Estimation of Parameters in Tra c Flow Models Using Data Assimilation

by Amy Jackson

The University of Reading

The Department of Mathematics and Statistics

This dissertation is a joint MSc in the Departments of Mathematics and Meteorology and is submitted in partial ful lment of the requirements for the degree of Master of Science

September 2011

Acknowledgments

I would like to thank Dr. Amos Lawless and Dr. Peter Sweby for all the help and guidance they have given me.
I would also like to thank my family for their constant encouragement and support throughout this year.

Abstract

The Payne-Whitham model is a macroscopic tra c ow model, usually known as the two equation Payne's model, incorporating two independent parameters denoted by c_0 and .

It is implemented by producing an adjoint model, derived from the linearisation of the Harten, Lax and van Leer scheme (HLL).

The purpose of this dissertation is to nd the value of these parameters which give the optimal solution when using the model. This involves calculating the cost function, J, and minimising its gradient using data assimilation methods.

The results show that the system was insensitive to but demonstrated good resilience for c_0.

Declaration

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

Signed ....

Contents

1 Introduction

1.1 Background

Much work has been done, particularly over the past thirty years, on tra c ow problems generally based upon a stochastic approach. Just by sur ng the web, many studies based on varying theories may be found, among the most popular are those derived from Lighthill Whitham and especially H J Payne.

However, when the tra c density becomes high, the model can be considered as a continuum process (i.e. suitable for motorways or busy roads) but it must be remembered that such theory will not be accurate for low density tra c ow.

The purpose of this dissertation is to further the work begun by Danila Volpi in her dissertation `Estimation of parameters in tra c ow models using data assimilation' 2009 [1], University of Reading, which was based upon Roe's numerical scheme using a nite di erence method. The current dissertation uses an HLL (Harten, Lax van Leer) numerical scheme as an alternative to Roe for numerical modelling of the tra c ow and applies Data Assimilation techniques to estimate the parameters of the model.

1.2 Overview

Tra c problems are a serious global concern with pollution such as experienced in Mexico city, Los Angeles and tra c jams in most urban centres. These problems are set to increase given the forecast in growth of tra c in the coming decades.

Figures from the Department for Transport for the period 1994 - 2010 show a steady overall increasing number of licensed vehicles (excluding motor cycles

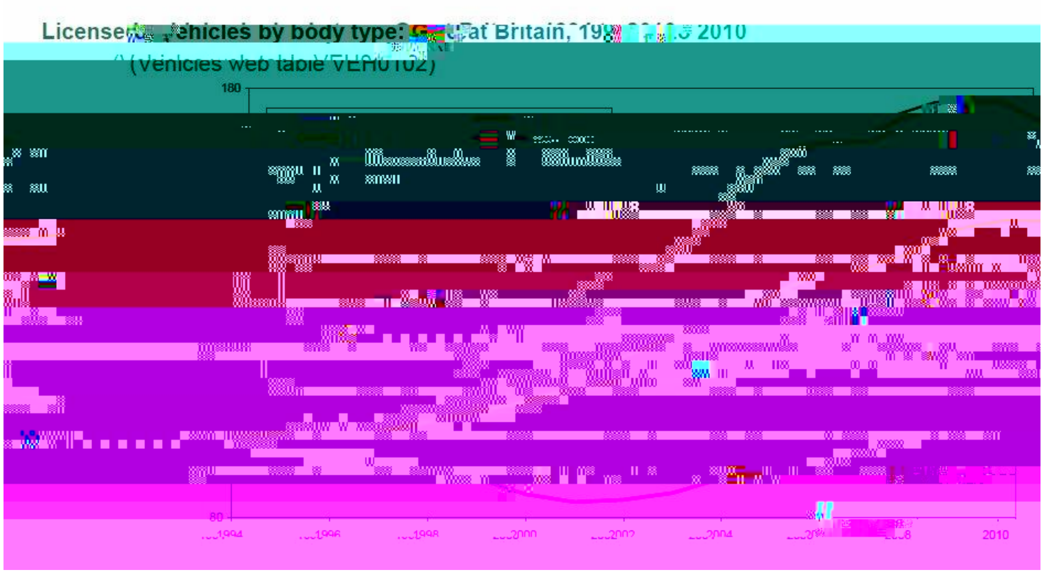


Figure 1: Graph from the Department for Transport [2]

and heavy goods). Although the current economic downturn is re ected in the last two years (see gure 1) the long term prediction is still more vehicles on the road in the UK and across the world generally. One recent estimate suggests an additional 5.7 million cars on the UK roads by 2031, a growth of 21% (Living Streets [3]).

This places increased pressure on the exi-279ue oaa79u 0ns(92(gro)27s0c5511)27rt [2]

2 Tra c Flow Models

2.1 Choice of the model

Apart from having stochastic qualities, three categories of simulation models for tra c have been developed:

- Microscopic models represent individual vehicle movements such as their velocity and position. Although precise, they are computationally very expensive when modelling a large number of vehicles.
- Mesoscopic models represent vehicle movements as groups sometimes

model can be represented essentially by the continuity equation:

$$_t + q_x = 0.$$

By de ning $q = \ v$ we can obtain the equation:

$$_t + (\ v)_x = 0; \tag{1}$$

where,

- $(x; t$

eigenvalues and is diagonalisable, or if it's eigenvalues are distinct real (J C Strikwerda [5]).

The eigenvalues of matrix $A(\mathrm{u})$ are:

$$ {}_1 = v + c_0; \qquad {}_2 = v - c_0 $$

and their corresponding eigenvectors are:

$$ e_1 = \begin{pmatrix} 1 \\ v - c_0 \end{pmatrix}; \qquad e_2 = $$

= 48 -30.97 Td [(and)-326(their)6593 Td [(0)]TJ
1cJ/F17 11.9552 Tf 8.054cJ/F117 [(cJ/F

and the scheme fails to capture the shock wave, then it may show vehicles continuing through the red light, which should not, and probably would not, happen in reality.

The numerical schemes considered for use in this project are discussed in sections §3 and §4.

3 Numerical Schemes- Roe's Algorithm

3.1 Introduction

Roe's algorithm (P L Roe [6]) is one of the most basic Riemann approximations. It is a well established scheme for aerodynamics which can be used to solve a simple form of a macroscopic traffic model, the Lighthill-Whitham 1-equation model (LW model). Beginning with this, the LW model is essentially the continuity equation (1),

$$\rho_t + (\rho v)_x = 0;$$

where $\rho(x; t)$ is the traffic density measured in vehicles per km, and ρv is the traffic flux, given in vehicles per hour.

3.2 The Theory

Equation (4) is a continuous function which is very difficult to solve and cannot be done analytically. In order to solve it numerically, Roe linearises the homogenious form in each interval $(x_{i-1}; x_i)$, replacing $A(\mathrm{u})$ by matrices $\tilde{A}(\mathrm{u}_{i-1}; \mathrm{u}_i)$ which, for any adjacent states u_L,u_R, the following three conditions are satisfied:

1. Hyperbolicity: $\tilde{A}(\mathrm{u}_L; \mathrm{u}_R)$ is diagonalisable with real eigenvalues,
2. Consistency: $\tilde{A}(\mathrm{u}_L; \mathrm{u}_R) \to A(\mathrm{u})$ as u_L and $\mathrm{u}_R \to \mathrm{u}$
3. Conservation: $\mathrm{f}(\mathrm{u}_L) - \mathrm{f}(\mathrm{u}_R) = \tilde{A}(\mathrm{u}_L; \mathrm{u}_R)(\mathrm{u}_L - \mathrm{u}_R)$
   where f(u

used in this work. However it did provide the starting point used for the HLL

A drawback to Roe's method is that is does not check if the solution is entropy satisfying, and so could be incorrect and contain unphysical discontinuities. To deal with this an entropy x must be added, which will ensure that the solution obtained is correct.

4 Numerical Schemes- Harten, Lax and van Leer

4.1 Introduction

Part of the data assimilation process is to obtain what is known as the tangent linear model which involves linearising the Roe Riemann solver. However Roe has absolute value signs in the numerical ux formula which are di cult to where

$$\mathbf{f}^{HLL}_{i\ \frac{1}{2}} = \frac{{}^{R}_{i\ \frac{1}{2}}\mathbf{f}_{i\ \ 1} - {}^{L}_{i\ \frac{1}{2}}\mathbf{f}_{i} + {}^{L}_{i\ \frac{1}{2}}\ {}^{R}_{i\ \frac{1}{2}}(\mathbf{u}_{i} - \mathbf{u}_{i\ \ 1})}{{}^{R}_{i\ \frac{1}{2}} - {}^{L}_{i\ \frac{1}{2}}}$$

and were used fo00

Where,

- u is a state vector
- *i* represents spacial points from 0 to *N* along the direction of ow in the computational grid
- is the tra c density
- *v* is the mean speed
- are eigenvalues of the matrix $A(\mathbf{u})$ de ned in §2.3
- f is the tra c ux
- *L* and *R* signify the left and right states respectively
- s is the source term
- U_{cap} is the equilibrium speed-density relationship, as in §2.3.

4.4 Results

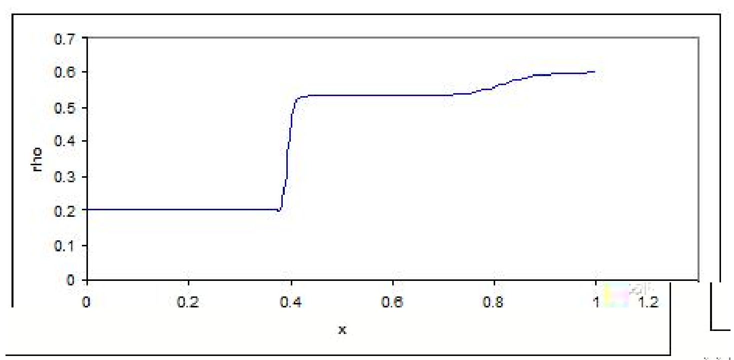


Figure 4: Density-space relationship at the nal time step.

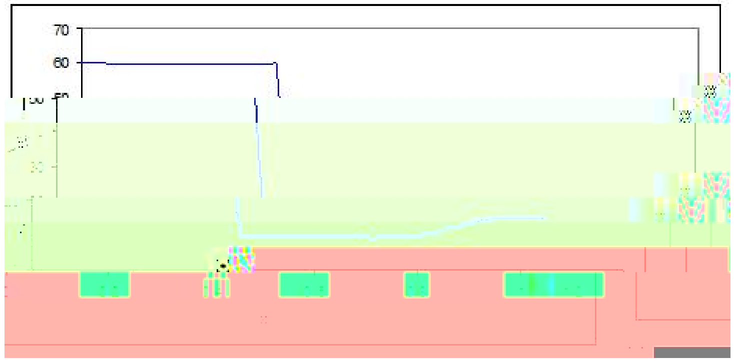


Figure 5: Velocity-space relationship at the nal time step.

The graphs produced appear almost identical to those for Roe. Since the HLL method is much simpler and entropy satisfying, no entropy x is required and so will be suitable for the remaining stages of the project.

5 Data Assimilation

`A data assimilation system consists of three components: a set of observations, a dynamical model, and a data assimilation scheme where the goal is to minimise a cost function with the constraints of the model equations and their parameters.'(A Robinson and P Lermusiaux [8])

It is an iterative process, in two senses, one using time steps where observations are joined with corresponding forecasts from the scheme, (in our case the HLL scheme) for input to the cost function. The other is in the minimisation process where the cost function is minimised to produce the best values for the system parameters forming the parameter vector, **p**.

The approach is based on an augmented state vector, **z**, comprising of the constants c_0 and plus the state variables and v. That is:

$$\mathbf{z} = (c_0 \quad ! \quad v)^T$$
$$= \begin{matrix} \mathbf{p} \\ \mathbf{u} \end{matrix}$$

The data assimilation begins with an initial state x_0, and incorporates observations (current and past) into a numerical model in order to produce a model state, known as the analysis, which most accurately represents the current state of the system.

The model uses the observations in time iterations, i.e. observations across the whole time window can be used (where the time window is given between $(t_0; : : : ; t_n)$). The analysis occurs at t_0 and best represents the actual true state and can be used to make future predictions, such as tra c forecasts for a road closure.

5.1 Four Dimensional Variational Assimilation ($4D-$Var)

$4D-$Var is a model based on the minimisation of an associated cost function which measures the di erence between the observations and the forecasts made, weighted by the accuracy of the measurements taken. This project will use $4D-$Var to estimate the state parameters and c_0.

5.1.1 Useful Notation

- $3D-$Var, Three-dimensional variational analysis
- $4D-$Var, Four-dimensional variational analysis
- Truth, x_T, The actual true state, e.g. the true temperature of a room
- Analysis, x_a, The analysis is our best estimate of this truth given the information available
- Background, x_b, Prior estimate of the truth before the observations are

5.1.2 Error Handling

Errors undoubtedly arise in observations which can be caused by numerous reasons, for example inaccuracies in taking the measurements or as a result of using faulty apparatus. If the errors are not dealt with then this will result in an unreliable analysis and so all errors must be minimised for the nal analysis.

Observation errors are often correlated especially if they are the result of measurements taken using the same apparatus. These error correlations can be represented in error covariance matrices for calculation purposes. The background error covariance matrix B (de ned only for t_0), and the observation error covariance matrix R are known but can be hard to produce when they are formed from multiple sources.

With a large number of observations these matrices are very expensive to invert as B is of size $n \cdot n$ and R is of size $p \cdot p$ where n is the size/length of the state vector and p is the number of observations. (E Kalnay [10])

5.2 Generic Cost Function

As previously mentioned, $4D-$Var is de ned as the minimisation of the following cost function which measures the di erence between the observations and the forecasts, weighted by the accuracy of the measurements, with the general formula given by:

$$J(\mathrm{x}) = (\mathrm{x} - \mathrm{x}_b)^T B^{-1}(\mathrm{x} - \mathrm{x}_b) + (\mathrm{y} - \mathrm{h}(\mathrm{x}))^T R^{-1}(\mathrm{y} - \mathrm{h}(\mathrm{x})). \tag{5}$$

So our analysis can be written as:

$$\mathrm{x}_a = min_{\mathbf{x}} J(\mathrm{x}).$$

5.3 Cost Function as applied

For simplicity the background term of the cost function has been removed in the application of this dissertation. The removal is permitted if enough observations exist, although it is also sensible to do where the background values are not known sufficiently accurate. Time permitting, we could experiment with including background values, however, as is said it is simpler to start without it.

Hence, the cost function (5) becomes:

$$
\begin{aligned}
J(\mathrm{p}) &= \sum_{i=0} (\mathrm{y}_i - \mathrm{h}_i[\mathrm{x}_i])^T R_i^{-1} (\mathrm{y}_i - \mathrm{h}_i[\mathrm{x}_i]) \\
&= \sum_{i=0} (\mathrm{y}_i - \mathrm{h}_i[\mathrm{x}_i])^2 R_i^{-1} \\
&= \sum_{i=0} (\ {}_i^o - \ {}_i)^2 R^{-1} + (v_i^o - v_i)^2 R_v^{-1}
\end{aligned}
$$

where

- y_i is the observed value of x_i
- h_i is the observation operator
- $\mathrm{x}_i \equiv \begin{pmatrix} \ {}_i \\ v_i \end{pmatrix}$
- ${}_i$ is the value of at position i
- v_i is the value of v at position i
- ${}_i^o$ is the observed value of at position i
- v_i^o is the observed value of v at position i

- R^{1} is the observation error covariance matrix for
- R_v^{1} is the observation error covariance matrix for v

This adaption of the cost function is calculated in the HLL scheme.

5.4 Minimisation of the Cost Function

The minimisation of the cost function is the method used to solve the paramwhere,

$$
\begin{aligned}
\frac{\partial x_i}{\partial p_0} &= \frac{\partial (m_{i-1}(x_{i-1}))}{\partial p_0} \\
&= \frac{\partial (m_{i-1}(x_{i-1}))}{\partial p_{i-1}} \frac{\partial (m_{i-2}(x_{i-2}))}{\partial p_0} \\
&= \frac{\partial (m_{i-1})}{\partial p_{i-1}} \frac{\partial (m_{i-2})}{\partial p_{i-2}} \frac{\partial (m_{i-3}(x_{i-3}))}{\partial p_0} \\
&= \frac{\partial (m_{i-1})}{\partial p_{i-1}} \frac{\partial (m_{i-2})}{\partial p_{i-2}} \cdots \frac{\partial (m_0)}{\partial p_0} \\
&= M_{i-1} M_{i-2} \cdots M_{i-p}
\end{aligned}
$$

For further details on the adjoint model, see section §7.

Perturbations c_0 and of c_0 and are allowed where c_0 and are % ($1 \leq | | \leq 10$) of the initial values assigned to c_0 and respectively. The new values ($c_0 + c_0$) and (+) are used by the HLL scheme to calculate a new value for the cost function $J(\mathrm{p})$, $\mathrm{p} = \begin{matrix} c_0 + c_0 \\ + \end{matrix}$.

The same values of the perturbations are used in the adjoint model to calculate the gradient of the cost function evaluated at and c_0 using the new values of the perturbations c_0 and .

For example,

$$ {}_n = \left. \frac{\partial J}{\partial} \right|_{t=n} $$

represents the gradient of J with respect to at time n.

It is this gradient and the corresponding cost function which are used in the minimisation to obtain the optimal parameters values.

5.5 Data Assimilation for Parameter Estimation

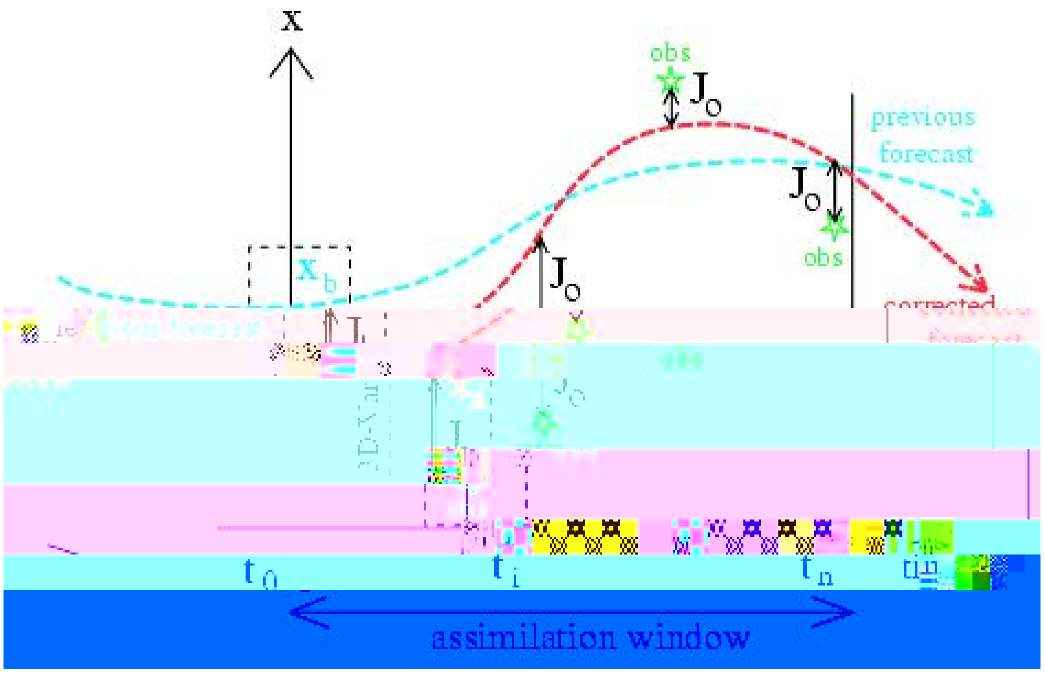


Figure 6: Example of $4D-$Var assimilation in a numerical forecasting system, graph from [13].

In the graphical representation of the $4D-$Var approach, Figure 6, the blue line is the previous forecast and the red is the corrected forecast after the model has been run. It shows that the model is minimising the distance between the forecast trajectory and the observations. This distance is measured by the cost function, $J(\mathbf{p})$, which calculates the weighted sum of the squares of these distances. $4D-$Var is used re-iteratively to minimise the cost function with respect to $\mathbf{p}$, the parameter vector.

By nding the minimum value of this sum we are able to obtain the required values for and c_0.

6 Tangent Linear Model

6.1 Introduction

The Tangent Linear Model is essentially the Jacobian of the nonlinear model operator and is therefore derived directly from the HLL model (A Lawless [11]) by keeping the statements of variables themselves unchanged and then adding the related derivatives of these statements. It is required as an intermediate stage to obtain the adjoint model, however note that any logical comparisons remain as in the HLL model. This stage involved the linearisation of the HLL model, for further background theoretical details see N Nichols [14].

6.2 Theory of the TLM

6.2.1

A typical linearisation model is of the form

$$\begin{aligned} \mathrm{x}_i &= \mathrm{m}_i(\mathrm{x}_{i-1}) \\ &= \mathrm{m}_{i-1}\mathrm{m}_{i-2}\ldots\mathrm{m}_0(\mathrm{x}_0) \\ &= \mathrm{m}(\mathrm{x}_0, t_i, t_0) \end{aligned}$$

where m is the non-linear model and x_0 is the state at the initial time t

ture positions and what appear to be random uctuations in the constants, i.e. the relaxation and anticipation constants when observing real tra c events. This is achieved by including a randomly generated value for p in the model. Hence,

$$x_i + \quad x_i = m(x_0; p + \quad p; t_i; t_0): \tag{9}$$

Applying a Taylor series expansion to (9) gives:

$x_i + \quad x_i = m(x_0; p; t_i; t_0) + i) +$

where

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

where x_1 to x_n are the variables required to obtain Z.

Then the tangent linear code for this is:

$$Z = \partial f$$

v. Both plots should tend to zero.

The non-linear model (HLL) is run twice, once with unperturbed values for and c_0 giving $m(\mathrm{p})$, and once with perturbed values for and c_0, giving $m(\mathrm{p} + \ \mathrm{p})$. The perturbation vector is given by $\mathrm{p} = \ c_0$. Following this, the tangent linear model is run once with the perturbed and c_0 to give $M \ \mathrm{p}$.

The total perturbation, $m(\mathrm{p} + \ \mathrm{p}) - m(\mathrm{p})$, is then compared with it's linear component, $M \ \mathrm{p}$. (Y Li et al [17]) This is performed by calculating the relative error as shown below.

The relative error is given by:

$$\frac{\|m(\mathrm{p} + \ \mathrm{p}) - m(\mathrm{p}) - M \ \mathrm{p}\|}{\|M \ \mathrm{p}\|} \cdot 100$$

where, $\|\mathrm{x}\| = \sqrt{\sum_{i=1}^{P} x_i^2}$ is the L_2 norm.

Once this was calculated, the logarithmic relative error was plotted against decreasing perturbation sizes, , where $= 10^0; 10^{\ 1}; 10^{\ 2}; 10^{\ 3}; 10^{\ 4}; 10^{\ 5}$ for the velocity plot, and from 10^0 to $10^{\ 6}$ for the tra c density plot. The graphs obtained are shown below:

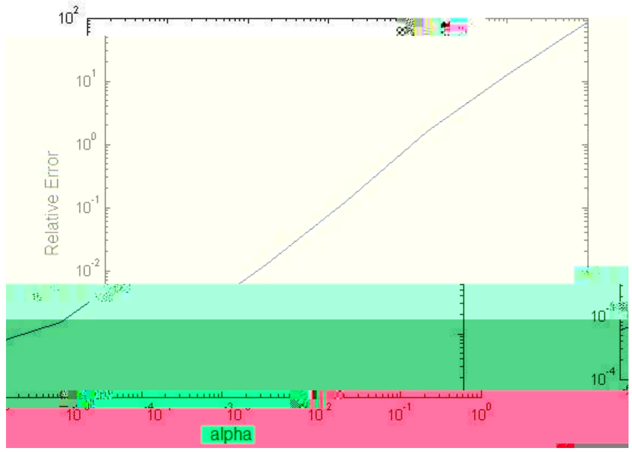


Figure 7: A graph to show the correctness of the TLM after 200 timesteps for tra c density.

7 Adjoint Model

7.1 Introduction

The adjoint model produces the gradient of the cost function which is an intrinsic part of the minimisation process.

The adjoint variables represent the gradient of the cost function with respect to the model variables. The TLM starts from the initial HLL values and calculates the variation between the unperturbed and the perturbed values for the tangent linear model at $t = t_{max}$; the adjoint model will start with the nal values produced by TLM and run backwards in both space and time to estimate the initial state vector values i.e. at $t = 0$.

7.2 Theory of the Adjoint Model

In order to obtain the adjoint of a linear model, it must be presented in the form $x^{n+1} = M x^n$. The adjoint is then $\hat{x}^n = M^T \hat{x}^{n+1}$. Adjoint models are very useful for computing the derivatives of a function which has numerous input variables, and so is particulary good for parameter and/or state vector estimation.

The Adjoint model was generated from the TLM model by `reversing' the logic/code and exchanging the derivatives on either side of statements; where they did not exist, the statements remain unchanged.

7.3 Examples

7.3.1 General Example

For example, take the following linear model of two variables $y; z$ with

$$y^{n+1} = y^n + z^n$$

$$z^{n+1} = y^n + z^n:$$

This can be written as $\begin{pmatrix} y \\ z \end{pmatrix}^{n+1} = \begin{pmatrix} \; \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix}^{n};$

i.e. $\mathrm{x}^{n+1} = M\mathrm{x}^{n}$ where $\mathrm{x} = \begin{pmatrix} y \\ z \end{pmatrix}$ and $M = \begin{pmatrix} \; \end{pmatrix}$ f 9.49.350

$$\begin{pmatrix} \hat{u}(j;i) \\ \hat{h}(j;i+1) \\ \hat{h}(j;i) \\ \hat{s} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{t}{x} & 1 & 0 & 0 \\ \frac{t}{x} & 0 & 1 & 0 \\ t & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{u}(j;i) \\ \hat{h}(j;i+1) \\ \hat{h}(j;i) \\ \hat{s} \end{pmatrix} :$$

Therefore the corresponding code for the adjoint model of u (in TLM) is:

$$\begin{aligned} \hat{u}(j;i) &= \hat{u}(j;i) \\ \hat{h}(j;i+1) &= \hat{h}(j;i+1) - \frac{t}{x} \cdot (\hat{u}(j;i)) \\ \hat{h}(j;i) &= \hat{h}(j;i) - \frac{t}{x} \cdot (-1) \cdot (\hat{u}(j;i)) \\ \hat{s}(j;i) &= \hat{s}(j;i \end{aligned}$$

where the triangular brackets denote the inner product.

The procedure for the validity test is as follows, where M is an operator and M^T is its adjoint.

1. We begin with a random perturbation x.
2. Then, the TLM code is applied, giving $M\ x$.
3. The adjoint model is then applied to $M\ x$ to obtain $M^T M\ x$.
4. Calculate $\langle M\ x; M\ x \rangle$.
5. Calculate $x; M^T M\ x$.
6. Check that $\langle M\ x; M\ x \rangle =\ x; M^T M\ x$.

When this test was done, $\langle M\ x; M\ x \rangle$ and $x; M^T M\ x$ produced the same value, from which it was concluded that the adjoint had been coded correctly. (Lawless et al [19])

8 Implementing the Minimisation

When the adjoint model had been obtained and verified, the gradient of the cost function J was calculated and tested to ensure correctness.
The gradient, ∇J, of the cost function, as determined by the HLL scheme, is calculated within the adjoint model. This value is then used in the gradient test, described in §8.1, to ensure that the adjoint model is working correctly. The gradient, ∇J, is given by equation (8).

The outputs from the adjoint model are and c_0 where

$$ ^n = \left.\frac{\partial J}{\partial }\right|_{t=n}; \qquad c_0^n = \left.\frac{\partial J}{\partial c_0}\right|_{t=n}. $$

As explained in §5.4, ^{n} is the gradient of J with respect to at time n and c_0^n the gradient of J with respect to c_0 at time n.
To be confident that these values are the correct gradient calculations of J, the gradient test was applied.

8.1 Gradient Test

The HLL scheme calculates the cost function $J(\mathrm{x}; \mathrm{p})$ for given values of c_0 and . It then uses perturbed values of these parameters to produce a perturbed value of J, given by:

$$ J(\mathrm{x}; \mathrm{p} + \ \mathrm{p}). $$

Using the Taylor expansion:

$$ J(\mathrm{x}; \mathrm{p} + \ \mathrm{p}) = J(\mathrm{x}; \mathrm{p}) + \ \mathrm{p}^T \nabla J(\mathrm{x}; \mathrm{p}) + O(\ ^2). $$

Rearranging:

$$ \frac{J(\mathrm{x}; \mathrm{p} + \ \mathrm{p}) - J(\mathrm{x}; \mathrm{p})}{\mathrm{p}^T \nabla J(\mathrm{x}; \mathrm{p})} = 1 + O(\ ). $$

Now de ne

$$(\quad) = \frac{J(x; p + \quad p) - J(x; p)}{p^T \nabla J(x; p)}$$

where takes the values $1; 0.1; 0.01; \ldots; 10^{\;\;11}$ and $p = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$.

For values of that are small, but quite not zero, ( ) should show a constant value close to 1 (Navon et al [20]). Figure 9 clearly shows ( ) = 1 over an interval of 6 orders of magnitude. 's lower limit is restricted by the accumulation of rounding errors arising in the computer used.

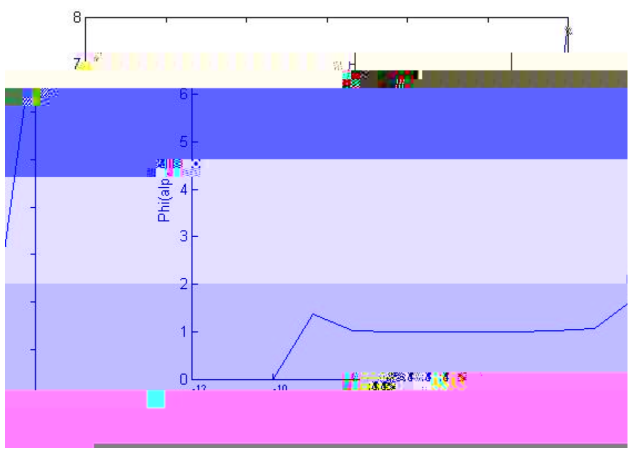


Figure 9: A graph showing ( ) against .

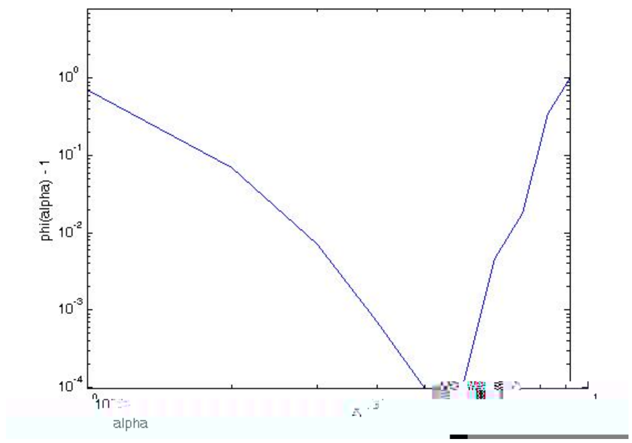


Figure 10: A graph showing (( ) − 1) against .

9 Results

The purpose of the project is to determine the optimal values for and c_0 which produce the closest match between the forecast and observed measurements.

To do so, the HLL scheme and the adjoint model are combined with the computer module CONMIN to minimise the cost function, J, and the modulus of it's gradient, ∇J, for all specified time steps. To verify that $4D-$Var is suitable to obtain the parameters, it is necessary to investigate the effects of altering the initial `guess' values of and c_0, together with changing the size of the time window, t, and adjusting the amplitude of the noise on the observations, represented by the observation error covariences (and $_v$).

For testing purposes, a random number generator was used to generate fluctuations of the u values to produce the observation values used by the cost function.

Although ∇J is one of the factors of the minimisation process, it should be noted that it is not possible to solve exactly for $\nabla J = 0$, which is the reason a user specified convergence attribute, EPS, is embedded within CONMIN.

Note: in CONMIN, EPS imposes convergence when

$$\|\nabla J\| \leq \cdot \max\{1; \|\mathrm{p}\|\};$$

where is the value of the EPS attribute.

9.1 Verification

The figures below are graphical representations of J and $\|\nabla J\|$ against the CONMIN iteration number. Figure 11 clearly shows J decreasing and then converging to a value after three iterations. Whereas Figure 12 illustrates $\|\nabla J\|$ decreasing steadily and approaching zero after seven iterations. These two graphs demonstrate the minimisation process and affirm that CONMIN has been implemented correctly.

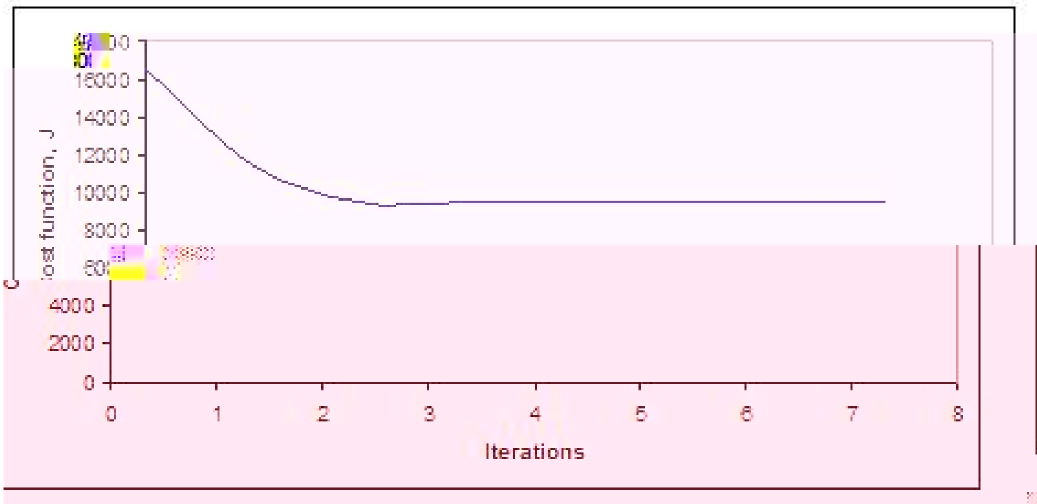
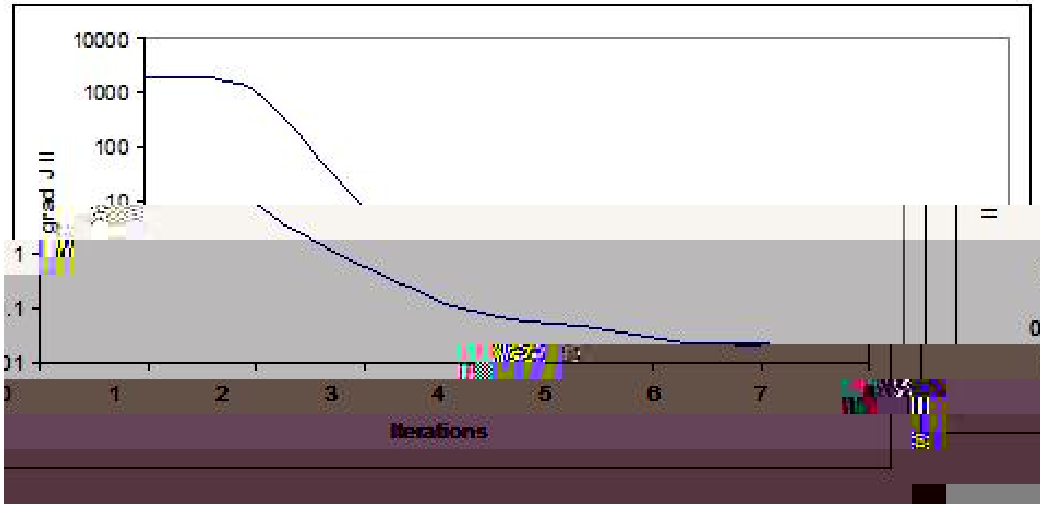


Figure 11: A graph to show J against the iterations.

The data for the graphs was produced using the values:

$$
\begin{aligned}
c_0^{true} &= 50 \qquad c_0^{guess} = 55 \\
{}^{true} &= 5 \qquad {}^{guess} = 4.5 \\
&= 1 \cdot 10^{5}
\end{aligned}
$$

and the time window, $t = 3.75 \cdot 10^{3}$.


10000
1000
100
10
1
0.1
0.01
grad J
0
1
2
3
4
5
6
7
Iterations

low iteration numbers.

9.2.1 Varying c_0

In the results below the superscripts ^{g}, ^{e} and ^{t} stand for guess, estimate and true values respectively, and is the CONMIN iteration number.
It is expected that when c_0^g and ^{g} are given values with a greater di erence from their true values, the minimisation will take longer to process, i.e. more iterations will be required to produce a close estimate. If they deviate too far, then the model might not produce close values.

c_0^g	10	20	40	50	55	60	80	100
^{g}	4.5							
t	$5 \cdot 10^{3}$							
	10^{6}							
c_0^e	50	50	50	50	50	50	50	50
^{e}	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000
J	10447.3	10447.3	10447.3	10447.3	10447.3	10447.3	10447.3	10447.3
∇J	1113.6	24162.6	7919.5	932.8	5551.7	3906.1	10738.4	4876.2
	20	11	13	13	15	10	14	9

c_0^g	10	20	40	50	55	60	80	100
^{g}	4.5							
t	$5 \cdot 10^{\ \ 3}$							
	0.7							
c_0^e	49.443	49.443	49.443	49.443	49.443	49.443	49.443	49.443
^{e}	124469.6	190869.8	161542.0	167884.9	188998.3	216759.4	169437.1	136860.9
J	12546.3	12546.3	12546.3	12546.3	12546.3	12546.3	12546.3	12546.3
∇J	0.8	0.6	0.7	0.7	0.6	0.4	0.7	0.8
	32	29	27	26	28	33	29	29

Table 2: c_0 varying with = 0:7

This is a general point concerning the results, that the solution may be reached when the speci ed level of convergence is obtained, irrespective of whether minimisation has been completed.

When the same values are used with less perfect observations, the model still manages to reach a close value for c_0, of 49.443. However this change has caused ^{e} to have extremely large values (see Table 2) which do not appear to correlate with the values of c_0. However, J converges to 12546.3 and ∇J tends to zero suggesting that minimisation has occurred.

9.2.2 Varying

Varying ^{g} had no e ect on the value of c_0^e in the tests conducted as can be seen in Tables 3 and 4. In each of these cases, a good value of ∇J that approached zero was recorded, although the values of ^{e} were far from the true value especially so for the higher value of .
From this, it can be concluded that minimisation was achieved giving a good value of c_0 despite varying and hence the process appeared to be highly insensitive to the parameter (for reasonable values).

c_0^g	55							
^{g}	3	4	5	6	7	10	15	30
t	$5 \cdot 10^{3}$							
	0.01							
c_0^e	50.01	50.01	50.01	50.01	50.01	50.01	50.01	50.01
^{e}	48.92	48.29	48.29	48.28	48.29	48.29	48.29	48.29
J	10447.8	10447.8	10447.8	10447.8	10447.8	10447.8	10447.8	10447.8
∇J	0.051	0.017	0.045	0.158	0.028	0.005	0.084	0.037
	29	24	21	22	22	25	22	16

Table 3: varying with = 0:01

c_0^g	55							
^{g}	3	4	5	6	7	10	15	30
t	$5 \cdot 10^{3}$							
	0.7							
c_0^e	49.443	49.443	49.443	49.443	49.443	49.443	49.443	49.443
^{e}	[illegible]	131612	187058	99929	162814			

Despite expecting the time interval to have an e ect on the results, Table 5 shows that the model attained very good estimates for the parameters, regardless of t. Table 6, with a larger value of , produce similar values for c_0 and the cost function, however uctuated widely and the gradient to a lesser extent.

It should be noted that as t increases, since t remains constant throughout, more observations are used in the cost function and hence in the minimisation calculations.

c_0^g	55							
^{g}	4.5							
t	0.00025	0.00125	0.0025	0.00375	0.005	0.00625	0.0075	0.0125
	$10\;^{6}$							
c_0^e	50	50	50	50	50	50	50	50
^{e}	5.007	5.001	5.001	5.001	5.001	5.000	5.000	5.000
J	$4{:}8 \cdot 10^{2}$	$2{:}5 \cdot 10^{3}$	$5{:}1 \cdot 10^{3}$	$7{:}8 \cdot 10^{3}$	$1{:}0 \cdot 10^{4}$	$1{:}3 \cdot 10^{4}$	$1{:}6 \cdot 10^{4}$	$2{:}7 \cdot 10^{4}$
∇J	2.6	$4{:}3 \cdot 10^{6}$	$8{:}9 \cdot 10$	$1{:}7 \cdot 10^{6}$	$5{:}6 \cdot 10^{3}$	$1{:}2 \cdot 10^{7}$	$1{:}4 \cdot 10^{7}$	$4{:}4 \cdot 10^{6}$
	11	31	11	28	27	18	30	20

Table 5: t varying with all other parameters remaining constant, using $=$ $10\;^{6}$.

c_0^g	55							
^{g}	4.5							
t	$5 \cdot 10^{\ 3}$							
	$10^{\ 6}$	$10^{\ 4}$	$10^{\ 3}$	$10^{\ 2}$	$10^{\ 1}$	10^0	2	5
c_0^e	50.000	50.000	50.001	50.009	49.938	49.185	48.237	44.023
^{e}	5.000	5.045	5.493	48.290	1211698	123999.7	23747	163.2
J	10447.3	10447.3	10447.3	10447.8	12158.0	12563.6	12563.6	12425.6
∇J	5551.7	64.3	2.5	0.0	4.6	0.5	4.8	0.4
	15	13	14	28	35	28	28	14

Table 7: varying with all other parameters remaining constant.

9.2.5 Further Investigations

Further investigations were conducted, some of which are included in Tables 8 and 9 below. The tests involved varying two or more parameters at the same time and using more extreme values for the parameters. This tests the model's resilience when using exaggerated values.

The rst observation from Table 8 is that c_0 was estimated accurately irrespective of the values of c_0, or , within the given ranges. also approximated well for small even though it's initial value deviated signi cantly from the true value. However for larger values of , as ^{g} approached it's true value, the estimated value became highly inaccurate.
This agrees with the previous comments about the insensitivity of which receives further support from the results in Table 9, where extreme values of ^{g}, providing that ≤ 1, did not prevent the attainment of very good values for c_0^e. For higher values of (> 1), as increased, the values for c_0^g deteriorated despite good values for ∇J and an increasing time window.

c_0^g	80	74	68	62	56	50
^{g}	3	3.4	3.8	4.2	4.6	5
t	$5 \cdot 10^{3}$					
	10^{6}	10^{5}	10^{4}	10^{3}	10^{2}	10^{1}
c_0^e	50.000	50.000	50.000	50.001	50.009	49.938

11 Conclusion

Many other tests were run apart from those results included in section §9 and in most cases a good estimate value for c_0 was obtained. However, the values for were often very far from the true value as was ∇J from the expected value of near zero. Even when ∇J was near zero, still did not approach it's true value.

We explain the result of varying and having marginal a ect on the other values by saying the system is insensitive.

We can further explain the failure of ∇J to tend to zero in all cases by the convergence factor in CONMIN, that resulted in the process completing with good (convergent) values for c_0 and the cost function before minimisation of ∇J had been achieved. With a more powerful and accurate computer it is possible to increase the machine accuracy parameter and adjust the EPS value for CONMIN which would allow convergence to occur at a later stage in the processing, thereby allowing more opportunity for minimisation to occur.

The programs, although tested as well as could be in the time allowed, are still likely to contain errors which further testing could eradicate if time permitted. Also more tests could be made with di erent combinations and magnitudes for the parameters used by the model to improve the performance.

The model exhibited resilience to a wide range of guess values (for c_0^g and ^{g}), generally achieving a good estimate for c_0. Good resilience was also recorded in the test for values of the covarience factor ≤ 1.

It was also concluded that had insigni cant in uence on the minimisation of J and hence in estimating the state variable.

It had been hoped to use real data collected from the M25 motorway but this stage was not reached because of time restriction, but it is anticipated that this will be done following the completion of this dissertation.

12 Appendix A- Glossary

Anticipation: the changes made by drivers to changing tra c conditions around them. Used in Payne's equations.

Continuity equation: di erential equation that describes the conservation of a conserved quantity.

Convection: changes in the mean tra c velocity caused be vehicles joining or leaving the ow. Used in Payne's equations.

Cost function: measures the bias between the observations and the model.

Data Assimilation: is the incorporation of observational data into a numerical model to produce a model state which most accurately describes the observed reality.

EPS: is the user supplied convergence parameter to CONMIN.

Hyperbolic: a system of partial di erential equations is hyperbolic if for $x_p + A(x)x_q; A(x)$ is diagonalisable and has real eigenvalues.

Insensitive Parameter: ability of a model or system to be una ected by widely ranging values of the parameter.

Relaxation: the tendency of tra c ow to approach an equilibrium velocity. Used in Payne's equations.

Stochastic process: a process whose behaviour is essentially non-deterministic, that is the system's subsequent state is a combination of the process's predictable actions and a random element.

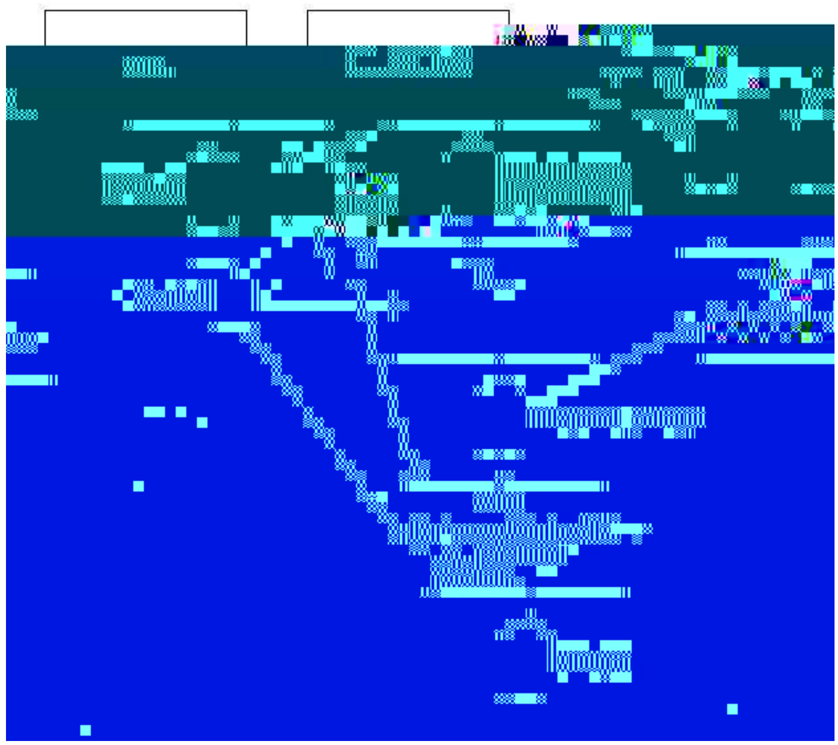
Tangent Linear approximation: An assumption used in applications of tangent linear models and adjoint models that the evolution of small perturbations in nonlinear models may be approximated by tangent linear (and adjoint) equations for nite time intervals (ref: 1).

TLM: Tangent Linear Model. A model, comprising tangent linear equations, that maps a perturbation vector, $x(t_1) = M\ x(t_0)$, from initial time t_0 to forecast time t_1. Where, M is the tangent linear operator and x is the model state vector.

Tra c ow model: formulates the relationships between tra c ow characteristics

13 Appendix B

13.1 Data Flow Diagram



13.2 Program Outlines

13.2.1 Introduction

All programs were written in Fortran using the Plato development application, a brief description of their function and the data input/output les is given below. A Control le is maintained of the initial values used by HLL together with other parameter values that are required to be passed between programs.

13.2.2 Roes

The rst program of the unlinearised model from which the HLL program was developed.

13.2.3 HLL_2 les

The Fortran implementation of the HLL model starts from initial values of the state vector and produces an output le of u at the nal time step. The constants c_0 and are then randomly perturbed and the model re-run. The cost function is calculated for each condition.

Output:
Control le output le containing parameter values
HLL_lin_state output le of containing values of u at each x position for each time step
HLL_01 output le of nal values of u using unperturbed values
HLL_02 output le of nal values of u using perturbed values
HLL

na28960(fro27(8( )28(t9)TJ/F28 118(t9)`1.9552 Tf 160.372 0 T03.153]TJ/F17 11.9552 Tf 9.99 0 Td 613.3]TJ/F17 '27(8(

each time step

Output:
TLM_01 containing nal time step values of u

13.2.5 ADJ

The program performs the Adjoint model processing, but in the reverse order to processing by HLL and TLM. (i.e. starting at the nal time and working backwards to the start time, and within each time step beginning with the greatest x value and working backwards to the smallest.) Again the equivalent HLL values of u are used at the start of each ${}^{0}x^{0}$ step (HLL values being accessed in reverse order to which they were produced) together with the nal time step values of u from TLM for the rst time step only.

The program includes the calculation of the inner products using the output of TLM and ADJ, to verify the correct working of the system.
Calculates inner product 1 = $(\ c_0)^2 + (\ \ )^2 + (\ u)^2$ (using the TLM_ nal_delta_u le values). Equivalent of $(M\ \ x) : (M\ \ x)$.

Calculates inner product 2 = $:$

The program includes the calculation of the gradient of the Cost function for use by CONMIN as detailed below:

$$\nabla J = 2 \cdot (\hat{c}_0 + \hat{\ })$$

Input:
Control le containing parameter values
HLL_lin_state containing all values of unlinearised **u** (to be used at the start of each `x' step)
TLM_ nal_delta_u containing nal time step values of u.
HLL_observs containing values of observations values at each x position for each time step

Output:
Control le updated to contain nal values of $\hat{c}_0$ and ^
Results of inner products calculations.

13.2.6 Merge_err

13.2.7 CONMIN

CONMIN is a computer subroutine that nds the values of c_0 and that produce a minimisation of the cost function. In summary, it is called from a small Fortran program and then repeadily calls HLL and ADJ to calculate the J and ∇J respectively until convergence of the values is achieved.

13.3 Control le values

Record 1: N, c_0,

14 Bibliography

References

[1] Danila Volpi, Estimation of Parameters in Tra c Flow Models Using Data Assimilation, 2009

[2] Department for Transport, vehicle licensing statistics, 2010, p.2

[3] Living Streets, http://www.livingstreets.org.uk/news/uk/-/driven-to-excess

[4] Chin Jian Leo, Robert L. Pretty, Numerical simulation of macroscopic continuum tra c models, 1990

[5] John C Strikwerda, Finite di erence schemes and partial differential equations, second edition, 2004

[6] P L Roe, Approximate Riemann solvers, parameter vectors, and di erence schemes, 1981

[7] A Harten, J M Hyman, and P D Lax, On nite di erence approximations and entropy conditions for shocks, Comm. Pure and Appl. Math., 1976, 29:297322

[8] Allan R Robinson, Pierre F J Lermusiaux, Overview of data assimilation, Harvard University, 2000

[9] N K Nichols, Data assimilation for the Earth system, p.11

[10] E Kalnay, Atmospheric Modelling, Data assimilation and predictability, 2003, p.154

[11] A Lawless, S Gratton, N Nichols, An investigation of incremental $4D-$Var using non-tangent linear models, 2004, p.6-7

[12] D F Shanno, K H Phua, Remark on algorithm 500 - a variable method subroutine for unconstrained nonlinear minimization. *ACM Trans. on Mathematical Software,*

[21] Ronald M. Errico, What is an adjoint model, 1997

[22] J F Muller, T Stavrakou, Inverse Modelling of CO Emissions, Belgian Institute for Space Aeronomy, 2006