University of Reading

Applications of Parameter Estimation to Meteorology and Food Processing

Peter Taylor

1 September 2004

Submitted to the Department of Mathematics, University of Reading, in partial fulfilment of the requirements for the Degree of Master of Science.

Abstract

The use of parameter estimation in the fields of meteorology and food processing has been investigated. A methodology has been built and tested against the parameters in the Lorenz system of equations from the field of meteorology. With suitable adaptations, the same algorithm has been used to retrieve small numbers of parameters in simple enzyme reactions. Using the analysis from these two independent problems, parameters have also been retrieved for two different models of a chemical reaction from a food processing problem. These parameter values have been obtained for two different modelling procedures - with and without observational data incorporated. Together with optimal parameter values, possible areas for improvement have also been offered.

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The process of using dynamic, experimental data in the formulation of mathematical models has developed alongside the demand for dynamic system models. The production of high speed computers has fed the demand for such models. This field relies on optimisation theory, simulation knowledge and data handling techniques.

In many models the values of some parameters are often determined by trial and error. They may contain information about the flow properties and characteristics or numerical stability conditions. The values of the parameters impact upon the predictive performance of the model, and can make a huge difference in matching observations.

This dissertation reviews the parameter estimation techniques used in the field of meteorology, and applies them to the Lorenz equations. The Lorenz system is particularly sensitive to parameter values. For certain choices of parameters, the system is unstable. We have attempted to retrieve the same choice of parameters in such a chaotic case.

Flavour chemists working in the Food Industry have carried out significant study and analysis of the final colours and flavours in steady state at the end of batch testing. Ideally the data collected could be used to extract the various rate constants associated with the reaction network, thus enabling quantitative numerical experiments to replace costly practicals and laboratory studies.

We have investigated the ways in which the values of parameters involved in flavour and colour batch testing in the Food Industry could be estimated. The chemical experiments we wish to simulate involve mixing a sugar with an amino acid at a specified temperature, and can take several hours. The temperature is usually at a number of steady states.

In fitting data to experiments in the Food Industry, the aim is to recover the forcing parameters from the experimental observations to improve the model's prediction, using parameter estimation. Currently two techniques are employed by the Food Biosciences Department at the University of Reading to do this. The first uses ASPEN, a commercial modelling package from the chemical industry. It is treated as a black box and a significant amount of human intervention takes place in order to retrieve admissible values. The second uses MathCAD, and involves sequential curve-fitting. Both techniques suffer from the limited nature of the data, so a better testing strategy is sought.

This dissertation is organised as follows: some of the possible minimisation techniques used in parameter estimation, and the reasons for our choice of method, are described in Chapter 2. The chosen minimisation algorithm is then applied to the Lorenz system and a simple chemical reaction in Chapters 3 and 4 respectively. Chapter 5 analyses the food problem and provides parameter values for certain models of the reactions involved. Finally, further improvements to the parameter estimation process are presented in Chapter 6.

Chapter 2

Parameter Estimation Methods

2.1 Introduction

All parameter estimation problems involve minimisation. The choice of algorithm is problem-dependent. We will now discuss some of the possible choices of minimisation techniques, and their typical applications.

2.2 Downhill Simplex Method

The *downhill simplex method* is due to Nelder and Mead [9]. It is a direct search method for multidimensional unconstrained minimisation. According to [3], the Nelder-Mead algorithm is especially popular in the field of chemistry, and this is why it is chosen and discussed here.

The algorithm is based on minimising a scalar-valued nonlinear function of n real variables by evaluating it at the $n+1$ vertices of a *simplex.* A simplex is the geometrical figure consisting, in n dimensions, of $n+1$ vertices and all their interconnecting line segments and polygonal faces. No derivative information is required (explicit or implicit), so this simplifies the modelling process.

For a general minimisation procedure, the simplex adapts to the local landscape by elongating to move down long gentle slopes, or by contracting around the final minimum. The method can be considered geometrically. We assume the simplex is nondegenerate, and encloses a finite inner n-dimensional nonzero volume (so it is a convex hull of $n+1$

vertices). We take one point on the figure to be the origin.

The algorithm in more than one dimension needs a starting guess, namely an n-vector of independent variables as the first point to try. One or more test points are computed, along with their function values. A simplex is then set up near the initial guess, and the input guess is placed in the simplex. The algorithm then attempts to make its own way downhill until it finds a (at least local) minimum. According to [11] we take a starting point $\mathbf{P}_0$ and take the other n points to be

$$\mathbf{P}_i = \mathbf{P}_0 + \lambda \mathbf{e}_i \tag{2.1}$$

where the $\mathbf{e}_i$'s are n unit vectors, and where λ is a constant guess of the characteristic length scale.

The downhill simplex method now takes a series of steps or *reflections*, moving the point of the simplex where the function is largest (*highest point*) through the opposite face of the simplex to a lower point. The reflections must conserve the volume of the simplex. Wherever possible the method expands the simplex in one or another direction to take larger steps. When it reaches a *valley floor*, the method contracts itself in the transverse direction and tries to flow down the valley. If the valley is too tight, the simplex contracts itself in all directions, pulling itself around its lowest (best) point.

2.2.1 The Algorithm

We now describe the downhill simplex method algorithm presented in [11], with mathematical details in the case of strictly convex functions in 1-D and 2-D from [3]. Four scalar parameters must be specified to define a complete Nelder-Mead method: coefficients of *reflection* (ρ), *expansion* (χ), *contraction* (γ) and *shrinkage* (σ) with

$$\rho > 0, \qquad \chi > 1, \qquad \chi > \rho, \qquad 0 < \gamma < 1, \qquad \text{and} \qquad 0 < \sigma < 1. \tag{2.2}$$

At the start of the kth iteration, $k \geq 0$, a nondegenerate simplex is given, along with its $n+1$ vertices, each of which is a point in the n-space. Iteration k begins by ordering and labelling these vertices as $\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{n+1}^{(k)}$, such that for the function f,

$$f_1^{(k)} \leq f_2^{(k)} \leq \ldots \leq f_{n+1}^{(k)}, \tag{2.3}$$

where $f_i^{(k)}$ denotes $f(\mathbf{x}_i^{(k)})$. The kth iteration generates a set of $n+1$ vertices that define a different simplex for the next iteration. Because we seek to minimise f, we refer to $\mathbf{x}_1^{(k)}$ as the *best* point or vertex, and to $\mathbf{x}_{n+1}^{(k)}$ as the *worst* point. Similarly, we refer to $f_{n+1}^{(k)}$ as the worst function value, and so on.

The result of each iteration is either (1) a single new vertex - the *accepted point* - which replaces $\mathbf{x}_{n+1}$ in the set of vertices for the next iteration, or (2) if a shrink is performed, a set of n new points that, together with $\mathbf{x}_1$, form the simplex at the next iteration. Here is the algorithm for one iteration (we drop the k superscript for simplicity).

Step One - Order

Order the $n+1$ vertices to determine which point is the highest (worst), next-highest, and lowest (best). Use a vector whose components are pre-initialised to the values of the function evaluated at the $n+1$ vertices. Algebraically, order the vertices to satisfy

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \ldots \leq f(\mathbf{x}_{n+1}). \tag{2.4}$$

Step Two - Reflect

Compute the *reflection point* $\mathbf{x}_r$ by extrapolating by a factor of ρ through the face, that is, reflect the simplex from the high point with

$$\mathbf{x}_r = \mathbf{x}_b + \rho(\mathbf{x}_b - \mathbf{x}_{n+1}) = (1+\rho)\mathbf{x}_b - \rho\mathbf{x}_{n+1}. \tag{2.5}$$

The term $\mathbf{x}_b = \sum_{i=1}^{n} \mathbf{x}_i/n$ is the centroid of the n best points (all vertices except for $\mathbf{x}_{n+1}$), that is, the vector average or centre of the face of the simplex across from the high point. Evaluate the function at the reflected point, so $f_r = f(\mathbf{x}_r)$. If $f_1 \leq f_r < f_n$, then accept the reflected point $\mathbf{x}_r$ and terminate the iteration.

Step Three - Expand

If the result is better than the best point, so $f_r < f_1$, try an additional extrapolation by a factor χ and calculate the *expansion point* $\mathbf{x}_e$

$$\mathbf{x}_e = \mathbf{x}_b + \chi(\mathbf{x}_r - \mathbf{x}_b) = \mathbf{x}_b + \rho\chi(\mathbf{x}_b - \mathbf{x}_{n+1}) = (1 + \rho\chi)\mathbf{x}_b - \rho\chi\mathbf{x}_{n+1} \tag{2.6}$$

using (2.5), and evaluate the function there, so $f_e = f(\mathbf{x}_e)$.

If $f_e < f_r$, then the additional extrapolation succeeded, and $\mathbf{x}_e$ replaces the high point so the iteration terminates. Otherwise (if $f_e \geq f_r$), the additional extrapolation failed but the reflected point $\mathbf{x}_r$ can still be used and the iteration terminated.

Step Four - Contract

If the reflected point $\mathbf{x}_r$ is worse than the second-highest, so that $f_r \geq f_{n+1}$, perform a *contraction* between $\mathbf{x}_b$ and the better of $\mathbf{x}_{n+1}$ and $\mathbf{x}_r$.

(a) Outside

If the point is strictly better than the highest, so $f_n \leq f_r < f_{n+1}$, then replace the highest but look for an intermediate lower point. That is, perform an *outside contraction* of the simplex along one dimension. Calculate

$$\mathbf{x}_{oc} = \mathbf{x}_b + \gamma(\mathbf{x}_r - \mathbf{x}_b) = \mathbf{x}_b + \gamma\rho(\mathbf{x}_b - \mathbf{x}_{n+1}) = (1 + \rho\gamma)\mathbf{x}_b - \rho\gamma\mathbf{x}_{n+1}, \tag{2.7}$$

using (2.5) again, and evaluate $f_{oc} = f(\mathbf{x}_{oc})$. If the contraction gives an improvement, so $f_{oc} \leq f_r$, then accept $\mathbf{x}_{oc}$ and terminate the iteration; otherwise, perform a shrink (Step Five).

(b) Inside

If $f_r \geq f_{n+1}$, then perform an *inside contraction* by calculating

$$\mathbf{x}_{ic} = \mathbf{x}_b - \gamma(\mathbf{x}_b - \mathbf{x}_{n+1}) = (1 - \gamma)\mathbf{x}_b + \gamma\mathbf{x}_{n+1}, \tag{2.8}$$

and evaluate $f_{ic} = f(\mathbf{x}_{ic})$. If $f_{ic} < f_{n+1}$, then accept $\mathbf{x}_{ic}$ and terminate the iteration; otherwise go to Step Five.

Step Five - Perform a shrink step

If the high point cannot be improved upon, then contract around the lowest (best) point. Evaluate f at the n points $\mathbf{v}_i = \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1), i = 2, \ldots, n+1$. The (unordered) vertices of the simplex at the next iteration consist of $\mathbf{x}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n+1}$.

Figure 2.1 shows the effects of reflection, expansion, contraction, and shrinkage for a simplex in two dimensions (a triangle).

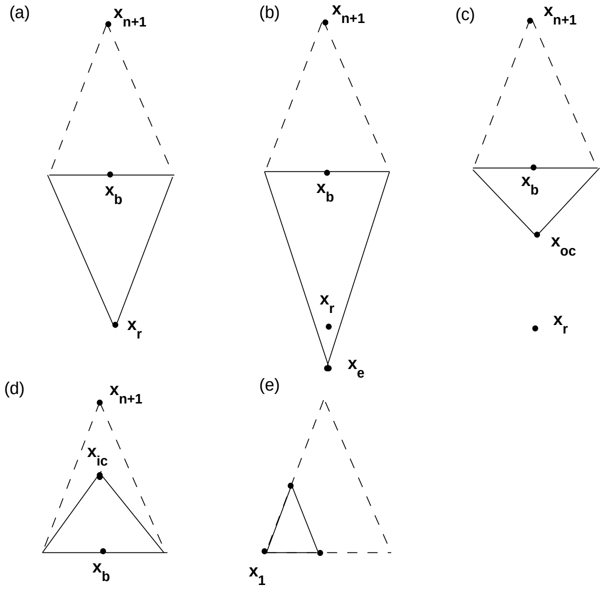


Figure 2.1: Possible outcomes for a step in the downhill simplex method in 2-D. The Nelder-Mead simplex at the beginning of the step, here a triangle, is shown with a dashed line. The simplex at the end of the step (shown with a solid line) can be either (a) a reflection away from the high point $\mathbf{x}_{n+1}$, (b) a reflection and expansion away from the high point, (c) an outside contraction along one dimension from the high point, (d) an inside contraction along one dimension from the high point, or (e) a shrink - a contraction along all dimensions toward the low point. An appropriate sequence of such steps will always converge to a minimum of the function.

It is important to note that except in a shrink, the one new vertex always lies on the (extended) line joining $\mathbf{x}_b$ and $\mathbf{x}_{n+1}$. Additionally the simplex shape undergoes a noticeable change during an expansion or contraction with the standard coefficients. Possible problems occur if the reflection gives a middling point. Tie-breaking rules are needed to order points in the case of equal function values. This is discussed in [3].

As stated in [3] and [11], the standard coefficients are $\rho = 1, \chi = 2, \gamma = 1/2$ and $\sigma = 1/2$. The maximum number of iterations depends on the number of variables. We iterate until the diameter of the simplex is less than a certain tolerance τ_1, *and* the function values differ from the minimum by less than another tolerance τ_2. These criteria might be fooled by a single anomalous step that failed to get anywhere. Therefore, it is often a good idea to *restart* a multidimensional minimisation routine at a point where it claims to have found a minimum. Therefore n of the $n + 1$ vertices of the simplex should be reinitialised by (2.1), with $\mathbf{P}_0$ being one of the vertices of the claimed minimum.

In [9] the method is shown to be effective and computationally compact. The problems we will consider are multidimensional, and the downhill simplex method is suitable in such a situation. It is not very efficient in terms of the number of function evaluations that it requires so it may be slower than other more complex methods. Nevertheless the method requires only function evaluations, not derivatives or gradients. It is suitable for nonlinear, locally linear problems, and is therefore suitable for the scope of this work.

2.3 A Brief Overview of Other Methods

There are many possible minimisation methods in multidimensions. *Direction set (Powell's)* methods use multiple one-dimensional line minimisations in n-space. The function's gradient need not be calculated. Directional vectors are used to find the minimum. The method cycles through the whole set of prescribed directions as many times as necessary, until the function stops decreasing. The aim is to find *conjugate directions*, that is, a set of n linearly independent, "non-interfering" directions. The two most famous examples are *Powell's quadratically convergent* method, and *Powell's method discarding the direction of largest decrease*. Powell's method is usually faster than the downhill simplex method in most applications.

Conjugate gradient methods in multidimensions calculate the gradient (vector of first partial derivatives) of a function. It is advantageous to use gradient information because a factor of n improvement in computational speed can be made. The *steepest descent* method is an example of conjugate gradients. The problem with it is that right-angle turns must be made, and this does not in general lead down to a minimum. The convergence is therefore slower than other methods. The *Fletcher-Reeves* and *Polak-Ribiere* methods are further examples, and both construct a sequence of directions.

Variable metric methods in multidimensions are sometimes called *quasi-Newton* methods - they are similar to conjugate gradient methods. In [8], the *limited-memory quasi-Newton* technique is preferred. They accumulate information from successive line minimisations to get the exact minimum of a quadratic form. The function's gradient must be computed, and an $n \times n$ Hessian matrix stored (conjugate gradient methods require storage of order n). Two examples are the *Davidon-Fletcher-Powell* (DFP) algorithm, and the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) algorithm. All of these methods are discussed further in [11], and we now discuss gradient methods in more detail.

2.3.1 Gradient Methods

A numerical weather forecast model involves a number of parameters that are determined empirically. The model is based on the integration of a dynamic system of partial differential equations modelling the behaviour of the atmosphere, in order to get a dataset. Parameter estimation is frequently carried out in data assimilation to find the solution to a constrained numerical weather prediction forecast model. Data assimilation is the process of recovering both the initial state and a set of forcing parameters from a real set of observations by fitting the model to observed data over an interval of time. The parameter set which gives the optimal solution is determined.

If we have some experimental data $x(t_i)$ and we want to fit a curve $y(t)$ to it, then it is relatively straightforward to minimise the difference between the two values at each time step in the case that $y(t)$ can be approximated by a polynomial. The gradients are also straightforward to compute. However, if y is a solution of an ODE, then it is more difficult to obtain the gradient of the difference function. One way this can be carried out is by using an *adjoint* method.

There are several examples of the use of adjoint parameter identification in the field of meteorology. According to [16], Chavant was amongst the first to propose parameter estimation using the adjoint method. He recognised that this inverse problem is often ill-posed, in that it is characterised by nonuniqueness and instability of the identified parameters.

2.3.2 An Example

To assess the impact of parameter estimation on a numerical model, Zou and Navon [16] present the continuous, forward forecast model

$$\frac{\partial \mathbf{X}}{\partial t} = \boldsymbol{f}(\mathbf{X}, \boldsymbol{\alpha}, t) \qquad \mathbf{X}_0 = \mathbf{X}(0) \tag{2.9}$$

with its full-physics adjoint, where $\mathbf{X}$ are the forecast variables and $\boldsymbol{\alpha}$ the parameter set. Three constant parameters as well as the initial condition are recovered.

Variational parameter identification involves minimising a cost function. A cost function measures the discrepancy between the observations and the corresponding model variables over time. Given constrained parameters α_i, the cost function in [16] is written as

$$\mathcal{J}(\mathbf{X}, \boldsymbol{\alpha}) = \frac{1}{2}\int_{t_0}^{t_R} \langle \mathbf{W}(\mathbf{X} - \mathbf{X}^{\exp}), (\mathbf{X} - \mathbf{X}^{\exp})\rangle dt + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{\alpha}), \tag{2.10}$$

where t_0 and t_R denote the assimilation window, $\langle\rangle$ denotes spatial integration (an inner product), $\mathbf{W}$ is a diagonal weighting matrix, $\mathbf{X}$ represents the state variable vector, $\mathbf{X}^{\exp}$ the observation vector, $\boldsymbol{\lambda}$ is the penalty coefficient vector and $\boldsymbol{g}(\boldsymbol{\alpha})$ is a function only of the violated constraints. The second term on the right-hand-side ensures that the retrieved parameters lie within prescribed bounds.

The adjoint of a model reduces the expense of the variational assimilation by calculating all of the components of the gradient of the cost function with respect to the initial conditions and $\mathbf{X}$ by one time integration. The decrease in the magnitude of the gradient of the cost function gives a better indication of how close the solution is to a minimum. In [16] the adjoint model is presented in the form

$$\frac{\partial \mathbf{P}}{\partial t} = -\left(\frac{\partial \boldsymbol{f}(\mathbf{X}, \boldsymbol{\alpha}, t)}{\partial \mathbf{X}}\right)^T \mathbf{P} - \mathbf{W}(\mathbf{X} - \mathbf{X}^{\exp}), \qquad \mathbf{P}(t_R) = 0, \tag{2.11}$$

where $\mathbf{P}$ represents the *adjoint variables.* Together with (2.9), (2.11) forms the simultaneous system of *Euler Lagrange* equations. Griffith and Nichols [2] provide a good analysis of the calculus of variations, and the derivation of the Euler Lagrange equations.

As carried out in [8], (2.11) is integrated backwards in time. The gradients of the cost function with respect to the initial conditions and parameters are calculated using

$$\begin{aligned}\nabla_{\mathbf{X}_0}\mathcal{J} &= \mathbf{P}(0) \\ \nabla\ \ \mathcal{J} &= \int_{t_0}^{t_R}\left[\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\alpha}}\right)^T \mathbf{P}\right]dt + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\alpha}}.\end{aligned} \tag{2.12}$$

Finally, a minimisation algorithm is used with the cost function and gradient values to find the optimal solution. Full details of how this is carried out can be found in [16].

The impact of an adjoint model was assessed by Navon in [7] and [16] by testing the impact of the model for a sufficiently long period, thus ensuring that no degradation of the forecast is caused by the optimally estimated parameter. The best forecasts were obtained by experiments in which all of the optimal values were used simultaneously. With a different model tested in [8], there was a significant reduction in both the cost function and the norm of its gradient during the minimisation process.

From [16], if a parameter is found to be very close to the estimated value, then it means either the initial guess is good, or the model is not very sensitive to the parameter. The returned value in parameter estimation therefore does not represent the true physical value. Rather, it represents the optimal value for the particular model.

The adjoint model can be difficult to derive. Variables can have eight orders of magnitude difference in such a meteorological problem, and it is hard to specify the weighting matrix **W**. Much more work is needed in this area. The creation of such an adjoint model is beyond the scope of this dissertation.

Within the scope of the project and the expected size of the problems considered, the gradient-free simplex method is preferable. Whilst gradient methods are faster, involve fewer function evaluations and are used for problems involving hundreds of parameters, they have to be defined separately for each problem. However, a general simplex method can be used for a variety of problems.

Chapter 3

Solving the Lorenz Equations Taking a Parameter Estimation Approach

3.1 A Model of the Atmosphere

In 1961 the scientist E.N. Lorenz was modelling the atmosphere, and investigating the predictability of the weather. His results disappointed weather forecasters, in that the solutions of the equations had a “sensitive dependence on initial conditions”. Small differences in initial conditions led to large differences in outcome - the much talked about ‘butterfly effect’. Lorenz noted from his findings that “precise very-long-range forecasting would seem to be non-existent” ([1], p223).

With certain parameters the Lorenz system is chaotic with no steady state. For the Lorenz system we retrieve parameters for a certain time, and carry out a longer forecast with the information obtained from the first forecast.

3.2 Lorenz Equations

From [4], the nonlinear Lorenz system is represented by three ODEs involving three parameters,

$$\begin{aligned}\frac{dx}{dt} &= -\sigma(x-y),\\ \frac{dy}{dt} &= \rho x - y - xz,\\ \frac{dz}{dt} &= xy - \beta z,\end{aligned} \tag{3.1}$$

where $x = x(t), y = y(t)$ and $z = z(t)$. According to Sparrow [14], they represent a 2-D fluid cell being warmed from below and cooled from above. The resultant convective motion is modelled by a partial differential equation. The quantity x measures the rate of convective overturning, y the horizontal temperature variation, and z the vertical temperature variation.

3.3 A Method to Retrieve Parameter Values

The method sets out to solve the differential equation

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}, t) = \mathbf{K}\boldsymbol{x} \tag{3.2}$$

by minimising the absolute difference between the numerical solution and experimental observations. However if experimental observations are not available, then suitable data must be generated. To do this we select the values of the parameter set P to be estimated. We then integrate the model forwards in time and minimise the approximation

$$\mathcal{J} = \frac{1}{\gamma}\left(\int \left(\mathbf{x}(t) - \mathbf{x}^{\exp}(t)\right)^2 dt\right) \approx \sum_{j=1}^{N_{\text{time}}} \frac{1}{\gamma_j}\left(\sum_{i=1}^{N_{\text{time}}}\left(x_j(t_i) - x_j^{\exp}(t_i)\right)^2\right) \tag{3.3}$$

where $\mathbf{x}$ represents the j components or species $(x_1, \ldots, x_N)$, $\mathbf{x}^{\exp}$ is the experimental data, and each species x_j is evaluated at time t_i. The values γ_j are chosen such that each component of the sum is of equal magnitude.

As discussed in Chapter 2 we use the downhill simplex Nelder-Mead method. The method starts at the initial guess of the vector of parameters P_0 and finds a local minimiser P of the supplied function. The function accepts input P and returns a scalar function value evaluated at P. The method iterates until a certain relative error tolerance in successive function evaluations is met. We hope that the final outputted value is as close as possible to the value of the experimental parameter set $P^{\exp}$. The idea behind this method is that if instead we have 'true', observed values of the variables rather than the known parameter values, the algorithm can be easily adapted so that the 'true' parameters are retrieved rather than a numerical solution of the governing equations. We now present the algorithm, written in MATLAB, which is implemented throughout this dissertation.

3.3.1 The Parameter Estimation Algorithm

Step One

Solve the system of ODEs using observed data and/or experimental values and known initial conditions. MATLAB's medium-order *ode45* solver can be used for non-stiff differential equations, otherwise *ode15s* works well for stiff systems.

Step Two

Input an initial estimate of the parameters and use the same ODE solver to calculate the new numerical solution values.

Step Three

Calculate the function value of the sum of the difference between the solution from the observed data and that from the estimated parameter values (see (3.3)).

Step Four

Use the MATLAB downhill simplex method *fminsearch* to iterate and find the parameter values which minimise the difference in Step Three. At the end of each iteration go back to Step Two and repeat Steps Three and Four until either (a) the difference falls below some tolerance band or (b) the maximum number of iterations is completed.

We solve the function in (3.1) using an ODE solver. It is a medium order method for non-stiff differential equations. It is an implementation of the explicit Runge-Kutta $(4, 5)$ pair of Dormand and Prince for approximating

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}, t), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{3.4}$$

by integrating the system from time t_0 to t_{final}.

The function argument to the method is a column vector corresponding to $\boldsymbol{f}(\boldsymbol{x}, t)$ so from (3.1) we have in the Lorenz case

$$\frac{d\boldsymbol{x}}{dt} \equiv \boldsymbol{x}' = \begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \boldsymbol{f}(\boldsymbol{x}, t) = \begin{cases} -\sigma(x_1 - x_2) \\ \rho x_1 - x_2 - x_1 x_3 \\ x_1 x_2 - \beta x_3 \end{cases} \tag{3.5}$$

where $x_1 = x, x_2 = y$ and $x_3 = z$.

Each row in the solution array of $\boldsymbol{x}$ corresponds to a time returned in the time column vector. Parameter values can also be passed to the ODE function. Thus, the ODE solver can be used to solve the system for both experimental data and repeating iterations with initial guesses of the parameters.

3.4 Results

For the experimental parameter values, numerical solutions oscillate apparently forever in a 'chaotic' manner. The ODEs determine a unique flow valid for all t. However, the Lorenz model is very sensitive to small changes in the parameters. Only slightly perturbed initial conditions in x and y with small enough step size lead to the numerical solutions for x and z becoming stable and decaying. There is no transient behaviour in the 'true' solution, where the trajectory continues to wind around, never settling down to periodic or stationary behaviour.

In the Data Assimilation Research Centre test case described in [4], the parameter set $[\sigma, \rho, \beta]$ is chosen to have the value [10, 28, 8/3]. These are taken to be the 'true' values. For the first run we make a 'guess' of the parameter values of $[x, y, z]$=[9, 30, 3] with an assimilation period of 20 seconds. The initial values of x, y and z are set to 2, as in [5]. The retrieved parameter values are [10.0163, 27.0641, 2.8356]. Clearly these are close to the 'true' values of [10,28,8/3]. The final value of the minimised function is 0.0910. Figure 3.1 shows these plots. The spikes in the plots of the difference between the solutions occur over very short times, so this means that the discrepancies are not significantly contributing to the sum.

The assimilation window is then doubled to 40 seconds. This process mirrors the system of data assimilation carried out in [15]. The final estimated parameter values are [9.8879, 27.7067, 2.9310]. The results of this run are shown in Figure 3.3.

From these tests it can be seen that the minimisation algorithm is reasonably accurate in reproducing the 'true' solution from the ODE solver. However as the assimilation period is increased, the retrieved parameter values lead to less accurate predictions. We see this by the fact that the function value after the same number of iterations for the doubled window is 0.3388, compared with 0.0910 for the shorter window. The solutions for the extended window are reasonably accurate for the first quarter (half of the original window), but the differences from the 'true' solutions increase with time. This is because the ODE method delivers less accuracy for problems integrated over longer intervals.

To conclude, we have a method which requires an initial guess of some 'known' parameter values. For the Lorenz system, the method produces more accurate estimates of the chaotic solutions which remain valid for the duration of the assimilation period. If the assimilation period is increased, the convergence becomes slower.

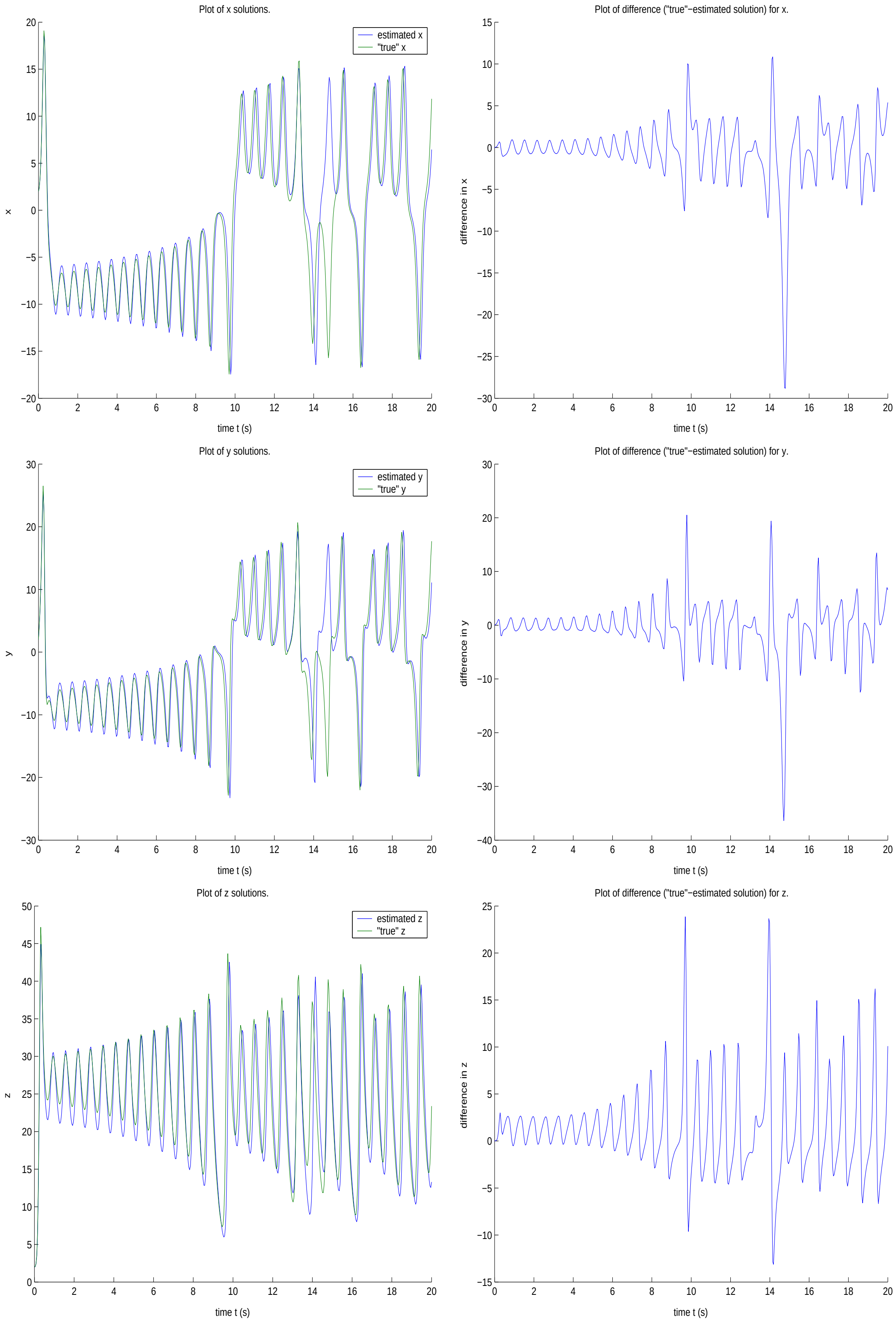
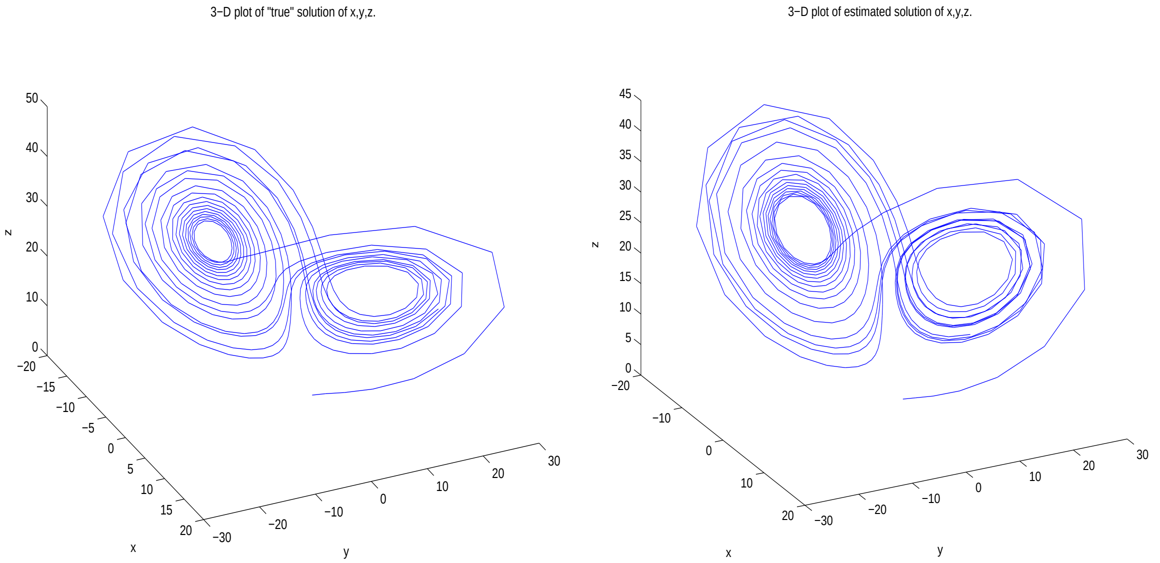


Figure 3.1: Plots of ‘true’ and estimated solutions of x, y and z, and the differences between the ‘true’ and estimated solutions, versus time.



3−D plot of "true" solution of x,y,z.

3−D plot of estimated solution of x,y,z.

Figure 3.2: Plots of 'true' and estimated solutions of x, y and z in 3-D.

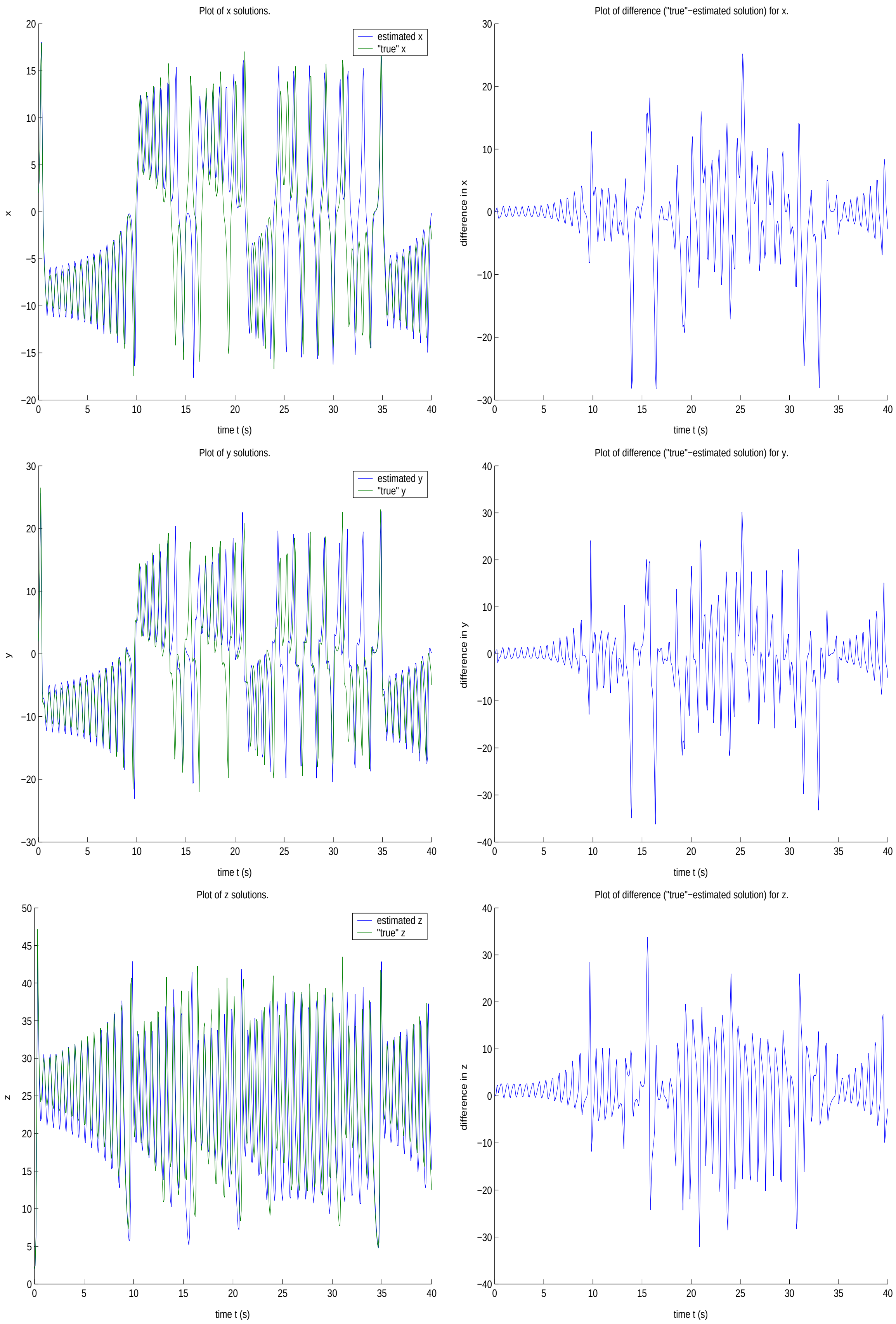


Figure 3.3: Plots of 'true' and estimated solutions of x, y and z, and the differences between the 'true' and estimated solutions, versus time, after doubling the assimilation window.

Chapter 4

Chemical Reactions

4.1 Introduction

At the end of Chapter 3 we made simple meteorological forecasts using a parameter estimation minimisation method. We will now turn to parameter estimation in another process which is relevant to the Food Industry. In this chapter we will consider a simple chemical reaction in order to be better able to deal with a larger batch test in the next chapter.

Chemical reactions are always occurring in the real world. Most of them involve proteins called *enzymes*. Enzymes are important in regulating any chemical processes, for example as catalysts or inhibitors in a reaction. Enzymes react selectively on definite compounds called *substrates*.

To understand the role of an enzyme in a food experiment, we need to consider *enzyme kinetics*. This is the study of rates of reactions, the temporal behaviour of the various reactants and the conditions which influence them. A biological example is given by Murray in [6]. The difficulty comes from the fact that the catalytic effectiveness of enzymes is reflected in the small concentrations needed in their reactions as compared with the concentrations of the substrates involved.

In order to qualitatively understand the processes involved in a chemical reaction, the development of a simplifying model is worthwhile. For such a model we should nevertheless use reaction mechanisms which are as realistic as possible. In this chapter we discuss a model reaction mechanism which will form the base of a more complicated process to be investigated in a later chapter.

4.2 Basic Enzyme Reaction

One of the most basic enzymatic reactions, first proposed by Michaelis and Menten and cited by Murray [6], involves a substrate S reacting with an enzyme E to form a complex SE which in turn is converted into a product P and the enzyme. We represent this schematically by

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} SE, \qquad SE \overset{k_2}{\rightarrow} P + E. \tag{4.1}$$

Here k_1, k_{-1} and k_2 are constant parameters associated with the rates of reaction. The symbol $\rightleftharpoons$ states that the reaction is reversible while the single arrow $\rightarrow$ indicates that the reaction can go only one way. Such one-way reactions often allow us to write the system as a relatively straightforward mathematical matrix system of equations.

The overall mechanism is a conversion of the substrate S, via the enzyme catalyst E, into a product P. The *Law of Mass Action* says that the rate of a reaction is proportional to the product of the concentration of the reactants. Applied to (4.1), it leads to one equation for each reactant and hence the system of nonlinear equations

$$\frac{ds}{dt} = -k_1 es + k_{-1}c, \qquad \frac{de}{dt} = -k_1 es + (k_{-1} + k_2)c$$
$$\frac{dc}{dt} = k_1 es - (k_{-1} + k_2)c, \qquad \frac{dp}{dt} = k_2 c \tag{4.2}$$

where the concentrations of the reactants S, E, SE and P are denoted by s, e, c and p respectively. The k's, called *rate constants*, are constants of proportionality. We also require initial conditions which we take here as those at the start of the process which converts S to P, so

$$s(0) = s_0, \qquad e(0) = e_0, \qquad c(0) = 0, \qquad p(0) = 0. \tag{4.3}$$

The solutions of (4.2) with (4.3) then give the concentrations, and hence the rates of the reactions, as functions of time. Equations (4.2) are not all independent. The last equation can be decoupled from the first three, in that p can be determined once $c(t)$ is known. In the mechanism (4.1) the enzyme E is a catalyst, which only facilitates the reaction, so its total concentration, free plus combined, is a constant. As in [6], this conservation law for the enzyme leads to the system of ordinary differential equations reducing to only two for s and c,

$$\begin{aligned} \frac{ds}{dt} &= -k_1 e_0 s + (k_1 s + k_{-1})c \\ \frac{dc}{dt} &= k_1 e_0 s - (k_1 s + k_{-1} + k_2)c, \end{aligned} \tag{4.4}$$

with initial conditions $s(0) = s_0, c(0) = 0$.

In [6], Murray goes further to nondimensionalise the system (4.4) with the initial conditions. This is a very common technique but it is not used in this work. Unlike the meteorological example discussed in Chapter 2, we already have the initial conditions so these do not have to be retrieved.

4.3 Method to Retrieve Parameter Values

We adapt the method used to approximate the Lorenz system numerically with the right-hand-side of the equations obviously changed. The same algorithm as that in Section 3.3.1 is used.

From the original reaction (4.1), which converts S into a product P, we have that the final steady state is both the substrate and the substrate-enzyme complex concentrations are zero. We are interested here in the time evolution of the reaction so we need the solutions of the nonlinear system (4.4), which we cannot solve analytically in a simple closed form. However we can see what the numerical solutions of s and c look like qualitatively using our method's ODE solver.

The function argument to the method must return a column vector corresponding to $\boldsymbol{f}(\boldsymbol{x}, t)$ so from (4.4) we have in this case

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{x}' = \begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = f(t, \boldsymbol{x}) = \begin{cases} -k_1 e_0 x_1 + (k_1 x_1 + k_{-1}) x_2 \\ k_1 e_0 x_1 - (k_1 x_1 + k_{-1} + k_2) x_2 \end{cases} \tag{4.5}$$

where $x_1 = s$ and $x_2 = c$. We try to fit parameters to get the model to match the observations. We minimise a similar norm function as that in (3.3).

4.4 Results

The initial values of s and c are chosen to be 1 and 0 respectively. The 'true' value of the parameter set $[k_1, k_2, k_{-1}, e_0]$ is taken to be [2, 0.8, 0.7, 0.4]. There is no reason why such figures should be chosen over numbers of a greater order of magnitude. Nor is it known whether these values are realistic. However the aim of the exercise is to retrieve values as close to these as possible using the simplex method.

As a perturbation to the selected values, the initial guess of the parameters is set to [3, 0.7, 0.8, 0.6] in the method so that they are all of the same order of magnitude. After iterating, the method returns the initial 'true' parameter values [2, 0.8, 0.7, 0.4]. Therefore the 'true' and retrieved solutions are the same to a certain degree of accuracy, as shown in Figure 4.1. Referring to the solution obtained for c in Figure 4.1, many time steps have been taken between the '×' marks. However, we have chosen to plot a low number of points representing the experimental data points.

The differences between the 'true' and estimated solutions increase with time, as is to be expected. Nevertheless, the magnitude of the differences for both s and c is small. Comparing these solutions qualitatively with the equivalent dimensionless quantities in Figure 5.1 of [6], we see that they are very similar. The final value of the minimised norm function is 5.7643×10^{-7}. Therefore we conclude that our minimisation technique is sufficient to provide good approximations to a system of ODEs.

Further tests were carried out, with guesses further away from the picked 'true' parameter values. For a parameter set an order of magnitude greater than the true values, such as [20, 14, 10, 12], the method is unable to meet integration tolerances without reducing

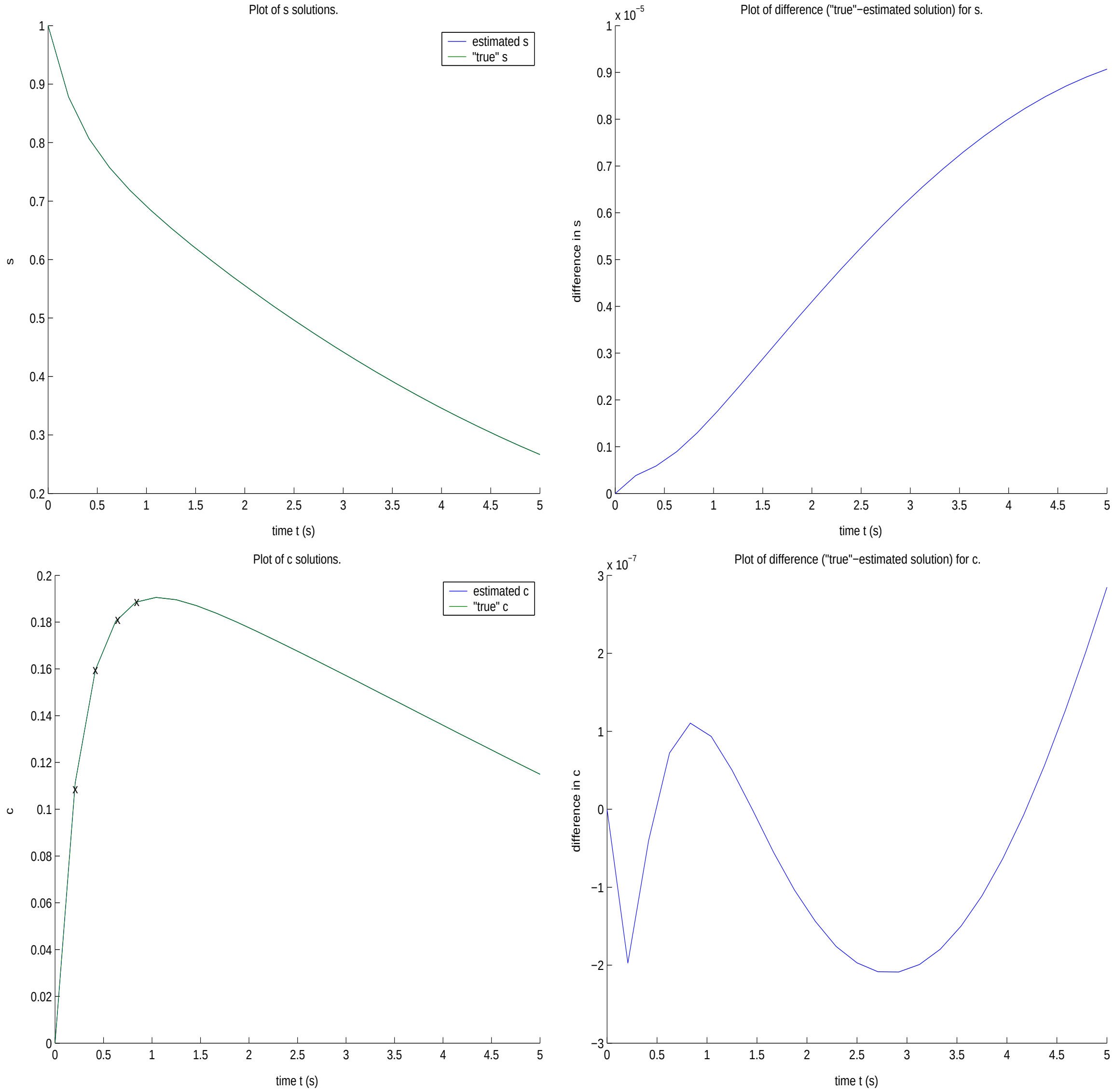


Figure 4.1: Plots of ‘true’ and estimated solutions for concentrations s of substrate S and c of complex SE, and the differences between the ‘true’ and estimated solutions, over a period of 5 seconds.

the step size below the smallest value allowed at time t. Therefore, as we would expect, the closer the initial guess to the ‘true’ values, the more accurate the retrieved parameters are.

As stated in Chapter 3, if we had no information on the true parameters $[k_1, k_2, k_{-1}, e_0]$ but instead only the observed concentration values s and c at certain times, then we could use an initial selection of the concentrations and obtain the optimal parameter values.

Enzymes are very effective in small concentrations compared with concentrations of substrates. This makes the system less straightforward to solve. It is possible to obtain a very accurate asymptotic solution by looking for regular Taylor expansion solutions to s and c. As shown in [6], a small parameter relating to the initial conditions of the enzyme and substrate multiplies the nondimensionalised form of (4.5). Therefore it is a *singular perturbation problem*. If the parameter is very small then the order of the system is reduced and it cannot in general satisfy all the initial conditions. Singular perturbation techniques can determine asymptotic solutions of such systems for small parameters.

Since the solution does not satisfy the initial conditions for small parameters, at least one of the solutions for s or c is not an analytic function of the parameter as it tends to zero. More appropriate timescales are needed around $t = 0$. The effect is to magnify the neighbourhood of $t = 0$. The thin layer near $t = 0$ is sometimes called the *boundary layer* and is the domain where there are very rapid changes in the solution. It arises because of the mixture of slow and fast length scales.

Murray describes the singular perturbation method. There are two parts to the resulting solution - the *singular* or *inner* solution for s and c and is valid for small t, and the *nonsingular* or *outer* solution valid for all t not in the immediate neighbourhood of $t = 0$. Whilst in the case presented in [6] the singular solutions are determined completely, this is not generally the case in singular perturbation problems.

The rapid change in the substrate-enzyme complex takes place in a very short time. As all the action is at the start, matching does not get any more difficult for the remainder of the period. For many experimental purposes it is not measurable. Thus in many experiments the singular solutions are never observed. The reaction for the complex is essentially in a steady state. That is, the reaction is so fast it is more or less in equilibrium at all times. This is Michaelis and Menten's *pseudo-steady state hypothesis*. Therefore, it is important to note that something is lost in applying experimental results to a theory which cannot satisfy all the initial conditions.

Chapter 5

The Food Problem

5.1 Introduction

The motivation for this section is work carried out by Professor Leo Pyle et al. of the University of Reading's Food Biosciences Department. His paper [12] summarises a simplified approach to quantitative modelling of the so-called Maillard reaction. It is for this reaction which we wish to retrieve parameter values. Whilst the boundaries of the reaction network are the same as those reported by Desclaux (cited in [12]), the reaction scheme initially posed is a simplification. In the full version 14 reagents and intermediates are included in the model, with 19 independent parameters to be estimated and fitted in the equations.

After private communications with Leo Pyle [13], it was decided to further simplify the reaction scheme to consider only the five main reagents and intermediates. This approach was taken for two reasons. The first was to avoid problems with non-measurable state variables. Secondly it allowed for quicker solutions to be obtained, so that the method itself could be tested thoroughly, rather than its completeness. As will be seen later, including more parameters in a minimisation algorithm reduces the accuracy of the method.

5.1.1 Definitions

An *amino acid* is an organic compound containing an amino group (NH_2), a carboxylic acid group (COOH), and any of various side groups, especially any of the twenty compounds that have the basic formula $NH_2CHRCOOH$. The groups are linked together by bonds to form proteins, or that function as chemical messengers and as intermediates in metabolism.

In this context, a *stoichiometric* relationship is assumed between reactants and products in a chemical reaction. *Arthromyces ramosus peroxidase* (ARP) is an enzyme involved in the chemical reaction which has a crystal structure. The experiments are pH-dependent. Therefore *iso-pH* refers to a uniform numerical value of the pH level.

Molar mass is a unit that enables the weight of any chemical substance to be calculated, be it an element or a compound. Molar mass is the sum of all of the atomic masses in a formula. If a substance is pertaining to, or formed from, two molecules, then it is said to be *bimolecular*. For example a bimolecular reaction is a reaction between two molecules.

5.2 Limitations and Assumptions in the Model

The main assumptions made in developing this model are outlined below.

1. The amino acid is assumed to play no direct role as a reagent in the first stages of the Maillard reactions.

2. All reactions are assumed to follow first order kinetics with stoichiometric coefficients equal to one.

3. The rate of reaction of the ARP intermediates is assumed to be very fast so that the ARP concentration equals zero. This allows us to simplify the model.

4. The model assumes batch, well-mixed, isothermal, iso-pH conditions (so that rate constants do not change during a particular experiment). These are significant assumptions.

5. The model is written in molar units; species concentrations are thus in kmolem^{-3}; rate constants have units mins^{-1}.

6. The model is first order, when in fact a bimolecular, second order model would be more accurate.

One of the main problems is that the data is limited for the measurable components. Some of the intermediates have very low concentrations so are hard to capture, and thus the experimental data which we have tried to fit to predicted measurements to estimate the parameters will be flawed.

5.3 Current Strategy to Test Models and Estimate Parameters

It is important to understand the approach currently undertaken by the Food Biosciences Department to estimate the parameters in a batch reaction, so that possible improvements can be put forward.

The following models are tested sequentially against the individual concentration-time profiles. The parameters are currently estimated in two ways. Firstly, using the modelling package ASPEN, the ODEs are solved simultaneously by assuming mass balance on each component. If unexpected results are obtained, then the parameters are often forced by the user. The second method involves minimising the error sequentially, with the use of curve-fitting to give improved forecasts. The solution is assumed to be parabolic. A fourth-order Runge-Kutta scheme is used.

Reactions are carried out at certain temperatures with a pH level assumed to be constant, and known initial concentrations up to a maximum temperature. In practice time is needed to reach a certain temperature so more data is required. Temperature effects will be included by assuming Arrhenius kinetics (see Section 5.5); estimations of the current activation energies will be derived from the models and the data at different temperatures.

Experiments are carried out at different final temperatures. The species are contained in metal test tubes fitted into an electrically-heated block, alongside a dummy experiment. Synchronous time measurements are then made. When the temperature reaches 120°C, the power is turned on to full. The block cannot be preheated. At 100°C, the automatic controller is switched on, which holds the temperature steady without any heating effects.

5.4 The Simplified Chemical Reaction

We start with a reagents pair, A being the sugar *xyclose* and B an amino acid. We assume no other species are initially present. The start of the batch reaction network is taken to be of the form shown in Figure 5.1.

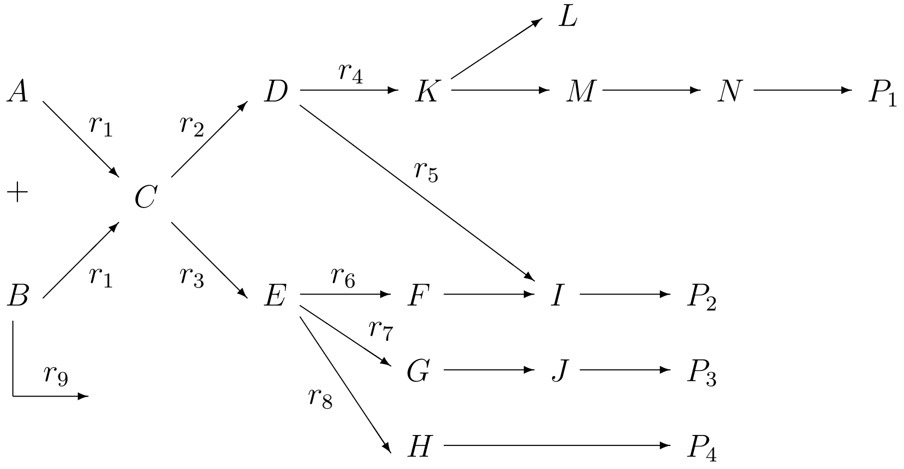


Figure 5.1: A first (simplified) version of the Maillard reaction.

The species C to N are some of the intermediates in the Maillard reaction, with products P_1 to P_4. The sugar and amino acid have initial concentrations $x_A(0)$ and $x_B(0)$ respectively and all other initial concentrations of the intermediates are zero in all of the modelling.

We assume that all the components are measurable, so that the balance on each component gives n ODEs for the concentrations x of the various species of the form

$$\begin{aligned} \frac{dx_A}{dt} &= -r_1 \\ \frac{dx_B}{dt} &= -r_9 \\ \frac{dx_C}{dt} &= r_1 - r_2 - r_3 \\ &\vdots \end{aligned}$$

where

$$r_1 = k_1 x_A x_B$$

and all other r's are first order

$$r_j = k_j x_j \tag{5.1}$$

with x_j being the concentration of the j-th reagent/intermediate, and k_j the corresponding rate constant.

What is of interest chemically is the *rate of reaction*, or the rate of uptake; that is dx/dt when $x(t)$ has been found. It is usually determined experimentally by measuring the dimensional substrate concentration at various times, then extrapolating back to $t = 0$ and the magnitude of the initial rate calculated.

5.5 Theory of Reaction Rates and Activation Energies

The theory for the food problem has been developed using work carried out on the *distributed activation energy model* (DAEM), with reference to [10]. It is applicable because, as with the sugar and amino acid in Figure 5.1, the behaviour of the numerous reactants in the DAEM is described by a distribution of activation energies. Methods are therefore needed to evaluate solutions quickly and efficiently. The solution method provides a rapid and highly effective method appropriate to parameter estimation and distribution function estimation.

In [10], the model describes the pyrolysis of various coals under differing temperature histories. The calculation of solutions to this model creates significant numerical difficulties.

The problem involves a single block of coal and the time evolution of its constituent parts, averaged over the whole block. Similar to the reagents and intermediates in Figure 5.1, the coal constituents are numbered $j = 1...n$, and for a first order rate of change of mass (pyrolysis), the reaction rate k_j (or rate constant as used in (5.1)) is taken to be Arrhenius in form,

$$k_j(t) = k_{0_j} e^{-\Delta E_j / RT(t)}, \tag{5.2}$$

where k_{0_j} is the pre-exponential or frequency factor (mins^{-1}), ΔE_j is the apparent activation energy for constituent j (kJ/mol), $R = 8.3$ is the universal gas constant (J/mol/K), and $T(t)$ is the time-dependent absolute temperature of the coal (K). We therefore take k_j in (5.1) to be of the form in (5.2) throughout this paper.

If $j = 1$ then the model is referred to as the *single first-order reaction model* (SFOR). In contrast the DAEM allows for a more complicated set of reactions by considering a continuous distribution of reactants.

Applying this model to the food problem where the volatiles are now food chemicals, we model the activation energies ΔE_j to only depend on the j-th species itself, and not on the temperature. The same is true of the values of the pre-exponential factors k_{0_j}. In practice these quantities *do* depend on temperature. The rates at which species disappear are the main time-dependent quantities. We aim to estimate the (constant) parameters k_{0_j} and ΔE_j so that the rate constant k_j can be found for each species j. We aim to do this over a range of input concentrations x_A and x_B, and temperatures T.

To do this, we can firstly assume the initial distribution of chemicals and the pre-exponential factors $k_0(E)$ and then find the resulting time dependence of the chemicals. Alternatively we can solve the inverse problem, where the reaction rates dx/dt are measured and the distribution of chemicals must be determined. This second problem is one of parameter estimation and as discussed in [10] there are significant difficulties in determining accurately both $k_0(E)$ and E as they are highly correlated. The important point is that we are not measuring the often coupled k_0 or ΔE values. Rather, we get the data from the experiments and infer the reaction rates.

5.6 The Models and Their Solutions

5.6.1 Model 1

The first step to estimate the parameters involved in the batch reaction in Figure 5.1, is to make the simplification of considering only the reagents A and B and intermediates C to E, as shown in Figure 5.2.

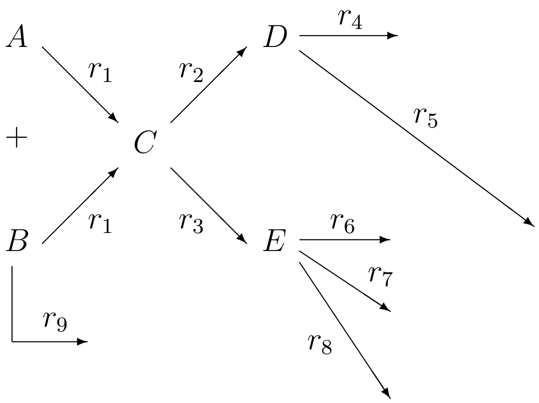


Figure 5.2: Chemical reaction for Model 1.

Using the notation used in Section 5.4 and with the assumptions of Section 5.2, the rates of change of these five species concentrations are given by

$$\begin{aligned}
\frac{dx_A}{dt} &:= \frac{dx_1}{dt} = -r_1 = -k_1 x_A x_B = -k_{0_1} e^{-\Delta E_1/RT} x_A x_B \\
\frac{dx_B}{dt} &:= \frac{dx_2}{dt} = -r_9 = -k_9 x_B = -k_{0_9} e^{-\Delta E_9/RT} x_B \\
\frac{dx_C}{dt} &:= \frac{dx_3}{dt} = r_1 - r_2 - r_3 = k_1 x_A x_B - k_2 x_C - k_3 x_C \\
&\quad = k_{0_1} e^{-\Delta E_1/RT} x_A x_B - \Big(k_{0_2} e^{-\Delta E_2/RT} + k_{0_3} e^{-\Delta E_3/RT} \Big) x_C \\
\frac{dx_D}{dt} &:= \frac{dx_4}{dt} = r_2 x_C - r_4 x_D - r_5 x_D = k_2 x_C - k_4 x_D - k_5 x_D \\
&\quad = k_{0_2} e^{-\Delta E_2/RT} x_C - \Big(k_{0_4} e^{-\Delta E_4/RT} + k_{0_5} e^{-\Delta E_5/RT} \Big) x_D \\
\frac{dx_E}{dt} &:= \frac{dx_5}{dt} = r_3 x_C - r_6 x_E - r_7 x_E - r_8 x_E = k_3 x_C - k_6 x_E - k_7 x_E - k_8 x_E \\
&\quad = k_{0_3} e^{-\Delta E_3/RT} x_C - \Big(k_{0_6} e^{-\Delta E_6/RT} + k_{0_7} e^{-\Delta E_7/RT} + k_{0_8} e^{-\Delta E_8/RT} \Big) x_D. \qquad (5.3)
\end{aligned}$$

It must be noted that the final products or final transient intermediates are not needed in this simplified set of equations. They do not appear in the ODEs, but ideally the rates at which they form should be measured by changing the temperature or initial substances. In total there are 18 parameters included in this first model - 9 pre-exponential factors, and 9 activation energies.

5.6.2 Testing of Model 1

We use a similar algorithm to that used for the simple chemical reaction (4.4) and detailed in Section 3.3.1. Estimates of the 18 parameter values are made and the method solves the system of ODEs numerically to obtain the 'correct' values of the parameters. Then the simplex method minimises the sum of the squares between the guessed and 'true' concentration solutions.

In order to motivate the systematic simplifications of the equations, it is useful to consider the typical values of the parameters and functions on which it depends. From [10], for the coal pyrolysis problem, the pre-exponential factors are typically in the range $k_0 \sim$

$10^{10} - 10^{13}$ s^{-1}, while the activation energies of interest are in the region $1 \times 10^5 - 3 \times 10^5$ J/mol. However the temperatures for the coal are much greater than for those for the sugar and amino acid. Concentrations are in molar masses, to which an order of 10^{-4} mol/litre can be detected.

At this point in the investigation no 'correct' parameter values were known. Therefore estimates had to be made, and then perturbations away from these estimates inputted. As stated in [10], a common assumption to solve the equations is to take all the pre-exponential factors, k_{0_j} to have the same value k_0. This simplifies much of the later analysis and is reasonable given much of the uncertainty over the reactant distributions. Therefore, to model the behaviour of the solutions qualitatively, the values of k_0 and ΔE were taken to give rate constant k values equal to the order of unity. The picked values are displayed in Table 5.1, together with the retrieved values.

From [13] it was found that the initial concentrations of the sugar and amino acid are 0.24 mol/l. Testing is carried out from typical temperatures of $T_0 \approx 50^{\circ}$C to an absolute maximum $T_1 \approx 140^{\circ}$C. The tests run for up to two hours. The temperature T in (5.3) is ramped up in the code with time using the first order (exponential) rise

$$T(t) = T_1 - (T_1 - T_0)e^{t/\tau} \tag{5.4}$$

with constant *time constant* τ. The experiment usually reaches T_1 after between 5 and 10 minutes, so a small value of τ of 3 minutes was set for a small rise time and a quickly-isothermal test. Such an exponential rise over two hours is shown in Figure 5.3. In this case and for all of our tests, the start temperature T_0 is taken to be 323K ($\approx 50^{\circ}$C), and the final temperature T_1 to be 393K ($\approx 120^{\circ}$C).

The value of R, the universal gas constant, is taken to be 8.3 J/mol/K, with each value of $k_{0_j} = 1 \times 10^{-2}$mins^{-1}, and $\Delta E_j = 2.5 \times 10^5$ kJ/mol. All of our models for food testing experiments are integrated forwards in minutes.

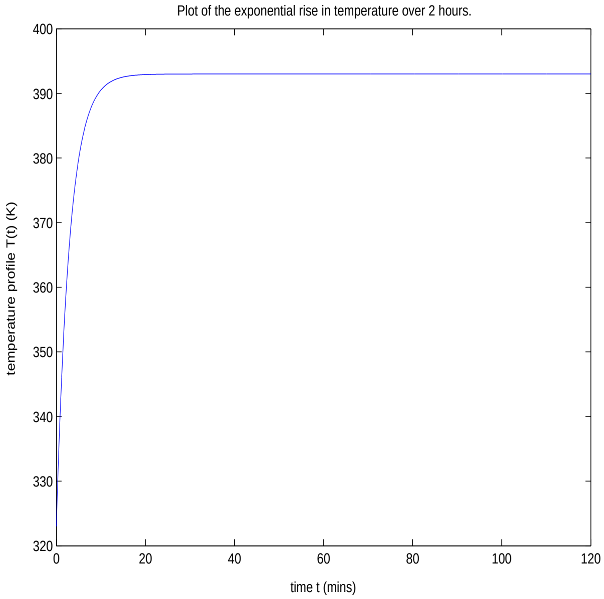


Figure 5.3: Plot of the exponential rise in temperature over time incorporated in the model.

The method minimises in a least squares sense the approximation

$$\begin{aligned} \mathcal{J} &= \frac{1}{(x^{\max})^2}\int (x - x^{\exp})^2. \\ &\approx \frac{1}{N}\sum_i \left(\frac{x_1(i) - x_1^{\exp}(i)}{x_1^{\max}(i)}\right)^2 + \ldots + \frac{1}{N}\sum_i \left(\frac{x_5(i) - x_5^{\exp}(i)}{x_5^{\max}(i)}\right)^2, \end{aligned} \tag{5.5}$$

where N is the size of the vectors holding the concentrations of each species (the number of times the concentrations are measured), and $x^{\exp}$ is the experimental data. There are five species x_1 to x_5 where x is the iterated solution. The maximum refers to the picked parameters (the experimental data). This weights the sum depending on its maximum value.

5.6.3 Interpretation of Results of Model 1

The picked 'true', initial guess and retrieved parameter values (to five significant figures) for the model in Figure 5.2 are displayed in Table 5.1.

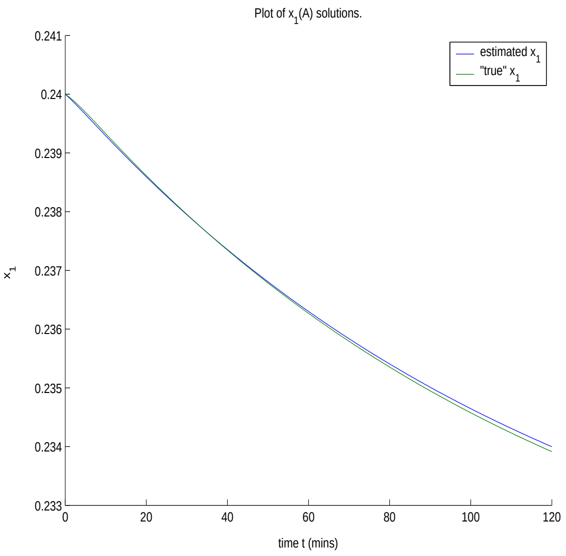
Parameter	‘True’ value	Guess	Retrieved value
k_{0_1}	1×10^{-2}	2×10^{-2}	4.5663×10^{-3}
k_{0_2}	1×10^{-2}	3×10^{-2}	1.8809×10^{-2}
k_{0_3}	1×10^{-2}	2×10^{-2}	2.1136×10^{-2}
k_{0_4}	1×10^{-2}	1.3×10^{-2}	3.0249×10^{-3}
k_{0_5}	1×10^{-2}	1.1×10^{-2}	1.8792×10^{-2}
k_{0_6}	1×10^{-2}	2×10^{-2}	1.0531×10^{-2}
k_{0_7}	1×10^{-2}	3×10^{-2}	1.8394×10^{-2}
k_{0_8}	1×10^{-2}	2×10^{-2}	1.9823×10^{-2}
k_{0_9}	1×10^{-2}	1.3×10^{-2}	2.0455×10^{-4}
E_1	2.5×10^5	3×10^5	1.3593×10^5
E_2	2.5×10^5	1.5×10^5	9.7770×10^4
E_3	2.5×10^5	1×10^5	1.8939×10^4
E_4	2.5×10^5	2.8×10^5	1.1881×10^5
E_5	2.5×10^5	2×10^5	4.1831×10^5
E_6	2.5×10^5	2.7×10^5	1.5530×10^5
E_7	2.5×10^5	2.2×10^5	2.4164×10^5
E_8	2.5×10^5	3.2×10^5	1.9256×10^5
E_9	2.5×10^5	2.8×10^5	2.1456×10^5

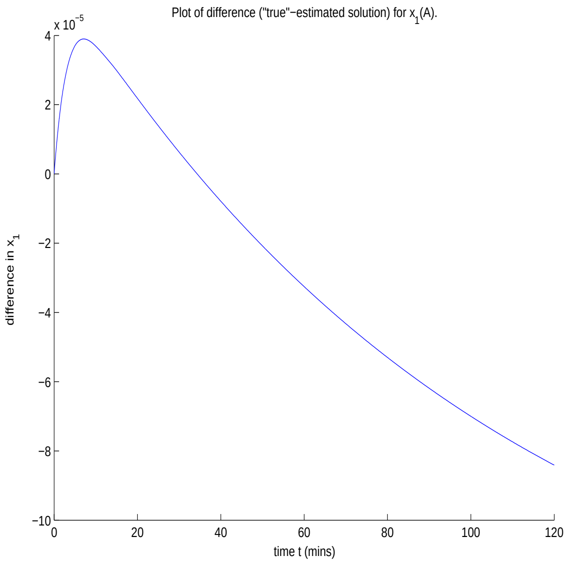
Table 5.1: Parameter input and output values for Model 1.

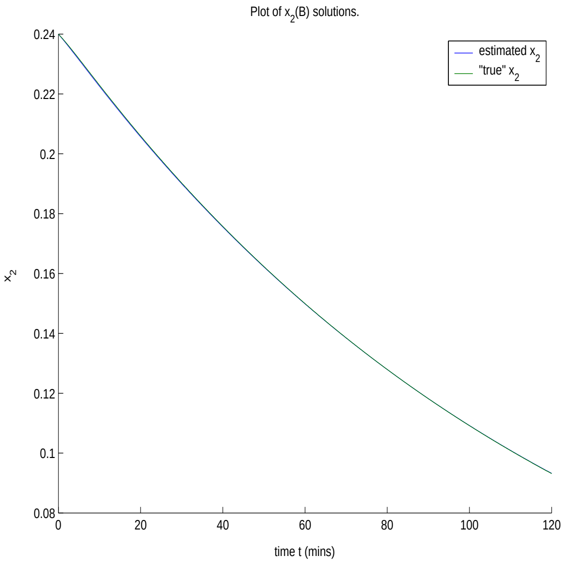
The smaller the value of k_0 and therefore the rate constant k_j, the faster the reaction is. A low value of ΔE means a non-varying system. With the exception of k_{0_1}, k_{0_4}, k_{0_9}, E_2 and E_3, all the retrieved parameters are of the same order of magnitude as those taken to be the ‘true’ values. This means either our initial guess was realistic, or the parameters do not have a large impact on the model, or the values are compensating for errors elsewhere in the solutions.

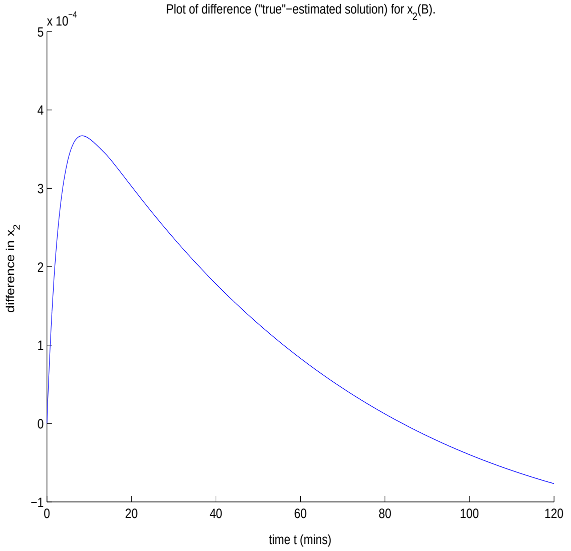
The results obtained are presented in Figure 5.4. The model reaches a steady state after a certain time period. Whilst the issues of identifiability and uniqueness are ever present, and there is variability between the ‘actual’ and retrieved numerical solutions, even with noisy observations the parameters are recovered to an acceptable degree of accuracy.

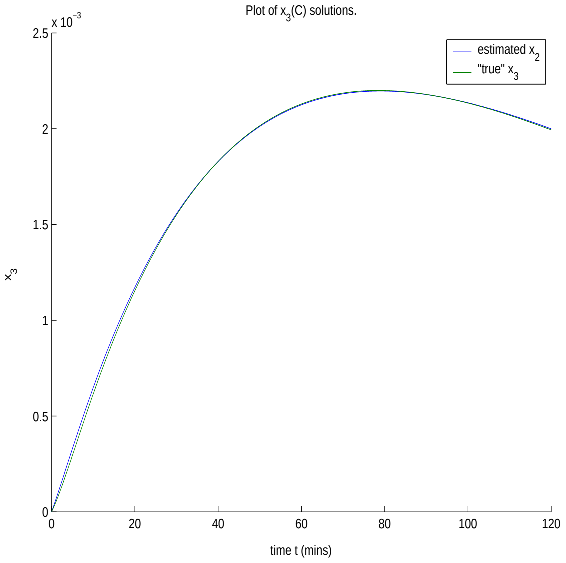
According to Navon [7], if an optimally estimated parameter attains unphysical values, one can deduce that either an overfitting of the data took place, or that this parameter is not identifiable with the data available.

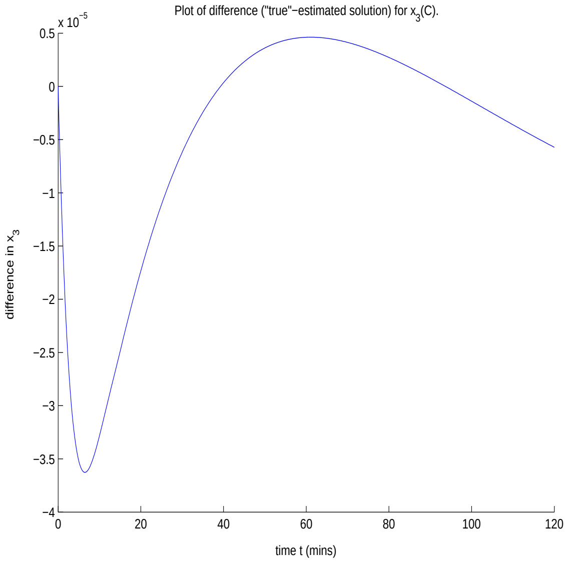

Plot of x_1(A) solutions.
estimated x_1
"true" x_1
x_1
0.241
0.24
0.239
0.238
0.237
0.236
0.235
0.234
0.233
0
20
40
60
80
100
120
time t (mins)

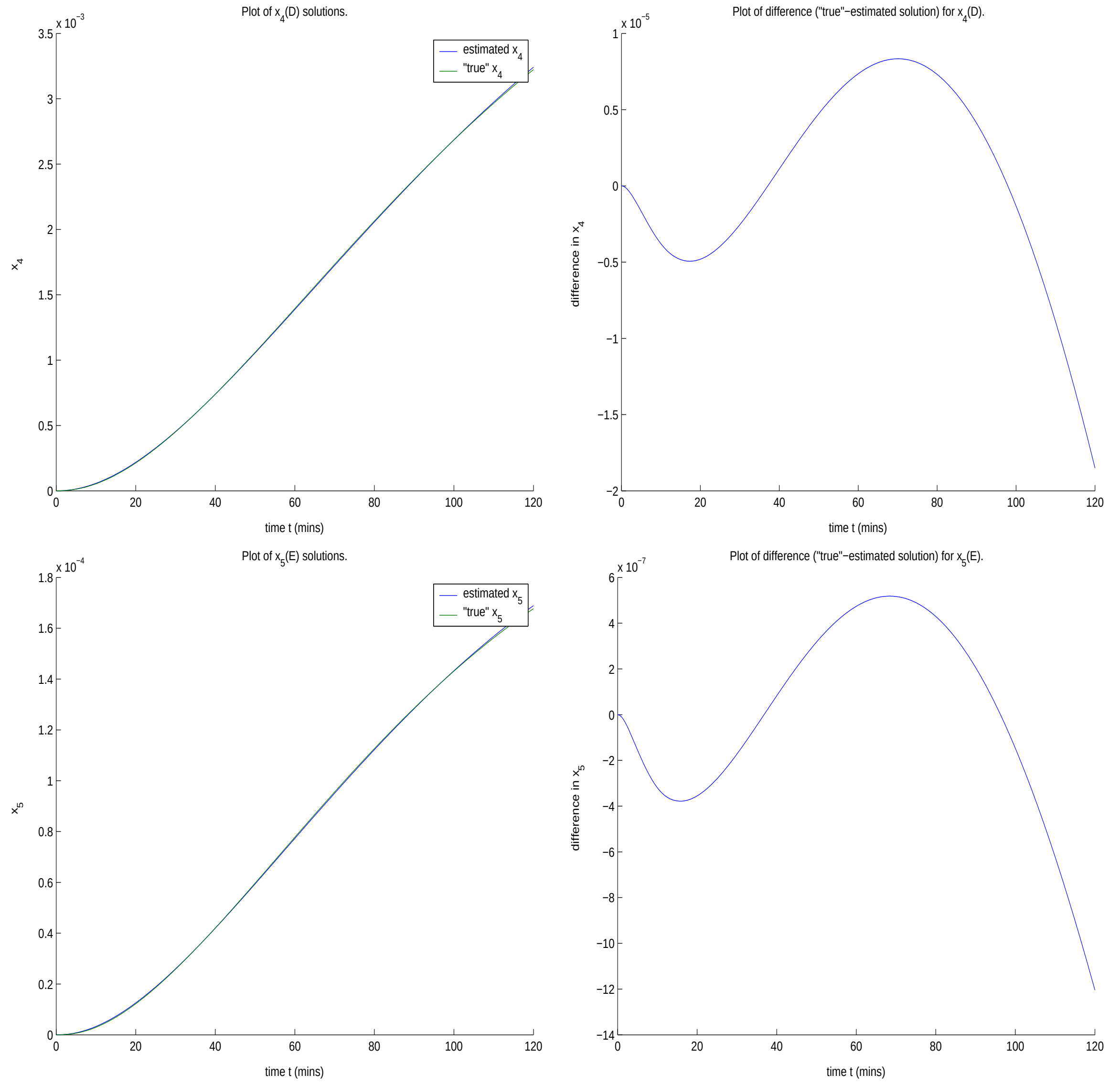

Plot of difference ("true"−estimated solution) for x_1(A).
x 10^{-5}
difference in x_1
4
2
0
−2
−4
−6
−8
−10
0
20
40
60
80
100
120
time t (mins)


Plot of x_2(B) solutions.
estimated x_2
"true" x_2
x_2
0.24
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0
20
40
60
80
100
120
time t (mins)


Plot of difference ("true"−estimated solution) for x_2(B).
x 10^{-4}
difference in x_2
5
4
3
2
1
0
−1
0
20
40
60
80
100
120
time t (mins)


Plot of x_3(C) solutions.
x 10^{-3}
estimated x_2
"true" x_3
x_3
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
120
time t (mins)


Plot of difference ("true"−estimated solution) for x_3(C).
x 10^{-5}
difference in x_3
0.5
0
−0.5
−1
−1.5
−2
−2.5
−3
−3.5
−4
0
20
40
60
80
100
120
time t (mins)


Plot of x_4(D) solutions.

Plot of difference ("true"−estimated solution) for x_4(D).

Plot of x_5(E) solutions.

Plot of difference ("true"−estimated solution) for x_5(E).

Figure 5.4: Plots of estimated and 'true' solutions for species $A(x_1)$, $B(x_2)$, $C(x_2)$, $D(x_4)$ and $E(x_5)$, and the differences between the 'true' and estimated solutions, versus time.

5.6.4 Model 2

After additional private communications [13], a further simplification to the model in Figure 5.2 was made, as depicted in Figure 5.5.

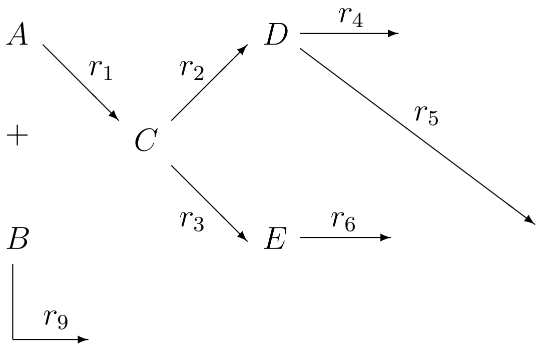


Figure 5.5: Chemical reaction for Model 2

The parameters corresponding to k_7 and k_8 have been made redundant. With the same assumptions, the rates of change of the species concentrations are given by the set of *linear* equations

$$\begin{aligned}
\frac{dx_A}{dt} &= -r_1 = -k_1 x_A = -k_{0_1} e^{-\Delta E_1/RT} x_A \\
\frac{dx_B}{dt} &= -r_9 = -k_9 x_B = -k_{0_9} e^{-\Delta E_9/RT} x_B \\
\frac{dx_C}{dt} &= r_1 - r_2 - r_3 = k_1 x_A - k_2 x_C - k_3 x_C \\
&= k_{0_1} e^{-\Delta E_1/RT} x_A - \left(k_{0_2} e^{-\Delta E_2/RT} + k_{0_3} e^{-\Delta E_3/RT} \right) x_C \\
\frac{dx_D}{dt} &= r_2 x_C - r_4 x_D - r_5 x_D = k_2 x_C - k_4 x_D - k_5 x_D \\
&= k_{0_2} e^{-\Delta E_2/RT} x_C - \left(k_{0_4} e^{-\Delta E_4/RT} + k_{0_5} e^{-\Delta E_5/RT} \right) x_D \\
\frac{dx_E}{dt} &= r_3 x_C - r_6 x_E = k_3 x_C - k_6 x_E \\
&= k_{0_3} e^{-\Delta E_3/RT} x_C - \left(k_{0_6} e^{-\Delta E_6/RT} \right) x_E. \qquad (5.6)
\end{aligned}$$

The set of equations (5.6) can be written more compactly in vector-matrix form as

$$\frac{d\boldsymbol{x}}{dt} = \mathbf{A}\boldsymbol{x}. \qquad (5.7)$$

The system matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \begin{pmatrix} -k_1 & 0 & 0 & 0 & 0 \\ 0 & -k_9 & 0 & 0 & 0 \\ k_1 & 0 & -(k_2+k_3) & 0 & 0 \\ 0 & 0 & k_2 & -(k_4+k_5) & 0 \\ 0 & 0 & k_3 & 0 & -k_6 \end{pmatrix}. \tag{5.8}$$

With initial conditions $\boldsymbol{x} = \boldsymbol{x}(0)$ this has solution

$$\boldsymbol{x}(t) = \boldsymbol{x}(0)e^{\mathbf{A}t}$$

where $e^{\mathbf{A}t}$ is the state transition matrix. Therefore $\boldsymbol{x}$ is defined in terms of a double exponential, as the terms in the matrix $\mathbf{A}$ (the rate constants) are also in terms of exponentials.

In this case the structure of the equation set, from the structure of $\mathbf{A}$, is such that the analytic solution for each of the state variables can be easily (if laboriously) found.

5.6.5 Testing of Model 2

More realistic parameter values were sought for this particular model. From tests carried out by the University of Reading's Food Biosciences Department, the rate constants were found to vary from an order of 10^{-8}, through to 400. The values of the individual pre-exponential factors and activation energies provided by Leo Pyle [13] are listed in Table 5.2. They were provided with a considerable amount of uncertainty in their accuracy. For example, there was concern with E_2 actually being 0 (we changed this to 1000) as it is unrealistic to have no temperature dependence. No data on the parameters for the amino acid were available, so the associated parameters were assumed to be half the values of those for the sugar. However, as the aim is to provide a methodology to retrieve parameters, the accuracy of the 'true' values was not a major issue.

k_{0_j}	Value	ΔE_j	Value
k_{0_1}	7×10^7	E_1	9.9×10^4
k_{0_2}	2×10^{-7}	E_2	$\sim 1 \times 10^3$
k_{0_3}	2.5×10^{-6}	E_3	1.2×10^4
k_{0_4}	1×10^7	E_4	8×10^4
k_{0_5}	2.4×10^7	E_5	8×10^4
k_{0_6}	400	E_6	~ 100
k_{0_9}	3.5×10^7	E_9	9.9×10^4

Table 5.2: Observed k_{0_j} and ΔE_j values at a temperature of 120°C.

As a result of these experimental values, the rate constants k_j vary by fourteen orders of magnitude, which leads to a stiff problem. It is very difficult to capture both very fast and very slow reactions. The algorithm used previously in Section 3.3.1 therefore needed to include a stiff ODE solver.

In order to check the reliability of the method for this stiff (linear) problem, the analytic solutions are found as a comparison. Using the values of the 14 parameters provided by [13] and a constant temperature value of 400K, the seven rate constants k_j can be calculated using

$$k_j = k_{0_j} e^{-\Delta E_j / 8.3 \times 400}. \tag{5.9}$$

The k_j values are then entered into the state transition matrix $\mathbf{A}$ in (5.8). The eigenvalues and corresponding eigenvectors can be easily calculated. The eigenvalues are

$$\boldsymbol{\lambda} = \begin{pmatrix} -1.1 \times 10^{-3} & -388.1 & 2 \times 10^{-7} & -7.8 \times 10^{-6} & -3.9 \times 10^{-6} \end{pmatrix}. \tag{5.10}$$

As expected the eigenvalues are very different, which means we have a stiff problem (due to the larger order of magnitude differences between the values of the pre-exponential factors and activation energies). The problem associated with stiff systems is that extreme timescales require small time steps, and this limits the stability of the approximation. Therefore the ODE solver in the method uses a quasi-constant step size.

With $\boldsymbol{x}' = \mathbf{A}\boldsymbol{x}$ we look for a solution of the form

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \sum_j \alpha_j \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}_{\lambda_j} e^{(\lambda_j t)}, \tag{5.11}$$

a linear combination of solutions where the α_j's are constants determined by the initial conditions, λ_j's the eigenvalues of $\mathbf{A}$ and $(\boldsymbol{x})_{\lambda_j}$ the corresponding eigenvectors.

The analytic solution of the linear problem in Figure 5.5 is therefore approximated to three significant figures as

$$\begin{aligned} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} &= 2.34 \times 10^{-7} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} e^{-1.1\times10^{-3}t} + 4.24 \times 10^{-17} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} e^{-388t} \\ &+ 0.246 \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1.36 \times 10^{-4} \\ 1.73 \times 10^{-10} \end{pmatrix} e^{-2\times10^{-7}t} + 0.344 \begin{pmatrix} 0.698 \\ 0 \\ -0.716 \\ -9.84 \times 10^{-5} \\ -1.24 \times 10^{-10} \end{pmatrix} e^{-7.8\times10^{-6}t} \\ &+0.24 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} e^{-3.9\times10^{-6}t}. \end{aligned} \tag{5.12}$$

From this, the order of magnitude of action is only $\mathcal{O}(e^{-1\times10^{-6\text{ or}-7}})$. Comparing the magnitudes of these analytic solutions after, say, 30 minutes, it can be seen that species $A(x_1)$ and $B(x_2)$ have the highest concentrations and have changed little since the start time. Therefore, they are *slow* species. Species $D(x_4)$ has the next smallest concentration, whilst species $C(x_3)$ and $E(x_5)$ have concentrations order of magnitude $\mathcal{O}(10^{-10})$. Therefore species C and E are *fast* species.

5.6.6 Interpretation of Results of Model 2

Table 5.3 shows the ‘true’, guessed and retrieved parameter values for Model 2 in Figure 5.5 at the termination point of the method.

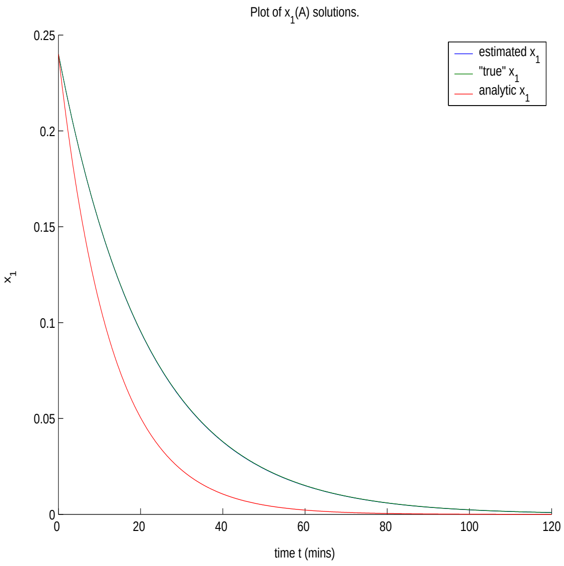
Parameter	‘True’ value	Guess	Retrieved value
k_{0_1}	7×10^7	6×10^7	8.4666×10^7
k_{0_2}	2×10^{-7}	1×10^{-7}	2.1789×10^{-7}
k_{0_3}	2.5×10^{-6}	2.3×10^{-6}	4.1588×10^{-6}
k_{0_4}	1×10^7	2×10^7	1.0192×10^6
k_{0_5}	2.4×10^7	2.2×10^7	9.4160×10^6
k_{0_6}	400	300	185.03
k_{0_9}	3.5×10^7	3×10^7	2.3803×10^7
E_1	9.9×10^4	9×10^4	9.9614×10^4
E_2	1×10^3	800	812.16
E_3	1.2×10^4	1.5×10^4	1.6287×10^4
E_4	8×10^4	8.5×10^4	6.8152×10^4
E_5	8×10^4	7.5×10^4	8.7404×10^4
E_6	100	200	192.01
E_9	9.9×10^4	9.6×10^4	9.7727×10^4

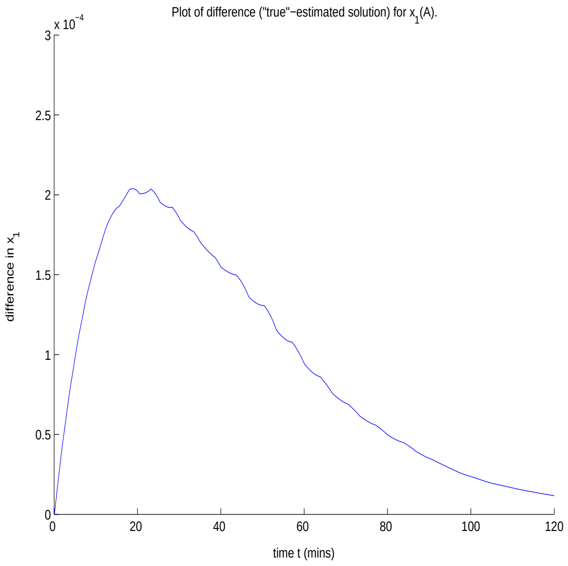
Table 5.3: Parameter input and output values for Model 2.

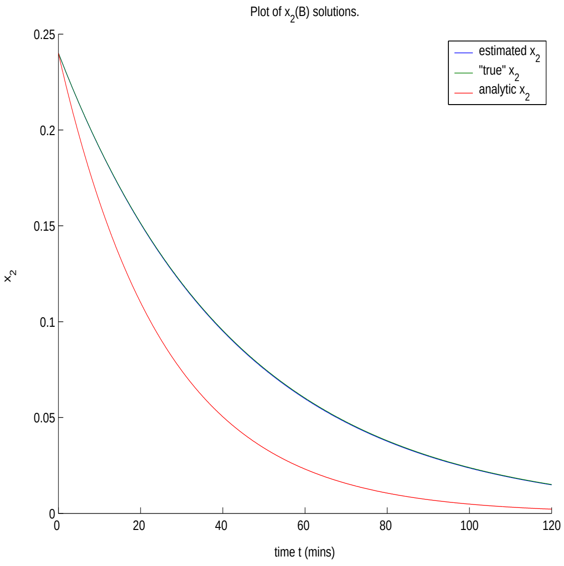
Comparing the retrieved parameters with those from the nonlinear model (which did not use observed parameter values), we note that the orders of magnitude are more consistent in this linear case. This would be expected, as the number of parameters has decreased. The retrieved values are closer to the ‘true’ values than in the 18-parameter case. The results are plotted in Figure 5.6.

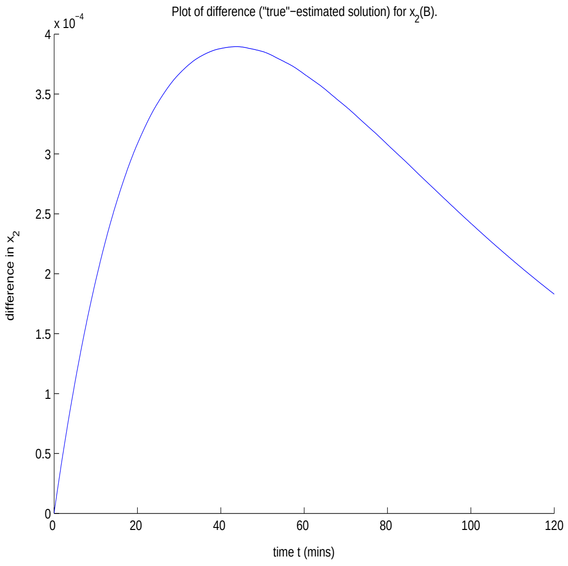
As can be seen from the plots, the blue and green lines representing the ‘true’ and iterated numerical solutions are on top of each other. They are much closer than in the nonlinear case. Additionally the analytic solutions are similar in form to those using the minimisation scheme for ramped temperature. The deviation is due to the fact that the analytic solution was calculated assuming a constant temperature whereas the numerical solution uses an exponential increase in temperature.

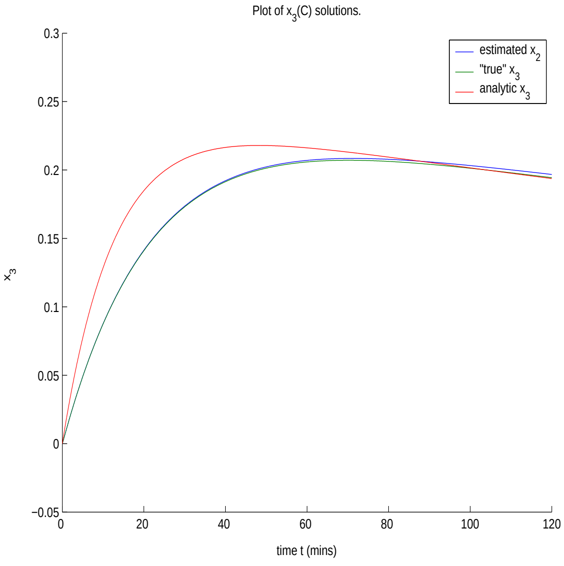
An important consequence of the original model annotated in [12] is that the model parameters can in theory be found sequentially from the concentration profiles for the

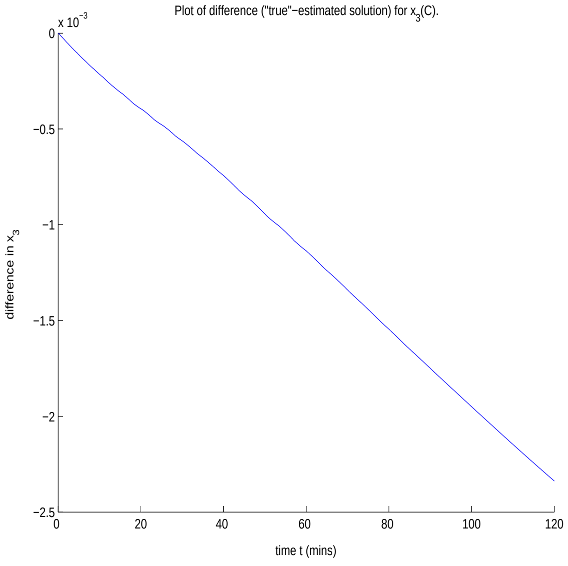

Plot of x_1(A) solutions.
estimated x_1
"true" x_1
analytic x_1
x_1
0.25
0.2
0.15
0.1
0.05
0
0
20
40
60
80
100
120
time t (mins)


Plot of difference ("true"−estimated solution) for x_1(A).
x 10^{-4}
difference in x_1
3
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
120
time t (mins)


Plot of x_2(B) solutions.
estimated x_2
"true" x_2
analytic x_2
x_2
0.25
0.2
0.15
0.1
0.05
0
0
20
40
60
80
100
120
time t (mins)


Plot of difference ("true"−estimated solution) for x_2(B).
x 10^{-4}
difference in x_2
4
3.5
3
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
120
time t (mins)


Plot of x_3(C) solutions.
estimated x_2
"true" x_3
analytic x_3
x_3
0.3
0.25
0.2
0.15
0.1
0.05
0
−0.05
0
20
40
60
80
100
120
time t (mins)


Plot of difference ("true"−estimated solution) for x_3(C).
x 10^{-3}
difference in x_3
0
−0.5
−1
−1.5
−2
−2.5
0
20
40
60
80
100
120
time t (mins)

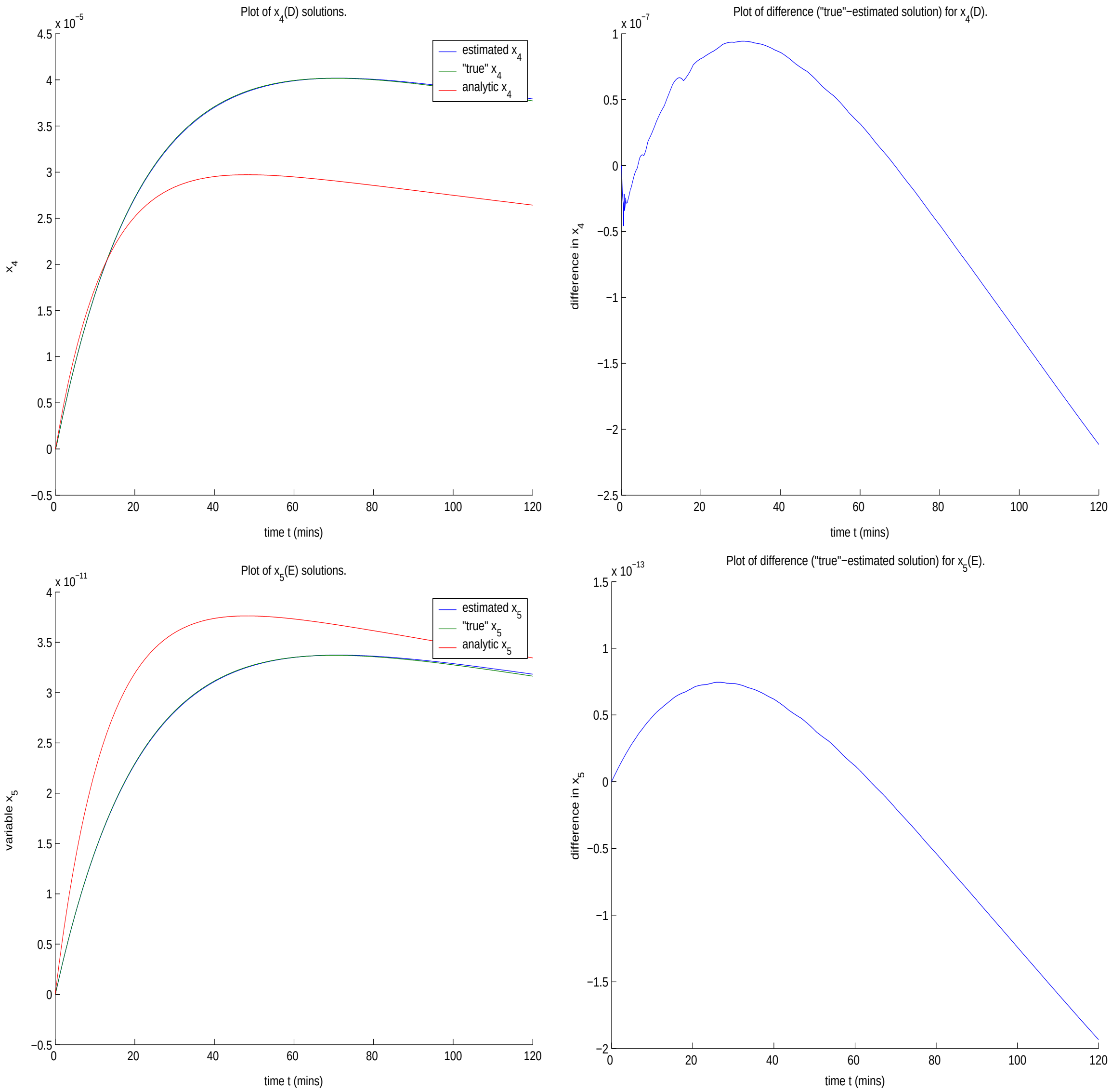


Figure 5.6: Plots of estimated, 'true' and analytic solutions for species $A(x_1)$, $B(x_2)$, $C(x_2)$, $D(x_4)$ and $E(x_5)$, and the differences between the 'true' and estimated solutions, versus time.

individual components; this also means that the individual models can be tested (more or less) independently.

5.7 Solutions Using Observed Concentration Data

Observed concentration values were obtained from [13] for the five species under consideration in Model 2 (Figure 5.5). Due to species C being very reactive and quickly transforming into intermediates D and E, no data for C was obtained. It is therefore taken out of the sum and assumed to be zero throughout the method. For our model the concentration measurements at a temperature of 120°C and pH 8 are used as this set of results is complete and we wish to test our model for the highest temperatures used in the laboratory. The values obtained from Leo Pyle are listed in Table 5.4.

Time	$A(x_1)$	$B(x_2)$	$D(x_4)$	$E(x_5)$
0 min.	0.24	0.24	0	0
30 min.	0.1678	0.1066	5.5153×10^{-4}	1.2182×10^{-3}
60 min.	0.0666	0.0640	4.2729×10^{-4}	4.9396×10^{-4}
90 min.	0.0266	0.0480	3.7274×10^{-4}	3.5456×10^{-4}
120 min.	0.0213	0.0427	2.8486×10^{-4}	2.2516×10^{-4}

Table 5.4: Observed concentrations of species A, B, C and D in mol/litre at a temperature of 120°C and pH 8.

Concentration recordings of the five species were taken at five times each, including the initial concentrations. To gain a complete profile of these observations at all times, and to estimate a curve through the five points for each species, *cubic spline interpolation* is used in the method. The values of the underlying function (in this case the set of observed concentrations) are found at the time points required, and a plot of the spline is then taken as the 'true' data in the algorithm in Section 3.3.1. Usually five points give a reasonably smooth spline with the possibility of an inflection.

A cubic spline is a spline constructed of piecewise third-order polynomials which pass through a set of m control points. The first and second derivatives are smooth, but the third are not. The second derivative of each polynomial is commonly set to zero at the endpoints, since this provides a boundary condition that completes the system of $m-2$ equations. This produces a so-called 'natural' cubic spline and leads to a simple tridiagonal system which can be solved easily to give the coefficients of the polynomials. However, this choice is not the only one possible, and other boundary conditions can be used instead.

With the real data, the model has to handle many orders of magnitude. The splines assume continuity. The parameters' 'true' values are no longer needed. Rather, selections are made in combination with the real observed data. Nevertheless, the initial guess of the parameter values is taken to be the same as those listed in Table 5.2 with two amendments. The values of ΔE_2 and ΔE_6 are set to one, in order to get some form of retrieved solutions for species D and E. The initial guess and retrieved parameter values are displayed in Table 5.5. The results are plotted in Figure 5.7.

Parameter	'Guessed' value	Retrieved value
k_{0_1}	7×10^7	1.2238×10^9
k_{0_2}	2×10^{-7}	4.6533×10^{-7}
k_{0_3}	2.5×10^{-6}	-1.0385×10^{-4}
k_{0_4}	1×10^7	5.2329×10^7
k_{0_5}	2.4×10^7	6.5298×10^8
k_{0_6}	400	-4.5705×10^3
k_{0_9}	3.5×10^7	7.7587×10^7
E_1	9.9×10^3	8.0518×10^4
E_2	1	-9.2124
E_3	1.2×10^4	-2.0735×10^4
E_4	8×10^4	1.7403×10^5
E_5	8×10^4	1.6858×10^6
E_6	1	13.371
E_9	9.9×10^4	7.0823×10^4

Table 5.5: Parameter input and output values for Model 2 using observed concentration data.

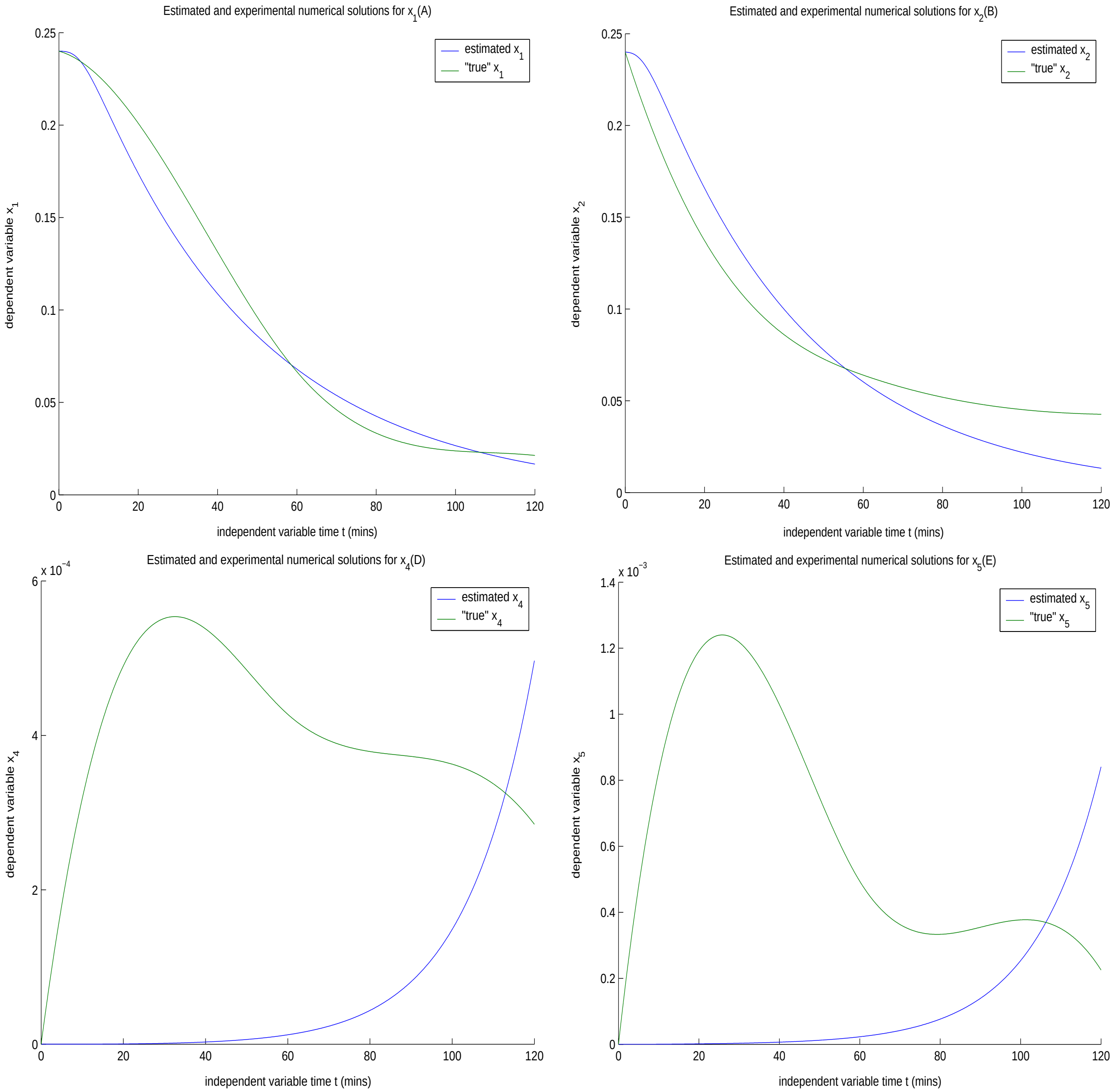
From combustion theory it is known that activation energies are relatively straightforward to measure. This is supported by these results, where the retrieved activation energies are closer to the initial guesses than those for the k_0 values. In practice k_0 values are impossible to measure. The retrieved values for $k_{0_2}, k_{0_4}, k_{0_9}, E_1$ and E_9 in particular are close to those that were observed in the Food Biosciences Department and formed our initial guess. They are of the same order of magnitude. With the parameters involved in the transition of species C to E and beyond being up to two orders of magnitude larger than the observed values, further observations are needed. It must be noted that these parameter values are valid for only one maximum temperature T_1 and may not be consistent for different temperatures.

Compared with the other tests, there is more variety in the orders of magnitude between the guessed and retrieved parameter values. Many of the retrieved values are larger by up to two orders of magnitude, suggesting our guesses (based on observations originally) are under-estimates. For the first time negative parameters are also obtained. This can be interpreted in a number of ways. Either our initial guesses are not very realistic, or the retrieved values have been affected by dataset uncertainty. There are two sources of dataset uncertainty: structured uncertainty (from the method - the main source), and the residual uncertainty (the method's finiteness of data). It is a sequential and highly sensitive method, so the errors cascade down. The method struggles to work with concentration data containing points of inflection as is the case for species D and E. This is because the inflections may not be part of the ODE's solution set.

5.8 Summary and Further Work

One of the main problems the Food Biosciences Department faces is the fact that all concentration data is limited. Parameter values are very sensitive to initial values, so a great deal of human intervention and forcing is required in order to obtain reasonable values. The algorithm used here has required no forcing and produces reasonably accurate solutions using limited data and splines. Little or no human intervention was necessary to obtain the plots for the different models.

If more data was obtained then fitting would be more accurate. However, another limitation is that the same test cannot be carried out twice. For example, modelling the effect of pH has not yet been successful. The Food Biosciences Department has recently started to continuously feed the reagents into the experiment, rather than carry out batch tests. They found that very stable steady states in colouring are eventually reached. Therefore



Estimated and experimental numerical solutions for x_1(A)

Estimated and experimental numerical solutions for x_2(B)

Estimated and experimental numerical solutions for x_4(D)

Estimated and experimental numerical solutions for x_5(E)

Figure 5.7: Species A, B, D and E's concentration profiles from cubic splines of observed data and iterated solutions using estimated parameters.

time derivatives could be replaced by algebraic equations in the future, which would be much easier to fit. Tests are ongoing in the department to investigate these balances.

Whether the schemes in the methods to solve the ODEs conserve mass and therefore satisfy the theory of conservation of mass implicit in the reaction models is a source of debate. There are many different minimisation techniques to choose from, but only one has been tested here.

Other issues for study include the number of observations required for a given mesh resolution in order to achieve satisfactory parameter identification. Ill-posedness of the problem of parameter estimation together with the problems of identifiability and uniqueness also need to be investigated. We do not claim that the solutions produced are unique. From the anomalies with the observed data it would suggest that we cannot defer the conditions on the k_{0_j} and ΔE_j values uniquely from the data. Nevertheless, even with noisy observations, the parameters have been recovered to an acceptable degree of accuracy for cases where only non-observational data is used.

The issues of ramping of the temperature and how often measurements of the concentrations of the different species are taken have not been addressed yet. The maximum concentrations x are required, but the k_j's can only be estimated from a limited number of points for three temperature experiments. An optimum experimental strategy to estimate the k_0 and ΔE values is still sought by the Food Biosciences Department.

The current approach is to keep the temperature at a fixed value such as 120°C. It has to be assumed that the pH level remains constant, as the real-life situation is hard to model. However, the actual tests can never be fully isothermal, because the equipment is always starting from cold.

In a similar approach to that taken by Niksa and Lau (cited in [10]), we have found analytic solutions with linear ramping and numerical solutions with exponential ramping temperature histories. The question to be posed is whether it would be better to carry out non-isothermal tests. This gives rise to a second minimisation problem - the first was to estimate the parameters, the second an optimal temperature function $T(t)$ to recover maximum information from the model.

Chapter 6

Parameter Estimation: Further Work and Possible Improvements

6.1 Ill-posedness

The key difficulty in developing successful numerical techniques for identifying spatially dependent parameters resides in the fact that such problems are ill-posed. Ill-posedness follows from the fact that the differentiation operator is not continuous with respect to any physically meaningful observations. For example the problem of identifying spatially-dependent coefficients appearing in the differential operator of a partial differential equation is in general both nonlinear and ill-posed. We do not know the covariance in this case, but we do know the order of magnitude if it is ill-posed in order to make it more well-posed.

6.2 Identifiability

The uniqueness problem in parameter estimation is related to the issue of parameter identifiability. Even if curves are successfully matched, it does not imply that the choice of parameters is unique. The model in Chapter 4 with only two parameters provided the best fit, but even in this case we cannot claim that the retrieved values are unique. Further work is needed to determine whether each unknown parameter in the batch test is 'identifiable', that is, whether it can be determined uniquely in all points of its domain by using the input-output relation of the system and the input-output data (Kitamura and Nakarigi, cited in [7]).

6.3 Scaling

For complicated functions, difficulties may be encountered in choosing suitable scaling factors. There is no general rule to determine the best scaling factors for all minimisation problems. According to [8], good scaling is problem-dependent. A basic rule is that the variables of the scaled problem should be of similar magnitude and of order unity. This is because, within optimisation routines, convergence tolerances are based on an implicit definition of 'small' and 'large'. Thus, variables with widely varying orders of magnitude may cause difficulties for some minimisation algorithms (Gill et al. cited in [16]). One simple direct way to determine the scaling factor is to use the typical values for different fields (for instance, 1 can be used as the scaling factor of concentration in Chapter 5).

6.4 Sensitivity Analysis

Sensitivity analysis is an efficient tool in parameter estimation in meteorology and oceanography. Whether it can be used in such a problem in the Food Industry as presented here is a source of further work. The relative sensitivity demonstrates the measure of the importance of the input parameter. The higher the relative sensitivity, the more important the input parameter in question. Thus, one of the crucial aspects of sensitivity analysis is to identify the most important input parameters whose changes impact the most chosen response.

The magnitudes of relative sensitivities can serve as a guide to ranking the importance of model parameters for use in choosing candidates for optimal parameter estimation. For models such as that in the food problem that involve a large number of parameters and comparatively few responses, sensitivity analysis can be performed very efficiently by using deterministic methods based on adjoint functions.

Chapter 7

Conclusion

We have tested a minimisation algorithm against a number of different models. For the Lorenz system in Chapter 3, the method found it difficult to match all of the chaotic detail. For the simple chemical reaction in Chapter 4, all of the action is at the start, and there is not as much detail to match, so the parameter iterations converge much more quickly. Over the shorter timescales the singular solutions are never observed.

In the food problem of Chapter 5, unlike the Lorenz case, the model reaches a steady state after a certain time. For the nonlinear food model, the profiles of the solutions produced by our minimising method closely resemble the form of the 'true' solutions found numerically. When the model is linearised by assuming the amino acid B does not transform into intermediate C, the iterated solutions using the optimal parameters are even closer to the 'true' solutions. Further, the form of the iterated solutions closely resembles the form of the analytic solutions found in the linear case. This validates the code itself, as well as the optimal parameters that are retrieved in both the linear and nonlinear case.

Combining the linear Model 2 with observed concentration data complicates the results. Some of the species are present in such small quantities that they are hard to capture. The solutions for the sugar and amino acid reagents are close to the observed profiles (with splines used to complete the profile of the 'true' data). However, the small-scale solutions of intermediates D and E do not resemble the observed concentrations. This would suggest that either the proposed linear model is an over-simplification of reality, or our minimising method cannot handle the large variety in orders of magnitude.

A better strategy is therefore required to determine *when* measurements should be taken.

Acknowledgements

I would like to thank Dr Mark Wakefield and Professor Mike Baines for their unending support with this paper and the work involved. I would also like to thank Professor Leo Pyle, whose research stimulated part of this investigation. Without our several meetings, this paper could not have been completed. I would like to thank Professor Nancy Nichols and Dr Amos Lawless for their input on the data assimilation issues discussed in the early part of the investigation. I acknowledge the financial support received from NERC.

Bibliography

[1] J. Cox, Storm Watchers - The Turbulent History of Weather Prediction from Franklin's Kite to El Niño, Wiley (2002)

[2] A. K. Griffith, N. K. Nichols, Data Assimilation Using Optimal Control Theory, The University of Reading (1994)

[3] J. C. Lagarias, J. A. Reeds, M.H. Wright, P. E. Wright, Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal of Optimization,* **9(1)** 112-147 (1998)

[4] A. S. Lawless, User guide for oscillating system and Lorenz models, Data Assimilation Research Centre

[5] M. Martin, A. S. Lawless, lorenz_menu.m (code), Data Assimilation Research Centre

[6] J. D. Murray, Mathematical Biology, 3rd ed, Springer (2002)

[7] I. M. Navon, Practical and Theoretical Aspects of Adjoint Parameter Estimation and Identifiability in Meteorology and Oceanography. *Dynamics of Atmospheres and Oceans,* **27** 55-79 (1997)

[8] I. M. Navon, X. Zou, J. Derber, J. Sela, Variational Data Assimilation with an Adiabatic Version of the NMC Spectral Model. *Mon. Wea. Rev.,* **120** 1433-1446 (1991)

[9] J. A. Nelder, R. Mead, A Simplex Method for Function Minimization. *Computer Journal,* **7** 308-313 (1965)

[10] C. P. Please, M. J. McGuinness, D. L. S. McElwain, Approximations to the DAEM for Coal Pyrolysis, The University of Southampton

[11] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, Numerical Recipes - The Art of Scientific Computing, Cambridge University Press (1986)

[12] L. Pyle, Modelling the Maillard Reaction Network, The University of Reading (2003)

[13] L. Pyle, Food Biosciences Department, The University of Reading, private communications (2004)

[14] C. Sparrow, The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors, Springer-Verlag (1982)

[15] M. Wlasak, A. S. Lawless, lorenz4d.m (code), Data Assimilation Research Centre

[16] X. Zou, I. M. Navon, Impact of Parameter Estimation on the Performance of the FSU Global Spectral Model Using Its Full-Physics Adjoint. *Monthly Weather Review,* **127** 1497-1517 (1999)