

4. Assessment of model input uncertainties with 4D-VAR chemical data assimilation (Task 3)

In this section we report on the new CMAQ/4D-VAR developments performed at Virginia Tech, University of Houston, and EPA as Task 3 of the H98 project. This section presents a brief overview of these developments.

A new CMAQ/4D-VAR system has been implemented. This system is based on CMAQ version 4.5. It includes the CB4 and SAPRC99 chemical mechanisms. Control variables for the optimization include initial conditions, boundary conditions and boundary scaling factors, and emission scaling factors. The system runs in parallel (the forward model, the adjoint model, and the observation operators are all parallelized).

The “factor of 2” problem has been explained, but not fully resolved. The “factor of 2” problem refers to the fact that, in some point to point comparisons, sensitivities obtained by the adjoint approach are almost twice as large as the sensitivities obtained by the finite difference approach. The differences appear to be due to the nonlinearity of the PPM method used to solve advection. A possible solution is to use discrete advection adjoints instead of continuous advection adjoints.

As part of the solution we have developed discrete advection adjoints for both the horizontal and vertical advection processes. These adjoints have been carefully tested in both sensitivity analysis and the 4D-VAR data assimilation setting. The counter-intuitive conclusion is that:

- The gradients given by the continuous advection adjoint have larger discrepancies from the finite difference sensitivities, but lead to a very fast convergence of the 4D-VAR data assimilation.

The gradients given by the discrete advection adjoint show an improved agreement with the finite difference sensitivities in point-to-point tests. However, they lead to a much slower convergence of the optimization iterations during 4D-VAR data assimilation.

Sensitivities with respect to boundary conditions have been derived in the adjoint framework. They include sensitivity of a cost functional with respect to a boundary perturbation, and with respect to a scaling factor applied to all the boundary condition values in a given time window. These sensitivities have been tested extensively in a 1-D framework, and have been also implemented and tested in CMAQ/4D-VAR.

To validate the new CMAQ/ADJ system data assimilation tests have been carried out for a chosen domain that covers parts of Southeast Texas and West Louisiana, and for the period August 30-September 1 2006. Assimilation of initial ozone conditions greatly improves the agreement between model predictions and observations. The assimilation of boundary scaling factors helps the predictions to a lesser degree.

4.1 Development of CMAQ/4D-Var modeling codes

One goal of the current project has been to incorporate the 4D-VAR (four dimensional variational) data assimilation method into CMAQ, test the resulting model and study actual applications. Initial implementation has been limited to the simulation of photochemical air quality emphasizing the transport processes in CMAQ. Additional modification of aqueous-phase and aerosol process codes is needed, but deferred for future work.

4.1.1 Overall Structure of CMAQ/4D-VAR Software

The 4D-VAR method that has been implemented in CMAQ invokes an optimization algorithm (L-BFGS-B version 2.1: *Byrd et al.*, 1995 and *Zhu et al.*, 1997) to iteratively adjust selected inputs to CMAQ, called control variable here, with the goal of minimizing a “cost” function. For each step of the iteration the optimization algorithm requires the value of the cost function and the gradient of the cost function with respect to the control variables at the current estimate for the control variables.

For data assimilation, the cost function is usually the sum of two parts: an observation term, which measures model prediction errors, and a background term, which is a penalty for deviation from a priori estimates of the control variables

$$J = \frac{1}{2} [y - h(c)]^T R^{-1} [y - h(c)] + \frac{1}{2} [y - 1]^T L^{-1} [y - 1] + \frac{1}{2} [\varepsilon - 1]^T E^{-1} [\varepsilon - 1] + \frac{1}{2} [C_0 - C_b]^T B^{-1} [C_0 - C_b]$$

R, L, E, and B are error covariance matrices for observations, lateral boundary concentration scaling factors y , emission scaling factors ε , *a priori* (background) initial states C_b in discrete spaces, respectively. At the current stage, they are assumed to be diagonal matrices. h is a projection operator, calculating the observation vector $y=y(x,t)$ from the model concentration space $C(x,t)$. In the current study the initial concentrations $C_b=C(x,t=0)$, emission scaling factors ε , or lateral boundary concentration scaling factors y can be chosen as the control parameters.

In other words, the observation term is a measure of the distance between model calculated values of selected trace species and the corresponding observed values of those species, while the background term is a measure of the distance between model calculated values of control variables and a priori, or background, estimates of those control variables.

The adjoint model provides the most efficient way to calculate the gradients of the cost functional J with respect to a large number of control parameters chosen.

The base version of CMAQ for CMAQ/4D-VAR was version 4.5 with

- PPM (Piecewise Parabolic Method) used for horizontal and vertical advection
- Eddy diffusion (K-theory) used for horizontal and vertical diffusion
- CB4 or SAPRC99 gas chemistry mechanism

The first modification of CMAQ was to replace the chemistry solvers with solvers generated by the Kinetic PreProcessor (KPP) [*Sandu et al.*, 2003]. The solvers generated by KPP support both the forward solution and the adjoint solution that is used in the gradient computation needed by the optimization algorithm.

For the H98 project development started from the version of CMAQ/ADJOINT completed for HARC Project H59. This is described in *Sandu and Singh*, [2007], and *Singh et al.*, [2007]. It included

sensitivities to initial conditions and emissions and simple 4D-VAR data assimilation with fake observations (generated by CMAQ) and just initial conditions as control variables. Also see *Hakami et al.*, [2007], for a description of a related implementation of adjoint sensitivities in CMAQ, which was based on the YAMO option for horizontal and vertical advection.

The sensitivity of a cost function to a control variable is just the partial derivative of the cost function with respect to the control variable, i.e., a component of the gradient vector of the cost function. The adjoint method for calculating sensitivities is a receptor based, or backward, method that calculates all the sensitivities of the cost function at the same time and is computationally efficient when the cost function depends on time dependent data. Thus, CMAQ/ADJOINT, which focuses on sensitivities, i.e., the gradient computation, is a big step towards development of CMAQ/4D-VAR. Once adjoint sensitivities for a cost function, i.e., the gradient of the cost function with respect to the control variables, can be calculated, the remaining steps needed to implement CMAQ/4D-VAR are

- 1) Set up the calculation of the value of the cost function and its gradient to take varying estimates of the control variables as inputs
- 2) Make these calculations subroutines that can be called during an iteration of the optimization algorithm

The following is an outline of the top level driver for CMAQ/4D-VAR:

- Standard CMAQ setup & initialization
- Setup & initialization of additional data needed for using varying estimates of the control variables as inputs (see the module FOURDVAR_MOD.F) including
 - List of control species names
 - Current values of control variables
 - Gradient of cost function with respect to the control variables
 - Background values of control variables
- Get the synchronization time step
- Setup & initialization for L-BFGS-B
- Create necessary adjoint files and open for update
- Save the base initial conditions and "background" values
- Set the initial values of the control variables
- Optimization loop
 - Call the L-BFGS-B code to adjust the control variables and determine whether a new evaluation of the cost function and its gradient is needed
 - Either exit the loop or continue
 - Restore the base initial conditions
 - Transfer the control variables from L-BFGS-B arrays to CMAQ arrays
 - Perform a forward CMAQ simulation to evaluate the evolution in time of the state of the system with the new control variables and save necessary checkpoint states
 - Do a backward (adjoint) solution to calculate the value of the cost function and compute the gradient of the cost function with respect to the control variables
 - Transfer the gradient of the cost function from CMAQ arrays to L-BFGS-B arrays
 - Return to the top of the loop to call L-BFGS-B again

4.1.1.1 Time Steps

Currently CMAQ/4D-VAR uses a user-input synchronization time step that is constant for a run. This is necessary because checkpoints are written every synchronization step and the IO/API requires a constant output time step. The synchronization time step is set by the user in the run script using the environment variable CTM_SYNC_STEP and read into CMAQ by syncstep.F.

However, the horizontal advection time step is recalculated each output time step so that the Courant condition is satisfied and may vary with the model layer. Other sub-synchronization time steps are calculated as in standard CMAQ.

4.1.1.2 Multiday Simulations

It is assumed that input files are for one Julian date, spanning a 24 hour period starting at midnight UTC. To handle simulations that span more than one date, a subroutine named SET_INPUT_ENV has been added that is called from the drivers for the forward and backward solvers whenever the date changes. SET_INPUT_ENV resets the values of the input environment variables to point to the input files for the current day. This procedure assumes that, for each input file type, the run script has set up a sequence of environment variables whose values are paths to the input files for the days spanned by the simulation. For example, if the simulation spans NN days, then for the MET_CRO_3D files, a sequence of environment variables MET_CRO_3D_01, ..., MET_CRO_3D_NN must have been defined in the run script.

Of course, there are a number of possible alternatives to this particular strategy for handling multiday simulations, including having a preprocessor concatenate the input files so that a single file of each input type spans the time period of the simulation. The concatenation strategy was rejected primarily because it was not clear how to concatenate the photolysis data files.

On the other hand, when the control variables are boundary condition time series, it is desirable to make a copy of the base boundary condition input files so that the control variables can be updated without overwriting the original inputs and the changing boundary condition file can be read by the standard CMAQ RDBCON subroutine. While the initial copy is being written, the boundary condition files for the different species classes are combined into one file and concatenated in time, if the simulation spans more than one date.

The output files for multiday simulations all cover the full period of the simulation. This can create some very large output files, in particular, the chemistry checkpoint file, but the checkpoint files can be deleted as soon as the simulation is finished.

4.1.1.3 Forward Solver

A CMAQ/4D-VAR forward solution is essentially just a standard CMAQ run, except that checkpoints for science processes required by the backward solver are written to file every synchronization time. The types of checkpoints that need to be saved depend on the properties of the solver used for the process and the type of adjoint solver that is used. For example, checkpoints are needed for X, Y and Z advection when the discrete adjoint is used, but not when the continuous adjoint is used. Checkpoints not needed for horizontal or vertical diffusion, even though a discrete adjoint is used, because the operators for those processes are linear. When a checkpoint file is needed for a process, it is written at the beginning of the top level driver for the solver for that process. An observation checkpoint file is written between synchronization steps to hold the simulated counterparts of observations.

4.1.1.4 Backward Solver

The CMAQ/4D-VAR backward (adjoint) solution updates the value and gradient of the cost function for the current estimate of the optimal control variables. It looks a lot like simply reversing everything in the forward solver, including time, with process solvers replaced by their adjoints.

When the continuous adjoint for advection is chosen and the PPM solver for advection is used in the forward solution, it is assumed that together the forward solution sequence COUPLE, [XYZ]ADV, ADJADV, and DECOUPLE solves the 3D equation

$$\frac{\partial q}{\partial t} = -V \cdot \nabla q$$

where q is mixing ratio, which says that mixing ratio is constant along trajectories determined by the wind field. The adjoint problem corresponding to this is the 3D equation

$$\frac{\partial \lambda}{\partial t} = -\nabla \cdot (V\lambda)$$

which can again be solved, using operator splitting, by the PPM algorithm. Therefore, checkpoints, COUPLE / DECOUPLE and ADJADV are not needed in the backward solution sequence.

4.1.1.5 Control Variables

Currently the available control variables are

- Initial conditions – the concentration of each selected species in every cell inside the model's domain at the starting time for a simulation
- Boundary condition scaling factors – a multiplicative factor, constant in time, for each selected species in every boundary cell around the model's domain
- Boundary condition time series – a time series, in output time step increments, of the concentration of each selected species in every boundary cell
- Emissions scaling factors – a multiplicative factor, constant in time, for the emission rate of each selected species in every cell inside the model's domain

In the future, it would be possible to add many other CMAQ inputs to the list of available control variables, including

- Meteorological inputs such as wind velocity, temperature and air density
- Model parameters such as diffusion coefficients, mixing heights, chemical reaction rates

Although conceptually the same thing is done for each type of control variable, the details of setting up the calculation of the value of the cost function and its gradient to take varying estimates of the control variables as inputs are highly dependent on the type of control.

- **Initial condition controls:**

Before the forward driver is called during an optimization iteration the control species in the base initial conditions are over-written with the current estimate of the control variables; after the backward solution, the gradient with respect to the control species is extracted from the full gradient with respect to all species, called LGRID, which is always calculated regardless of the choice of control variables

- **Boundary condition scaling factor controls:**

The boundary condition scaling factors, BC_SF, and the gradient with respect to them, LGRID_BC_SF, are passed via the module FOURDVAR_MOD. In XADV and YADV, the drivers for the forward x and y advection solvers, BC_SF simply scales the base boundary conditions after they have been read and processed to convert them from mixing ratios to mass concentrations. In XADV_CAD and YADV_CAD, the drivers for the continuous adjoint x and y advection solvers, after advecting LGRID backwards in time into the boundary array LGRID_BC_SF, it is necessary to multiply by the base boundary condition in mixing ratio units.

- **Boundary condition time series controls:**

In standard CMAQ the boundary conditions may be input in four different files, one for each species class (gas chemistry, aerosols, non-reactive and tracer) that CMAQ supports, although the default run script assumes that there is just one file that contains all the species classes.

In order to be able to overwrite the base boundary condition control variables with updated estimates, the subroutine INIT_BNDY_COND

- copies the base boundary condition files for the different species classes into one file
- concatenates the combined files in time, if the run spans more than one day
- initializes the global boundary condition control variables
- saves the initial values of the boundary condition control variables as "background" values, in the array BG_BC_TS
- resets environment variables containing the paths to the boundary condition files to the path to the combined and concatenated copy, whose formal name is ADJ_BC_TS

The subroutine UPDATE_BNDY_COND updates the control variables in the ADJ_BC_TS file and updates the local CNL_BC_TS arrays. The arrays CNL_BC_TS, BG_BC_TS and the gradient, LGRID_BC_TS, are passed via the module FOURDVAR_MOD.

Since the copy of the boundary condition file always holds the current estimates of the boundary conditions and the environment variables for boundary condition input files point to the copy, the forward solvers for horizontal advection, XADV and YADV, can be used without any modification. In XADV_CAD and YADV_CAD, after advecting LGRID backwards in time into the boundary, the synchronization step value is distributed back to the time series array LGRID_BC_SF, which is in output time step increments, by reversing the linear interpolation in time that is done when boundary conditions are used in the forward direction.

- **Emissions scaling factors:**

The treatment of these scaling factors is fairly similar to the treatment of boundary condition scaling factors, except that the process solvers involved are the forward and discrete adjoint solvers for vertical diffusion. Other than cosmetic changes, the only change made to the CMAQ/ADJOINT code was to use the multiplicative emissions scaling factor in the forward vertical diffusion solver – which is unnecessary if you only want sensitivities, because the initial estimate of scaling factors is one.

4.1.2 Parallelization

Parallelization of standard CMAQ v4.5 is based on the Message Passing Interface (MPI) and horizontal domain decomposition. For I/O, each processor reads all data for the full domain from files and extracts data for its local domain; only the root processor writes to files, so the other processors must pass data for their local domains to the root processor which gathers the local data into global arrays before writing. While the parallelization of standard CMAQ supports reading from boundary files, it does not support writing to boundary files.

Since the optimization package, L-BFGS-B version 2.1, is serial code and is used as a “black box”, it must be called only from the root processor in parallelized CMAQ/4D-VAR. Each call to L-BFGS-B requires

- the current value of the cost function
- the array of current values of the control variables
- the gradient of the cost function at the current values of the control variables (an array of the same size as the array of control variables)
-

The calls to L-BFGS-B return an updated estimate of the optimal values of the control variables.

In parallel mode, for L-BFGS-B these must all be global values for the whole domain, while for the forward and backward runs of CMAQ and CMAQ/ADJOINT that evaluate the cost function and its gradient they must be localized to processor sub-domains. To perform the necessary communication between the root processor and the local processors some new subroutines were developed:

- GATHER_GRDD_4D and GATHER_BNDY_3D use MPI functions to gather gridded 4D arrays and 3D boundary arrays from local processors into a global array on the root processor
- SCATTER_GRDD_4D and SCATTER_BNDY_3D use MPI_BCAST to broadcast global gridded 4D arrays and 3D boundary arrays from the root processor to the local processors which then extract the data needed for the local domain

These tasks are similar to what is currently done for parallel I/O in standard CMAQ, but they do not involve external files.

Depending on the type of control variable, the global array of initial estimates of the control variables may have to be gathered from individual sub-processors:

- For initial condition control variables, this is done using GATHER_GRDD_4D
- For boundary condition time series control variables this is done by the subroutine INIT_BNDY_COND

When the control variables are boundary condition or emissions scaling factors, the initial estimate of the scaling factor is always 1.0, so nothing special needs to be done.

Inside the optimization loop

- after L-BFGS-B updates the global array of control variables, the global array has to be scattered to local arrays
 - when the control variables are boundary condition time series this is done inside the subroutine UPDATE_BNDY_COND

- after the backward (adjoint) solution, which computes the cost function and its gradient on local domains, the local values of the cost function have to be summed and the local gradients have to be gathered before calling L-BFGS-B again

In parallel mode, in order to use the IO/API subroutine WRITE3 on the root processor to write a global array that has already been gathered by GATHER_GRDD_4D or GATHER_BNDY_3D, it was necessary to introduce a wrapper subroutine called BWRITE3 which is defined in the file bwrite3.f (note the lower case f). The reason for this is that parallel mode build scripts set a preprocessor (which is invoked during a make for source files with the extension upper case F) directive that replaces WRITE3 with PWRITE3.

Due to the use of horizontal domain decomposition, the horizontal transport (diffusion and advection) routines require each processor to receive “ghost” data from its neighbors.

A discrete adjoint is used for horizontal diffusion. The original serial version was created using the automatic differentiation tool TAMC [see *Giering*, 1999]. However, the code that TAMC generated did not work properly with horizontal domain decomposition. Fortunately, starting from a cleaned-up version of the forward horizontal diffusion solver, HDIFF, it was easy to manually write an adjoint solver (because the diffusion operator is self-adjoint) that did work properly in parallel mode.

For the continuous adjoint of horizontal advection no changes were actually required for the forward solver routines XADVPPM, YADVPPM and HPPM, but cleaned up versions were already available from another project and using those versions made parallelizing the continuous adjoint counterparts XADV_CAD and YADV_CAD somewhat easier (HPPM can be used without change). The discrete adjoint of horizontal advection has not been successfully parallelized yet.

4.1.3 Construction of advection adjoints and analysis of their properties

4.1.3.1 The “factor of 2” problem

The “factor of 2” problem refers to the fact that the sensitivities obtained by adjoint approach are almost twice as large as the sensitivities obtained by the finite difference approach for several specific point receptor and point source pairs.

Our first task in H98 was to elucidate the source of these differences, and to find an appropriate solution.

4.1.3.2 Experiments revealing the “factor of 2” problem

One of the examples where this problem manifests itself is the following point –to –point test case reported by Dr. Peter Percell. The settings are as follows:

Source: OLE in column 24, row 12 and layer 1 at 13:00 UTC (7:00 CST)

Receptor: O₃ in column 15, row 13 and layer 1 at 21:00 UTC (15:00 CST)

All the input files can be found at: http://geosun2.geosc.uh.edu/web/ppercell/fd_sns/

The results reported by Dr. Percell are:

Absolute ADJ_SNS (ppbV/ppmV): 10.08652

BF_SNS (ppbV/ppmV): 5.08575

To confirm factor of 2 problem, we have repeated the original test and have also considered additional test cases with different settings. The results of these tests are summarized next.

- (1) The original test
Independent results at VT using the same input files and test settings confirm Peter's findings:
adjoint sensitivity (ppbV/ppmV): 9.93048
finite-difference (ppbV/ppmV): 5.35007
- (2) Additional test with different source species.
We have changed the source species from O₃ to NO₂ .
Result:
adjoint sensitivity (ppbV/ppmV): -4.7202925
finite-difference (ppbV/ppmV): -2.3772469

Additional tests, not reported here, have shown similar results. Note that the relative discrepancy between the adjoint sensitivities and the finite difference sensitivities depends strongly on the choice of the source and receptor pairs. For the "worst case" scenarios the discrepancy is at the "factor of 2" level.

4.1.3.3 Identifying the cause of the "factor of 2" problem

Using the same test case for the factor of 2 problem, we find that the sensitivities obtained by the adjoint approach and finite difference approach are consistent with each other after switching off vertical advection (zadv). The results are shown below for two different source species.

- (1) Source species: OLE
adjoint sensitivity (ppbV/ppmV): 6.2011527
finite-difference (ppbV/ppmV): 7.0461372
- (2) Source species: NO₂
adjoint sensitivity (ppbV/ppmV): -3.1010953
finite-difference (ppbV/ppmV): -3.178566

Despite the better agreement of these sensitivities, we note that there still is a 11% relative difference in the OLE test case, which indicates that there might be an additional source of error besides the vertical advection adjoint. It turns out that these remaining differences are caused by relative inaccuracies in the horizontal advection continuous adjoint.

We have performed a finite difference test for horizontal advection only (all other processes turned off), and compare the sensitivities obtained by continuous adjoint and finite difference method for the same point-to-point test cases. For Peter Percell's test we find that there is a 7.8% relative difference between the sensitivities as shown in Table 4.1.

Table 4.1 Finite difference test for horizontal advection only, Peter Percell's test case.

| Test Case | $\delta\psi$ Finite Difference | $\delta\psi$ Continuous Adjoint | Relative Error |
|---|--------------------------------|---------------------------------|----------------|
| Only horizontal advection process (with continuous adjoint) | 84.45 | 105.42 | 7.8 % |

The experiments above reveal that the differences between finite difference sensitivity and adjoint sensitivity come from the continuous adjoint of the advection process. Most of the difference is due to the continuous adjoint vertical advection; a smaller, but still sizeable part of the difference is due to the continuous adjoint of the horizontal advection.

4.1.3.4 Explanation of the “factor of 2” problem

Advection is a linear process and its adjoint is also a linear process. In short, the forward advection moves a plume forward along the Lagrangian trajectories (characteristics) defined by the three-dimensional wind field. Assuming the wind field (u,v,w) is divergence free, the adjoint of the advection process moves a plume backward in time along the same characteristics; the adjoint of advection is itself an advection equation, which propagates information backwards along the characteristics.

Based on the above considerations one can regard the adjoint of the advection as an advection equation driven by a wind field with reversed signs $(-u,-v,-w)$. This observation is the basis of the “continuous adjoint” implementation of the adjoint advection in CMAQ. The advection in the forward model is solved by the “piecewise parabolic method” (PPM), which transports concentrations using the forward wind field (u,v,w) . The same PPM numerical routine is called in the adjoint model and performs the transport of the adjoint variables using the reverse wind field $(-u,-v,-w)$.

The crucial element to consider is that PPM is a nonlinear numerical discretization applied to solve the linear advection PDE. This nonlinear numerical scheme is applied again in the adjoint model. Consider a very narrow (delta-shaped) plume and let it evolve under advection only forward in time. If advection is solved exactly, the plume is moved to another location (and possibly deformed in a conservative way). Now transport the final plume for the same time interval using the reversed wind field. If this reverse (adjoint) advection is solved exactly, the plume will come back to its initial position and its initial shape (the delta-shape will not be modified by the forward and reverse advection; the plume just travels forward and backwards along the same characteristic). Because of the linearity of the advection process the same considerations apply to perturbations (consider the plume above to be a perturbation of the concentration field; the forward evolution of the perturbation and its backward evolution lead to the same results).

Next, instead of solving the advection equation exactly, consider a numerical solution using the PPM method. The delta-shaped plume is transported forward using the forward wind fields and the PPM solution; the final plume will show some spread due to the numerical diffusion built into PPM. The final plume is then transported using the reverse wind fields and the PPM solution again. Because of the nonlinearity of the PPM scheme the numerical plume will show a spread and will therefore lose the original delta-shape. The plume will not be exactly at the same location as the original plume either. These differences are the ones observed between the adjoint and the finite difference gradients and lead to the “factor of 2” problem.

Consider now the case where we have a base concentration profile and a perturbed concentration profile at the initial time. We refer to the difference of the two profiles as the “perturbation field”. Both the base and the perturbed profiles are evolved forward in time; their difference gives the evolved perturbation field. A linearized model can evolve the perturbation field directly from the initial time to the final time. Note that if the advection equation is solved exactly the two results agree: the difference of two fields at final time is the same as their difference at the initial time, evolved forward by exact advection. This property is due to the linearity of the exact advection. If a nonlinear numerical scheme like PPM is used, however, the difference of the two concentration fields at the final time can be very different from the perturbation at the initial time, evolved by PPM. Thus, the finite difference approach and the linearized model (continuous adjoint) approach can lead to significant differences.

These considerations are also supported by the findings of Dr. Tianfeng Chai. In his report “Investigation of CMAQ-ADJ factor-2 problem” he concludes as well that the reason for the factor of

2 problems is the nonlinear property of PPM scheme. He provides a detailed explanation of nonlinearity of PPM scheme in his report.

We next explain in detail the link between the nonlinearity of the PPM scheme and factor 2 problem.

First, we perform a test similar to Tianfeng’s to show the nonlinearity of PPM. We consider the linear 1D advection PDE: $u_t + a(x)u_x = 0$, where $a(x)$ is the wind velocity. To show the PPM method is not linear, we consider the transport of a delta function and show its profile at initial time and final time (Figure 4.1). Next, we consider a “base” smooth profile and advect it forward in time. We add a delta perturbation to the smooth profile, and transport the perturbed profile forward in time. The differences between the perturbed and the “base” profiles are shown in Figure 4.2. The transported delta function (Figure 4.1b) and the difference of the base and perturbed profiles (Figure 4.2b) are very different, which indicates the PPM scheme is nonlinear. (If advection was solved exactly then these two profiles would have been the same due to linearity).

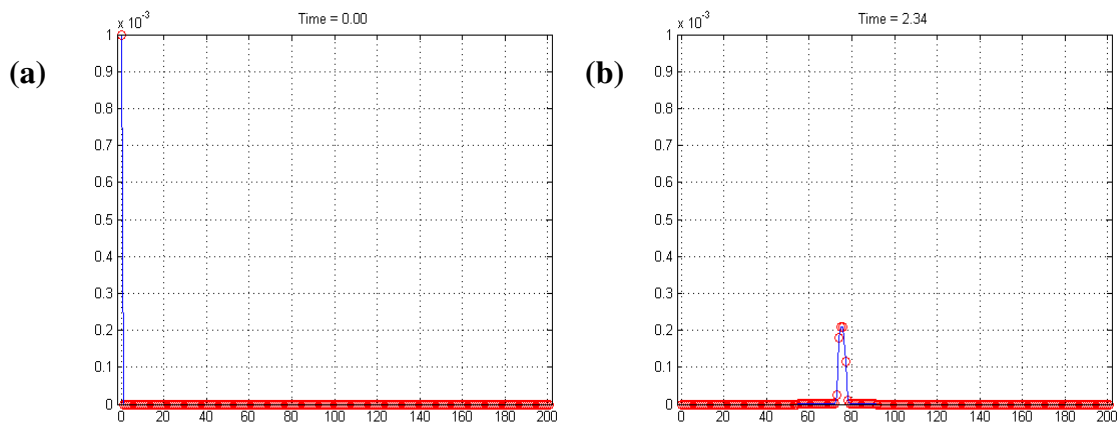


Figure 4.1 Delta function transported in time using PPM to solve a one-dimensional advection equation: (a) at initial time (b) at final time.

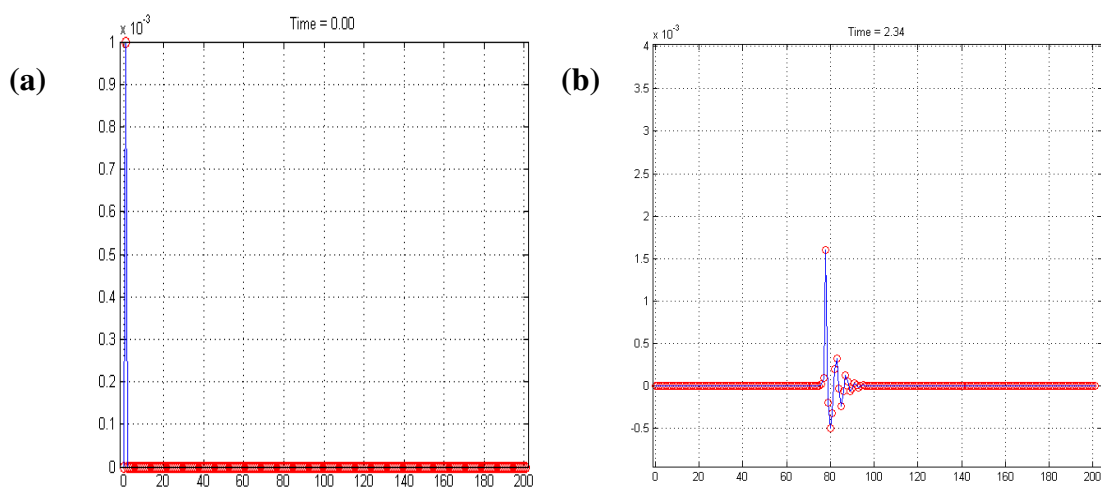


Figure 4.2 Perturbation of the initial condition (difference between perturbed solution and base solution) transported in time along 1D axis: (a) difference at initial time (b) difference at final time.

Our question is how this nonlinearity of the PPM scheme affects adjoint sensitivity and finite difference sensitivity (therefore the validation). To answer this question, we did another validation of the adjoint sensitivity with different receptors. The sensitivity analysis problem is defined as follows.

The cost function/receptor is

$$\psi = \sum_{i=a}^b c(x_i, t_F)$$

where $c(x,t)$ is the solution of the 1D advection PDE: $c_t + ac_x = 0$ with the initial condition $c(x,0) = 1 + \sin(x)$, and boundary condition $c(0,t) = c(2\pi,t) = 1$, $x \in [0, 2\pi]$. i is the computational grid index. When solving the 1d advection PDE, we chose $N=200$ inner grids (so $x_i = i * dx = i * \frac{2\pi}{201}$) and $a=1$. The source (perturbation location) is $c(x_1,0)$, initial condition at x_1

We want: $\frac{\partial \psi}{\partial c(x_1,0)}$, the sensitivity of the cost function (receptor) w.r.t. the initial condition at the x_1 (source).

Two approaches are used to get $\frac{\partial \psi}{\partial c(x_1,0)}$. One is finite difference approach and the other one is adjoint approach. Table 4.2 shows the comparison between finite difference sensitivity and adjoint sensitivity. For finite difference test, the perturbation of the source is 0.1%.

Table 4.2 Finite Difference Sensitivity vs. Continuous Adjoint Sensitivity with different receptors for the 1D advection equation example.

| Receptor | $\Delta\Psi$ Finite Difference | $\Delta\Psi$ Continuous Adjoint | Relative Error |
|-----------------|--------------------------------|---------------------------------|----------------|
| a = 0 , b = 70 | 0.000000 | 2.6772819E-08 | 0.000 % |
| a = 71 , b = 97 | 9.9945068E-04 | 9.9997257E-04 | 0.000 % |
| a = 98 , b =202 | 0.000000 | 4.7424572E-25 | 0.000 % |
| a = 75 , b = 80 | 1.0094643E-03 | 9.5583033E-04 | -5.587368 % |
| a = 81, b = 86 | -1.2683868E-04 | 1.2357767E-08 | 1026388 % |
| a = 75, b = 85 | 9.4318390E-04 | 9.5822528E-04 | 1.592401 % |

The results in Table 4.2 show that the agreement between the adjoint and finite difference sensitivities depends strongly on the choice of the receptor. This is caused by the effect of nonlinearity of PPM to the finite difference results. To explain this we revisit Figures 4.1 and 4.2. Figure 4.2 shows the difference of the perturbed and base solutions and corresponds to the finite difference sensitivity case in Table 4.1. Figure 4.1 shows what the difference of two solutions would be if the PPM scheme was linear, and corresponds to the adjoint (linearized sensitivity) case of Table 4.1. In Figure 4.2, there are wiggles resulted from nonlinearity of PPM which do not appear in the linearized solution (Figure 4.1). This implies that if receptors are chosen at grids where wiggles are present, the finite difference sensitivity is not accurate this leads to a big disagreement of finite difference sensitivity and continuous adjoint sensitivity. For example, when we choose receptor as the sum of the solution at grids 81 to 86, the relative difference is very large. However, this big discrepancy is mainly due to the inaccurate finite difference sensitivity. From Figure 4.1 (the linearized case), we see that $\delta\psi$ should be very close to 0 - but in finite difference test the change in the cost function large.

Summary.

The “factor of 2” problem is caused by nonlinearity of PPM scheme. Specifically, the finite difference sensitivity (the difference between a perturbed and a base solution) is a wiggly profile that depends not only on perturbation, but on the base solution as well. The continuous adjoint sensitivity (the PPM scheme applied to propagate the initial perturbation) is a smooth solution due to the monotonic properties of the scheme.

We expect the large differences between the profile difference and the linearized profiles are to lead to differences between the adjoint and the finite difference sensitivities. The magnitude of those differences greatly depends on the choice of the receptor. Good agreement is observed when the receptors include the real change in the solution profile, but they do not include the spurious wiggles resulting from the nonlinearity of the scheme. On the other hand when a pointwise receptor is located on a wiggle the finite difference result and the linearized (adjoint) result can be arbitrarily different; for example, the finite difference can give a negative result while the adjoint sensitivity is positive.

One practical way to overcome this problem is to always choose a large receptor area (e.g., an integral of concentrations over several grid cells). In this case one “integrates” over the wiggles and cancels out their spurious contribution, while capturing the real change in the concentration profile (and therefore, in the cost function).

4.1.3.5 Discrete adjoints of advection

Since continuous adjoint sensitivities may have relatively large differences from the finite difference sensitivities, we have developed discrete adjoint advection routines. The development of discrete adjoints uses TAMC (the Tangent and Adjoint Model Compiler) as well as a considerable hand coding effort. The discrete advection adjoints are now available in CMAQ. The user can choose between the discrete and the continuous adjoints when configuring a CMAQ-ADJ application.

4.1.3.5.1 Continuous adjoint of vertical advection

In order to better understand the behavior of the continuous adjoint of vertical advection, we consider three test cases and compare continuous adjoint sensitivities against finite difference sensitivities. The latter are obtained using central finite differences and a perturbation of 5%.

The test cases differ in their selection of the sources and receptors. Each test case use ozone (species index 4). The test cases are summarized in the Table 4.3. Sources and sinks are chosen both at ground level (vertical layer 1) and in the higher elevations (vertical layers 7 and 10).

Table 4.3 Three test cases for vertical advection. The simulation time is 13:00 UTC to 21:00 UTC on 2000238 and the species is ozone. In a single model run independent tests are carried out in each model column.

| Test Case | Source (at initial time) | Receptor (at final time) |
|-----------|--------------------------|--------------------------|
| 1 | (1:NCOLS,1:NROWS,1,4) | (1:NCOLS,1:NROWS,3,4) |
| 2 | (1:NCOLS,1:NROWS,3,4) | (1:NCOLS,1:NROWS,1,4) |
| 3 | (1:NCOLS,1:NROWS,7,4) | (1:NCOLS,1:NROWS,10,4) |

Since only vertical advection is considered the evolution in each column is independent of the other columns. Thus $NCOLS \times NROWS$ independent point-to-point sensitivity coefficients are obtained in each column using both approaches. To compare continuous adjoint sensitivities against central finite differences we draw scatterplots and present them in Figure 4.3. Each point in the scatterplot corresponds to one sensitivity coefficient in a particular model column. The x-coordinate is the continuous adjoint sensitivity value, while the y-coordinate is the finite difference value for the same sensitivity. If the two coefficients agree well then all the points are aligned with the main diagonal of the first quadrant.

The results in Figure 4.3 show a rather weak correlation between the continuous adjoint and the finite difference values. The weakest correlations are for the test case 1 (source at ground level) and the test case 2 (receptor at ground level). Somewhat better correlations are observed for test case 3 when both the receptor and the source are away from the ground level. Even in this case, however, there are large differences between the continuous adjoint sensitivities and their finite difference counterparts.

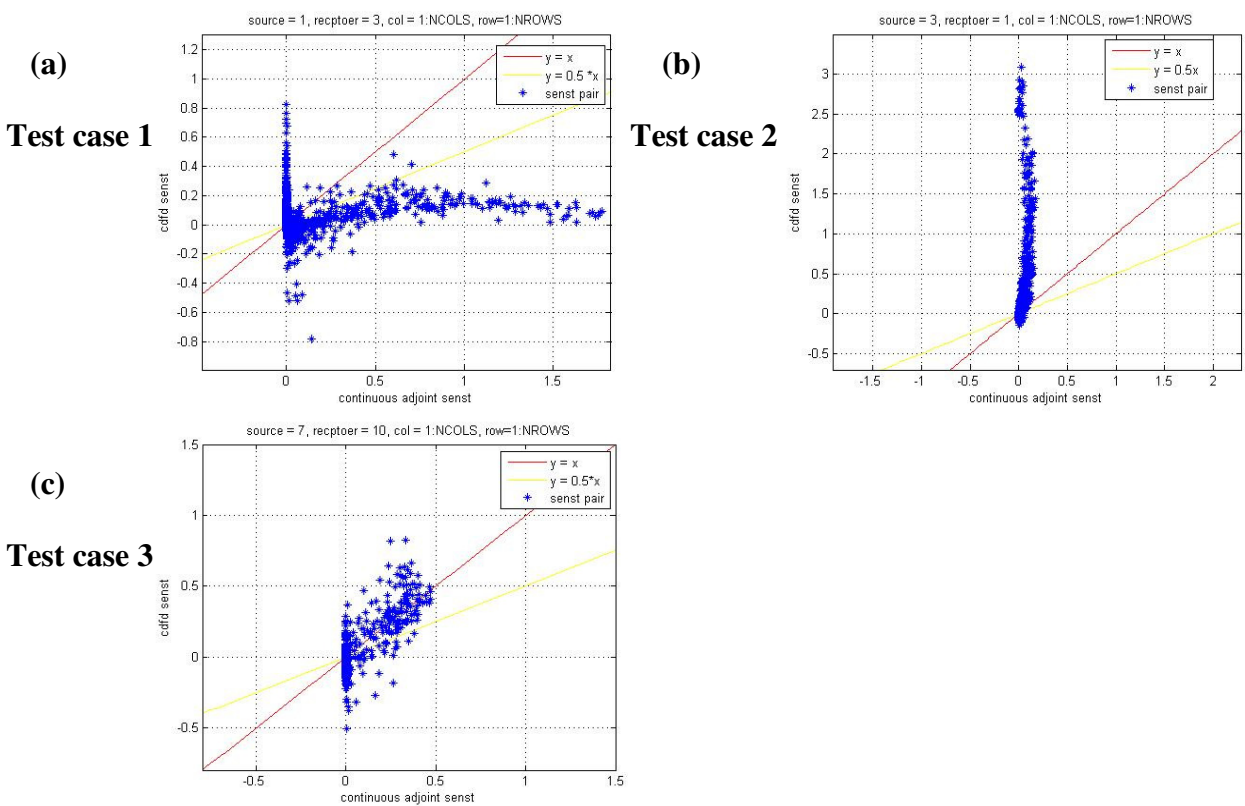


Figure 4.3 Scatter plot of finite difference sensitivity and continuous adjoint sensitivity for vertical advection (vppm). Each point represents one sensitivity coefficient in a different column, with the x coordinate the value obtained by the adjoint method and the y coordinate the value obtained by finite differences. Species: O_3 ; Simulation time: 13:00 UTC to 21:00 UTC on 2000238.

4.1.3.5.2 Discrete adjoint of vertical advection

Because of the large differences between the continuous adjoint sensitivity and finite difference sensitivity, we have decided to develop discrete adjoint for vertical advection. The code development uses the automatic differentiation tool TAMC plus a considerable amount of hand coding.

(1) Point – to –point tests

The tests presented in Table 4.3 have been repeated with the discrete adjoint of the vertical advection. All grids at the source and receptor layers are chosen. Central difference method is used for finite difference sensitivity and the perturbation is 5% of the original source.

The scatterplot results are presented in Figure 4.4. A comparison with Figure 3,3 reveals that the discrete adjoint sensitivities are in much better agreement with finite differences than are the continuous adjoint sensitivities.

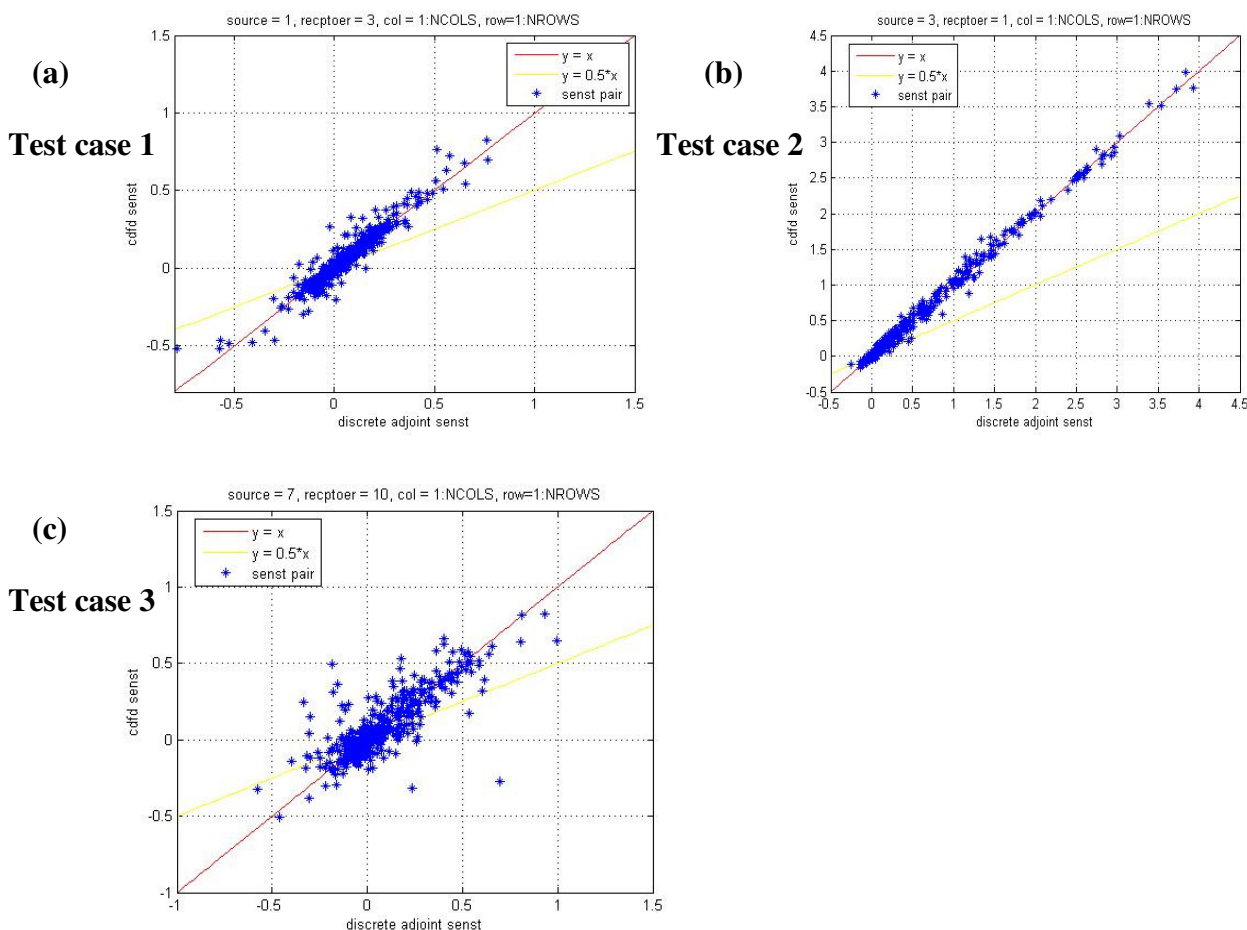


Figure 4.4 Scatter plot of finite difference sensitivity and discrete adjoint sensitivity for vertical advection (vppm); Species: O₃; Simulation time: 13:00 UTC to 21:00 UTC on 2000238.

(2) Peter Percell’s original test

We repeat Peter Percell’s original test but with discrete adjoint of vertical advection replacing the continuous adjoints. As expected, the results are in good agreement with finite differences. In particular there is no “factor of 2” problem when using discrete advection adjoints.

- adjoint sensitivity (ppbV/ppmV): 5.3482050
- finite-difference (ppbV/ppmV): 5.3885686

(3) *Finite Difference Validation*

In order to fully validate the discrete adjoint of vertical advection, we perform a comparison of the change in the cost function ($\delta\psi$) obtained by finite difference approach and adjoint approach.

The change in a cost function, due to a change in initial conditions, is obtained as the difference between the two function values:

$$\partial\psi = \psi_2 - \psi_1, \dots (c_0 \rightarrow \psi_1, c_0 + \delta c_0 \rightarrow \psi_2)$$

To first order the change in cost function is approximated by the dot product of the adjoint variable (at the initial time) and the perturbation of the initial concentration:

$$\partial\psi = \lambda \cdot \delta c_0 = \sum \lambda_i * (\delta c_0)_i = \sum \left(\frac{\partial\psi}{\partial c_0} \right)_i * (\delta c_0)_i$$

The results are shown in Table 4.4.

Table 4.4 Finite difference test for the discrete adjoint of vertical advection. Simulation time 4hrs.

| Test Case | $\delta\psi$ via Finite Difference | $\delta\psi$ via Adjoint | Relative Error |
|---|------------------------------------|--------------------------|----------------|
| only vertical advection process (with discrete adjoint) | 149.91 | 148.31 | -1.08 % |
| only advection process (with discrete vertical advection and continuous adjoint horizontal advection) | 86.29 | 105.42 | 18.15 % |
| only horizontal advection process (with continuous adjoint) | 84.45 | 105.42 | 7.7 % |

We see that only vertical advection test with discrete adjoint shows a very good agreement with finite differences. The agreement deteriorates when we add horizontal advection, which also suggests that the continuous adjoint of horizontal advection may not be accurate. Therefore, we have decided to develop the discrete adjoint of horizontal advection.

4.1.3.5.3 Continuous and discrete adjoints of horizontal advection

Similar to the implementation of discrete adjoint for vertical advection, the discrete adjoint of horizontal advection is constructed using a combination of automatic differentiation (TAMC) and hand coding. We next discuss tests of discrete horizontal advection adjoints.

(1) *X-advection tests*

Because x advection and y advection are equivalent (they are implemented by calling the same hppm routine, and provide similar test results), we only show x advection results in this section. To generate point-to-point sensitivities test, we choose the source as cells at column 25 and all rows and layers for species O₃ at initial time. The receptor is defined to measure ozone in all cells at column 23 for the same rows and layers as source at final time. Since only x advection is involved, each component of sensitivity of the receptor w.r.t. the source is the same as the sensitivity of the single cell at column 23 for a certain row and layer at final time w.r.t. the single cell at column 25 for the same row and layer at initial time. Therefore, we have the NROWS x NLAYS independent pairs of sensitivities (finite difference sensitivity and adjoint sensitivity).

Figure 4.5 presents the scatter plots of adjoint versus finite difference sensitivities. Each point represents the sensitivity in a different column. The x-coordinates are the adjoint values and the y-coordinates are the finite difference values of the same sensitivity coefficient. Figure 4.5a results use the continuous horizontal advection adjoint, while Figure 4.5b presents the discrete horizontal advection results. A comparison of the two panels clearly shows that the discrete adjoint results are in better agreement with the finite difference results than the continuous adjoint values.

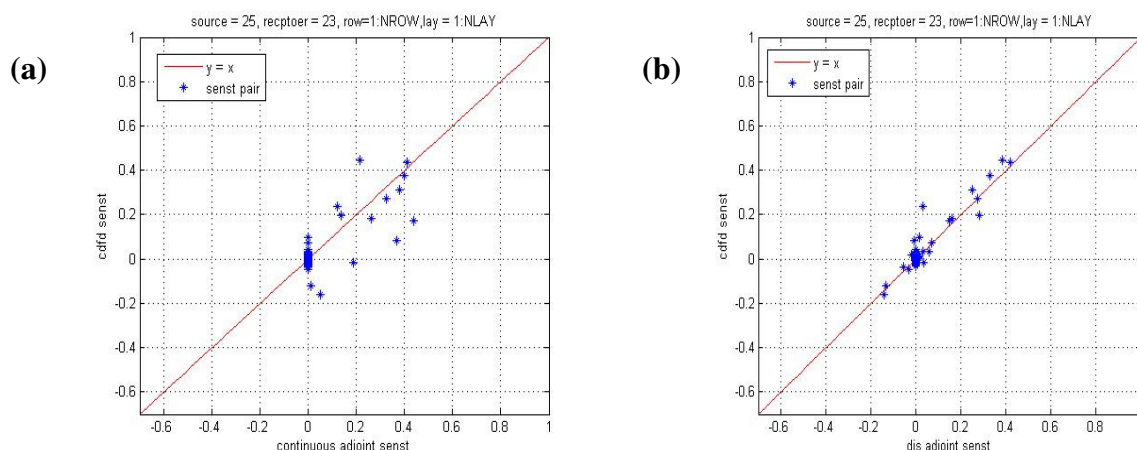


Figure 4.5 Scatter plot of finite difference sensitivity and adjoint sensitivity for horizontal advection (hppm); Species: O₃; Simulation time: 00:00 UTC to 08:00 UTC on 2000238. (a) Continuous adjoint of horizontal advection. (b) Discrete adjoint horizontal advection.

(2) *Finite Difference Validation*

We repeat the finite difference tests for a given cost function using the discrete adjoint of horizontal advection. The results are shown in Table 4.5. A comparison with Table 4.4 reveals that the discrete adjoints for horizontal advection are in much better agreement with finite differences than the continuous adjoint results.

Table 4.5 Finite difference test for discrete adjoint of horizontal advection.

| Test Case | $\delta\psi$ via Finite Difference | $\delta\psi$ via Adjoint | Relative Difference |
|--|------------------------------------|--------------------------|---------------------|
| only horizontal advection process (discrete adjoint for x,y, 4hrs) | 84.45 | 84.61 | 0.19 % |
| only advection(x,y,z) test (discrete adjoint for x,y,z advection, 22hrs) | 170.77 | 175.28 | 2.57 % |
| all science process test (discrete adjoint for x,y,z advection, 22hrs) | 121.57 | 124.25 | 2.16 % |

In conclusion, we have found that the sensitivities given by discrete adjoint of CMAQ advection (implemented by the PPM scheme) are more consistent with finite difference sensitivities than those given by continuous adjoint of advection. This is especially true in the point-to-point case where both the receptor and the source are defined at single grid points.

4.1.4 Impact of advection adjoint type on 4-Var data assimilation

4.1.4.1 Mathematical aspects of 4D-VAR data assimilation

4D-VAR data assimilation is posed as an optimization problem where the best estimate of initial conditions minimizes the following cost function:

$$J(c^0) = \frac{1}{2} \sum_{k=1}^N (c^k - c^{k,obs})^T R_k^{-1} (c^k - c^{k,obs}) + \frac{1}{2} (c^0 - c^b)^T B^{-1} (c^0 - c^b)$$

where c_0 is the initial concentration, c^k is the model prediction at time step k , $c^{k,obs}$ is the observation at time step k , R^k is the observation error covariance matrix, c^b is the background concentration and B is the background error covariance matrix. The cost function measures the misfit between model predictions and observations as well as the misfit between initial conditions and background concentrations.

To minimize the cost function, we apply L-BFGS, a limited memory quasi-Newton method for solving large scale optimization problems. L-BFGS is an iterative method which requires, at each iteration step, the cost function value and its gradient at that point, and returns another point which is a better approximation to the optimal solution. The whole process continues until the convergence criteria are satisfied. In the implementation of CMAQ/4D-VAR we interface L-BFGS with the CMAQ-ADJ. The cost function value and gradient value are obtained through a forward run followed by an adjoint run of CMAQ-ADJ.

The computational flow of the 4D-VAR optimization proceeds as follows:

- (1) Start the iteration with $x_0 = c_p^0 = c_0 + \delta c_0$
- (2) For each iteration $i = 0, \dots, k$
 - a. Using x_i as initial condition, perform a forward run and adjoint run to get the cost function f_i and gradient of the cost function g_i at x_i .
 - b. Provide (f_i, g_i) to L-BFGS to get the next iteration start point x_{i+1} . L-BFGS $(f_i, g_i, x_i) \rightarrow x_{i+1}$
 - c. Stop the iteration when a certain termination criterion is met (e.g., convergence, or the max iteration number has been exceeded).

4.1.4.2 Numerical investigation of 4D-VAR with various adjoint gradients

In this section, we compare the performance of the continuous adjoint and discrete adjoint in the context of 4D-VAR data assimilation.

We consider a 4D-VAR assimilation problem with synthetic observations in the “twin experiment” framework. A “reference run” of the model provides the “true state” and is used to obtain the synthetic observations. The initial conditions are then perturbed to obtain a “best initial guess”. The optimization starts with the perturbed initial conditions and, using the information contained in the observations, is attempting to recover the reference initial field.

The setting of the twin experiment problem is explained next.

(1) Observations

The computation domain for our simulation has 38 columns, 38 rows and 6 layers in total. We choose $10*10*3$ grid cells as observation grids and choose O_3 as the species that we want to retrieve. A reference forward run is performed, starting from the reference initial condition. During this run the ozone concentrations at all the observation grid points are saved. They are the synthetic observation values.

Table 4.6 The observation grid consists of a subset of the model grid points.

| Columns | Rows | Layers | Species |
|----------------------------|----------------------------|--------|-------------|
| 1,5,9,13,17,22,26,30,34,38 | 1,5,9,13,17,22,26,30,34,38 | 1,3,5 | 4 (O_3) |

(2) Observation error covariance matrix and background error covariance matrix

Both covariance matrices are diagonal. The entries are the error variances at the corresponding grid points. These variances correspond to standard deviations equal to 10% of the reference concentration values for the background errors, and 1% of the reference concentration values for the observation errors.

(3) Perturbation

The best guess, i.e., the starting point for the iteration is

$$c_p^0 = c_0(1.3 + 9 * eps), eps = 0.001$$

(4) Simulation parameters are set as follows:

- Assimilation time window: 12 hrs
- Maximum number of model runs: 26
- Science Processes: advection, diffusion, and chemistry

(5) Results

Two different optimization experiments are carried out. One uses gradients computed via a continuous advection adjoint, the second uses the gradients computed via the continuous advection adjoint. In both cases (the same) discrete adjoints are used for the diffusion and chemistry processes.

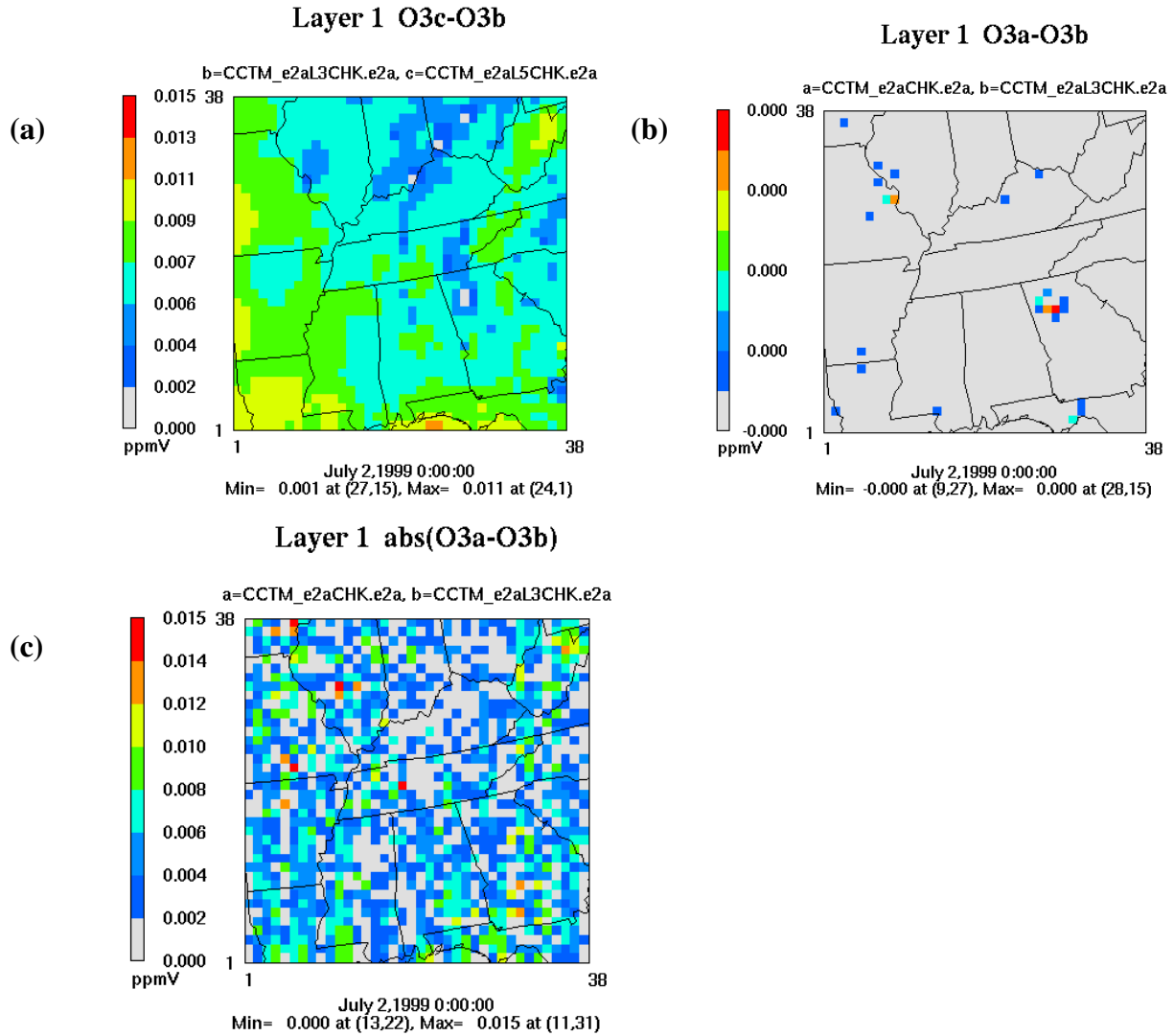


Figure 4.6 4D-VAR data assimilation carried out for 12 hours to optimize the initial O₃ values. (a) Difference between perturbed and reference concentration at initial time; (b) Difference between optimized and reference concentration at initial time using continuous advection adjoint; (c) Difference between optimized and reference concentration at initial time using discrete advection adjoint.

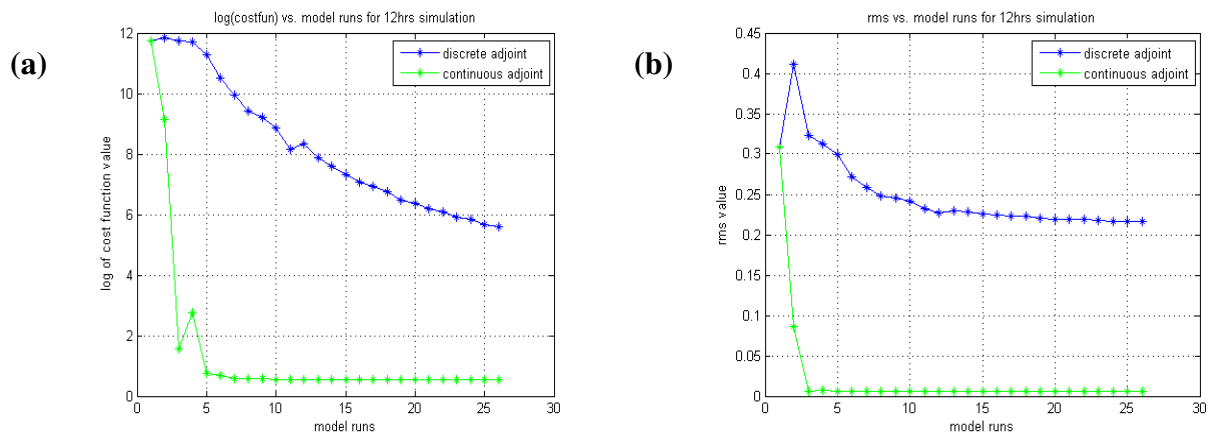


Figure 4.7 Convergence of L-BFGS optimization. (a) Log of the cost function values vs. model runs; (b) RMS values vs. model runs.

Figure 4.6a presents the difference between the best guess (perturbed initial condition) and reference initial condition before data assimilation. The initial field after data assimilation is compared against the reference initial field; if data assimilation process converges than the initial field after data assimilation should approximate well the reference initial condition. This is indeed the case with the solution obtained using a continuous advection adjoint, presented in Figure 4.6b. However, the optimized initial field using the discrete advection adjoint shows large differences from the reference. This can be seen in Figure 4.6c.

Figure 4.7 presents the evolution of the optimization process. Specifically, Figure 4.7a shows the decrease of the cost function with the number of iterations. Figure 4.7b shows the decrease of the root mean square error (difference between the initial conditions found by optimization and the reference solution) with the number of iterations. For both metrics the optimization converges faster when continuous advection adjoints are used (green lines). The solution obtained with discrete advection adjoints (blue lines) show a much slower convergence.

The counter-intuitive conclusion is that:

- The gradients given by the continuous advection adjoint have large discrepancies from the finite difference sensitivities, but lead to a very fast convergence of the 4D-VAR data assimilation.
- The gradients given by the discrete advection adjoint show an excellent agreement with the finite difference sensitivities in point-to-point tests. However, they lead to a much slower convergence of the optimization iterations during 4D-VAR data assimilation.

4.1.5 Adjoint sensitivity with respect to the boundary conditions

We now discuss the work on developing sensitivity w.r.t. boundary conditions and boundary scaling factor for CMAQ. First, we give the formula used to compute boundary sensitivities. We then show validation results first in a simple 1D advection problem, then for the current implementation of boundary sensitivity in CMAQ.

4.1.5.1 Sensitivity to boundary conditions and boundary scaling factors for 1D advection

4.1.5.1.1 Sensitivity to boundary conditions

Let $\psi(C_i(t))$ be a cost function defined on $\{C_i(t), i=1, \dots, N\}$. $C_i(t)$ is the concentration at time t at gridpoint i . Let $C_0(t)$ and $C_{N+1}(t)$ be the boundary conditions. The sensitivity of the cost function to the boundary conditions depends on adjoint variables as follows:

$$\lambda_0 = \left(\frac{\partial \psi(C_i(t))}{\partial C_0(t)} \right)^T = \left(\sum_{i=1}^N \frac{\partial \psi}{\partial C_i} \cdot \frac{\partial C_i}{\partial C_0} \right)^T = \sum_{i=1}^N \left(\frac{\partial C_i}{\partial C_0} \right)^T \cdot \left(\frac{\partial \psi}{\partial C_i} \right)^T$$

Since

$$\left(\frac{\partial \psi}{\partial C_i} \right)^T = \lambda_i, i=1, \dots, N$$

we have that:

$$\lambda_0 = \left(\frac{\partial \psi(C_i(t))}{\partial C_0(t)} \right)^T = \sum_{i=1}^N \left(\frac{\partial C_i}{\partial C_0} \right)^T \cdot \lambda_i$$

Therefore, in order to obtain λ_0 , we need the derivatives $\frac{\partial C_i}{\partial C_0}$. These derivatives represent the dependency of different concentrations on the boundary condition at the current time step. These derivatives depend on the particular numerical discretization used for one time step of the advection equation.

Considering the advection process and assuming the numerical scheme is the first order upwind scheme, we have:

$$c_1^{new} = c_1^{old} - U_1 \Delta t \frac{c_1^{old} - c_0^{old}}{\Delta x}$$

So

$$\frac{\partial C_i}{\partial C_0} = \frac{U_1 \Delta t}{\Delta x} \text{ and } \frac{\partial C_j}{\partial C_0} = 0, j = 2, \dots, N$$

Therefore,

$$\lambda_0 = \left(\frac{\partial C_i}{\partial C_0} \right)^T \cdot \lambda_1 = \frac{U_1 \Delta t}{\Delta x} \cdot \lambda_1$$

Similarly, for λ_{N+1} , we have

$$\lambda_{N+1} = \left(\frac{\partial \psi(C_i(t))}{\partial C_{N+1}(t)} \right)^T = \sum_{i=1}^N \left(\frac{\partial C_i}{\partial C_{N+1}} \right)^T \cdot \lambda_i$$

Since

$$c_N^{new} = c_N^{old} - U_N \Delta t \frac{c_{N+1}^{old} - c_N^{old}}{\Delta x}$$

We have:

$$\frac{\partial C_N}{\partial C_{N+1}} = \frac{U_N \Delta t}{\Delta x} \text{ and } \frac{\partial C_j}{\partial C_{N+1}} = 0, j = 1, \dots, N-1$$

Therefore,

$$\lambda_{N+1} = \left(\frac{\partial C_N}{\partial C_{N+1}} \right)^T \cdot \lambda_N = \frac{U_N \Delta t}{\Delta x} \cdot \lambda_N$$

When the numerical discretization is not the first order upwind, these formulas represent the sensitivities with respect to boundary conditions within an approximation error.

4.1.5.1.2 Sensitivity to scaling factors of the boundary conditions

Let B^{old} be the base boundary conditions and a be the scaling factor of the base boundary conditions. Consequently, $b=aB^{old}$ is the new boundary conditions after the base boundary scaled by a . The same (fixed) scaling factor is applied to the boundary values at all time steps. We are interested to find the sensitivity of a cost function with respect to such boundary scaling factors.

The sensitivity of the cost function (ψ) w.r.t. the scaling factor is derived as follows:

$$\left. \frac{\partial \psi}{\partial a} \right|_m = \frac{\partial \psi}{\partial b} \cdot \frac{\partial b}{\partial a} = \frac{\partial \psi}{\partial b} \cdot \frac{\partial (aB^{old})}{\partial a} = B^{old} \cdot \left. \frac{\partial \psi}{\partial b} \right|_m$$

$\left. \frac{\partial \psi}{\partial b} \right|_m$ is the sensitivity of the cost function w.r.t. the boundary condition at time t_n . Therefore, we can easily get the sensitivity to scaling factor after we get sensitivity to the boundary conditions by just multiplying it with B^{old} .

Since a is constant in time, we have

$$\left. \frac{\partial \psi}{\partial a} \right|_{overall} = \sum_{timesn} \left. \frac{\partial \psi}{\partial a} \right|_m$$

4.1.5.2 Validation on the 1-D advection PDE

We consider a 1-D advection PDE:

$$c_t + a(x)c_x = 0$$

where $c(x,t)$ is the concentration function and $a(x)$ is the wind velocity, $x \in [0, 2\pi]$, the initial condition is $c(x,0)=1+\sin(x)$, and the boundary condition is $c(0,t)=c(2\pi,t)=1$ (only the inflow condition is actually used). We solve this 1-D advection using PPM method implemented in CMAQ (hppm.F). The boundary sensitivities computed with the formulas discussed above are approximations of the discrete boundary sensitivities oh PPM.

(1) *Validation of sensitivities with respect to boundary conditions*

Consider the cost function:

$$\psi = \sum_{i=a}^b c(x_i, t_F)$$

where a and b are grid indices from a total of 200 internal gridpoints.

We want to validate the computation of the sensitivity: $\left. \frac{\partial \psi}{\partial b} \right|_m$
 $\psi(b + \delta b) - \psi(b)$

To test continuous adjoint sensitivity we compare the finite difference result $\psi(b + \delta b) - \psi(b)$ against the product between the sensitivity coefficient and the perturbation in the boundary, $\left. \frac{\partial \psi}{\partial b} \right|_{r_0} * \delta b$.

Here $\left. \frac{\partial \psi}{\partial b} \right|_{r_0}$ is the adjoint sensitivity to boundary value. The results are shown in Table 4.7. We notice that for some receptor choices the results are very accurate (e.g., gridpoints 71-97) while for other choices the results are very inaccurate (e.g., gridpoints 75-80).

a) *Validation results for the left boundary sensitivity are shown in Table 4.7.*

Table 4.7 Left boundary Sensitivity: Finite Difference vs. Adjoint.

| Receptor | $\Delta\psi$ Finite Difference | $\Delta\psi$ Continuous Adjoint | Relative Error |
|-----------------|--------------------------------|---------------------------------|----------------|
| a = 0, b = 70 | 0.0000 | 2.0079612E-07 | 0.0000 % |
| a = 71, b = 97 | 7.4882E-03 | 7.4997935E-03 | 0.1526 % |
| a = 98, b = 202 | 0.0000 | 3.5568429E-24 | 0.0000 % |
| a = 75, b = 80 | -2.442E-03 | 7.1687265E-03 | 134.06 % |
| a = 81, b = 86 | 9.2855E-03 | 9.2683244E-08 | -1.0E+7 % |
| a = 75, b = 85 | 5.7688E-03 | 7.1866894E-03 | 19.732 % |

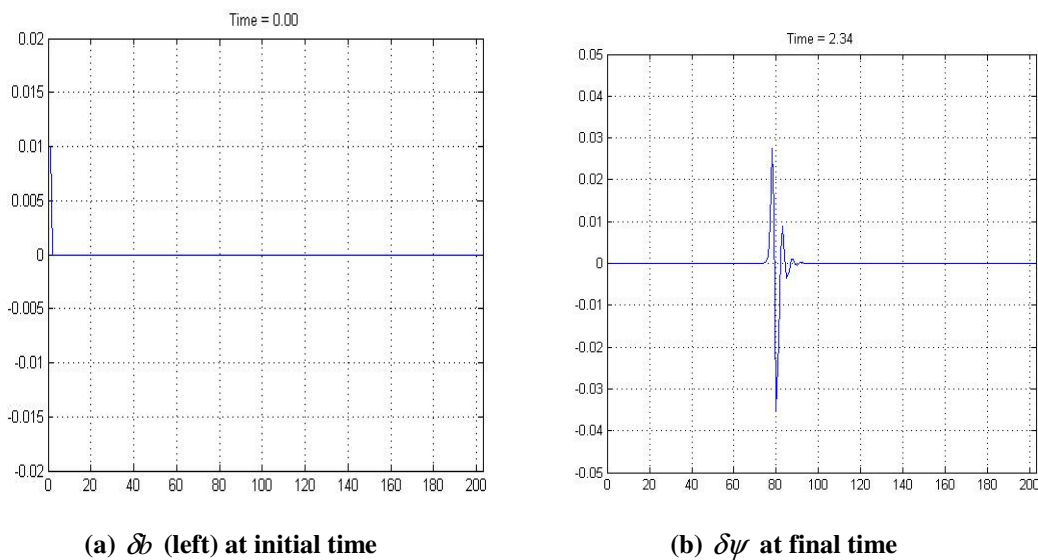


Figure 4.8 Evolution of a perturbation in the left boundary condition (inflow boundary). (a) Perturbation at left boundary at initial time. (b) Difference between the perturbed and base solutions at the receptor at final time.

In order to better understand how a perturbation of the boundary condition at the initial time affects the 1-D advection PDE solution, we plot the difference between the solutions with perturbed and base boundary values at the final time. The results are shown in Figure 4.8. For meaningful finite

difference results the receptor needs to be chosen such as to include the grids whose values are changed by the change in the boundary condition. In our test case, we see that solutions at grids 71-97 are changed. When the receptor includes the entire area where the solution changes, a good match between finite difference and adjoint is obtained. However, if we choose other receptors, we may get very large differences between finite differences and adjoint. This is again is caused by the nonlinearity of the PPM method.

b) *A similar analysis was carried out for the right boundary condition, with similar conclusions. Validation results for the right boundary sensitivity are shown in Table 4.8.*

Table 4.8 Right Boundary Sensitivity: Finite Difference vs. Adjoint.

| Receptor | $\Delta\psi$ Finite Difference | $\Delta\psi$ Continuous Adjoint | Relative Error |
|-------------------|--------------------------------|---------------------------------|----------------|
| a = 0, b = 104 | 0.000000 | 1.7369E-23 | 0.000000 % |
| a = 105 , b = 130 | 7.4996948E-03 | 7.4997E-03 | 0.000000 % |
| a = 131, b = 202 | 0.000000 | 2.0079E-07 | 0.000000 % |
| a =105, b = 110 | 2.3841858E-07 | 1.2994E-18 | -1.834E+13 % |
| a =110, b = 115 | 1.7404556E-04 | 6.5507E-13 | -2.656E+10 % |
| a =118, b = 123 | -2.2854805E-03 | 9.4818E-05 | 2510.455 % |
| a =118, b = 130 | 7.7905655E-03 | 7.4998E-03 | -3.878384 % |

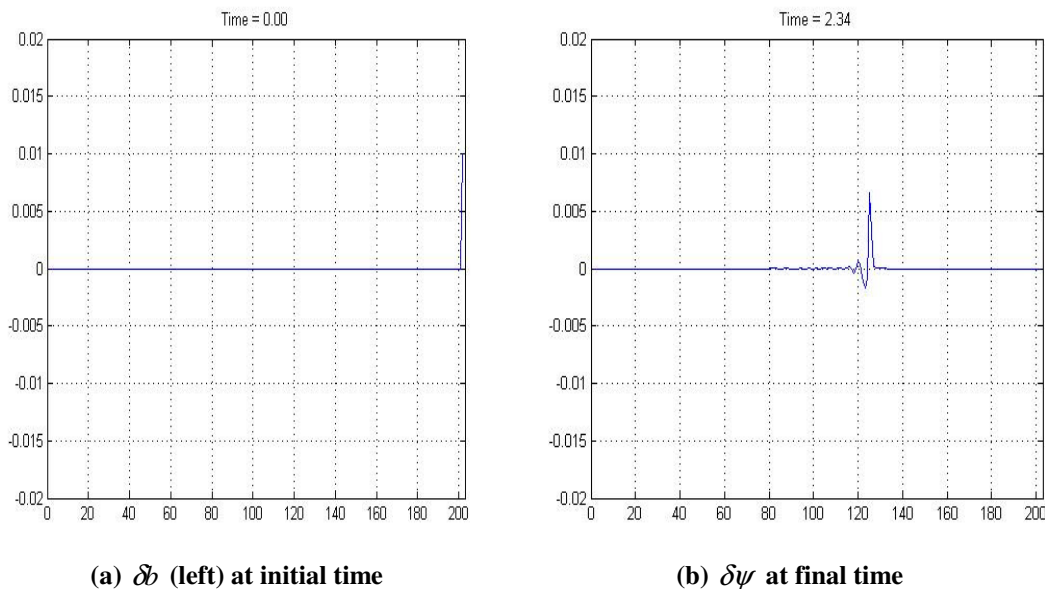


Figure 4.9 Evolution of a perturbation of the right boundary condition. (a) Perturbation at right boundary at initial time. (b) Difference between perturbed and base solutions at final time.

The conclusions of the right boundary perturbation test are similar to the ones obtained from the left boundary tests. The agreement between adjoint and finite differences depends on how the receptor is chosen. If we choose the receptor to include all the changes in the solution at the final time, then the adjoint result and the finite difference result are close to each other.

(2) Validation of sensitivities with respect to boundary scaling factors

For these tests we choose a cost function of the form:

$$\psi = \sum_{i=a}^b c(x_i, t_F)$$

We want to validate the sensitivities $\left. \frac{\partial \psi}{\partial a} \right|_m$ and $\left. \frac{\partial \psi}{\partial a} \right|_{overall}$ where a is the scaling factor of the boundary conditions.

- a) Validation results for $\left. \frac{\partial \psi}{\partial a} \right|_{t_0}$ are shown in Table 4.9. Like in the case of additive boundary perturbations, the agreement between the adjoint sensitivity and the finite difference sensitivity depends on the choice of the receptor (the choice of a,b in the definition of the cost function).

Table 4.9 Sensitivity to West boundary Scaling Factor: Finite Difference V.S. Adjoint.

| Receptor | $\Delta\psi$ Finite Difference | $\Delta\psi$ Continuous Adjoint | Relative Error |
|------------------|--------------------------------|---------------------------------|----------------|
| a = 0, b = 71 | 0.000000 | 1.2726824E-06 | 0.000000 % |
| a = 72, b = 99 | 7.5759E-03 | 7.5736693E-03 | -5.04E-02 % |
| a = 100, b = 202 | 0.000000 | 5.7291103E-26 | 0.000000 % |
| a = 75, b = 80 | 5.8774E-03 | 7.2404137E-03 | 18.82214 % |
| a = 80, b = 85 | 3.1566E-04 | 9.4773031E-07 | -33206.97 % |
| a = 75, b = 80 | 6.3409E-03 | 7.2585563E-03 | 12.63935 % |

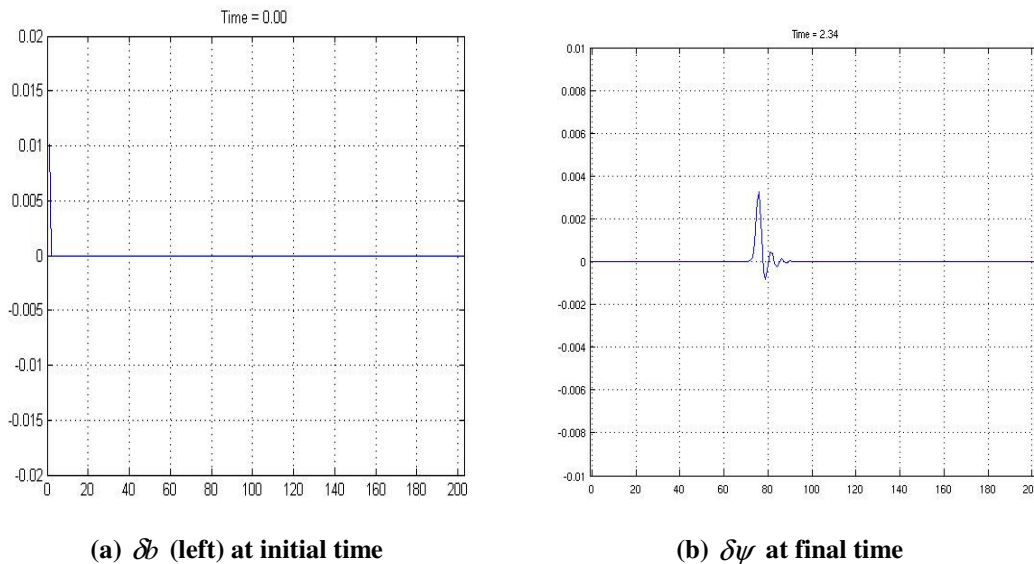


Figure 4.10 Evolution of the perturbation (with a scaling factor) at left boundary. (a) Perturbation at left boundary at initial time. (b) Solution difference in the receptor at final time.

- b) Validation of $\left. \frac{\partial \psi}{\partial a} \right|_{overall}$ In this case, the boundary conditions are perturbed by scaling the boundary values at each time step. The scaling factor is the same for all time steps, and we

seek to estimate the impact of this scaling to the cost function. For the base case the scaling factor is equal to 1.

The propagation of the boundary perturbation to the solution is plotted in Figure 4.11. We see that the change affects a wider area (due to the fact that the perturbation in boundary conditions is applied at all time steps). The chances of a receptor to capture the change in solution are therefore better. The results in Table 4.10 show a better agreement between the adjoint and the finite difference sensitivities. This agreement is less sensitive to the choice of receptor than in the case of perturbations applied to individual boundary conditions.

Table 4.10 Overall Boundary Scaling Factor Sensitivity: Finite Difference vs. Adjoint.

| Receptor | $\Delta\psi$ Finite Difference | $\Delta\psi$ Continuous Adjoint | Relative Error |
|-----------------|--------------------------------|---------------------------------|----------------|
| a = 0, b = 94 | 0.76759 | 0.7575000 | -1.332 % |
| a = 95, b = 202 | 0.00000 | 3.2700958E-22 | 0.000 % |
| a = 70, b = 90 | 6.06231E-02 | 6.8175033E-02 | 11.07 % |
| a = 50, b = 70 | 0.21209 | 0.2120999 | 3.597E-03 % |
| a = 30, b = 33 | 4.03995E-02 | 4.0399998E-02 | 1.180E-03 % |

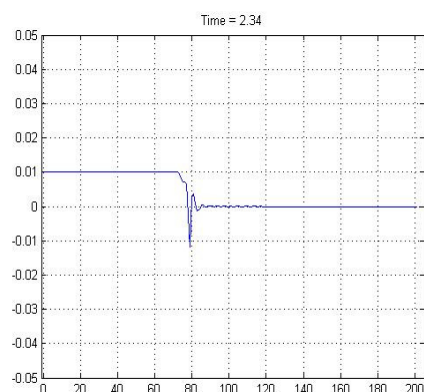


Figure 4.11 Evolution of perturbation of boundary at overall time: $\delta\psi$ at final time.

4.1.5.3 Sensitivity to boundary conditions and boundary scaling factors for CMAQ

4.1.5.3.1 Validation

We now show validation results for the sensitivity of a cost function w.r.t. western boundary condition perturbations at initial time. Both the cost function and western boundary condition are only considered for ozone. We switch off all other science processes except advection. This test is similar to the 1-D advection test. Specifically, each western boundary gridpoint $BCON(ROW, LAY)$ is perturbed individually (acts as a single grid source). The receptor is chosen as $\sum_{COL=1}^{NCOLS} CGRID(COL, ROW, LAY)$, the sum of the concentrations in the whole line of gridpoints having the same row and layer number as the source. Therefore, there are $NROWS \times NLAYS$ source-receptor pairs for which sensitivities are computed independently of each other. We calculate the sensitivities for

each source and receptor pair using both finite difference and adjoint approach, and we compare the results using scatter plots and relative differences.

To obtain the sensitivity to boundary condition, we first perform the sensitivity run (choose the simulation mode in run.cctm as snst). The results, LGRID_B, are stored in BSENST_WEST_ADJ_O3.txt.

To obtain the finite difference results, we perform the fd run (choose the simulation mode in run.cctm as fd which use the driver fd_driver.F). The perturbed run uses boundary conditions changed by Delta_B; the boundary perturbation is 1% of the nominal boundary values. In this run, the difference of cost functions before and after perturbing the boundary conditions are calculated. The cost function difference is compared against the (LGRID_B, Delta_B) dot product. LGRID_B are read from BSENST_WEST_ADJ_O3.txt.

The simulation is performed for 4hrs. The perturbation for finite difference test is 1% of the original boundary concentrations. We repeat the tests using both the continuous and the discrete adjoints of xadvection in the computation of boundary sensitivities. Continuous adjoint results are reported in Figure 4.12, and discrete adjoint results in Figure 4.13.

Figures 4.12a and 4.13a show the scatter plot of the finite difference sensitivity and continuous adjoint sensitivity/discrete adjoint sensitivity respectively, for each grid cell at western boundary. Figure 4.12b and 4.13b shows the relative difference between finite difference sensitivities and adjoint sensitivities.

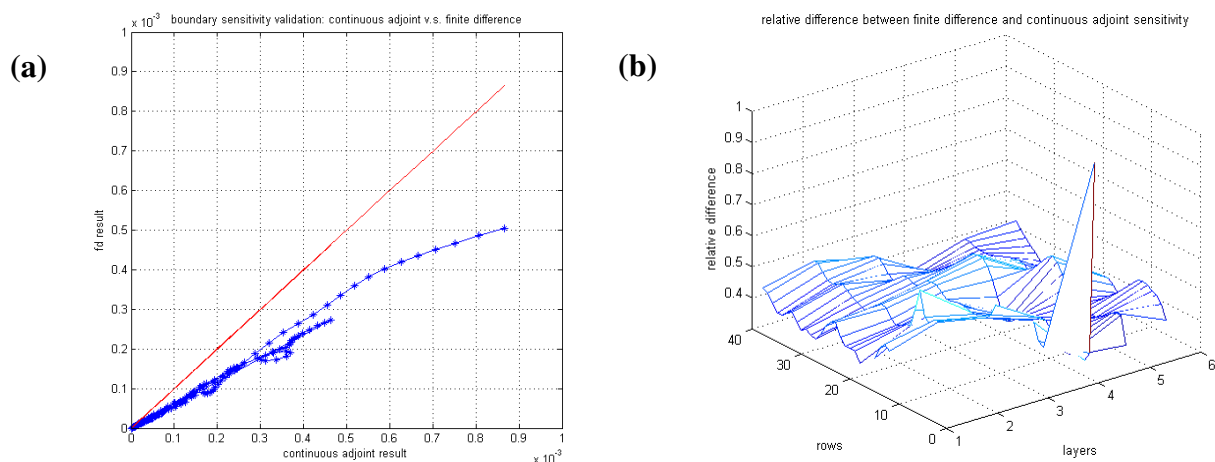


Figure 4.12 Western boundary continuous adjoint sensitivities validation. (a) Scatter plot of finite difference sensitivity (y-axis) and continuous adjoint sensitivity (x-axis). (b) Relative difference between continuous adjoint sensitivity and finite difference sensitivity.

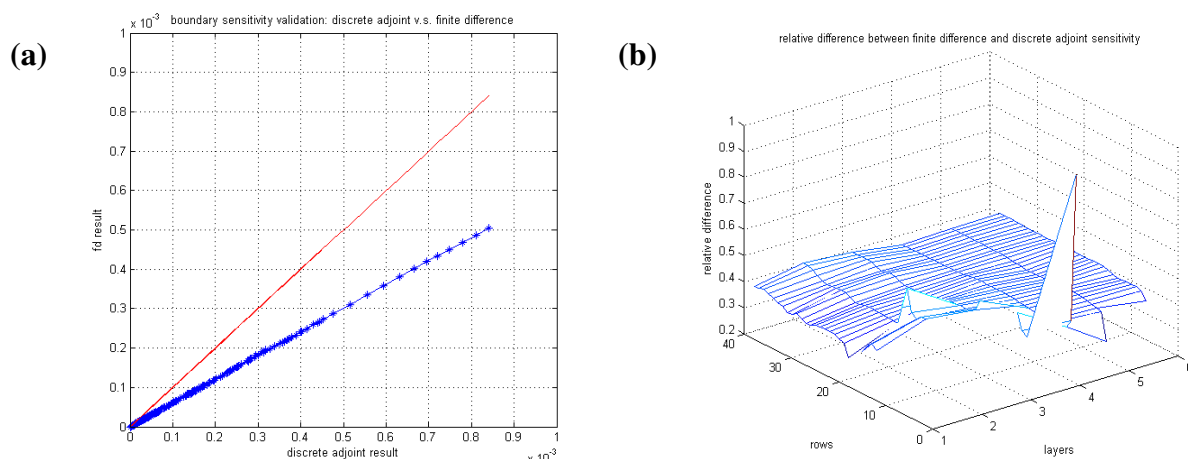


Figure 4.13 Western boundary discrete adjoint sensitivities validation. (a) Scatter plot of finite difference sensitivity and discrete adjoint sensitivity. (b) Relative difference between discrete adjoint sensitivity and finite difference sensitivity.

The results reported in Figures 4.12 and 4.13, indicate that in our current implementation the sensitivity to boundary conditions is about 40% off the finite difference results. This difference seems to be systematic (points are aligned well in the scatterplot. Further work is needed to better understand these differences and to improve the results.

4.1.5.3.2 Additional comments on boundary sensitivities

- (1) Different units need to be properly accounted for in the computation of sensitivities. In finite difference test, we have $\delta b(kg) \rightarrow \delta \psi(ppm)$, so we want to have the sensitivity $\frac{\delta \psi}{\delta b(kg)}$. To get this value, LGRID is decoupled before being passed to the stob subroutine where we calculate LGRID_B (boundary sensitivities) using LGRID information.
- (2) The current code calculates boundary sensitivities at synchronization times. However, in the formula of getting the LGRID_B, DT is the advection time. We can show that the sensitivity at the synchronization time is the sum of the sensitivities at all the advection times within that synchronization step. If the wind velocity and LGRID do not change considerably between different advection times, we can further approximate the sensitivity at the synchronization time as follows:

$$\frac{\partial \psi}{\partial b_w} = \sum U_{1advection} \frac{\Delta t_{advection}}{\Delta x} \cdot \lambda_1(advection_time) \approx \frac{U_{1syn} \Delta t_{syn}}{\Delta x} \cdot \lambda_1(syn_time)$$

4.1.6 References

- Byrd, R. H., P. Lu, J. Nocedal and C. Zhu, 1995: "A limited memory algorithm for bound constrained optimization", *SIAM J. Scientific Computing* 16 (1995), no. 5, pp. 1190-1208.
- Giering, R., 1999: TAMC: Tangent linear and Adjoint Model Compiler, <http://autodiff.com/tamc>
- Hakami, A., D.K. Henze, J.H. Seinfeld, K. Singh, A. Sandu, S. Kim, D. Byun, and Q. Li, 2007: "The Adjoint of CMAQ", *Environmental Science and Technology*, Vol. 41, No. 22, p. 7807-7817.
- Sandu, A., D. Daescu, and G.R. Carmichael, 2003: "Direct and Adjoint Sensitivity Analysis of Chemical Kinetic Systems with KPP: I -- Theory and Software Tools", *Atmospheric Environment* 37 (2003), pp. 5083-5096.

- Sandu, A, and K. Singh, 2007: "Development of an Adjoint Model for CMAQ-4.5", Report for Project H45C, Prepared for Houston Advanced Research Center, 44pp.
- Singh, K., A. Sandu, A. Hakami, and J. Seinfeld., 2007: "CMAQ v4.5 Adjoint User's Manual: An adjoint model for CMAQ to perform 4D-var Data Assimilation and SensitivityAnalysis", 24pp, Available from http://people.cs.vt.edu/~asandu/Software/CMAQ_ADJ/CMAQ_ADJ.html
- Zhu, C., R.H. Byrd, and J. Nocedal, 1997: "L-BFGS-B – Fortran routines for large scale bound constrained optimization", *ACM Trans. Math. Software*, **23**(4), p 550-560

4.2 Application of CMAQ/4D-VAR modeling codes to investigate model input uncertainties

4.2.1 Observation Inputs

4.2.1.1 Observations

Ozone measurements from the following three different data sets are used for the 4D-VAR data assimilation tests:

- Hourly averaged ozone measurements from Continuous Ambient Monitoring Stations (CAMS) operated by Texas Commission on Environmental Quality (TCEQ). There are 347 sites available in the TEXAS domain. Data are organized in daily files for the period of 8/29-9/3, 2006. That is, each daily file contains 24 time steps and the first time step is the averaged ozone between 00:00:00 UT to 1:00:00 UT.
- Hourly averaged ozone measurements from AIRNow network. Note that the AIRNow data include 188 TCEQ sites (114 sites with ozone monitoring). Other than TCEQ sites, the AIRNow data set also include U.S. EPA, NOAA, National park Services (NPS), and tribal operated monitoring stations. Data are organized in daily files for the period of 8/29-9/3, 2006. Following the convention of AIRNow data sets, each daily file contains 24 time steps and the first time step is the averaged ozone between 03:00:00 UT to 4:00:00 UT.
- Ozonesonde measurements were made at University of Houston, Houston, TX (29.97N, 95.33W, 19m) (<http://croc.gsfc.nasa.gov/intexb/ions06.html>) by Gary A. Morris of Valparaiso University. The Six launches listed below were processed. Data of the first 5 launches are downloaded from MILAGRO/INTEX/IMPEX archive (<http://www-air.larc.nasa.gov/cgi-bin/arcstat-b>) and they are revision-R1 version (last updated on Feb. 27, 2007). Data on 9/01/2006 were from Bernhard Rappenglück.
 - 8/28/2006, launched at 18:06:04 UT
 - 8/29/2006, launched at 18:03:00 UT
 - 8/30/2006, launched at 18:13:50 UT
 - 8/31/2006, launched at 18:29:34 UT
 - 8/31/2006, launched at 12:00:48 UT
 - 9/01/2006, launched at 18:54:16 UT

4.2.1.2 Data Processing

Original data from different sources are organized quite differently and they are not associated with the computational grid information. Complicating the 4D-VAR data assimilation code with processing routines to deal with different data formats should be avoided. In addition, such processing need to completed once and it is not efficient to include that inside the assimilation code, especially for the re-analysis tests for the current project. So, the processing of the data are made beforehand to generate a universal input format which translates the geographic locations into the grid indices. Data of three different sources are processed separately following the steps below.

- Convert the longitude, latitude, and altitude information into grid indices. As the current computational grid generated using MM5 assumes the earth being a sphere shape, a compatible

projection is made to convert latitude, longitude into cross points of the computational domain using mm5ll_xy subroutine. CAMS and AIRNow data are placed inside the first layer. Layer height information inside GRIDCRO3D is used to project the vertical index for ozonesonde data. The horizontal shift during the balloon rising is calculated using the wind speed and direction provided along with the ozone measurements. Such shift is used to calculate the horizontal locations of the ozone measurements at upper levels.

- The time index is given assuming the simulation starts at 0:00:00 UT. A time shift is needed if simulations start at a different time. Ozone observations are in the units of ppbv (parts per billion). Observations flagged as being invalid and those outside the computational domain are taken out.
- Remove questionable data from the data sets. For the ozonesonde data, measurements of ozone greater than 1000 ppbv are removed from the data sets.
- Reduce data according to the model resolution. The measurements contain more temporal and spatial variation than the model can resolve at the current resolutions. For instance, the ozonesonde has a temporal resolution of 1 second and the model advection time step is normally longer than 300 seconds. In space, there are multiple measurements inside certain grids cell for both CAMS and AIRNow data. To avoid providing conflicting information to the 4D-VAR system a four-dimensional grid averaging is performed to generate the final data input files. Observations inside one grid cell could vary significantly and this is called representative error. It is normally much larger than the measurement error [Chai *et al.*, 2007]. In this study, the total error, including the measurement error and representative error, is assumed as 8% uniformly.

4.2.2 4D-VAR data assimilation for initial ozone concentrations

4.2.2.1 Settings

- (1) Grid description and input files.
The grid name is uh_dw04_01 with 82 x 64 grid points. Input data used corresponds to this grid definition.(Bcon: Bcon_dw04km; Icon: Cmaq_cb4)
- (2) Background error covariance matrix and Observation error covariance matrix.
Both matrices are diagonal. The BG_WGT variable in GET_BG_FG subroutine is set to 1.0 and the UNC_OBS variable in GET_OBS_FG subroutine is set to 0.08.
- (3) Observation data.
We were using original Airnow observation data from Tianfeng Chai for the 82 x 64 domain without retrieving the 0 values.
- (4) Assimilation period and assimilation windows.
Assimilation period is from 2006242:040000 (08/30/2006, 4am) to 2006244:240000 (09/01/2006, midnight). Note that the start time at the first day is 4am since the observation data for that day starts from 4am. All times are UTC.
The typical assimilation window used in this study is 6 hours. The data assimilation runs are initialized with the "analysis" from the previous 6-hour run. Specifically, the initial condition (guess) for current 6-hour run is from the analysis for the previous 6-hour run. We perform cyclic 4D-VAR data assimilation for the following windows:
 - 1) 2006242:040000 – 2006242:100000;

- 2) 2006242:100000 – 2006242:160000;
 - 3) 2006242:160000 – 2006242:240000 (8hrs);
 - 4) 2006243:000000 – 2006243:060000;
 - 5) 2006243:060000 – 2006243:120000;
 - 6) 2006243:120000 – 2006243:180000;
 - 7) 2006243:180000 – 2006243:240000;
 - 8) 2006244:000000 – 2006244:060000;
 - 9) 2006244:060000 – 2006244:120000;
 - 10) 2006244:120000 – 2006244:180000;
 - 11) 2006244:180000 – 2006244:240000;
- (5) Control variables.
This experiment is for O₃ initial conditions (at the beginning of each subinterval).
- (6) Optimization settings.
The maximum number of model runs (forward and backward runs) is set to 15.

4.2.2.2 4D-VAR data assimilation results

We use both scatter plots and time series plots to compare the ozone concentrations from observations, background model run, and model analysis. Model analysis is the model prediction using optimized initial condition for each assimilation window.

For each space-time point (column, row, layer, hour) where an observation is available we read the corresponding model background and analysis predictions. Scatter plots for pairs of observation-background values and observation-analysis values are constructed. They illustrate the improvements in the model-observation agreement after the assimilation, and therefore illustrate the quality of the analysis. For each case we also compute the R-square mismatch factor between analysis and observations.

For each observation location, we plot the concentration values of observation, background and analysis as time evolves to see if the analysis is really an improvement compared to background. Six specific locations are chosen for each day and results for these locations are reported. The location of these stations is shown in Figure 4.14.

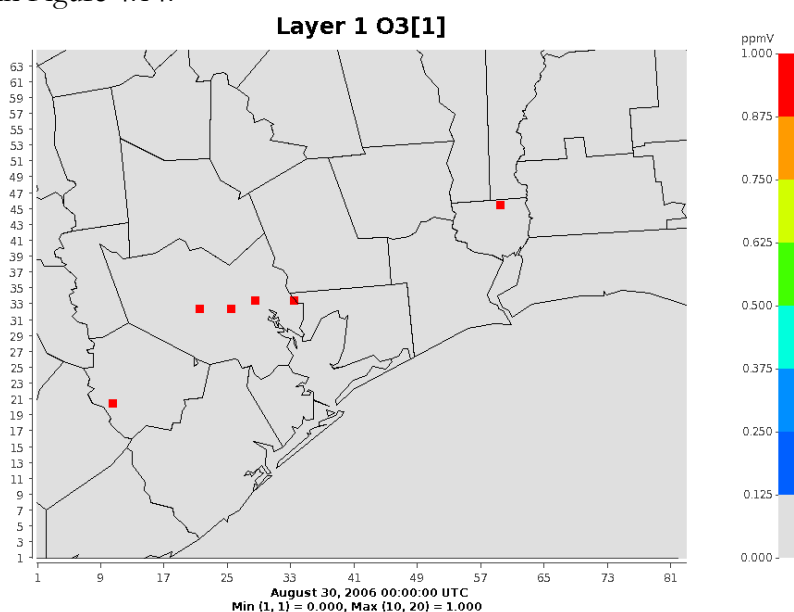


Figure 4.14 Location of stations where time series are shown.

The background value are obtained by running the forward model 3 days using the model input at the beginning of the first day as the initial condition.

The results are presented next.

(1) *First day (Aug. 30, 2006): 2006242:040000 – 2006242:240000.*

The scatter plots for the first day are shown in Figure 4.15. The model-observation agreement, as quantified by the R-square factor, increases from the background value of 0.65 to the analysis value of 0.83.

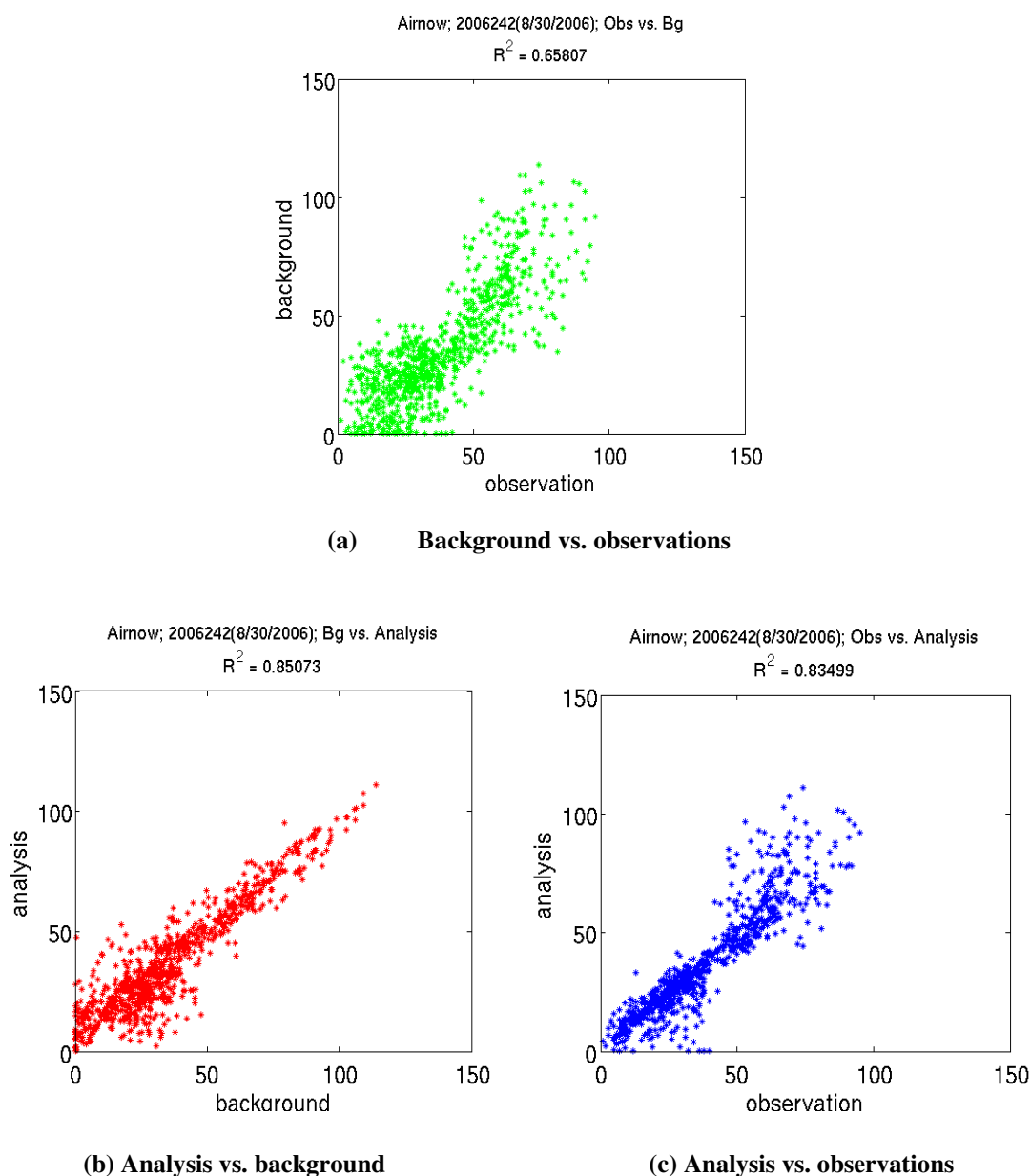
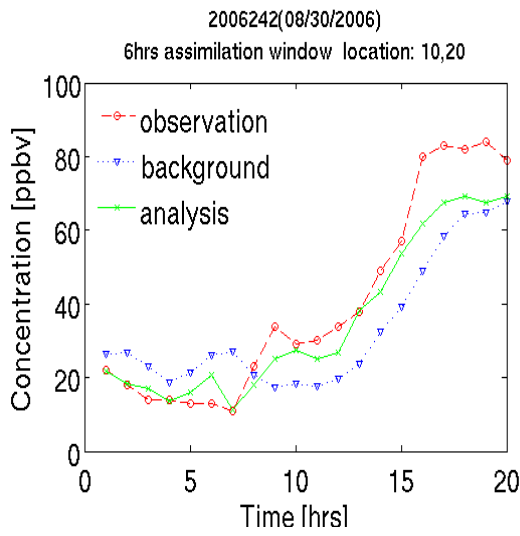


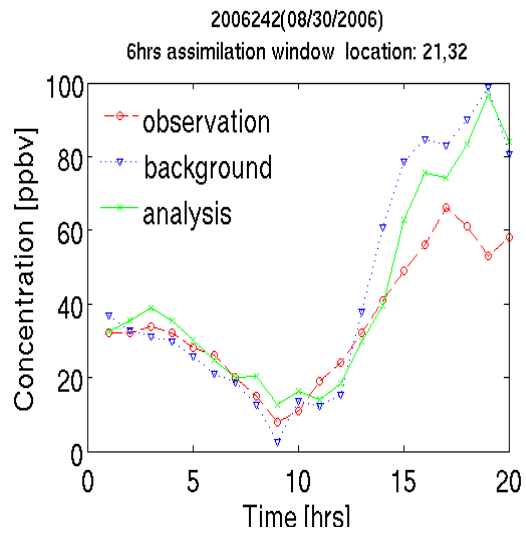
Figure 4.15 Scatter plots of observations and the corresponding model predictions for Aug. 30, 2006.

Time series of observations, background model prediction, and analysis model predictions at selected stations during the first day are shown in Figure 4.16. For each station the analysis is consistently closer to the observations. Recall that each analysis trajectory is obtained from four 6-hour

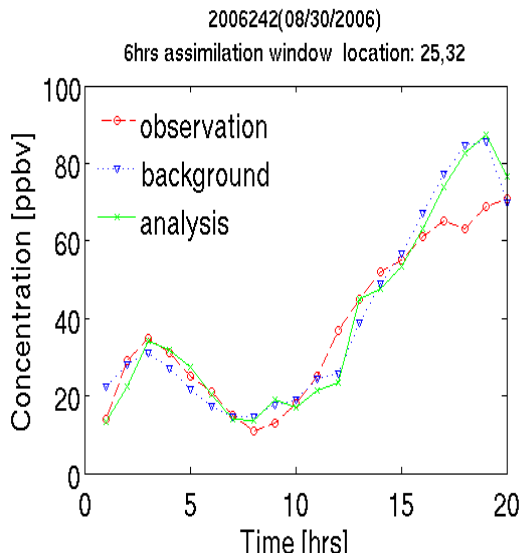
assimilation windows, and the analysis profiles may not be smooth at the beginning of a new assimilation window.



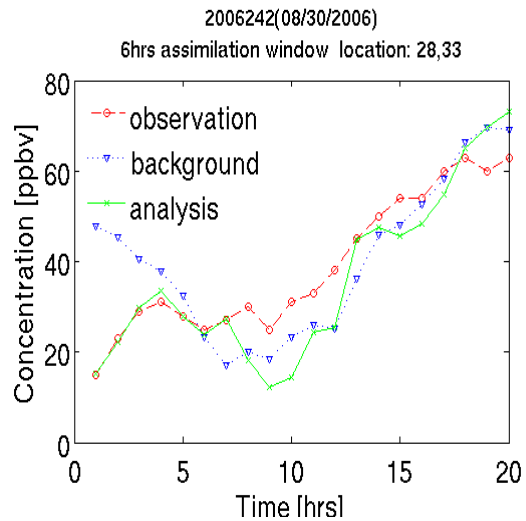
(a) location (col=10,row =20)



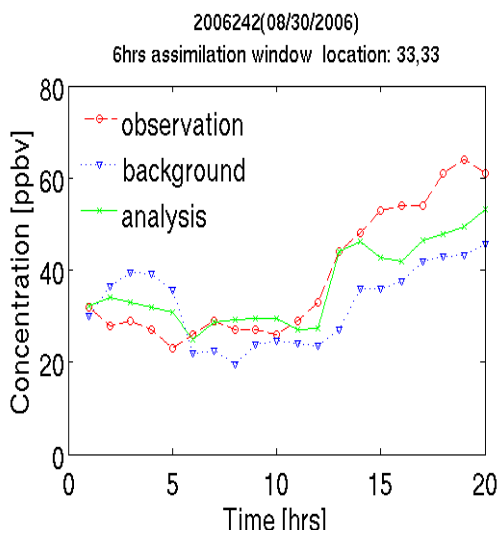
(b) location (col=21,row =32)



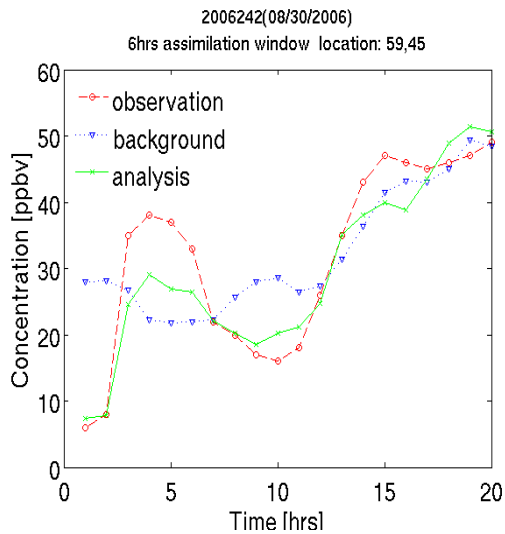
(c) location (col=25,row =32)



(d) location (col=28,row =33)



(e) location (col=33,row =33)



(f) location (col=59,row =45)

Figure 4.16 Time series of observations, background model prediction, and analysis model predictions at selected stations on Aug. 30, 2006.

(2) *Second day (Aug. 31, 2006): 2006243:000000 - 2006243:240000*

The scatter plots for the second day are shown in Figure 4.17. The model-observation agreement, as quantified by the R-square factor, increases from the background value of 0.54 to the analysis value of 0.94.

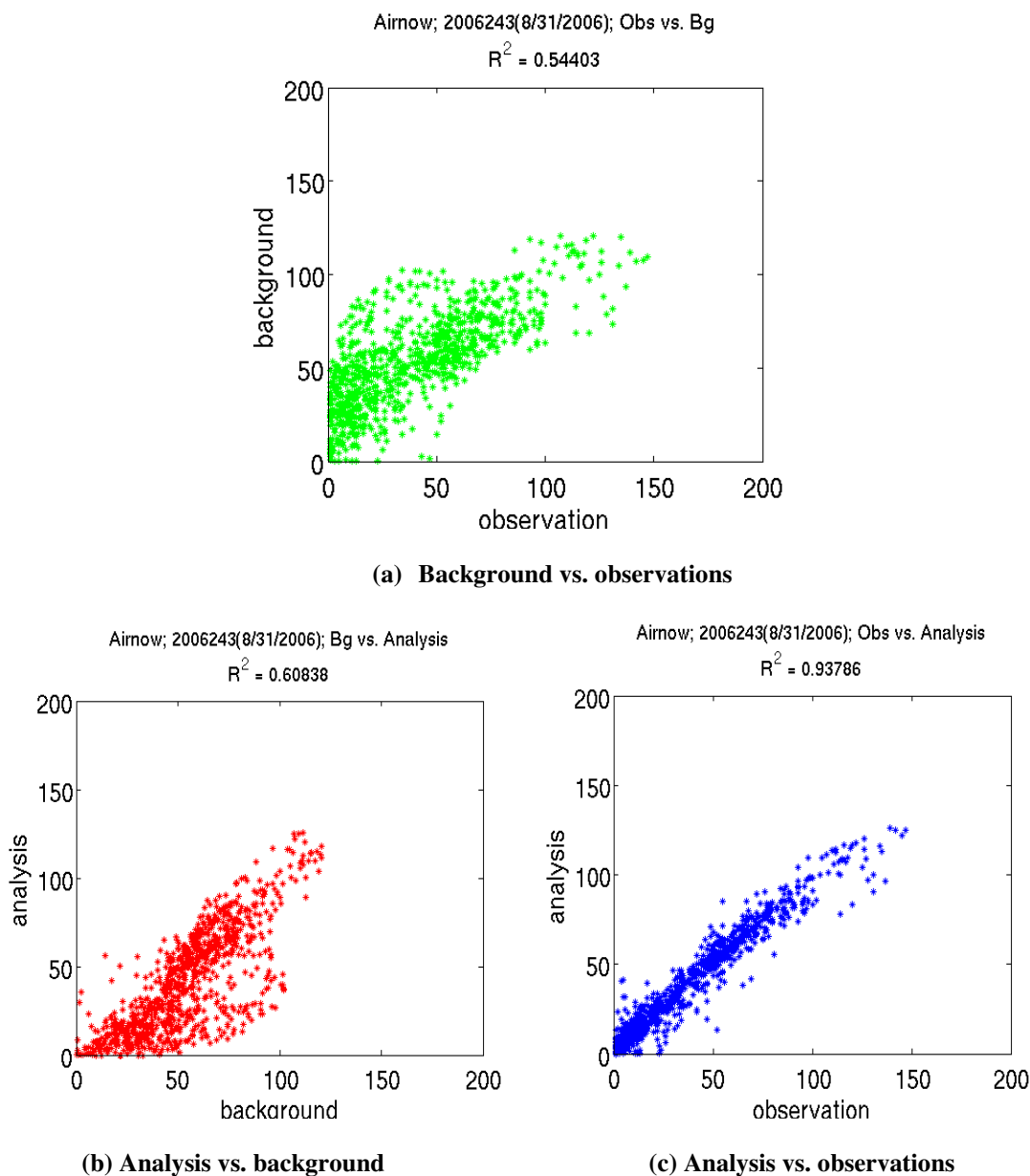


Figure 4.17 Scatter plots of observations and the corresponding model predictions for Aug. 31, 2006.

Time series of observations, background model prediction, and analysis model predictions at selected stations during the second day are shown in Figure 4.18. For each station the analysis is slightly closer to the observations. Recall that each analysis trajectory is obtained from four 6-hour assimilation windows, and the analysis profiles may not be smooth at the beginning of a new assimilation window.

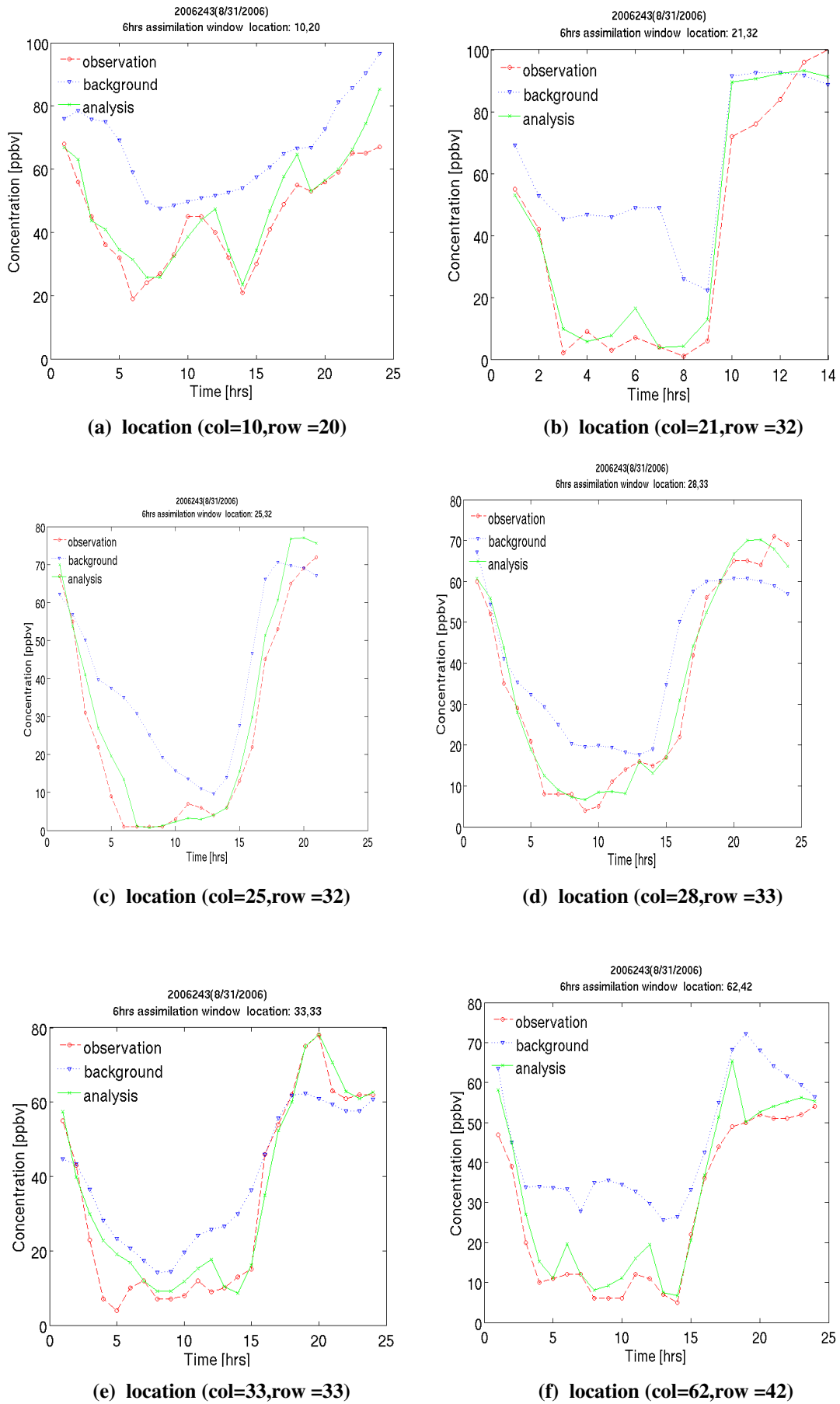


Figure 4.18 Time series of observations, background model prediction, and analysis model predictions at selected stations on Aug. 31, 2006.

(3) *Third day (Sep. 1, 2006): 2006244:000000 -- 2006244:240000*

The scatter plots for the third day are shown in Figure 4.19. The model-observation agreement, as quantified by the R-square factor, increases from the background value of 0.57 to the analysis value of 0.92.

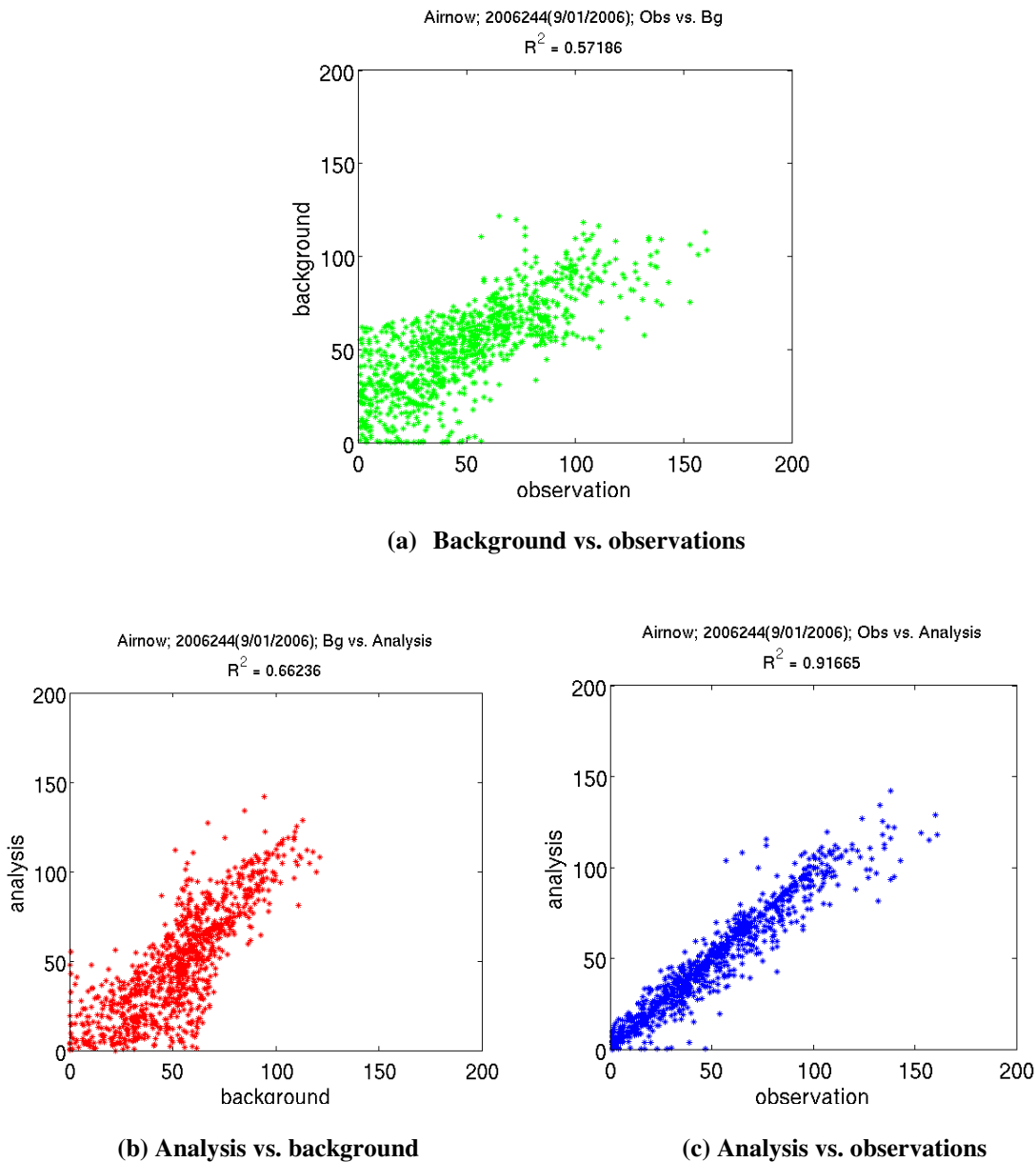


Figure 4.19 Scatter plots of observations and the corresponding model predictions for Sep. 1, 2006.

Time series of observations, background model prediction, and analysis model predictions at selected stations during the third day are shown in Figure 4.20. For each station the analysis is slightly closer to the observations. Recall that each analysis trajectory is obtained from four 6-hour assimilation windows, and the analysis profiles may not be smooth at the beginning of a new assimilation window.

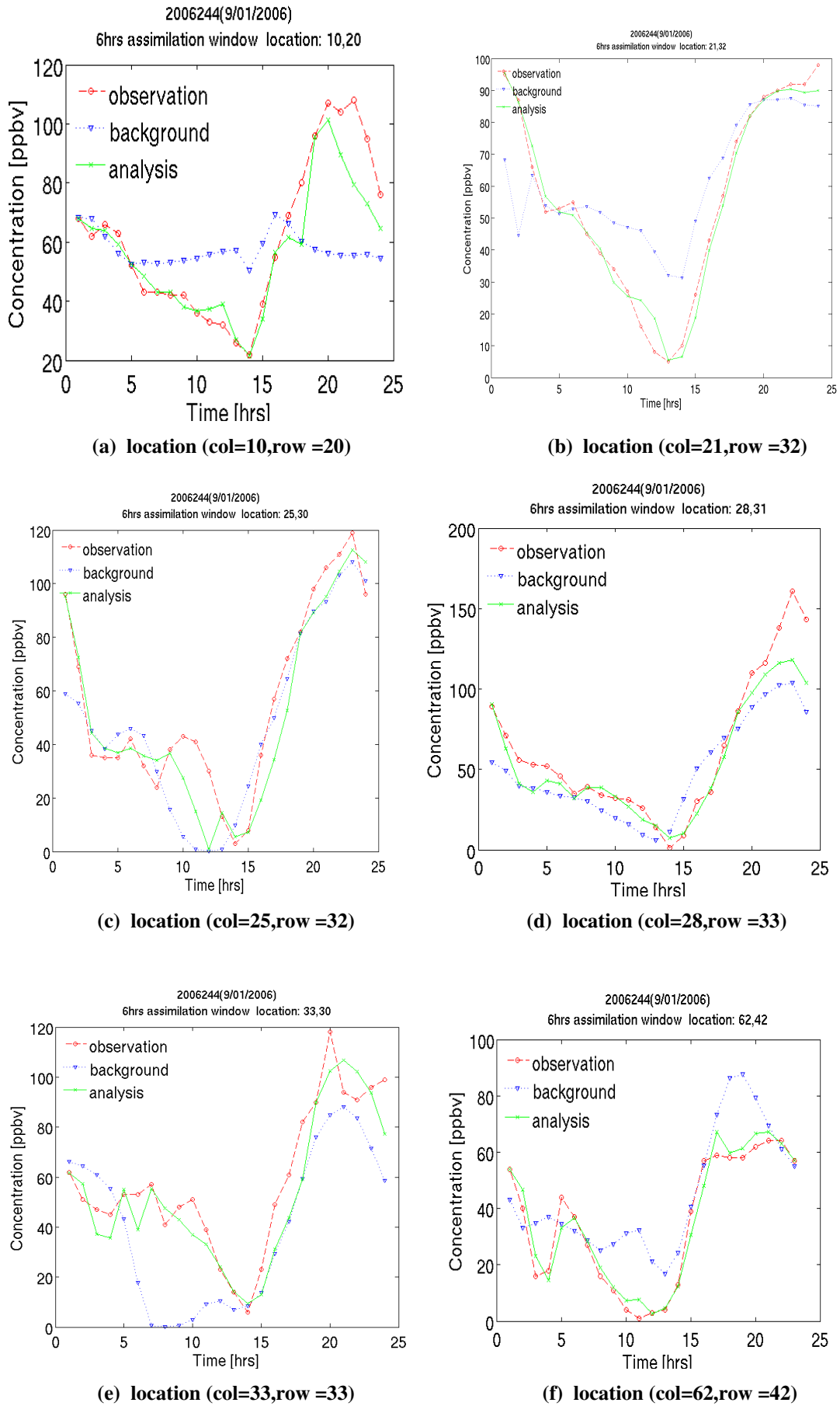


Figure 4.20 Time series of observations, background model prediction, and analysis model predictions at selected stations on Sep. 1, 2006.

4.2.2.3 Comparison of assimilation results against CAMS data

The CAMS observations were **not used directly** in data assimilation. After assimilation of AirNow data the agreement of model predictions and CAMS data is improved. Note that the CAMS and the AirNow data are not fully independent as many of the locations overlap. Nevertheless, the test provides a measure of the beneficial effects of assimilation.

(1) *First day (Aug. 30, 2006): 2006242:040000 – 2006242:240000.*

The scatter plots for the first day are shown in Figure 4.21. The agreement between model predictions and CAMS observations, as quantified by the R-square factor, remains about the same with a background value of 0.373 and an analysis value of 0.378.

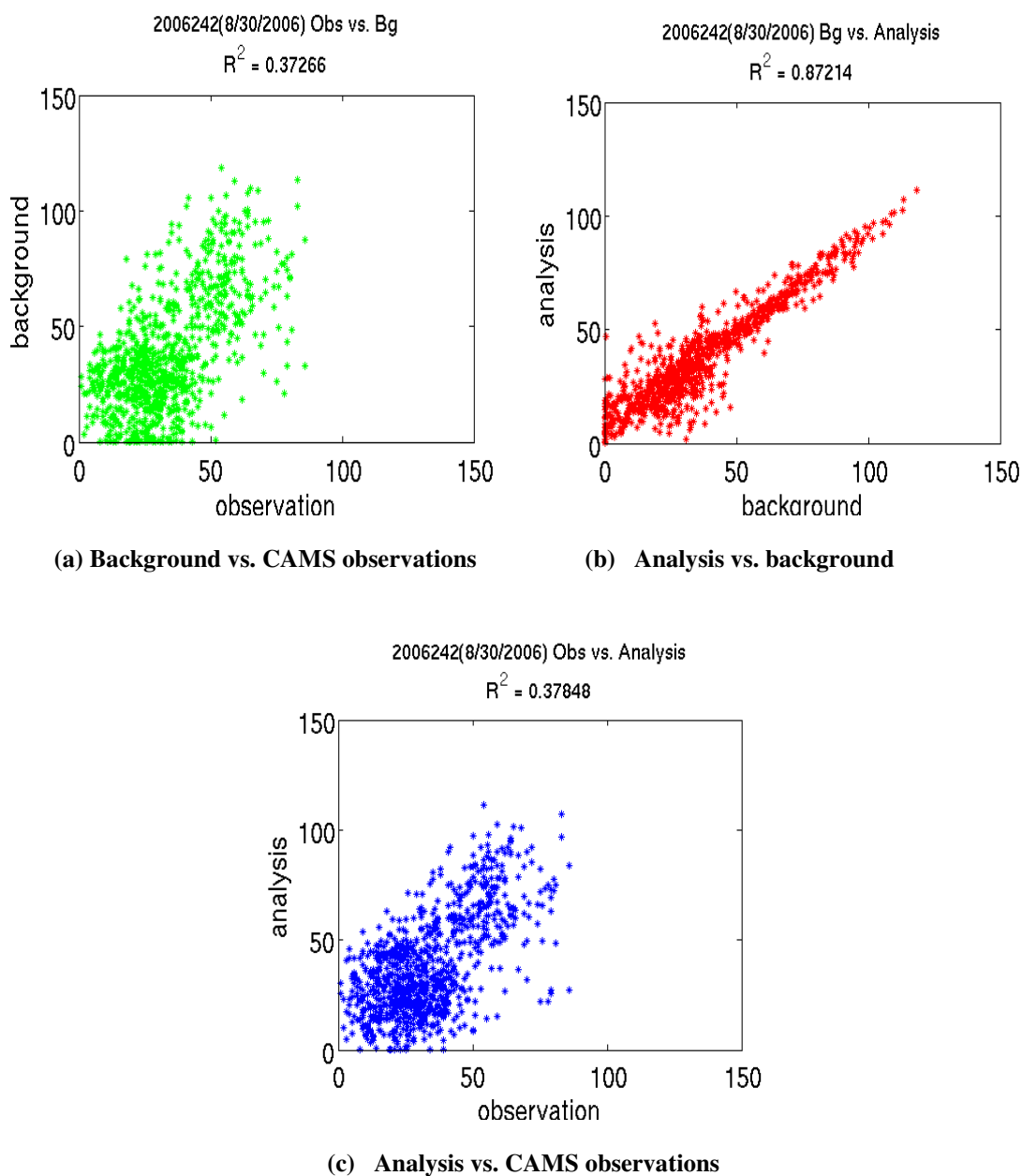


Figure 4.21 Scatter plots of CAMS observations and the corresponding model predictions for Aug. 30, 2006.

(2) *Second day (Aug. 31, 2006): 2006243:000000 - 2006243:240000*

The scatter plots for the second day are shown in Figure 4.22. The agreement between model predictions and CAMS observations, as quantified by the R-square factor, increases from the background value of 0.48 to the analysis value of 0.82.

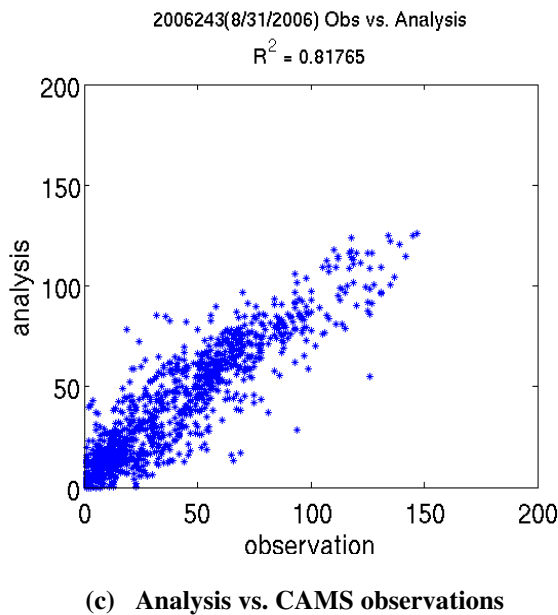
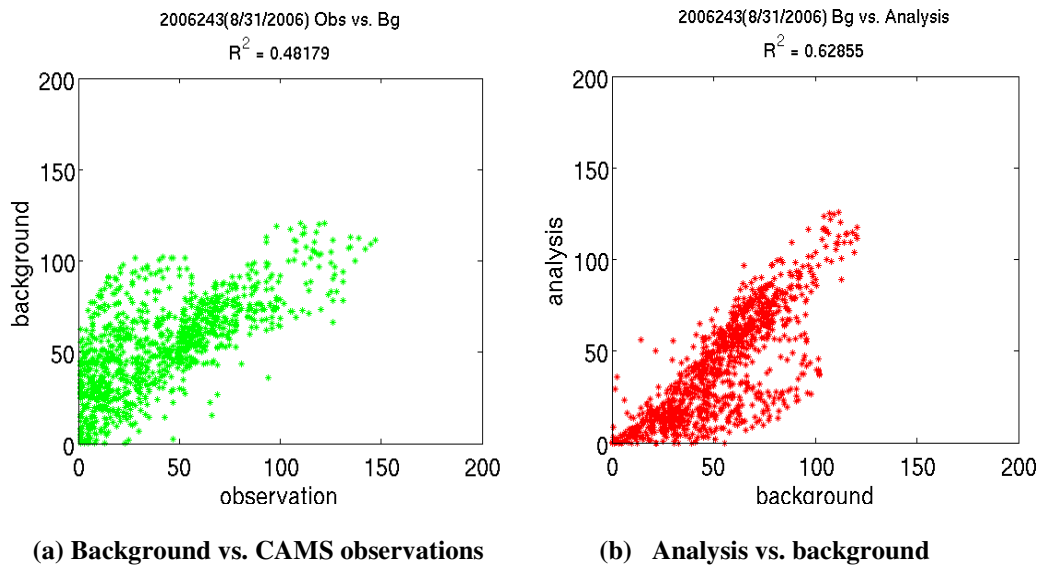


Figure 4.22 Scatter plots of CAMS observations and the corresponding model predictions for Aug. 31, 2006.

(3) *Third day (Sep. 1, 2006): 2006244:000000 -- 2006244:240000*

The scatter plots for the third day are shown in Figure 4.23. The agreement between model predictions and CAMS observations, as quantified by the R-square factor, increases from the background value of 0.55 to the analysis value of 0.75.

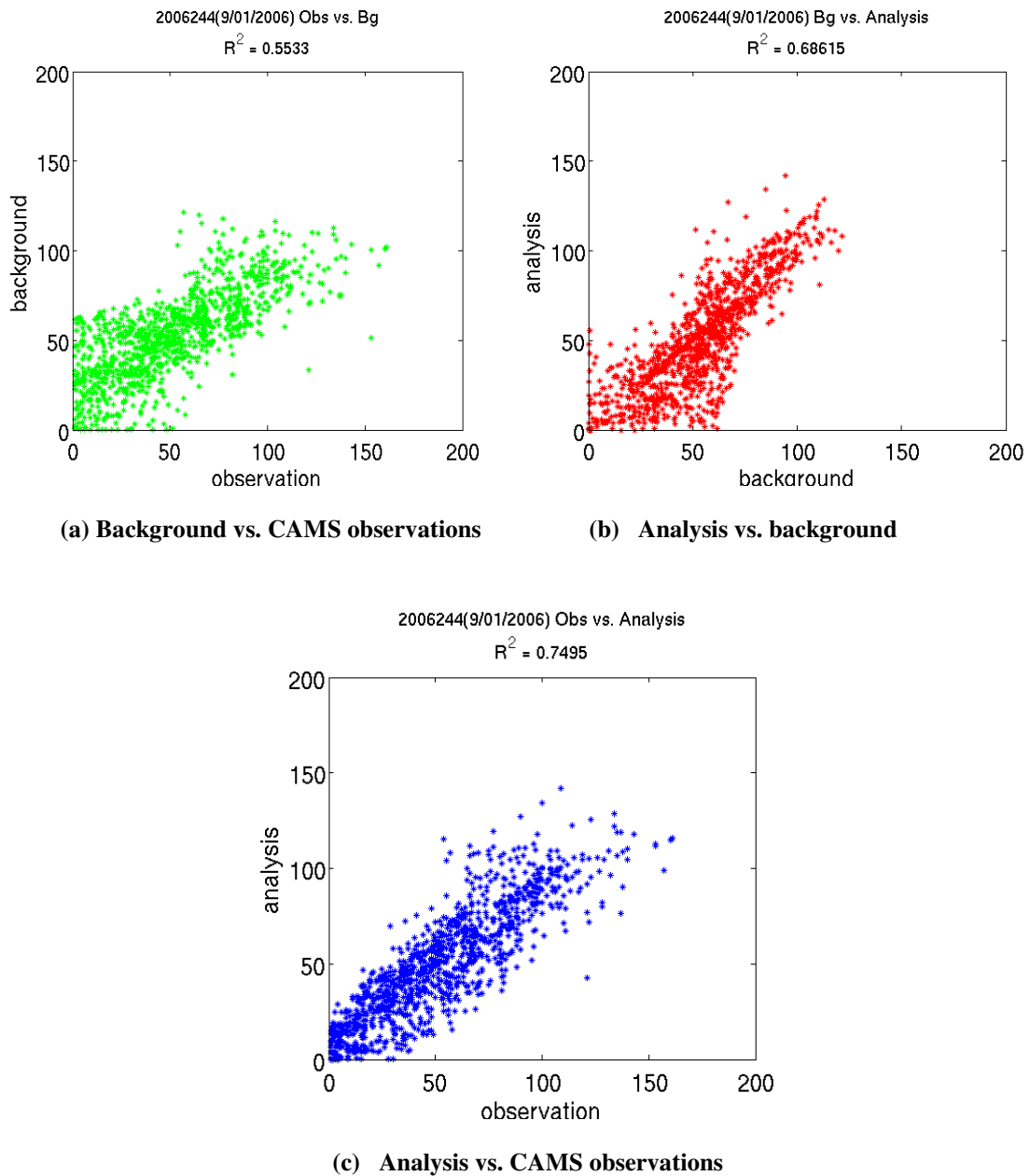


Figure 4.23 Scatter plots of CAMS observations and the corresponding model predictions for Sep. 1, 2006.

4.2.3 4D-VAR with boundary condition scaling factor as control variable

In this experiment, we want to further improve the model prediction based on the optimized initial condition after 4D-VAR using initial condition as control variable.

The results show that, although the cost function decrease by 40%, the scatter plots and time series plots indicates that the new model prediction after this 4D-VAR experiment doesn't get closer to the observations than the background. The reason need to be investigated in the future. One possible reason is that the scaling factors are considered as const at this time so scaling factors as a function of time may need to be used in the future.

4.2.3.1 Settings

- (1) Grid description and input files.
The grid name is uh_dw04_01 with 82 x 64 grid points. Input data used corresponds to this grid definition.(Bcon: Bcon_dw04km; Icon: Cmaq_cb4)
- (2) Background error covariance matrix and Observation error covariance matrix.
Both matrices are diagonal. The BG_WGT variable in GET_BG_FG subroutine is set to 1.0 and the UNC_OBS variable in GET_OBS_FG subroutine is set to 0.08.
- (3) Observation data and background.
We were using the latest Airnow observation data from Tianfeng for the 82 x 64 domain.
The background is the optimized initial concentration after 4D-VAR using the initial condition as control variable.
- (4) Assimilation period and assimilation window.
Assimilation period is from 2006243:000000 (08/31/2006, 0am) to 2006244:24000 (09/01/2006, midnight). All times are UTC. The assimilation window is 48 hrs.
Note that the reasons for choosing this period and assimilation window is that we want the boundary conditions take more impact on the cost function so a long assimilation window would be helpful.
- (5) CMAQ-ADJ codes:
Peter's latest codes with parallel version.
- (6) Control variable.
This experiment is for 4D-VAR with boundary scaling factor as control variable.
- (7) Optimization settings.
Max number of model runs (forward and backward runs) is set to 15.

4.2.3.2 4D-VAR data assimilation results

LBFGS log

RUNNING THE L-BFGS-B CODE

it = iteration number
 nf = number of function evaluations
 nint = number of segments explored during the Cauchy search
 nact = number of active bounds at the generalized Cauchy point
 sub = manner in which the subspace minimization terminated:
 con = converged, bnd = a bound was reached
 itls = number of iterations performed in the line search
 stepl = step length used
 tstep = norm of the displacement (total step)
 projg = norm of the projected gradient
 f = function value

* * *

Machine precision = 2.220D-16

N = 6808 M = 5

| it | nf | nint | nact | sub | itls | stepl | tstep | projg | f |
|----|----|------|------|-----|------|---------|---------|-----------|-----------|
| 0 | 1 | - | - | - | - | - | - | 1.500D+00 | 5.101D+05 |
| 1 | 2 | 2103 | 2102 | --- | 0 | 1.0D+00 | 5.0D+01 | 2.500D+00 | 3.184D+05 |
| 2 | 3 | 2 | 1482 | bnd | 0 | 1.0D+00 | 2.3D+00 | 2.496D+00 | 3.168D+05 |
| 3 | 4 | 27 | 1507 | con | 0 | 1.0D+00 | 2.0D+00 | 2.484D+00 | 3.162D+05 |
| 4 | 5 | 14 | 1516 | bnd | 0 | 1.0D+00 | 1.7D+00 | 2.500D+00 | 3.157D+05 |
| 5 | 6 | 19 | 1527 | bnd | 0 | 1.0D+00 | 1.8D+00 | 2.474D+00 | 3.154D+05 |
| 6 | 7 | 29 | 1555 | bnd | 0 | 1.0D+00 | 1.6D+00 | 2.500D+00 | 3.152D+05 |
| 7 | 8 | 13 | 1561 | bnd | 0 | 1.0D+00 | 1.3D+00 | 2.494D+00 | 3.151D+05 |
| 8 | 9 | 12 | 1569 | bnd | 0 | 1.0D+00 | 9.3D-01 | 2.500D+00 | 3.150D+05 |
| 9 | 10 | 8 | 1571 | bnd | 0 | 1.0D+00 | 1.1D+00 | 2.474D+00 | 3.149D+05 |
| 10 | 11 | 10 | 1574 | bnd | 0 | 1.0D+00 | 1.0D+00 | 2.500D+00 | 3.148D+05 |
| 11 | 12 | 4 | 1572 | bnd | 0 | 1.0D+00 | 1.2D+00 | 2.500D+00 | 3.146D+05 |
| 12 | 13 | 13 | 1578 | bnd | 0 | 1.0D+00 | 9.3D-01 | 2.500D+00 | 3.144D+05 |
| 13 | 14 | 2 | 1575 | bnd | 0 | 1.0D+00 | 6.3D-01 | 2.497D+00 | 3.143D+05 |
| 14 | 15 | 16 | 1589 | bnd | 0 | 1.0D+00 | 1.3D+00 | 2.495D+00 | 3.140D+05 |

STOP: Total no. of F and G evaluations exceeds limit

Total User time 3.177E+04 seconds.

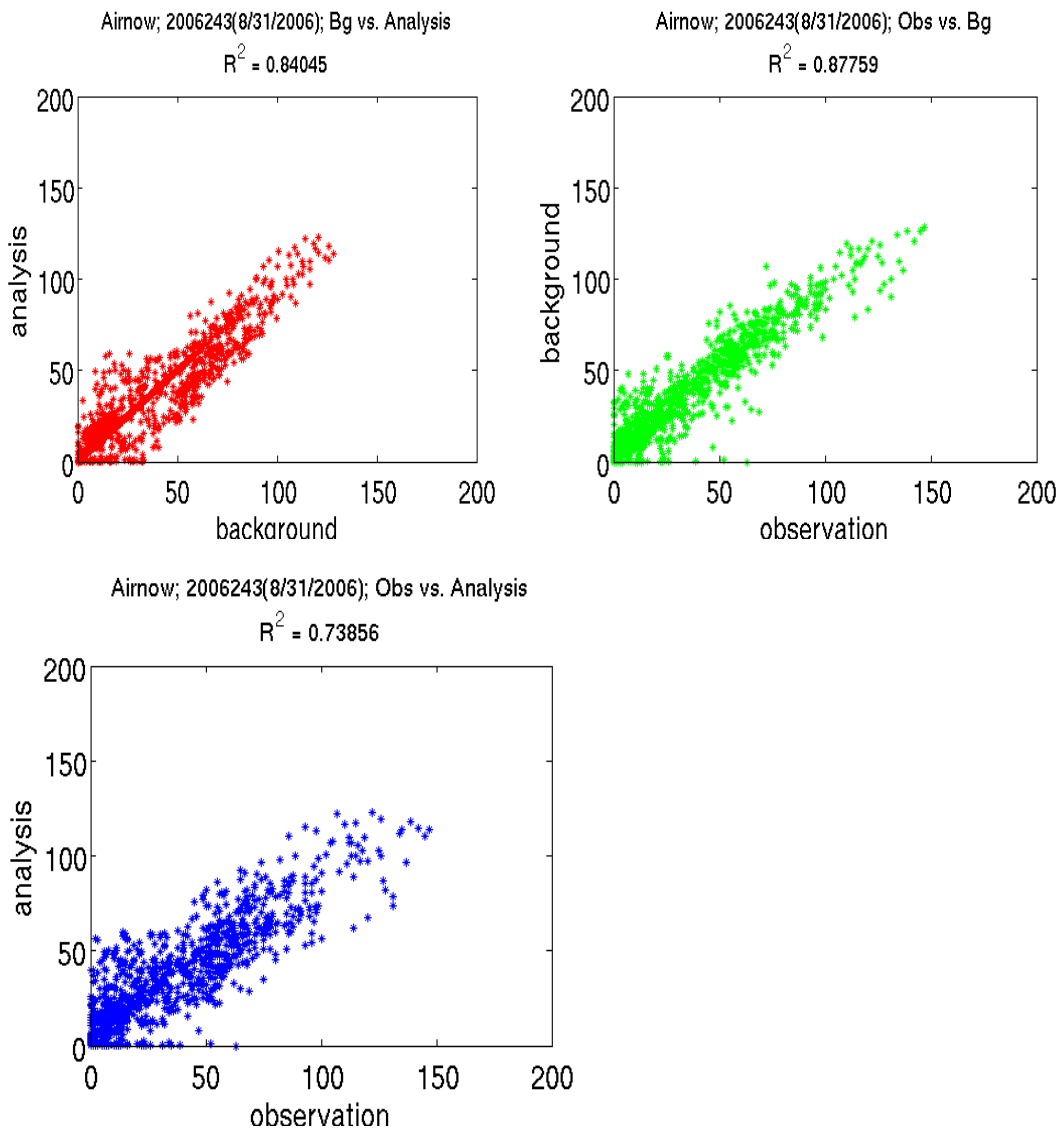


Figure 4.24 Scatter plots for Aug. 31, 2006.

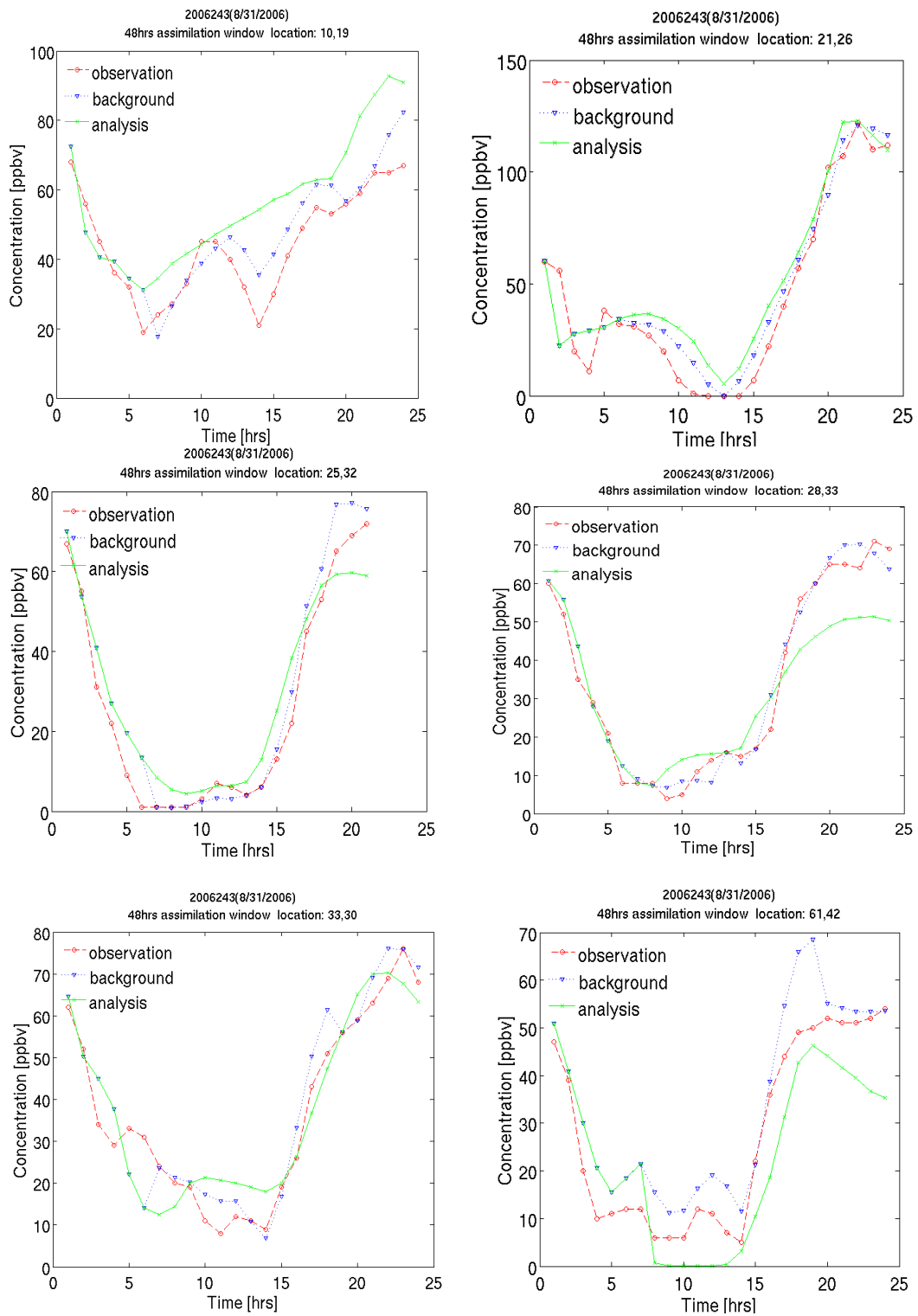


Figure 4.25 Time series for Aug. 31, 2006.

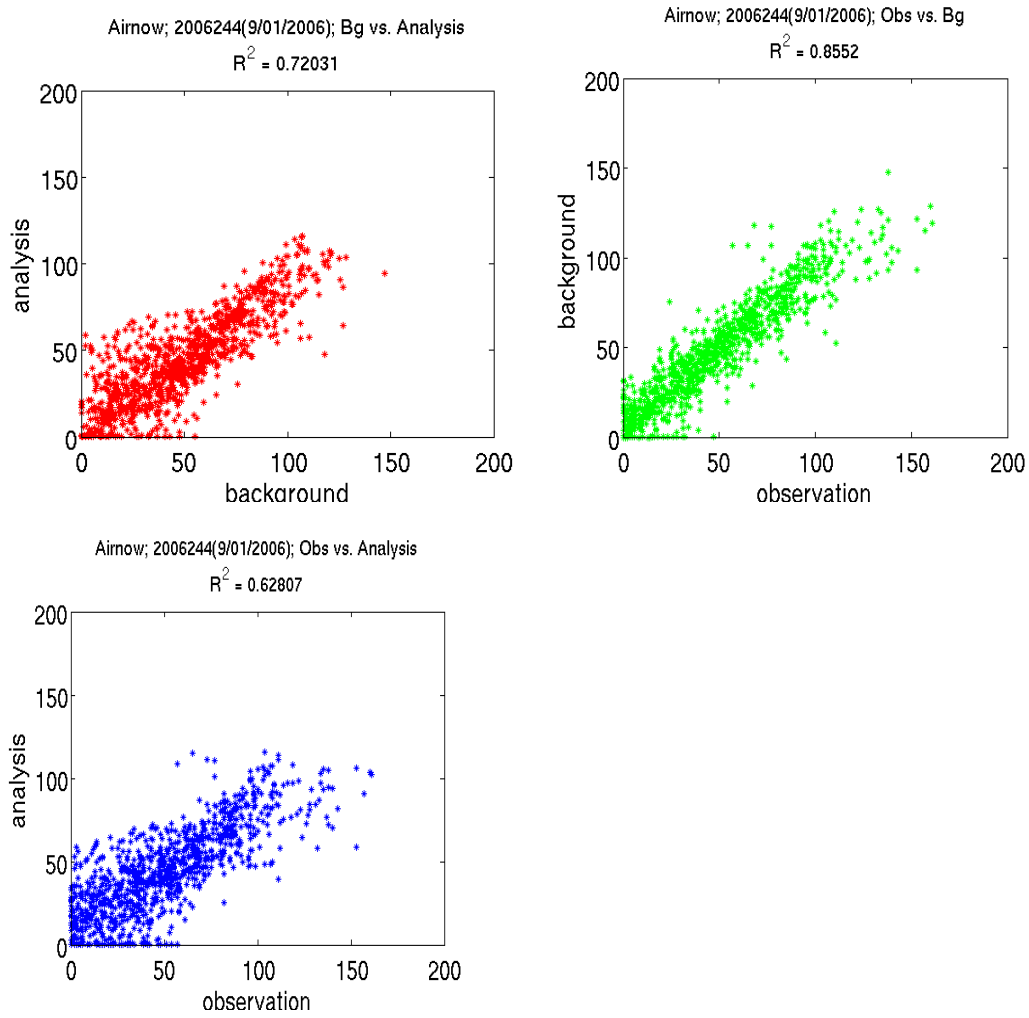


Figure 4.26 Scatter plots for Sep. 1, 2006.

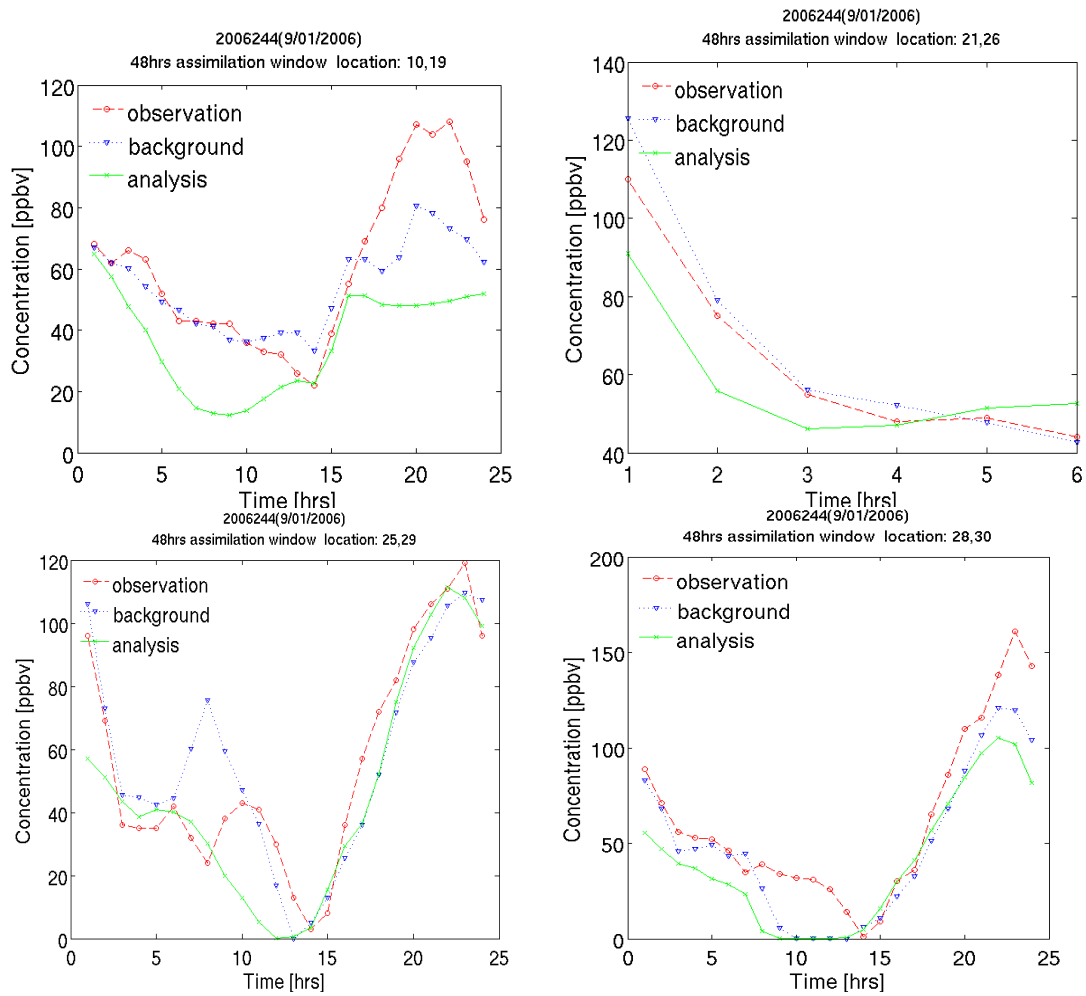


Figure 4.27 Time series for Sep. 1, 2006.

4.2.4 References

Chai, T., G. R. Carmichael, Y. Tang, A. Sandu, M. Hardesty, P. Pilewskie, S. Whitlow, E. V. Browell, M. A. Avery, P. Nedelec, J. T. Merrill, A. M. Thompson, and E. Williams, 2007: "Four Dimensional Data Assimilation Experiments with ICARTT (International Consortium for Atmospheric Research on Transport and Transformation) Ozone Measurements," *J. Geophys. Res.*, **112**, D12S15, doi:10.1029/2006JD007763