIS also deals with the divergent disaggregation problem present in processes and materials. This is done by assigning every process and material a parent field and an `is_separator` flag. The parent field contains the process or materials at the next higher level of aggregation. For example the parent field for a blue car material would be car. As a result the processes and materials stored by STAFDB form a tree structure with the `is_separator` flag represent on edges in the tree to detect divergent disaggregation. This thesis focuses on the relationship between flows and processes and therefore considerations about disaggregation are out of scope, but it is likely that the program I have produced will be able to be extended to navigate the aggregation tree and ensure that divergent disaggregation does not occur within a UMIS diagram.

1.2.3 Mass Balancing

MFA, IOA and LCA all share core modelling concepts. Each methodology involve a separation background and foreground systems, with the background system acting as a more generic supplier of inputs and outputs to the more detailed foreground system [?]. The foreground system is often modelled through a system of equations and constraints, but one common concept to all methodologies is that of mass balancing. The idea of this is that throughout the entire system, matter must be conserved. Therefore the total mass of a given material entering a process must be equal to the total mass leaving [?]. This is defined by the following equation:

Where $i, j \in \{1, \dots, n\}$ are the processes in the system, f_{ij} is a flow from process i to j , q_i and o_i are the

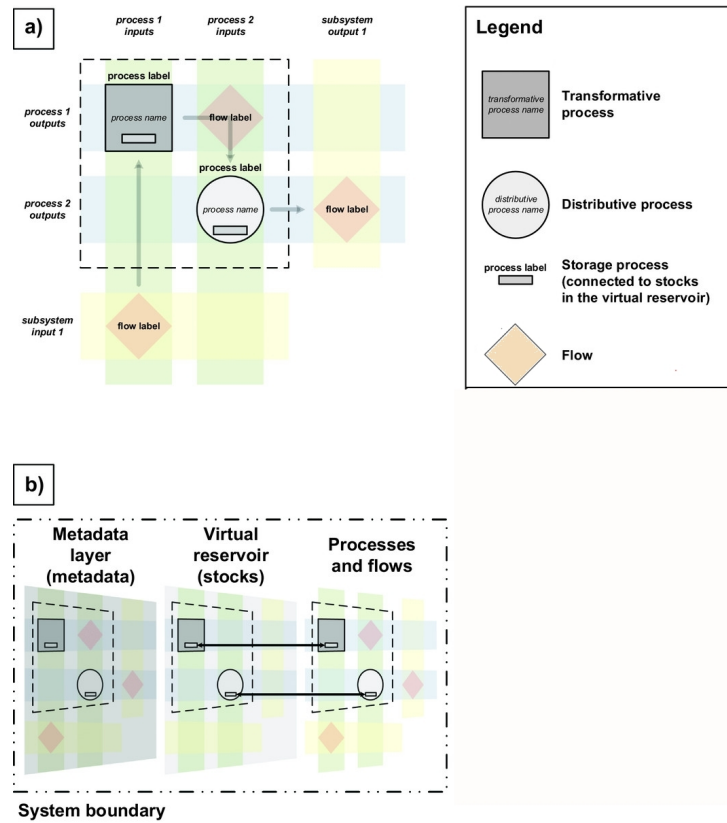


Figure 1.1: (a) Key aspects of a UMIS diagram, visualizing it in a matrix style. Contains transformation, distribution and Storage processes as well as 3 flows. The processes lie on the diagonal and the flows are on the row of the origin process and the column of the destination process (b) The orientation of the virtual reservoir and metadata layer in reference to processes and flows. Flows depicted by grey arrows in (a) and conceptual linkages denoted by black arrows in (b) are for illustration and are not properties of UMIS diagrams. UMIS = Unified Materials Information System. Adapted from [?]

inflow and outflow between process i and the background system respectively, and δs_i is the stock being supplied to or coming from storage:

$$\forall i, q_i + \sum_j^n f_{ji} = \sum_j^n f_{ij} + o_i + \delta s_i \quad (1.1)$$

Further constraints can be supplied in the form of transfer coefficients which allocate the total amount of material flowing into the process amongst the flows leaving the process. With a_{ij} being the transfer coefficient of the flow from process i to j , s_i is stock coming from storage (which may be 0) and z_i as the total of all material entering process i , this can be defined as:

$$z_i = q_i + \sum_j^n f_{ji} + s_i \quad (1.2)$$

$$f_{ij} = z_i a_{ij} \quad (1.3)$$

Concentration equations can also be used when processes extract specific goods from substances. For example, in a system modelling the flow of iron, a process may be used to model the extraction of iron from iron oxide ore. This is denoted by the following equation where f_s is the flow of substance s (e.g iron oxide ore) into a process, f_g is the flow of good g (e.g iron) out of a process and concentration coefficient c_{sg} is the concentration of good g in substance s :

$$f_s = f_g c_{sg} \quad (1.4)$$

Whilst Myers et al. have demonstrated how material systems in LCA, MFA, IOA studies can be visualized as UMIS diagrams [?], no work has been done to use UMIS structured data in a computational model. In order to demonstrate UMIS's suitability as a structuring system for data in a computational model I will develop a calculation engine for propagating uncertainty through UMIS structured systems.

1.3 Uncertainty in IE

1.3.1 What is uncertainty?

Common practice in IE research is to first define the system in terms of stocks, flows and processes and then identify values for the quantities of stock and flow. These observed values have an associated uncertainty which leads to an uncertainty in model outputs. As IE studies can be used to guide economic and political decisions, it is important to incorporate uncertainty into the results. A study by Danus and Burström [?] found that initial conclusions drawn when studying the flow of nitrogen were inconclusive when data uncertainties were taken into account. Therefore in cases where a comparison of systems are being conducted, the uncertainty around both systems must be taken into account to ensure that

1. Establish mathematical model
 - (a) Define system elements and relationships between the elements
 - (b) Define equations based on mass balance principle
2. Characterize data uncertainty
 - (a) Evaluate information about data (model parameters, inputs and outputs)
 - (b) Define characterising functions for uncertainty
3. Combine data and mathematical model
 - (a) Balance model and cross-check data
 - (b) Evaluate plausibility and reconcile data (iterative)
 - (c) Produce a calibrated model using all available data
4. Calculate uncertainty for calibrated model
 - (a) Propagate uncertainty through the model and calculate uncertainty of stocks and flows
 - (b) Interpret uncertainty estimates for resultant values from model
5. Analyze sensitivity & develop scenarios
 - (a) Identify critical model parameters
 - (b) Change parameters to perform scenario analysis

Figure 1.2: Adapted from: [?]

the results found are significant and cannot be attributed to variation in underlying data. As such, uncertainty is an important and well studied topic in the fields of LCA, IOA and MFA [?, ?, ?].

The sources of uncertainty can be separated into two kinds [?]. The first is aleatory uncertainty which comes from randomness in the underlying value. This is where either the method of measuring the value has an uncertainty around it so the precise value is unknown, or the value itself is non-deterministic. The second kind is epistemic which refers to generalizations about the value. One cause can originate from when values vary over space and time, therefore the value for a time period (e.g one year) or a geographic area (e.g the entire UK) can have an associated uncertainty. Another source can be subjective judgement such as when an unknown value is estimated based on a known value. Other causes can include scientific disagreement over the true value of a quantity, imprecise language on what the quantity refers to (e.g a material being said to be mostly carbon), and approximation in parameters in order to describe the system as a mathematical model. Epistemic uncertainty is potentially reducible through further investigation, however due to feasibility constraints it is sometimes unavoidable [?].

Further forms of epistemic uncertainty can be caused by possibilities in how the system is defined or described, however as we are looking at a general algorithm for propagating uncertainty in systems, this can be seen as out of scope. Instead, I will focus on uncertainty in specific parameter values.

1.3.2 Incorporating Uncertainty into UMIS

In [?], Laner et al. describe a five step procedure for incorporating uncertainty into MFA studies. These can be found in figure ??:

As the MFA, IOA and LCA share constraints surrounding mass balancing in their system structure, steps 1-4 in the above procedure can be applied to all three methodologies. Where they diverge is in the fifth stage, where the model is used to calculate further values and perform analysis on the system.

Therefore providing support for stages 1-4 can be seen as a relevant and useful addition to UMIS and STAFDB.

UMIS provides a method for structuring stocks and flows data into the system described in step one of this procedure however does not explicitly specify a unified way for specifying how the uncertainty around data values must be described. Current development on STAFDB and in the Metabolism of Cities project favours an approach investigated by Laner in [?]. This strategy combines the use of a pedigree matrix to classify data quality, as defined by Weidema and Wesnaes [?], and the application of data quality to uncertainty by Hedbrandt and Sörme [?]. The pedigree matrix provides five criteria of data quality (reliability, completeness, temporal correlation and geographical correlation). When a data point is recorded it is given a score for each criteria. The score is combined with a sensitivity level to provide a coefficient of variation. The data point can then be modelled as a normally distributed random variable with a standard deviation related to its coefficient of variation. As stocks and flow values can never be negative and can have asymmetrical properties, it can be appropriate to model the data point as a log-normal random variable. Therefore Laner also provides a method for characterising data quality as an uncertainty factor which relates to the standard deviation of a log-normal distribution. Therefore Laner's work could be used to convert descriptions of a data point's uncertainty from quality scores to probability density functions.

Whilst the above approach provides a method for characterising the uncertainty of individual data points independently, it does not provide support for considering them in respect to the entire system. This is key to steps 3 and 4a of the procedure in figure ???. To do this the data must be arranged as a mathematical model and the uncertainties of each model parameter must be calculated under the mass balance constraints. Typically this is performed using statistical approaches through either possibilistic methods, probabilistic methods or sensitivity analysis.

As a single UMIS diagram structures data in terms of a single time snapshot, a useful addition to UMIS would be to provide a program to propagate uncertainty through a UMIS diagram and infer unknown stocks, flows and transfer coefficients. Programs to support analysis such as in steps 4b, and 5 or in situations unique to IOA and LCA would have to be unique to the study and therefore are out of scope for this thesis.

1.4 Aims

Below are the aims for this dissertation:

1. Investigate literature in propagating uncertainty through industrial ecology models
2. Develop a program to create a model from UMIS formatted data and propagate uncertainty through it
3. Add support for representing uncertainty through normal, log-normal and uniform distributions
4. Evaluate the accuracy and performance of the program

Chapter 2

Technical Background

There are a variety of different methods for propagating uncertainty throughout MFA models. These typically fall under four main approaches, Gaussian error propagation, possibility theory, probability theory and sensitivity analysis. [?]. Uncertainty propagation can also be seen to involve three tasks, that of data reconciliation, error propagation, and data model consistency.

Data reconciliation involves altering data values so that they agree with all constraints in the mathematical model. As the data values are inherently uncertain, cases occur where their most likely values do not satisfy mass balance constraints, but one of their possible values do. For example take three flows A, B and C where flows A and B are entering a process and flow C is leaving. The mass balance equation is $A + B = C$. Say we model our prior uncertain knowledge of each flow by representing each as a random variable where $A \sim \mathcal{N}(30, 10^2)$, $B \sim \mathcal{N}(5, 2^2)$ and $C \sim \mathcal{N}(32, 4^2)$. As the expected values $A = 10, B = 5, C = 32$ do not satisfy the mass balance constraint, we can use our prior knowledge of the data values and the knowledge of how the material should behave in the system to reduce the uncertainty of all three parameters to a point where their expected values all agree with the constraint.

Error propagation involves inferring the uncertainties of values calculated by the model using the uncertainties of parameters supplied to the model. An example would be inferring the expected value and standard deviation of flow C in the above case.

Cases can occur when the independent data values obtained from the model do not agree at all with the model structure. The degree to which how well the data agrees with the model is known as data model consistency. A model which is completely consistent will have prior data values that agree with the mass balance constraints whilst a completely inconsistent model will have no way of reconciling the data so that agreement is reached. Between these two extremes are where values had to be reconciled to move into agreement.

As the mathematical model involves linear (Eqs. ??, ??) and non-linear (Eqs. ??, ?? constraints, situations arise where two uncertain and therefore variable parameters are multiplied together. This adds greater complexity to the task.

2.1 Sensitivity Analysis

A more model specific approach to propagating uncertainty is that of sensitivity analysis [?]. This technique puts a focus on evaluating the effect of a specific parameter's uncertainty on the model's results. The parameter is varied throughout its possible values and the different result values are recorded, producing an uncertainty interval for the results. This can be repeated for multiple parameters to try and identify what has the greatest effect on the results and therefore identify "hotspots" where changes in the real world should be enacted. This approach is less general than other approaches to dealing with uncertainty and therefore is not appropriate for an integration into a general system such as UMIS.

2.2 Gaussian Error Propagation

Gaussian Error Propagation is used by STAN, a free software for performing MFA [?], developed by Oliver Cencic. STAN provides a GUI for creating graphical MFA models comprising of stocks, flows and processes. The graphical model is translated into a mathematical model using equations ??, ??, ?? and ?. Each parameter in this equation may be considered to be unknown (\mathbf{y}), known with an uncertainty (\mathbf{x}) or exactly known (\mathbf{z}). Uncertainty is characterised by STAN through the use of 68 % confidence intervals around a "true" value. This is modelled as a normally distributed random variable with the "true" value as the mean (μ) and the 68% confidence intervals as the standard deviation (σ).

In STAN, uncertainty propagation is performed in two stages [?]. First \mathbf{x} are reconciled giving a new mean and standard deviation for each parameter (μ^* and σ^*). Next, the unknown parameters are calculated from μ^* and σ^* of known values. It is possible to also calculate the confidence interval of the unknown values, as all uncertainties are assumed to be modelled as normal distributions. This allows for uncertainty to be propagated to model results. By measuring the distance each parameter has been reconciled from its original value, we can get a measure of data-model consistency. We can impose a limit relative to a data value's variance to determine when a data point has been reconciled so far that the model can be seen to no longer be in agreement.

Data reconciliation is performed in the form of a weighted least squares optimization problem. The optimization is of minimizing the objective function ??:

$$F(\mu^*) = (\mu - \mu^*)^T \mathcal{Q}^{-1} (\mu - \mu^*) \quad (2.1)$$

Where \mathcal{Q} is a matrix containing the confidence intervals of the known parameters on the diagonal. \mathcal{Q} provides weightings in this minimization resulting in the more uncertain parameters being reconciled "further" than the more certain parameters.

The minimization of the objective function ?? is done in respect to mass balance constraints obtained from the graphical model. These equations may be non linear as they can involve flow parameters being multiplied with transfer or concentration coefficient parameters. The constraints of the objective function must be in linear form, therefore a linear approximation of the mass balance is obtained using a first order Taylor series expansion on the on-linear constraints. Full details can be found in [?].

The advantages of using this approach is that it is relatively fast in comparison to Bayesian approaches. It also performs validation to ensure that all unknown parameters can be inferred by the algorithm and does not try to calculate them if not enough known parameters are supplied. This is done by using Gaussian elimination to convert the set of constraint equations into reduced row echelon form. This allows you to check certain rows of the matrix to ensure that there exists at least one equation with only one unknown parameter for each unknown parameter.

Disadvantages to this approach is that it enforces all uncertainty to be characterised as normal distributions. This has shown to be too restrictive for a great deal of IE applications in section ???. The data reconciliation algorithm has also been shown to fail when parameters in non-linear equations are modelled to have a large uncertainty. To allow for more flexible and robust representations of uncertainty, possibilistic or probabilistic approaches should be used instead.

2.3 The Possibilistic Approach

Possibilistic approaches are based on fuzzy set theory where model parameters and their uncertainty are represented by fuzzy sets. A fuzzy set is used to model a "vaguely perceived or imprecisely defined quantitative piece of information" [?] and involves a membership function which maps a value to the degree to which the value belongs in the set: $f : X \rightarrow [0, 1]$. Intersection, union, addition and subtraction operations are supported over fuzzy sets [?]. Model parameters are represented by special cases of membership functions where $X = \{x|x \geq 0\}$, and x is mapped to the likelihood the parameter would take that value. Džubur et al. propose a data reconciliation and error propagation algorithm using fuzzy sets and demonstrate it on a case study of the Austrian wood system in 2011 [?]. In this study, the uncertainty of the m prior known model parameters ($x \in \{x_1, \dots, x_m\}$) are characterised through trapezoidal or triangular membership functions, but can be arbitrary as long as they are convex and normalised to 1. In cases where there are multiple data sources for a parameter, x_i it is defined by the intersection of each data source's membership function.

The fuzzy set theory approach deals with non-linear operations first before reconciling values according to linear constraints. Membership functions for each stock or flow in the model are calculated from the prior known values, these can be already known or may have to be calculated through non-linear operations. If stocks or flows are specified in multiple ways then the intersection of the membership functions is used. This step results in n model parameters ($\hat{x} \in \{x_1, \dots, x_n\}$).

Next Džubur et al. present a 3 step procedure for reconciling stock and flow values using the mass balance constraints (Eq. ??) around each process. Their procedure makes a distinction between internal flows (flows between processes in the model) and external flows (flows between a process and outside the model). We will refer to both stock and flows as flows for the procedure:

1. Calculate each internal flow i , as 3 different membership functions ($\gamma_1, \gamma_2, \gamma_3$)

- $\gamma_1 := \hat{x}_i$
- $\gamma_2 :=$ mass balancing the origin process of the flow using \hat{x}

- $\gamma_3 :=$ mass balancing the destination process of the flow using \hat{x}
 - $\bar{x}_i^* := \cap(\gamma_1, \gamma_2, \gamma_3)$
 - $\alpha_i :=$ the peak of \bar{x}^*
2. Calculate each external flow j , as 2 different membership functions (γ_1, γ_2)
- $\gamma_1 := \hat{x}_j$
 - $\gamma_2 :=$ mass balancing the internal process of the flow using \hat{x} (for other external flows) and \bar{x}^* (for internal flows)
 - $\bar{x}_i^* := \cap(\gamma_1, \gamma_2)$
 - $\alpha_j :=$ the peak of \bar{x}^*
3. Each γ^* is normalised to 1

As a result of this procedure we have the reconciled and calibrated data values with respect to constraints \hat{x}^* as well as a global measure of data-model consistency as the minimum α value. This α value provides validation for the model as if it is 0 at any intersection there is explicit information that there is either a problem with the data values defined or the way the model has been structured.

The running time of this method is relatively low at $O(n * d)$ where d is a variable related to the number of "cuts" of a membership function used when performing operations on membership functions. This therefore appears faster than the Gaussian error propagation algorithm which requires the Gaussian elimination and therefore has at least $O(n^3)$ time. The two are not directly comparable however as the fuzzy set approach requires the model of constraints to be stated explicitly, whilst the algorithm employed by STAN automatically develops a mathematical model.

Whilst fuzzy sets are one method of representing the uncertainty around data values, they are limited in that they are forced to be bounded. Therefore they struggle to represent the far tail end of values which may result in unlikely scenarios. By basing the representation on arbitrary probability density functions (pdfs) such as those used in Bayesian inference approaches, you have greater flexibility in how you choose to characterise uncertainty.

2.4 Probabilistic approaches

Another tactic for propagating uncertainty throughout a model is using pdfs. Whilst the Gaussian error propagation approach propagates uncertainty when assuming normal distributions, this provides an inflexibility over how uncertainty can be characterised. The need for both normal and log-normal distributions can be seen by Laner's work as discussed in section ??, but many studies characterise uncertainty of parameters through a wide variety of probability distributions[?, ?], including triangular, trapezoidal and uniform. The uncertainty represented by these distributions needs to be calibrated to

the mass balance constraints from the structure of the model as well as propagated to any model results by calculating their pdfs. Pdfs are more informative than using just mean and variance or fuzzy sets as metrics such as percentiles, skewness and correlation between parameters can be calculated.

2.4.1 Monte Carlo Simulations

In [?], Gottschalk et al. explores a probabilistic approach to model uncertainty in a MFA study investigating the environmental exposure of nano-particles. This mathematical model was built in terms of inflows to the system and transfer coefficients between processes in the system. Uncertainty of the inflows was characterised with a lognormal distribution and the transfer coefficients with triangular and uniform distributions. Stocks and flows values were seen as the results of the system and were calculated through matrix algebra.

External inputs to the system were organised into a column vector $\mathbf{q} \in \mathbb{R}^{n \times 1}$ where q_i is the amount of material flowing into process i . Transfer coefficients were arranged into a matrix $A \in \mathbb{R}^{n \times n}$ where t_{ij} is the transfer coefficient for the flow from process i to j (see Eqs. ?? & ??). The unknown total flow into each process z_i was arranged as $\mathbf{F} \in \mathbb{R}^{n \times 1}$. The mass balance constraints were enacted as:

$$(I - A)\mathbf{z} = \mathbf{q} \quad (2.2)$$

Where I is the $n \times n$ identity matrix.

Monte Carlo (MC) simulations were used to calculate pdfs for the stock and flow values of the results with $r = 100,000$ samples. r values for each transfer coefficient $a \in A$ are sampled from their pdfs. The values are used to calculate corresponding stock and flow values by solving for the total flow into a process, \mathbf{z} and substituting that in equation ?. Pdfs of the stocks and flows that account for the uncertainty in the model inflows and transfer coefficients can then be constructed from their r samples. Whilst this technique does generate pdfs of stock and flow values when transfer coefficients are known, it does not incorporate any prior knowledge of the stocks and flows values. To accomodate this, Gottschalk et al. extended this approach with Bayesian inference.

2.4.2 Bayesian Inference

Bayesian inference is a technique that can be used to infer pdfs of model parameters in non-linear systems [?]. When building an MFA system you implicitly define prior knowledge about the parameters within it. This may come from the system structure in the form of dependencies from mass balance constraints, or from a prior belief of a range parameter will lie in. For example when modelling a system, the researcher will know a flow value will be somewhere between 0 and the total input into the system. Once a MFA model has been converted into a mathematical model I , the joint prior probability distribution of the model parameters can be written as $P(\theta|I)$. As we have seen, model parameter values may be measured or estimated with an uncertainty (D). Bayes theorem allows us to infer the pdfs of reconciled model parameters (or *posterior* $P(\theta|D, I)$) given the *likelihood* of observations of parameter values [?]:

$$p(\theta|D) = \frac{p(D|\theta, I)p(\theta|I)}{p(D|I)} \quad (2.3)$$

where the denominator (the marginal likelihood) is defined as:

$$p(D|I) = \int p(D|\theta, I)p(\theta|I)d\theta \quad (2.4)$$

This allows us to calculate the pdfs of each parameter value weighted by to the likelihood of these observations. As the number of model parameters can often be large it can be complex to calculate the marginal likelihood, but by using Monte Carlo Markov Chain (MCMC) algorithms we can generate samples from the posterior distribution and use those samples to infer the posterior distributions of model parameters.

Monte Carlo Markov Chain Algorithms

Monte Carlo Markov Chain algorithms are techniques which allow for sampling from unknown posterior distributions. Basic MCMC algorithms require two things. The first is a proposal distribution ($g(\theta)$) for the model parameters. The second is a distribution proportional to the posterior distribution [?]. As the denominator in equation ?? is constant for all model parameters, our proportional distribution is:

$$f(\theta) = p(\theta|I)p(D|\theta, I) \propto p(\theta|D) \quad (2.5)$$

or the probability of observing the data given the model multiplied by the prior [?]. MCMC algorithms construct a Markov Chain with a stationary distribution equivalent to the target posterior density. This is done by drawing a proposal sample vector of all parameters in the model θ^* from $g\theta$ and using an acceptance probability to determine if the sample is representative of the posterior distribution. The acceptance probability is:

$$\alpha(\theta^i, \theta^{i-1}) = \min(1, \frac{f(\theta^i)g(\theta^{i-1})}{f(\theta^{i-1})g(\theta^i)}) \quad (2.6)$$

Therefore the probability of being accepted is proportional to how likely the new proposal belongs to the posterior compared to the the previous proposal. By using the ratio: $\frac{f(\theta^i)}{f(\theta^{i-1})}$ and using equation ??, we can see that the marginal likelihood constant cancels. Algorithm ?? shows a procedure for obtaining samples of model parameters from their posterior distribution.

Gottschalk et al. use such a sampler to improve on their Monte Carlo simulation results. However, due to a scarcity of nano-particle data, Gottschalk et al. had to estimate values for their observations of stocks and flows values. Because of this, they did not provide the likelihood functions of their model parameters for their MCMC sampler, only describing the approach they used and presenting their more certain, posterior distributions for stock and flow values.

Algorithm 2.1: Monte Carlo Markov Chain Sampler

```

1 for  $i = 1$  to  $n$  samples do
2   Draw candidate vector of proposal parameters  $\theta^*$  from  $g(\theta)$ 
3   Compute acceptance probability  $a = \alpha(\theta^*, \theta^{i-1})$ 
4   Draw uniform random number  $u \in [0, 1]$ 
5   if  $u \leq a$  then
6     Accept  $\theta^*$ 
7     Store  $\theta^*$ 
8      $\theta^i = \theta^*$ 
9   else
10    Reject  $\theta^*$ 
11    Store  $\theta^{i-1}$ 
12     $\theta^i = \theta^{i-1}$ 
13   end
14 end

```

Independence Sampler

In [?, ?], Cencic improves on his Gaussian propagation approach by describing a Bayesian framework for data reconciliation over MFA models. His approach involves using the observations of all model parameters as priors and then inferring the posterior distribution of those parameters when constrained by the mass balance equations. Therefore this method differs from that of Gottschalk as the posterior is not inferred from the likelihood of an observation, but from using the mass balance constraints. In his technique he divides model parameters into three groups:

- $\mathbf{w} \in W^{n_w}$ - Observed free variables, model parameters with a known prior distribution $p_w(\mathbf{w})$
- $\mathbf{u} \in U^{n_u}$ - Observed dependent variables, model parameters with a known prior distribution $p_u(\mathbf{u})$ that are functions of \mathbf{w}
- \mathbf{y} - Unobserved dependent variables, results of the model calculated from model parameters

The functions that define the dependent variables are $\mathbf{u} = \mathbf{h}(\mathbf{w}) = \begin{bmatrix} h_1(\mathbf{w}) \\ \vdots \\ h_{n_u}(\mathbf{w}) \end{bmatrix}$

and $\mathbf{y} = \mathbf{h}(\mathbf{w})$. Therefore $\mathbf{h}(\mathbf{w})$ can be thought of as $n_u = |\mathbf{u}|$ non-linear or linear constraint equations. Cencic also introduces the concept of a constraint manifold. If model parameters \mathbf{w} and \mathbf{u} can be thought of as having an independent joint prior pdf in a space $D \subseteq \mathbb{R}^{n_w+n_u}$, the model equations can be seen to define a constraint manifold $S \subset D$ where the equations are satisfied and model parameters are valid. Therefore the posterior distribution of model parameters conditional on model equations ($\pi_s(\mathbf{w})$) is the pdf of S . Cencic uses the example shown in figure ?? where $\mathbf{w} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\mathbf{u} = [x_3]$ and $\mathbf{h}(\mathbf{w}) = [h_{x_3}(\mathbf{w})] = [x_1 + x_2]$, as a visualisation of S as a plane. By using a MCMC sampler, the posterior distribution of S can be inferred using proposals of \mathbf{w} . The accepted values for \mathbf{w} can be used then to calculate the dependent variables.



Figure 2.1: Visualisation of prior distributions with constraint manifold. Top left: prior pdfs of observed variables. Top right: Pdf of variables under constraints. Bottom: Marginalised densities of variables under constraints. Source: [?]

The MCMC sampler described is known as an independence sampler as its proposals for \mathbf{w} are drawn independently from their marginal distributions, $g(\mathbf{w}) = p_w(\mathbf{w})$. The proportional distribution $f(\mathbf{w}^i, \mathbf{w}^{i-1})$ of the sampler must be proportional to the pdf of the constraint manifold S . As prior distributions of \mathbf{u} and \mathbf{w} are assumed to be independent, $\pi_s(\mathbf{w}) = \frac{p_u(\mathbf{h}(\mathbf{u})p_w(\mathbf{w})V(\mathbf{w})}{\int_W p_u(\mathbf{h}(\mathbf{u})p_w(\mathbf{w})V(\mathbf{w})d\mathbf{w}}$. The term $V(\mathbf{w}) = \sqrt{|I + \mathbf{H}^T \mathbf{H}|}$

is a Lebesgue measure of the constraint manifold S where $\mathbf{H} = \begin{bmatrix} \frac{\partial h_1 \mathbf{w}}{\partial w_1} & \cdots & \frac{\partial h_1 \mathbf{w}}{\partial w_{n_w}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{n_u} \mathbf{w}}{\partial w_1} & \cdots & \frac{\partial h_{n_u} \mathbf{w}}{\partial w_{n_w}} \end{bmatrix}$ and is evaluated at \mathbf{w} .

In his paper, Cencic evaluates his approach over small MFA models of at most two processes where the functions $\mathbf{h}(\mathbf{w})$ are calculated explicitly. Therefore, in generalising it to generic MFA models, a system must be created for selecting which observed variables are to be considered free and which are dependent. Another task is the construction of n_u constraint functions, which incorporate all the mass balancing information to express each dependent variable.

Incremental MFA

In [?], Lupton presents an incremental approach to reducing uncertainty in an MFA model tracking the flow of steel in Austria in 2015. The external inputs \mathbf{q} and transfer coefficients \mathbf{A} can be thought of as free variables, and observations of a subset of the flow parameters as dependent variables. This approach did not attempt to infer the posterior distribution of a constraint manifold, but rather uses the likelihood of dependent observations to accept or reject free variable proposals. Characterisations of flow value observations as normally distributed random variables were used to model the likelihood of dependent flows belonging to the observed distribution. A full description of the construction of the model can be found in section ???. To construct their model and perform the MCMC sampling, Lupton



Figure 2.2: Visualisation of Hamiltonian Monte Carlo Sampling along a 2-Dimensional posterior probability density function. The red line shows the leapfrogging step from the previous accepted proposal value, the green box. Source: [?]

used a variation of a Hamiltonian Monte Carlo sampler called the No-U-Turn (NUTS) sampler, provided by pymc3 [?].

Hamiltonian Monte Carlo and the No-U-Turn sampler

Hamiltonian Monte Carlo (HMC) [?] provides a useful method for sampling a new proposal vector sample from the previous accepted proposal. The Hamiltonian sampler can be thought of treating the multi-dimensional posterior distribution of model parameters as a concave surface and its log likelihood as a negative potential energy function. An accepted proposal parameter $\theta^i \in \boldsymbol{\theta}^i$ can be thought of as a coordinate on that surface with r the momentum of a particle at that coordinate. L "Leapfrog" steps (shown in algorithm ??) calculate a new proposal parameter θ^* from θ^{i-1} by "rolling" this "particle" along the surface with momentum r and distance ϵ . To ensure variation in the next parameter proposed, r is initialised randomly before performing the leapfrogging steps.

Algorithm 2.2: Leapfrog step:

Finds new proposal parameter $\tilde{\theta}$ from previous parameter θ , with previous momentum r , step size ϵ and log-likelihood of posterior distribution \mathcal{L}

- 1 Given $\boldsymbol{\theta}, r, \epsilon$
 - 2
 - 3 Set $\tilde{r} = r + (\epsilon/2)\Delta_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$
 - 4 Set $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} + \epsilon\tilde{r}$
 - 5 Set $\tilde{r} = r + (\epsilon/2)\Delta_{\boldsymbol{\theta}}\mathcal{L}(\tilde{\boldsymbol{\theta}})$
 - 6 **return** $\tilde{\boldsymbol{\theta}}, \tilde{r}$
-

L and ϵ must be tuned correctly during the algorithm to ensure efficient sampling. If ϵ is too large, the

proposal vectors drawn will have been drawn too far away from the previous value and therefore will be rejected too often, wasting computation time. Also if L is too large, the leapfrogging will perform a "U-Turn" and start moving θ towards its original value. If L and ϵ are too small then the proposed values will be too similar to each other and therefore not explore the full space of the posterior distribution. The No-U-Turn sampler (NUTS) [?] automatically tunes L and ϵ to avoid these issues.

In this thesis, I will use adapt Lupton's approach to propagating uncertainty through MFA models for UMIS structured systems. This technique will involve automatically developing a mathematical model from a system defined by stocks and flows with independently observed data values and then using a NUTS sampler provided by pymc3 to infer the posterior distributions of these values in respect to mass balance constraints over the model.

Chapter 3

Project Execution

3.1 Overview

The aim for this thesis is to prove UMIS's suitability as a basis for performing computation over industrial ecology systems. To show this I have developed a Bayesian inference engine to propagate uncertainty through UMIS systems and display the calibrated uncertainty values for stocks and flows data. This task is formed of three sections:

1. Extract stocks and flows data from STAFDB and arrange in a UMIS diagram
2. Convert the UMIS diagram into a mathematical model and propagate uncertainty through it
3. Display calibrated uncertainty values for stocks and flows data

3.2 Constructing the UMIS diagram

3.2.1 STAFDB Prototype

As STAFDB has not yet been released, I have developed a prototypical version (STAFDB-P) for my implementation. I was provided an Entity Relationship Diagram (ERN) for in-progress STAFDB by Zoë Petard from the University of Edinburgh. As only a subset of the entities in the database are necessary for the inference engine, the STAFDB-P only contains that subset. An ERN describing the prototype can be seen in figure ??.

STAFDB is designed around describing all industrial ecology systems in terms of stocks and flows which are consolidated into a single entity, STAF. STAFs have associated attributes describing the space, material, timeframe and processes they are in reference to. A STAF is associated with at least one data value. The data value contains information regarding the quantity, unit and specific material that

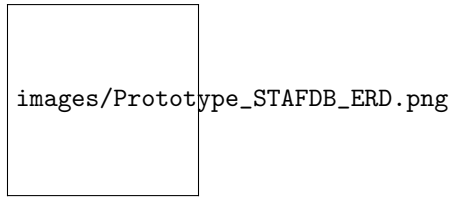


Figure 3.1: Entity Relationship Diagram showing the schema of STAFDB-P

is flowing or being stored. STAFs are associated with multiple data values when they are describing a composite material; each data value is used to describe a fundamental component of the composite material. STAFDB also stores information regarding the provenance of the data, such as the data's quality and the study it has originated from. In [?], Laner describes a method for expressing a data's quality as uncertainty in the form of normal and log-normal distributions, however no definitive method has been proposed yet for STAFDB. Therefore I have replaced information regarding the provenance of the data with JSON strings describing the data's uncertainty. The uncertainty can be characterised as uniform with a lower or upper bound, or as normally or lognormally distributed with a mean and standard deviation.

3.2.2 UMIS Diagram

A UMIS diagram is used to represent the relationship between stocks, flows and processes. STAF records are first extracted by their ID from STAFDB-P and parsed into sets of **Stock** and **Flow** python objects as well as their related attributes. Stocks and flows are separated from STAFs in a UMIS diagram as they have differing behaviour. Both **Stocks** and **Flows** inherit from a **Staf** class which has attributes for time frame, material, origin process and destination process. Stock and flow also have a dictionary which maps a material to its **StockValue** and **Value** respectively. These values are the information regarding the quantity of that material being stored or flowing. Both **StockValue** and **Value** store the quantity, unit and uncertainty around the data. The uncertainty is simply the serialized JSON string described in section ???. **StockValue** contains an additional field describing the "stock type". This is because stocks in STAFDB may refer to the total amount stored at that process or instead the net transfer of material to or from storage during this time frame. Origin and destination processes for stocks and flows are parsed into **UmisProcess** objects. In STAFDB, processes are generalised so that one process record can be used to describe the same event in various locations. For example a manufacturing process in STAFDB can be used to represent manufacturing in Spain or the UK or the USA. In a UMIS diagram, processes must be differentiated by location (space) as flows between the same processes but in different reference spaces are supported. Therefore **UmisProcess**'s have a unique diagram ID formed of concatenating their process ID and their reference space ID. Processes also store information about their type and name. When a flow is parsed, its origin and destination process is validated to ensure that they are between a transformation and distribution process. Likewise, stocks are validated to ensure that they are between a storage and a transformation or distribution process.

UMIS diagrams are implemented as the class **UmisDiagram**. They are constructed from sets of external flows entering the diagram, internal stocks and flows in the diagram, and flows exiting the diagram. In agreement with the findings of Myers and Pauliuk [?, ?], the diagram follows a graph pattern [?]. Therefore, external inflows and external outflows to the system are stored as separate sets, whilst the internal flows and stocks in the diagram are stored as a dictionary mapping each internal process to



Figure 3.2: UML Class diagram of UMIS diagram data models

its outflows and its stock if it has one. The intention is to develop a python representation of a UMIS diagram that is decoupled both from STAFDB and from the Bayesian inference engine. Therefore if further computational models or visualisation tools are developed, they can build directly off of the python classes, without having to write new methods for extracting the data from STAFDB. A UML diagram of the structure of `UmisDiagram` and its components can be seen in figure ??.

Now that stocks and flows have been organised into a UMIS diagram, they can be used to construct a mathematical model, whose parameters can then be inferred with accurate, calibrated uncertainty using Bayesian inference.

3.3 Constructing the Mathematical Model

3.3.1 A System of Equations

In [?, ?], Lupton and Gottschalk use the following equation to model the mass balance constraints of the system:

$$(I - A) \bullet z = (q) \quad (3.1)$$

where I is the identity matrix, A is a matrix of transfer coefficients (TCs) and $a_{ij} \in A$ is the TC from equation ??, representing the proportion of throughput from process i that moves to process j . z is a vector of process throughputs as mentioned in equation ??, q is a vector of external inflows to each process from outside the system, and \bullet is the dot product operation.

From equation ??, we know:

$$F = A \cdot z \quad (3.2)$$

where $f_{ij} \in F$ is the flow from process i to process j and \cdot is the elementwise multiplication operation. Therefore equations for each outflow from processes in Lupton and Gottschalk's models can be derived as:

$$F = A \cdot (I - A)^{-1} \bullet q \quad (3.3)$$

In the context of Cencic's framework, A and q can be seen as free observed parameters. Matrix F can be seen as a matrix of dependent observed parameters constructed from free parameters.

Whilst this equation was sufficient for Lupton and Gottschalk's case studies, UMIS systems can include the use of stocks where material flow in and out of a virtual reservoir. Furthermore, UMIS systems can allow for the storage and flow of composite materials. Therefore, concentration coefficients (as in equation ??) must be included in order to reconcile the flow of composite materials into a reference material.

To accommodate the introduction of stocks, stock values can be separated into two types. In the first, material is stored into the virtual reservoir (s^-). These are treated as an extra outflow from an existing process into a new storage process. As such they have their own TCs (A') and staf equations ($F' = \begin{bmatrix} F & s^- \end{bmatrix}$). In the second type, material enters the system from the virtual reservoir (s^+); these are now treated as another form of inputs to processes in the system ($Q = \begin{bmatrix} q & s^+ \end{bmatrix}$).

In order to ensure that the system of equations is mass balancing correctly and that the likelihoods of dependent parameters are being calculated in terms of the same material, concentration coefficients are needed to reconcile staf values for composite materials into the reference material. Input parameters (Q) must be converted into the reference material for use in the equations through:

$$Q_r = Q \cdot C_Q \quad (3.4)$$

The results from staf equations must also be converted from the reference material into the material that is observed through:

$$F'_o = \frac{F'}{C_{F'}} \quad (3.5)$$

Therefore the system of equations for my model is:

$$F'_o = \frac{A' \cdot (I - A)^{-1} \bullet \text{sum}(Q \cdot C_Q)}{C_{F'}} \quad (3.6)$$

where F'_o are dependent observed parameters whilst A', Q, C_Q and $C_{F'}$ are free observed parameters.

3.3.2 Representing Model Parameters

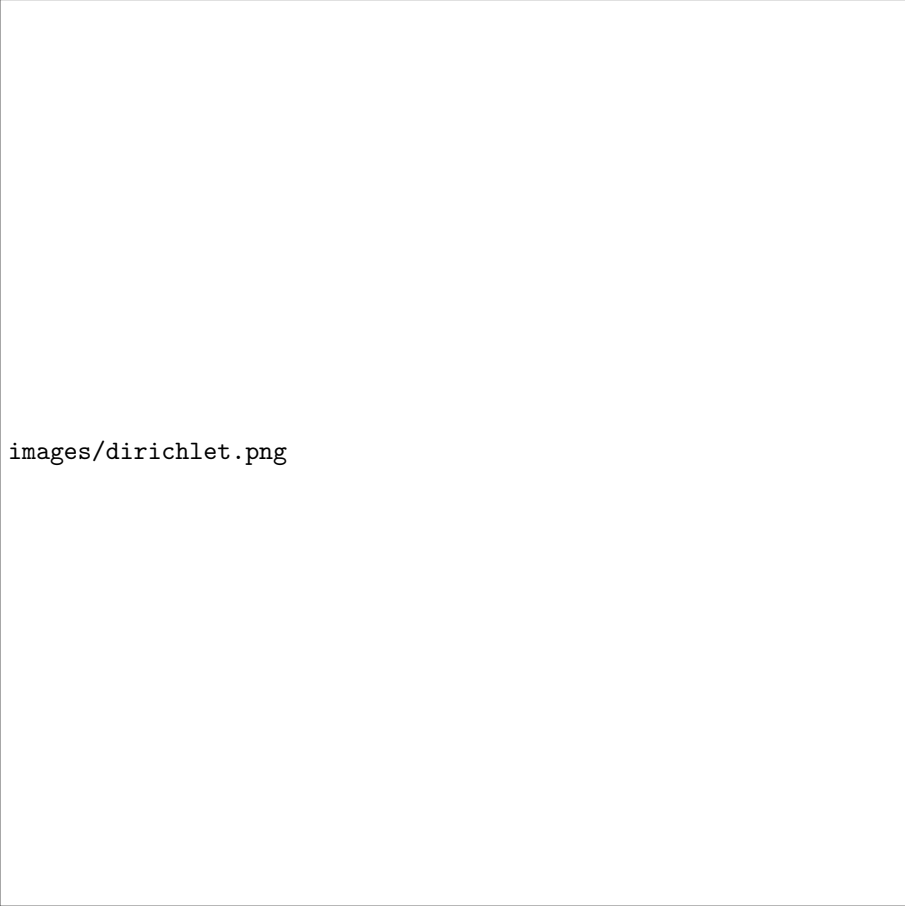
Each parameter in the model has a known prior distribution $p(x)$ which characterises the uncertainty around that parameter's true value. For Q, F'_o, C_Q and $C_{F'}$ these can be represented as normally, log-normally or uniformly distributed stochastic variables whose distribution parameters are $\{\mu_n, \sigma_n\}$, $\{\mu_l, \sigma_l\}$ and $\{l_u, u_u\}$ respectively. The values for F'_r and Q are supplied when constructing the UMIS diagram and take the form of **Uncertainty** attributes in the corresponding **Stock** and **Flow** objects. A material reconciliation table is supplied to the system and is used to map materials to the prior distributions of CCs.

Representing TCs

TCs have a more interesting prior distribution as their value relies on the structure of the model and differ according to their process type. Lupton also recognises the division of transformation and distribution processes and proposes characterising the TCs of each process type through different distributions. For both types if a process only has one outflow then its TC must be 1, whilst if there are no outflows, the TC must be 0. The modeller may have prior knowledge of TC values that they wish to supply to the model. This can be done through the use of a TC lookup table which maps the origin process and destination process IDs of the TC to an **Uncertainty** object.

Transformation processes can have at most 2 outflows: a stock and an outflow to another process. In this case, the TCs must be a and $1 - a$. If no prior knowledge of the TC exists, then a is represented as a uniform distribution between 0 and 1. If prior knowledge about one TC exists in the lookup table, a is represented as the stochastic variable from that **Uncertainty**. If prior knowledge about both exists, then one is arbitrarily selected to model a .

Distribution processes can have many outflows, including to stock. Lupton models these TCs as parameters from a Dirichlet distribution. A Dirichlet distribution is a continuous multivariate distribution with two parameters α and θ [?]. Drawing from a Dirichlet distribution results in $\{\theta_1, \dots, \theta_n\} = \theta$ where



images/dirichlet.png

Figure 3.3: Effect of α on a three parameter Dirichlet distribution. Source: [?]

$\sum_i \theta_i = 1$ and $\theta_i \geq 0$. This makes it ideal for representing TCs. The share parameters $\{\alpha_1, \dots, \alpha_n\} = \alpha$ indicate a weighting to a certain configuration of θ . Figure ?? shows the effect of α on a three parameter Dirichlet distribution. If no prior knowledge of TCs exist, each α_i is set to 1 resulting in every possible configuration of TCs being equally likely. If prior knowledge of any TC exists in the lookup table, its α is set to the mean of the `Uncertainty` object.

3.3.3 Model Construction

In his implementation, Lupton used Pymc3 to create his mathematical model from stochastic variables, observe the likelihood of his dependent parameters and sample from the resultant posterior distribution. He used Theano operations to define the equations in his system. We will use the same tools in our approach. Construction of the model before sampling can be split into three stages: Defining the parameter priors, creating the model parameters and equations, and observing the dependent parameters to calculate their likelihood. This structure is illustrated in figure ??.

images/inference_engine_structure.png

Figure 3.4: Structure of model construction for the Bayesian inference engine

Stage 1: Defining Parameter Priors

The first stage is devoted to creating python objects which are in turn used to create the stochastic random variables which represent parameters in the model. It takes as input the python object representation of the UMIS diagram, a material reconciliation table, a TC observation table and reference attributes. The material reconciliation table maps a material to its concentration coefficient. The reference attributes refer to the specific time frame and material over which the system is mass balanced. On processing stocks and flows, if they belong to the wrong time frame then they are ignored. If the stock or flow does not have a value for the reference material, it undergoes material reconciliation. This checks to see if the stock or flow has a value for a material corresponding to an entry in the material reconciliation table. If so, that value is used and that materials CC is stored for use later. If a stock or flow has no material that can be reconciled, it is ignored. If no material reconciliation is necessary a deterministic CC of 1 is used. The TC observation table is used to store prior information about TC values. Its design and use is discussed in section ???. The python representation of the UMIS diagram consists of the list of External Inflows, dictionary of processes to internal stafs (Internal Stafs Dictionary) and list of External Outflows discussed in section ???.

The first python objects constructed are the Math Processes and the Math Process Dictionary. This maps the diagram ID of the process to its corresponding math process. This allows for looking up the Math Processes later which is useful for placing model parameters in the correct positions in parameter matrices. On creation of a new Math Process it is assigned an automatically increasing process index which refers to its row in the parameter matrices. Construction is performed by iterating over the Internal Stafs Dictionary and External Outflows.

Outflows are added to a process as ParamPrior objects. These contain a parameter's type (e.g TC), origin process ID, destination process ID and an uncertainty attribute which describes its distribution. ParamPrior objects have a method to construct a Pymc3 stochastic variable corresponding to the parameter's uncertainty. For these ParamPrior objects, the uncertainty refers to the prior knowledge of the TC for the outflow. Math Processes have a method to use its outflows to create a list of process IDs that receive flows from it, and a corresponding list of stochastic variables representing its TCs. These stochastic variables are constructed in accordance with section ???. Any stocks that are coming from the virtual reservoir (s^+) are ignored, whilst stock going to the virtual reservoir (s^-) are represented as an outflow to a process has no outflows.

Next the InputPriors object is created. This contains two dictionaries, one mapping process IDs to their inputs from External Inflows (External Inflows Dictionary) and the other mapping process IDs to the inputs from Stock Inputs (Stock Inputs Dictionary). Each input is represented by two ParamPrior objects, one representing the un-reconciled staf value flowing into the system ($\{q_i | s_i^+\} \in Q$) and the other representing its CC ($c_q \in C_q$). The External Inflows are iterated over to construct the External Inflows Dictionary whilst Stock Inputs Dictionary is constructed from the Internal Stafs Dictionary.

The final python object is the Dependent Staf Priors. This is a list of Dependent Staf Prior objects. Each Dependent Staf Prior consists of two ParamPrior objects, one representing the un-reconciled observed distributions of the dependent model parameters ($f'_o \in F'_o$) and the other representing its CC ($c_r \in C_{F'}$). Dependent Staf Priors separate the observed parameters into three lists by their distributions.

Stage 2: Create Model Parameters and Equations

Once the python objects have been created they are used to develop a Pymc3 mathematical model. The matrices in the right hand side of equation ?? are constructed as Theano tensors and then populated either by Pymc3 stochastic variables or constants.

First the TC matrix (A') is constructed from the Math Process Dictionary. It is initialised as an $N_p \times N_p$ dimensional Theano matrix of zeros, where N_p is the number of processes in the system (including new ones created from stock values). The Math Process Dictionary is then used to create the Pymc3 stochastic random variables for each TC and place them in the correct positions in the matrix. Similarly, the Staf Inputs Matrix (Q) and its corresponding Input Coefficients Matrix (C_Q) are constructed as Theano tensors and populated by stochastic random variables from the Input Priors. By adding these random variables to the Pymc3 model they are set as free variables in a mathematical model. Later when this model is sampled from, proposals will be drawn as a vector of these free variables.

Each Theano matrix as a whole is set as a named Pymc3 deterministic variable. When this model is sampled, Pymc3 will then store the value of each matrix for each sample. This allows us to extract the posterior samples of each matrix from their variable name and then access specific parameters using their location in that matrix.

Chapter 4

Critical Evaluation

Chapter 5

Conclusion

Appendix A

Appendix A

A.1 STAFDB Entity Relationship Diagram