561

Data Assimilation in Weather Forecasting: A Case Study in PDE-Constrained Optimization

M. Fisher, J. Nocedal[1], Y. Trémolet and S. J. Wright[2]

Research Department

[1]Department of Electrical Engineering and Computer Science, Northwestern University, Evanston IL 60208-3118, USA.
[2]Department of Computer Sciences, 1210 W. Dayton Street, University of Wisconsin, Madison WI 53706, USA.

April 2008

Abstract

Variational data assimilation is used at major weather prediction centers to produce the initial conditions for 7- to 10-day weather forecasts. This technique requires the solution of a very large data-fitting problem in which the major element is a set of partial differential equations that models the evolution of the atmosphere over a time window for which observational data has been gathered. Real-time solution of this difficult computational problem requires sophisticated models of atmospheric physics and dynamics, effective use of supercomputers, and specialized algorithms for optimization and linear algebra. The optimization algorithm can be accelerated by using a spectral preconditioner based on the Lanczos method. This paper shows how practical demands of the application dictate the various algorithmic choices that are made in the nonlinear optimization solver, with particular reference to the system in operation at the European Centre for Medium-Range Weather Forecasts.

1 Introduction

One of the most challenging problem areas in numerical optimization is the solution of very large problems whose major constraints are partial differential equations (PDEs). Typically, the goal is to choose parameters, inputs, or initial values for the system governed by the PDEs so as to optimize the behavior of the system in some way. The optimization procedure requires repeated execution of the PDE simulation, a task that is by itself complex and computationally intensive.

Although PDE-constrained optimization problems arise in several forms, such as optimal control, optimal design, and parameter identification, they share a number of challenges from the perspective of numerical optimization. Often, the problems are too large or complex to allow off-the-shelf optimization software to be used, and optimization algorithms that can be adapted to the characteristics of the PDE and its solver must be implemented. Derivative information may not always be exact and theoretical guarantees of performance must be sacrificed for the sake of practical efficiency.

In this paper, we discuss an important PDE-constrained optimization problem known as *variational data assimilation*, which arises in atmospheric and oceanic sciences and other applications. Data assimilation has become an important tool in forecasting at major weather prediction centers, where it is used to produce initial values for regional and global weather forecasts several times a day. The PDE in the data assimilation problem describes the evolution of the atmosphere from a given initial state, while the unknown is the state of the atmosphere at a given time point in the recent past. The objective function measures the goodness of fit between the simulated state and actual observations of the atmospheric state at various time points. The optimal state obtained from the data assimilation problem is then used as the initial condition in the evolution equations that produce a forecast.

In this paper, we discuss the key issues in variational data assimilation from the perspective of numerical optimization, highlighting the algorithmic choices that can be made for various aspects of the problem. In Section 2 we describe the problem and its basic characteristics. Possible algorithmic approaches are discussed in Section 3; a Gauss-Newton approach based on a nonlinear elimination formulation is found to be well suited for this application. Section 4 provides a description of a multilevel implementation of the Gauss-Newton approach, including the use of a spectral preconditioner in solving the subproblem to obtain the steps. We summarize the method in Section 5 and briefly discuss related work in Section 6. In Section 7, we mention several planned extensions to the model, together with the algorithmic challenges they present.

2 Variational Data Assimilation

2.1 Meteorological Formulation

Major weather prediction centers, such as the European Centre for Medium-Range Weather Forecasts (ECMWF), Météo-France, and the U.S. National Centers for Environmental Prediction (NCEP), produce a medium-range global weather forecast several times a day. The quality of forecasts, gauged by the number of days for which the forecast is accurate by some measure, has improved steadily during the past 10 years Simmons and Hollingsworth (2002). Advances are due to improved models of the atmosphere, greater availability of atmospheric data, increased computational power, and the continued improvement of variational data assimilation algorithms.

If the state of the atmosphere (temperatures, pressures, humidities, wind speeds and directions) were known accurately at a certain timepoint in the recent past, a forecast could be obtained by integrating the atmospheric model equations forward, past the present time and into the future. In practice, however, the state of the atmosphere is known only incompletely through observations that are distributed nonuniformly in space and time and subject to error. Before producing a forecast, an optimization calculation must be performed to find the atmospheric state at the chosen timepoint that is most consistent with the observations that have been made during the past few hours. This calculation is the data assimilation problem.

The *assimilation window* is a time interval (typically 6 to 12 hours) that encompasses the observations to be assimilated. The objective function is of weighted least-squares type, constructed from the differences between the actual atmospheric observations and the values predicted by the model. It also includes a term that penalizes deviation from a prior "background" state, obtained from the previous forecast run. The technique is known more specifically as *four-dimensional* variational data assimilation (abbreviated as 4D-Var) because the model contains three spatial dimensions and one time dimension, with the observations being distributed nonuniformly in all four dimensions. Once the state at the start of the assimilation window has been determined by solving the data assimilation problem, one can integrate the atmospheric model equations forward in time (over an interval of 10-15 days) to produce a forecast. Because a forecast must be produced in a timely fashion in order to be useful, the entire assimilation/forecast procedure must be carried out in real time; typically, a few hours.

In giving further detail on the 4D-Var problem, we focus on the system that is currently in operational use at ECMWF. We denote by $\bar{x}$ the atmospheric state at the start of the assimilation window, which is the key unknown in the data assimilation problem. The state is specified by the values of atmospheric variables (such as velocity, humidity, pressure, and temperature) at each node of a three-dimensional grid covering the globe. Atmospheric flow is described by the Navier-Stokes equations (see for example Haltiner and Williams (1980)), which are evolved over the assimilation window to obtain coefficients describing state at later time points in the window. We are particularly interested in the states at N equally spaced time points in the assimilation window, denoted by $t_1, t_2, \ldots, t_N$ (with the initial time point denoted by t_0). We denote the state vectors at these time points by $x_1, x_2, \ldots, x_N$, and denote the (nonlinear) operator that integrates the model from t_{i-1} to t_i by $\mathscr{M}_i$, so that

$$x_i = \mathscr{M}_i(x_{i-1}), \quad i = 1, 2, \ldots, N; \qquad x_0 = \bar{x}. \tag{1}$$

The Navier-Stokes equations are solved using a spectral method, so that the actual components of the state vector x_i are the coefficients of the spectral representation of the various atmospheric quantities on the specified grid at time t_i.

We denote the vector of observations made at time point t_i by y_i. Observations are distributed across the whole assimilation window, and include measurements of temperature, pressure, humidity, windspeed, and radiances taken from satellites, buoys, planes, boats, and ground-based sensors. (In practice, observations are assigned to

the time point t_i nearest the time at which they are actually measured.) Since the observations are obtained in physical space, we require an *observation operator* to perform the mapping from the spectral space in which x_i resides to the observation space. This operator $\mathcal{H}_i$ also incorporates the interpolations that map state values at the physical grid to the spatial location of each observation. The observations are nonuniformly distributed around the globe, tending to be clustered on satellite paths and densely populated areas.

The objective function for the 4D-Var problem can now be specified as follows:

$$J(\bar{x}, x) = \frac{1}{2}(\bar{x} - x^b)^T B^{-1}(\bar{x} - x^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathcal{H}_i(x_i) - y_i)^T R_i^{-1}(\mathcal{H}_i(x_i) - y_i), \tag{2}$$

where x^b is the background state, B is the background error covariance matrix, R_i is the observation error covariance matrix at time t_i, and x is a vector formed by concatenating $x_1, x_2, \ldots, x_N$. The background state x^b is obtained by integrating the solution obtained in the previous assimilation-forecast cycle to the starting time t_0 of the current assimilation window. This term is essential both because it carries forward information from previous assimilation windows and also because the problem would be underdetermined without it. The number of observations in any given assimilation cycle is much smaller than the number of variables—though the nonuniformity noted above results in there being many *more* observations than grid points in some geographical areas.

The choice of the covariance matrix B greatly influences the assimilation process, as its properties determine the components of the solution that correspond to areas with few or no observations. By altering the weight of the background term relative to the observation terms, we can tune the degree to which the analysis will hew to the observations.

2.2 Characteristics of the Optimization Problem

The 4D-Var problem, as formulated in the ECMWF system, exhibits a number of notable features.

1. The number of variables is extremely large. Atmospheric values at all nodes of a physical grid covering the globe are maintained for each time point in the assimilation window. In the current ECMWF system, the grid points are separated by approximately 25 km in the two surface dimensions, with 91 (nonuniformly spaced) levels in the vertical direction, covering the atmosphere to an altitude of approximately 80km. In the spectral representation of this physical grid of values, the vector x_i at each time point has about 3×10^8 components. Additionally, there are $N = 24$ time points (equally spaced at intervals of 30 minutes over a 12-hour assimilation window), making a total of about 7×10^9 variables.

2. There is a strict time budget available for solving the problem. Forecasts must be produced on a 12-hour cycle, less than two hours of which is available for solving the data assimilation problem.

3. The objective function $J(\bar{x}, x)$ is nonconvex and the constraints (1) are nonlinear.

4. Evaluation of the constraints via forward integration of the model (1) is expensive and can be performed only a few times in each forecast cycle. For given values of $x_0, x_1, \ldots, x_N$, evaluation of the observation operators $\mathcal{H}_i$ is relatively less expensive.

5. An adjoint approach has been used to code the action of the derivatives of $\mathcal{M}_i$ and $\mathcal{H}_i$ on given vectors. However, these coded derivatives are not exact. Some terms are omitted due to the difficulty of writing their adjoints, and certain discontinuities are ignored. The calculations needed to operate with the Jacobian of $\mathcal{M}_i$ on arbitrary vectors requires two to three times as much computation as an evaluation of $\mathcal{M}_i$ alone.

6. Calculation of the second derivatives of the operators $\mathscr{M}_i$ and $\mathscr{H}_i$ is not practical.

Because of these features, the 4D-Var problem is well beyond the capabilities of general-purpose optimization algorithms and software. A practical optimization algorithm for 4D-Var must exploit the characteristics of the problem to estimate the solution using very few function evaluations. Fortunately, a good estimate of the solution is available from the previous forecast, a property that makes it possible for 4D-Var to find a reasonable approximate solution in few steps.

3 Optimization Approaches

3.1 Algorithms for Data Fitting with Equality Constraints

To discuss algorithms for solving the 4D-Var problem, we abstract the problem slightly and consider a data-fitting problem with equality constraints. We partition the set of variables as $(\bar{x}, x)$, where x has m components, and write the equality constraints as the algebraic system $c(\bar{x}, x) = 0$, where c is a vector function with the same number of components (m) as x. The general data-fitting problem is

$$\min_{(\bar{x},x)} \frac{1}{2} f(\bar{x},x)^T R^{-1} f(\bar{x},x) \quad \text{subject to } c(\bar{x},x) = 0,$$

or more compactly

$$\min_{(\bar{x},x)} \frac{1}{2} \|f(\bar{x},x)\|^2_{R^{-1}} \quad \text{subject to } c(\bar{x},x) = 0, \tag{3}$$

where f is a vector function with p components and R is a $p \times p$ symmetric positive definite matrix.

The 4D-Var problem is a special case of (3) in which the unknowns are the states $\bar{x} = x_0$ and $x = (x_1, x_2, \ldots, x_N)$, while

$$c(\bar{x},x) = \begin{bmatrix} x_1 - \mathscr{M}_1(\bar{x}) \\ x_2 - \mathscr{M}_2(x_1) \\ \vdots \\ x_N - \mathscr{M}_N(x_{N-1}) \end{bmatrix}, \quad f(\bar{x},x) = \begin{bmatrix} \bar{x} - x^b \\ y_0 - \mathscr{H}_0(\bar{x}) \\ y_1 - \mathscr{H}_1(x_1) \\ \vdots \\ y_N - \mathscr{H}_N(x_N) \end{bmatrix}, \tag{4}$$

and $R = \text{diag}(B, R_0, R_1, \ldots, R_N)$.

3.1.1 SAND Approach

The first approach we consider for solving (3) is sequential quadratic programing (SQP) in which we linearize the constraints about the current point and replace the objective by a quadratic model (see, for example (Nocedal and Wright, 2006, Chapter 18)). Specifically, if D_z denotes differentiation with respect to z, we solve the following subproblem to obtain the step $(\bar{d}, d)$:

$$\min_{(\bar{d},d)} \frac{1}{2} \begin{bmatrix} \bar{d} \\ d \end{bmatrix}^T W \begin{bmatrix} \bar{d} \\ d \end{bmatrix} + \begin{bmatrix} \bar{d} \\ d \end{bmatrix}^T \begin{bmatrix} D_{\bar{x}} f(\bar{x},x)^T \\ D_x f(\bar{x},x)^T \end{bmatrix} R^{-1} f(\bar{x},x) \tag{5a}$$

$$+ \frac{1}{2} f(\bar{x},x)^T R^{-1} f(\bar{x},x) \tag{5b}$$

$$\text{subject to} \quad D_{\bar{x}} c(\bar{x},x)\bar{d} + D_x c(\bar{x},x)d = -c(\bar{x},x), \tag{5c}$$

for some symmetric matrix W. This matrix could be defined as the Hessian of the Lagrangian for (3), that is,

$$
\left[\begin{array}{c} D_{\bar{x}}f(\bar{x},x)^T \\ D_xf(\bar{x},x)^T \end{array} \right] R^{-1} \left[\begin{array}{cc} D_{\bar{x}}f(\bar{x},x) & D_xf(\bar{x},x) \end{array} \right] \tag{6}
$$
$$
+ \sum_{j=1}^{p} [R^{-1}f(\bar{x},x)]_j \nabla^2 f_j(\bar{x},x) + \sum_{l=1}^{m} \lambda_l \nabla^2 c_l(\bar{x},x),
$$

for some estimate of the Lagrange multipliers $\lambda_1, \lambda_2, \ldots, \lambda_m$, where $\nabla^2 f_j$ denotes the Hessian of f_j with respect to $(\bar{x},x)$, and similarly for $\nabla^2 c_l$. In the 4D-Var problem it is not practical to calculate second derivatives, so it is necessary to use an approximation to (6).

One option for the W matrix in (5a) is a (limited-memory) quasi-Newton approximation of the Hessian (6). Although this approach has proved effective in various PDE-optimization applications Biegler et al. (2003), it is not appropriate for the 4D-Var problem, because the small number of iterations does not allow time for curvature information to be accumulated rapidly enough. Moreover, a quasi-Newton approximation to (6) ignores the "least-squares" structure of this Hessian, namely, that the first term in (6) involves only first-derivative information. A Gauss-Newton approach would be to approximate the Hessian by its first term, thus avoiding the need for Lagrange multiplier estimates or second derivatives. Under this scheme, the approximate Hessian in (5a) is

$$
W = \left[\begin{array}{c} D_{\bar{x}}f(\bar{x},x)^T \\ D_xf(\bar{x},x)^T \end{array} \right] R^{-1} \left[\begin{array}{cc} D_{\bar{x}}f(\bar{x},x) & D_xf(\bar{x},x) \end{array} \right], \tag{7}
$$

and with this definition we can write the quadratic program (5) as follows:

$$
\min_{(\bar{d},d)} \frac{1}{2} \left\| f(\bar{x},x) + D_{\bar{x}}f(\bar{x},x)\bar{d} + D_xf(\bar{x},x)d \right\|_{R^{-1}}^2 \tag{8a}
$$

$$
\text{subject to } D_{\bar{x}}c(\bar{x},x)\bar{d} + D_xc(\bar{x},x)d = -c(\bar{x},x). \tag{8b}
$$

The formulation (8) has been used extensively; see for example Bock and Plitt Bock and Plitt (1984) for applications in control.

This approach is a particular instance of "simultaneous analysis and design" (SAND); see e.g. Biegler and Waecther (2003). It is characterized by the fact that the optimization is performed with respect to all the variables $(\bar{x},x)$, including the state vectors at the time points inside the assimilation window. The SQP method based on (8) does not enforce the constraint $c(\bar{x},x) = 0$ at intermediate iterates; it requires only the linearizations of the constraints to be satisfied at every iteration.

3.1.2   NAND Approach

In the *nonlinear elimination* or *nested analysis and design (NAND)* approach Biegler and Waecther (2003), the constraint $c(\bar{x},x) = 0$ is used to eliminate x from the problem, thus reformulating it as an unconstrained nonlinear least squares problem in $\bar{x}$ alone. We emphasize the dependence of x on $\bar{x}$ by writing $x = x(\bar{x})$, and reformulate (3) as

$$
\min_{\bar{x}} \frac{1}{2} \| f(\bar{x}, x(\bar{x})) \|_{R^{-1}}^2 . \tag{9}
$$

Sufficient conditions for smoothness of x as a function of $\bar{x}$ include that the $m \times m$ Jacobian $D_xc(\bar{x},x)$ is uniformly nonsingular in a neighborhood of the set of feasible $(\bar{x},x)$ in the region of interest.

We can consider standard approaches for solving the unconstrained nonlinear least squares problem (9), excluding methods that require second derivatives. Alternatives include Gauss-Newton, Levenberg-Marquardt,

and specialized quasi-Newton schemes that, in some cases, exploit the fact that part of the Hessian of (9) can be obtained from first partial derivatives of $f(\bar{x}, x)$; see the discussion in (Nocedal and Wright, 2006, Chapter 10).)

The Gauss-Newton approach yields the following linear-least-squares subproblem:

$$\min_{\bar{d}} \frac{1}{2} \left\| f(\bar{x}, x) + \left[D_{\bar{x}} f(\bar{x}, x) + D_x f(\bar{x}, x) \frac{dx}{d\bar{x}} \right] \bar{d} \right\|_{R^{-1}}^2 . \tag{10}$$

By differentiating $c(\bar{x}, x) = 0$ with respect to $\bar{x}$, we obtain

$$D_{\bar{x}} c(\bar{x}, x) + D_x c(\bar{x}, x) \frac{dx}{d\bar{x}} = 0,$$

from which we deduce that

$$\frac{dx}{d\bar{x}} = - \left(D_x c(\bar{x}, x) \right)^{-1} D_{\bar{x}} c(\bar{x}, x). \tag{11}$$

Therefore, we can write the Gauss-Newton subproblem (10) as

$$\min_{\bar{d}} \frac{1}{2} \left\| f(\bar{x}, x) + \left[D_{\bar{x}} f(\bar{x}, x) - D_x f(\bar{x}, x) (D_x c(\bar{x}, x))^{-1} D_{\bar{x}} c(\bar{x}, x) \right] \bar{d} \right\|_{R^{-1}}^2 . \tag{12}$$

One of the distinguishing features of the NAND approach is that it generates strictly feasible iterates $(\bar{x}, x)$, while SAND attains feasibilty only in the limit. One can show, however, that the Gauss-Newton NAND method (12) is identical to a *feasible* variant of the Gauss-Newton SAND approach (8). This variant would solve (8) to obtain $\bar{d}$ and set $\bar{x}^+ \leftarrow \bar{x} + \bar{d}$, then choose x^+ to maintain feasibility; that is, $c(\bar{x}^+, x^+) = 0$. In this scheme, the d component of the solution of (8) is discarded. Moreover, if this feasibility adjustment is performed at every iteration, the right-hand side of (8b) is always zero. Eliminating d from (8), we obtain $d = (D_x c(\bar{x}, x))^{-1} D_{\bar{x}} c(\bar{x}, x)$; substituting in (8a) gives the subproblem (12).

Note that by defining the null space matrix Z for the constraints (8b) as follows:

$$Z = \begin{bmatrix} I \\ -(D_x c(\bar{x}, x))^{-1} D_{\bar{x}} c(\bar{x}, x) \end{bmatrix},$$

we can rewrite (12) as

$$\min_{\bar{d}} \frac{1}{2} \left\| f(\bar{x}, x) + [D_{\bar{x}} f(\bar{x}, x) \quad D_x f(\bar{x}, x)] Z \bar{d} \right\|_{R^{-1}}^2 . \tag{13}$$

Hence, these equations define a Gauss-Newton step in which the step is confined explicitly to the null space of the constraints.

In summary, nonlinear elimination of x followed by application of the Gauss-Newton method (the NAND approach (12)) is equivalent to a certain feasible variant of SAND with linear elimination of the step d in the subproblem (8b). This observation is not new (see for example Young et al. (2002)), but it is relevant to the discussion that follows.

3.2 SAND vs NAND

We now consider the question: How do we choose between the SAND and NAND approaches for the 4D-Var problem? There are three key issues: storage, stability, and computational cost.

For the SAND formulation (8), it is usually necesary to store all the variables $(\bar{x}, x)$ along with data objects that (implicitly) represent the Jacobians $D_{\bar{x}} c(\bar{x}, x)$ and $D_x c(\bar{x}, x)$. These storage requirements are too high at present

for operational use. Since the atmospheric state vector x_i at each time point contains on the order of 3×10^8 components, the complete state vector $(\bar{x}, x)$ contains about 7×10^9 components. Therefore from the point of view of memory requirements, the NAND approach is to be preferred. (See §7 for further discussion on this topic).

In terms of stability, the SAND approach has two potential advantages. First, it provides more flexibility and potentially more accuracy at the linear algebra level. The quadratic program (8) can be solved using either a full-space method applied to the KKT conditions of this problem or a null-space approach (see (Nocedal and Wright, 2006, Chapter 16)). In the latter case, null-space SQP methods aim to find a basis of the constraint Jacobian

$$[D_{\bar{x}} c(\bar{x}, x) \quad D_x c(\bar{x}, x)]$$

that promotes sparsity and is well conditioned. This choice of basis can change from one iteration to the next. In contrast, in the NAND approach (12) the choice of basis is fixed at $D_x c(\bar{x}, x)$. If this basis is ill-conditioned, the subproblem (12) can be difficult to solve, particularly if iterative linear algebra techniques are used. In this case, the optimization iteration can contain significant errors that can slow the iteration and can even prevent it from converging. In other words, by eliminating x prior to solving the minimization problem for $\bar{x}$, we are imposing an elimination ordering on the variables that can introduce unnecessary ill-conditioning in the step computation procedure.

The second stability-related advantage of the SAND approach is that it does not require us to compute the solution $x(\bar{x})$ of the model equations $c(\bar{x}, x) = 0$ at every iteration of the optimization algorithm; we need only to evaluate c. For the NAND formulation, the integration of the differential equations for a poor choice of $\bar{x}$ may lead to blow-up and failure. Indeed, the atmospheric dynamics contain "increasing modes," by which we mean that small perturbations in the initial state $\bar{x}$ can lead to larger changes in later states x_i. (Over the time scale of the 12-hour assimilation window, the instability is not particularly significant, though it could become an issue if a longer assimilation window were to be used.) Such numerical difficulties could be handled by retaining the states at a selection of timepoints in the assimilation window as explicit variables and eliminating the rest, yielding a hybrid between the SAND and NAND approaches.

Another important distinction between the SAND and NAND approaches arises from the cost of satisfying the constraints. In the NAND approach, for every trial value of $\bar{x}$ we must solve the nonlinear equations (to reasonable accuracy) $c(\bar{x}, x) = 0$ to obtain x, while the SAND requires only the evaluation of $c(\bar{x}, x)$ for a given vector $(\bar{x}, x)$. In some PDE applications, there is a large difference in cost between these two operations. In 4D-Var, however, the two operations cost essentially the same; evaluation of the operators $\mathcal{M}_i$, $i = 1, 2, \ldots, N$ is required in both cases. (A similar situation occurs in control problems, where $\bar{x}$ represents input/control variables and x represents the system state.) Nevertheless, the SAND approach has the possible advantage that the operators $\mathcal{M}_i$ can be applied in parallel, rather than sequentially as in the NAND approach.

Taking all these factors into account, we conclude that the most attractive approach for the the current ECMWF system is the Gauss-Newton method applied to the unconstrained formulation (12): the NAND approach. We now discuss a variety of features that are necessary to make this approach effective in the restrictive practical environment of operational weather forecasting.

4 A Multilevel Inexact Gauss-Newton Method

Using the following definitions for the Jacobians of the nonlinear operators $\mathcal{M}_i$ of (1) and $\mathcal{H}_i$ of (2):

$$M_i = \frac{d\mathcal{M}_i}{dx_{i-1}}, \ i = 1, 2, \ldots, N; \qquad H_i = \frac{d\mathcal{H}_i}{dx_i}, \ i = 0, 1, \ldots, N, \tag{14}$$

we can write the Gauss-Newton subproblem (12) for the 4D-Var problem (1), (2) as follows:

$$\min_{\bar{d}} q(\bar{d}) = \frac{1}{2}(\bar{x} + \bar{d} - x^b)^T B^{-1}(\bar{x} + \bar{d} - x^b) \tag{15}$$

$$+ \frac{1}{2}\sum_{i=0}^{N}(f_{i+1} - H_i M_i M_{i-1}\ldots M_1\bar{d})^T R_i^{-1}(f_{i+1} - H_i M_i M_{i-1}\ldots M_1\bar{d}),$$

where the f_i denote the subvectors of f given in (4), that is,

$$f_{i+1} = y_i - \mathscr{H}_i(x_i), \quad i = 0, 1, \ldots, N; \qquad x_0 = \bar{x}.$$

After an approximate solution $\bar{d}$ of (15) has been computed, we set $\bar{x} \leftarrow \bar{x} + \bar{d}$ and update the remaining states x via successive substitution into the model equations (1), which corresponds to forward integration through the assimilation window.

The (strictly convex) quadratic function $q(\bar{d})$ defined in (15) can be minimized using an iterative method, such as conjugate gradients (CG), thus avoiding the need to form and factor the Hessian of q. The CG method requires only that we compute products of the matrices M_i, H_i, and their transposes with given vectors Golub and Van Loan (1989). In fact, the ECMWF system includes codes that perform these operations, and also codes that perform multiplications with the Cholesky factor of B (see (16) below).

A straightforward application of the conjugate gradient (CG) method is, however, too expensive for operational use, as a large number of CG iterations is required for the unpreconditioned method, and the cost of each such iteration is approximately four times higher than the cost of evaluating the objective function J in (2). However, as we discuss in this section, we can make the CG approach practical by computing the Gauss-Newton step at a lower resolution and by introducing a preconditioner for CG. Specifically, the CG iteration is preconditioned by using a few eigenvalues and eigenvectors of the Hessian of a lower-resolution version of the quadratic model q. The spectral information is obtained by applying the Lanczos method instead of CG. (The two methods are equivalent in that they generate the same iterates in exact arithmetic Golub and Van Loan (1989), but the Lanczos method provides additional information about the eigenvalues and eigenvectors of the Hessian.) This preconditioner is constructed in the first Gauss-Newton iteration and carried over to subsequent iterations in a manner described in Subsection 4.2 below.

4.1   The Multilevel Approach

A breakthrough in 4D-Var was made by implementing the Gauss-Newton method in a multilevel setting Courtier et al. (1994). Essentially, the quadratic model q in (15) is replaced by a lower-resolution model $\hat{q}$ whose approximate solution $\hat{d}$ can be found approximately at much lower cost, then extended to a step $\bar{d}$ in the full space.

Multiresolution schemes require a mechanism for switching between levels, restricting and prolonging the vector $(\bar{x}, x) = (x_0, x_1, \ldots, x_N)$ as appropriate. We describe this process by means of a restriction operator S that transforms each higher-resolution state vector x_i into a lower-resolution state vector $\hat{x}_i$, that is,

$$\hat{x}_i = Sx_i.$$

The pseudo-inverse S^+ is a prolongation operator, which maps a low-resolution $\hat{x}_i$ into a high-resolution x_i. When the components of x_i are spectral coefficients, as in the current ECMWF 4D-Var system, the operator S simply deletes the high-frequency coefficients, while the pseudo-inverse inserts zero coefficients for the high frequencies.

We obtain the lower-resolution analog $\hat{q}$ of q in (15) by replacing H_i, M_i, and B^{-1} by reduced-resolution approximations. The Lanczos method is applied to compute an approximate minimizer $\hat{d}$ of $\hat{q}$, and the new (higher-resolution) iterate is defined as follows:

$$\bar{x}^+ = \bar{x} + S^+\hat{d}.$$

Additional savings are obtained by simplifying the physics in the definition of $\hat{q}$. The linear operators M_i are not the exact derivatives of $\mathcal{M}_i$ (even at reduced resolution), but rather approximations obtained by simplifying the model of some physical processes such as convection, solar radiation, and boundary-layer interactions. Moreover, importantly, the lower-resolution model uses a much longer timestep than the full-resolution model. Since the computational cost of the gradient calculation is dominated by the cost of integrating the linear model and its adjoint, the use of lower resolution and simplified physics in the subproblems considerably reduces the cost of solving them.

Note that the Gauss-Newton steps modify only the lower-frequency coefficients, which correspond to large-scale atmospheric effects. Smaller-scale processes are, to some extent, controlled by the larger scales. For example, the (large-scale) jet stream in the upper troposphere has a significant influence on the development and evolution of smaller scale weather systems. The smaller scales are not ignored in the forecast/assimilation process, however, as modifications to the larger scales produce changes at the smaller scales when the nonlinear model is integrated to obtain the forecast.

4.2 Preconditioning

Construction of an effective preconditioner for the Gauss-Newton subproblems (15) is not straightforward. The matrices involved are very large and unstructured, due to the nonuniformly distributed nature of the observation locations. Moreover, the matrices are not defined explicitly in terms of their elements, but rather in computer code as functions that evaluate matrix-vector products. These factors preclude the use of many common, matrix-based preconditioning techniques.

The ECMWF system uses two levels of preconditioning. The first level employs a linear change of variable

$$\bar{d} = L\tilde{d}, \quad \text{where} \quad B = LL^T; \tag{16}$$

that is, L is the Cholesky factor of the background covariance matrix B. This change of variable transforms the Hessian of (15) to

$$I + \sum_{i=0}^{N}(H_iM_iM_{i-1}\ldots M_1L)^T R_i^{-1}(H_iM_iM_{i-1}\ldots M_1L), \tag{17}$$

and thus ensures that all eigenvalues of the transformed Hessian are at least 1. Moreover, since the rank of the summation term in (17) is much smaller than the dimension of $\bar{d}$ (and $\tilde{d}$), there are many unit eigenvalues. Preconditioning using L reduces the condition number of the Hessian (17) considerably, but it remains in the range $10^3 - 10^4$. The leading 45 approximate eigenvalues of (17) for a typical case are shown in Figure 1.

A second level of preconditioning takes advantage of the fact that the spectrum of (17) has well-separated leading eigenvalues that can be approximated by means of the Lanczos algorithm. When the Lanczos method is used to minimize the quadratic model (15), then after k iterations, it generates a $k \times k$ tridiagonal matrix $T^{(k)}$ along with a rectangular matrix $Q^{(k)}$ with k orthogonal columns. These matrices have the property that the eigenvalues of $T^{(k)}$ converge to the leading eigenvalues of the Hessian, while the eigenvectors of $T^{(k)}$, when pre-multiplied by $Q^{(k)}$, converge to the eigenvectors of the transformed Hessian (17).
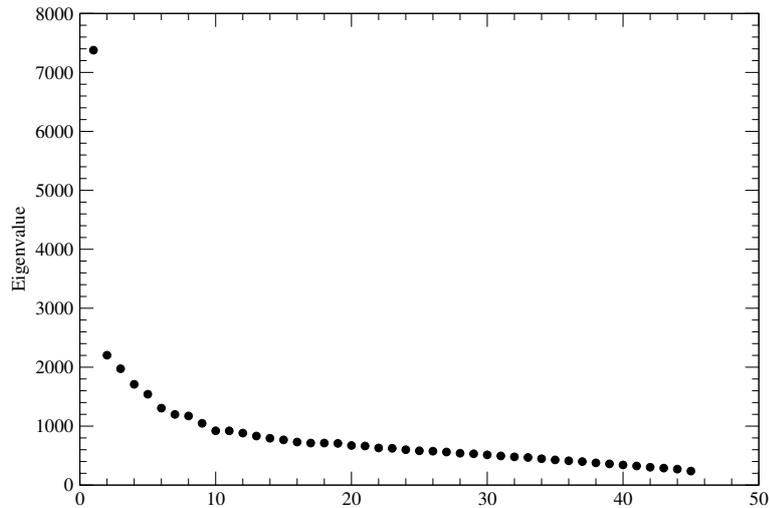
Figure 1: Leading approximate eigenvalues of ([17](#)), as determined by the Lanczos method.

The kth iterate $\chi^{(k)}$ of the Lanczos method satisfies

$$T^{(k)}z^{(k)} = -\left(Q^{(k)}\right)^T L^T \nabla q, \tag{18a}$$

$$\chi^{(k)} = Q^{(k)}z^{(k)}. \tag{18b}$$

The columns of $Q^{(k)}$ are normalized versions of the gradients produced by CG, where the normalization coefficient is a simple function of elements of $T^{(k)}$.

Let the estimates of the k leading eigenvalues obtained from $T^{(k)}$ be denoted by $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_k$, while the corresponding eigenvector estimates are denoted by $\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k$. The preconditioner for ([17](#)) is chosen to be

$$I + \sum_{j=1}^{m} (\mu_j - 1)\hat{v}_j\hat{v}_j^T, \tag{19}$$

for some integer $m \leq k$, where μ_j is typically defined as $\mu_j = \min(\hat{\lambda}_j, 10)$. If the first m eigenvalue and eigenvector estimates have converged reasonably well (which happens when the spectrum of the Hessian has well-separated leading eigenvalues, as in our case), the condition number is reduced from about λ_1 to λ_{m+1}. (Recall that the smallest eigenvalue of ([17](#)) is 1, so the condition number is simply the largest eigenvalue.)

The leading few eigenvectors and eigenvalues of ([17](#)) are calculated for the first Gauss-Newton subproblem at lowest resolution. The eigenvalue / eigenvector information obtained in the course of the first subproblem is used to precondition the subsequent Gauss-Newton subproblems, which are formulated at higher resolution. (See Morales and Nocedal (2000) for a related discussion.) Three Gauss-Newton steps are computed in the current operational system. This suffices because the availability of a good initial guess produces locally a problem that is only weakly nonlinear, at least for the larger scales that are important for predictability. Thus, the initial guess is in the domain of fast local convergence for the Gauss-Newton method, and we can expect a signficant improvement in accuracy after only 3 steps.

An additional advantage of using the Lanczos algorithm in place of CG is that the orthogonalization of the Lanczos vectors needed to prevent convergence of spurious copies of the leading eigenvectors also accelerates the minimization. The reduction in the number of iterations is in line with the argument by Notay (1993) who showed that for isolated eigenvalues at one or both ends of the spectrum, the number of extra iterations of CG

required to overcome the effects of rounding error is equal to number of duplicate eigenvectors that would be generated by an unorthogonalized Lanczos algorithm.

5   Summary of the Method

Here we summarize the algorithm whose major elements are described above.

Algorithm (One Cycle of 4D-Var: ECMWF System)

> Obtain x^b by integrating the previous forecast to time t_0;
> Define $\bar{x} = x_0 \leftarrow x^b$;
> **for** $k = 1, 2, \ldots, K$
>> Integrate the model (1) at high resolution to obtain $x_1, x_2, \ldots, x_N$
>>> and compute the cost function $J(\bar{x}, x)$ given by (2);
>> Choose S to define the restriction operator that maps high-
>>> resolution states to a lower resolution, with S^+ as the
>>> corresponding prolongation operator;
>> Compute the Jacobians M_i and H_i from (14) at lower resolution;
>> Using as preconditioner the background matrix (16), composed
>>> with the Lanczos preconditioner (19) obtained from the
>>> approximate eigenpairs gathered at previous iterations,
>>> perform l_k steps of preconditioned conjugate gradient on the
>>> quadratic objective (15) at the lower resolution to obtain the
>>> approximate lower-resolution solution $\hat{d}$;
>> Take step $\bar{x} \leftarrow \bar{x} + S^+ \hat{d}$;
> **end (for)**
> Starting from $x_0 = \bar{x}$, integrate the model (1) at highest resolution
>> to obtain the operational forecast.

No line search is performed at the major iterations (indexed by k), so there is no guarantee that J decreases. (A recent study Trémolet (2005) identifies those aspects of the simplification and resolution reduction that can lead to a failure of descent.) The current system uses $K = 3$ major iterations. For the first Gauss-Newton subproblem ($k = 1$), $l_1 = 70$ Lanczos iterations are performed, and $m = 25$ Lanczos vectors are saved to form the preconditioner (19) for the second and third iterations. (This choice of m was made empirically to optimize the tradeoff between the increased rate of convergence that results from using more eigenvectors and the computational cost of manipulating the vectors and applying the preconditioner.) During the second and third Gauss-Newton iterations, about 30 CG iterations are used.

Figure 2 shows the reduction in the norm of the gradient of the cost function for the second Gauss-Newton subproblem for a typical case. The dashed curve is for a minimization preconditioned using the background matrix B from (16) only. The solid curve is for a minimization that is also preconditioned using Lanczos eigenvectors calculated during the first minimization.

One could consider preconditioning the first outer iteration using eigenvalue and eigenvector information obtained from a previous forecast, but since the CG computations for the first Gauss-Newton problem are performed at low resolution, the savings would not be significant.

The 4D-Var system currently operational at ECMWF runs on an IBM cluster supercomputer using 1024 processors, organized as 128 MPI tasks with 8 Open-MP threads each. The overall run-time allowed for one 12-hour
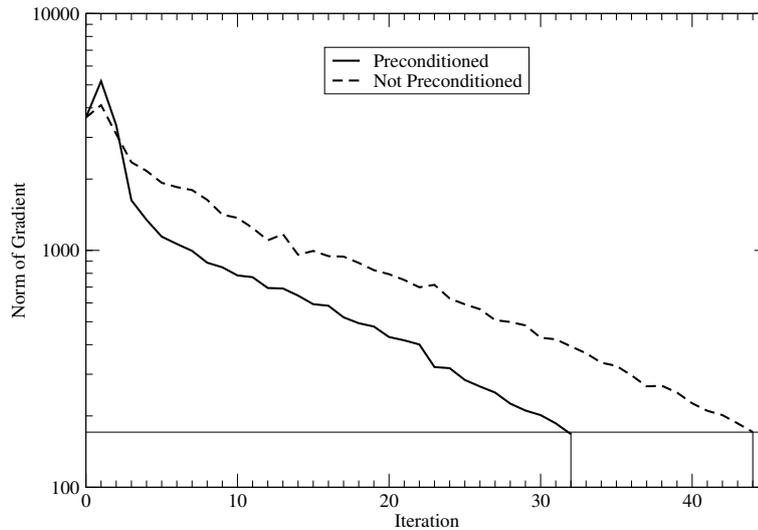
Figure 2: The effect of the eigenvector-based preconditioner on the convergence of the Gauss-Newton subproblem at the second major iteration. Vertical and horizontal lines show that the preconditioner reduces the number of iterations required to achieve a prescribed reduction in gradient norm by around 27%.

	Spectral Resolution	Physical Resolution	*J* evaluation (seconds)	Total (seconds)
Nonlinear	T799	25 km	350	2000
Minimization 1	T95	200 km	1.7	407
Minimization 2	T159	120 km	6.7	711
Minimization 3	T255	80 km	14.2	1381

Table 1: Average runtime per evaluation of the nonlinear ($\mathscr{H} \circ \mathscr{M}$) and linearized ($H \circ M$) operators in operational 4D-Var with 1024 processors.

4D-Var cycle is 1h15min. Since the full nonlinear operator ($\mathscr{H} \circ \mathscr{M}$) requires almost 6 minutes of run time, only a few evaluations of it can be performed. The reduced resolution of the successive minimizations make 4D-Var possible within the operational schedule; see Table 1. The computational cost of the 10-day forecast at resolution T799 is approximately the same as the overall cost of 4D-Var.

6   Related Work

The literature on PDE-constrained optimization is by now quite extensive and continues to grow rapidly, while the literature on preconditioning is vast. We mention here several papers that describe techniques related to the ones discussed in this work.

Young et al. Young et al. (2002) describe aerodynamic design applications, which like 4D-Var are problems in which the main constraint is a PDE that can be used to eliminate the state variables. They have found that the NAND approach is most effective, with various choices of approximate Hessian matrices for the subproblem. In 4D-Var, given $\bar{x}$, we can recover the eliminated variable x exactly (to within model and computational precision) at the cost of a single evaluation of the model (1); in Young et al. (2002), the equality constraint is different in nature and iterative techniques are required to approximately recover x from $\bar{x}$. Line searches are used to

improve global convergence properties.

The Lagrange-Newton-Krylov-Schur approach of Biros and Ghattas Biros and Ghattas (2005) can be viewed as a SAND approach in that it takes steps in the full space of variables $((\bar{x}, x)$, in our notation) and attempts to satisfy only a linearized version of the constraint $c(\bar{x}, x) = 0$ at each iteration. The focus in this work is on solution of the equality-constrained quadratic program at each iteration, which is posed as a linear system involving steps in the space of variables and Lagrange multipliers (the KKT system). Various choices are proposed for the Hessian approximation and structured preconditioners for the KKT system are constructed, preliminary to applying the Krylov method to solve this system.

Nabben and Vuik Nabben and Vuik (2004) discuss spectral preconditioners like (19) but with a general choice of the vectors $\hat{v}_j$. They obtain estimates for the spectrum of the preconditioned matrix and test their approach on a porous-flow application in which the vectors $\hat{v}_j$ are chosen to capture the smallest eigenvalue. Giraud and Gratton Giraud and Gratton (2006) analyze the eigenvalues of the preconditioned matrix as a function of the accuracy of the estimates μ_j and $\hat{v}_j$. For purposes of their analysis, these estimates are taken to be the *exact* eigenvalues and eigenvectors of a perturbed matrix, and bounds on deviation of the eigenvalues of the preconditioned matrix from their ideal values are given in terms of the size of the perturbation.

7 Final Remarks

The area of four-dimensional data assimilation is evolving continually, adapting to newer supercomputer platforms, better modeling methodology, new data sources, and increased demands on accuracy and timeliness. We conclude the paper by summarizing some of the challenges in the area and current directions of research.

With increasing resolution of weather forecasting models, smaller-scale phenomena will be resolved and more physical processes will be represented in the model. In particular, moist processes, which tend to be highly nonlinear, will be included in the models. The increased demand for environmental monitoring and forecasting should lead to additional atmospheric constituents to be estimated (such as CO_2) and the inclusion of additional chemical processes. Observation operators will also become more nonlinear as more humidity-related observations such as clouds and precipitation observations are included in the system. As a result of the increasing nonlinearity, additional outer-loop iterations (beyond the three iterations currently taken) will become necessary to produce a better approximate solution to the more nonlinear objective.

4D-Var as presented in Section 3 assumes that the numerical model representing the evolution of the atmospheric flow is perfect over the length of the assimilation window, or at least that model errors can be neglected when compared to other errors in the system, in particular, observation errors. This assumption can be alleviated by taking model error explicitly into account, imposing the model as a *weak constraint*. In addition to lifting a questionable assumption, an additional motivation for developing weak constraint 4D-Var is that when the assimilation window becomes long enough, it becomes equivalent to a full-rank Kalman smoother; see Fisher et al. (2005).

Several formulations are possible to implement weak-constraint 4D-Var; some are described by Trémolet (2006). The most general approach comprises keeping the four dimensional state variable as the control vector. The atmospheric model is then imposed as an additional constraint that relates the successive three dimensional states. A model error term in the cost function penalizes the difference

$$x_i - \mathcal{M}_i(x_{i-1}).$$

It is no longer possible to use the model constraints to eliminate all variables except $\bar{x}$ from the problem. We

rewrite the objective function (2) as a function of x as follows:

$$
\begin{aligned}
J(x) &= \frac{1}{2}(x_0 - x_b)^T B^{-1}(x_0 - x_b) + \frac{1}{2}\sum_{i=0}^{n}[\mathscr{H}(x_i) - y_i]^T R_i^{-1}[\mathscr{H}(x_i) - y_i] \\
&+ \frac{1}{2}\sum_{i=1}^{n}[x_i - \mathscr{M}_i(x_{i-1})]^T Q_i^{-1}[x_i - \mathscr{M}_i(x_{i-1})]
\end{aligned}
$$

where Q_i, $i = 1, 2, \ldots, N$ are the model error covariance matrices. The weak-constraint formulation is currently being developed at ECMWF. This would be the SAND approach. Because of the computational cost, and because of the limited amount of information available to define the model error covariance matrices and to solve the problem, the control variable would be defined by the state at some but not every timepoint in the assimilation window. The forecast model is used to define the state between the times when the control variable is defined. This would be the hybrid between the SAND and NAND approaches discussed in section 3.2.

References

L. Biegler and A. Waecther. DAE-constrained optimization. *SIAG/OPT Views-and-News*, 14(1):10–15, April 2003.

L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders. *Large-Scale PDE-Constrained Optimization*, volume 30. Springer Verlag, Heidelberg, Berlin, New York, 2003. Lecture Notes in Computational Science and Engineering.

G. Biros and O. Ghattas. Parallel lagrange-newton-krylov-schur methods for pde-constrained optimization. part i: The krylov-schur solver. *SIAM Journal on Scientific Computing*, 27(2):687–713, 2005.

H.-G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Conference*, pages 242–247, Budapest, 1984. Pergamon Press.

P. Courtier, J.-N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4D-Var, using an incremental approach. *Q. J. R. Meteorol. Soc.*, 120:1367–1387, 1994.

M. Fisher, M. Leutbecher, and G. Kelly. On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Q. J. R. Meteorol. Soc.*, 131:3235–3246, 2005.

L. Giraud and S. Gratton. On the sensitivity of some spectral preconditioners. *SIAM Journal of Matrix Analysis and Applications*, 27(4):1089–1105, 2006.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, second edition, 1989.

G. J. Haltiner and R.T. Williams. *Numerical prediction and dynamic meteorology*. John Wiley & Sons, 2nd edition, 1980.

J. L. Morales and J. Nocedal. Automatic preconditioning by limited memory quasi-newton updating. *SIAM Journal on Optimization*, 10(4):1079–1096, 2000.

R. Nabben and C. Vuik. A comparison of deflation and coarse grid correction applied to porous media flow. *SIAM Journal on Numerical Analysis*, 42(4):1631–1647, 2004.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, second edition, 2006.

Y. Notay. On the convergence rate of the conjugate gradients in presence of rounding errors. *Numerische Mathematik*, 65:301–317, 1993.

A. Simmons and A. Hollingsworth. Some aspects of the improvement in skill of numerical weather prediction. *Q. J. R. Meteorol. Soc.*, 128:647–677, 2002.

Y. Trémolet. Incremental 4D-Var convergence study. Technical Memorandum 469, European Center for Medium-Range Weather Forecasts, July 2005.

Y. Trémolet. Accounting for an imperfect model in 4D-Var. *Q. J. R. Meteorol. Soc.*, 132:In press, 2006.

D. P. Young, W. P. Huffman, R. G. Melvin, C. L. Hilmes, and F. T. Johnson. Nonlinear elimination in aerodynamic analysis and design optimization. In *Proceedings of the First Sandia Workshop on Large-scale PDE Constrained Optimization*, Lecture Notes in Computational Science and Engineering, Heidelberg, Berlin, New York, 2002. Springer Verlag.