Application of a New Adjoint Newton Algorithm to the 3-D ARPS Storm Scale Model Using Simulated Data

Zhi Wang, *Kelvin K. Droegemeier and **L. White

Center for Analysis and Prediction of Storms,

*School of Meteorology

and

**Department of Mathematics,

University of Oklahoma, Norman, OK 73019

E-mail: dwang@tornado.gcn.uoknor.edu

I. M. Navon

Department of Mathematics and Supercomputer

Computations Research Institute, Florida State University

Submitted to *Mon. Wea. Rev.*

3 Dec. 1996

Abstract

The adjoint Newton algorithm (ANA) is based on the first- and second-order adjoint techniques allowing one to obtain the "Newton line search direction" by integrating a "tangent linear model" backward in time (with negative time steps). Moreover, the ANA provides a new technique to find "Newton line search direction" without using gradient information. The error present in approximating the Hessian (the matrix of second order derivatives) of the cost function with respect to the control variables in the quasi-Newton type algorithm is thus completely eliminated, while the storage problem related to storing the Hessian no longer exists since the explicit Hessian is not required in this algorithm. The ANA is applied here, for the first time, in the framework of 4-D variational data assimilation to the adiabatic version of the Advanced Regional Prediction System (ARPS), a 3-dimensional, compressible, nonhydrostatic storm-scale model. The purpose is to assess the feasibility and efficiency of the ANA as a large scale minimization algorithm in the setting of 4-D variational data assimilation.

Numerical results using simulated observations indicate that the ANA can efficiently retrieve high quality model initial conditions. It improves upon the efficiency of the usual adjoint method employing the LBFGS algorithm by more than an order of magnitude in terms of both CPU time and number of iterations for our test problems. Numerical results also show that the ANA obtains a fast linear convergence rate.

1 Introduction

4-D variational data assimilation (Zupanski, 1993a, Thépaut *et al.*, 1993, Yang *et al.*, 1996, Zupanski and Mesinger, 1995 etc) is widely acknowledged as the most promising approach for implementing real-time analyzed data into an operational weather forecast system. In this approach, the numerical procedure attempts to generate both a close fit to the data and consistency with the dynamic model over a period of time. With present computer power, the only practical way to carry out 4-D variational data assimilation is through an appropriate use of the so-called adjoint of the assimilation model, which is used to calculate the gradient of the cost function with respect to control variables. The computational cost to operationally implement 4-D variational data assimilation is still very expensive due to the computer power, the required large storage size of meteorology problems and the inefficiency of large-scale unconstrained minimization algorithms. Therefore, it is very important to improve the existing algorithms or to develop new ones. While proper scaling and preconditioning (Zupanski, 1993 and 1996) may significantly improve the efficiency of a large-scale unconstrained minimization algorithm, they are not the focus points of this work. This paper studies the application of the adjoint Newton algorithm (ANA) to 4-D variational data assimilation problems using the Advanced Regional Prediction System (ARPS), a 3-dimensional, compressible, nonhydrostatic storm-scale model.

The second-order adjoint technique provides information that is related to Hessian (the matrix of second order derivatives) of the cost function with respect to the control variables. One integration of the second-order adjoint model yields a Hessian vector product (Wang *et al.*, 1992) which can be used to improve the efficiency of the truncated Newton algorithm (Wang *et al.*, 1993 and 1995). The application of the second-order adjoint technique may also be used to derive the Newton descent direction, which leads to an adjoint Newton algorithm (Wang *et al.*, 1996), which will be applied to the ARPS.

The adjoint Newton algorithm requires integrating a tangent linear model backward in time (Wang *et al.*, 1996). The reversing (integrating tangent linear model backward in time) is approximated by running the tangent linear model (TLM) of the nonlinear forecast model with a negative time step, but reversing the sign of friction and diffusion terms, in order to avoid the computational instability that would be associated with these terms if they were run backwards (Pu *et al.* 1996). Pu *et al.* (1996, 1996a and 1996b) also show that the backwards integration of a TLM can be used to improve the future forecast skill using past forecast error. They found that the forecast improvement obtained by the quasi-inverse linear method (integrating the TLM backwards in time) is considerably better than that obtained with a single adjoint integration, and similar to the one obtained using 5 iterations of the adjoint method.

The diffusion process and the equation that describes it are irreversible. But it is not true that the solution of the diffusion equation cannot be reversed in time. Just as a film showing the diffusion process may be reversed in time, the equivalent numerical trick may be found to produce the same effect (Smolarkiewicz, 1983). His technique (not used in this paper) may be used in the future.

Current popular methods for large-scale unconstrained minimization are

1. Limited memory conjugate gradient method (Shanno and Phua, 1980, Navon and Legler, 1987);

2. Quasi-Newton type algorithms (Davidon, 1959, Gill and Murray, 1972, O'Leary, 1983);

3. Limited-memory quasi-Newton methods such as the LBFGS algorithm (Nocedal, 1980, Liu and Nocedal, 1989) and

4. Truncated-Newton type algorithms (Nash, 1984a, b and 1985, Nash and Sofer, 1989, Schlick, 1992a and b, Navon *et al.*, 1992a and b, Wang, 1995).

All those methods find the descent direction using gradient of the cost function. The adjoint Newton algorithm provides a new approach to find the Newton descent direction by integrating a tangent linear model backward in time (with negative time steps). In Subsection 2.2, we will introduce the adjoint Newton algorithm, which is illustrated using a simple example in Appendix B. If the tangent linear model can be accurately integrated backward in time, the adjoint Newton algorithm has a quadratic convergence rate. For completeness, the theory of the adjoint Newton algorithm will be provided in Appendix A. In the process of the proof of the adjoint Newton algorithm, we derive a relation between the first and second order adjoint variables. The ARPS as well as its adjoint will be briefly described in Section 3. Also in Section 3, different issues of backward integration of tangent linear models will be addressed. Cost function, weights and scaling factors are addressed in Section 4. Numerical results for different experiments are shown in Section 5. There the ANA is compared with the usual adjoint method employing the LBFGS algorithm of Liu and Nocedal (1989) in terms of both CPU time and quality of the retrieved fields. Since the backward tangent linear model of the ARPS is not well posed and a modified version of it is used, the adjoint Newton algorithm achieves just a fast linear convergence rate (Appendix C) for our test problems. Summary, conclusions and limitations as well as topics for further research are presented in Section 6.

In the general framework of 4-D variational data assimilation, there are observation errors and/or model errors. Our first experiment (Section 5) uses simulated (model generated) observations without error while the second experiment uses simulated observations with random errors. Therefore the first experiment is not truly a 4-D variational data assimilation problem but is carried out in a setting closely related to 4-D variational data assimilation. Since this work represents the first step in the application of the adjoint Newton method using a 3-dimensional model, we proceed by assuming that the model is perfect.

2 Method

2.1 Definitions

Of interest is the large-scale unconstrained minimization of a functional assuming the following form

$$\min_U \ J(U) = \min_U \left\{ \frac{1}{2} \int_{t_0}^{t_f} \left(CX(t) - X^o(t)\right)^T W \left(CX(t) - X^o(t)\right) dt \right\}. \tag{1}$$

Here $X : t \longmapsto X(t)$ is the state variable that is a function from $[t_0, t_f]$ into n-dimensional Euclidean space R^n. The operator C: $R^n \longmapsto R^m$ represents a projection from the space (R^n) of the model solution X to the m-dimensional space (R^m) of observations. The linearity in C is assumed for simplicity but is not required. The superscript T indicates a transpose. The components of X are values of the various model fields (wind, temperature, pressure etc) at each of the model's grid locations. The number of components of X is denoted by n, and m represents the number of observations at any given time. In general, n is not equal to m. $[t_0, t_f]$ denotes the assimilation window where t_0 and t_f are the initial and final time and W is a positive definite and symmetric weighting function, which will be defined later. The cost function $J(U)$ is the weighted sum of squares of distance between the model solution and available observations distributed in space and time. The observation variable $X^o : t \longmapsto X^o(t)$ is a function from $[t_0, t_f]$ into R^m. The control variable $U \in R^n$ and the state vector X satisfy semi-discrete model equations, i.e. a set of ordinary differential equations (usually nonlinear)

$$\frac{dX}{dt} = F(X), \tag{2}$$

$$X(t_0) = U, \tag{3}$$

where t is the time, and F is model vector function of X. For simplicity, F is assumed to be at least twice differentiable. The control to state mapping (from R^n to R^n) is given by $U \longmapsto X(t, U)$.

It is important to realize that the control variables belong to a subset of the control space. To determine the admissible set of control one may use physical information about the control variable. *For simplicity, only initial conditions are taken as control variables in this paper.* However, our method holds when the control variables are initial plus boundary conditions. It is not clear whether parameters (Wang, 1993) other than initial and boundary conditions can be a part of the control variables vector for the ANA. This topic is the subject of further investigations.

Let $X' = [DX(U)]\,U'$ where $U' = X'(t_0) \in R^n$ be a perturbation on the initial condition U in Eq. (3) and $DX(U) : R^n \longmapsto R^n$ be the derivative of $X(U)$ with respect to U. Then the tangent linear model is defined as

$$\frac{dX'}{dt} = DF(X)X', \tag{4}$$

$$X'(t_0) = U'. \tag{5}$$

The first- and second-order adjoint models (Le Dimet and Talagrand, 1986, Wang *et al.*

1992 and 1995) may be defined, respectively, as

$$-\frac{dP}{dt} = (DF(X))^T P + C^T W (CX - X^o), \tag{6}$$

$$P(t_f) = 0, \tag{7}$$

$$-\quad \frac{d\hat{P}}{dt} = (DF(X))^T \hat{P} + \left(D^2F(X)X'\right)^T P + C^T W C X', \tag{8}$$

$$\hat{P}(t_f) = 0. \tag{9}$$

Here[1] P and $\hat{P} \in R^n$ are the first- and second- order adjoint variables. $D^2F(X) : R^n \times R^n \longmapsto R^n$ denotes the second derivative of $F(X)$ with respect to X.

It can be proven (Le Dimet and Talagrand, 1986, Wang *et al.*, 1992 and 1995) that

$$P(t_0) = DJ(U), \tag{10}$$

$$\hat{P}(t_0) = D^2J(U)U'. \tag{11}$$

2.2 Adjoint Newton algorithm

The current algorithms used to solve problem (1) first find its gradient by integrating the adjoint Eqs. (6-7) and then use the gradient to find a descent direction. There are other alternatives to calculate the gradient such as finite differencing approach (although not practical at all for meteorology problems). Through the study of the first and second order adjoint techniques, we found that the solution of a backward integration of a tangent linear model with a "suitable final condition" yields "a descent direction", which is comparable to the Newton descent direction. We will call this direction the estimated Newton descent direction. This approach is schematically presented in Fig. 1. Now we introduce the adjoint Newton algorithm here.

In the usual Newton algorithm (Berger, 1977, Stoer and Bulirsch, 1976), the function $J(U)$ being minimized is *approximated* locally by a quadratic function, and this *approximated*

[1] The definition of adjoint variable P differs from the definition in some references by a sign. We think that this notation is more consistent with our presentation.

function is minimized exactly. Thus near U we can approximate $J(U)$ by the *truncated* Taylor series (Luenberger, 1984)

$$J(U+d) \approx J(U) + DJ(U)d + \frac{1}{2}d^T D^2 J(U)d, \tag{12}$$

where $d \in R^n$ is a descent direction. The minimum of the right hand side of Eq. (12) will be achieved if d is the stationary point of the right hand side of Eq. (12), i.e. if d satisfies Newton's equations

$$D^2 J(U)d = -DJ(U). \tag{13}$$

Setting $U'_N = -d$, then Eq. (13) may be written as

$$D^2 J(U)U'_N = DJ(U). \tag{14}$$

Our underlying assumption is that the Hessian matrix $D^2 J(U)$ is positive definite and Eq. (14) is uniquely solvable. That is, the hypotheses for the convergence of Newton's method hold (Berger, 1977, Stoer and Bulirsch 1976). The adjoint Newton algorithm (ANA) amounts to a reformulation of Eq. (14) so that the estimated Newton descent direction is obtained using techniques related to the adjoint and there is no need to explicitly calculate or update the Hessian.

According to Eqs. (10) and (11), Eq. (14) implies there *exists a unique initial condition* U'_N such that

$$\hat{P}(t_0) = P(t_0), \tag{15}$$

where recall $\hat{P}(t_0) = D^2 J(U)U'_N$. We seek an approximation U'_e to U'_N. It turns out that by integrating a tangent linear model backward from a "suitable final condition", one can obtain just such an initial condition U'_e. Indeed, if the tangent linear model can be integrated backwards in time[2], then (proof is given in Appendix A) the estimated Newton descent direction is given by $d_e = -Y'(t_0) = -U'_e$ where $Y'(t_0)$ is the solution of the following

[2]The backward integration of a tangent linear model will be addressed in Subsection 3.3.

backward problem:

$$\frac{dY'}{dt} = DF(X)Y', \tag{16}$$

$$Y'(t_f) = C^{-1}\left(CX(t_f) - X^o(t_f)\right). \tag{17}$$

If U_m is the minimum and

$$U_m = U - U', \tag{18}$$

where U is the current estimate, then the following hold

$$\|U' - U'_N\| = O(\|U'\|^2), \tag{19}$$

$$\|U'_e - U'_N\| = O(\|U'\|^2), \tag{20}$$

$$\|U'_e - U'\| = O(\|U'\|^2). \tag{21}$$

Here C is assumed to be invertible and $\|U\|$ denotes the Euclidean norm of U.

Remarks:

1. From Eq. (18), we know that $-U'$ is the step to the minimum of $J(U)$ while Newton's direction $-U'_N$ is the step to the minimum of the right hand side of Eq. (12).

2. Since U'_e is as good as U'_N in the sense of Eqs. (19-21), we will also call $-U'_e$ the Newton descent direction.

3. Since we only reformulate the Newton's equation, Newton method's theoretical convergence rate should remain the same. Its convergence proof can be found in Luenberger (1984). A numerical convergence analysis will be provided in Appendix C using the ARPS as an example.

4. For simplicity it is assumed that the tangent linear model may be integrated backwards in time. If the nonlinear model is a mixture of hyperbolic and parabolic types, its corresponding tangent linear model can not be directly integrated backward in time. Modifications have to be carried out such that the tangent linear model can be integrated backward in time. This issue will be addressed in Subsection 3.3.

5. The final condition (17) denotes the "forecast error" at t_f. The solution of the backward problem given by Eqs. (16) and (17) at $t < t_f$ is an approximation to the "forecast error" at t. That is if the TLM can be accurately integrated backward in time and if there is no observation error, $Y'(t) \approx C^{-1}(CX(t) - X^o(t))$. Hence under these conditions, if the observations at $t < t_f$ are available, one may let $Y'(t) = C^{-1}(CX(t) - X^o(t))$ and then continue to integrate Eq. (16) to the initial time. Clearly, if the observations at t_0 are available, setting $d_e = -Y'(t_0) = -C^{-1}(CX(t_0) - X^o(t_0))$ the ANA will find the minimum in one step since $d_e = C^{-1}X^o(t_0) - X(t_0)$ is the step to the minimum, i.e. starting from an initial guess $U = X(t_0)$, one obtains $X(t_0) + d_e = C^{-1}X^o(t_0)$, which is the minimum.

6. The computational cost and storage for integrating the tangent linear model backward in time, is similar to that required for integrating the first order adjoint model.

7. The second order adjoint model is used to derive the ANA but it is not used in the ANA. In the current version of the ANA, the gradient is only used to check the convergence criteria and line search conditions (see (79) and (77-78) in Appendix A). Convergence criteria and line search conditions without using the gradient information (such as golden section search etc (Luenberger, 1984)) are under investigation. Namely, the next version of the ANA will not use the gradient of the cost function nor the adjoint model.

A simple example is provided in Appendix B to illustrate the process of implementing the ANA.

3 Model descriptions

3.1 Nonlinear model

The Advanced Regional Prediction System (ARPS) is a 3-dimensional, compressible, nonhydrostatic model developed for storm-scale prediction (Droegemeier *et al.*, 1995, Xue *et al.*, 1995). It is formulated in a curvilinear coordinate system that is orthogonal in the horizontal. The curvilinear coordinates can be defined numerically as well as analytically, making it more flexible than conventional terrain-following coordinates. The governing equations are the result of a direct transformation from the Cartesian system, and are expressed in a fully conservative form.

The current adiabatic version of the ARPS includes the Coriolis force, artificial divergence damping, total buoyancy and subgrid turbulence mixing. The governing equations are discretized on an Arakawa C-grid. Since the model atmosphere described by the governing equations is compressible, the meteorologicaly unimportant acoustic waves must be handled efficiently to avoid unnecessary restriction in the time step. The ARPS achieves this goal through the use of a splitting time integration technique reported by Klemp and Wilhelmson (1978). This technique divides a leapfrog integration time step into a number of small time steps and updates the acoustically active term every small time step while computing all the other terms only every leapfrog (big) time step. These acoustically active terms are the perturbation pressure gradient terms in the momentum equations and the divergence term in the pressure equation.

In the ARPS, a base state can be defined as either horizontally homogeneous or inhomogeneous. The thermal energy and pressure equations are written as prognostic equations for potential temperature and pressure. The adiabatic version of the ARPS with periodic boundary conditions is used in this study. Its governing equations are

$$\begin{aligned}\frac{\partial(\rho^* u)}{\partial t} &= -\left[(\rho^* u)\frac{\partial u}{\partial \xi} + (\rho^* v)\frac{\partial u}{\partial \eta} + (\rho^* W^c)\frac{\partial u}{\partial \zeta}\right] - \left[\frac{\partial(\bar{p}J_3)}{\partial \xi} + \frac{\partial(\bar{p}J_1)}{\partial \zeta}\right] \\ &- \left[\frac{\partial(J_3(p' - \alpha Div^*))}{\partial \xi} + \frac{\partial(J_1(p' - \alpha Div^*))}{\partial \zeta}\right] + \rho^* f v - \rho^* \tilde{f} w + G^{\frac{1}{2}} D_u \quad (22)\end{aligned}$$

$$\begin{aligned}\frac{\partial(\rho^* v)}{\partial t} &= -\left[(\rho^* u)\frac{\partial v}{\partial \xi} + (\rho^* v)\frac{\partial v}{\partial \eta} + (\rho^* W^c)\frac{\partial v}{\partial \zeta}\right] - \left[\frac{\partial(\bar{p}J_3)}{\partial \eta} + \frac{\partial(\bar{p}J_2)}{\partial \zeta}\right] \\ &- \left[\frac{\partial(J_3(p' - \alpha Div^*))}{\partial \eta} + \frac{\partial(J_2(p' - \alpha Div^*))}{\partial \zeta}\right] - \rho^* f u + G^{\frac{1}{2}} D_v \quad (23)\end{aligned}$$

$$
\begin{aligned}
\frac{\partial(\rho^* w)}{\partial t} &= -\left[(\rho^* u)\frac{\partial w}{\partial \xi} + (\rho^* v)\frac{\partial w}{\partial \eta} + (\rho^* W^c)\frac{\partial w}{\partial \zeta}\right] \\
&- \frac{\partial(p' - \alpha Div^*))}{\partial \zeta} + \rho^* B + \rho^* \tilde{f} u + G^{\frac{1}{2}} D_w && (24) \\
\frac{\partial(\rho^* \theta)}{\partial t} &= -\left[(\rho^* u)\frac{\partial \theta}{\partial \xi} + (\rho^* v)\frac{\partial \theta}{\partial \eta} + (\rho^* W^c)\frac{\partial \theta}{\partial \zeta}\right] + G^{\frac{1}{2}} D_\theta && (25) \\
\frac{\partial(G^{\frac{1}{2}} p')}{\partial t} &= -\left[(G^{\frac{1}{2}} u)\frac{\partial p'}{\partial \xi} + (G^{\frac{1}{2}} v)\frac{\partial p'}{\partial \eta} + (G^{\frac{1}{2}} W^c)\frac{\partial p'}{\partial \zeta}\right] \\
&- \left[(G^{\frac{1}{2}} u)\frac{\partial \bar{p}}{\partial \xi} + (G^{\frac{1}{2}} v)\frac{\partial \bar{p}}{\partial \eta}\right] + G^{\frac{1}{2}} W^c \bar{\rho} g \\
&- -\bar{\rho}\bar{c}^2 \left[\frac{\partial(G^{\frac{1}{2}} u)}{\partial \xi} + \frac{\partial(G^{\frac{1}{2}} v)}{\partial \eta} + \frac{\partial(G^{\frac{1}{2}} W^c)}{\partial \zeta}\right] && (26)
\end{aligned}
$$

where

- Variables with an overbar denote base state quantities that are functions only of hight,
- u, v, w and W^c are two components of horizontal velocity, vertical velocity and vertical contravariant velocity, respectively,
- $G^{\frac{1}{2}}$ (Droegemeier *et al.*, 1995, Xue *et al.*, 1995) is the determinant of the Jacobian matrix of the transformation from (ξ, η, ζ) system to (x, y, z) system. J_1, J_2 and J_3 are the non constant Jacobian of the transformation from (x, y, z) to (ξ, η, ζ),
- $\rho^* = G^{\frac{1}{2}} * \bar{\rho}$, and $\bar{\rho}$ is the base state density,
- p' and $\bar{p}$ denote the perturbation and base state pressure, respectively,
- αDiv^* is an artificial divergence damping term designed to attenuate acoustic waves,
- $f = 2\Omega \sin(\phi)$ and $\tilde{f} = 2\Omega \cos(\phi)$, where Ω is the angular velocity of the earth, and ϕ is latitude,
- D_u, D_v,D_w and D_θ are subgrid scale turbulence mixing,
- B is the total buoyancy,
- θ is potential temperature,

- $\bar{c}$ is the sound speed.

The control variables of the cost function for our numerical tests are the perturbation variables at the initial time, defined as

$$
\begin{aligned}
u'(x,y,z,t_0) &= u(x,y,z,t_0) - \bar{u}(z) \\
v'(x,y,z,t_0) &= v(x,y,z,t_0) - \bar{v}(z) \\
w'(x,y,z,t_0) &= w(x,y,z,t_0) \\
\theta'(x,y,z,t_0) &= \theta(x,y,z,t_0) - \bar{\theta}(z) \\
p'(x,y,z,t_0) &= p(x,y,z,t_0) - \bar{p}(z)
\end{aligned}
\tag{27}
$$

For simplicity, we drop the prime in subsequent discussion.

3.2 Model initialization

In this paper, the base state used follows that described by Weisman and Klemp (1982) and is horizontally homogeneous, hydrostatic and time invariant. The computational domain extends 10 km in both the east-west and north-south directions, and 5 km in the vertical, with horizontal and vertical grid spacings of 1 km and 0.5 km, respectively. Convection is initiated with five buoyant thermal disturbances ("bubbles") placed in the boundary layer. The maximum amplitude of these "bubbles" is 3 K. The centroid locations of the "bubbles" are (x_c, y_c, z_c)= (5km, 5km, 1.5km), (3km, 3km, 1.5km), (7km, 3km, 1.5km), (7km, 7km, 1.5km) and (2km, 7km, 1,5km). Above the ground, horizontal and vertical radius are (x_r, y_r, z_r)=(3km, 3km, 1.5km). The "bubbles" overlaps. Starting with this initial condition, we integrate the ARPS for 15 minutes and use the output at this time as the initial conditions to obtain the model-simulated "observations" at the end of assimilation window. The initial fields at $z = 3.5km$ of v and p are shown in Fig. 2. All fields of v and p and difference fields between reference and retrieved fields will be shown at $z = 3.5km$ and we will mention this no more. The assimilation window extends from 15 to 17 minute. The simulated observations are available only at the end of assimilation window. The simulated observations at the initial time will be used only to check the retrieved fields. The length of assimilation window is two minutes for our test problems since it is about the optimal assimilation length, which is obtained by looking at the graph (not shown) of RMS errors between the retrieved and reference fields for different assimilation lengths. Sun (1992) discussed in detail the optimal assimilation length for a model similar to ours.

3.3 Tangent linear model and its backward integration

A tangent linear model (TLM) describes the evolution of perturbations in a forecast model. It uses approximations that are linear with respect to perturbations of the model fields. It is called a tangent model because the linearization is around a time-evolving solution, and therefore, the coefficients of the linear model are defined by slopes of tangents to the nonlinear model trajectory in phase space (Errico *et al.*, 1993). For the validity of a TLM, readers may consults Lacarra and Talagrand (1988), Vukićević (1991) and Errico *et al.* (1993). We will focus on the backward integration of a TLM for the purpose of the ANA application.

If a TLM can be integrated backward in time, given a perturbation (forecast error) at the end of forecast period a backward integration of a TLM determines what perturbation (initial error), say at the initial time, causes the perturbation at the end of forecast period. At the end of the integration, the initial error (sensitivity) of the forecast error is obtained.

Unfortunately, in general, most TLMs may not be integrated backward in time due to the presence of irreversible processes such as mixing (diffusion) and divergence damping. However, simple modifications (See Subsection 3.3.1) to TLMs lead to modified TLMs that can be integrated backward in time and yield approximations to the initial errors (sensitivities) of the forecast errors. In the following, we describe the backward integration of TLMs.

3.3.1 Quasi-inverse of a tangent linear model

Most numerical prediction systems such as ARPS are a mixture of hyperbolic, elliptic and parabolic systems although hyperbolic terms (advection terms) dominant. Because of irreversible mixing and damping terms (parabolic in nature), their corresponding TLMs can not be integrated backward in time. However, a modified TLM with reversing signs of mixing and damping terms is well-posed and can be integrated backward in time. The backward integration of a such modified TLM is called quasi-inverse of the TLM (Pu *et al.*, 1996, 1996a and 1996b). Reversing signs of mixing terms in a TLM leads to mixings backward in time in the quasi-inverse of the TLM. The difference between the initial condition of a well-posed forward TLM and the solution at the initial time of the quasi-inverse of the TLM starting from a solution of the well-posed TLM at a future time is hard to quantify since it is the difference of fields from two different models. Empirical results show that they are "close". We will demonstrate this using the ARPS. A similar approach in integrating TLMs backward in time is used by Pu *et al.* (1996, 1996a and 1996b) to calculate sensitivity of forecast error to forecast sensitivity patterns.

In the quasi-inverse of the TLM of the ARPS, the artificial divergence damping terms and subgrid scale turbulence mixing terms have opposite signs as those in the ARPS. Namely, αDiv^* and $-G^{\frac{1}{2}}D_u$, $-G^{\frac{1}{2}}D_v$, $-G^{\frac{1}{2}}D_w$ and $-G^{\frac{1}{2}}D_\theta$ are used in the quasi-inverse of the TLM instead of $-\alpha Div^*$ $G^{\frac{1}{2}}D_u$, $G^{\frac{1}{2}}D_v$, $G^{\frac{1}{2}}D_w$ and $G^{\frac{1}{2}}D_\theta$ in the TLM of the ARPS. Let U' be the same initial fields (Fig. 2) used to obtain the "simulated observations". The TLM solution fields at 17 minute, $X'(t_f)$, is used as the final condition for the quasi-inverse of the TLM. The solution fields of v and p of the quasi-inverse of the TLM at 15 minute are shown in Fig. 3. Comparing Fig. 2 with Fig. 3, one concludes that the solution of the quasi-inverse of the TLM of the ARPS approximately yields the initial perturbations fields. Although the initial perturbation fields thus obtained are different from the exact fields corresponding to U', their over all eddy structures, magnitudes and signs are "similar" to those in the exact fields. The correlation coefficients of anomaly between the exact u, v, w, θ and p at time t_0 and those obtained by the quasi-inverse of the TLM are 0.9997, 0.9996, 0.9729, 0.9306 and 0.8275, respectively. This indicates that the quasi-inverse of the TLM of the ARPS yields meaningful and useful results "direction-wise".

The validity of the quasi-inverse of a TLM may depend on the validity of the TLM itself. The study of the time length for which a quasi-inverse of a TLM yields "meaningful" results is an important issue, which is under investigation.

3.4 Adjoint model development

The adjoint of the ARPS was developed. The verification of the correctness of the gradient is conducted as follows. A Taylor series expansion applied in the direction $U' = W(X(t_f) - X^o(t_f))$ yields (Navon *et al.*, 1992)

$$\phi(\alpha) = \frac{J(U + \alpha U') - J(U)}{\alpha U'^T \cdot DJ(U)} = 1. + O(\alpha \|U'\|). \tag{28}$$

For small α but not too close to machine accuracy, the value of $\phi(\alpha)$ should be close to unity if the gradient $DJ(U)$ is correctly calculated. Numerical results (not shown) indicate that this is exactly the case.

4 Cost function, Weighting and Scaling

If the "simulated observations" are available at the beginning of the assimilation window without error, the ANA would find the minimum in one iteration since there is no need to integrate the quasi-inverse of the TLM, whose solution, $X(t_0) - X^o(t_0)$, is known in this case and $X^o(t_0) - X(t_0)$ is the step to the minimum (see remark 5 in Subsection 2.2). Therefore, in order to test the robustness of the ANA, "simulated observations" are made available at the end of the assimilation window. For simplicity, we assume that the simulated observations are of the same dimension as the model state. It is worth noting that although the ANA is presented in a semi-discrete form, to carry out numerical calculations only the cost function needs to be defined as a sum instead of an integral. For instance, in our experiments, the cost function is defined as

$$J(U) = \frac{1}{2}(X(t_f) - X^o(t_f))^T W (X(t_f) - X^o(t_f)), \tag{29}$$

where $X = (u, v, w, p, \theta)$ is state vector including all grid point values of model solution at a time and W is weighting matrix. Since there is no statistics available for the ARPS, a constant diagonal weighting matrix $W = diag\{W_u, W_v, W_w, W_p, W_\theta\}$ is used where

$$W_u^{-1} = \frac{1}{N}\sum_{i=1}^{N}\left[u_i(t_f) - u_i^o(t_f) - \frac{1}{N}\sum_{j=1}^{N}\Big(u_j(t_f) - u_j^o(t_f)\Big)\right]^2, \tag{30}$$

and N is the number of u-grid point values. Model solution $X = (u, v, w, p, \theta)$ is obtained by integrating the ARPS from zero initial guess. The weights thus defined nondimensionalize the cost function.

Scaling is a crucial issue in the success of nonlinear unconstrained optimization problems, and considerable research has been carried out on scaling nonlinear problems. It is well known that a badly scaled nonlinear programming problem can be almost impossible to solve (Navon and de Villiers 1983, Navon *et al.* 1992). An effective automatic scaling procedure would ease these difficulties and could also render problems that are well scaled, easier to minimize by improving condition number of their Hessian matrix (Thacker 1989).

Scaling by variable transformation converts variables from units that reflect physical properties to units that display desirable properties for the minimization process. Given a diagonal scaling matrix $S = diag\{S_u, S_v, S_w, S_p, S_\theta\}$ where S_u, S_v, S_w, S_p, and S_θ are constant diagonal sub-matrices, the general scaling procedure may be written as

$$X = SX^s, \tag{31}$$

$$\vec{g}^s = S\vec{g}, \tag{32}$$

$$H^s = S^T HS, \tag{33}$$

where H is the Hessian matrix. The constant diagonal elements of sub-matrix S_u will be calculated by

$$S_u = \left\{\frac{1}{N}\sum_{i=1}^{N}\left[u_i(t_f) - u_i^o(t_f) - \frac{1}{N}\sum_{j=1}^{N}\Big(u_j(t_f) - u_j^o(t_f)\Big)\right]^2\right\}^{\frac{1}{2}}, \tag{34}$$

and similarly for S_v, S_w, S_p and S_θ.

For complicated functions, difficulties may be encountered in choosing suitable scaling factors. Good scaling is problem dependent. A basic rule is that the variables of the scaled problem should be of similar magnitude and of order unity because within optimization routines convergence tolerances and other criteria are necessarily based on an implicit definition of "small" and "large", and, thus variables with widely varying orders of magnitude may cause difficulties of convergence for minimization algorithms (Gill and Murray 1981). One simple direct way to determine the scaling factor is to use the typical values for different fields. This issue is closely related to the issue of preconditioning (Yang *et al.* 1996).

5 Numerical results

5.1 Experiment Description

Two experiments are carried out to access the feasibility and efficiency of the ANA as a large scale minimization algorithm in the framework of 4-D variational data assimilation. In all experiments, the length of the assimilation window is 2 minutes and the big and small time steps are 6 and 1 seconds, respectively. The two experiments are:

1. "Simulated observations" without error.

   The model generated fields of u, v, w, θ and p at 17 minute (the end of the assimilation window) are used as simulated observations. *The initial guess is zero for all variables*, i.e. for the three components of the 3-D wind, pressure and potential temperature fields. *It is emphasized that the simulated observations are available in all fields and at final time t_f only.*

2. "Simulated observations" with random errors.

   Same as experiment 1 except that a 20% random error is added to the simulated observations at the *final time.* Since random "observation errors" are just random noise, their distributions are not shown.

The ANA will be compared with the usual adjoint method employing the LBFGS algorithm of Liu and Nocedal (1989, which will be called LBFGS method for simplicity). The full description of the LBFGS method may be found in Liu and Nocedal (1989). The same convergence criteria

$$\|g_{k+1}\| \leq 10^{-2}\|g_0\|, \tag{35}$$

will be used for both ANA and LBFGS method. This convergence criteria yields a reasonable quality of the retrieved fields in terms of RMS errors and correlation coefficients of anomaly relative to the reference fields (Fig. 2) of u, v, w, θ and p at the initial time.

The gradient of the cost function with respect to control variables is calculated by integrating the adjoint of the ARPS. In the LBFGS method, the gradient is used to find the descent direction and to check the convergence criteria (35) and line search conditions (77-78) while it is only used to check the convergence criteria (35) and line search conditions (77-78) in the ANA. In order to be consistent, for all experiments, the ANA and LBFGS method of Liu and Nocedal (1989) are fixed. No fine tuning is allowed. The number of updates in the LBFGS method is set to 5 at where the LBFGS algorithm performs best for our test problems. Computations are performed using the Cray C90 at the Pittsburgh Supercomputing Center.

5.2 Results and comparison of the ANA and LBFGS algorithms

For experiment 1, ANA requires 5 iterations, 11 function calls and 26.847s of CPU time to satisfy the convergence criteria (35) while the LBFGS method requires 78 iterations, 224 function calls and 431.131s to satisfy the same convergence criteria (35). Therefore the ANA is more than an order of magnitude faster than the LBFGS method in terms of both number of iterations and CPU time in this experiment. The variation of the cost function scaled by its initial value (J/J_0) as well as those of the norm of the gradient also scaled by its initial value ($\|\vec{g}\|/\|\vec{g}_0\|$) as functions of the number of iterations are displayed in Fig. 4, which indicates that the cost function obtained using ANA decreases one more order of magnitude than that obtained by using LBFGS method. The quality of the retrieved fields obtained by using ANA is also much better than that obtained by using LBFGS method, which can be seen by comparing (given below) the retrieved fields with reference fields.

Figs. 5 and 6 show that the retrieved fields obtained using the ANA are almost identical to the reference fields (Fig. 2) while those obtained by using the LBFGS method display clear differences with the reference fields (Fig. 2). Difference fields of v and p between the retrieved and reference fields are displayed in Figs. 7 and 8, respectively. After the minimization, the maximum difference values of v and p obtained using the ANA are about an order of magnitude smaller than those obtained using the LBFGS method. The difference fields of u, w and θ are similar (not shown).

Table 1 shows the RMS errors between the retrieved u, v, w, θ and p and their corresponding reference fields at the end of assimilation using both ANA and LBFGS methods while Table 2 shows the corresponding correlation coefficients of anomaly. Table 1 clearly indicates that the RMS errors in all fields obtained using ANA are an order of magnitude smaller than those obtained using the LBFGS method. The correlation coefficients of the anomaly (Table 2) obtained using ANA are larger than those obtained using LBFGS method in all fields.

In experiment 2 we test the feasibility and efficiency of the ANA in the case of "simulated observations with random errors". As in experiment 1, the ANA is very robust comparing with LBFGS method. It requires 7 iterations, 15 function calls and 36.556s of CPU time to satisfy the convergence criteria (35) while the LBFGS method requires 108 iterations, 295 function calls and 568.006s to satisfy the same condition. At the end of assimilation, the cost function obtained using ANA decreases by one more order of magnitude than that obtained by using the LBFGS method (Fig. 9).

The RMS errors and correlation coefficients of anomaly in Tables 1 and 2 indicate that the ANA yields more accurate results than the LBFGS method does, i.e. smaller RMS errors and larger correlation coefficients of anomaly in all fields but the differences are smaller comparing with those in experiment 1.

Figs. 10 and 11 show that the retrieved fields obtained using the ANA are a little closer to the reference fields (Fig. 2) than those obtained by using the LBFGS method. After the minimization, the difference between the maximum and minimum values of v and p in the difference fields of v and p between the retrieved and reference fields are .266m/s and 59.8 Pa when the ANA is used and .354m/s and 62.9Pa when LBFGS method is used. The quality of the retrieved fields in this case is similar although the ANA yields slightly better results (Tables 1 and 2).

6 Summary and conclusions

A new ANA suitable for 4-D variational data assimilation was applied to the ARPS. The new ANA finds the Newton descent direction by integrating the quasi-inverse of a TLM backward in time. Since most TLMs of numerical weather prediction models are not well posed when they are integrated backward in time due to mixing and damping effects, they have to be modified by reversing the signs of the mixing and damping terms such that the quasi-inverses of the TLMs are well posed when they are integrated backward in time. The solution of a quasi-inverses of a TLM at the initial time is only an approximation to Newton descent direction, which leads to a fast linear convergence rate of the ANA for our experiments. Numerical experiments using the ARPS indicate that the ANA is suitable for 4-D variational data assimilation problems encountered in meteorology. It is very efficient in terms of both number of iterations and CPU times. Therefore it offers an efficient alternative to large-scale unconstrained minimization algorithms for meteorology problems. Comparing with the LBFGS method of Liu and Nocedal (1989), the ANA is a clear winner for our test problems both in terms of efficiency as well as in terms of quality of retrieved fields.

The weakness of Newton's method (hence the ANA as well) is that when the initial guess is "too" far away from the minimum, it may fail. If such situations occur, one may use other minimization algorithms for one or two iterations and then revert to using the ANA.

The current ANA has not considered how to deal with the following:

1. Parameters other than initial and boundary conditions as part of control variables;

2. Non invertible operator C. This limitation can be avoided by reformulating the ANA as pointed out in the first remark following Theorem 2 in Appendix A;

3. How the validity of the backward integration of a TLM may depend on the validity of the TLM itself?

4. The backward integrations of the TLMs with physical processes;

5. Model errors.

How to formulate the ANA in these settings is under investigation.

Convergence criteria and line search conditions in the current version of the ANA can be carried out without the information of the gradient using techniques such as golden section search etc (Luenberger, 1984). Namely, the next version of the ANA to be tested in the near future will not use the gradient and the adjoint model. In this paper, the ANA is only tested using an adiabatic model and simulated observations. Our next paper will also deal with real data cases using models which include additional physical processes. The backward integration of TLMs with physical processes will also be discussed in a follow-up paper.

Acknowledgement

The first author gratefully acknowledges the insightful comments and useful discussions with Qin Xu and thanks Prof. S. Lakshmivarahan and Prof. Alan Shapiro at the University of Oklahoma for their interests and insightful comments. The first author is also thankful to Andrew Crook and T. Vukićević at NCAR for their interests and encouragement in this project. This research was supported by NSF grant ATM-912009. Supercomputing was provided by the Pittsburgh Supercomputing Center.

Appendices

A The theory of the adjoint Newton algorithm

For completeness, in this appendix we provide the mathematical foundations of the ANA (Wang *et al.*, 1996). Notations same as those in the text will be used. For simplicity, the proof is presented under the best and even unrealistic scenario by assuming, the model is perfect, the TLM can be integrated backward in time and the operator C is invertible. However, these assumptions may not limit the applications of the ANA as shown in the previous sections and the remarks below. We also assume that F: $R^n \longmapsto R^n$ is sufficiently differentiable to carry out our arguments. Discontinuous physical processes are very important. However, we do not deal with them at this time. $\Diamond$ will be used to denote the end of a proof.

Theorem 1. *Let:*

a. *Variables X, X', P and $\hat{P} \in R^n$ satisfy Eqs. (2), (4), (6) and (8), respectively.*

b. *The cost function defined by (1) and function $F(X)$ in Eq. (2) be sufficiently differentiable and the operator C: $R^n \longmapsto R^m$ in (1) be invertible.*

c. *There are no observation errors and the cost function defined by (1) has a unique minimum, U_m, such that $CX(U_m)(t) = X^o(t)$ for any time $t \in [t_0, t_f]$ where $X(U_m)(t)$ is the model solution with initial condition U_m.*

Then for a given U near the global minimum[3] U_m with

$$U' = U - U_m, \tag{36}$$

and for any time $t \in [t_0, t_f]$ we have

1. *The forcing terms in the first and second order adjoint models satisfy*

$$C^T W \left(CX(t) - X^o(t)\right) = \left(D^2 F(X) X'(t)\right)^T P(t) + C^T W C X'(t) + O(\|U'\|^2); \tag{37}$$

[3] See remark 1 after the proof of Theorem 1 for the meaning of "near the global minimum U_m".

2. *The first and second order adjoint variables satisfy*

$$P(t) = \hat{P}(t) + O(\|U'\|^2). \tag{38}$$

Proof.

1. The nonlinear model and tangent linear model solutions, X and X' obtained with initial conditions U and U', satisfy Eqs. (2) and (4), respectively. Using Taylor expansion (Berger, 1977), one has

$$X(U) = X(U_m) + DX(U)U' + O(\|U'\|^2), \tag{39}$$

i.e.

$$X(U) = X(U_m) + X' + O(\|U'\|^2). \tag{40}$$

Using Eq. (40) and $CX(U_m)(t) = X^o(t)$, one obtains

$$\begin{aligned} CX(U)(t) - X^o(t) &= C\left(X(U_m)(t) + X'(t) + O(\|U'\|^2)\right) - CX(U_m)(t) \\ &= CX'(t) + O(\|U'\|^2). \end{aligned} \tag{41}$$

Hence $C^T W\left(CX(t) - X^o(t)\right) = O(\|U'\|)$. From Eq. (6), one obtains (Dieudonne, 1960)

$$P(t) = O(\|U'\|). \tag{42}$$

Since F is sufficiently differentiable, $\|D^2F(X)\|$ is bounded. Hence

$$\left(D^2F(X)X'(t)\right)^T P(t) = O(\|U'\|^2),$$

which together with Eq. (41) leads to

$$\begin{aligned} &C^T W\left(CX(t) - X^o(t)\right) - \left(\left(D^2F(X(t))X'(t)\right)^T P + C^T WCX'(t)\right) \\ &= C^T W\left(CX'(t) + O(\|U'\|^2)\right) - O(\|U'\|^2) - C^T WCX'(t) \\ &= O(\|U'\|^2). \end{aligned} \tag{43}$$

Eq. (43) yields Eq. (37).

2. Subtracting Eq. (6) from Eq. (8), one obtains that $\hat{P}(t) - P(t)$ satisfies

$$\begin{aligned} -\frac{d\left(\hat{P}(t) - P(t)\right)}{dt} &= (DF(X))^T \left(\hat{P}(t) - P(t)\right) \\ &+ \left(D^2F(X)X'\right)^T P(t) + C^TWCX' - C^TW\left(CX - X^o\right), \end{aligned} \tag{44}$$

with final condition

$$\hat{P}(t_f) - P(t_f) = 0. \tag{45}$$

Eq. (44) is a linear equation with a forcing term satisfying Eq. (43). Hence Eqs. (44-45) yield Eq. (38) (Dieudonne, 1960).

$\diamondsuit$

Remarks:

1. Under the assumption of sufficient conditions for the convergence of Newton's method (Berger, 1977, Stoer and Bulirsch, 1976), we know that there exists a neighborhood, B, of U_m such that if $U \in B$, the Newton's method converges. "For a given U near the global minimum U_m ..." in Theorem 1 means $U \in B$.

2. In most cases, Eqs. (37) and (38) will be approximately satisfied. In Appendix B, we will show that Eqs. (37) and (38) can be exactly satisfied using a linear example. For nonlinear model equations, Eqs. (37) and (38) are satisfied with indicated accuracy. Interested readers may use

$$\frac{dX}{dt} = -X^2, \tag{46}$$

$$X(0) = U, \tag{47}$$

where time $t \in [0, 1]$ to verify Theorem 1.

Corollary 1. *If the same assumptions in Theorem 1 hold, then*

$$X'(t_f) = C^{-1}\left(CX(t_f) - X^o(t_f)\right) + O(\|U'\|^2), \tag{48}$$

where C^{-1} is the inverse of C.

Proof. From Theorem 1, we know that Eq. (37) is true at any time. Hence it is true at the time t_f in particular, i.e.

$$C^T W \left(CX(t_f) - X^o(t_f) \right) = \left(D^2 F(X) X'(t_f) \right)^T P(t_f) + C^T W C X'(t_f) + O(\|U'\|^2). \quad (49)$$

Noticing that $P(t_f) = 0$ given by Eq. (7) and C is invertible, solving Eq. (49) for $X'(t_f)$ one obtains Eq. (48) as a result. $\diamondsuit$

Theorem 2. *If the same assumptions in Theorem 1 hold and the tangent linear model can be integrated backwards in time, then the estimated Newton descent direction is given by* $d_e = -Y'(t_0) = -U'_e$ *where* $Y'(t_0)$ *is the solution of the following backward problem:*

$$\frac{dY'}{dt} = DF(X)Y', \quad (50)$$

$$Y'(t_f) = C^{-1} \left(CX(t_f) - X^o(t_f) \right), \quad (51)$$

such that

$$\|U' - U'_N\| = O(\|U'\|^2), \quad (52)$$

$$\|U'_e - U'_N\| = O(\|U'\|^2), \quad (53)$$

$$\|U'_e - U'\| = O(\|U'\|^2). \quad (54)$$

Proof. Since the tangent linear model can be integrated backwards in time, Eqs. (50) and (51) indicate that

$$Y' = DX(U)U'_e. \quad (55)$$

According to Eq. (41) and noticing $X' = O(\|U'\|)$, one has

$$Y'(t_f) = C^{-1} \left(CX(t_f) - X^o(t_f) \right) = CX'(t_f) + O(\|U'\|^2). \quad (56)$$

Hence

$$Y'(t) = O(\|U'_e\|) = O(\|U'\|). \quad (57)$$

Since the tangent linear model can be integrated backwards in time, the solution of the following equations

$$\frac{dZ'}{dt} = DF(X)Z', \tag{58}$$

$$Z'(t_f) = X'(t_f) = \left[DX(U)U'\right]_{t_f}, \tag{59}$$

is the same as that of Eqs. (4) and (5). Namely,

$$Z' = DX(U)U' = X'. \tag{60}$$

Eq. (41) at t_f yields

$$C^{-1}\left(CX(t_f) - X^o(t_f)\right) - X'(t_f) = O(\|U'\|^2), \tag{61}$$

which is a relationship between final conditions (51) and (59). Hence,

$$Y'(t) - X'(t) = Y'(t) - Z'(t) = DX(U)U'_e - DX(U)U' = O(\|U'\|^2). \tag{62}$$

At the minimum, the gradient of the cost function with respect to control variables is zero, i.e.

$$0 = DJ\left(U_m\right) = DJ(U - U') = DJ(U) - D^2J\left(U\right)U' + O(\|U'\|^2). \tag{63}$$

and

$$D^2J\left(U\right)U' = DJ(U) + O(\|U'\|^2). \tag{64}$$

Subtracting Eq. (14) from Eq. (64), one obtains

$$D^2J\left(U\right)\left(U' - U'_N\right) = O(\|U'\|^2). \tag{65}$$

Since by assumption $D^2J(U)$ is positive definite, Eq. (65) yields Eq. (52). Now consider the forcing terms of Eq. (44) with X' replaced by Y' and apply Eqs. (57), (42), (62) and (41), one obtains

$$\begin{aligned}
&\left\|\left(\left(D^2F(X)Y'\right)^T P(t) + C^TWCY'\right) - C^TW\left(CX - X^o\right)\right\| = \\
&\left\|\left(D^2F(X)Y'\right)^T P(t) + C^TWC(Y' - X') + C^TW\left(CX' - (CX - X^o)\right)\right\| \\
&= O(\|U\|^2) + O(\|U'\|^2) + O(\|U'\|^2) = O(\|U'\|^2)
\end{aligned} \tag{66}$$

That is

$$\hat{P}(t) - P(t) = D^2 J\,(U)\,U'_e - DJ(U) = O(\|U'\|^2). \tag{67}$$

Using Newton's equation Eq. (14), Eq. (67) becomes

$$D^2 J\,(U)\,(U'_e - U'_N) = O(\|U'\|^2), \tag{68}$$

which yields Eq. (53). ¿From Eqs. (52) and (53), one obtains

$$\|U'_e - U'\| \leq \|U'_e - U'_N\| + \|U'_N - U'\| + O(\|U'\|^2), \tag{69}$$

which yields Eq. (54). If U' is sufficiently small, U'_e is a descent direction since

$$\begin{aligned} U'^T U'_e &= U'^T (U'_e - U' + U') \geq \|U'\|^2 - \|U'\| \|U'_e - U'\| \\ &= \|U'\|(\|U'\| - \|U'_e - U'\|) = \|U'\|(\|U'\| - O(\|U'\|^2)) > 0. \end{aligned} \tag{70}$$

$\diamondsuit$

Remarks:

1. For simplicity, the operator C is assumed to be invertible. However, C is not necessarily invertible except in case it is applied to certain discrete systems. In this case, one may reformulate the Newton's equation

   $$D^2 J(U) U'_N = DJ(U), \tag{71}$$

   as a minimization problem of the following form

   $$\min_{U'} \frac{1}{2} \|D^2 J(U) U' - DJ(U)\|^2, \tag{72}$$

   where U' is the initial condition of tangent linear model given by Eqs. (4) and (5). Minimization problem (72) is quadratic and is similar to problem (1). All techniques such as penalty, model error techniques etc, suitable for problem (1) are also suitable for the quadratic problem (72). In this case, the tangent linear model is integrated forward. This is related to generalized inverse problem (Bennett, 1992, Caradus, 1974, Golub and Van Loan, 1989, Rao and Mitra, 1971) and is successfully applied to the ARPS. Our next paper will deal with this general case.

2. For completeness, the algorithm form of the ANA is listed here:

   (a) Choose an initial guess U_0 and set $k = 0$.

   (b) Calculate the gradient of the cost function with respect to initial conditions

$$g_k = g(U_k) = DJ(U_k), \tag{73}$$

   by integrating the first order adjoint model equations.

   (c) Obtain a Newton line search direction:

$$d_k = -Y'(t_0), \tag{74}$$

   by integrating the backward tangent linear model given by Eqs. (16) and (17).

   (d) Set

$$U_{k+1} = U_k + \alpha_k d_k, \tag{75}$$

   where α_k is the step-size obtained by conducting a line search

$$J(U_k + \alpha_k d_k) = \min_{\alpha} J(U_k + \alpha d_k), \tag{76}$$

   using Davidon's cubic interpolation method for the line search of the stepsize and which satisfies the following Wolfe conditions (Liu and Nocedal, 1989):

$$f(U_k + \alpha_k d_k) \leq f(U_k) + \beta' \alpha_k g_k^T d_k, \tag{77}$$

   and

$$\frac{g_k(U_k + \alpha_k d_k)^T d_k}{g_k^T d_k} \leq \beta, \tag{78}$$

   where $\beta' = 0.0001$, $\beta = 0.9$.

   (e) Check the convergence condition. Given a tolerance criterion ϵ: $10^{-14} \leq \epsilon \leq 10^{-1}$, and if

$$||g_{k+1}|| \leq \epsilon ||g_0||, \tag{79}$$

   stop. Otherwise, set $k = k + 1$ and go to step b.

3. The adjoint Newton method requires the user to provide a subroutine to calculate the cost function and *a subroutine to calculate the Newton direction by integrating the backwards tangent linear model.* In the current version of the ANA, the gradient is only used to check the convergence criteria (79) and line search conditions (77-78). Those criteria and conditions could be carried out without the gradient information (Luenberger, 1984). In addition to above calculations, each iteration of the adjoint Newton iteration requires two stages: setup (performed once per iteration), and line search iteration. The numerical costs of the two stages are approximately 2n flops and 4n flops (Nash and Nocedal, 1989), respectively, where "flops" denotes additions, subtractions, multiplications or divisions. The ANA is a large-scale unconstrained minimization Newton algorithm which does not have the limitation of the storage problem since it does not require the calculation of either the Hessian or its inverse. However it may be sensitive to the initial guess and may only be applied to the problems where their corresponding backward problems could be solved with "reasonable accuracy".

B A simple example

We first show that Eqs. (37) and (38) are exactly satisfied by using a very simple linear example and then we illustrate the ANA by using the same example.

Let us consider the following 1-dimensional model equation

$$\frac{dX}{dt} = -X, \tag{80}$$

$$X(0) = U, \tag{81}$$

where time $t \in [0, 1]$. The solution of the model is

$$X = Ue^{-t}. \tag{82}$$

Suppose that the simulated observations are model generated with the initial condition

$$X(0) = 1, \tag{83}$$

then from Eq. (82), the simulated observation may be written as

$$X^o = e^{-t}. \tag{84}$$

Let us define the cost function as

$$J(U) = \frac{1}{2}\int_0^1 (X - X^o)^2 dt, \tag{85}$$

then

$$J(U) = \frac{(U-1)^2}{4}(1 - e^{-2}). \tag{86}$$

The first order adjoint model of Eqs. (80) and (81) may be written as

$$-\frac{dP}{dt} = -P + (X - X^o), \tag{87}$$

i.e.

$$-\frac{dP}{dt} = -P + (U-1)e^{-t}, \tag{88}$$

$$P(1) = 0, \tag{89}$$

where P is the adjoint variable. The gradient of the cost function with respect to the initial conditions is given by

$$DJ(U) = P(0). \tag{90}$$

Eq. (88) has an analytic solution of the following form

$$\begin{aligned} P(t) &= \overline{P}e^t - e^t \int_0^t e^{-\tau}(U-1)e^{-\tau} d\tau \\ &= \overline{P}e^t + \frac{(U-1)}{2}(e^{-t} - e^t), \end{aligned} \tag{91}$$

where $\overline{P}$ is a constant to be determined by the final condition (89). Therefore $\overline{P} = ((U-1)/2))(1 - e^{-2})$, and Eq. (91) yields

$$P(t) = \frac{(U-1)}{2}(1 - e^{-2})e^t + \frac{(U-1)}{2}(e^{-t} - e^t), \tag{92}$$

and

$$P(0) = \frac{(U-1)}{2}(1 - e^{-2}). \tag{93}$$

Eq. (93) is exactly the gradient of the cost function (86) with respect to the initial condition U.

Let us now consider a perturbation, U', on the initial condition U for X. Then the tangent linear and second order adjoint models are:

$$\frac{dX'}{dt} = -X', \tag{94}$$

$$X'(0) = U', \tag{95}$$

and

$$-\frac{d\hat{P}}{dt} = -\hat{P} + X', \tag{96}$$

$$\hat{P}(1) = 0, \tag{97}$$

respectively. They have the following exact solutions

$$X' = U'e^{-t}, \tag{98}$$

and

$$\hat{P}(t) = \frac{U'}{2}(1 - e^{-2})e^{t} + \frac{U'}{2}(e^{-t} - e^{t}), \tag{99}$$

respectively. $\hat{P}(0) = U'(1 - e^{-2})/2$ is exactly Hessian vector product $D^2J(U)U'$.

It can be shown that

$$\hat{P}(0) = U'\frac{(1 - e^{-2})}{2} = \frac{(U - 1)}{2}(1 - e^{-2}) = P(0)$$

if $U' = U - 1$, i.e. Eq. (38) can be exactly satisfied in this linear case. In nonlinear cases, Eq. (38) is only an approximation. When $U' = U - 1$, the forcing terms of Eqs. (88) and (96) are exactly equal, $(U - 1)e^{-t} = U'e^{-t}$, which indicates that Eq. (37) is exactly satisfied for this problem.

The variational data assimilation aims to find the best initial condition, $X(0) = 1$, which minimizes the cost function given by (86). It should be realized that since the model itself is linear, its tangent linear model is identical to itself. Since we can solve it exactly, an exact Newton descent direction could be obtained and the minimum point is found in one step, i.e.:

1. Given an arbitrary initial guess U_0.

2. Solving the backward problem

$$\frac{dY'}{dt} = -Y', \tag{100}$$

$$Y'(1) = X(t_f) - X^o(t_f) = (U_0 - 1)e^{-1}, \tag{101}$$

one obtains

$$d_0 = -Y'(0) = -(U_0 - 1). \tag{102}$$

3. Let

$$U_1 = U_0 + \alpha_0 d_0, \tag{103}$$

where α_0 is the step-size obtained by the following line search

$$J(U_0 + \alpha_0 d_0) = \min_{\alpha} \left\{ \frac{(U_0 - \alpha(U_0 - 1) - 1)^2}{2}(1 - e^{-2}) \right\}, \tag{104}$$

which has an exact solution $\alpha_0 = 1$.

4. Using the newly found α_0 we obtain that

$$U_1 = U_0 + \alpha_0 d_0 = U_0 - (U_0 - 1) = 1, \tag{105}$$

which is the best initial condition that minimizes the cost function given by Eq. (86), i.e.

$$J(U_1) = 0. \tag{106}$$

So the adjoint Newton algorithm algorithm has found the minimum in a single step.

C Convergence analysis

For a quick assessment of the rate of convergence, a table can be constructed of

$$\xi_k = J_{k-1} - J_k$$

for the last few values of k (Gill and Murray, 1972). In our experiment, J_k denotes the value of the cost function at the kth iteration scaled by J_0. Superlinear convergence would be indicated if $\xi_{k+1} \approx \xi_k^r$, where $r > 1$. Fast linear convergence would be indicated if $\xi_{k+1} \approx \xi_k/M$, where $M > 2$.

In forming the sequence $\{\xi_k\}$, the user needs to be aware that eventually all algorithms either fail to make further progress or display slow linear convergence near the limiting accuracy of the solution. What may occur with a superlinearly convergent algorithm is that the sequence $\{\xi_k\}$ will demonstrate superlinear convergence for a few iterations only, and then lapse into slow linear convergence when limiting accuracy has been achieved. Therefore, it is important, especially if a failure has been indicated, to examine the sequence $\{\xi_k\}$ at iterations that sufficiently precede the final stage (Gill and Murray, 1972).

After 18 iterations, the cost function decreased 8 orders of magnitude when the ANA is used. We construct Table 3 for iterations 1 to 5 and 14 to 18. This table indicates that the ANA has a fast linear convergence rate for the first 5 iterations with average $M = 8.77$ while its convergence rate slows down for the last five iterations where average $M = 1.79$. Similar studies are carried out for LBFGS method. It is found out that the LBFGS has a linear convergence rate with average $M = 2.38$ for the first five iterations and $M = 0.63$ for iterations 74 to 78.

As pointed out in the third remark in Subsection 2.2, ANA has a theoretical quadratic convergence rate. It is a Newton method. However, its numerical convergence rate in this example is only fast linear. This is due to the fact that once modifications are introduced in the backward integration of the tangent linear model, its solution at the initial time will not be as good as the Newton descent direction. Hence the corresponding ANA will not display a quadratic convergence rate.

REFERENCES

Bennett, A. 1992: *Inverse problem in physical oceanography.* Cambridge University Press, pp. 346.

Berger, M. S., 1977: *Nonlinearity and functional analysis.* Academic Press, New York, pp. 417.

Caradus, S. R., 1974: *Operator theory of the pseudo-inverse.* A Queen's Papers in Pure and Applied Mathematics, No. **38**, Queen's University, Kingston, Ontario, Canada, pp. 67.

Droegemeier, K., M. Xue, K. Johnson, M. O'keefe, A. Swadey, G. W. Sabot, S. Wholey, K. Mills, and N. Lin, 1995: Weather Prediction: A Scalable Storm-Scale Model. *High performance computing.*, Addison Wesley, 45-92.

Davidon, W. C., 1959: Variable metric method for minimization. A. E. C. *Research and Development Report*, ANL-5990 (Rev.).

Dieudonne, J., 1960: *Foundations of modern analysis.* Academic Press, New York, pp. 361.

Errico, R. M., T. Vukićević and K. Raeder, 1993: Examination of the accuracy of a tangent linear model. *Tellus,* **45A**, 462-477.

Gill, P. E. and W. Murray, 1972: Quasi-Newton methods for unconstrained optimization. *J. Inst. Maths Applics*, **9**, 91-108.

Gill, P. E. and W. Murray, 1981: *Practical optimization.* Academic Press, 401 pp.

Golub, G. H. C. F. Van Loan, 1989: *Matrix computations.* 2nd Edition, The Johns Hopkins University Press, Baltimore and London, pp. 642.

Klemp, J.B., and R.B. Wilhelmson, 1978: The simulation of three-dimensional convective storm dynamics. *J. Atmos. Sci.*, **35**, 1070-1096.

Lacarra, J. F. and O. Talagrand, 1988: Short-range evolution of small perturbations in a barotropic model. *Tellus,* **40A**, 81-95.

Le Dimet, F. X. and O. Talagrand, 1986: Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects. *Tellus*, **38A**, 97-110.

Liu, D. C and Jorge Nocedal, 1989: On the limited memory BFGS method for large scale minimization. *Math. Prog.*, **45**, 503-528.

Luenberger, David G., 1984: *Linear and nonlinear programming.* 2nd edition, Addison-Wesley Inc., Reading, MA, pp. 491.

Navon, I. M. and D. M. Legler, 1987: Conjugate gradient methods for large scale minimization in meteorology. *Mon. Wea. Rev.*, **115**, 1479-1502.

Navon, I. M. , X. L. Zou, J. Derber and J. Sela, 1992: Variational data assimilation with an adiabatic version of the NMC Spectral Model. *Mon. Wea. Rev.*,**122**, 1433-1446.

Navon, I. M., X. Zou, M. Berger, P. K. H. Phua, T. Schlick and F. X. Le Dimet, 1992a: Numerical experience with Limited Memory Quasi-Newton and truncated Newton Methods. Optimization Techniques and Applications, Vol 1, Edited by K.H. Phua et al. World Scientific Publishing Co., 33-48.

Navon, I. M., X. Zou, M. Berger, P. K. H. Phua, T. Schlick and F. X. Le Dimet, 1992b: Testing for Reliability and Robustness of Optimization Codes for Large Scale Optimization Problems. Optimization Techniques and Applications, Vol 1, Edited by K.H. Phua et al. World Scientific Publishing Co., 445–480.

Navon, I. M. and R. de Villiers, 1983: Combined penalty multiplier optimization methods to enforce integral invariants conservation. *Mon. Wea. Rew.*, **111**, 1228-1243.

Nash, S. G., 1984a: Newton-type minimization via the Lanczos method. *SIAM J. Numer. Anal.*, **21** (4), 770-788.

Nash, S. G., 1984b: Truncated-Newton methods for large-scale function minimization. *Applications of Nonlinear Programming to Optimization and Control,* H.E. Rauch (ed.), Pergamon Press, Oxford, 91-100.

Nash, S. G., 1985: Preconditioning of truncated-Newton methods. *SIAM J. Sci. Stat. Comput.*, **6** (3), 599-616.

Nash, S. G. and A. Sofer, 1989: Block truncated-Newton methods for parallel optimization. *Math. Prog.*, **45**, 529-546.

Nash, S. G. and Jorge Nocedal, 1989: A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *Tech. Rep. NAM*, **02**, Dept. of Electrical Engineering and Computer Science, Northwestern University, Dec., pp. 19.

Nocedal, J., 1980: Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, **35**, 773-782.

Rao, C. R. and S. K. Mitra, 1971: *Generalized inverse of matrices and its applications to statistics.* John Wiley & and Sons, pp. 240.

O'Leary, D. P., 1983: A discrete Newton algorithm for minimizing a function of many variables. *Math. Prog.*, **23**, 20-23.

Pu, Z., E. Kalnay and J. Sela, 1996: Sensitivity of forecast error to initial conditions with a quasi-inverse linear method. Accepted for publication by *Mon. Wea. Rev.*.

Pu, Z., E. Kalnay, J. Derber and J. Sela, 1996a: Using past forecast error to improve the future forecast skill. *Preprint of the 11th Conference on Numerical Weather Prediction*, 19-23, Aug., 1996, Norfolk, Virginia, USA, 142-143.

Pu, Z., E. Kalnay and J. Derber, 1996b: Use of tangent linear model in the study of the sensitivity of forecast error to initial conditions. WMO Weather Prediction Research Programs. CAS/JSC Working group on Numerical Experimentation, Research Activities in Atmospheric Oceanic Modeling, Edited by A. Staniforth, Report No. 23, Feb. 1996. WMO/ID-No. 734, 626-627.

Shanno, D. F. and K. H. Phua, 1980: Remark on algorithm 500 – a variable method subroutine for unconstrained nonlinear minimization. *ACMTOMS*, **6**, 618-622.

Schlick, T. and A. Fogelson, 1992a: TNPACK–A truncated Newton minimization Package for large-scale problems: I. Algorithm and usage. *ACMTOMS*, **18** (1), 46-70.

Schlick, T. and A. Fogelson, 1992b: TNPACK–A Truncated Newton minimization package for large-scale problems: II. Implementation examples. *ACMTOMS*, **18** (1), 71-111.

Smolarkiewicz, Piotr K., 1983: A simple positive definite advection scheme with small implicit diffusion. *Mon. Wea. Rev.* **111**, 479-486.

Stoer, J. and R. Bulirsch, 1976: *Introduction to numerical analysis.* 2nd Ed., Springer-Verlag, New York, pp. 659.

Sun, J., 1992: *Convective scale 4-D data assimilation using simulated single Doppler radar* *observations.* Ph.D. Thesis. University of Oklahoma, Norman, OK, pp. 172.

Thacker, W. C.,1989: The Role of Hessian Matrix in Fitting models to Measurements. *J. Geophys. Res.*, **94**, 6177-6196.

Thépaut, J. D. Vasiljevic and P. Courtier, 1993: Variational assimilation of conventional meteorological observations with a multilevel primitive-equation model. *Q. J. R. Meteorol. Soc.*, **119**, 153-186

Vukićević, T., 1991: Nonlinear and linear evolution of initial forecast error. *Mon. Wea. Rev.*, **119**, 1602-1611.

Wang, Zhi, I. M. Navon, F. X. Le Dimet and X. Zou, 1992: The Second Order Adjoint Analysis: Theory and Application. *Meteorol. and Atmos. Phy.*, **50**, 3-20.

Wang, Zhi, 1993: Variational data assimilation with 2-D shallow water equations and 3-D FSU glob al spectral models. *Tech. Rep. FSU-SCRI-93T-149,* Florida State University, Tallahassee, Florida, 235 pp.

Wang, Zhi, I. M. Navon, X. Zou and F. X. Le Dimet, 1995: The adjoint truncated Newton algorithm for large-scale unconstrained optimization. *Computational Optimization and Applications,* **4** (3), 241-262.

Wang, Zhi, K. Droegemeier and L. White, 1996: The adjoint Newton algorithm for large-scale unconstrained optimization in meteorology applications. *Accepted for publication by Computational Optimization and Applications.*

Weisman, M.L., and J.B. Klemp, 1982: The dependence of numerically simulated convective storms on vertical wind shear and buoyancy. *Mon. Wea. Rev.*, **110**, 504-520.

Xue, M., K. Droegemeier, V. Vong, A. Shapiro and K. Brewster, 1995: *ARPS Version 4.0 User's Guide.* Center for the Analysis and Prediction of Storm, University of Oklahoma, pp. 380.

Yang, W., I. M. Navon and P. Courtier, 1996: A new Hessian preconditioning method applied to variational data assimilation experiments using NASA General Circulation Models. *Mon. Wea. Rev.*, **124** (5), 1000-1017.

Zupanski, D. and F. Mesinger, 1995: Four-dimensional variational data assimilation of precipitation data. *Mon. Wea. Rev.*, **123**, 1112-1127.

Zupanski, M., 1993: A preconditioning algorithm for large scale minimization problems. *Tellus*, **45A**, 578-592.

Zupanski, M., 1993a: Regional four-dimensional variational data assimilation in a quasi-operational forecasting Environment. *Mon. Wea. Rev.*, **121**, 2396-2408.

Zupanski, M., 1996: A preconditioning algorithm suitable for a general four-dimensional data assimilation. *Preprint of the 11th Conference on Numerical Weather Prediction*, 19-23, Aug., 1996, Norfolk, Virginia, USA, 241-242.

Figure captions

Fig. 1. Schematic presentations of adjoint and tangent linear approaches in finding descent directions. Here U^{new}, U^{old}, a and d denote the new and old estimates for U, step length and descent direction, respectively. The update step, $U^{new} = U^{old} + ad$, is not a part of the process of finding descent direction but for completeness it is shown here

Fig. 2. Initial perturbation fields (reference fields) of v (top panel) and p (bottom panel) at $z = 3.5km$. Zero contour lines are kept since they clearly separate positive and negative contour lines.

Fig. 3. Initial sensitivity fields of v (top panel) and p (bottom panel) at $z = 3.5km$ obtained by integrating the modified TLM backward in time.

Fig. 4. Variations of the log of the scaled cost function (J_k/J_0, top panel) and scaled gradient norm ($||\vec{g}_k||/||\vec{g}_0||$, bottom panel) with the number of iterations using algorithms: adjoint Newton algorithm (ANA) and LBFGS algorithm, respectively in experiment 1.

Fig. 5. Retrieved fields of v (top panel) and p (bottom panel) at the initial time and $z = 3.5km$ using the ANA in experiment 1.

Fig. 6. Same as Fig. 5 except the LBFGS is used.

Fig. 7 Difference fields between the retrieved fields of v (top panel) and p (bottom panel) at $z = 3.5km$ using ANA method and their corresponding reference fields in experiments 1, respectively.

Fig. 8. Same as Fig. 7 except the LBFGS is used.

Fig. 9. Same as Fig. 4 except they are for experiment 2.

Fig. 10. Same as Fig. 5 except they are for experiment 2.

Fig. 11. Same as Fig. 6 except they are for experiment 2.

Table captions

Table 1. RMS errors at the initial time between the retrieved u, v, w, θ and p fields and corresponding reference fields normalized by their initial values in experiments 1 and 2, respectively. Iter. denotes number of iterations.

Table 2. Correlation coefficients of anomaly at the initial time for u, v, w, θ and p between the retrieved and corresponding reference fields for experiments 1 and 2, respectively. Corr. denotes correlation coefficient.

Table 3. Convergence analysis information for adjoint Newton algorithm in experiment 1.

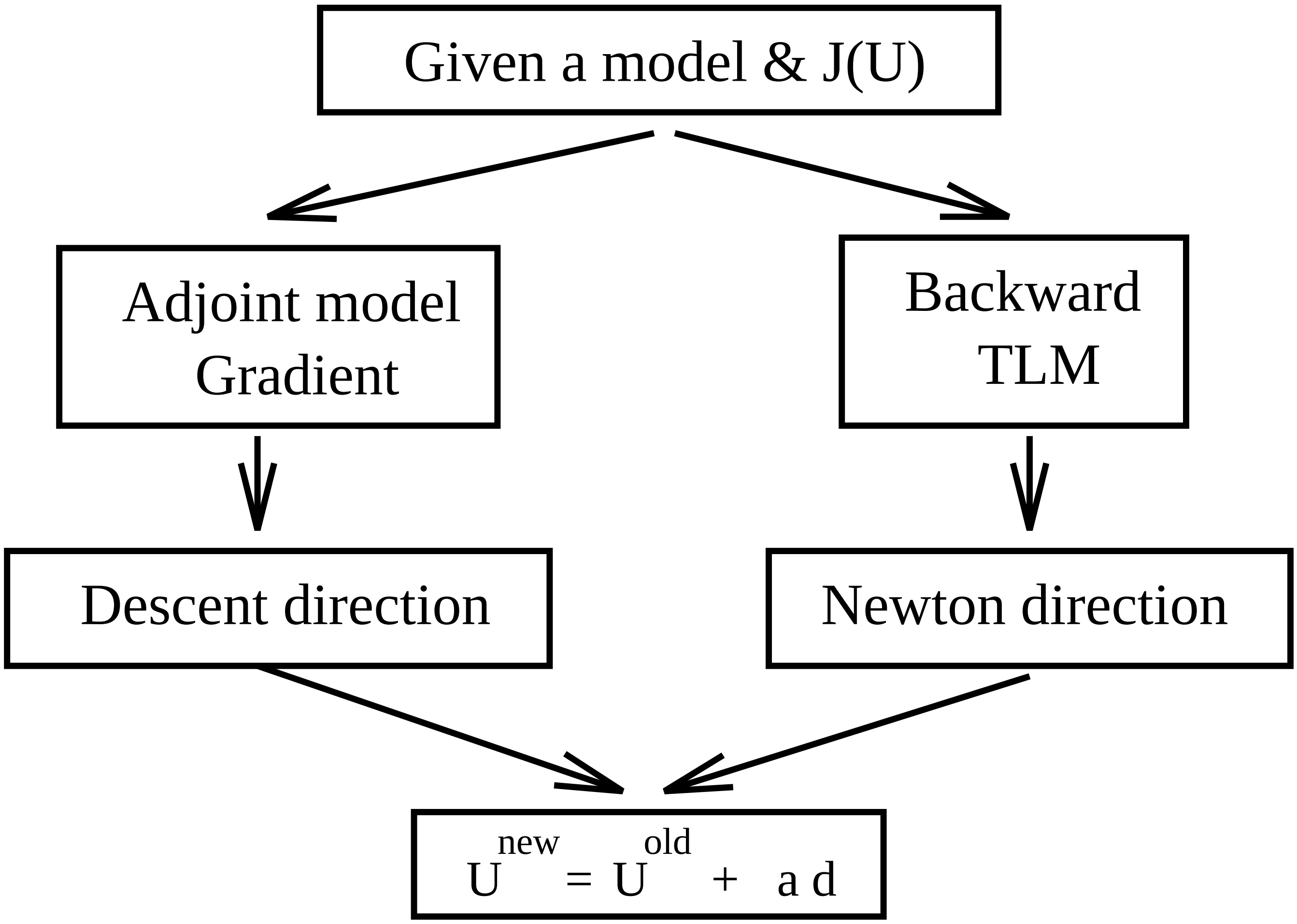


Figure 1: Schematic presentations of adjoint and tangent linear approaches in finding descent directions. Here U^{new}, U^{old}, a and d denote the new and old estimates for U, step length and descent direction, respectively. The update step, $U^{new} = U^{old} + ad$, is not a part of the process of finding descent direction but for completeness it is shown here.

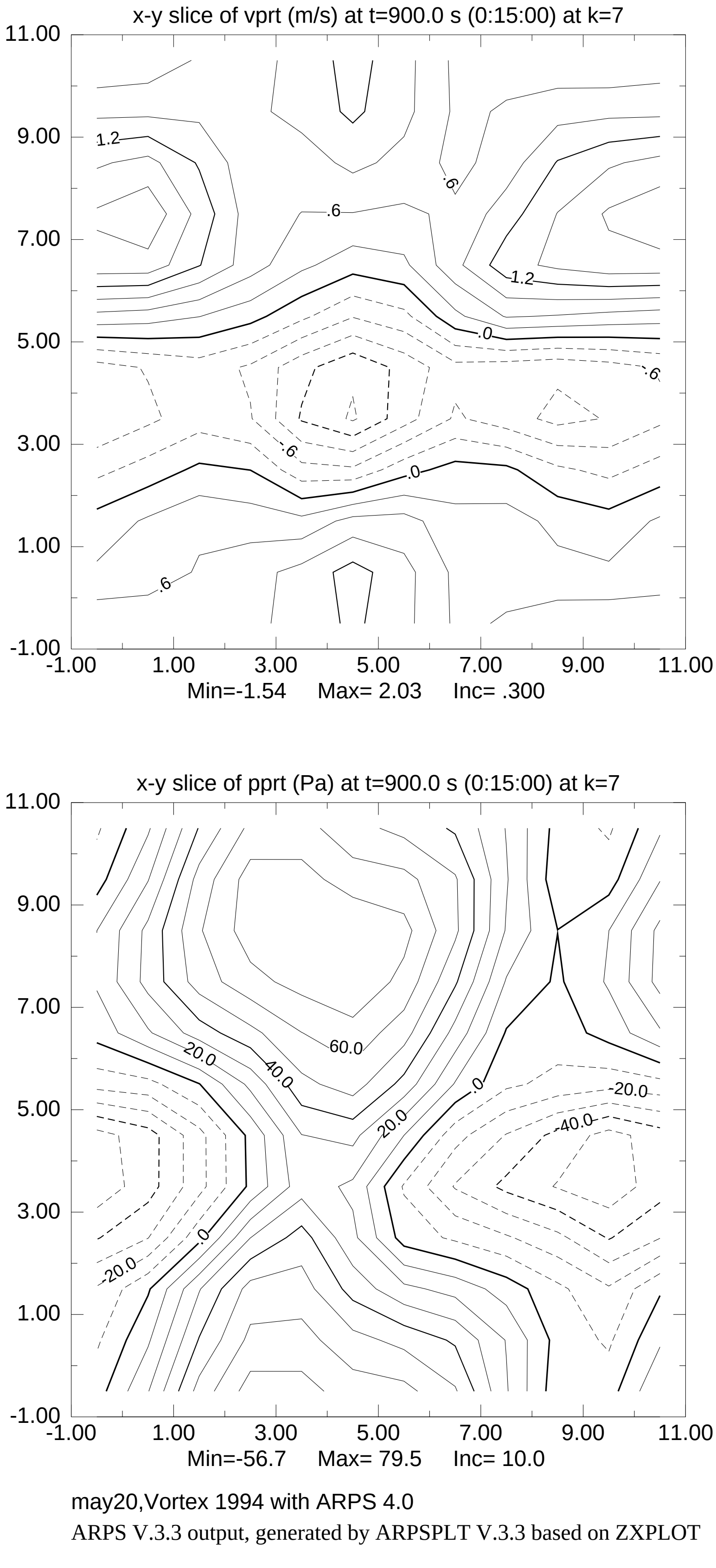


Figure 2: Initial perturbation fields (reference fields) of v (top panel) and p (bottom panel) at $z = 3.5km$. Zero contour lines are kept since they clearly separate positive and negative contour lines.

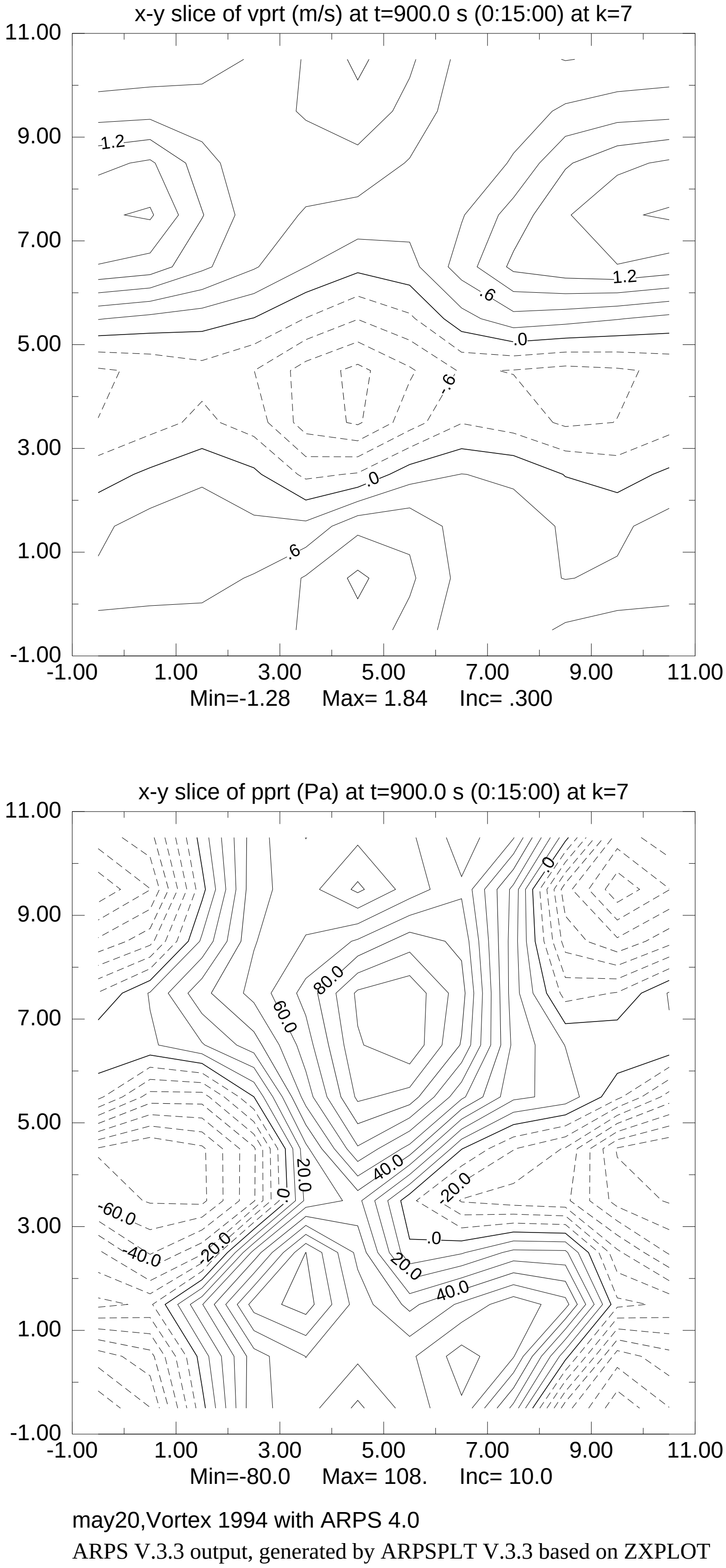


Figure 3: Initial sensitivity fields of v (top panel) and p (bottom panel) at $z = 3.5km$ obtained by the quasi-inverse of the TLM.

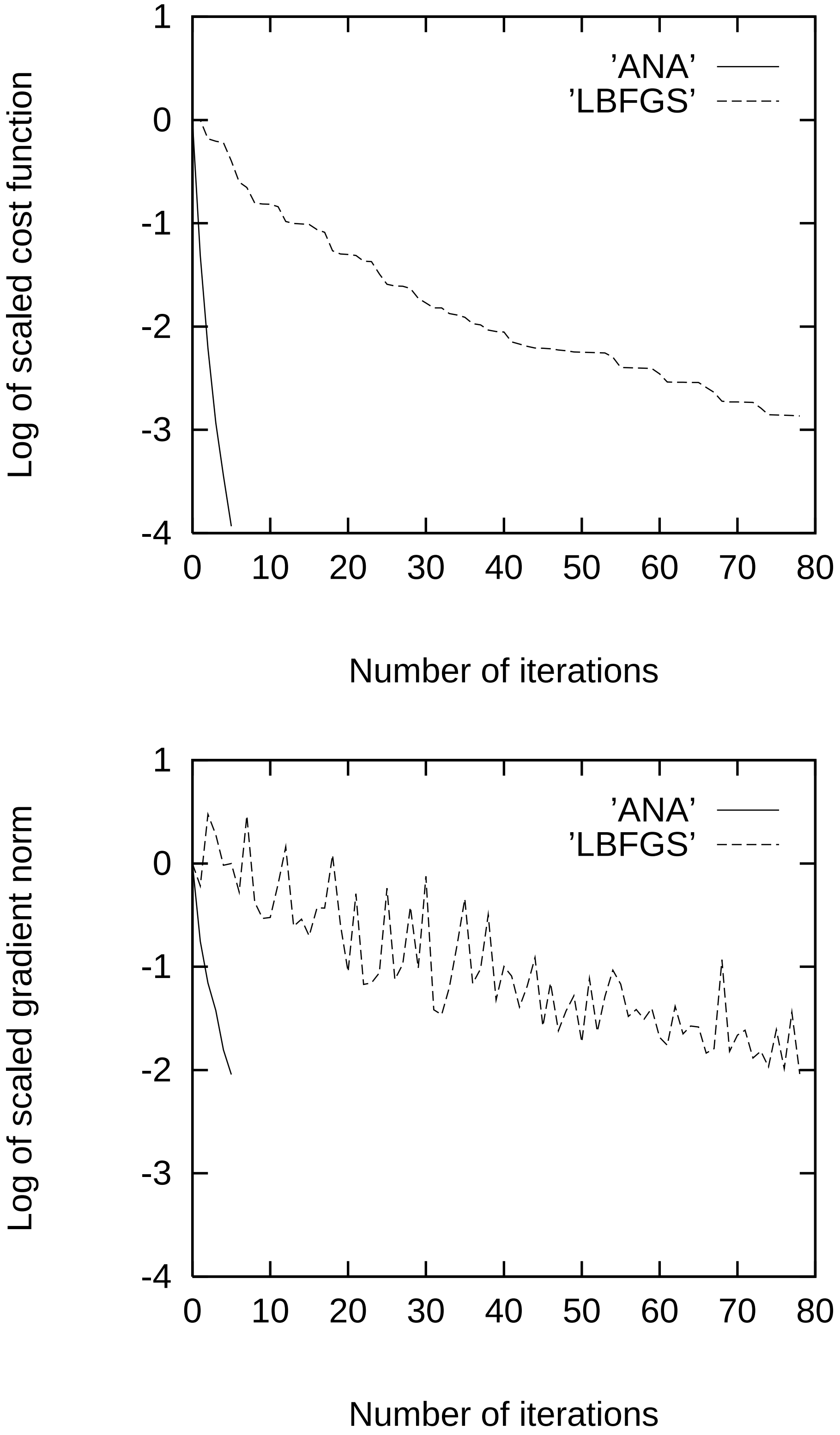


Figure 4: Variations of the log of the scaled cost function (J_k/J_0, top panel) and scaled gradient norm ($\|\vec{g}_k\|/\|\vec{g}_0\|$, bottom panel) with the number of iterations using algorithms: adjoint Newton algorithm (ANA) and LBFGS algorithm, respectively in experiment 1.

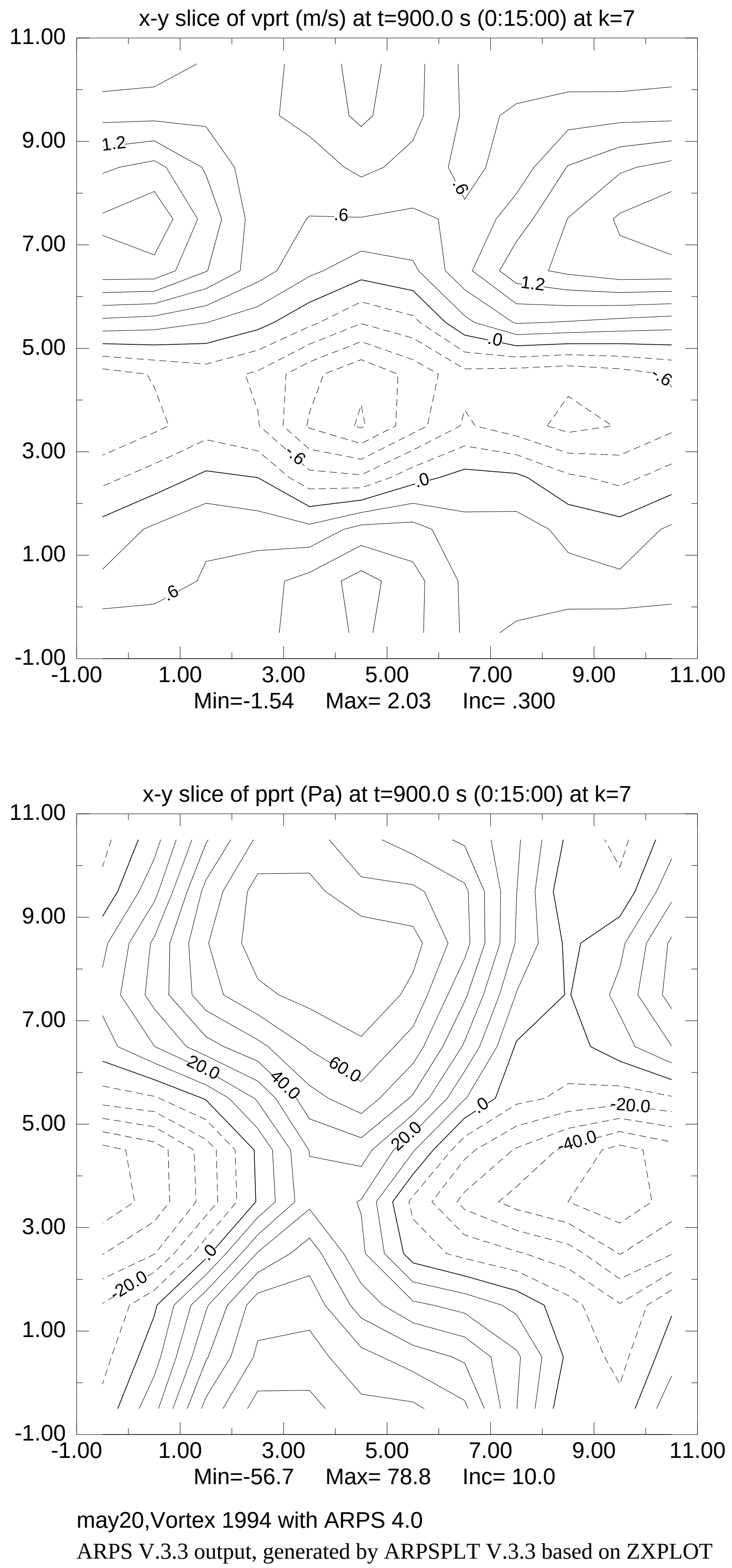
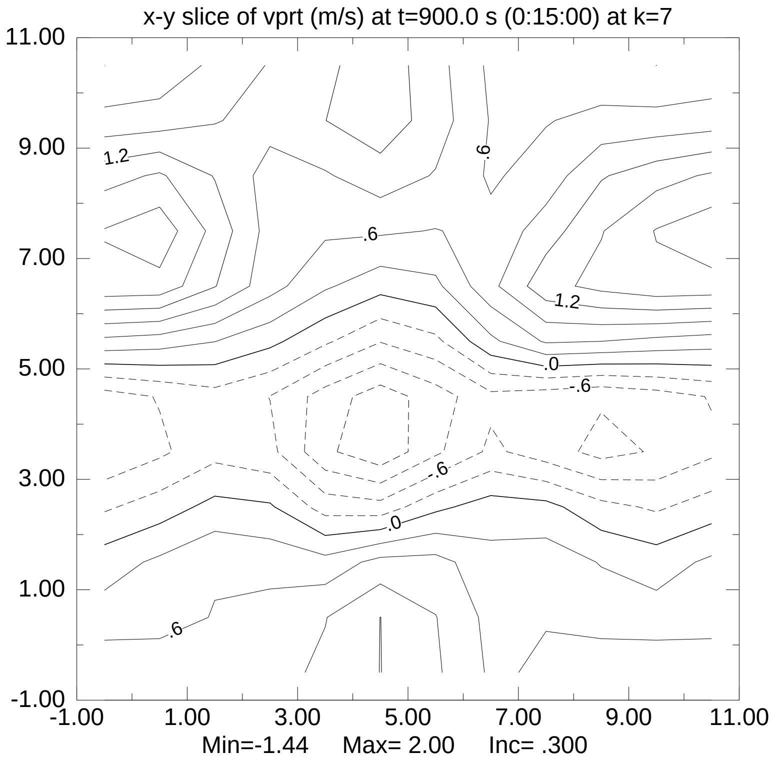


Figure 5: Retrieved fields of v (top panel) and p (bottom panel) at the initial time and $z = 3.5km$ using the ANA in experiment 1.



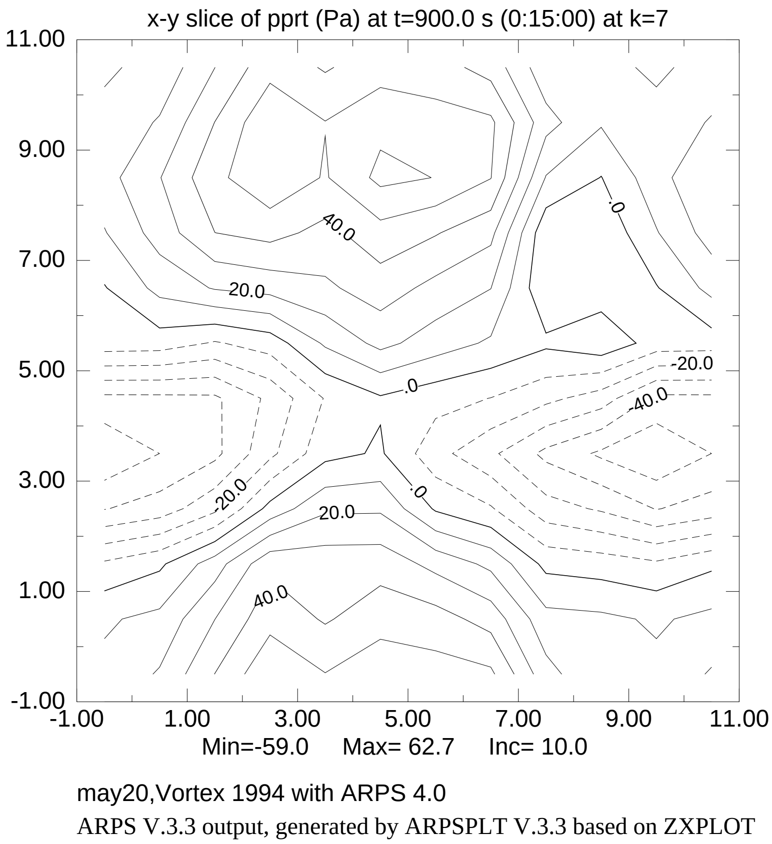


Figure 6: Same as Fig. 5 except the LBFGS is used.

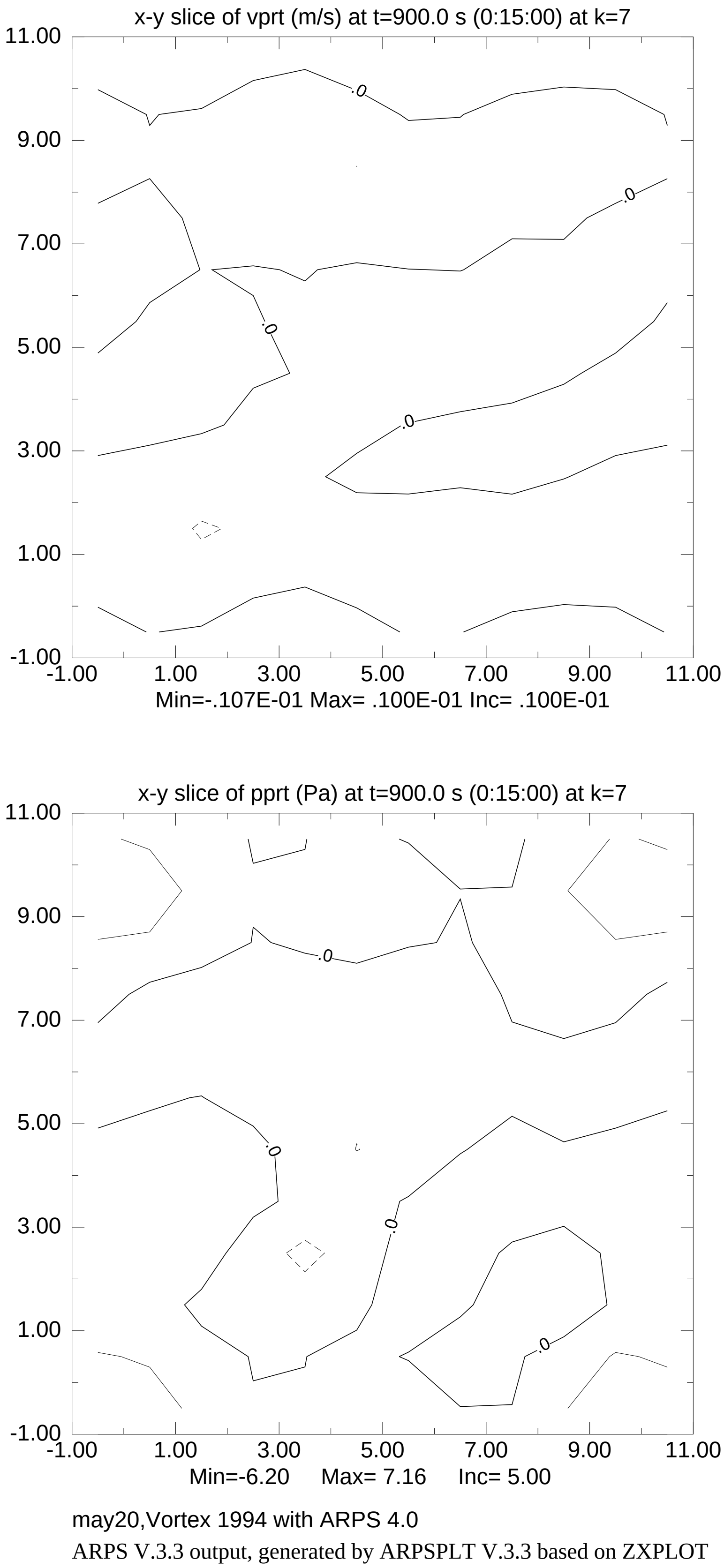


Figure 7: Difference fields between the retrieved fields of v (top panel) and p (bottom panel) at $z = 3.5km$ using ANA method and their corresponding reference fields in experiments 1, respectively.

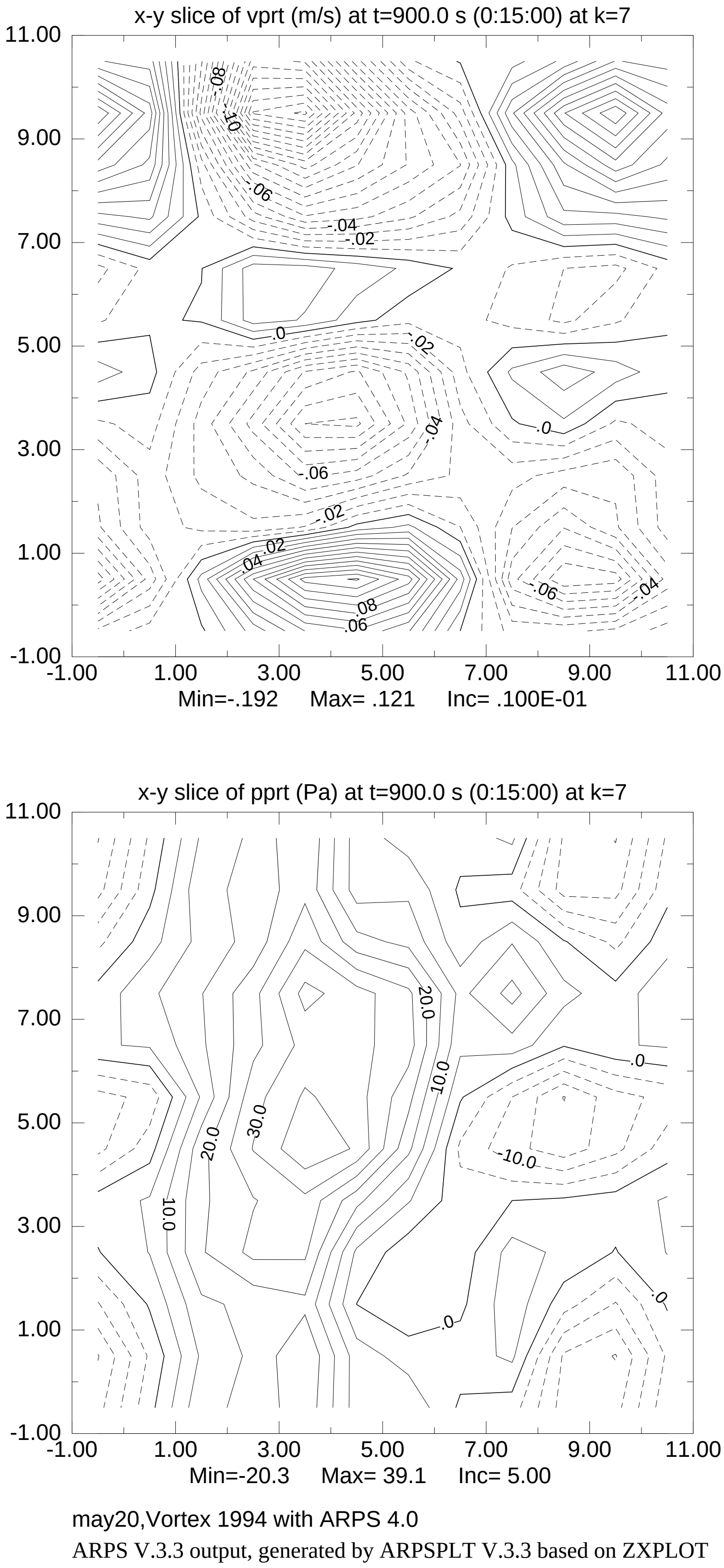


Figure 8: Same as Fig. 7 except the LBFGS is used.

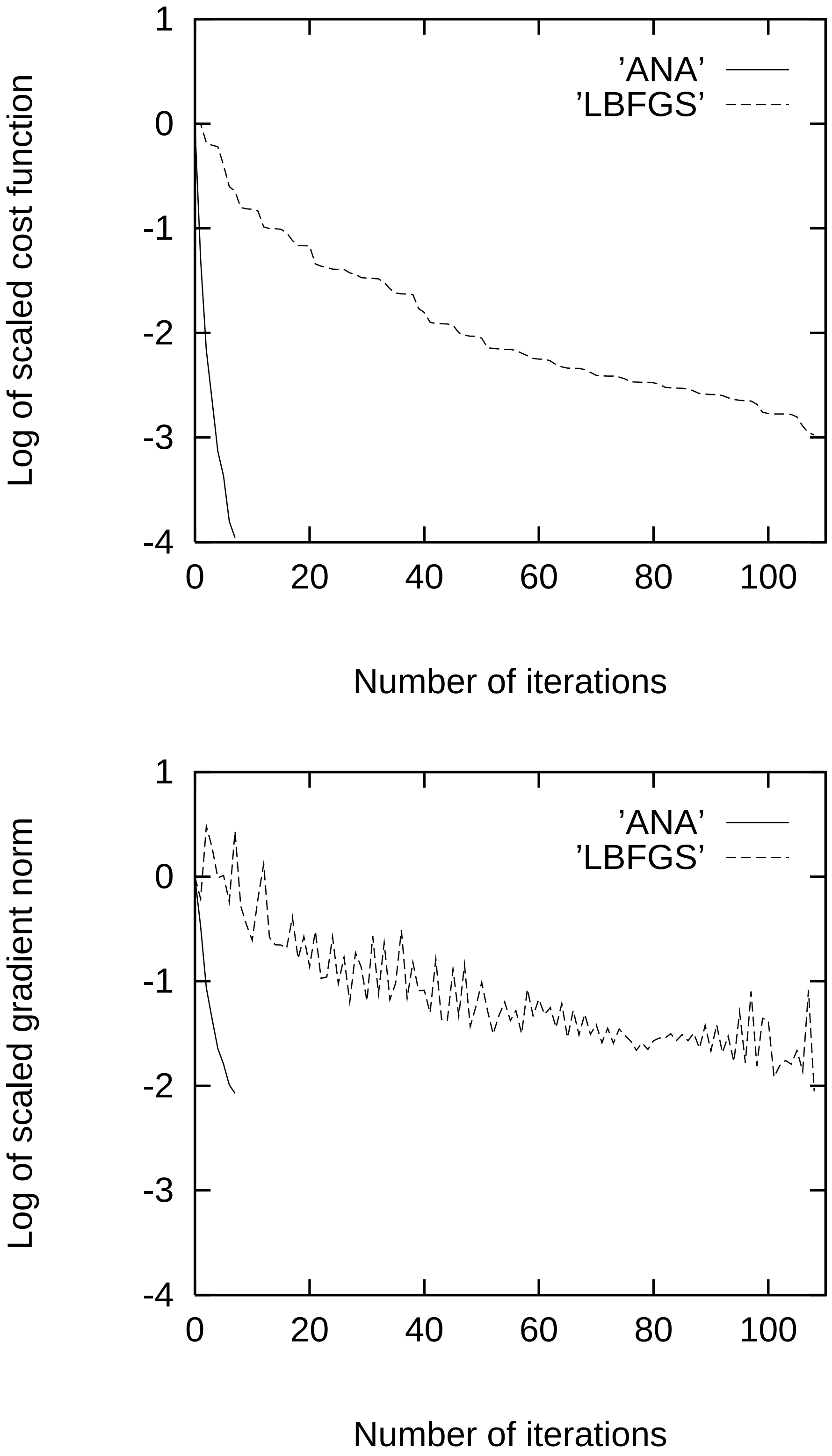
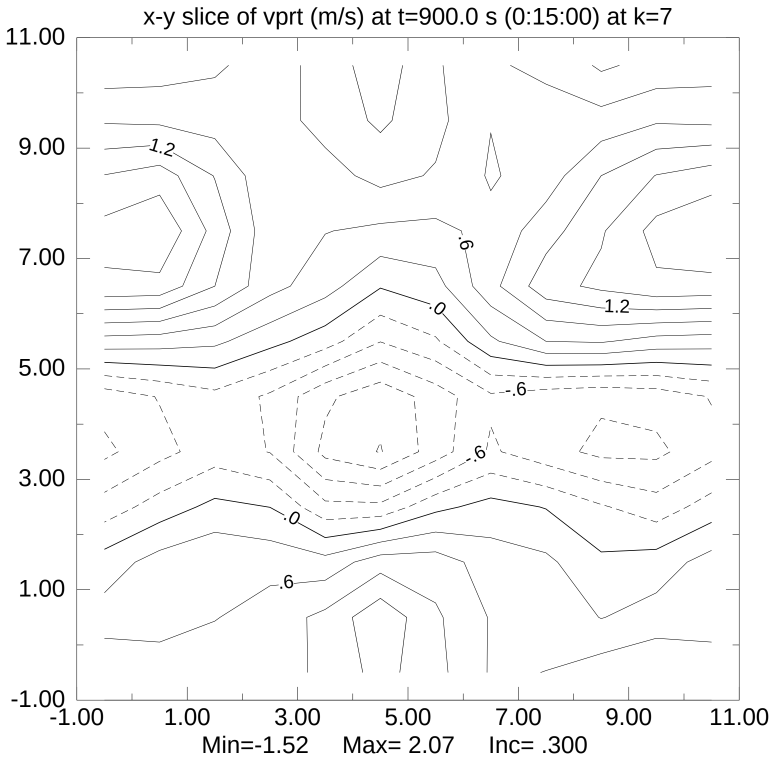


Figure 9: Same as Fig. 4 except they are for experiment 2.



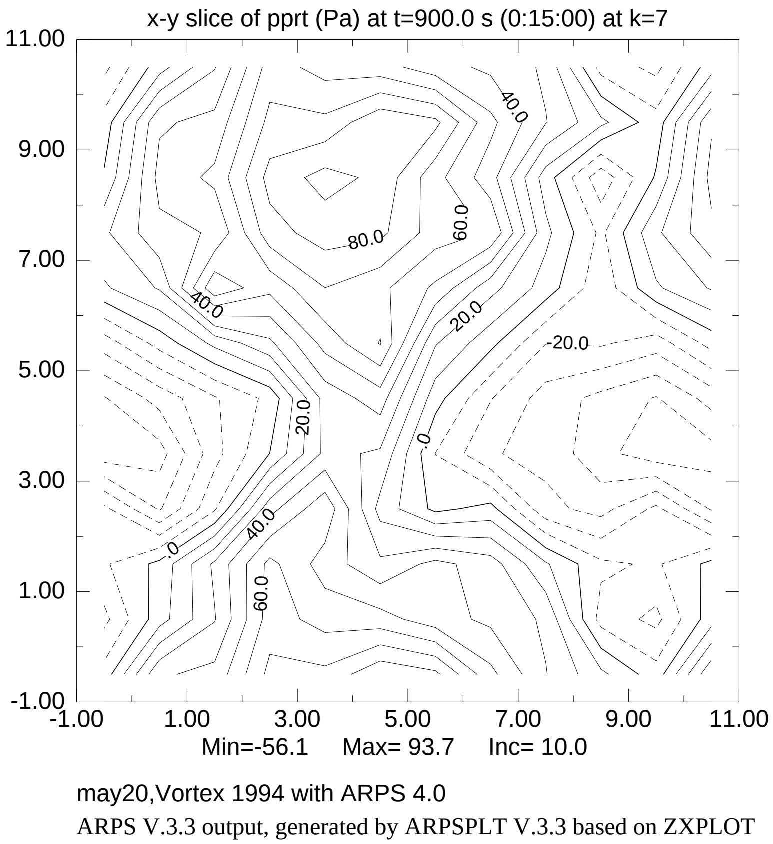


Figure 10: Same as Fig. 5 except they are for experiment 2.

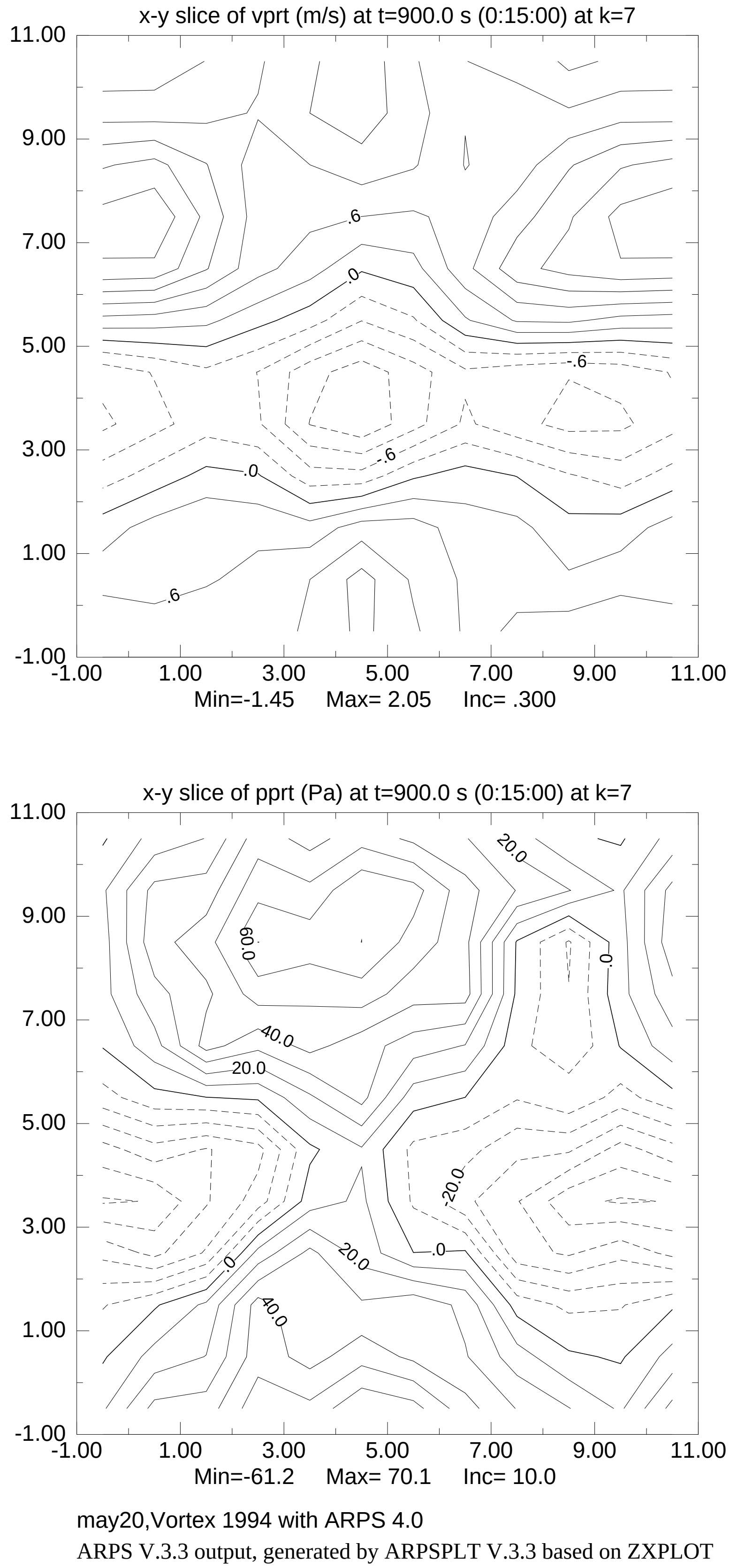


Figure 11: Same as Fig. 6 except they are for experiment 2.

Table 1: RMS errors at the initial time between the retrieved u, v, w, θ and p fields and corresponding reference fields normalized by their initial values in experiments 1 and 2, respectively. Iter. denotes number of iterations.

Experiments	Algorithms	RMS in u	v	w	θ	p	Iter.	CPU
1	ANA	.0054	.0045	.0090	.0026	.0463	5	26.847s
1	LBFGS	.1340	.0619	.0760	.0314	.2344	78	431.131s
2	ANA	.0560	.0421	.0973	.0485	.2076	7	36.556s
2	LBFGS	.1125	.0644	.1133	.0530	.2451	108	568.006s

Table 2: Correlation coefficients of anomaly at the initial time for u, v, w, θ and p between the retrieved and corresponding reference fields for experiments 1 and 2, respectively. Corr. denotes correlation coefficient.

Experiments	Algorithms	Corr. in u	v	w	θ	p
1	ANA	.99998	.99999	.99996	1.	.99892
1	LBFGS	.99292	.99792	.99756	.99966	.97411
2	ANA	.99825	.99903	.99529	.99876	.97966
2	LBFGS	.99406	.99776	.99413	.99857	.97256

Table 3: Convergence analysis information for adjoint Newton algorithm in experiment 1

k	J_k	ξ_k	$\xi_k/\xi_{k+1} = M$
1	4.847×10^{-2}	.952	22.50
2	6.183×10^{-3}	4.229×10^{-2}	8.438
3	1.170×10^{-3}	5.012×10^{-3}	6.176
4	3.590×10^{-4}	8.116×10^{-4}	3.847
5	1.165×10^{-4}	2.424×10^{-4}	3.781
14	1.060×10^{-7}	8.804×10^{-8}	1.809
15	5.734×10^{-8}	4.867×10^{-8}	2.069
16	3.382×10^{-8}	2.352×10^{-8}	1.585
17	1.897×10^{-8}	1.484×10^{-8}	2.001
18	1.155×10^{-8}	7.417×10^{-9}	1.460