Adjoint Working Group

Adjoint modeling clinic

- lots of excited new users!

- overview of applications best suited for adjoints (MK)

- reviewed current applications (MK)

- grassroots distribution, development, credit
  ~ GEOS-Chem forward model (DH)

- code demonstration of GCv7 adjoint (KS)

- google “geos chem adjoint wiki” for more info

Adjoint Working Group

Working group

"ask not what the adjoint can do for you, but what you can do for the adjoint"

Adjoint Working Group

Working group

- adjoint codes must and will be merged
  - we have largely moved beyond R&D stages for "current" capabilities
  - necessary step for getting new folks involved
  - multiple stages, completed by end of summer 2009

- reviewed interests of new users
  - covers almost everything!
  - mostly easily doable with existing code

Adjoint Working Group

What most users will need to contribute (~5%)

- adjoints calculate:

$$\frac{\partial J}{\partial p}$$

- defining the cost function, *J* (observational operator)
  - several already implemented
  - can be shared

- defining control parameters, *p*, of interest
  - a post processing step

Adjoint Working Group

Outstanding issues for development

- GEOS-5
  - large bookkeeping task
  - minimal adjoint code dev task

- nested, high resolution capabilities
  - current is 4x5, 2x2.5
  - nesting would be entirely “new”, not just for GCadj

- diagnostics

- aerosol adjoints
  - size
  - optics, RT

- generic tagged adjoint? (PK)

Adjoint Working Group

Long term integration issues

- GEOS-Chem version 8?

- ESMF?

- do we continue to play catch-up or try to leap frog?

Adjoint Working Group

adjoint code developing parallel to forward GC because:

- integration requirements (FD, inverse checks)
  - requires adjoint-skilled gatekeeper (DH)

- distribution of current adj code experts

- shift towards distributed management of GC

Obtaining the adjoint codes

Step 1: Sign up on the adjoint group mailing list:

geos-chem-adjoint@seas.harvard.edu

- stay informed of bug patches
- new releases (many coming out soon after this meeting)

Step 2: Peruse the adjoint wiki:

http://wiki.seas.harvard.edu/geos-chem/index.php/GEOS-Chem_Adjoint

- find out which adjoint version you need
- read the manual (linked from the wiki page)

Step 3: Get code from CVS server

- email daven.henze@colorado.edu for an account
- CVS server is moving from Caltech to CU Boulder very soon
- instructions for accessing are in the users manuals

Obtaining the adjoint codes

...Step 3: CVS is good for:

- tracking changes you've made to the code
- upgrading to changes we've made to the code

e.g., adding 2x2.5 support:

```
106,107c102
<       REAL*4, ALLOCATABLE :: QC_SO2_CHEMT(:,:,:,:)
<       REAL*4, ALLOCATABLE :: QC_SO2_DYNT(:,:,:,:)
---
>       REAL*4, ALLOCATABLE :: QC_SO2_CHK(:,:,:,:)
158a154
> !  (11) Now completely split dynamic from chemical time step checkpoints
   !         (dkh, 02/01/09)
354,365c350,362
```

Caveats about the code

Code is distributed following GEOS-Chem grass roots policy

- open access

- get code from the source

- share what you've done too

- give proper credit

Running the code

Benchmark simulations

- following links from wiki, download benchmark inputs / outputs
- try it on your computer!

Benchmark packages and results

- ADJv29_bench_opt_geos3.tar.gz
- ADJv29_bench_opt_geos3_lin.tar.gz
- ADJv29_bench_opt_geos4.tar.gz
- ADJv29_bench_fd_geos3.tar.gz
- ADJv29_bench_fd_geos3_short.tar.gz
- ADJv29_bench_fd_geos4.tar.gz

Supporting tools for plotting in IDL / matlab

- tools.tar.gz

Supporting files

- IDL / matlab tools for plotting results

Adjoint model setup

Define your active variables: (CMN_ADJ)

```
!=============================================================
      !          **** ACTIVE VARIABLE SELECTION ****
      ! Set type of inverse problem to solve. Only uncomment one of
      ! the following sections.
      !=============================================================
      ! INITIAL CONDITIONS
!     CHARACTER(LEN=10)   :: ACTIVE_VARS = 'TRACERS'
!     INTEGER, PARAMETER  :: NOPT       = IIPAR * JJPAR * LLPAR * NADJ
      !-------------------------------------------------------------
      ! EMISSION SCALE FACTORS
!     CHARACTER(LEN=10)   :: ACTIVE_VARS = 'EMISSIONS'
!     INTEGER, PARAMETER  :: NOPT       = IIPAR * JJPAR * MMSCL * NNEMS
      !-------------------------------------------------------------
      ! FINITE DIFFERENCE TEST
      CHARACTER(LEN=10)   :: ACTIVE_VARS = 'FDTEST'
      INTEGER, PARAMETER  :: NOPT       = IIPAR * JJPAR * MMSCL * NNEMS
```

Adjoint model setup

Pick a type of cost function (cpp directives in CMN_ADJ):

```
!#define JACOBIAN      'JACOBIAN'
#define PM_ATTAINMENT 'PM_ATTAINMENT'
!#define O3_ATTAINMENT 'O3_ATTAINMENT'
!#define NO2_SAT_OBS   'NO2_SAT_OBS'
!#define IMPROVE_OBS   'IMPROVE_OBS'
!#define CASTNET_OBS   'CASTNET_OBS'
```

$$\mathcal{J}_a = \frac{1}{2} \sum_{i \in \mathrm{US}} \sum_{\mathrm{day}\ j} \theta(a_{i,j}) a_{i,j}^2,$$

where

$$a_{i,j} = \left(\sum_{\hat{k}} \bar{c}_{i,j,\hat{k}} \right) - \bar{c}_a, \quad \hat{k} = \{SO_4^{2-}, NO_3^-, NH_4^+\}, \quad \theta(a) = \begin{cases} 0 & a \le 0 \\ 1 & a > 0 \end{cases}$$

and some other options:

```
#define LOG_OPT       'LOG_OPT'
!#define FD_GLOB      'FD_GLOB'
```

$$p = p_a e^{\sigma}$$

Adjoint model setup

Set the number of iterations and folder locations in the run script:

```
# Set the start (or current ) iteration number
X=1

# Set the stopping iteration number
XSTOP=1

# Give every run a unique name (default is $PBS_JOBID)
RNAME=ADJv29d_f04C0
```

Set some “classic” geos-chem input files (input.geos, input.ctm)

```
TIMING VARIABLES--4 x 5rRUN------+-------+-------+---------+-------+--------+
020401 000000 020408 000000     INYMDb  INHMSb  INYMDe    INHMSe  I       I
 21600   1800 000000                INDT    INTDT   INDIAGTIMEI       I       I
GEOS-CHEM FLAGS-----------------+-------+-------+---------+-------+--------+
T   T   T   T   F          ILEMIS  ILDRYD  ILCHEM    ILTRAN  ILTPFV   I
T   T   T   F   T          ILTURB  ILCONV  ILWETD    ILDBUG  ILMONOT  I
F   T   T   F   F          ILWAIT  ILBBSEA ILUNZIP   ILSVGLB ILTOMSAI I
F   T   F   F   T          ILMFCT  ILFILL  ILSTDRUN  ILDEAD  ILSHIPSO2I
T   T   T   T   F          ILSULF  ILCARB  ILDUST    ILSSALT ILATEQ   I
```

Results

in the gctm.gdt.01 file:

```
IDL> gamap, file='gctm.gdt.01'
      CATEGORY ILUN TRCNAME   TRC       UNIT      TAU0(DATE)       DIMENSIONS
  1 : IJ-GDE-$   21 dJdSOx1  9801        none 179664.00(2005070100)  72 46  1
  2 : IJ-GDE-$   21 dJdSOx2  9802        none 179664.00(2005070100)  72 46  1
  3 : IJ-GDE-$   21 dJdSO2s  9803        none 179664.00(2005070100)  72 46  1
  4 : IJ-GDE-$   21 dJdSO2b  9804        none 179664.00(2005070100)  72 46  1
  5 : IJ-GDE-$   21 dJdSO2b  9805        none 179664.00(2005070100)  72 46  1
  6 : IJ-GDE-$   21 dJdNH3b  9806        none 179664.00(2005070100)  72 46  1
  7 : IJ-GDE-$   21 dJdNH3b  9807        none 179664.00(2005070100)  72 46  1
  8 : IJ-GDE-$   21 dJdNH3a  9808        none 179664.00(2005070100)  72 46  1
  9 : IJ-GDE-$   21 dJdNH3n  9809        none 179664.00(2005070100)  72 46  1
```

note: you will need the tracerinfo.dat and diaginfo.dat files provided in the “tools” package

Developing new capabilities

Overall, adjoint models calculate gradients

$$\frac{\partial J}{\partial p}$$

For your application, what is *J*? Does GEOS-Chem already calculate *J*?

What is *p*? Name the GEOS-Chem specifically variable.

e.g., I want sensitivity w.r.t NOx emissions. OK. which NOx emissions?

- REMIS
- RRATE
- GEMISNOX2
- NOXTOT
- EMIS_BL
- EMISRRN
- ...

Adding your own code

So you added a new (tracer, process, emission, reaction) to the adjoint. GREAT!

But did you

- check the adjoint with finite difference comparison?

- try your inverse model with pseudo observations?

Validating code (IDL/matlab scripts provided)