Data Assimilation Using Optimal Control Theory

Anne K. Griffith and N.K. Nichols

Numerical Analysis Report 10/94

Department of Mathematics
P.O.Box 220
University of Reading
Whiteknights
Reading
RG6 2AX
United Kingdom

Abstract

This r port giv s a bri f introduction to data assimilation, and a summary of th calculus of variations and its application to optimal control th ory. It th n consid rs how data assimilation can b xpr ss d as an optimal control probl m.

An algorithm is d scrib d for th num rical solution of th optimal control probl m, which involv s using th mod l and its adjoint to find th gradi nt of th cost functional. This gradi nt is th n us d in a d sc nt algorithm to produc an improv d stimat of th control variabl .

Th algorithm is t st d for a simpl ODE and a simpl PDE mod l. For ach mod l diff r nt discr tisations ar consid r d, and th corr sponding discr t adjoint quations ar found dir ctly.

cknowledgements

The work described in this report was carried out under and EPSRC CASE studentship with the UK Meteorological Office. We would like to thank Andrew Lorenc of the Met. Office for his help and support with this work.

C ntents

1 Intr ducti n t Data Assimilati n

Data assimilation is a process for integrating observed data into a forecast model. The crudest such method would be direct substitution of the observed values to replace the predicted values they represent. However, if the value at an observation point is changed in this way, it no longer agrees with values at neighbouring grid points. Data assimilation schemes therefore aim to modify the model predictions so that they are consistent with the observations.

Data assimilation has been widely used in various forms in meteorological and oceanographic modelling since the 1950's. The various forms use ideas from different branches of mathematics; notably probability theory, optimization and control theory. It is interesting, however, that although the problem may be formulated using different disciplines of mathematics, the resulting schemes have many common features and properties. (See [6] for an overview of different data assimilation techniques, and an extensive list of references.)

The different approaches to data assimilation could be categorised in many different ways, but choosing just three categories, data assimilation techniques can be classed as simple correction schemes, statistical schemes and variational schemes.

Simple correction schemes

The simple correction schemes invlove weighting functions to add some proportion of a correction to grid points surrounding an observation, the "correction" being the difference between the observation and the corresponding model value. In the simplest cases, these weights depend on distance from the observations alone (see [5] and [3] for examples).

Statistical schemes

Statistical schemes, for example statistical interpolation or optimal interpolation (see [9]), use the error covariances of the observations and of the model predictions to find the "most likely" linear combination of the two. The Kalman filter provides perhaps the most sophisticated approach to this, but is very expensive to run and is not easily extended for use in nonlinear models.

Variational schemes

The idea behind variational data assimilation is to minimize some "cost functional" expressing the distance between observations and the corresponding model values using the model equations as constraints. The result is the model solution which fits "closest" to the observations, with the measure of closeness defined by the cost function (see [8], [11] and [12]).

In the case of data assimilation for a meterological forecast model for example, variational data assimilation would provide means for choosing initial conditions in such a way that the resulting "analysis" (model output) is as close as possible to the specified observed values, whilst satisfying the model equations. Variational schemes are based on optimal control theory.

Section 2 presents some results from the calculus of variations as background to optimal control theory. Section 3 introduces optimal control theory in the context of data assimilation, and describes an algorithm for the numerical solution of optimal con

2 Overview f the Calculus f Variati ns

The aim of this section is to give some background results in the calculus of variations which are used in optimal control theory. For a more thorough treatment of the subject, see any text book on optimal control or the calculus of variations, eg [1], [2], [7], and [10]. The theorems and definitions quoted below can also be found in these texts, although set out in a different way.

2.1 Cost Functionals

The "fundamental problem of the calculus of variations" is:

Find the function $y(t)$ in the set of admissible functions $\mathcal{A}$ which minimizes the cost functional

$$\mathcal{J} = \int_{t_0}^{t_1} F(t, y(t), y'(t))dt.$$

Definition 1:

The functional $\mathcal{J}(y)$ has an *extremal* at $\hat{y}$ if $\exists\varepsilon > 0$ such that $\mathcal{J}(y) - \mathcal{J}(\hat{y})$ has just one sign $\forall$ y such that $||y - \hat{y}|| < \varepsilon$.

Theorem 1:

A necessary condition for $\hat{y}\in\mathcal{A}$ to be an extremal is that $\delta\mathcal{J} = 0$ for all choices of δy and $\delta y'$.

2.4 Necessary Conditions for an Extremal

Any necessary conditions ensuring that $\delta\mathcal{J} = 0$ give necessary conditions for an extremal. However, to avoid evaluating the variation of (often complicated) cost functionals, the Euler Lagrange equations give the required necessary conditions for many problems in a neat form.

The Euler Lagrange Equations

The first variation of $\mathcal{J}$ is:

$$\delta\mathcal{J} = \int_{t_0}^{t_1} \frac{\partial F}{\partial y}\delta y + \frac{\partial F}{\partial y'}\delta y' dt. \tag{2.5}$$

Therefore

$$\delta\mathcal{J} = \int_{t_0}^{t_1} \frac{\partial F}{\partial y}\delta y dt + \left[\frac{\partial F}{\partial y'}\delta y\right]_{t_0}^{t_1} - \int_{t_0}^{t_1} \frac{d}{dt}\left(\frac{\partial F}{\partial y'}\right)\delta y dt, \tag{2.6}$$

and hence

$$\delta\mathcal{J} = \int_{t_0}^{t_1} \left(\frac{\partial F}{\partial y} - \frac{d}{dt}\left(\frac{\partial F}{\partial y'}\right)\right)\delta y dt + \left[\frac{\partial F}{\partial y'}\delta y\right]_{t_0}^{t_1}. \tag{2.7}$$

From this it can be seen that for $\delta\mathcal{J} = 0$ we require

$$\left[\frac{\partial F}{\partial y'}\delta y\right]_{t_0}^{t_1} = 0 \tag{2.8}$$

and

$$\frac{\partial F}{\partial y} - \frac{d}{dt}\left(\frac{\partial F}{\partial y'}\right) = 0, \tag{2.9}$$

which is known as the *Euler Larange Equation.*

Notice that if $\mathcal{A}$ restricts admissible functions y to those with fixed end points, $y(t_0) = y_0$, $y(t_1) = y_1$, then $\delta y(t_0) = 0 = \delta y(t_1)$, and so

$$\left[\frac{\partial F}{\partial y'}\delta y\right]_{t_0}^{t_1} = 0. \tag{2.10}$$

Otherwise, at a "free end", we must enforce

$$\frac{\partial F}{\partial y'} = 0. \tag{2.11}$$

Simplified forms of the Euler Lagrange equations can be derived in the case where $\mathcal{J}$ does not depend on t explicitly, or when $\mathcal{J}$ is independent of t and y.

The Vector Case

If the functional $\mathcal{J}$ is defined in terms of an N dimensional vector $\mathbf{y}(t)$ and its derivative $\mathbf{y}'(t)$, so that

$$\mathcal{J} = \int_{t_0}^{t_1} F(t, \mathbf{y}(t), \mathbf{y}'(t))dt, \tag{2.12}$$

then we have N Euler Lagrange equations

$$\frac{\partial F}{\partial y_n} - \frac{d}{dt}\left(\frac{\partial F}{\partial y_n'}\right) = 0, \quad n = 1, 2, \ldots N. \tag{2.13}$$

2.5 Constraints and the Method of Lagrange Multipliers

Suppose we wish to minimize the functional $\mathcal{J} = \int_{t_0}^{t_1} F(t, y, y')dt$ subject to the constraint $G(t, y, y') = 0$.

Theorem 2:

If $y(t) \in \mathcal{A}$ is twice continuously differentiable and an extremal of $\mathcal{J}$ over members of $\mathcal{A}$ satisfying $G(y) = 0$, then $\exists \lambda \in \Re$ such that y is an extremal of the functional

$$\mathcal{L} = \int_{t_0}^{t_1} F(t, y, y') + \lambda G(t, y, y')dt. \tag{2.14}$$

Notes:

1 If y minimizes $\mathcal{J}$, then we don't know that y minimize[illegible]

3 $\mathcal{L}$ is called the *augmented functional.*

In general, if we have N constraints, $G_1, ...G_N$, Theorem 2 holds with N Lagrange multipliers $\lambda_1, ...\lambda_N$ and the augmented functional becomes:

$$\mathcal{L} = \int_{t_0}^{t_1} F(t, y, y') + \sum_{n=1}^{N} \lambda_n G_n(t, y, y')dt. \tag{2.15}$$

Necessary Conditions for Extremals, Adjoint Equations

The same analysis as for the unconstrained case can now be applied to the augmented functional.

With the notation $H = F + \lambda G$, (H is sometimes called the "Hamiltonian"), we have the following necessary conditions for an extremal y of $\mathcal{L}$:

$$G(y) = 0, \tag{2.16}$$

$$\frac{\partial H}{\partial y} - \frac{d}{dt}\left(\frac{\partial H}{\partial y'}\right) = 0, \tag{2.17}$$

$$\left[\frac{\partial H}{\partial y'}\delta y\right]_{t_0}^{t_1} = 0. \tag{2.18}$$

The second condition, the Euler-Lagrange equation, is sometimes called an *adjoint equation*, and Lagrange multiplier λ is sometimes called an *adjoint variable.*

3.2 Numerical Solution of the Optimal Control Problem

We need a numerical method for choosing the control variable so that the resulting model state satisfies the necessary conditions for an extremal. Therefore, given a first guess for the control and the resulting model output, we need a way to c where K interpolates the model states to the observations and W is a weighting matrix, which might for example weight components of $\mathbf{y} - K(\mathbf{x})$ according to the error covariances of the observations and of the interpolation.

We wish to find the vector $\mathbf{x}$ that minimizes $\mathcal{J}$, so $\mathbf{x}$ acts as the control variable here. If $\mathbf{x}$ is also required to satisfy the forecast model, then the problem is one of constrained minimization with the forecast model as a constraint. In general, this constrained minimization needs to be carried out n

4 An ODE Example

This section develops a method for solving the data assimilation problem in the case of a simple ODE model. In Section 4.1 the example is presented, and an algorithm based on the concepts given in Section 3.2 is developed to solve the problem. In Sections 4.2 and 4.3 the problem is tackled from a slightly different angle. For two different discretisations of the same model, the optimal control problem is solved by finding the adjoint equations directly for the discrete model equations. The example is concluded in Section 4.4, with a discussion of the results.

4.1 The ODE model

We suppose that our model is

$$\dot{y}(t) = ay(t) \qquad t \in [t_0, t_1], \tag{4.1}$$

with

$$y(0) = \alpha, \tag{4.2}$$

and that we have a set of observations corresponding to $y(t)$ which can be represented by the continuous function $\tilde{y}(t)$. If we choose to represent the distance between y and $\tilde{y}$ using the L_2 norm, then with α as the control variable, the optimal control problem is:

Choose α to minimize

$$\mathcal{J} = \int_0^1 (y(t) - \tilde{y}(t))^2 dt \tag{4.3}$$

subject to

$$ay - \dot{y} = 0, \tag{4.4}$$

$$y(0) = \alpha. \tag{4.5}$$

The augmented functional is

$$\mathcal{L} = \int_0^1 (y(t) - \tilde{y}(t))^2 + \lambda(t)(ay(t) - \dot{y}(t))dt, \tag{4.6}$$

and taking the first variation gives

$$\delta\mathcal{L} = \int_0^1 (2(y(t) - \tilde{y}(t$$

or

$$\delta\mathcal{L} = \int_0^1 (2(y(t) - \tilde{y}(t)) + \lambda(t)a)\delta y dt - [\lambda(t)\delta y(t)]_0^1 + \int_0^1 \dot{\lambda}(t)\delta y(t))dt. \quad (4.8)$$

From this the adjoint equation is found to be

$$-\dot{\lambda} = a\lambda + 2(y - \tilde{y}), \quad (4.9)$$

with

$$\lambda(1) = 0 = \lambda(0), \quad (4.10)$$

so the control problem can be written:

Find $\alpha = y(0)$ so that

$$\cdot \quad t$$

and

$$-\dot{\lambda} = a\lambda + 2(y - \tilde{y}), \tag{4.18}$$

but this time with just

$$\lambda(1) = 0, \tag{4.19}$$

then we are left with

$$\delta\mathcal{L} = \lambda(0)\delta y(0). \tag{4.20}$$

From this it can be seen that the gradient of $\mathcal{L}$ with respect to the control $y(0)$ is $\lambda(0)$, and that we need this to be zero for an optimal control.

This gives a method for finding the optimal control $\alpha = y(0)$ numerically. we first discretise (4.1) and (4.9), letting $y_j \approx y(j\Delta x)$ and $\lambda_j \approx \lambda(j\Delta x)$, for $j = 0, 1, ..J$, where $J = \frac{1}{\Delta x}$, and then use the following algorithm.

Algorithm 1

1 Guess α.

2 From $y_0 = \alpha$ calculate $y_j, \quad j = 1, ..J$.

3 Using y_j and starting from $\lambda_J = 0$, calculate $\lambda_j = 0, \quad j = J-1, ..0$.

4 Use the gradient λ_0 in a descent algorithm to guess a new α, and repeat from 2 until $|\lambda_0|$ is small enough.

From the "optimal" α found, the required approximation to the optimal $y(t)$ can be determined for $t \in [t_0, t_1]$.

Comments:

1) λ_0 is only an approximation to $\lambda(0)$, the gradient of $\mathcal{L}$ with respect to the control.

2) The discretised version of the adjoint equation may not be the true adjoint of the discretised version of (4.1). This point inspires the work of Section 4.2.

4.2 Euler's method applied to a simple ODE

As mentioned in Section 4.1, the discretized version of the adjoint equations may no longer be the true adjoint for the discrete version of the original equation.

This section describes how optimal control theory may be applied directly to the discretised ODE and PDE equations. The discrete adjoint equations derived in this way can be compared with the continuous ones derived previously. Test cases for these methods to find the "optimal control" solving a data assimilation type of problem are then described.

Euler's scheme discretises (4.1) with (4.2) as follows:

$$y_{j+1} = (1 + a\Delta t)y_j, \tag{4.21}$$

$$y_0 = \alpha. \tag{4.22}$$

We suppose we have observations $\tilde{y}_j$ approximating (4.14),

$$\tilde{y}_j = j\Delta t \quad \text{for} \quad j = 0, 1, ..J -$$

$$\delta\mathcal{L} = \sum_{j=0}^{J-1}[2(y_j - \tilde{y}_j)\Delta t + \lambda_{j+1}(1 + a\Delta t)]\delta y_j) - \sum_{j=1}^{J}\lambda_j\delta y_j \tag{4.29}$$

$$= [2(y_0-\tilde{y}_0)\Delta t+\lambda_1(1+a\Delta t)]\delta y_0+\sum_{j=1}^{J-1}[2(y_j-\tilde{y}_j)\Delta t+\lambda_{j+1}(1+a\Delta t)-\lambda_j]\delta y_j-\lambda_J\delta y_J. \tag{4.30}$$

So if we enforce

$$y_{j+1} = (1 + a\Delta t)y_j \qquad \text{for} \quad j = 0, 1, ..J - 1, \tag{4.31}$$

and

$$\lambda_j = (1 + a\Delta t)\lambda_{j+1} + 2(y_j - \tilde{y}_j)\Delta t \qquad \text{for} \quad j = J - 1, ..1, \tag{4.32}$$

with

$$\lambda_J = 0, \tag{4.33}$$

then we are left with

$$\delta\mathcal{L} = (\lambda_1(1 + a\Delta t) + 2(y_0 - \tilde{y}_0)\Delta t)\delta y_0, \tag{4.34}$$

so the gradient of $\mathcal{L}$ with respect to the control $\alpha = y_0$ is

$$\lambda_1(1 + a\Delta t) + 2(y_0 - \tilde{y}_0)\Delta t =$$

The steplength s is taken to be 1 originally, but if α^{k+1} is not better than α^k, that is, if $\left|\lambda_0^{k+1}\right| > \left|\lambda_0^k\right|$, the previous iteration is repeated with the stepsize halved, and the method continued using the smaller stepsize. This is carried out until $\left|\alpha^{k+1} - \alpha^k\right|$ is small enough (in this example until $\left|\lambda_0^k\right|$ is small enough).

In this way, the largest corrections are made to α on the first iterations, and then finer corrections are made as the iteration converges to the optimal α.

Implementation

A FORTRAN program was written implement Algorithm 1 using the Euler's scheme and its adjoint, with the steepest descent algorithm. The iteration is continued until $|\lambda_0| < 10^{-3}$. Table 1 presents the results from this program with $a = -2$, showing the solutions found and the num

4.3 Fourth Order Runge-Kutta Method for the ODE problem

In this section, the method described in Section 4.2 is repeated using the fourth order Runge-Kutta discretisation of (4.1) with (4.2), which is

$$y_{j+1} = \left(\frac{ah}{6}(6 + ah(3 + ah(1 + \frac{ah}{4}) + 1 \right) y_j, \qquad j = 0, 1, ..J - 1, \tag{4.37}$$

with

$$y_0 = \alpha. \tag{4.38}$$

Employing the same method as for the Euler example, we find that the adjoint equations are:

$$\lambda_j = \left(\frac{ah}{6}(6 + ah(3 + ah(1 + \frac{ah}{4}) + 1 \right) \lambda_{j+1} + 2(y_j - \tilde{y}_j)\Delta t, \qquad j = J - 1, ..1 \tag{4.39}$$

with

$$\lambda_J = 0, \tag{4.40}$$

which is consistent with (4.9) as $\Delta t \to 0$. The gradient of $\mathcal{L}$ with respect to $\alpha = y_0$ is λ_0 as before.

In this case the results for the same problem, which has analytic solution $\alpha = 0.6051$, are given in table 2.

Table 2

Δt	first guess of y_0	number of iterations	final value of y_0
$\frac{1}{100}$	0.6	4	0.5916
	0.5	7	0.5890
	1.0	9	0.5920
	10	14	0.5915
$\frac{1}{1000}$	0.6	2	0.6018
	0.5	7	0.6018
	1.0	9	0.6054
	10	14	0.6050
$\frac{1}{10000}$	0.6	3	0.6037
	0.5	7	0.6003

can expect inaccuracies in y_0 of an unknown size. Since Tables 1 and 2 show that the rate of convergence of y_0 to $y(0)$ decreases for smaller vaues of Δt, it seems that the errors in the iteration scheme dominate over the errors of the discretisation when Δt is small. This suggestion is backed up by the fact that the forth order Runge Kutta scheme should give a much better approximation to the analytic solution than the Euler scheme, and yet the results for both schemes are of similar accuracy for the same value of Δt.

The results could be improved by continuing the iteration until $|\lambda_0|$ satisfied a stricter tolerence. A more efficient descent algorithm could also be used to reduce the number of iterations needed. This is important in the context of data assimilation, because each iteration of the descent algorithm involves an integration of the model and of its adjoint. Since forecast models are very large, this will involve a lot of work. Therefore, the overall efficiency of any data assimilation scheme of this type will depend heavily on the number of iterations needed.

5 A PDE Example

In this section the model used is the linear advection equation in one dimension. This section follows a similar development to Section 4. Section 5.1 presents the problem, conditions for its solution and an algorithm for the numerical solution of the problem. Sections 5.2 and 5.3 treat two different discretisations of the linear advection equation, and describe the implementation of the given algorithm in each case. Section 5.4 discusses the results.

5.1 The PDE model

Suppose now that our model is the linear advection equation

$$u_t + cu_x = 0, \tag{5.1}$$

with

$$u(x,0) = \alpha(x), \qquad \text{and} \quad u(0,t) = u(1,t), \tag{5.2}$$

where $u = u(x,t)$, with $x \in [0,1]$ and $t \in [0,1]$.

Suppose we have observations corresponding to $u(x,t)$ which can be represented by the c

$u_{t\alpha} \mu u \mu f \mu \mu j \mu \mu u j \mu$

$$\delta\mathcal{L} = \int_0^1 \int_0^1 (2(u-\tilde{u}) - \lambda_t - c\lambda_x)\delta u dx dt + \int_0^1 [\lambda\delta u]_{t=0}^1 dx + \int_0^1 [c\lambda\delta u]_{x=0}^1 dt. \quad (5.6)$$

Necessary conditions for $\delta\mathcal{L} = 0$ are:

$$\lambda_t + c\lambda_x = 2(u-\tilde{u}), \quad (5.7)$$

with

$$\lambda(x,1) = 0 \qquad \text{and} \quad \lambda(0,t) = \lambda(1,t). \quad (5.8)$$

The control problem is now:

Find $\alpha(x) = u(x,0)$ *so that*

$$\lambda_t + c\lambda_x = 2(u-\tilde{u}), \quad (5.9)$$

with

$$\lambda(x,1) = 0, \qquad \lambda(x,0) = 0, \qquad \textit{and} \quad \lambda(0,t) = \lambda(1,t), \quad (5.10)$$

and

$$u_t + cu_x = 2(u-\tilde{u}), \quad (5.11)$$

with

$$u(0,t) = u(1,t). \quad (5.12)$$

We need a numerical scheme to do this. Following the development in section (3.2), we take

$$\lambda_t + c\lambda_x = 2(u-\tilde{u}) \quad (5.13)$$

with just

$$\lambda(x,1) = 0 \qquad \text{and} \quad \lambda(0,t) = \lambda(1,t), \quad (5.14)$$

where u satisfies (5.1) and (5.2), so that we are left with

$$\delta\mathcal{L} = -\int_0^1 \lambda(x,0)\delta u(x,0) dx. \quad (5.15)$$

Since the control variable $u(x,0)$ is a continuous function for $x \in [0,1]$, the "relevant inner product" in (3.9) is the L_2 inner product for $x \in [0,1]$.

Hence, the gradient of $\mathcal{L}$ with respect to the control $\alpha(x) = u(x,0)$ is $-\lambda(x,0)$. After discretising the original equation and its adjoints:

$u_j^n \approx u(j\Delta x, n\Delta t)$ and $\lambda_j^n \approx \lambda(j\Delta x, n\Delta t)$ for $j = 0, 1, ..J$ and $n = 0, 1, ..N$
Algorithm 2 can be used to find the optimal control:

Algorithm 2

1 Guess α_j for each $j = 1, ..., J-1$.

2 From $u_j^0 = \alpha_j$ calculate $u_j^n, \quad j = 1, ..J, \quad n = 1, ..N$

3 Using the u_j^n and starting from $\lambda_j^N = 0$, calculate $\lambda_j^n, \quad j = 0, 1, ..J,$ $n = N-1, ..0$.

4 Use λ_j^0 to guess new α_j, and repeat from step 2 until $||\lambda^0|| = \sum_j \left|\lambda_j^0 \Delta x\right|$ is small enough.

As in Section 5.2, rather than finding the adjoint equation of the continuous model equation and then discretising the model and adjoint equations, the discrete adjoint equations are found directly from a discretisation of the model. Algorithm 2 is then applied to the following test problem to examine the performance of the data assimilation.

5.1.1 A test problem for this scheme

Suppose the "observations" are given by the analytic solution $v(x,t)$ to $v_t + cv_x = 0$ with $v(0,t) = v(1,t)$ and with one of the following sets of initial conditions:

1.

$$v(x,0) = \begin{cases} -0.5 & x < 0.25 \\ 0.5 & 0.25 < x < 0.5 \\ -0.5 & x > 0.5 \end{cases} \tag{5.16}$$

2.

$$v(x,0) = \begin{cases} 0 & x < 0.25 \\ \cos^2(\frac{(x-0.375)\pi}{} & \\ & \end{cases}$$

After similar manipulations to those in the Euler example, we find (after a lot of fiddly algebra) that if we take as adjoint equations

$$\lambda_j^N = 0 \qquad \text{for} \quad j = 0, 1, ..J-1, \tag{5.26}$$

$$\lambda_j^n = (1-\mu)\lambda_j^{n+1} + \mu\lambda_{j+1}^{n+1} - 2(u_j^n - \tilde{u}_j^n)\Delta x \Delta t, \quad \text{for} \quad j = 0, 1, ..J-1, \quad n = N-1, ..1, 0;$$

5.3 The Lax Wendroff Scheme for the Linear dvection Equation

The work of Section 5.2 is now repeated using the Lax Wendroff scheme.

For the linear advection equation $u_t + cu_x = 0$ with $u(0,t) = u(1,$

$$\text{for} \qquad j = 0, 1, ..J-1, \quad n = 0, 1, ..N-1,$$

and the gradient of $\mathcal{L}$ with respect to the j^{th} component of the control, $\alpha_j = u_j^0$ is $-\lambda_j^0$.

The same test problem was carried out here as in the upwind scheme example. The results from using different values of Δx and Δt in this program are shown in Figures 5a-5c, and Figure 6 illustrates the behaviour of the Lax Wendroff scheme in the absense of data assimilation, with the analytic solution given for the initial condition.

5.4 Discussion of the results

The Upwind Scheme

The dissipation typical of the upwind scheme for $\frac{\Delta t}{\Delta x} = \frac{1}{2}$ is clearly seen in the solution. The data assimilation proceedure produces a vector of initial conditions for the upwind scheme, and inevitably, no matter what these are, dissipation will occur.

In Figures 1a-1c, it can be seen that the "optimal" initial condition produced by the assimilation over-exaggerates the corners of the square wave, so that after the dissipation occurs, the numerical scheme at later times is not so bad. Figure 3 shows similar effects for the different set of initial conditions. As typical with the upwind scheme, there is less dissipation when Δt and Δx are decreased, keeping $\frac{\Delta t}{\Delta x} = \frac{1}{2}$. Figure 2 shows the usual performance of the upwind scheme if the analytic solution is used for the initial conditions when Δx and Δt are the same as in Figure 1a. Comparing Figures 1a and 2 shows that instead of a good approximation for t close to the initial time and a bad one for t close to the end time, as usual for the upwind scheme; the assimilation scheme produces a solution which is on average not too far from the observations. This is what we expect as the solution to the optimal control problem: a numerical solution with minimum distance from the observations over the whole time interval.

The number of iterations needed is large, and increases as Δt and Δx decrease. When the tolerance on $||\lambda_0||$ is 10^{-3}, then 58 iterations are needed in the case $\Delta t = \frac{1}{80}$ and $\Delta x = \frac{1}{40}$, and 320 were needed when $\Delta t = \frac{1}{320}$, and $\Delta x = \frac{1}{160}$.

The assimilation has not resolved very well the fine features of the small spike in the third set of data, as Figure 4 shows. This indicates that the stopping criterion for the assimilation is too weak, and a smaller tolerance should be used.

The Lax Wendroff scheme

The results of the Lax Wendroff scheme with the first set of initial data for different values of Δt and Δx is shown in Figures 5a-5c. Again, the data assimilation produces initial conditions which modify the undesirable effects of the numerical solution at later times. Without data assimilation, the Lax Wendroff scheme with $\frac{\Delta t}{\Delta x} = \frac{1}{2}$ produces spurious oscillations behind a shock, as Figure 6 shows. The spurious oscillations produced when data assimilation is included are smaller, and now occur ahead of the shock for the initial time and behind the shock at the end time. Comparing Figure 5c with Figure 6 indicates the difference between using the optimal value of y_0 found by the data assimilation and using the analytic solution. The number of iterations needed increases as Δt and Δx decrease, and is similar to the number of iterations needed with the upwind scheme.

For both schemes, the results given here illustrate how the data assimilation scheme can use observations to counter some effects of model error. In the upwind scheme, the model error takes the form of dissipation, and in the Lax Wendroff scheme the model error consists of the spurious oscilations produced behind a shock. Both of these undesirable effects were modified in the solution by the choice of control variable.

6 C nclusi ns and Suggesti ns f r Further W rk

List f Figures

Figure 1a: The upwind scheme with data assimilation;

using the first set of initial conditions, and $\Delta t = \frac{1}{80}$, $\Delta x = \frac{1}{40}$

Figure 1b: The upwind scheme with data assimilation;

using the first set of initial conditions, and $\Delta t = \frac{1}{160}$, $\Delta x = \frac{1}{80}$

Figure 1c: The upwind scheme with data assimilation;

Figure 2: The upwind scheme: usual performance ***without*** **data assimilation;**

using the first set of initial conditions, and $\Delta t = \frac{1}{80}, \quad \Delta x = \frac{1}{40}$

Figure 3: The up

Figure 4: The up

Figure 5a:

The Lax Wendroff scheme with data assimilation;

using the first set of initial conditions, and $\Delta t = \frac{1}{80}, \quad \Delta x = \frac{1}{40}$

Figure 5b:

The Lax Wendroff scheme with data assimilation;

using the first set of initial conditions, and $\Delta t = \frac{1}{160}$, $\Delta x = \frac{1}{80}$

Figure 5c:

The Lax Wendroff scheme with data assimilation;

using the first set of initial conditions, and $\Delta t = \frac{1}{320}, \quad \Delta x = \frac{1}{160}$

Figure 6: The Lax Wendroff scheme: usual performance *without* **data assimilation;**

using the first set of initial conditions, and $\Delta t = \frac{1}{320}, \quad \Delta x = \frac{1}{160}$

References

[1] S. Barnett. *Introduction to Mathematical Control Theory.* OUP, 1975.

[2] R.G. Bartle. *The Elements of Real Analysis.* Wiley, 1964.

[3] P