# Development of GEOS-Chem v7 Adjoint

**********

June 2008

*Prepared for*

***AIST Project, NASA JPL***
***Lead by Dr. Meemong Lee***
***Dr. Kevin Bowman***

*Prepared by*

**Dr. Adrian Sandu**
**Kumaresh Singh, Ph.D. candidate**
**Paul Eller, M.S. candidate**

Computational Science Laboratory
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060

E-mail: sandu@cs.vt.edu
Phone: 540-231-2193
Fax: 540-231-9218

# 1 Executive Summary

This report presents the work performed by Virginia Tech to date to develop the adjoint of GEOS-Chem v7. The following elements have been addressed.

**Implementation of GEOS-Chem chemistry using the Kinetic PreProcesor KPP.**
We have developed new tools that allow to seamlessly implement GEOS-Chem chemistry using the Kinetic PreProcesor KPP. These tools include: (1) a first Perl parser to translate SMVGEAR inputs to KPP input files that describe the chemical mechanism and the list of species and reactions; (2) an extension of KPP syntax to instruct KPP to generate code specific for GEOS-Chem; (3) a second Perl parser to modify GEOS-Chem and it to call directly KPP-generated code; (4) modified chemical drivers that call KPP chemistry and automatically remap species and reactions between GC and the KPP-generated code.

The GEOS-Chem developers now can: (1) generate the Fortran simulation code for any chemical mechanism, and use it within GEOS-Chem; (2) take advantage of KPP's highly efficient stiff ODE solvers (Rosenbrock, Runge-Kutta, Liner multistep), and including the mechanism-specific sparse linear algebra routines; (3) take advantage of the highly efficient tangent linear and discrete adjoint solvers that have been already developed and are available in the KPP library.

**Full GEOS-Chem adjoint.**
We have implemented the full adjoint of GEOS-Chem which includes the following elements: (1) discrete adjoints of GEOS-3 and GEOS-4 convection routines; (2) discrete adjoint of dry deposition modules; (3) discrete adjoint derivatives with respect to emissions; (4) continuous adjoint of the advection routines (the adjoint advection PDE is solved with the same routine as the forward model advection PDE); (5) checkpointing mechanisms for the concentration fields, selected meteorological fields, and the reaction rate coefficients. Drivers are provided for checking the accuracy of the adjoint derivatives against finite differences. Extensive validation results are presented in this report.

**4D-Var data assimilation.**
We have implemented a 4D-Var data assimilation system that combines the GEOS-Chem adjoint with a numerical optimization routine (L-BFGS). The driver provided here performs 4D-Var in a twin experiment framework using synthetic data obtained from a reference model run; data assimilation results in this setting and the convergence of the optimization procedure are documented in this report. Real data assimilation studies can be done using this computational framework. In addition to the framework, for each particular real data assimilation study one needs to implement: (1) the observation operators that map the GEOS-Chem model space to the particular space of observations; and (2) more realistic model of background and observation error covariances.

**Parallelization aspects.**
The original GEOS-Chem v7 offers a shared memory parallel implementation based on OpenMP. Our implementation of the GEOS-Chem adjoint is also OpenMP parallel. New implementation issues include: (1) the development of parallel code with KPP-generated chemistry; and (2) the parallelization of the adjoint (backward in time) calculations. Parallel efficiency results for the new code are documented in this report.

# 2 Implementation of GEOS-Chem Chemistry with KPP

We first discuss the KPP based implementation of the chemical solver in GEOS-Chem; this alternative implementation allows the use of additional numerical integrators (Rosenbrock, Runge Kutta, Linear multistep) and provides the needed infrastructure for the implementation of the chemical adjoint.

The GEOS-Chem native chemistry solver is the Sparse Matrix Vectorized Gear (SMVGEAR) code which implements backward differentiation formulas (BDF) and uses sparse-matrix operations to reduce the CPU time associated with the solution of linear systems.

We have implemented the chemistry simulations in GEOS-Chem using the Kinetic Pre-Processor (KPP) [Sandu et al., 2003]. KPP provides a library of several Rosenbrock and Runge-Kutta chemical solvers, together with their tangent linear and adjoint integrators. KPP provides very effective sparse matrix computational kernels, which lead to high computational efficiency. The use of KPP in GEOS-Chem allows the user to:

- Automatically generate the Fortran simulation code for any chemical mechanism, and use it within GEOS-Chem;

- Take advantage of the highly efficient stiff ODE solvers implemented in KPP (Rosenbrock, Runge-Kutta, Liner multistep), and include the mechanism-specific sparse linear algebra routines;

- Take advantage of the tangent linear and discrete adjoint ODE solvers that have been already developed and are available in the KPP library.

We next review the main steps taken for the implementation of the KPP chemistry within GEOS-Chem.

## 2.1 Automatic translation of SMVGEAR inputs to KPP inputs using *"geos2kpp_parser.pl"*

KPP requires the description of the chemical mechanism in terms of the chemical species (specified in the **.spc* file) and the chemical equations (specified in the **.eqn* file). The syntax of these files is KPP specific, but is simple and intuitive. A simulation definitions file *(*.def*) sets the simulation parameters (like initial conditions and simulation time) and specifies the chemical species and chemical equations associated with the simulation. Based on this input KPP generates all the routines needed to carry out the numerical simulation of the chemical kinetic evolution with the numerical integrator of choice.

GEOS-Chem SMVGEAR uses the chemical mechanism specified in the input file "globchem.dat"; this file contains the list of species with their initial values and the list of chemical reactions. To use KPP one needs to translate the chemical mechanism specification from the SMVGEAR input format (in "globchem.dat" file) to the KPP input format (in "globchem.spc" and "globchem.eqn" files).

We have developed the Perl parser "*geos2kpp_parser.pl*" that reads in the "globchem.dat" file, parses the information about the chemical mechanism, and automatically generates the KPP input files "globchem.spc" and "globchem.eqn" which describe the same mechanism in KPP syntax. The use of the

Perl parser is illustrated below:

```
[user@local~] $ perl -w geos2kpp_parser.pl globchem.dat

Parsing globchem.dat
Creating globchem.def, globchem.eqn, and globchem.spc
```

Once these files are ready, the user runs KPP to generate the Fortran code that describes – and solves - the chemical mechanism.

## 2.2 Creation of KPP model to interface with GEOS-Chem

The next step involves creating the chemical mechanism using KPP. KPP requires the addition of specific subroutines and modification of previously implemented subroutines to correctly interface with GEOS-Chem. We created a new input command for KPP to specify that the model needs to interface with GEOS-Chem. The user adds the command #GEOSCHEM to the *.kpp input file, along with specifying details such as the name of the model created in the first step and integrator to use.

The KPP input file “gckpp.kpp” is prepared as follows:

```
#MODEL          globchem
#INTEGRATOR     Rosenbrock
#LANGUAGE       Fortran90
#DRIVER         none
#GEOSCHEM       on
```

The user then runs KPP with this input file

```
 [user@local~] $ kpp globchem.kpp
```

to generate all the Fortran subroutines associated with the globchem chemical mechanism (ODE function, Jacobian, Hessian, sparsity information, etc.) together with the ODE solver of Rosenbrock type. (This solver choice can be replaced by SDIRK, Runge-Kutta, etc.). The files generated by KPP are (automatically) named using the prefix gckpp_* for the forward model and the prefix gckpp_adj_* for the adjoint model. The set of files (the model) created by KPP is then copied into the GEOS-Chem source code directory.

In order to replace the call to SMVGEAR with a call to KPP generated code within GEOS-Chem additional changes are needed in both the generated KPP files and GEOS-Chem's chemistry subroutines. These are discussed next.

## *2.3* Automatic modification of GEOS-Chem to interface with KPP using *"gckpp_parser.pl"*

We have developed the Perl parser "gckpp_parser.pl" to automatically modify the relevant GEOS-Chem code to interface with KPP. This parser is run once in the GEOS-Chem code directory to modify existing files and create new files. Files are modified to call KPP generated code instead of SMVGEAR for the chemistry step and to add new subroutines which are used for the adjoint calculations. New files are created to assist with these calculations.

The use of this Perl parser is illustrated below:

```
[user@local~] $ perl -w gckpp_parser.pl

Modifying and creating GEOS-Chem files
Modifying time_mod.f
...
Modifying checkpoint_mod.f
Creating Makefile.ifort.4d
...
Creating dpmeps.f
Done modifying GEOS-Chem files
```

The resulting modified version of GEOS-Chem now uses the KPP generated chemistry code. Makefiles are included for the finite-difference driver, the sensitivity driver, and the 4D-Var driver.

In order to replace SMVGEAR with KPP further changes are needed in the generated KPP files and GEOS-Chem's chemistry subroutines. The chemical species and the chemical reactions have different indices in the KPP generated code than in the original SMVGEAR code. One transformation in the new GEOS-Chem code is the remapping of species and equations. This is discussed next.

## 2.4 Mapping of Chemical Species between GC and KPP

At the beginning of each time step the chemical concentrations for each grid cell are copied from the three-dimensional GC data structures into the KPP data structures. At the end of the time step the time-evolved chemical concentrations – for the particular grid cell - are copied from KPP code back into GC.

Since KPP performs a reordering of the species to enhance sparsity gains the chemical species indices in KPP are different than their indices in GC. The KPP-generated “kpp_Util.f90” file provides a subroutine `Shuffle_user2kpp` that maps the GEOS-Chem species array to KPP species array. The inverse mapping subroutine `Shuffle_kpp2user` is also generated automatically by KPP. This subroutine is used to initialize the VAR and FIX species of KPP with the CSPEC array values (for each grid cell JLOOP) in SMVGEAR.

`kpp_Initialization.f90`

```
USE gckpp_Util,      ONLY : Shuffle_user2kpp

CALL Shuffle_user2kpp(V_CSPEC,VAR)

DO i = 1, NFIX
   FIX(i) = 1.0d0
END DO
```

The FIX-ed species are assigned the value 1.d0. The reason is that the reaction rates are already multiplied with FIX-ed species concentrations in the `CALCRATE` subroutine.

## 2.5 Mapping of Chemical Reactions between GC and KPP

The reaction rates calculated every chemistry time step by the `CALCRATE` subroutine need to be copied for usage by KPP. In order to assign the reaction rates RRATE from SMVGEAR to RCONST in KPP, the RRATE vector is saved in a temporary vector RRATE_FORKPP in the `CALCRATE` subroutine and then reshuffled and assigned to RKPP vector in `PHYSPROC` subroutine. While storing the RRATE vector in `CALCRATE`, the mapping indices are stored in a vector IND(:) in the following way.

`Subroutine CALCRATE`

```
      I = 1                                           ! ‘courtesy: Daven Henze’
      DO NK          = 1, NTRATES(NCS)
         DO KLOOP      = 1, KTLOOP
            IF ( NEWFOLD(NK,NCS) > 0 ) THEN
               IF(KLOOP.eq.1)THEN
                  IND(I) = NK
                  I = I +1
               ENDIF
               RRATE_FORKPP(KLOOP,NK)   = RRATE(KLOOP,NEWFOLD(NK,NCS))
            ENDIF
         ENDDO
      ENDDO
```

`Subroutine PHYSPROC`

```
      DO KLOOP = 1, KTLOOP                            ! ‘courtesy: Daven Henze’
         JLOOP         = JREORDER(JLOOPLO+KLOOP)
         DO NK          = 1, NTRATES(NCS)
```

```
            RKPP(JLOOP,NK) = RRATE_FORKPP(KLOOP,NK)
         ENDDO
      ENDDO
```

Once, the RKPP vectors are saved in the PHYSPROC subroutine, they are mapped and assigned to RCONST (KPP's reaction rate vector). The automatic assignment of the RCONST vector done by KPP software is then commented out.

```
kpp_Rates.f90

USE COMODE_MOD

INTEGER :: N

DO N = 1, NREACT
   RCONST(N) = RKPP(JLOOP,IND(N))
ENDDO
```

## 2.6 The GEOS-Chem Chemical Driver

The kpp_Global.f90 and kpp_Parameters.f90 files are modified accordingly. Also, the Integrator subroutine is modified to carry out layer-wise integration.

```
Subroutine GCKPP_DRIVER()
DO i=1,NVAR
       RTOL(i) = 1.0d-4
       ATOL(i) = 1.0d-3
END DO

ICNTRL(:) = 0
RCNTRL(:) = 0.d0

ICNTRL(1) = 1    ! Autonomous

DT = GET_TS_CHEM() * 60.d0

T = 0d0
TIN = T

TOUT = T + DT

! Solve Chemistry

DO JJLOOP = 1,NTT

   JLOOP = JJLOOP
   ! Get 3D coords from SMVGEAR's 1D coords
   I = IXSAVE(JJLOOP)
   J = IYSAVE(JJLOOP)
   L = IZSAVE(JJLOOP)
   DO N =1, NVAR
      V_CSPEC(N) = CSPEC_FORKPP(JLOOP,N)
   END DO

   ! Pass tracer concs from V_CSPEC to VAR, FIX.
   CALL Initialize()

   ! Recalculate rate constants
   CALL Update_RCONST()    !*******************!

   CALL INTEGRATE( TIN, TOUT, ICNTRL, RCNTRL,
   &        ISTATUS, RSTATE)

   ! Set negative values to SMAL2
   DO N = 1, NVAR
      VAR(N) = MAX(VAR(N),SMAL2)
   ENDDO

   CALL Shuffle_kpp2user(VAR,V_CSPEC)

   DO N =1, NVAR
      CSPEC(JLOOP,N) = V_CSPEC(N)
   END DO

ENDDO

END SUBROUTINE GCKPP_DRIVER
```

The call to `SMVGEAR` in `PHYSPROC` subroutine is commented out and `GCKPP_DRIVER` subroutine is called in *`chemdr.f`* file to carryout KPP chemistry.

## 2.7 Validation of KPP Chemistry Implementation vs. SMVGEAR

In order to illustrate the correctness of GEOS-Chem simulations using KPP chemistry we compare the simulation results against the original simulation with SMVGEAR. The only difference between the two simulations is the implementation of gas phase chemistry, but all other code parameters and input files are identical. GEOS-Chem is run twice for 48 hrs. Concentrations are checkpointed every hour and then compared. GAMAP plots for 0th and 48th hour for CO, NOx, Ox and RCHO are provided below for comparison.

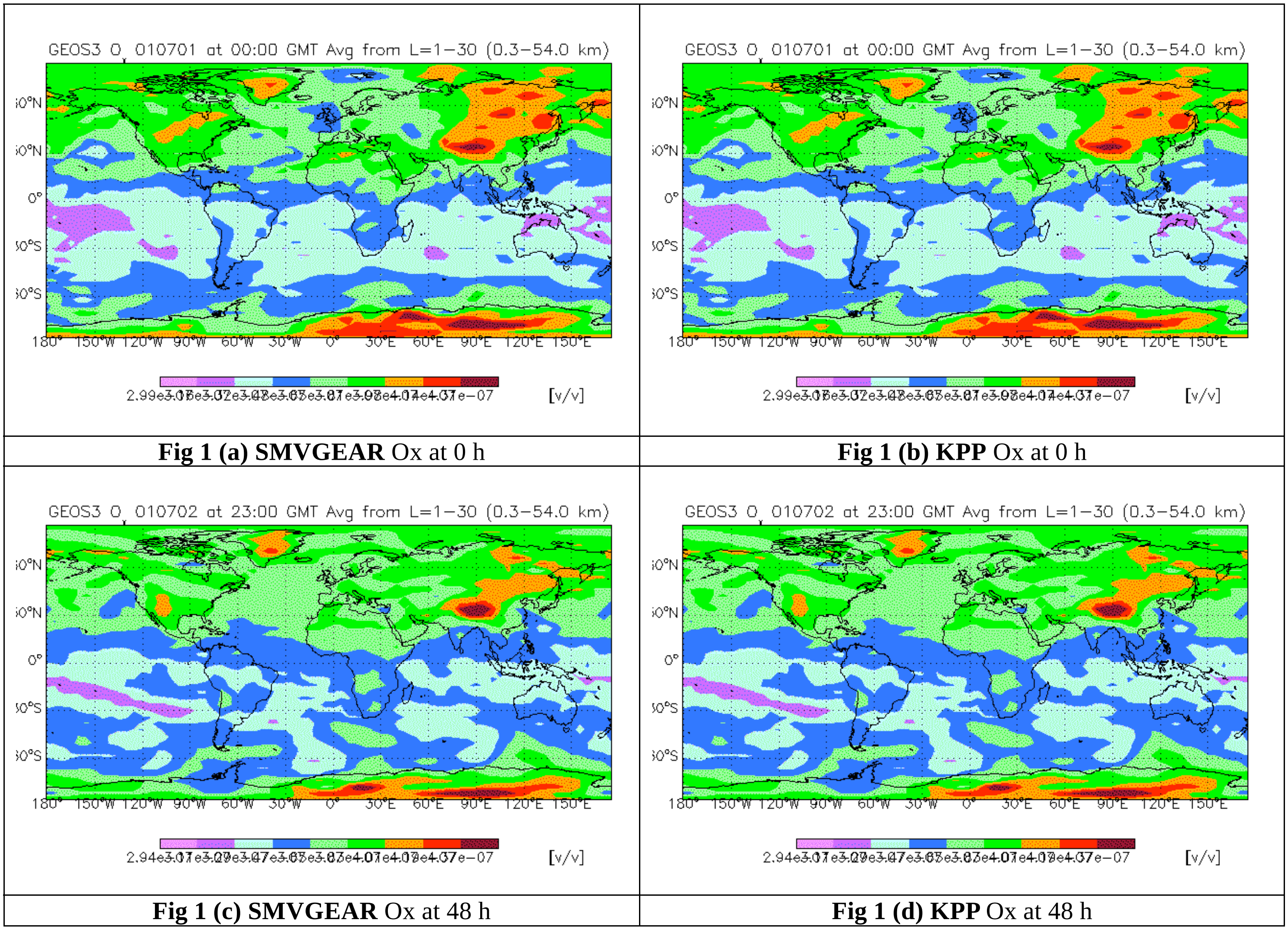

GEOS3 Ox 010701 at 00:00 GMT Avg from L=1–30 (0.3–54.0 km)
60°N
30°N
0°
30°S
60°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
2.99e3.0Be3.0Ze3.4Be3.6Be3.8Te3.9Be4.0Te4.3Te–07
[v/v]
GEOS3 Ox 010702 at 23:00 GMT Avg from L=1–30 (0.3–54.0 km)
2.94e3.0Te3.2Ge3.47e3.6Be3.8Ze4.0Te4.0Ge4.37e–07
[v/v]

**Fig 1 (a) SMVGEAR** Ox at 0 h

**Fig 1 (b) KPP** Ox at 0 h

**Fig 1 (c) SMVGEAR** Ox at 48 h

**Fig 1 (d) KPP** Ox at 48 h

***Figure 1. A comparison of 48-hour Ox simulation results obtained with KPP generated code and with SMVGEAR. The GAMAP filled contour plots use data averaged over all 30 layers. The plots are visually identical.***

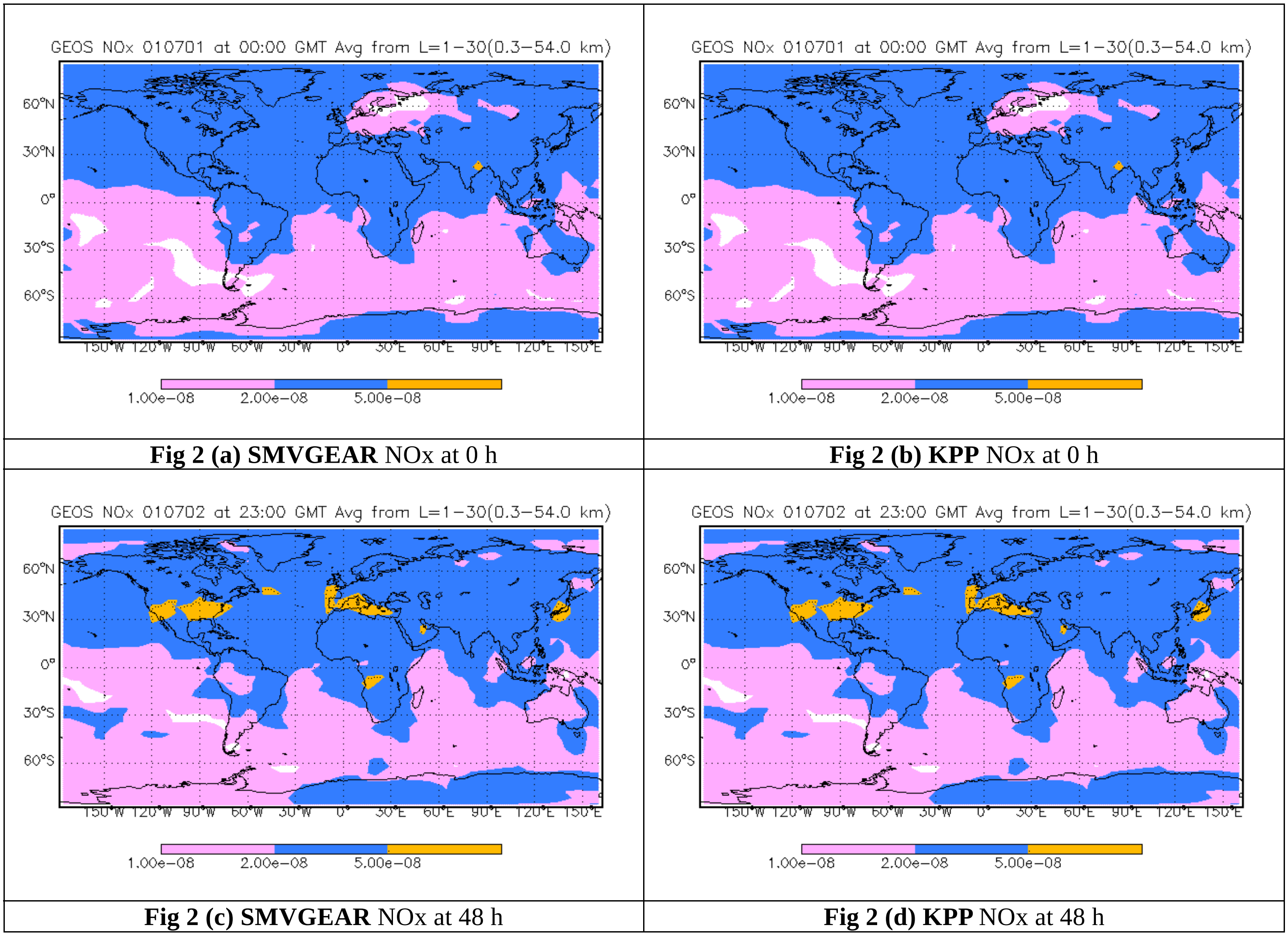

GEOS NOx 010701 at 00:00 GMT Avg from L=1−30(0.3−54.0 km)
GEOS NOx 010701 at 00:00 GMT Avg from L=1−30(0.3−54.0 km)
GEOS NOx 010702 at 23:00 GMT Avg from L=1−30(0.3−54.0 km)
GEOS NOx 010702 at 23:00 GMT Avg from L=1−30(0.3−54.0 km)
60°N
30°N
0°
30°S
60°S
150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
1.00e−08
2.00e−08
5.00e−08

**Fig 2 (a) SMVGEAR** NOx at 0 h

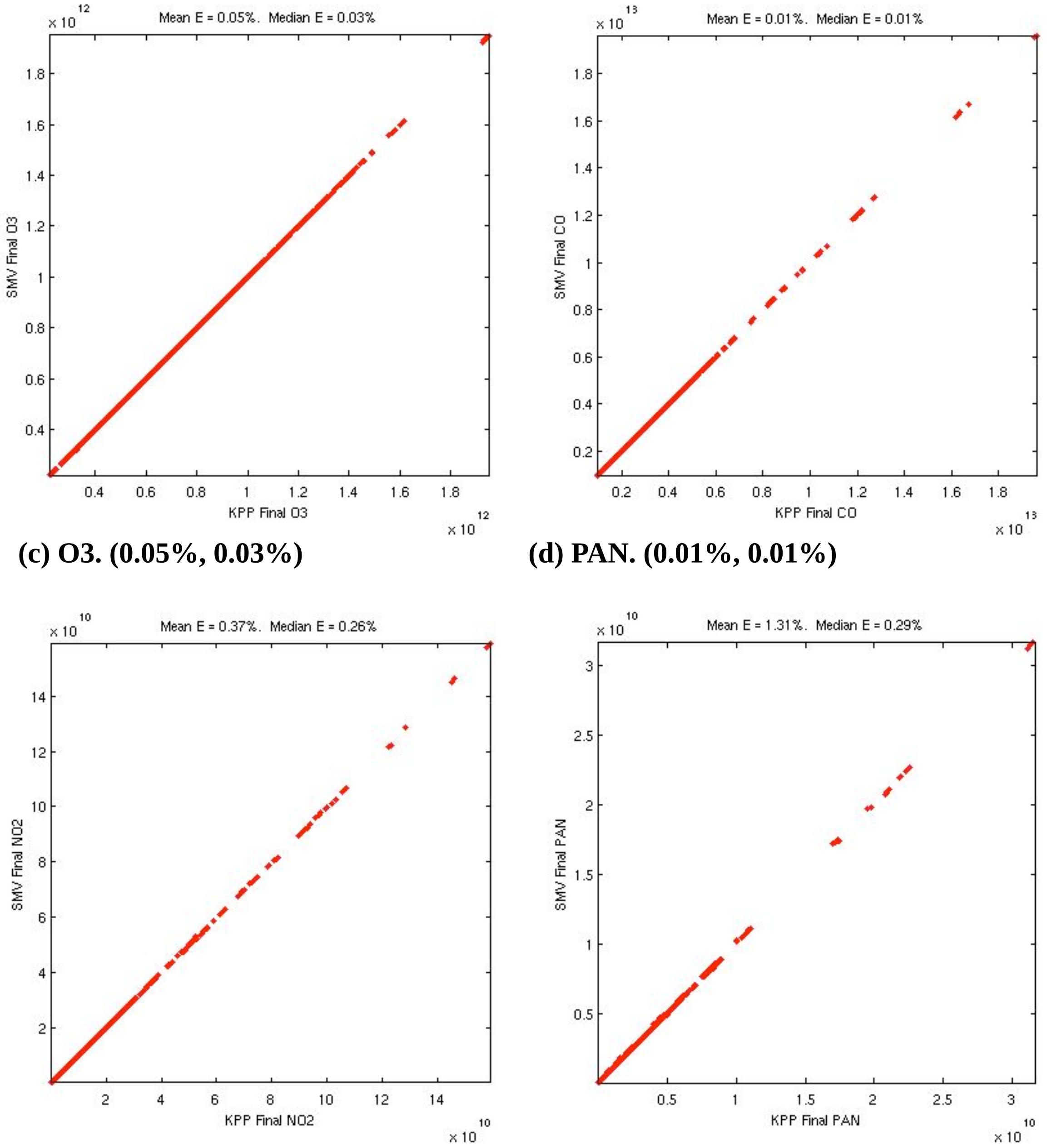
**Fig 2 (b) KPP** NOx at 0 h

**Fig 2 (c) SMVGEAR** NOx at 48 h

**Fig 2 (d) KPP** NOx at 48 h

***Figure 2. A comparison of 48-hour NOx simulation results obtained with KPP generated code and with SMVGEAR. The GAMAP filled contour plots use data averaged over all 30 layers. The plots are visually identical.***

A more stringent validation is the scatter plot of SMVGEAR vs KPP predicted concentrations. We next consider the GEOS-Chem results using KPP and using SMVGEAR chemistry for a one-week simulation interval, between 0 GMT on 2001/07/01 and 23 GMT on 2001/07/07. Both simulations are started with the same initial conditions and they differ only in the chemistry module. The tolerances for both integrators are Atol=0.1 and Rtol=0.001. The results are shown in Figure 3 for the final predicted concentrations of O3, CO, NO2, and PAN. Each point in the scatter plots corresponds to a different grid cell. The scatterplot aligns almost perfectly with the main diagonal, indicating a very good agreement between the GEOS-Chem results obtained with the KPP and with the SMVGEAR chemistry modules.



**(c) O3. (0.05%, 0.03%)** **(d) PAN. (0.01%, 0.01%)**

**(c) NO2. (0.37%, 0.26%)** **(d) PAN. (1.31%, 0.29%)**

***Figure 3. GEOS-Chem predictions after one simulation week. The scatter plots represent the SMVGEAR results (on the y-axis) versus the corresponding KPP results (on the x-axis). Each point corresponds to one grid cell. The percentage (Mean Error, Median Error) are given for each species.***

# 3 GEOS-Chem Adjoint

In this section we discuss the implementation of adjoints for individual GEOS-Chem processes: chemistry, convection, transport, emission, and dry deposition. After the adjoint code is integrated within GEOS-Chem code, it undergoes an extensive validation. In order to test the correctness of the adjoint code we recall that adjoint variables (at the initial time) represent derivatives of a cost function J with respect to changes in the initial concentrations $c_0$

$$\lambda_0 = \frac{\partial J(c_0)}{\partial c_0}$$

Testing for adjoint correctness requires multiple code runs with different initial conditions; for each run a different value of the cost function is obtained. Consider a vector of perturbations $\Delta c_0$ applied to the initial concentrations. The adjoint variables are compared against one-sided finite differences

$$\lambda_0^T \cdot \Delta c_0 \approx \frac{J(c_0 + \varepsilon \cdot \Delta c_0) - J(c_0)}{\varepsilon}$$

or against central finite differences

$$\lambda_0^T \cdot \Delta c_0 \approx \frac{J(c_0 + \varepsilon \cdot \Delta c_0) - J(c_0 - \varepsilon \cdot \Delta c_0)}{2\varepsilon}$$

For a correct adjoint we expect the finite difference and the adjoint results to differ only by O(ε). The difference between the one-sided and central approximations represents an estimate of the accuracy of the one-sided finite difference result.

## 3.1 Chemistry Adjoint

The Kinetic Pre-Processor generates the discrete adjoint chemistry model in a similar manner as with the forward chemistry model, and using the same chemical model file globchem.dat. The KPP generated files are interfaced with the GEOS-Chem adjoint code, updating the KPP global variables, parameters and initialization files. The chemistry adjoint integrator utilizes the rate coefficients and chemical concentrations from the forward run for that hour. There are separate checkpoint files for the chemical concentrations and for the rate coefficients; the CSPEC and the RRATE arrays are written at each hour before the beginning of the forward chemical step. They are retrieved in the backward run before each chemistry forward-adjoint hourly calculation.

For the finite difference calculations we perturb the initial concentration of tracer `TRAC1` (in each grid cell) and record the final concentration of tracer `TRAC2(`in each grid cell`).` Three forward runs are performed with: (1) the nominal values of the initial concentrations, (2) with the perturbation added to the initial concentrations, and (3) with the perturbations subtracted from the initial concentrations.

```
! Perturbation is a fraction of the reference initial conditions
! Typically FEPS = 0.1
      PERT(:,:,:,TRAC1) = STT(:,:,:,TRAC1)*FEPS

      IF (P1_CASE) THEN
         STT(:,:,:,TRAC1) = STT(:,:,:,TRAC1) + PERT(:,:,:,TRAC1)
      ELSE IF (P2_CASE) THEN
         STT(:,:,:,TRAC1) = STT(:,:,:,TRAC1) - PERT(:,:,:,TRAC1)
      END IF
```

The concentrations saved in the forward runs are used to generate the one- and central finite differences of tracer concentrations TRAC2 with respect to perturbations in tracer TRAC1.

```
      fd1 = (STT_P1(I,J,L,1)-STT_NOMINAL(I,J,L,1))/( PERT(I,J,L,1))
      fd2 = (STT_P1(I,J,L,TRAC1)-STT_P2(I,J,L,1))/( 2*PERT(I,J,L,1))
```

We consider adjoint derivatives of the final concentration of the TRAC2 species with respect to the initial concentration of the TRAC1 species (the cost function J is the final concentration of TRAC2). For this the adjoint variables are initialized at the final time as follows:

```
         STT_ADJ(:,:,:,:)     = 0.0
         STT_ADJ(:,:,:,TRAC2) = 1.0
```

This adjoint concentrations STT_ADJ(:,:,:,TRAC1) at the initial time are recorded and are then tested against the one-sided (fd1) and central (fd2) finite differences.

For the validation tests we choose three derivatives: dOx/dNOx, dPAN/dNOx, and dCO/dOx in layers 1, 10 and 15. Note that the forward and adjoint calculations in each grid cell are independent of one another.

The adjoint derivatives are compared against their finite difference counterparts. Six different plots are presented for each test case:

- Adjoint variable field pixel plot over the entire globe;
- Central finite difference pixel plot over the entire globe;
- Scatter plot of adjoint values vs central finite differences;
- Filled contour relative error plot of adjoint versus one-sided;
- Filled contour relative error plot of adjoint versus central finite differences;
- Filled contour relative error plot of one-sided finite differences versus central finite differences.

The plot (vi) serves to quantify the relative error of the finite difference approximation. We consider the adjoints to be accurate if they match the finite differences within this error margin.

The relative error between quantities A and B is computed in each grid cell as follows:

$$\text{Rel.Err.}\left(A \text{ vs } B\right) = \frac{|A-B|}{\max\left(|A|,|B|,\text{tol}\right)}$$

The tolerance is a small number that prevents division by zero.

NOTE: The tracer names in the relative error plots are of no significance, since the relative errors for the three different cases (iv), (v) and (vi) are dumped to the first three species concentrations of the checkpoint file using the bpch2 subroutine.

The one-sided FD test procedure discussed here. The central finite difference test is similar, and requires an extra forward run with a negative perturbation.

- ➔ In order to perform the **box model testing** for GEOS-Chem chemistry adjoint, we choose a chemical species (say S) with respect to which the derivative of the concentration of another tracer species (say I) has to be calculated, i.e., $dC_I/dC_S$. A forward run of GEOS-Chem is performed for the time period (T) under consideration with an initial perturbation ($pertC_{S0}$) in S. The concentrations from the final time ($pertC_{IT}$) are saved for finite difference calculations.

- ➔ A similar forward run is performed but this time with original C and the concentrations are checkpointed every chemistry time step to be used in the adjoint run which follows it. The concentrations for the final time ($origC_{IT}$) are again saved.

- ➔ In the adjoint mode, the adjoint tracer array for species I are initialized with unit concentration over the grid points under consideration at the final time (T) integrating backwards to the initial time. At T=0, the adjoint tracer concentrations for species S ($adjC_{S0}$) are compared with their finite difference approximations in the following manner:

$$adjC_{S0} \approx ( pertC_{IT} - origC_{IT} ) / pertC_{S0}$$

### 3.1.1 Chemical Adjoint Validation Results

The results in this section are obtained with the GEOS-Chem v7 adjoint, chemistry-only simulation, run for one week from 2001/04/01 to 2001/04/07. Adjoint derivatives of for SO4 and O3 with respect to NOx concentrations in vertical layer 10 are presented. The results indicate a good agreement between the adjoint values (ADJ) and the finite difference approximations. Similar results have been obtained for the vertical layers 1 and 15.

NOTE: The species names on some of the graphs are not correct since they are generated from temporary checkpoint files used to save the FD test results.

1) **Results for dSO4/dNOx**

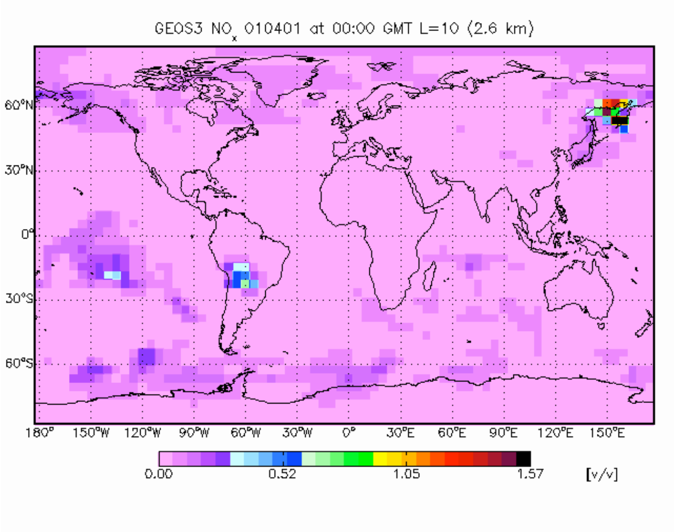


**Fig 4(a)** dSO4/dNOx ADJ at T=0

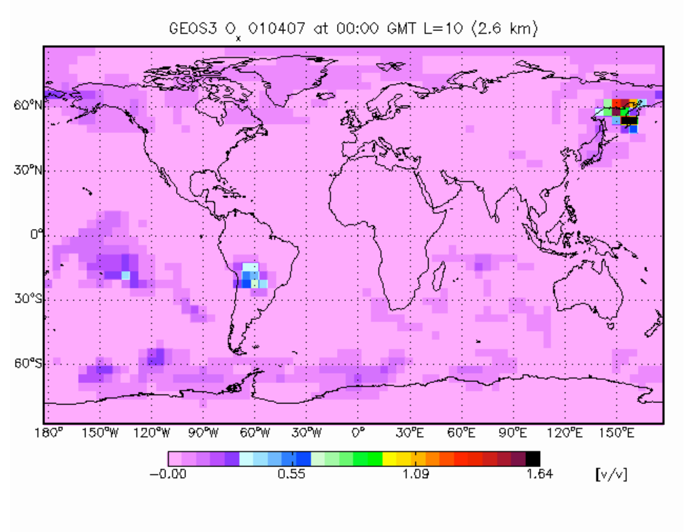


**Fig 4(b)** dSO4/dNOx Central-FD at T=0

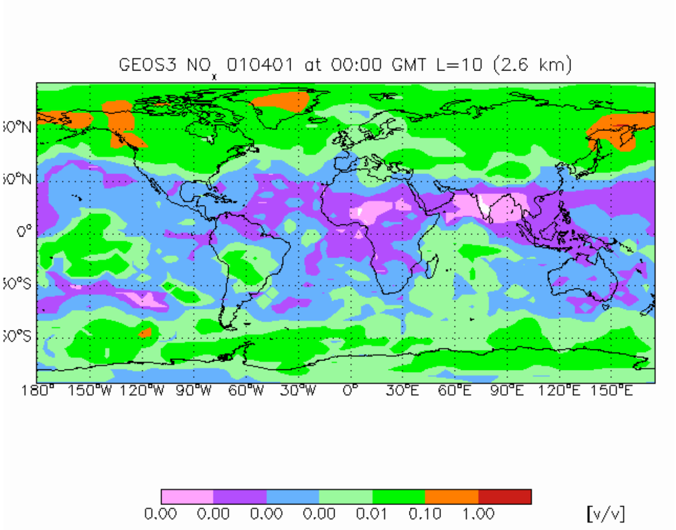


**Fig 4(c)** ADJ vs Central-FD relative error

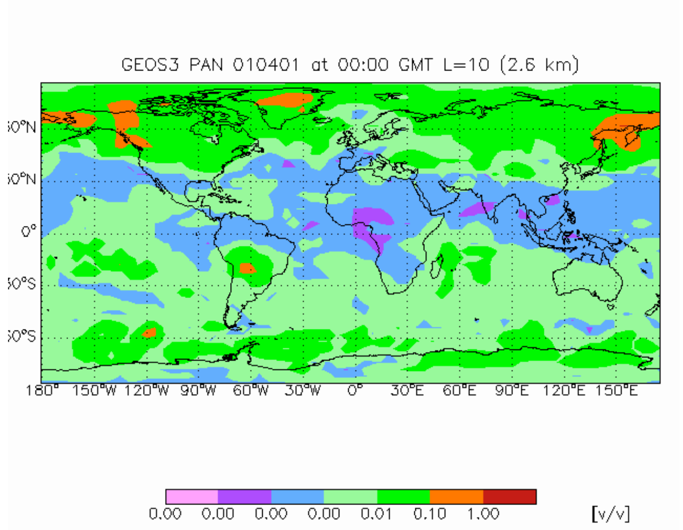


**Fig 4(d)** ADJ vs One-sided-FD relative error

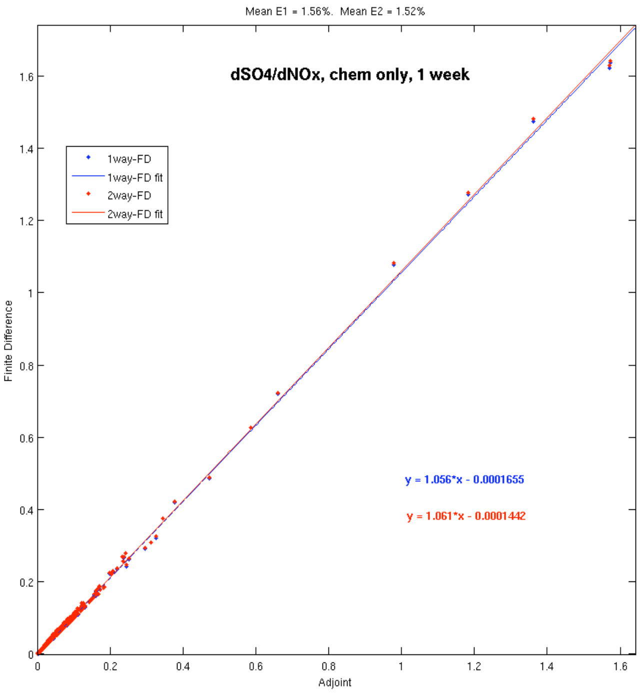


**Fig 4(e)** One-sided and Central-FD vs ADJ scatter plot with linear curve fitting. The mean errors for the two finite differences are also presented. E1 = One-sided-FD vs ADJ relative error in %.

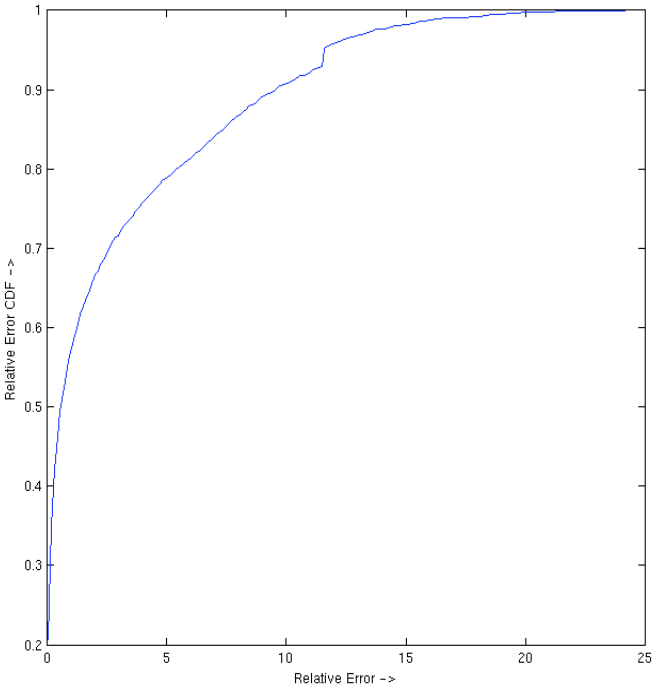


**Fig 4**(f) CDF of Central-FD and Adjoint relative error. Shows what percentages of total points are within what percentage of error.

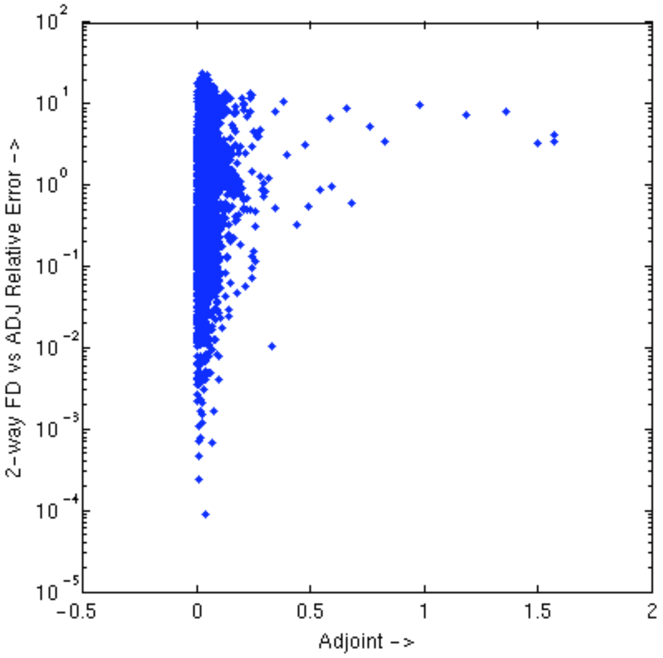


**Fig 4(g)** Semilogy plot of Central-FD to ADJ relative error vs. ADJ values.

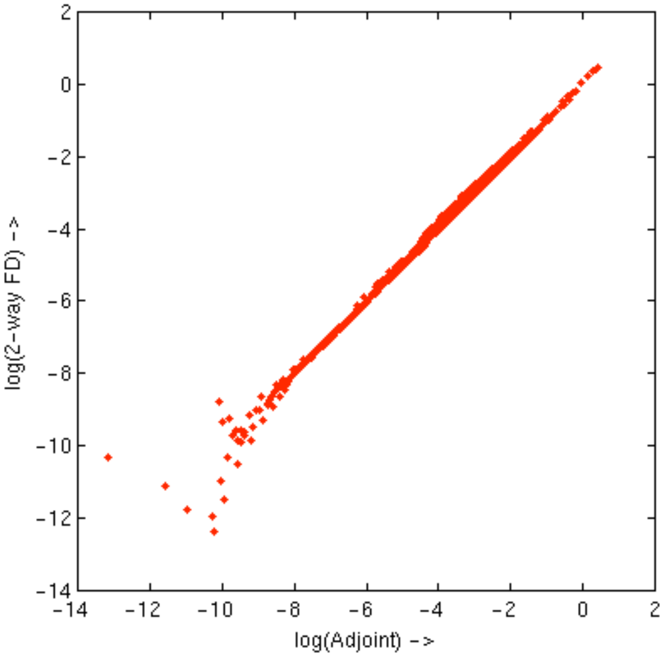


**Fig 4(h)** Log of abs(Central-FD) vs. Log of abs(ADJ)

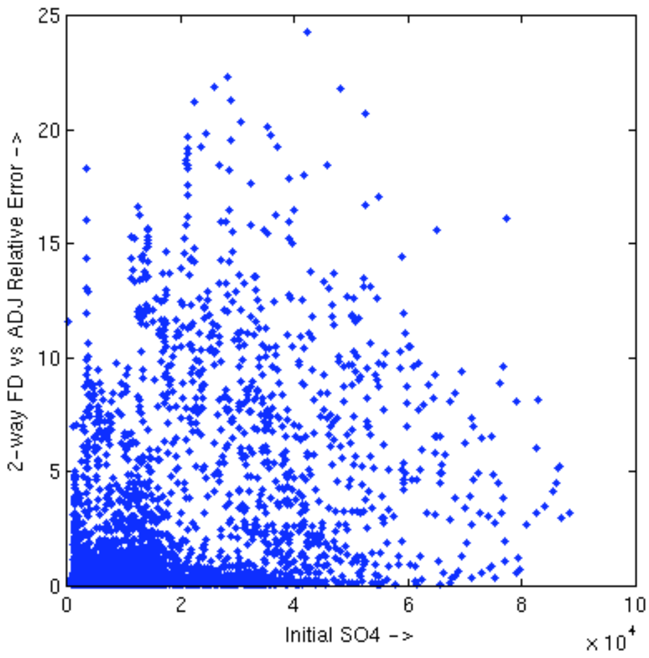


**Fig 4(i)** Central-FD to ADJ relative error vs. initial SO4 concentrations.

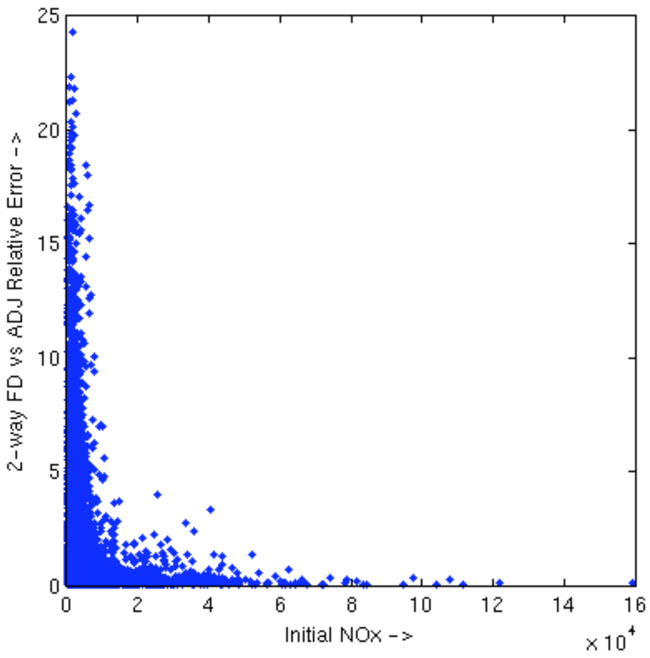


**Fig 5(j)** Central-FD to ADJ relative error vs. initial NOx concentrations.

## 2) Results for dO3/dNOx

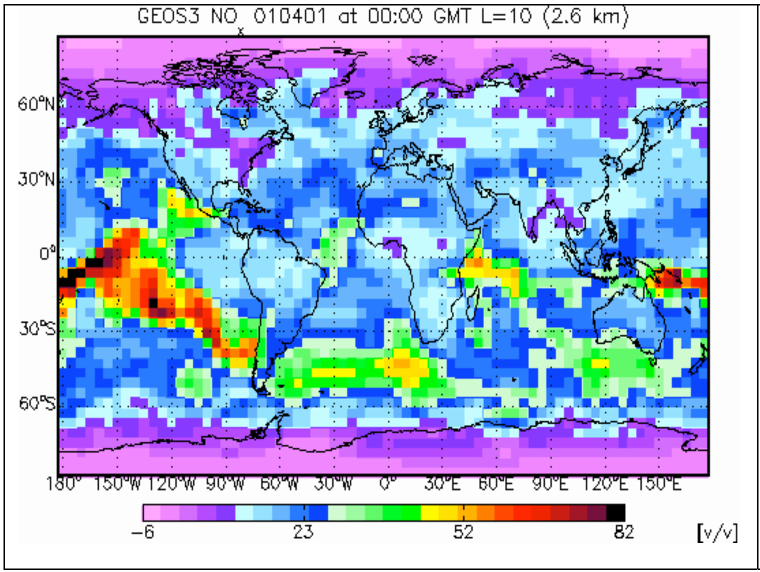

GEOS3 NO_x 010401 at 00:00 GMT L=10 (2.6 km)
60°N
30°N
0°
30°S
60°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
–6
23
52
82
[v/v]

**Fig 5(a)** dO3/dNOx ADJ at T=0

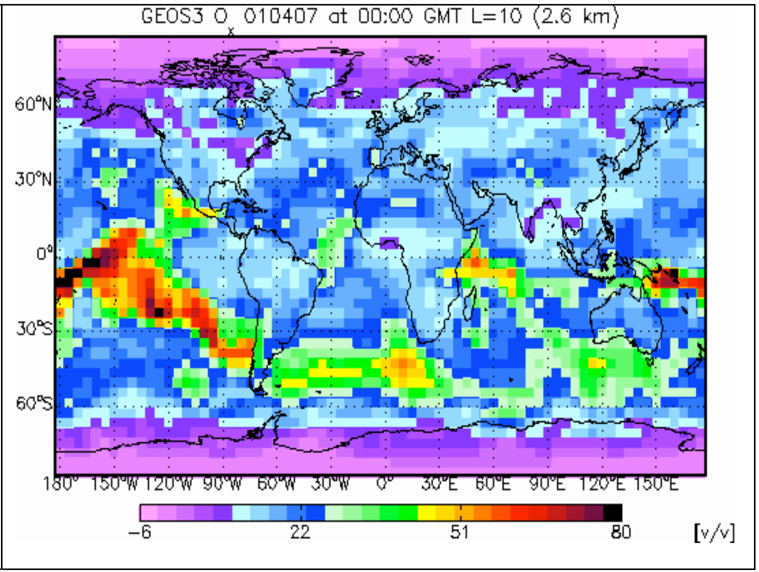
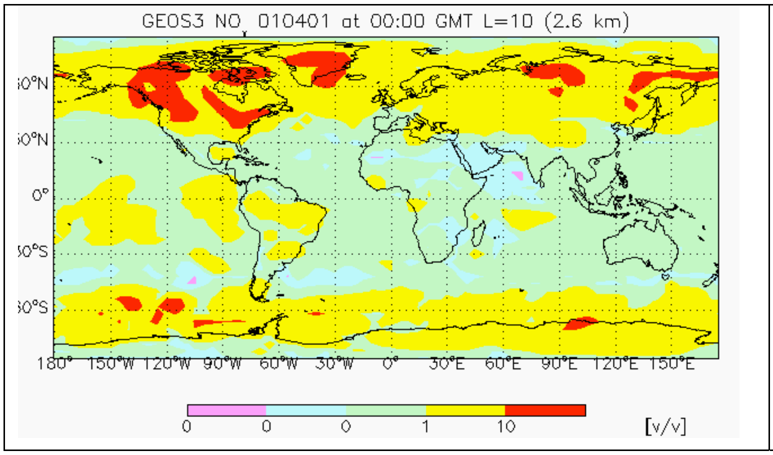

GEOS3 O_x 010407 at 00:00 GMT L=10 (2.6 km)
60°N
30°N
0°
30°S
60°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
–6
22
51
80
[v/v]

**Fig 5(b)** dO3/dNOx Central-FD


GEOS3 NO 010401 at 00:00 GMT L=10 (2.6 km)
60°N
30°N
0°
30°S
60°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
0 0 0 1 10
[v/v]

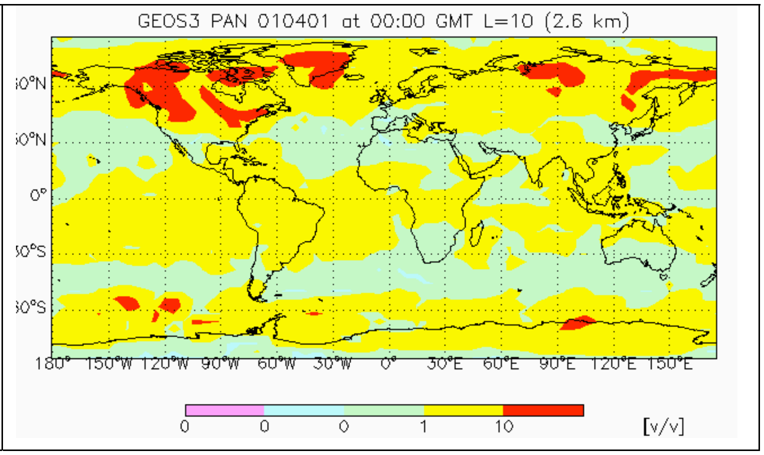
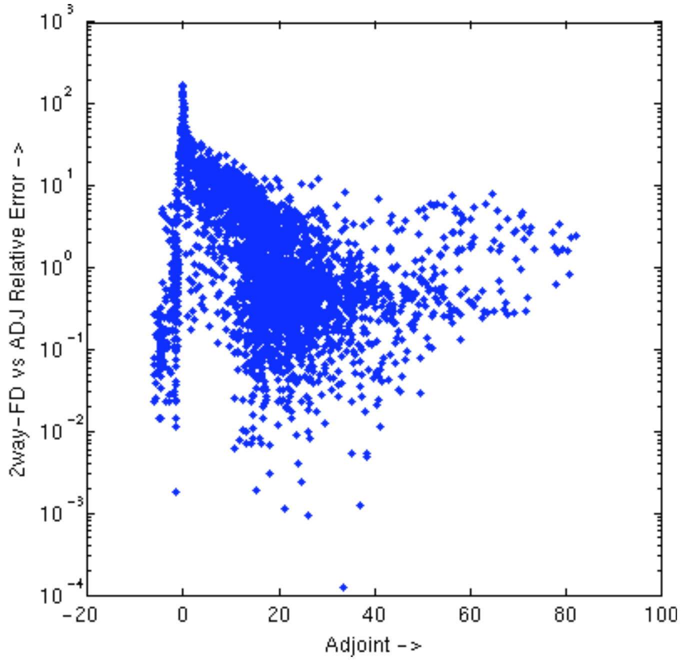

GEOS3 PAN 010401 at 00:00 GMT L=10 (2.6 km)
60°N
30°N
0°
30°S
60°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
0 0 0 1 10
[v/v]

**Fig 5(c)** ADJ vs Central-FD relative error filled filled contour plot

**Fig 5(d)** ADJ vs One-sided-FD relative error contour plot



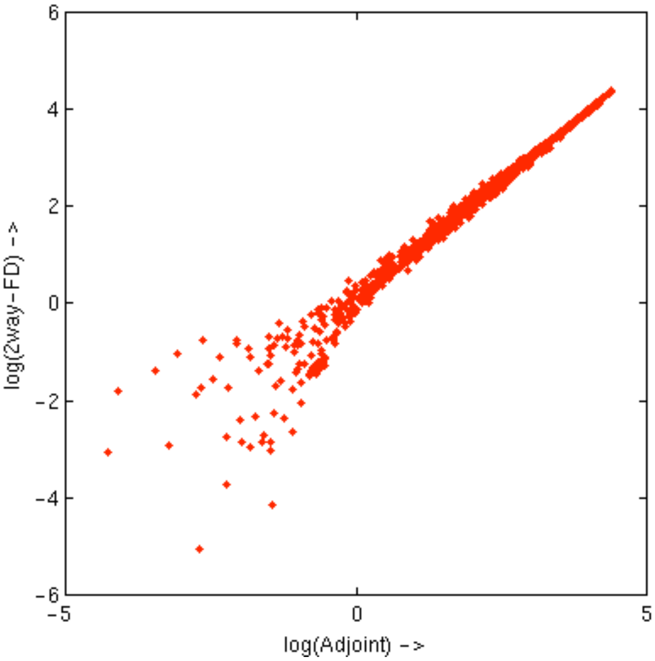


**Fig 5(e)** Semilogy plot of Central-FD and ADJ relative error vs. Adjoint values.

**Fig 5(f)** Log of abs(Central-FD) vs. Log of abs(ADJ)

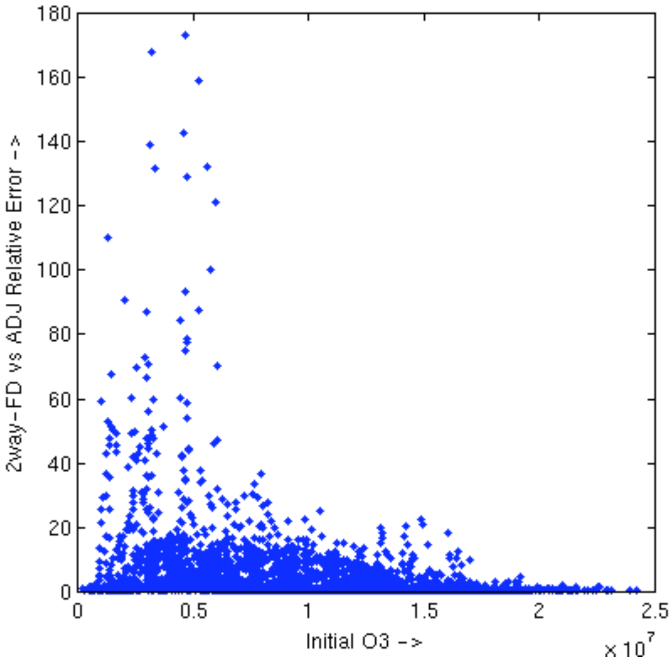


**Fig 5(g)** Central-FD to ADJ relative error vs. initial concentrations.

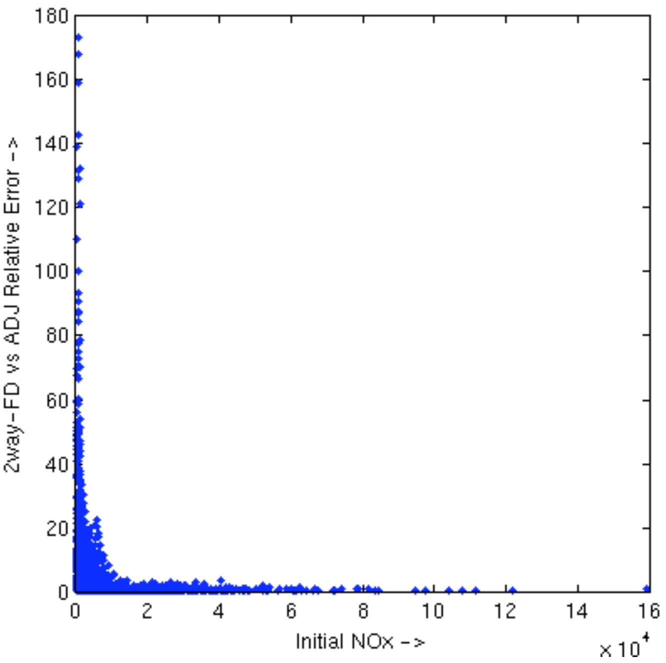


**Fig 5(h)** Central-FD to ADJ relative error O3 vs. initial NOx concentrations.

## 3.2 Convection Adjoint

Here we discus the one-sided FD test procedure; the central FD setup is similar, and requires one extra forward run with a different perturbation. The results we present include the comparison of adjoint vs. central finite difference approximations.

Geos-3 and Geos-4 convection subroutines are quite different and their adjoints have been handled separately. The discrete adjoints for each of these convection routines have been generated using TAMC. The results presented below show a good agreement between adjoints and the finite difference approximations. The execution times for GEOS-3 and GEOS-4 convection adjoints are similar to the GEOS-3 and GEOS-4 forward convection, respectively.

- ➔ The **column testing** for GEOS-Chem convection adjoint for a chemical species (S) is performed as follows. Consider two layers L and H, with H>L. We will calculate the derivative of the concentration of S at the high level and final time with respect to the concentration of the same species at the low level and initial time: $dS_T^H/dS_0^L$. A forward run of GEOS-Chem is performed for the time period (T) under consideration with an initial perturbation ($pertS_0^L$) added to S in layer L only. The concentrations at the final time ($pertS_T^H$) are saved for finite difference calculations.

- ➔ A similar forward run is performed but this time with original S and the concentrations are checkpointed every convection time step to be used in the adjoint run which follows it. The concentrations for the final time ($origS_T^H$) are again saved.

- ➔ In the adjoint mode, the adjoint variables associated with species S are initialized with unit concentration over layer H at the final time (T). The adjoint convection model is run backward in time. At T=0, the adjoint variables for species S ($adjS_0^L$) are compared with their finite difference approximations in the following manner:

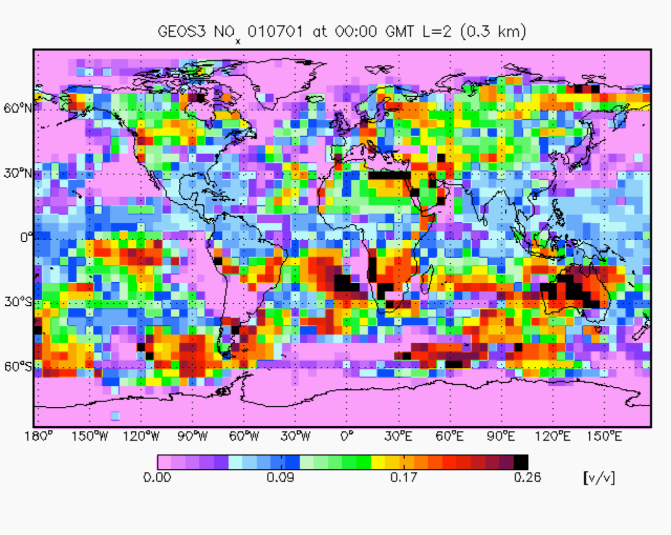
$$adjS_0^L \approx (pertS_T^H - origS_T^H) / pertS_0^L$$

**Results**

The following results are obtained with GEOS-Chem v7 adjoint, convection-only simulation, run for one week from 2001/07/01 to 2001/07/07. We consider the convection process acting on NOx concentrations, and test both the GEOS-3 and GEOS-4 adjoint convection schemes. The perturbation is introduced at layer L=2 and measured at layer H=9.

NOTE: The species names on some of the graphs are not correct since they are generated from temporary checkpoint files used to save the FD test results.

**<u>GEOS-3 results</u>**



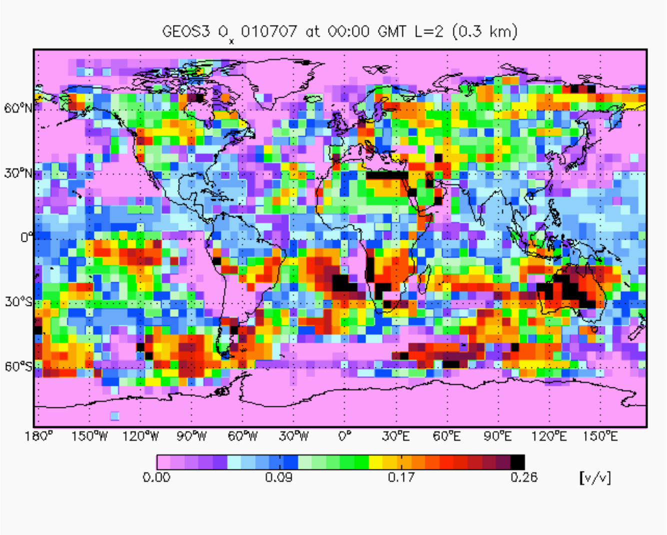
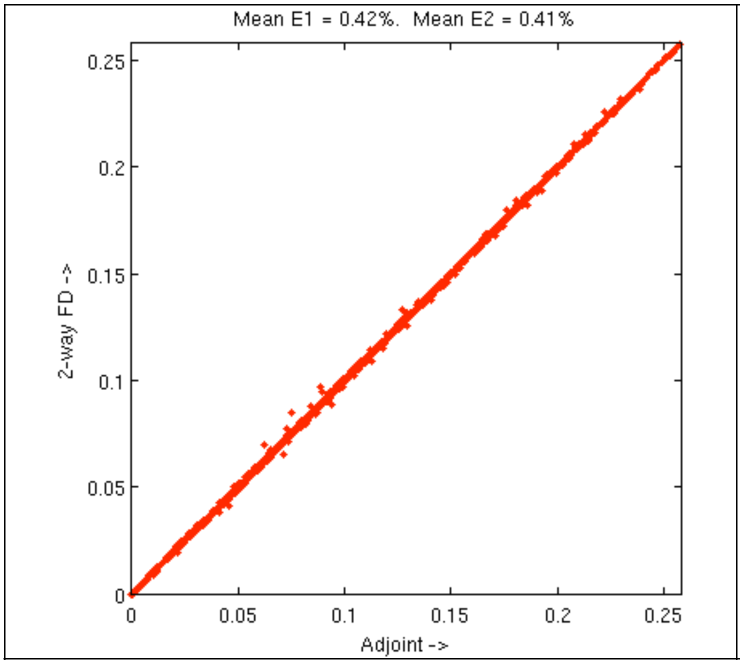


**Fig 6(a)** dNOx$_9$/dNOx$_2$ adjoint values at T=0.

**Fig 6(b)** dNOx$_9$/dNOx$_2$ Central-FD values.



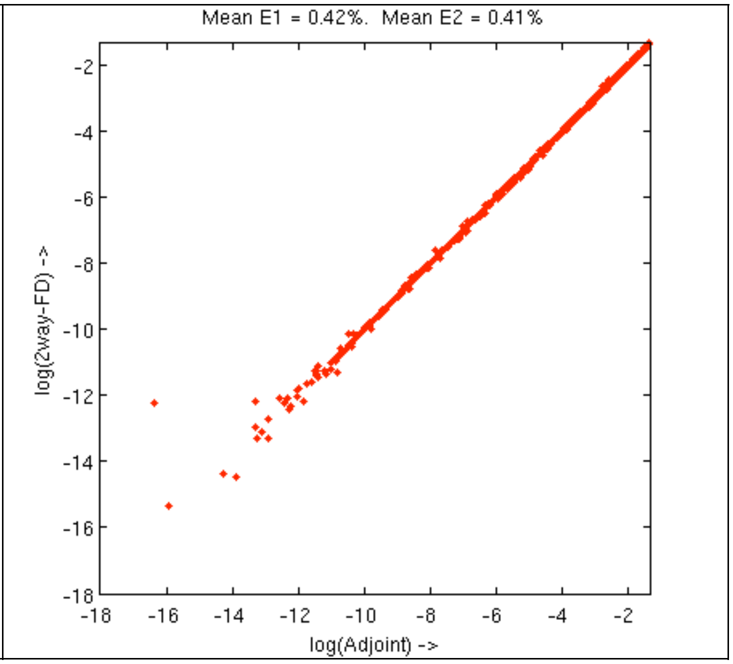
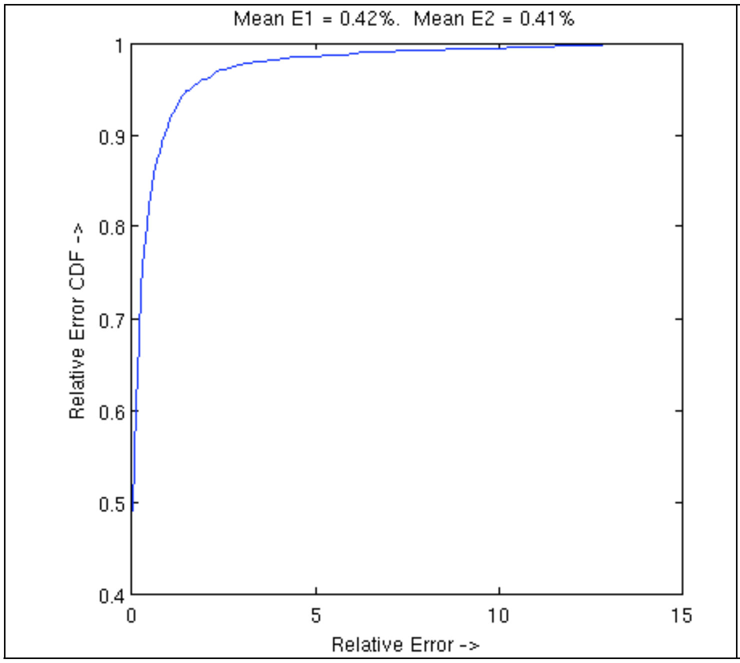


**Fig 6(c)** ADJ vs Central-FD scatter plot.

**Fig 6(d)** Log of abs(central-FD) vs. Log of abs(ADJ) scatter plot.



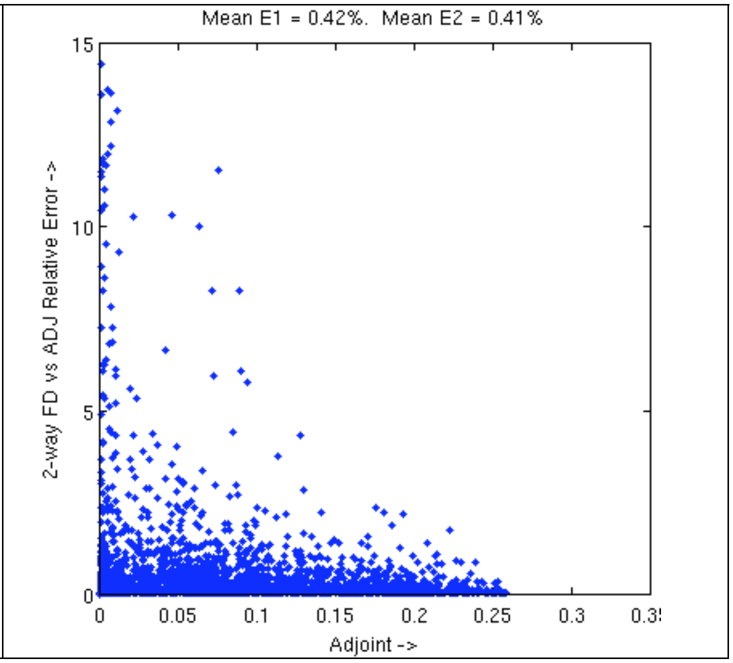


**Fig 6(e)** Central-FD to ADJ relative error CDF plot.

**Fig 3(f)** Central-FD to ADJ relative err. vs. ADJ.

## GEOS-4

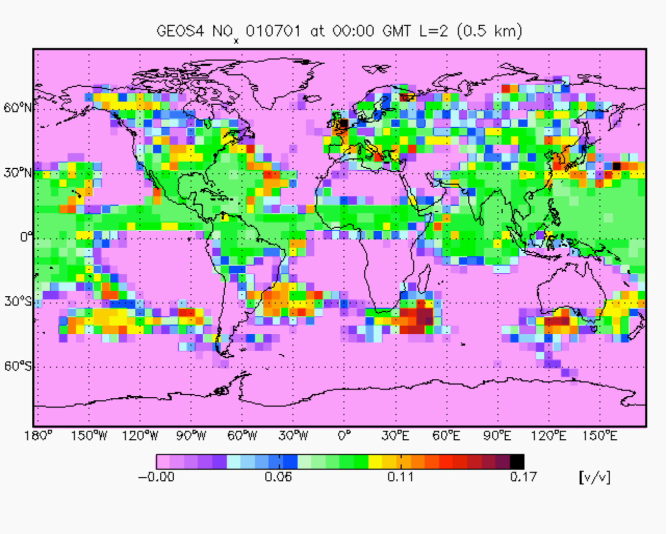


**Fig 7(a)** dNOx$_9$/dNOx$_2$ adjoint values at T=0.

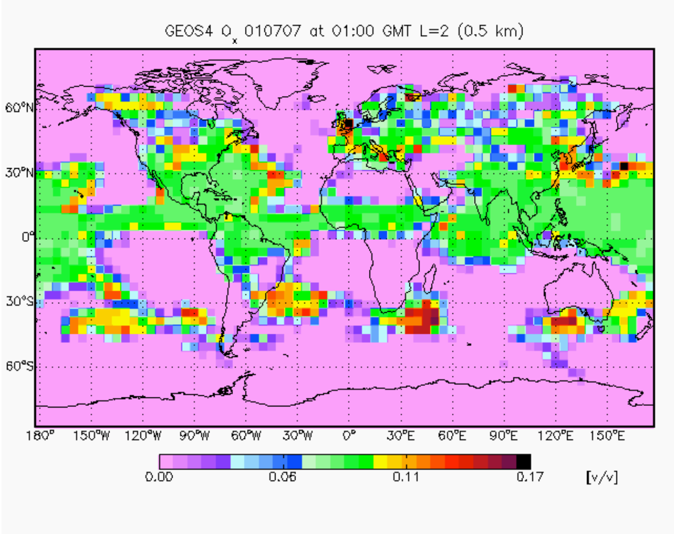


**Fig 7(b)** dNOx$_9$/dNOx$_2$ Central-FD values.

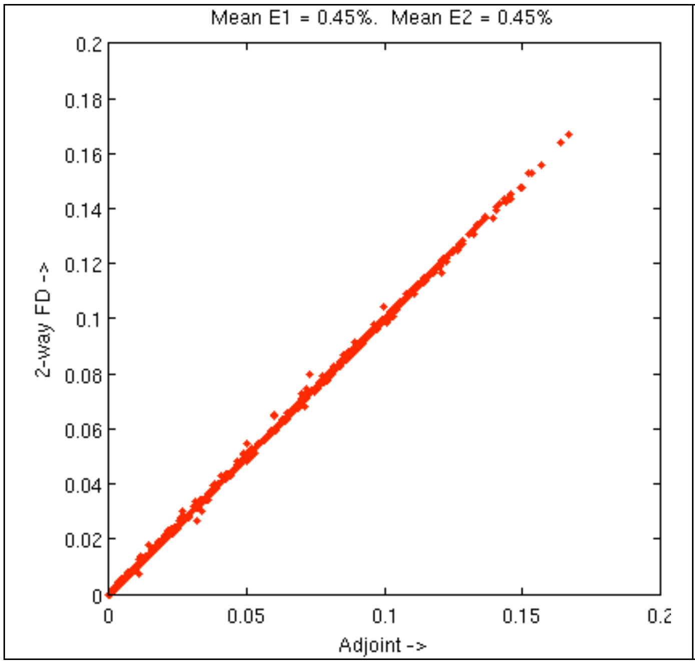


**Fig 7(c)** ADJ vs Central-FD scatter plot.

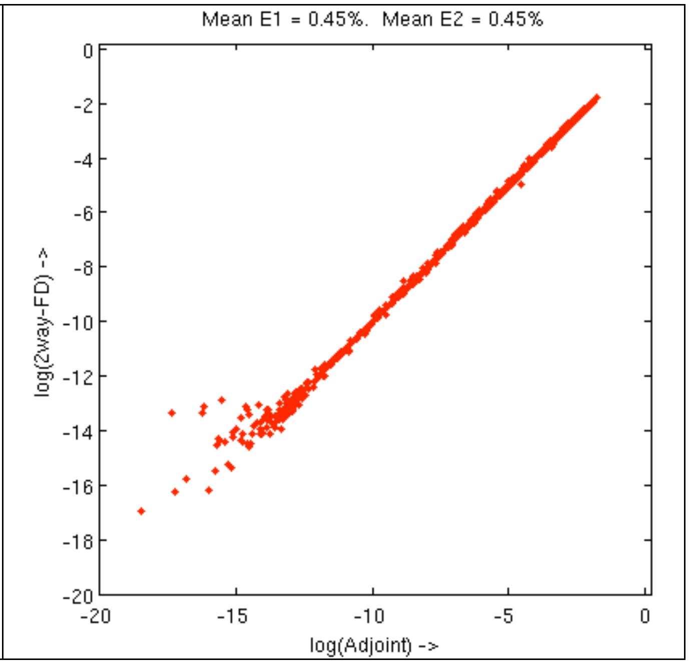
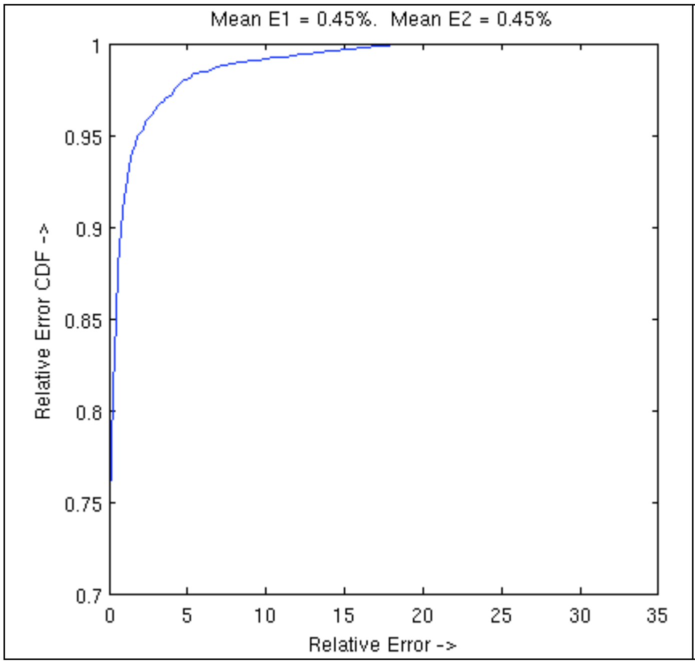


**Fig 7(d)** Log of abs(central-FD) vs. Log of abs(ADJ) scatter plot.



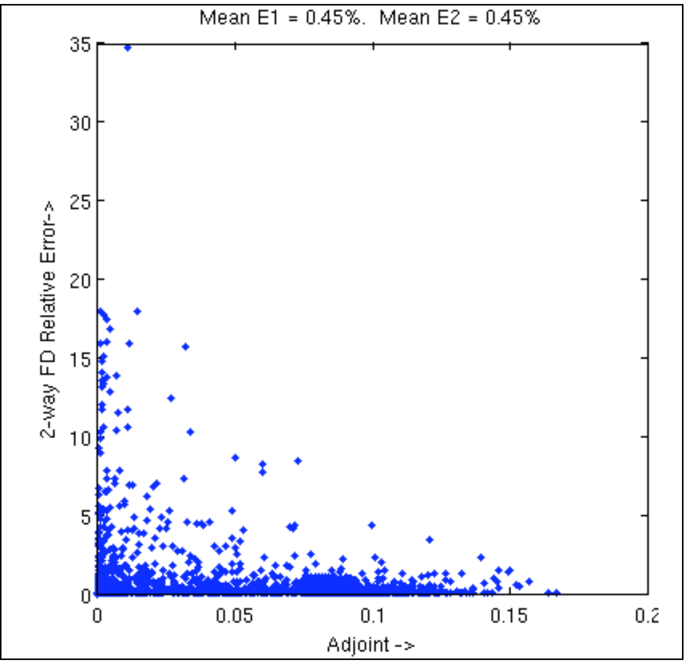


**Fig 7(e)** Central-FD to ADJ relative error CDF plot.

**Fig 7(f)** Central-FD to ADJ relative error vs. ADJ.

## 3.3 Transport Adjoint

GEOS-3 and GEOS-4 use TPCORE and TPCORE_FVDAS subroutines respectively for transport calculations. The continuous adjoints for the transport processes are constructed by calling the corresponding subroutines with reversed wind fields. The results presented below show a reasonably good agreement between the continuous adjoint solution and the finite difference approximations.

- ➔ In order to **test** the GEOS-Chem transportation adjoint, we choose a chemical species (say S), and a pair of grid points (say g and h). We calculate the derivative of the concentration of S at grid h and final time T with respect to the concentration of S at grid g and initial time, i.e., $dS_T^h/dS_0^g$. For a more comprehensive verification we choose multiple such sample grid points (day $G=\{g_1, g_2,\ldots, g_n\}$) and perform the finite different test for each of those.

- ➔ A forward run of GEOS-Chem is performed for the time period (T) under consideration, with an initial perturbation ($pertS_0^g$) in S at grid point g only. The concentrations from the final time ($pertS_T^h$) are saved for finite difference calculations.

- A similar forward run is performed but this time with the original S and the pressure is checkpointed every transportation time step to be used in the adjoint run which follows it. The concentrations for the final time (orig$S_T^h$) are again saved.

- In the adjoint mode, the adjoint associated with species S are initialized with unit concentration over grid point h only at the final time (T) transporting backwards to the initial time. At T=0, the adjoint tracer concentrations for species S (adj$S_o^g$) are compared with their finite difference approximations in the following manner:

$$\mathrm{adj}S_0^g \approx ( \mathrm{pert}S_T^h - \mathrm{orig}S_T^h ) / \mathrm{pert}S_0^g$$

## 48 HOURS SIMULATION RESULTS

The following results are obtained with GEOS-Chem v7 adjoint, transport-only simulation, run for 48 hours from 2001/07/01 : 000000 to 2001/07/03 : 000000. We consider the transport of NOx using GEOS-3 and GEOS-4 advection schemes. In order to perform multiple tests for different grid points in the same run, a set of grid points that are sufficiently far apart is chosen. A perturbation is introduced in each of these grid points and is tracked at corresponding receptor site at the final time.

## GEOS-3 (48h results)

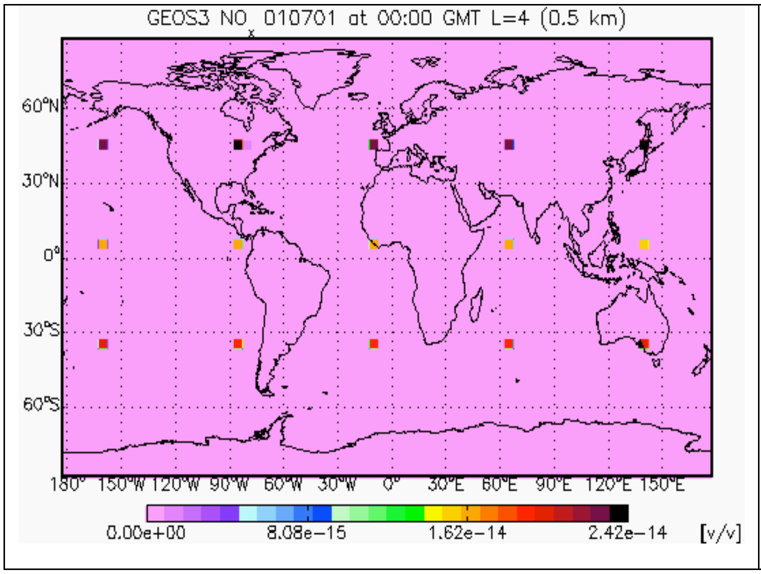

GEOS3 NO$_x$ 010701 at 00:00 GMT L=4 (0.5 km)
60°N
30°N
0°
30°S
60°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
0.00e+00
8.08e−15
1.62e−14
2.42e−14
[v/v]

**Fig 8(a)** Set of sample grid points at T=0.

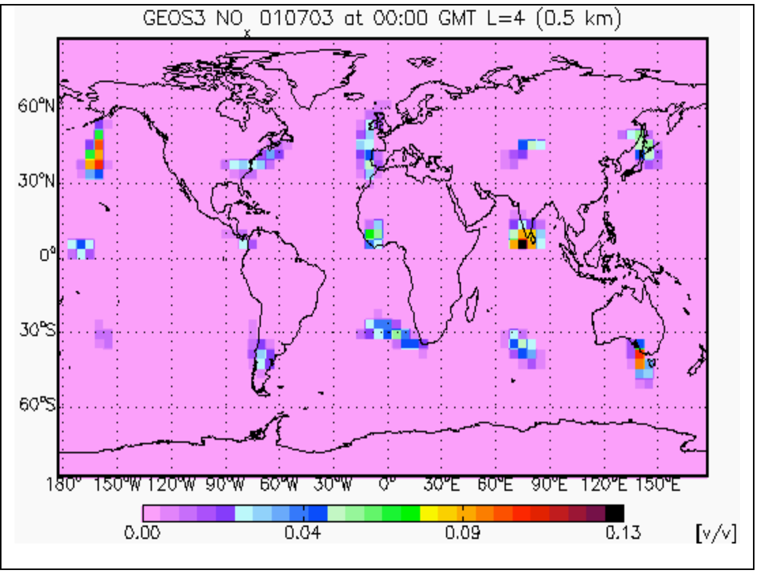
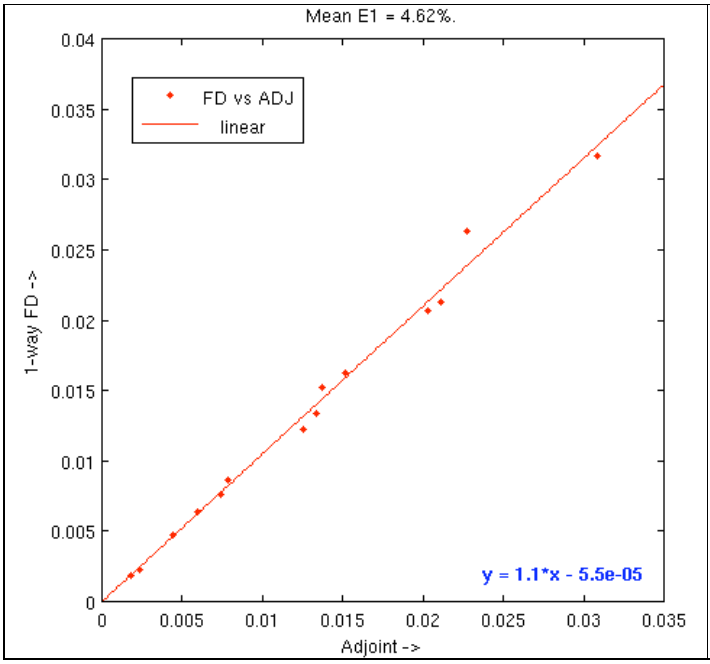

GEOS3 NO$_x$ 010703 at 00:00 GMT L=4 (0.5 km)
60°N
30°N
0°
30°S
60°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
0.00
0.04
0.09
0.13
[v/v]

**Fig b(b)** Sample grid point concentrations at final time.



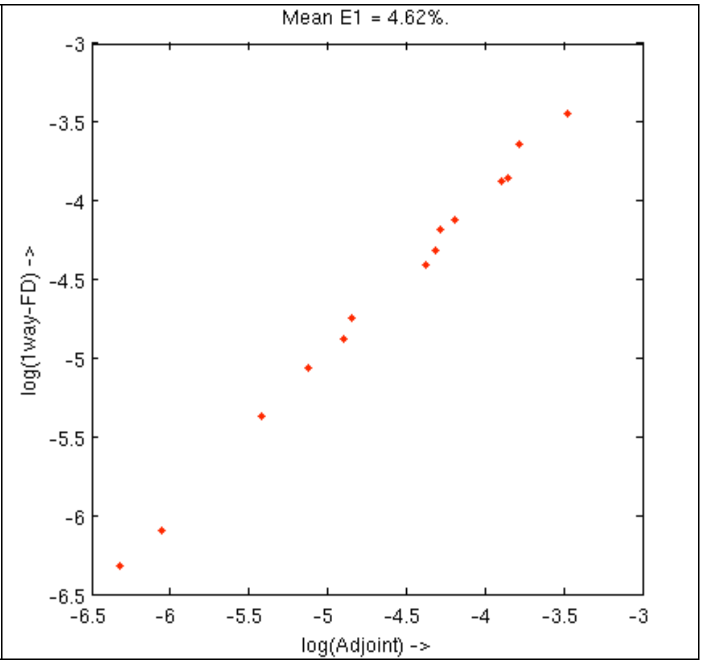
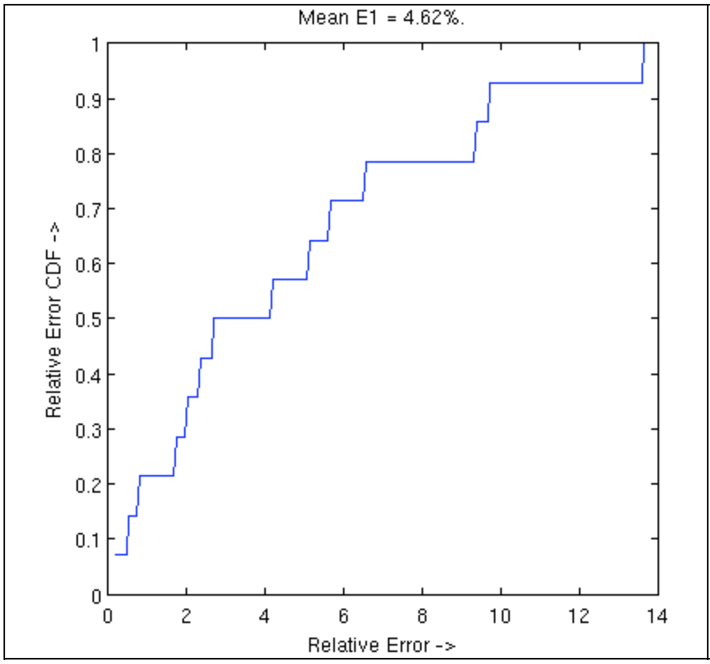


**Fig 8(c)** ADJ vs One-sided FD scatter plot.

**Fig 8(d)** Log of abs(one-sided FD) vs. Log of abs(ADJ) scatter plot.



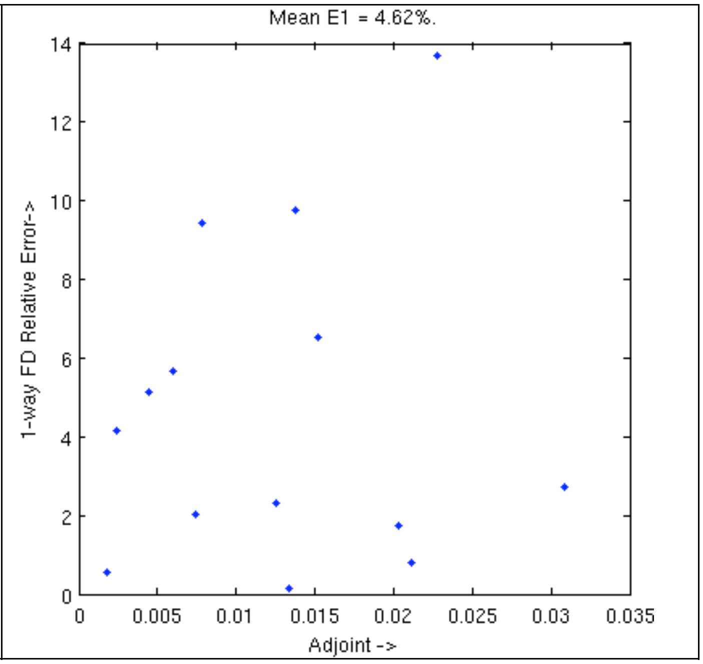
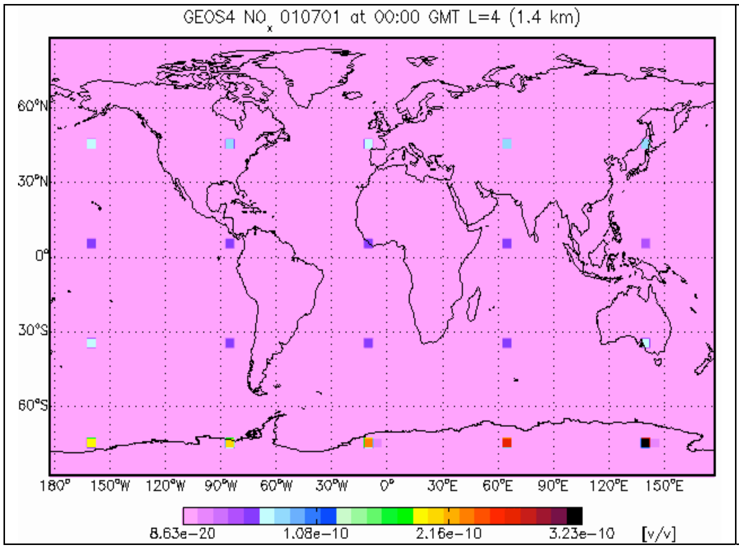


**Fig 8(e)** One-sided FD to ADJ relative error CDF plot.

**Fig 8(f)** One-sided FD to ADJ relative error vs. ADJ scatter plot.

## GEOS-4 (48h results)



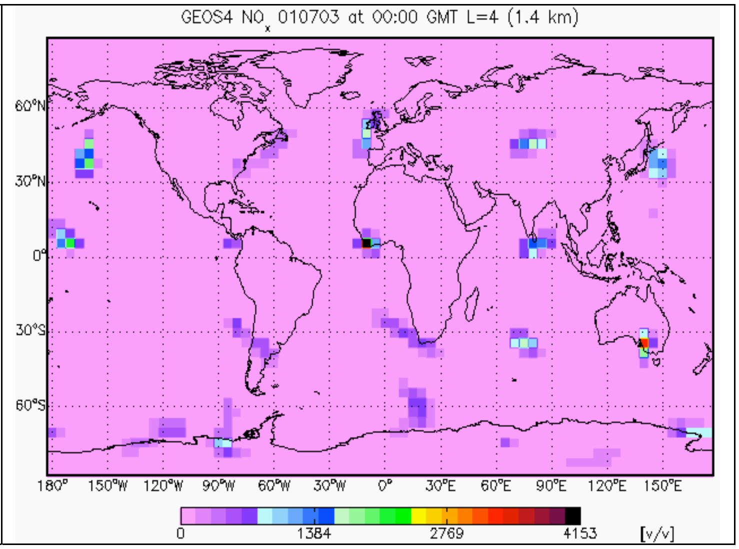
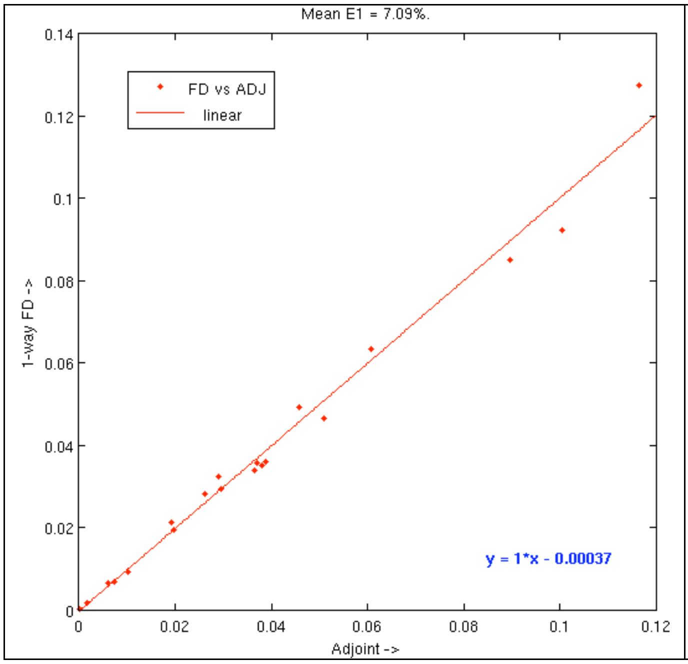


**Fig 9(a)** Set of sample grid points at T=0.

**Fig 9(b)** Sample grid point concentrations at final time.



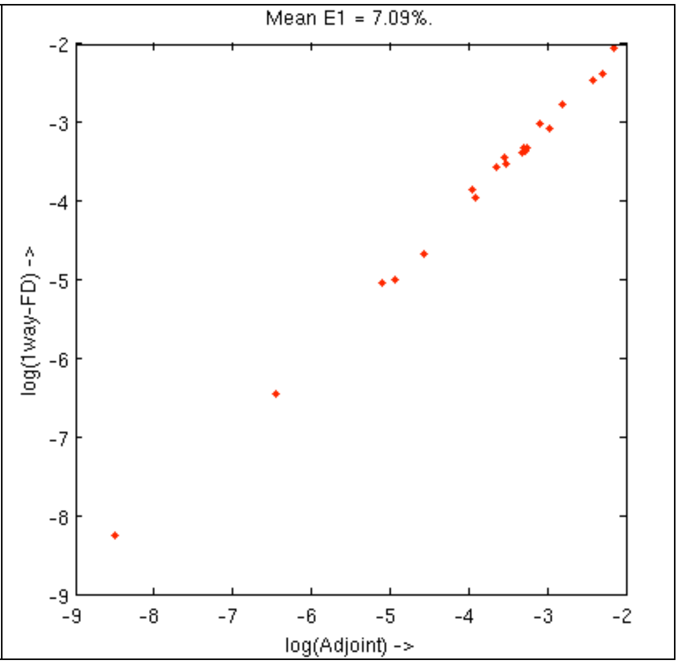


**Fig 9(c)** ADJ vs One-sided FD scatter plot.

**Fig 9(d)** Log of abs(one-sided FD) vs. Log of abs(ADJ) scatter plot.

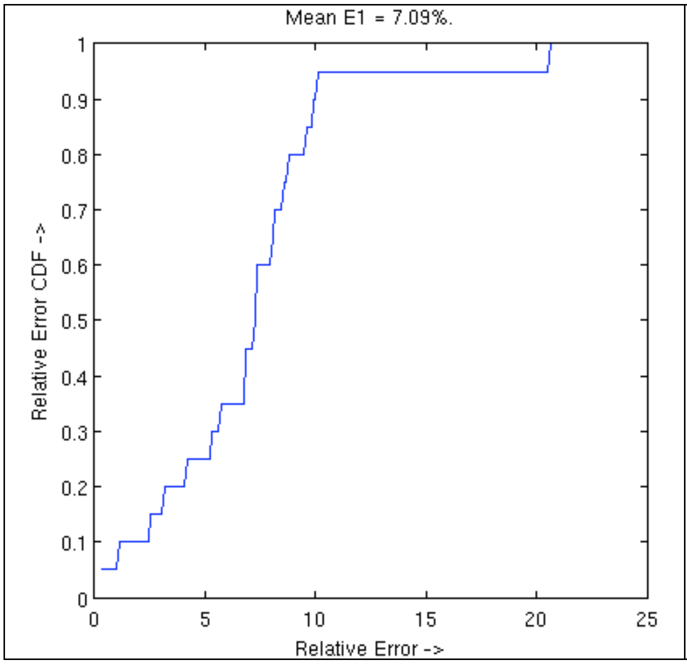


**Fig 9(e)** One-sided FD to ADJ relative error CDF plot.

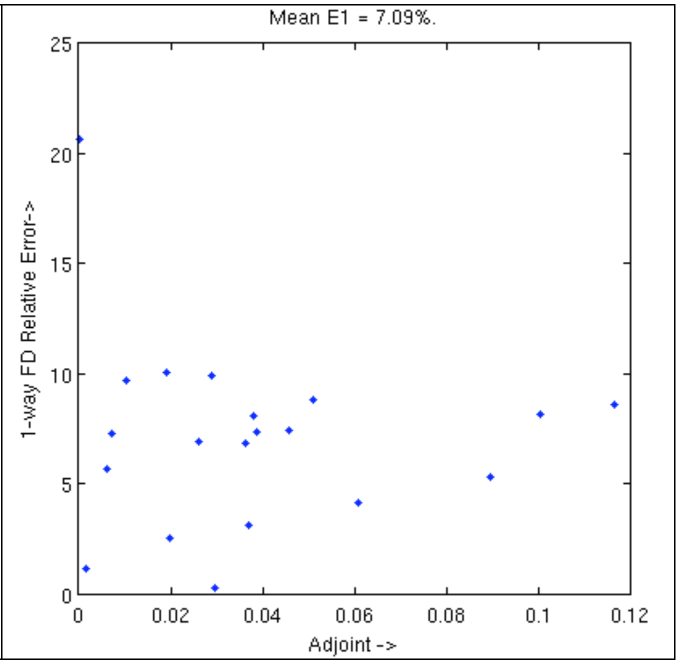
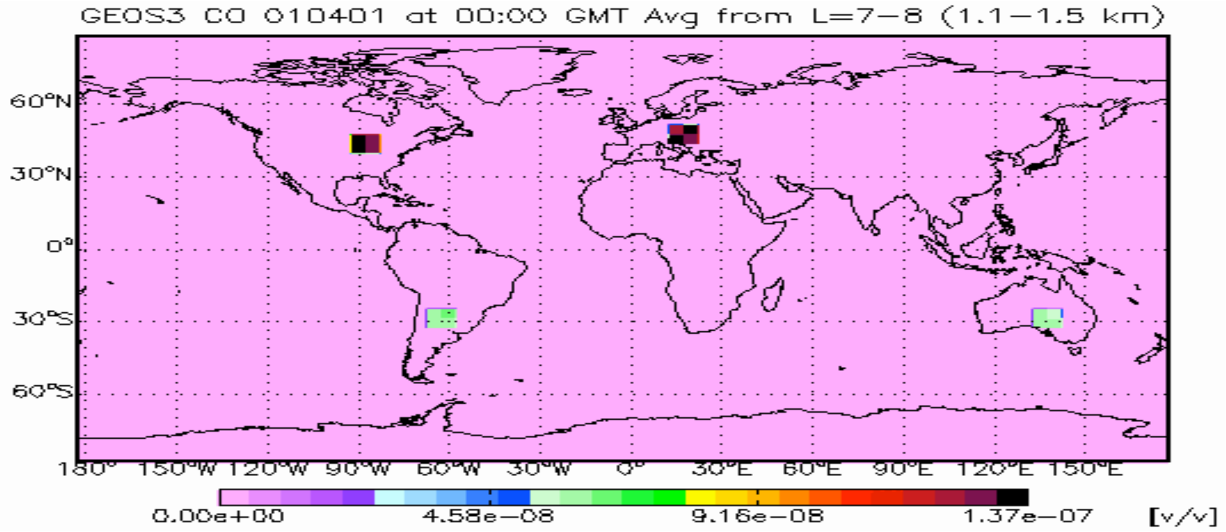


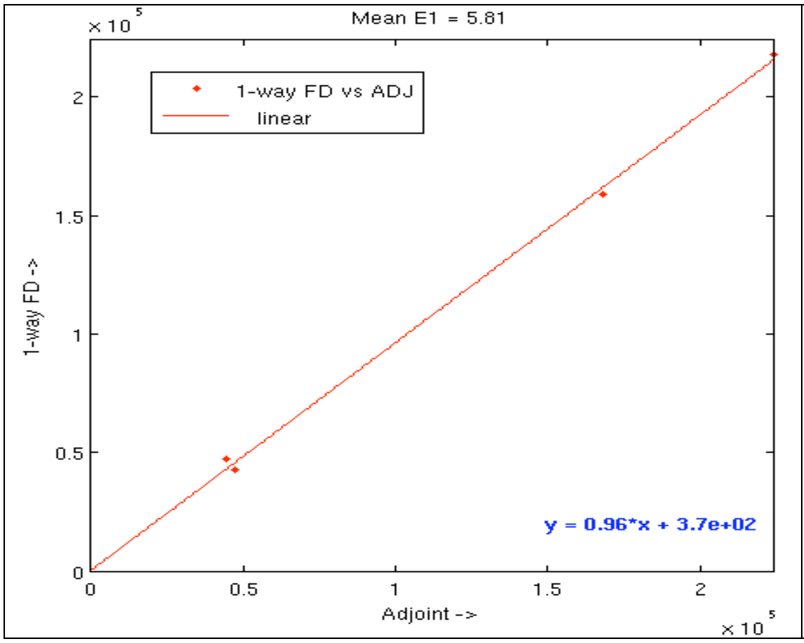
**Fig 9(f)** One-sided FD to ADJ relative error vs. ADJ scatter plot.

## ONE WEEK SIMULATION RESULTS

The next set of results are obtained with GEOS-Chem v7 adjoint, transporta-only simulation, run for one week from 2001/04/01 : 000000 for GEOS-3 and 2001/07/01:000000 for GEOS-4. We consider adjoints for CO concentrations in vertical layers 7-8. An approach similar to 48 hours adjoint test is carried out with four points in different continents around the globe as shown below:



## GEOS-3 (one week results)



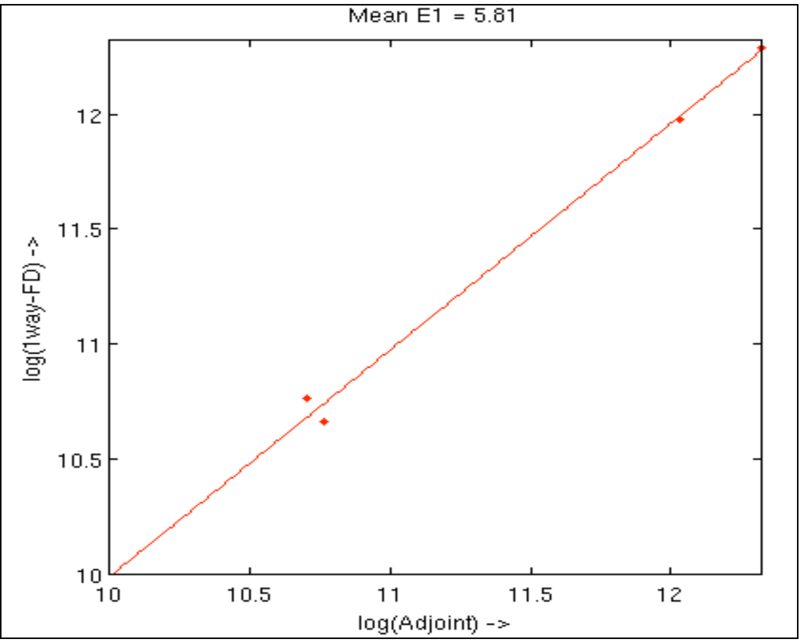
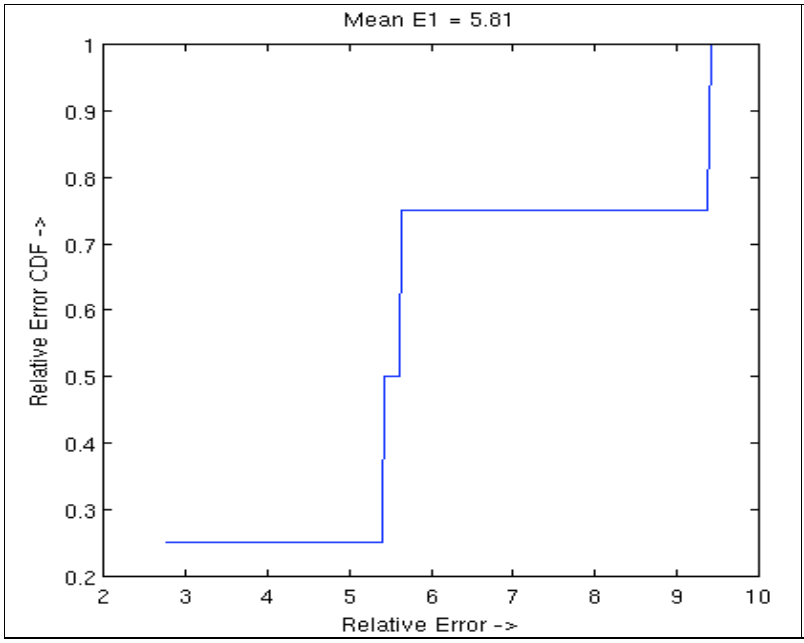


**Fig 10(a)** ADJ vs One-sided FD scatter plot.

**Fig 10(b)** Log abs(One-sided FD) vs. Log abs(ADJ) scatter plot.



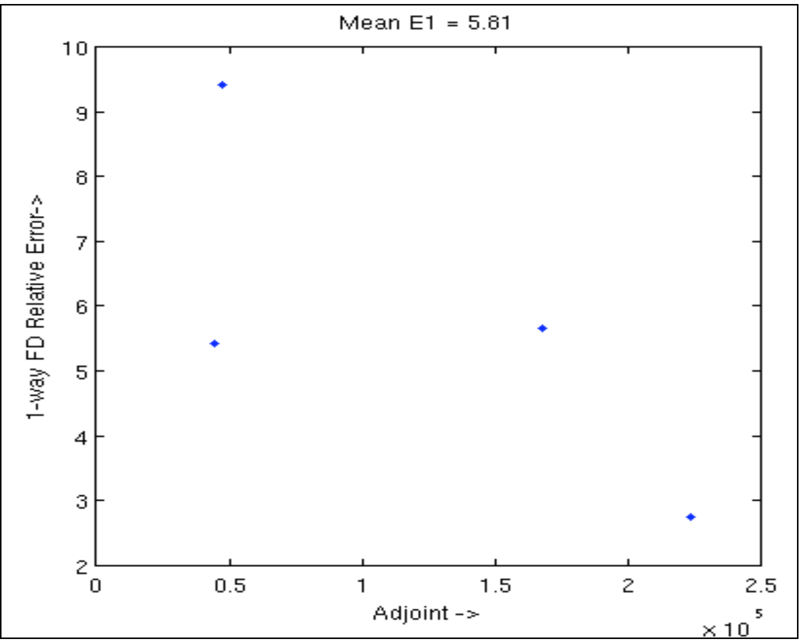
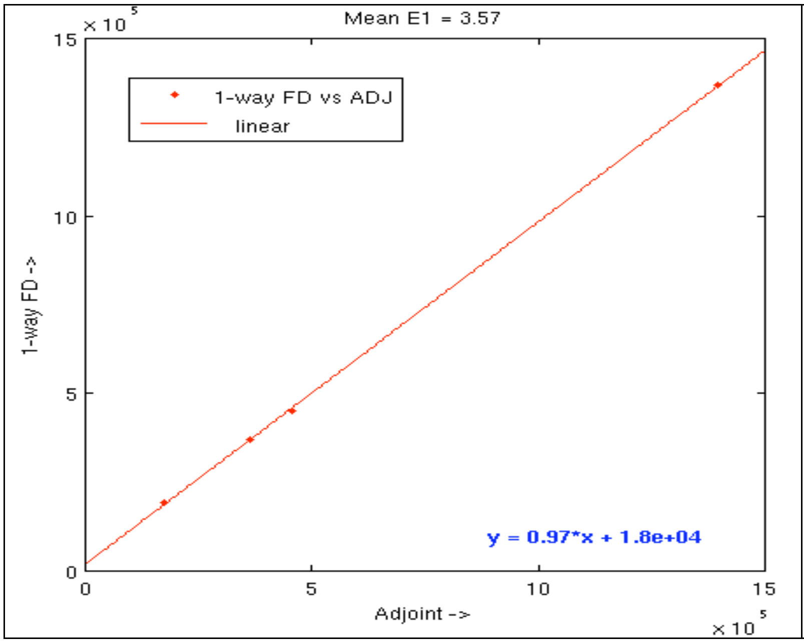


**Fig 10(c)** One-sided FD to ADJ relative error CDF plot.

**Fig 10(d)** One-sided FD to ADJ relative err vs. ADJ scatter plot.

## GEOS-4 (one week results)



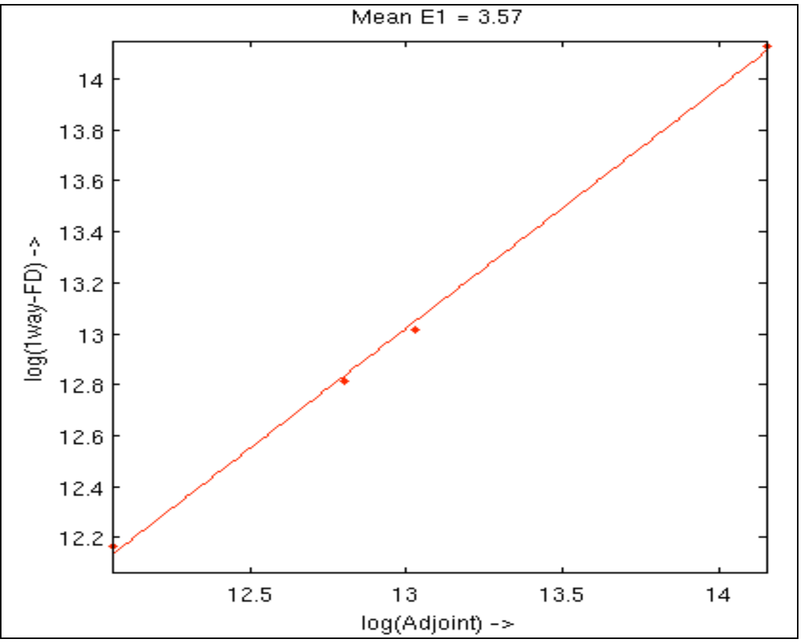
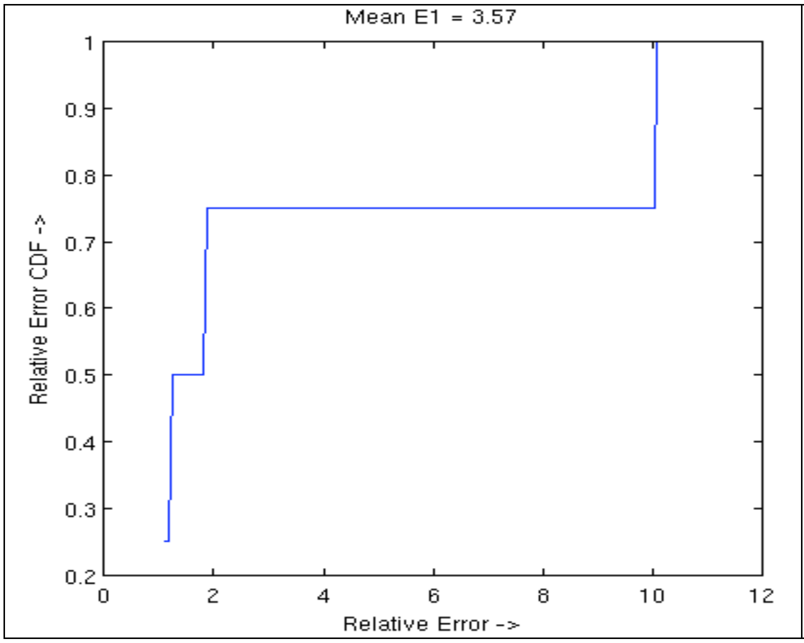


**Fig 11(a)** ADJ vs One-sided FD scatter plot.

**Fig 11(b)** Log of abs(One-sided FD) vs. Log of abs(ADJ) scatter plot.



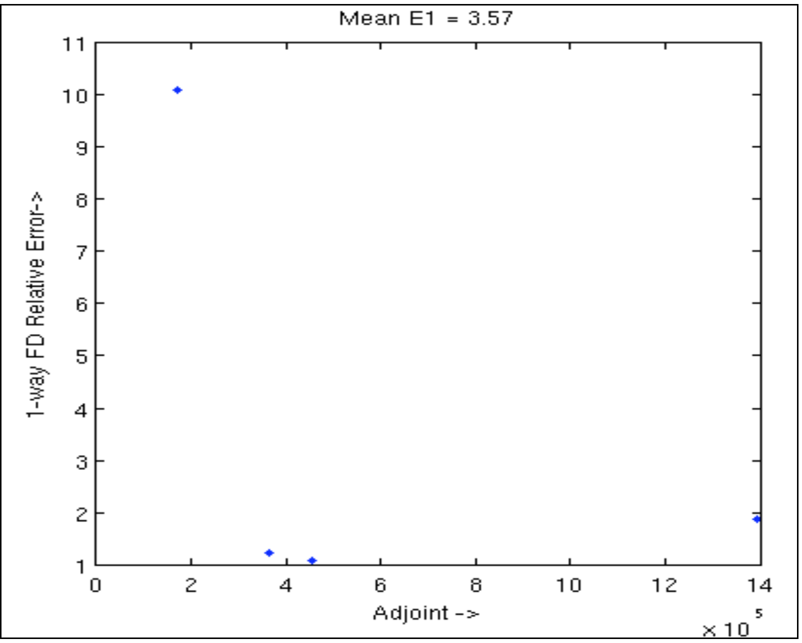


**Fig 11(c)** One-sided FD to ADJ relative error CDF plot.

**Fig 11(d)** One-sided FD to ADJ relative err vs. adjoint scatter plot.

## 3.4 Emission and Dry Deposition Adjoint

In GEOS-Chem emission and dry deposition are handled through chemistry via fake equations. The rates for these processes are calculated separately and then attached to the chemistry reaction rates. The adjoints of these subroutines are calculated using the adjoint integrator which provides adjoint derivatives with respect to the reaction rates. These adjoint variables are then scaled by the individual emission or deposition rates and are accumulated over time. The results presented below show a good agreement of adjoints and finite difference approximations.

- ➔ In order to **test** the GEOS-Chem emissions adjoint, we compute the adjoint derivative of the concentration of species P (at final time) with respect to the emission rate of species S at the same grid point (say h). For verification purposes, the finite difference test is repeated at all grid points.

- ➔ A forward run of GEOS-Chem is performed for the time period (T) under consideration, with a perturbation (pert$\alpha_0$) in the emission rates REMIS for species S, at all grid points, for all times {0,1,…T-1}. Specifically, the emission rate of species S is changed from REMIS(S) to (1+ pert$\alpha_0$)*REMIS(S). The concentrations at the final time (pert$P_T$) are saved for finite difference calculations.

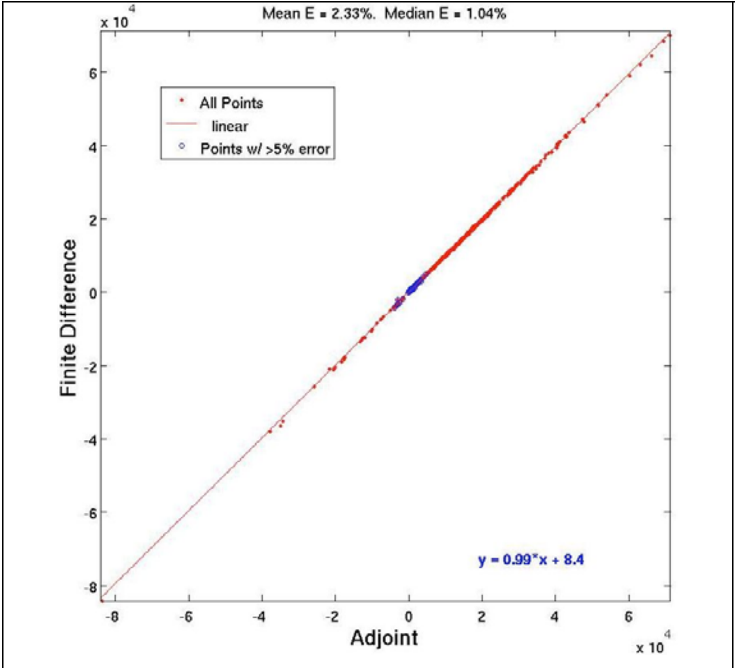
- ➔ A forward run is performed but this time with original emission rates REMIS. The concentrations for the final time (orig$P_T$) are again saved.

- ➔ In the adjoint mode, the adjoint associated with species P is initialized with unit concentration over all the grid points at the final time (T). At T=0, the adjoint derivatives with respect to REMIS(S) (adj$S_0$) are compared with their finite difference approximations in the following manner:

$$\mathrm{adj}S_0 \approx (\ \mathrm{pert}P_T - \mathrm{orig}P_T\ ) / \mathrm{pert}\alpha_0$$

**Results:**

The following results are obtained GEOS-Chem v7 adjoint, 48 hours run from 2001/07/01 : 000000 to 2001/07/03 : 000000. We consider derivatives of Ox concentrations with respect to changes in NOx emissions. A perturbation in the emission is introduced in each grid point.



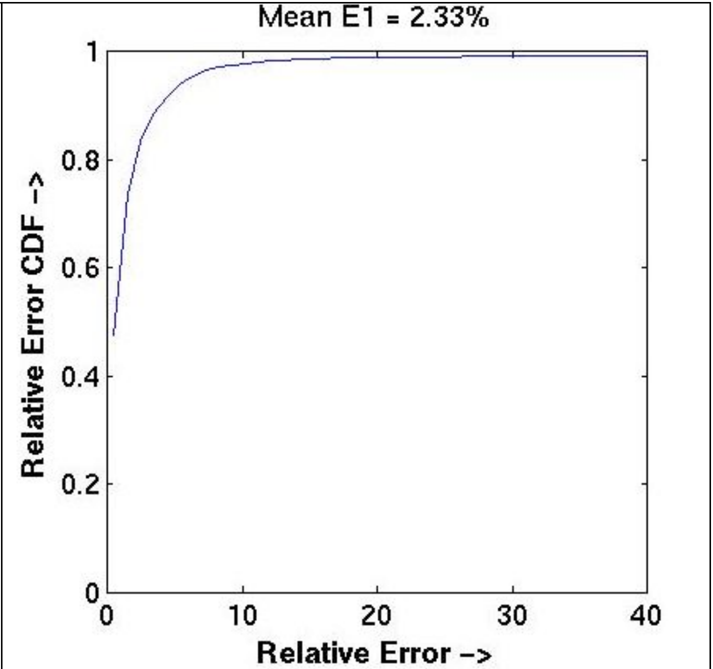


**Fig 12(a)** One-sided FD vs Adjoint scatter plot with linear curve fitting.

**Fig 12(b)** CDF of One-sided FD to ADJ relative error. Shows what percentages of total points are within what percentage of error.

## 3.5 Full GEOS-Chem Adjoint Sensitivities

We next illustrate the adjoint variables computed with the full GEOS-Chem adjoint. The cost function is the total ozone concentration under a hypothetical TES trajectory, as shown below. The adjoint sensitivities plotted below represent areas of influence. The TES ozone measurement will be sensitive to changes in emissions, etc. in these areas during the week before the measurement is taken.

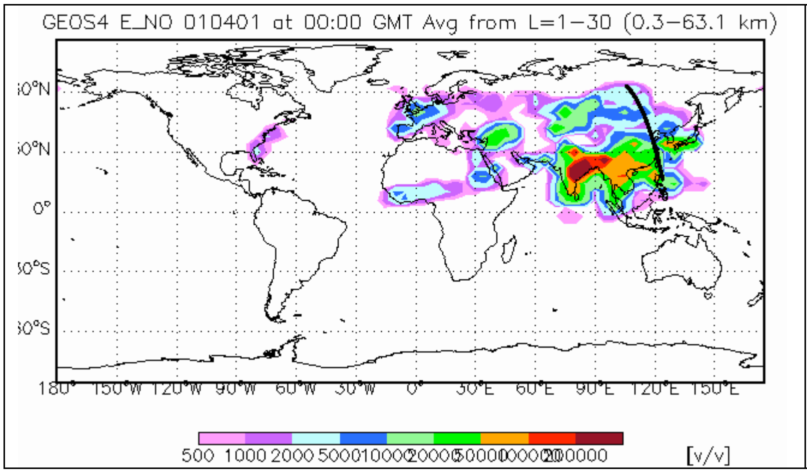

GEOS4 E_NO 010401 at 00:00 GMT Avg from L=1−30 (0.3−63.1 km)
0°N
0°N
0°
0°S
0°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
500 1000 2000 5000 10000 20000 50000 100000 200000
[v/v]

**Fig 13(a)** Ox Adjoint with respect to Total NOx emissions.

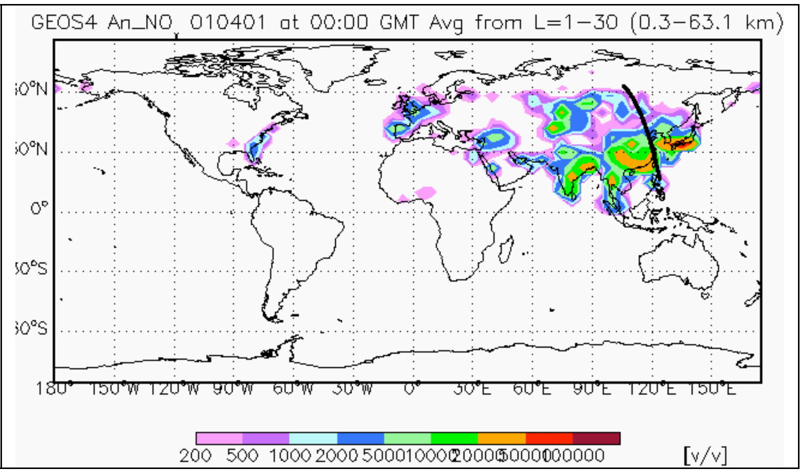

GEOS4 An_NO 010401 at 00:00 GMT Avg from L=1−30 (0.3−63.1 km)
0°N
0°N
0°
0°S
0°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
200 500 1000 2000 5000 10000 20000 50000 100000
[v/v]

**Fig 13(b)** Ox Adjoint with respect to Anthro NOx Emissions.

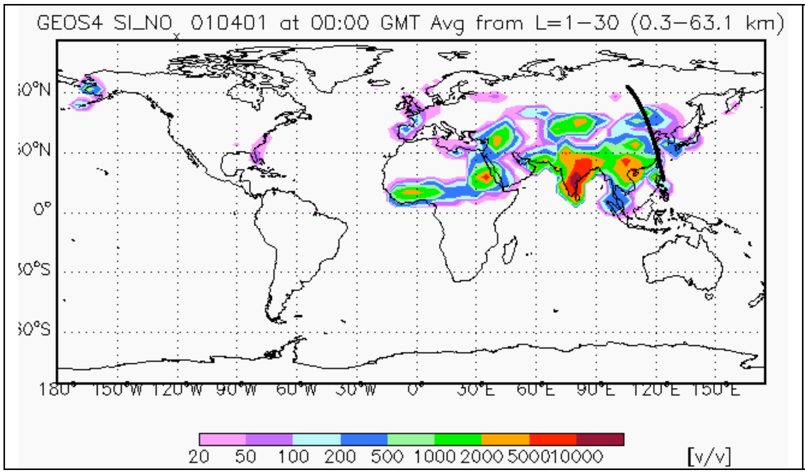

GEOS4 Sl_NO 010401 at 00:00 GMT Avg from L=1−30 (0.3−63.1 km)
0°N
0°N
0°
0°S
0°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
20 50 100 200 500 1000 2000 5000 10000
[v/v]

**Fig 13(c)** Ox Adjoint with respect to Soil NOx NOx emissions.

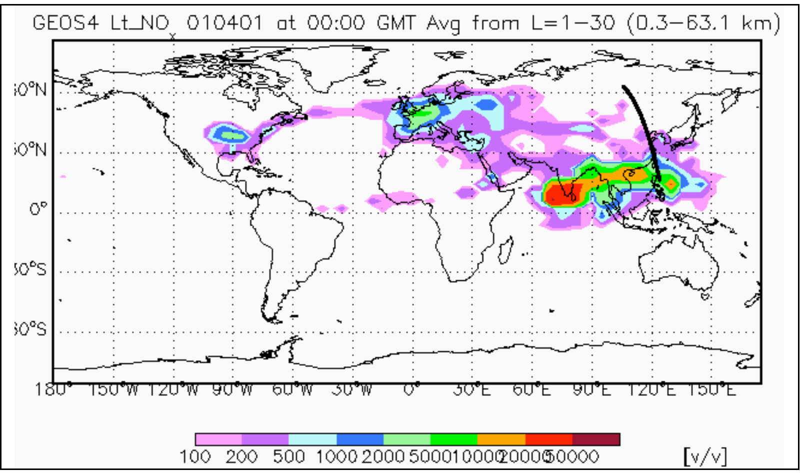
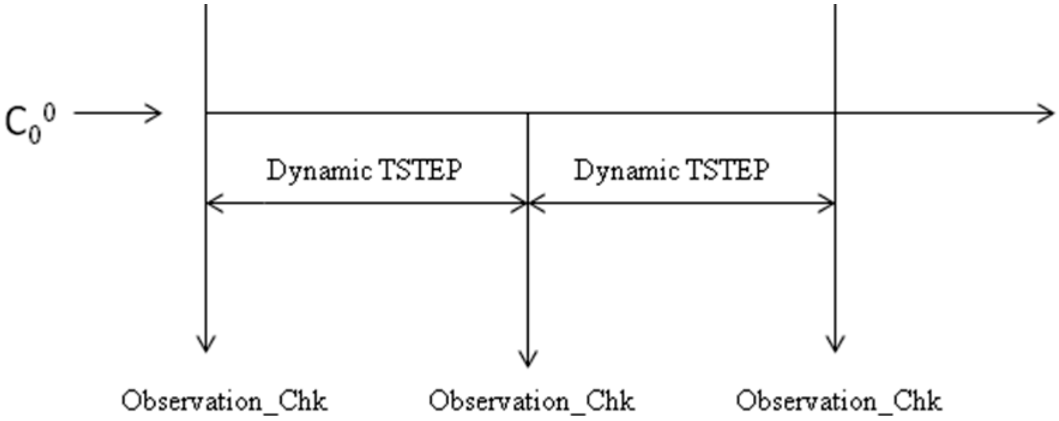

GEOS4 Lt_NO 010401 at 00:00 GMT Avg from L=1−30 (0.3−63.1 km)
0°N
0°N
0°
0°S
0°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
100 200 500 1000 2000 5000 10000 20000 50000
[v/v]

**Fig 13(d)** Ox Adjoint with respect to Lightning emissions.

# 4 4D-Var Data Assimilation

4D-Var data assimilation allows the optimal combination of three sources of information: an a priori (background) estimate of the state of the atmosphere, knowledge about the physical and chemical processes that govern the evolution of pollutant fields as captured in the chemistry transport model (CTM), and observations of some of the state variables.

In order to carry out chemical data assimilation with GEOS-Chem, an interface has been developed to integrate the adjoint code with the optimization routine L-BFGS. For testing purposes artificial observations are extracted from a reference forward GEOS-Chem run using the initial concentration field $c_0^0$.



A perturbation is then added to the initial concentrations $c_0^{ref}$ to produce $c_0^p$.

$$c_0^p = c_0^{ref} + \text{perturbation}$$

This perturbed concentration is considered the best guess of the initial conditions, and is adjusted in successive optimization iterations to obtain an improved estimate $c_{op}^0$ of the initial concentration $c_0^0$.

We denote by x the vector of control variables (the set of variables adjusted by the numerical optimization process). At iteration 0,

$$x_0 = c_p^0$$

At each subsequent iteration k (k≥1),

$$x_{k+1} \leftarrow \text{ L-BFGS } (x_k, f, g)$$

$$c_0^p \leftarrow x_{k+1}$$

$$(f, g) \leftarrow \text{ GEOS-Chem-ADJOINT } (c_0^p, \text{Observation\_Chk})$$

where, f is the cost function and g is the gradient of the cost function.

For our test case the cost function and its gradient are defined as:

$$f = \frac{1}{2} \sum_{k=1}^{Ntimes} \sum_{g=1}^{Ngrids} \left( c_{k,g}^{p} - c_{k,g}^{ref} \right)^{T} R_{k,g}^{-1} \left( c_{k,g}^{p} - c_{k,g}^{ref} \right)$$

and

$$g = \sum_{k=1}^{Ntimes} \sum_{g=1}^{Ngrids} R_{k,g}^{-1} \left( c_{k,g}^{p} - c_{k,g}^{ref} \right)$$

where k = 1,2,…,$T_{final}$ is the total number of time steps runs in the forward mode and g is a 4-tuple observation grid. The observation grid specifications and perturbation amounts used are as follows:

| ***Observation Grid*** | | | | ***Perturbation amount*** |
|---|---|---|---|---|
| ***Columns*** | ***Rows*** | ***Layer*** | ***Species*** | $c_p^0 = c_0^0 * (1+eps)$,<br>***eps = 0.1*** |
| ***x1=1,x2=IIPAR*** | ***x1=1,x2=JJPAR*** | ***1:NLAYS*** | ***Ox*** | |

In our experiment we are perturbing initial Ox, observing Ox, and retrieving initial Ox only.

## 4.1 Validation and Results

This data assimilation test is performed for GEOS-4 with chemistry, transport and convection, for the month of July 2001, for 12hrs, starting from 000000 hrs.

To validate the data assimilation test-bed, we plot the difference between the perturbed and the reference initial concentrations before the data assimilation (Fig. 14(a)) and after data assimilation (Fig 14(b)). The assimilated concentration fields are better estimates of the reference concentration fields. Data assimilation recovers well the reference initial zone concentration.

Figures 14(c) and (d) illustrate the convergence of the optimization algorithm. The decrease of the cost-function with the number of model runs is shown in Fig 14(c). The decrease of the root-mean square error of the perturbed initial conditions with the number of model runs is shown in Fig. 14(d). The root mean square error is defined at each iteration as

$$RMS = \sqrt{\frac{1}{Ngrids} \sum_{g=1}^{Ngrids} \left( c_{0,g}^{p} - c_{0,g}^{ref} \right)^{2}}$$

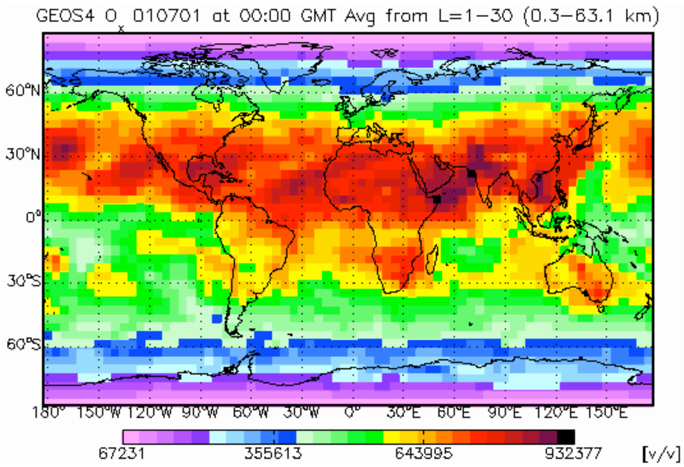

GEOS4 $O_x$ 010701 at 00:00 GMT Avg from L=1–30 (0.3–63.1 km)
60°N
30°N
0°
30°S
60°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
67231
355613
643995
932377
[v/v]

**Fig 14 (a)** Difference between the perturbed and the reference concentrations ($c_0^p$- $c_0^{ref}$).

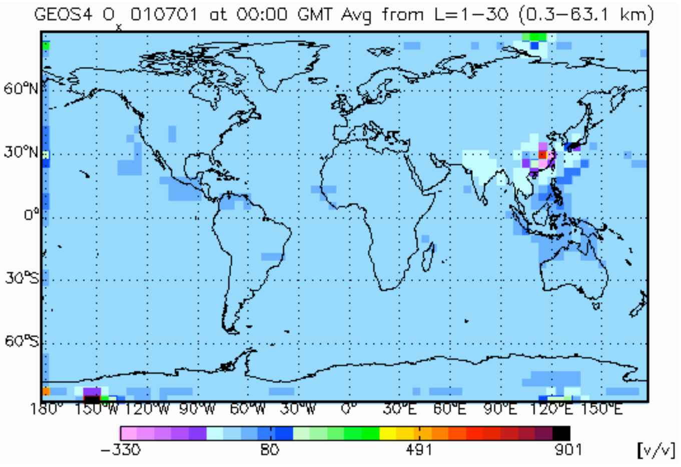

GEOS4 $O_x$ 010701 at 00:00 GMT Avg from L=1–30 (0.3–63.1 km)
60°N
30°N
0°
30°S
60°S
180° 150°W 120°W 90°W 60°W 30°W 0° 30°E 60°E 90°E 120°E 150°E
–330
80
491
901
[v/v]

**Fig 14(b)** Difference between the optimized and the reference concentrations ($c_0^{opt}$- $c_0^{ref}$).

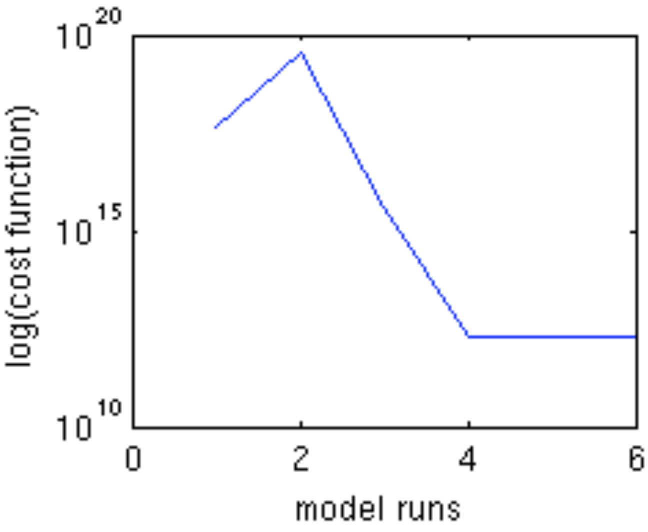


**Fig 14(c)** Decrease of the Cost function vs. number of model runs

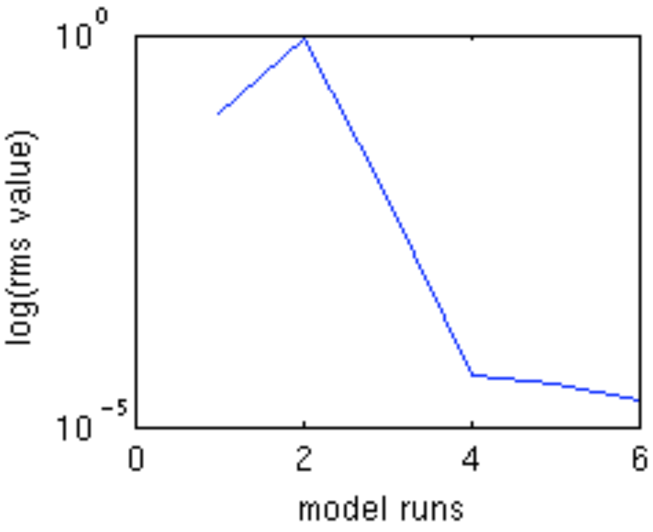


**Fig 2(b)** Decrease of the root-mean square error vs number of model runs.

# 5 Parallelization Aspects

## 5.1 Development of the OpenMP parallel KPP code

The forward and adjoint GEOS-Chem (with the KPP implemented chemistry) is parallelized using OpenMP. All the forward and backward time-stepping subdrivers have a modular layout that allows each phase of the simulation to be performed separately, with checkpointing code being placed in between each phase. Therefore in order to parallelize the chemistry using KPP, we only need to focus on the driver subroutines in chemistry_mod.f.

The GCKPP_DRIVER and GCKPP_DRIVER_ADJ subroutines are perform the forward and adjoint chemistry steps. These drivers each contain a loop which iterates over grid cells and calls the KPP integrator to solve the chemistry. The results of each iteration are independent of every other iteration, allowing us to parallelize this loop. The OpenMP statements used to parallelize the GCKPP_DRIVER and GCKPP_DRIVER_ADJ are very similar. The OpenMP statements used to parallelize the GCKPP_DRIVER are shown below.

```
!$OMP PARALLEL DO
      !$OMP+DEFAULT( SHARED )
      !$OMP+PRIVATE( JJLOOP, I, J, L, N, RSTATE, ISTATUS, IH, JH, LH, TID )
      !$OMP+FIRSTPRIVATE( RCNTRL, ICNTRL )
      !$OMP+COPYIN( TIME )
      !$OMP+SCHEDULE( DYNAMIC )
            DO JJLOOP = 1,NTT
                  //Forward Chemistry Driver Code
            ENDDO
!$OMP END PARALLEL DO
```

The next step is to modify the KPP generated code to make the necessary KPP global variables threadprivate. This ensures that each thread has a private copy of the simulation data to modify. Therefore each array and variable modified by the KPP integrator is made threadprivate in gckpp_Global.f90. Additionally there is a global variable declared in gckpp_Function.f90 which also needs to be declared threadprivate. The threadprivate declaration in gckpp_Global.f90 is shown below.

```
!$OMP THREADPRIVATE( VAR, V AR_ADJ, V_CSPEC, V_CSPEC_ADJ, FIX, JLOOP, RCONST,
TIME )
```

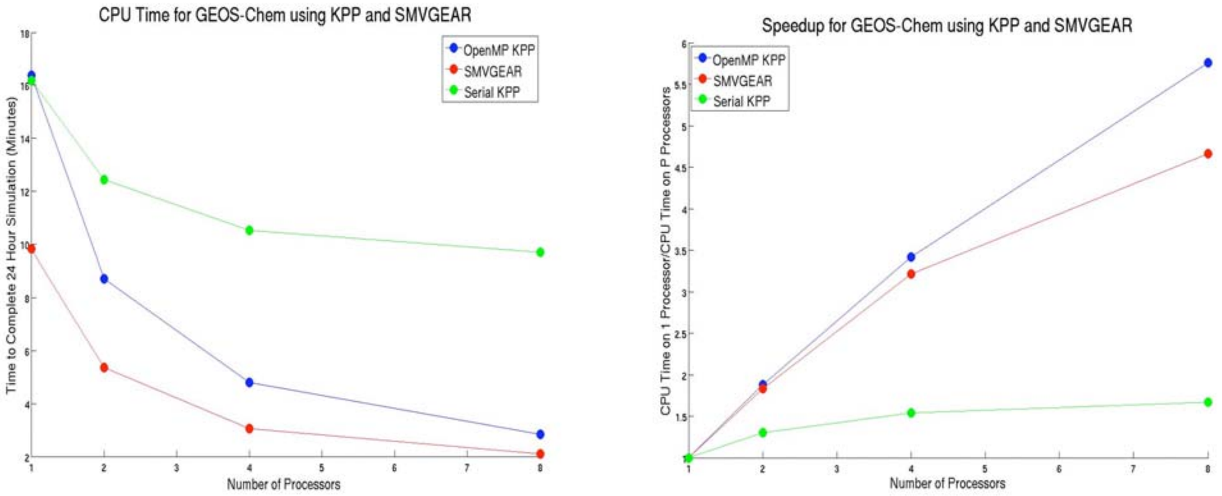
Parallelizing the GCKPP_DRIVER_ADJ subroutine requires the additional step of making the checkpointing variables contained in gckpp_adj_Integrator.f90 threadprivate. These variables are found near the beginning of the RosenbrockADJ subroutine. The thread private declarations are shown below.

```
!$OMP THREADPRIVATE( stack_ptr, chk_H, chk_T, chk_Y, chk_K, chk_J, chk_dY,
chk_d2Y )
```

Completing these steps successfully parallelizes the forward and adjoint KPP integrators.

## 5.2 Results for the parallel forward model with KPP

In order to demonstrate the performance improvements obtained by parallelizing the KPP code, we first test the forward OpenMP KPP code against the OpenMP SMVGEAR code and forward serial KPP code. We ran a 24-hour GEOS-Chem simulation for each set of code using 1, 2, 4, and 8 processors. CPU time and speedup results are shown in Fig 15 below.



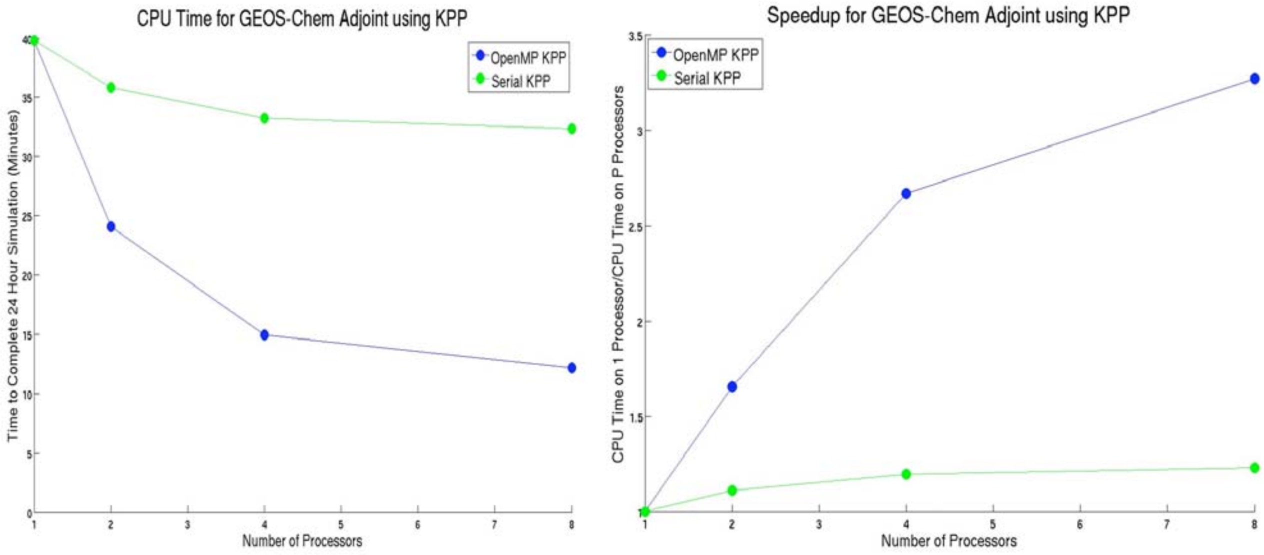
(a) CPU time

(b) Speed up

**Fig 15.** Parallelization results for a 24-hour GEOS-Chem forward model run with 1, 2, 4, and 8 cores using OpenMP KPP, OpenMP SMVGEAR, and Serial KPP.

## 5.3 Results for the parallel adjoint model

Similar tests are performed with the adjoint KPP code and comparing the parallel KPP adjoint to the serial KPP adjoint. We ran a 24-hour GEOS-Chem simulation for each code using 1, 2, 4, and 8 processors. The run times and the speed up are reported in Fig 16. The adjoint code consists of a forward part which checkpoints data and a backwards part which uses the check pointed data. Therefore in addition to the overall adjoint timing tests, we also time the forward and backwards sections separately and report the results in Fig 17.



(a) CPU time

(b) Speed up

**Fig 16.** Parallelization results for a 24-hour GEOS-Chem ADJOINT run with 1, 2, 4, and 8 cores using OpenMP KPP and Serial KPP. The difference illustrate the gains due exclusively to the parallelization of the chemistry.

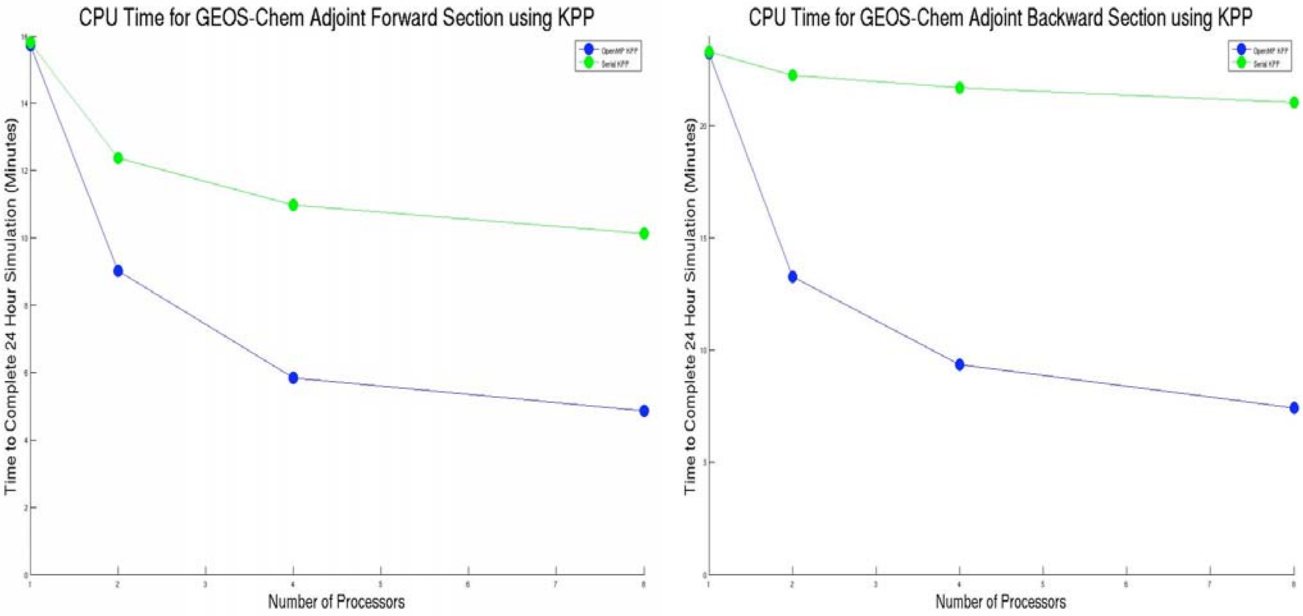


**Fig 17(a).** Parallelization results for a 24-hour GEOS-Chem ADJOINT run with 1, 2, 4, and 8 cores. Shown are the separate CPU times of the forward and of the backward sections of the adjoint.

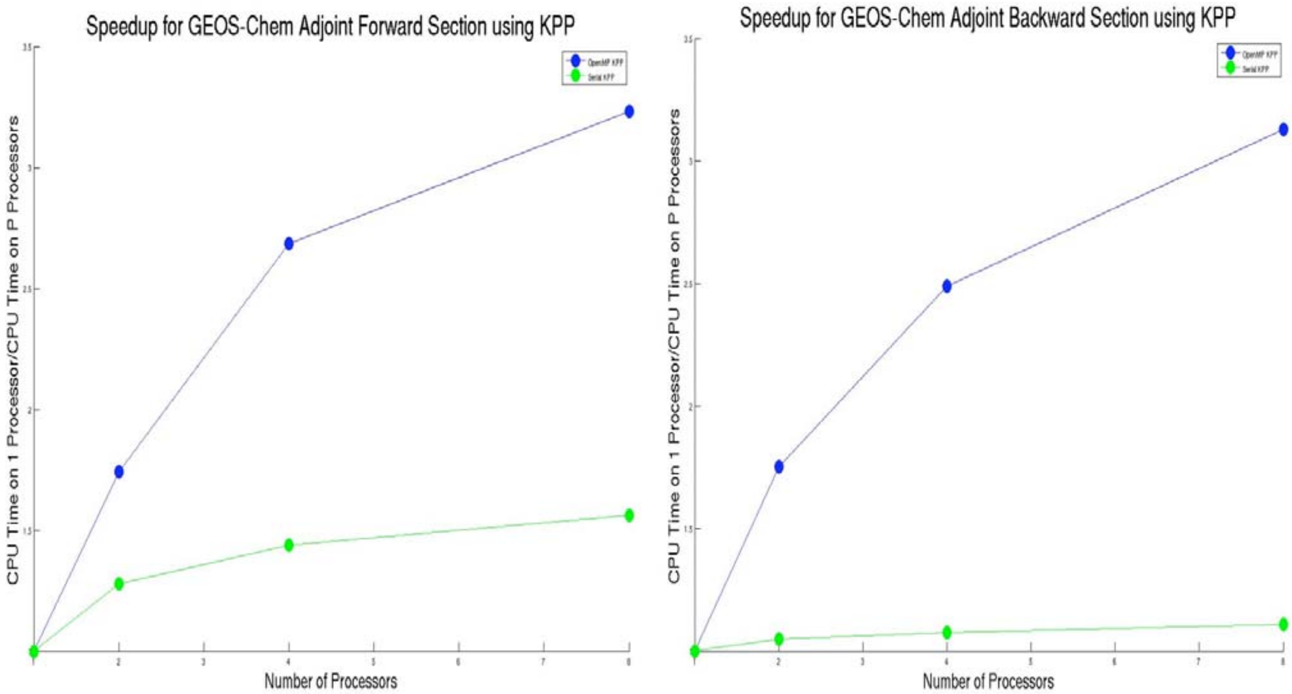


**Fig 17(b).** Parallelization results for a 24-hour GEOS-Chem ADJOINT run with 1, 2, 4, and 8 cores. Shown are the separate SPEEDUPS of the forward and of the backward sections of the adjoint.