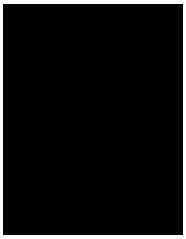
University of Reading

School of Mathematics, Meteorology & Physics



Estimation of parameters in traffic flow models using data assimilation

Danila Volpi

Supervisors: Dr Amos S. Lawless,

Dr Peter K. Sweby

Abstract

Payne Whitham model (PW) is a macroscopic second order tra□c model involving two parameters which directly influence the solution of the model, but which are independent from time and space.

Declaration

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

Signed..... Date.....

Contents

proximated by a linear model (tangent linear model). Obviously when the higher order terms become large this assumption does not hold anymore and we are out of the range of validity for the tangent linear model.

The implementation of 4D-Var consists of minimising the distance between the model trajectory and the observations (which is measured by the cost function). To minimise it we have used the conjugate gradient method which requires the availability of the gradient of the cost function.

We have approximated the gradient of the cost function with a finite difference scheme. This approximation will reveal to be quite crude and it will weigh on the minimisation accuracy. However, the results show the success of the optimal estimation: in fact the model trajectory estimated with the optimal parameters is more consistent with the true state.

The first chapter describes the main characteristic of the PW system and it briefly illustrates the numerical method which solves it. The second chapter introduces the technique of the parameter estimation for the general case using the 4D-Var.

In the third chapter it is presented the tangent linear model and it is investigated the validity time window through some tests.

A description of the 4D-Var implementation, the conjugate gradient method with its stopping criterion is presented in Chapter 4. Finally, the results of the assimilation are presented.

Chapter 1

The choice of the model

There are two different approaches for modelling the traffic depending on which aspects one is interested in.

There exist the microscopic models, which analyse the interactions between vehicles and in which attention is put on the behaviour of each car, its velocity and position; they are extremely precise, but the computational cost rises excessively with the increase of vehicles involved. The basic idea is that the velocity and acceleration of each car is adjusted according to the conditions ahead (car-following models).

On the other hand, the macroscopic approach aims to describe the general system, without looking specifically at each vehicle. This approach is definitely less accurate, but it enables the problem to be written in a mathematical compact system dependent on just a few variables, which makes it computationally efficient.

The macroscopic approach treats the traffic as a fluid flow, and applies the fluid dynamic laws in order to describe the evolution over time and space of

some average quantities such as the mass density and the mean velocity of the flux.[2] The analogy with fluid behaviour requires some assumptions:

the continuum hypothesis: we are going to consider the tra□c as a continuum fluid, uniformly distributed in the space. Obviously this condition is not physically satisfied by cars along a road; however, this assumption can be accepted as an approximation of the physical reality if the number of vehicles is large enough so that it makes sense to introduce macroscopic quantities;

the flux is conserved and there exists a conservation law (as for the fluid);

there exists a bijective correspondence between the density and the velocity and between the flux and the density; this latter correspondence is given by an equation of state[1]

In order to derive the equation for the conservation of the flux we consider an infinitesimal length [2] dx of road, as illustrated in figure (1.1), which is occupied at time t by dx vehicles (the analogue of the mass for a fluid), where is the density.

This quantity varies both when more vehicles arrive to occupy that infinitesimal length of road, or when some vehicles leave it. So the quantity of vehicles

[1] An equation of state is an equation describing the state of the matter under a given set of physical conditions. It provides a relation between two state functions associated with the matter.

[2] In fluid dynamics we usually consider the infinitesimal volume, but in this study we will consider a monodimensional road.

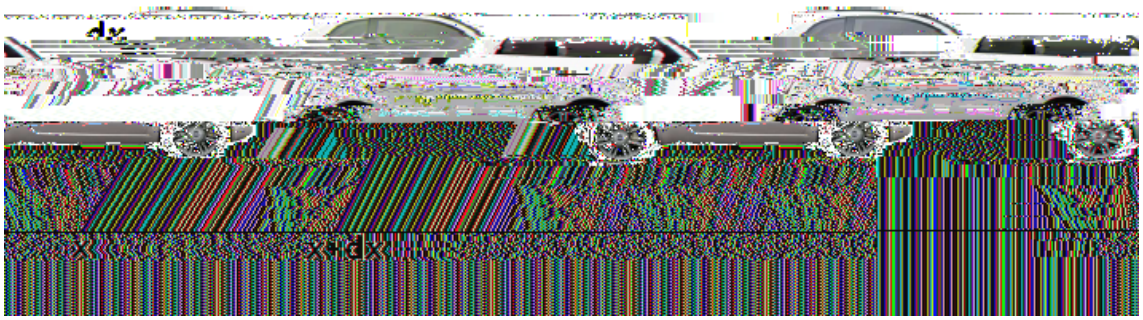


Figure 1.1: Infinitesimal rectangle of road

entering and leaving the length of road per unit time and space is given respectively by u where u is the velocity. We need to evaluate this quantity at the points x and $x + dx$. The derivative with respect of time is given by $\frac{@}{@t}dx$. We get:

$$\frac{@}{@t}dx = (\ u_x)dx \quad (\ u_{x+dx})dx$$

using the Taylor expansion for u_{x+dx} we get:

$$\frac{@}{@t}dx = (\ u_x)dx \quad (\ u_x)dx \quad (\frac{@\ u_x}{@x}dx) \qquad (1.1)$$

calling p21 7g90Tp2(F18 5.37206 0 Td [(p2 7g90TJ/F15 2g)]TJ/F22f11.9552 Tf 37.06 0

Therefore for a first order model we rewrite the equation (1.2) as:

$$\frac{@}{@t}$$

second order model is given by:

$$\begin{cases} \frac{\partial}{\partial t} + \frac{\partial}{\partial x}(\ u) = 0 \\ \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = a[\ ; u; \quad ; \quad u] \end{cases} \tag{1.6}$$

1.1 The Payne-Whitham model

We will use the Payne-Whitham (PW) model [12] that dates back to 1971, which is the prototype of the second order models.
The material model of the acceleration is the sum of relaxation and anticipation terms.

$$a[\ ; u; \quad ; \quad u] = a_r[\ ; u] + a_a[\ ; \ _x]$$

The first term models the tendency of the drivers to travel at the free flow speed $U(\ )$ depending on the congestion on the road:

$$a_r[\ ; u] = \frac{U(\quad u)}{} \tag{1.7}$$

where u is the actual speed and (> 0) is the relaxation time, i.e. the drivers' reaction time in adjusting their speed to the 'maximal and out of danger' velocity $U(\ )$

where c_0 (>0) is the traffic sound speed, which measures the ratio of the anticipation constant with the relaxation time.

We can now write the system for the PW model:

$$\begin{cases} \frac{d}{dt} + \frac{d}{dx}(\ u) = 0 \\ \frac{du}{dt} + u\frac{du}{dx} = \frac{U(\ )-u}{\ } \quad c_0^2 \frac{d}{dx} \end{cases} \tag{1.9}$$

In this system there appear two parameters c_0 and , which are not dependent on time and space, but which directly affect the results of the system.

In the next chapters they will be the target of our study, with the aim of estimating their optimal values for insertion into the model and produce an improved forecast.

In order to investigate the solutions of the system, let us show that it is hyperbolic[4].

The dynamic system (1.9) has the following matrix representation:

$$q_t + \begin{matrix} q \\ (u^2 + c_0^2) \end{matrix}_x = \begin{matrix} 0 \\ \frac{(U(\ )-u)}{\ } \end{matrix} \tag{1.10}$$

We write the system in the form $U_t + A(U)U_x = s(U)$, with $A(U)$ the jacobian matrix:

$$q_t + \begin{matrix} 0 & 1 \\ c_0^2 & u^2 \quad 2u \end{matrix} q_x = \begin{matrix} 0 \\ \frac{(U(\ )-u)}{\ } \end{matrix}$$

where the right hand side is the source term. The eigenvalues and the respective eigenvectors of the above matrix are:

$$_{1;2} = u \quad c_0 \qquad e_{1;2} = \begin{matrix} 1 \\ u \quad c_0 \end{matrix} \tag{1.12}$$

Since the eigenvalues are real and distinct the system is hyperbolic, and this implies that information propagates along characteristic directions and that discontinuities can develop spontaneously, and either persist or expand.
The solution consists of two waves each moving with a speed given by the characteristic. Since the flux fluctuations are nonlinear it is di□cult to derive exact solutions to the system for general initial data[5].
The two waves can either be a shock or a rarefaction:

shock: initial data may generate a discontinuity in the solution, that is not acceptable as a classic solution of a partial di□erential equation. Therefore a shock is introduced to continue the discontinuity propagation;

rarefaction: because of a discontinuity in the initial data the characteristics do not fill the whole plane $(x; t)$. Therefore a rarefaction is generated.

We therefore need a condition to select the correct physical solution: the entropy condition (which assures that characteristic goes into the shock).
The solutions of the PW model are not always acceptable; there can be cases of negative velocity and the analysis of characteristic speeds shows

[5] It is possible to derive the analytical solution that holds just at a certain time called the collapsing time [3].

that sometimes the vehicles dynamics can a□ect forward tra□c dynamics [2]. This leads to a violation of tra□c dynamics[6]

Even if the PW model has some limitations, we have chosen it as a starting point for the study of the parameters estimation in the tra□c flow model.

1.2 The numerical model

The hyperbolicity of the system illustrated above justifies the use of special schemes to approximate it numerically [5]. In particular schemes need to be:

able to capture shocks (i.e. the correct speeds of the discontinuities);

consistent with the original system;

stable over time (in order to a27(the)-326(disBtoidisBtot16b326Bto9.al)-326(systLsln)

Roe's method is not entropy satisfying and therefore it is necessary to add a standard entropy correction term.

Roe scheme is a first order upwind sheme, which assures that its accuracy of order t in time and x in space.

It follows a brief description of the Roe's flux difference splitting algorithm for Payne's model [1].

Let us call $u = \begin{matrix} u \\ q \end{matrix}$ where q is the flux u and let us consider f the flux function; we can write the PW model as:

$$u_t + f(u) = s(u)$$

where s is the source term; we are looking for an approximation such that:

$$\begin{aligned} u &= {}_1 e_1 + {}_2 e_2 \\ f &= {}_1 {}_1 e_1 + {}_2 {}_2 e_2 \end{aligned} \tag{1.13}$$

For the PW formulation the approximate values are:

$$\begin{aligned} \hat{}_1 &= \hat{u} \quad c_0 \qquad \hat{e}_1 = \begin{matrix} 1 \\ \hat{u} \quad c_0 \end{matrix} \qquad \hat{e}_2 = \begin{matrix} 1 \\ \hat{u} + c_0 \end{matrix} \\ \hat{}_2 &= \hat{u} + c_0 \\ \hat{}_1 &= \frac{1}{2} + \frac{1}{2}\frac{\hat{}}{c_0} \quad u \qquad \hat{}_2 = \frac{1}{2} \quad \frac{1}{2}\frac{\hat{}}{c_0} \quad u \\ \hat{u} &= \frac{p_l u_l + p_r u_r}{p_l + p_r} \qquad \hat{} = p_l p_r \\ \hat{}_1 &= x\frac{\hat{s}_1 \hat{}_2 \quad \hat{s}_2}{2c_0 \hat{}_1} \qquad \hat{}_2 = x\frac{\hat{s}_2 \hat{} hats_1 \quad {}_1}{2c_0 \hat{}_2} \\ \hat{s} &= \frac{1}{2}[s_l + s_r] \qquad s = \begin{matrix} s_1 \\ s_2 \end{matrix} \end{aligned} \tag{1.14}$$

where with 0 we are calculating the velocity in the middle of each cell and with the indices $_l$ and $_r$ we are indicating the velocity at the space point respectively behind and ahead.

The final difference scheme then becomes:

$$w_i^{n+1} = w_i^n \quad \frac{t}{x}[f_{i+1=2}^n \quad f_{i-1=2}^n] + \frac{t}{2}[\hat{s}_{i+1=2}^n + \hat{s}_{i-1=2}^n] \tag{1.15}$$

where the index $_i$ indicates the space step and $_n$ the time step. $f_{i+1=2}$ is the interface flux between cells and is given by:

$$f_{i+1=2}^n = \frac{1}{2}\left[f_{i+1}^n + f_i^n \right] \sum_{m=1}^{2} j \hat{} j_m \, (\hat{}_m + \hat{}_m)\hat{e}_m \, {}^n_{i+1=2} \tag{1.16}$$

1.3 The structure of the numerical model

Our numerical model is divided in four subroutines: the two bottom level subroutines are *Ve* and *PWdecomp*, which calculate respectively the out of danger velocity $U(\ )$ and the approximate values of equation (1.14). The subroutine *fluxes* computes the interface flux between cells at each space-step, using the two bottom level subroutines. Finally the top level program implements the finite difference scheme of equation (1.15) for all the time-steps, using the values calculated in *fluxes* (code in the Appendix A).

The figure (1.2) is the result of running the model for 200 time-steps and 100 space-steps, $t = 2{:}510^{-5}$ and $x = 0{:}01$. The initial velocity and density are two constant states with a discontinuity in $x = 0{:}5$.

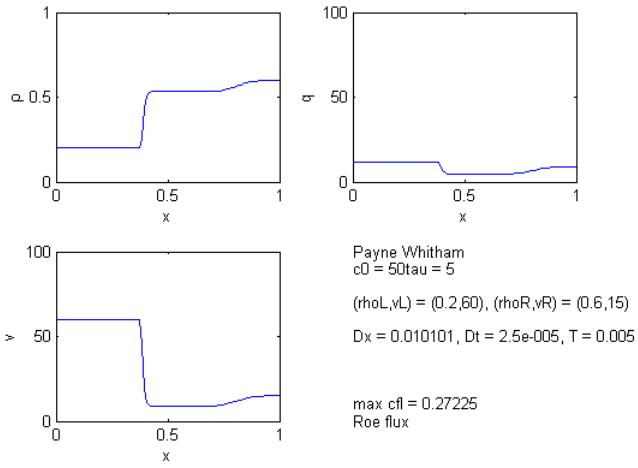


Figure 1.2: (from the top left) density, flux and velocity along space at the final time-step

Chapter 2

Data Assimilation

As we have mentioned previously the variables involved in the PW dynamic system are:

$(x;t)$ = tra□c density (vehicles per km).

$u(x;t)$ = mean speed (km per hour).

$q(x;t) = u$ = tra□c flow (vehicles per hour).

$U(\ )$= free flow velocity.

= relaxation time.

c_0= tra□c sound speed.

The last two variables are parameters that are usually determined empirically.

Their values directly impact upon the performance of the model. Thus in order to improve the numerical simulation, our aim is to estimate optimal

parameters making use of all the information available [7]. This information includes any kind of observational data available, information from previous forecasts (also called background), or any statistical information relating to their respective errors.

We have carried out the parameter estimation procedure through the technique of data assimilation.

Data assimilation has the purpose to incorporate all the available information into the numerical model in order to determine the state of the system as accurately as possible.

There are two main families of assimilation techniques: sequential and variational. While variational techniques proceed by the global fitting of an assimilation model to the available information, sequential assimilation steps through the observations sequentially in time.

The current success of variational methods is attributed to both their capability in handle large size models and the availability of e□cient minimization algorithms [6].

We have chosen to apply the four dimensional variational method (4D-Var). This technique allows us to estimate the parameters, taking into account both the information about the dynamics from the dynamical model and the information about the true state contained in the observations and the background.

Moreover the 4D-Var method enables us to incorporate into the numerical model the observations at their correct time and the method combines the previous data with the currently available ones. This guarantees the information recovery not only of the variables whose observations are available

2.1 4D-Var parameter estimation

Let us introduce some of the features that take part in the technique for the parameter estimation: [1]

Notation	Variables
$x_t = \begin{matrix} z_t \\ p_t \end{matrix}$	true state of velocity, density, flux (given from z_t) and the parameters c_0 and (from p_t).
p_b	background state. It comes from a good quality forecast
p_g	initial guess of the parameters c_0 and
p_a	final analysis of the parameters c_0 and
y	vector of observations
h	observation operator

For random, unbiased [2] Gaussian errors the minimum variance estimate is equivalent to the maximum likelihood estimate, which we can find by minimising the cost function (that measures the weighted sum of squares of distances to the background state p_b and the observations y distributed over a time interval $[t_0; t_n]$).

The cost function is given by:

$$J(\mathrm{p}) = (\mathrm{p}(t_0) \quad \mathrm{p}_b(t_0))^T B^{-1}(\mathrm{p}(t_0) \quad \mathrm{p}_b(t_0)) + \sum_{i=0}^{n} (\mathrm{y}_i \quad \mathrm{h}_i(\mathrm{x}_i))^T R_i^{-1}(\mathrm{y}_i \quad \mathrm{h}_i(\mathrm{x}_i)) \tag{2.1}$$

with x_i = m(c901 Tf 6.587 0 Td Mnh.993 Td [((203)]TJ/F22 11.9552 Tf 11.206 30844

su□cient condition for the minimum, which is unique).

To linearize m we use the Taylor expansion of the non linear model with a perturbation: let consider the non linear model at time t_i:

TLM does not retain the original features of the nonlinear system, therefore it is not a good approximation and the parameters estimated by minimising the cost function are not optimal values. It follows that the accuracy of the TLM determines the accuracy of the variational assimilation.

2.3 Correctness test

There follows a test to verify the correctness of the TLM [10].
Let consider the linear model with the parameter perturbed, which we can write as:

$$m(x_0; p + \quad p; t_i; t_0) \approx M \quad p + H{:}o{:}t + m(x_0; p; t_i; t_0)$$

then if we put:

$$p = \quad p^0 \tag{2.3}$$

where p^0 represents the perturbation of the parameters, we can write:

$$m(x_0; p + \quad p^0; t_0; t_i) \quad m(x_0; p; t_0; t_i) \approx M \quad p^0 + O(\ ^2) \tag{2.4}$$

On the left hand side we have the difference between the state originated by the perturbed parameters and the one originated by the true parameters; this gives us the perturbation of the state (z_{NL}). M p^0 is the perturbation of the state given by the TLM (z_L). Rearranging the eq. (2.4) we can write:

$$km(x_0; p + \quad p^0; t_0; t_i) \quad m(x_0; p; t_0; t_i) \quad M \quad p^0 k$$

that is equivalent to:

$$\frac{k z_{NL} \quad z_L k}{k z_L k} \approx O(\ ) \tag{2.6}$$

For small we have that $z_{NL} \approx z_L$ and therefore the left hand side of (2.5) is zero.

Therefore a standard method of validating the TLM is to show that the left hand side of (2.5) tends linearly to zero for small .

2.4 Validity test

As we explained in section (2.2) it is necessary to detect the validity time of the tangent linear hypothesis. Outside that interval the tangent linear model will not be applicable and our study becomes ine□cient.

The validity time is the period over which the linear model exhibits a behaviour similar to the non linear system. Therefore we are going to compare the trajectories of the perturbated state given by the linear and nonlinear model, and their di□erence [9].

2.5 Minimization of the cost function

Since the numerical method that we will use to minimize the cost function is a gradient based method we need to calculate $r J$:

$$r J = r J_b + r J_o$$

where J_b is the background part of the cost function and J_o is the observation part. The gradient of the background part is given by:

$$r J_b = 2B^{-1}(p \quad p_b)$$

while for the observation term the gradient is:

$$r J_o = \quad 2 \sum_{i=0} (H_i M_i M_{i-1} ::: M_1)^T R_i^{-1}(y_i \quad h_i(x_i)) \tag{2.7}$$

Therefore the gradient of the cost function evaluated at p is given by:

$$r J(p) = 2B^{-1}(p \quad p_b) \quad 2 \sum_{i=0} (H_i M_i M_{i-1} ::: M_1)^T R_i^{-1}(y_i \quad h_i(x_i)) \tag{2.8}$$

where $(M_i M_{i-1} ::: M_1)^T$ is given by the adjoint model [13].
This model is built from the linear model:

$$x^{n+1} = M x^n$$

then the adjoint model is given by

$$\hat{x}^n = M^T \hat{x}^{n+1} \tag{2.9}$$

In ou.096 4.2 Tf 159.288 -39.436 Td [(Td [(N16 11.9552 Tf 145.928 -5.312 Td [((2.-4.936 Td [(=)

calculate the gradient as:

$$r\,J(p_i) \approx \frac{J(p_i + @p_i) \quad J(p_i)}{@p_i} \tag{2.10}$$

In general if we needed parameter estimation for a large amount of parameters, or if we had a di□erent parameter for each spatial point, using the finite di□erence approach would mean run the scheme for each parameter, which would become computationally expensive; therefore in these cases the adjoint model would be preferred (of which we would need just one run). However in this study it is acceptable to use the finite di□erence scheme.

Chapter 3

Bulding of 4D-Var for the traffic model

We now want to apply the parameter estimation technique to the PW model. Since we do not have any previous forecast data, the background term does not appear.

Moreover we assume that the instruments which collect the information, can measure data of the velocity and the density only, thus the observation vector is given by $\begin{pmatrix} u_i^o \\ \rho_i^o \end{pmatrix}$ The cost function (2.1) has the form:

$$J(c_0) = \sum_{i=0}^{N} \begin{pmatrix} u_i^o \\ \rho_i^o \end{pmatrix} - \begin{pmatrix} u_i \\ \rho_i \\ q_i \end{pmatrix} \Big)^T R_i^{-1} \begin{pmatrix} u_i^o \\ \rho_i^o \end{pmatrix} - \begin{pmatrix} u_i \\ \rho_i \\ q_i \end{pmatrix} \Big) \quad (3.1)$$

where u_i is the vector state given by the i^{th} run of the non linear model. The gradient then follows from eq. (2.7):

$$r J \quad c_0 \quad = \quad 2 \sum_{i=0} (M_i M_{i-1} \quad M_1)^T H_i^T R_i^{-1} \quad u_i^o \quad H_i \begin{pmatrix} 0 & 1 \\ u_i & \\ & \\ i & \\ q_i & \end{pmatrix} \qquad (3.2)$$

We can proceed with the building of the tangent linear model. The TLM for our model is reported in Appendix B.

Particular attention needs to be paid to the linearisation of the interface flux between cells given by equation (1.16) because of the presence of the absolute value; we need to linearise an equation of the form:

$$y = a \quad j \quad j$$

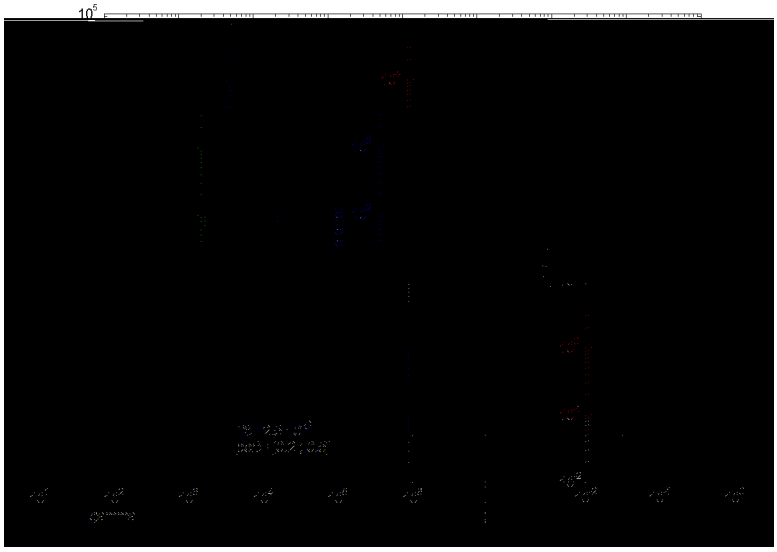
3.1 TLM correctness test

We perform the correctness test of the TLM presented in the section (2.3): we choose a perturbation vector $p = c_0$ and we run the non linear model twice, once with the initial parameters and then with the parameters perturbed. The difference between the two resulting states gives the perturbed state z_{NL}, which needs to be compared with the perturbed state given from the TLM z_L. The absolute error of the perturbed state is:

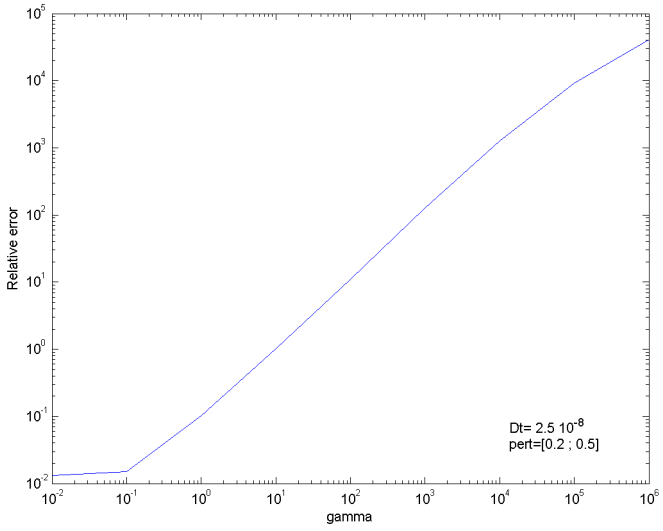
$$\text{abs err} = \underbrace{m(x; p + p; t_i; t_0) - m(x; p; t_i; t_0)}_{z_{NL}} - \underbrace{M_{\cdot} p}_{z_L}$$

The ratio between the absolute error and z_L gives the relative error. We then plot its L^2 norm (eq (2.5)) against , (defined in eq (2.3) as $p = p^0$) and we expect it to tend linearly to zero (fig (3.1)). For the following graphs we have used the values:

$$c_0 = 50 \qquad c_0 = 0.2$$
$$= 5 \qquad = 0.5$$



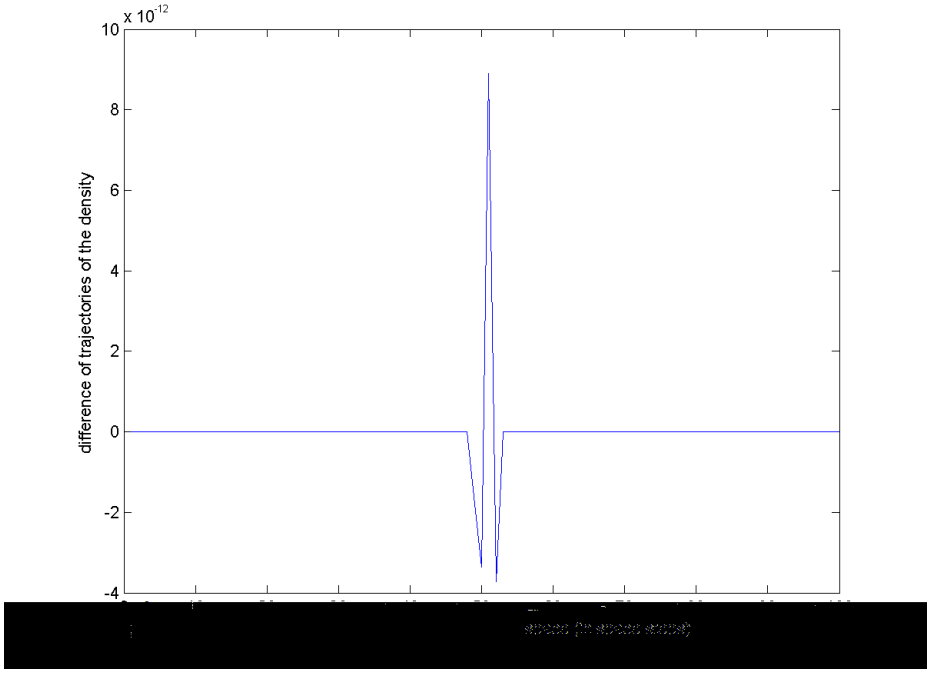
(a)



(b)

Figure 3.1: Graph to illustrate the correctness of the TLM after 10 timesteps for: (a) flux, (b) velocity

(a)



(b)

Figure 3.3: (a) Density perturbation trajectory (order of 10^{-8}) (b) difference between linear and non linear perturbation of the density (order of 10^{-12})

3.2 Time window test

In order to investigate the length of the time window for which the tangent linear hypothesis holds, we are going to calculate the relative error for di□erent time windows. We need to make sure that the numerical model remains stable (stability criterion: $\frac{}{t}$

For further proof we compare the values of z_{NL} and z_L at time 1000 and notice that the large relative error is due to the fact that perturbations given from the nonlinear model are positive values, while the ones given from the linear model are negative.

Moreover the relative errors of the density, the flux and the velocity oscillate quite regularly. Observing a sequence of local maximum and minimum we notice that the oscillation is due to the fact that the wave obtained from the TLM moves faster than the one from the NLM; therefore a local minimum of the relative error (fig. (3.4(a), 3.4(b),3.4(c))) corresponds to the instant in which the waves are in phase, and viceversa a maximum (in the relative error) occurs when the waves are out of phase.

This is shown by fig (3.5) that represents a sequence of local max-min-max-min-max in the relative errors of the density. The table (3.1) shows the maximum relative error for 900 time steps:

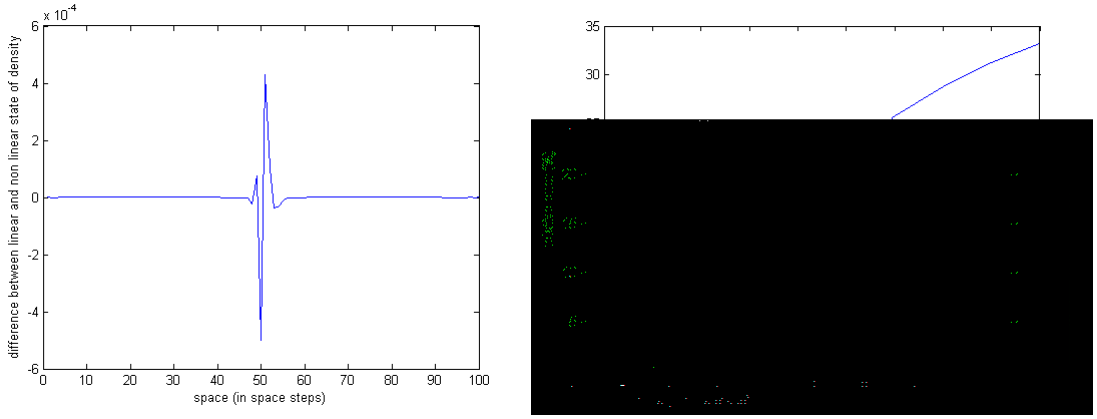
Variables	max rel.err (%))
density	48.7783
flux	

Time window	timesteps length	max rel. err. velocity (%)	max rel. err. density (%)	max rel. err. flux (%)
$5 \quad 10^{-6}$	$2{:}5 \quad 10^{-8}$	0.2907	1.1713	1.0933
$5 \quad 10^{-5}$	$2{:}5 \quad 10^{-7}$	3.3142	9.3612	10.6297
$5 \quad 10^{-4}$	$2{:}5 \quad 10^{-6}$	17.9544	33.8193	39.6312

Table 3.4: $p = {}^{c_0}_{} = {}^{50}_{5}$, $p = {}^{0:2}_{0:02}$, 200 time steps.

We notice that for a time window of the order of 10^{-4} the function of the relative errors is monotonically increasing (fig 3.6(b), (3.7(b)), (3.7(d))); this means that there is not any delay in the linear and non linear wave movement over time and the relative errors depend only on the time steps.

The figures (3.6(a)), (3.7(a)) and (3.7(c)) show the difference between the linear and non linear trajectories for a time window of $2{:}5 \quad 10^{-4}$



(a) (b)

Figure 3.6: (a) Difference between the states of the density, (b) relative error

3.3 Summary of the results

The tests illustrated in this chapter have led to the following conclusions:

The code for the tangent linear model is correct; however we get some rounding errors in testing the density due to the small order of magnitude (10^{-8}) ;

the tangent linear model is a good approximation for a maximum time window of 5 10^{-3} with the relative errors within the range 20-60 %. For bigger time intervals the higher order terms become large, and the tangent linear hypothesis is not guaranteed;

the oscillation of the relative error is due to a delay in the propagation of the nonlinear wave respect to the linear wave;

the parameter and its perturbation do not excessively a□ect the relative errors;

to decrease the size of the relative error we need to choose a time window of the order of 10^{-4}. In such a small time window the relative error increases monotonically and its maximum corresponds to the last time step.

Chapter 4

Minimization of J and parameter estimation

To code the 4D-Var we modify the full 4D-Var program for the Lorenz equations from [16].

In order to set the true state (that we assume to be known) we run the nonlinear model with the true parameters.

In this study we use artificial data, so we generate observations with some random noise and we set the linearised observation operator H.

Choosing some first guess parameters, we calculate the cost function (2.1) and its gradient. In order to evaluate the gradient using the finite di□erent scheme of eq (2.10) we choose some perturbation of the parameters and run the nonlinear model with the first guess parameters perturbed, and we calculate the cost function of the state perturbed.

The examples given in this chapter have the perturbation set at 10^{-4}. Appendix C illustrates the code to calculate the cost function and its gradient.

4.1 Minimisation

As mentioned before, we use the conjugate gradient method (CG) that is a gradient based method (i.e. it needs the availability of the values of the gradient). It is implemented following a line search strategy in the directon of the steepest descent (see more details about the method in chapter 5 of [15]). We use one of the variant of this method, which is the Polak-Ribière method [15].

Since the CG is an iteration method it is necessary to select a proper stopping criterion which ensures an accurate approximation of the minimum.

Our stopping criterion is going to be the Lawless-Nichols criterion [14] which is:

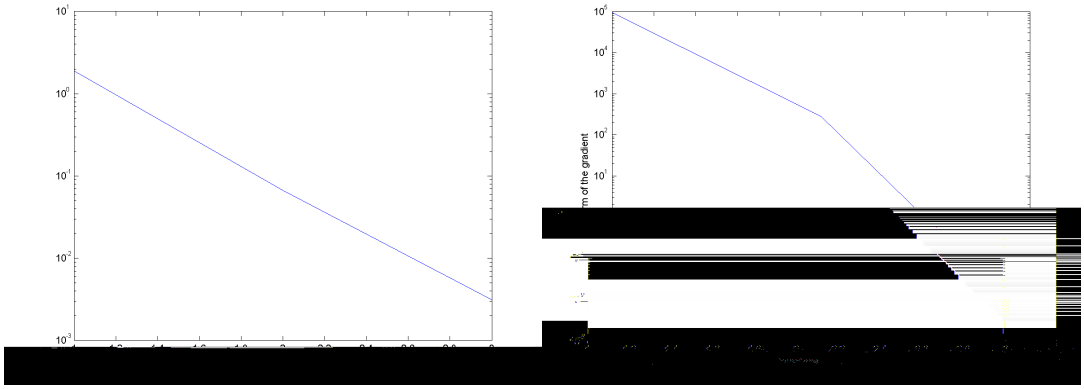
$$\frac{k\, r\, J^{i} k}{k\, r\, J^{0} k} < \qquad (4.1)$$

where is the user-set tolerance, and the index of the gradient is referred to the timestep.

However, since our approximation is crude, the error becomes significant before the stopping criterion is reached (see next section).

4.2 Results

As we exlpained in the previous section the crude approximation of the gradient makes the minimisation terminate. This causes less iteration and therefore the norm of the gradient reaches the order of 10^{-3} (figure (4.1)).



(a) (b)

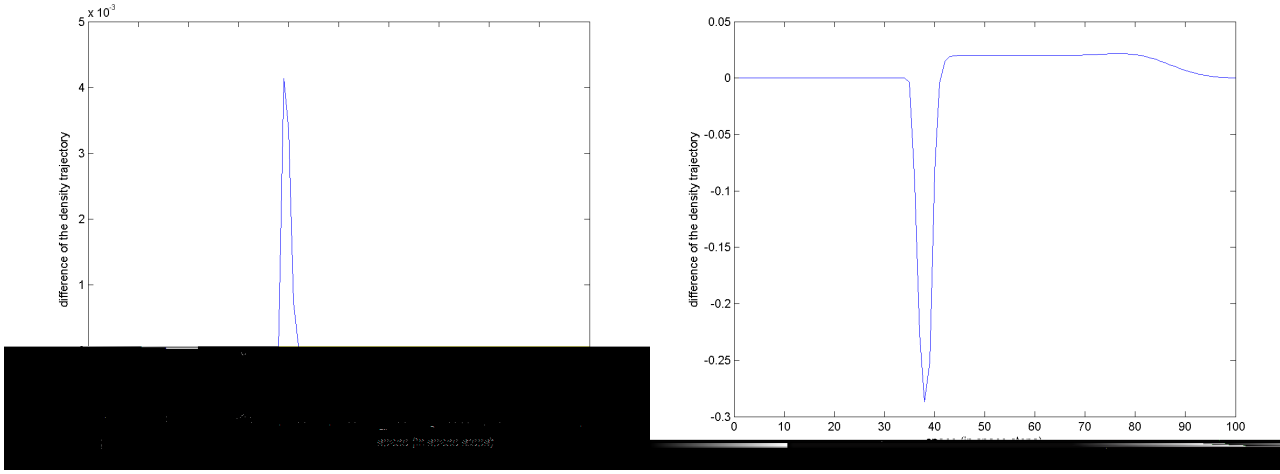
Figure 4.1: $k\,r\,Jk$ in its path towards 0. (a) time window of 2:5 10^{-4}, (b) time window of 5 10^{-3}

We report here the results of two experiments; the values used are:

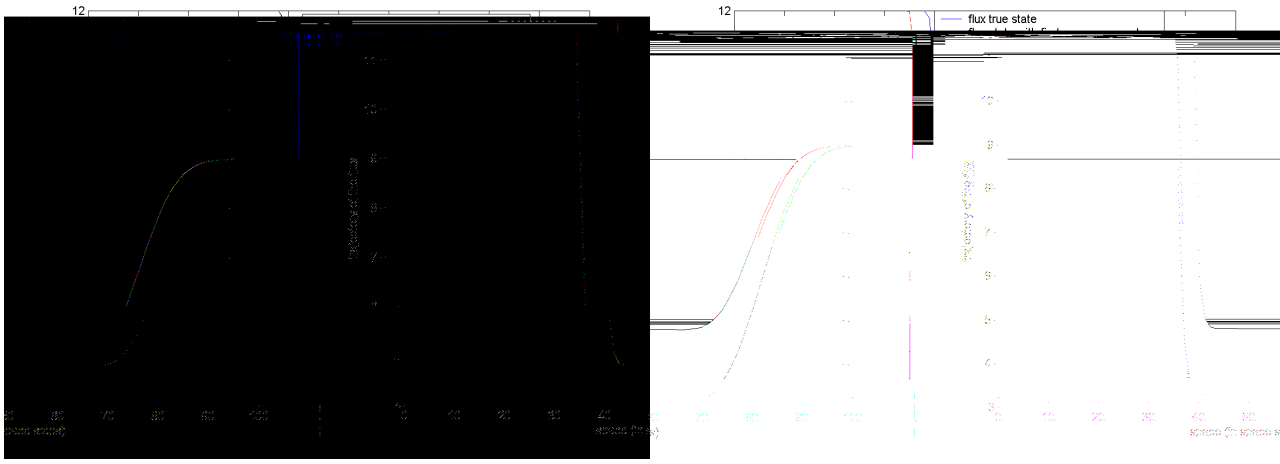
$$c_0^t = 50 \quad c_0^g = 55$$
$$^t = 5 \quad ^t = 7$$

where with ^{g} we indicate the first guess parameters and with ^{t} the true ones. The frequency of the observations is set to 1 (i.e. at each timesteps), and the variance for the observational noise is set to 4 10^{-4}

Case	T window



(a) (b)



(c) (d)

Figure 4.2: Results from case B. (a)(b)Di□erence of the density trajectories: (a) true state and state after the parameter analysis, (b) true state and state from the first guess; (c),(d) trajectories of the flux compared with the true trajectory (c) with parameter estimation (d with first guess parameters).

Chapter 5

Conclusions and further work

We succeeded in improving the forecast with the optimal parameters estimated. However, the approximation of the gradient of the cost function is crude and the error become significant before the stopping criterion is satisfied; this leads to a less accurate minimisation.

smaller time window; this happens because a larger time window leads also to more observations to include in the non linear model, and also more time for the waves to propagate.

Further, an interesting point would be investigate the reason for the delay of the non linear wave propagation respect to the linear one. This would help

Appendix A

Non linear model code

-Subroutine Ve

```
function [omega] = Ve(rho)
omega = (tanh(1./rho-2)-tanh(-2));
end
```

-Subroutine PWdecomp

```
function [lambda,X,alpha,beta,lamL,lamR]=PWdecomp(uL,uR,scell,c0)
% perforc=lL,4L[omega]perfoRoe[omegap
```

```
X(1,2) = 1;
X(2,2) = lambda(2);

alpha(1) = ((vtilde+c0)*(uR(1)-uL(1)) - (uR(2)-uL(2)))/(2*c0);
alpha(2) = (-(vtilde-c0)*(uR(1)-uL(1)) + (uR(2)-uL(2)))/(2*c0);

beta(1) = (lambda(2)*scell(1)-scell(2))/(2*c0);
beta(2) = (scell(2)-lambda(1)*scell(1))/(2*c0);

lamL(1) = (uL(2)/uL(1)-c0);
lamL(2) = (uL(2)/uL(1)+c0);
lamR(1) = (uR(2)/uR(1)-c0);
lamR(2) = (uR(2)/uR(1)+c0);

end
```

-Subroutine fluxes

```
function [h,s] = fluxes(u,Dt,Dx,c0,tau)
%calculates and applies Roe increments - WITHOUT entropy fix
%

imax = size(u,2);

evals=zeros(imax,1);

l1min=10000;l1max=-10000;l2min=10000;l2max=-10000;
```

```
f(1,:) = u(2,:);
f(2,:) = u(2,:).^2./u(1,:)+c0^2*u(1,:);
source = [zeros(1,imax);(u(1,:).*Ve(u(1,:))-u(2,:))/tau];
s = Dx*(source(:,1:imax-1)+source(:,2:imax))/2;

for i = 1:imax-1
    [lambda,X(:,:,i),alpha,beta,lamL,lamR] = PWdecomp(u(:,i),u(:,i+1),...
        s(:,i),c0);

    h(:,i) = (f(:,i)+f(:,i+1))/2 ...
               -((alpha(1)+beta(1)/lambda(1))*abs(lambda(1))*X(:,1,i)+ ...
                 (alpha(2)+beta(2)/lambda(2))*abs(lambda(2))*X(:,2,i))/2;



%   collect extremes of wavespeeds
    l1min = min(lambda(1),l1min);
    l1max = max(lambda(1),l1max);
    l2min = min(lambda(2),l2min);
    l2max = max(lambda(2),l2max);
end
end
```

-Subroutine RFtra□c

```
function [x,rho,q,v] = RFtraffic(rhoL,vL,rhoR,vR,imax,tmax,nmax,c0,tau)
% traffic Riemann problem
```

```
Dx = 1/(imax-1);
%Dt = tmax/nmax;
Dt=0.000000025;

x = 0:Dx:1;
for i = 1:imax
    if (x(i)<=0.5)
        u(1,i) = rhoL;
        u(2,i) = rhoL*vL;
    else
        u(1,i) = rhoR;
        u(2,i) = rhoR*vR;
    end
end
rho = u(1,:);
q = u(2,:);
v = u(2,:)./u(1,:);
for n=1:nmax
    [h,s] = fluxes(u,Dt,Dx,c0,tau);
    for i = 2:imax-1
        u(:,i) = u(:,i)-(Dt/Dx)*(h(:,i)-h(:,i-1))+...
        (Dt/Dx)*(s(:,i)+s(:,i-1))/2;
    end
    rho = u(1,:);
```

```
    q = u(2,:);
    v = u(2,:)./u(1,:);

end
```

Appendix B

Tangent linear model code

-Subroutine VeTL

```
function [omega_TL] = Ve_TL(rho,rho_p)
omega_TL = -(rho_p./(rho.^2)).*(1./(cosh(1./rho-2)).^2);
end
```

-Subroutine PWdecompTL

```
function [lambda,lambda_p,X,X_p,alpha, beta,alpha_p,beta_p,lamL_p,lamR_p]...
    =PWdecomp_TL(uL,uR,uL_p,uR_p,scell_p,scell,c0_p,c0)
vtilde = (uR(2)/sqrt(uR(1))+uL(2)/sqrt(uL(1)))/(sqrt(uR(1))+sqrt(uL(1)));

vtilde_p=(((uR_p(2)*sqrt(uR(1))-uR(2)*0.5*1/sqrt(uR(1))*uR_p(1))...
    /uR(1)+(uL_p(2)*sqrt(uL(1))-uL(2)*0.5*1/sqrt(uL(1))*uL_p(1))/uL(1))...
    *(sqrt(uR(1))+sqrt(uL(1)))-(uR(2)/sqrt(uR(1))+uL(2)/sqrt(uL(1)))...
    *(0.5*1/sqrt(uR(1))*uR_p(1)+0.5*1/sqrt(uL(1))*uL_p(1)))/...
    ((sqrt(uR(1))+sqrt(uL(1)))*(sqrt(uR(1))+sqrt(uL(1))));
lambda(1) = vtilde-c0;
```

```
lambda(2) = vtilde+c0;

lambda_p(1) = vtilde_p-c0_p;
lambda_p(2) = vtilde_p+c0_p;

X(1,1) = 1;
X(2,1) = lambda(1);
X(1,2) = 1;
X(2,2) = lambda(2);

X_p(1,1) = 0;
X_p(2,1) = lambda_p(1);
X_p(1,2) = 0;
X_p(2,2) = lambda_p(2);

alpha(1) = ((vtilde+c0)*(uR(1)-uL(1)) - (uR(2)-uL(2)))/(2*c0);
alpha(2) = (-(vtilde-c0)*(uR(1)-uL(1)) + (uR(2)-uL(2)))/(2*c0);

beta(1) = (lambda(2)*scell(1)-scell(2))/(2*c0);
beta(2) = (scell(2)-lambda(1)*scell(1))/(2*c0);

alpha_p(1) = (((vtilde_p+c0_p)*(uR(1)-uL(1))+(vtilde+c0)...
    *(uR_p(1)-uL_p(1))-(uR_p(2)-uL_p(2)))*2*c0-2*c0_p...
    *((vtilde+c0)*(uR(1)-uL(1)) - (uR(2)-uL(2))))/(4*c0*c0);
```

```
alpha_p(2)=((-(vtilde_p-c0_p)*(uR(1)-uL(1))+(vtilde-c0)...
    *(uR_p(1)-uL_p(1))+(uR_p(2)-uL_p(2)))*2*c0-2*c0_p...
    *(-(vtilde-c0)*(uR(1)-uL(1)) + (uR(2)-uL(2))))/(4*c0*c0);

beta_p(1) = (2*c0*(lambda_p(2)*scell(1)+lambda(2)*scell_p(1)-scell_p(2))...
    -2*c0_p*(lambda(2)*scell(1)-scell(2)))/(4*c0*c0);
beta_p(2) = (2*c0*(scell_p(2)-lambda_p(1)*scell(1)-lambda(1)*scell_p(1))...
    -2*c0_p*(scell(2)-lambda(1)*scell(1)))/(4*c0*c0);

lamL_p(1) = ((uL_p(2)*uL(1)-uL(2)*uL_p(1))/(uL(1)*uL(1))-c0_p);
lamL_p(2) = ((uL_p(2)*uL(1)-uL_p(1)*uL(2))/(uL(1)*uL(1))+c0_p);
lamR_p(1) = ((uR_p(2)*uR(1)-uR(2)*uR_p(1))/(uR(1)*uR(1))-c0_p);
lamR_p(2) = ((uR_p(2)*uR(1)-uR_p(1)*uR(2))/(uR(1)*uR(1))+c0_p);
end
```

- Subroutine fluxesTL

```
function [h_p,s,s_p] = fluxes_TL(u,u_p,Dt,Dx,c0_p,c0,tau,tau_p)
imax = size(u,2);

l1min=10000;l1max=-10000;l2min=10000;l2max=-10000;

[omega]=Ve(u(1,:));
[omega_p]=Ve_TL(u(1,:),u_p(1,:));

f(1,:) = u(2,:);
f(2,:) = u(2,:).^2./u(1,:)+c0^2*u(1,:);
```

```
f_p(1,:) = u_p(2,:);

for k=1:imax
f_p(2,k) = (2*u(2,k)*u_p(2,k)*u(1,k)-u_p(1,k)*(u(2,k)^2))...
    /(u(1,k)^2)+2*c0*c0_p*u(1,k)+c0^2*u_p(1,k);
end
source = [zeros(1,imax);(u(1,:).*omega-u(2,:))/tau];
s = Dx*(source(:,1:imax-1)+source(:,2:imax))/2;

source_p=[zeros(1,imax);
    (tau*(u_p(1,:).*omega+u(1,:).*omega_p-u_p(2,:))-...
    tau_p*(u(1,:).*omega-u(2,:)))/(tau*tau)];

s_p = Dx*(source_p(:,1:imax-1)+source_p(:,2:imax))/2;

for i = 1:imax-1
    [lambda,lambda_p,X(:,:,i),X_p(:,:,i),alpha,beta,alpha_p,beta_p,lamL_p,...
        lamR_p] = PWdecomp_TL...
        (u(:,i),u(:,i+1),u_p(:,i),u_p(:,i+1), s_p(:,i),s(:,i),c0_p,c0);

  h_p(:,i) =(f_p(:,i)+f_p(:,i+1))/2-0.5*((alpha_p(1)+(beta_p(1)*lambda(1)-...
    beta(1)*lambda_p(1))/((lambda(1))^2))*X(:,1,i)*abs(lambda(1))...
    +(alpha(1)+(beta(1))/(lambda(1)))*X_p(:,1,i)*abs(lambda(1))+...
    ((alpha(1)+(beta(1))/(lambda(1)))*X(:,1,i)*sign(lambda(1))*lambda_p(1))...
    +(alpha_p(2)+(beta_p(2)*lambda(2)-beta(2)*lambda_p(2))/((lambda(2))^2))...
```

```
    *X(:,2,i)*abs(lambda(2))+(alpha(2)+(beta(2))/(lambda(2)))*X_p(:,2,i)...
    *abs(lambda(2))+((alpha(2)+(beta(2))/(lambda(2)))*X(:,2,i)*sign(lambda(2))
    lambda_p(2)));
    end
end
```

- Subroutine RFtra□cTL

```
function [x,rho_p,q_p,v_p] = RFtraffic_TL(rhoL,vL,rhoR,vR,imax,tmax,nmax,...
    c0,tau,c0_p,tau_p)
% traffic Riemann problem
u_p=zeros(2,imax);
Dx = 1/(imax-1);
%Dt = tmax/nmax;
Dt=0.000000025;
 x = 0:Dx:1;
%
 for i = 1:imax
    if (x(i)<=0.5)
         u(1,i) = rhoL;
         u(2,i) = rhoL*vL;
    else
        u(1,i) = rhoR;
        u(2,i) = rhoR*vR;
    end
 end
```

```
rho = u(1,:);
q = u(2,:);
v = u(2,:)./u(1,:);
rho_p = u_p(1,:);
q_p = u_p(2,:);

    for k=1:imax
        v_p(1,k) = (u_p(2,k)*u(1,k)-u_p(1,k)*u(2,k))/((u(1,k)^2));
    end
for n=1:nmax
   [h,s] = fluxes(u,Dt,Dx,c0,tau);
    [h_p,s,s_p] = fluxes_TL(u,u_p,Dt,Dx,c0_p,c0,tau,tau_p);

 for i = 2:imax-1
        u(:,i) = u(:,i)-(Dt/Dx)*(h(:,i)-h(:,i-1))+...
        (Dt/Dx)*(s(:,i)+s(:,i-1))/2;

        u_p(:,i) = u_p(:,i)-(Dt/Dx)*(h_p(:,i)-h_p(:,i-1))+(Dt/Dx)...
            *(s_p(:,i)+s_p(:,i-1))/2;

    end

    rho = u(1,:);
    q = u(2,:);
    v = u(2,:)./u(1,:);
```

```
    rho_p = u_p(1,:);
    q_p = u_p(2,:);
    for k=1:imax
        v_p(1,k) = (u_p(2,k)*u(1,k)-u_p(1,k)*u(2,k))/((u(1,k)^2));
    end
end
```

Appendix C

Calculation of the cost function and the gradient

```
function [f,g] = calcfg(X,nmax,imax,tmax,datrho,datv,D,freq,truerhoL,...
    truevL,truerhoR,truevR)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  List of main variables
%    X:          Guess of c_0 and tau
%    nmax:       Number of time steps
%    imax:       Number of space steps
%    tmax:       Time window
%    D:          Observation weighting matrix
%    h:          Time step for numerical scheme
%    freq:       Frequency of observations
%    tstep:      Number of time steps to perform
%    [x,rho,q]:    Forward trajectory
%    [datrho,datv]: Observation values -no obs of the flux
%    [truerhoL,truevL,truerhoR,truevR]: initial state of density and
%    velocity
```

```
%  Output:
%    [f,g]: Cost function and gradient
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f=0.0d0;
f1=0.0d0;
f2=0.0d0;
guessc0=X(1,1);
guesstau=X(2,1);
%set perturbation
pert=[1d-4,1d-4];
%parameters perturbed
guessc0_p=pert(1)+guessc0;
guesstau_p=pert(2)+guesstau;
%run of the non linear model for first guess parameters
[x,rho,q,v]=RFtraffic(truerhoL,truevL,truerhoR,truevR,imax,tmax,nmax,guessc0,.
    guesstau);
[x1,rho1,q1,v1]=RFtraffic(truerhoL,truevL,truerhoR,truevR,imax,tmax,nmax,...
    guessc0_p,guesstau);
[x2,rho2,q2,v2]=RFtraffic(truerhoL,truevL,truerhoR,truevR,imax,tmax,nmax,...
    guessc0,guesstau_p);

%
% Calculate the cost function
for n=1:freq:nmax
```

Bibliography

[1] Chin Jian Leo and Robert L. Pretty Numerical simulation of macroscopic continuum tra□c models, *Transportation Research Part B*, 26B, No3, 1992, 207-220.

[2] B.Piccoli and A.Tosin A review of continuum mathematical models of vehicular tra□c, *Encyclopedia of Complexity and Systems Science*, R. A. Meyers, Ed. Springer, 2009

[3] M.D'Elia *Modelli continui monodimensionali per il tra co automobilistico [Monodimensional continuum models for vehicular tra c]*, Dissertation, Politecnico di Milano, 2004-2005 (in Italian).

[4] W.L.Jin and H.M.Zhang Solving the Payne-Whitham tra□c flow model as a hyperbolic system of conservation laws with relaxation, 2001.

[5] H.M.Zhang A finite di□erence approximation of a non-equilibrium traffic flow model, *Transportation Research Part B* , 35, 2001, 337-365.

[6] Zhijin and I.M Navon Optimality of variational data assimilation and its relationship with the Kalman filter and smoother, *Quarterly Journal of the Royal Meteorological Society*, 127, 2001, 661-683.

[7] Yanqiu Zhu and 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, 1999, 1497-1517.

[8] Zhijin Li and I.M.Navon Performance of 4D-Var strategies using the FSU global spectral model with its full physics adjoiny, *Monthly Weather Review*, 1998.

[9] Laura Watkinson *Four dimensional variational data assimilation for Hamiltonian problems*, thesis for the degree of Doctor of Philosophy, University of Reading, 2006 .

[10] Zhuo Liu and I.M.navon Documentation of the tangent linear and adjoint models of the MPI version of the FSU global spectral model, *Monthly Weather Review*, 2002.

[11] Yong Li, I.M. Navon, Weiyu Yang, Xiaolei Zou, J.R. Bates, S. Moorthi and R.W. Higgins Four-dimensional variational data assimilation experiments with a multilevel semi-lagrangian semi-implicit general circulation model , *Monthly Weather Review*, 122, 1994, 966-983.

[12] Payne, H.J Models of freeway tra□c and control, *Beckey, G.A (Ed.), Mathematical models of public systems, simulation councils proc. ser.*, 1[12]

[14] A.S. Lawless and N.K. Nichols Inner-loop stopping criteria for incremental four dimensional variational data assimilation, *Monthly Weather Review*, 134, 2006, 3425-3435.

[15] J.Nocedal and S.J. Wright *Numerical optimization*, Springer, 2006.

[16] *http* : ==*www:nceo:ac:uk=lorenz_system:php*, Lorenz system.