GEOS-Chem Adjoint User's Guide (gcadj_v35)

Questions: Yanko Davila (yanko.davila@colorado.edu),
Daven Henze (daven.henze@colorado.edu)

Contents

1 Getting started

1.1 Brief overview

The GEOS-Chem adjoint model is an adjoint model derived from the GEOS-Chem CTM. Although in terms of GEOS-Chem activities, it is one of many projects in terms of code itself, it is a super-structure, containing both, the forward GEOS-Chem and its derivative adjoint code. Great effort is made to keep the adjoint current with updates in the GEOS-Chem, which have to be implemented manually. Currently, there is not an adjoint equivalent of every part of GEOS-Chem.

The FORTRAN code that is GEOS-Chem adjoint can perform a number of calculations: sensitivity calculations (most efficient if y in $\partial y / \partial x$ is a scalar, and x is a 2-, 3- or 4-dimensional field). The adjoint code can also be used for inverse problems, although some code development might be required to interface with observational datasets. Currently, some observation operators are available for several species.

Original work on the adjoint of GEOS-Chem began in 2003, focusing on the adjoint of the offline aerosol simulation. By 2005, the adjoint was expanded to include a tagged CO simulation and a full chemistry simulation (*Kopacz et al.*, 2009a,b; *Henze et al.*, 2007, 2009); an adjoint of GEOS-Chem v7 was also developed in the following years (*Zhang et al.*, 2009; *Singh et al.*, 2009). Each of these branches of the adjoint code were been constructed in a hybrid fashion using a combination of automatic differentiation software (TAMC, KPP) and manual coding of both discrete and continuous adjoints. They shared many common elements yet had unique features for different applications. During the summer of 2009, the existing branches were merged and updated to bring the adjoint into alignment with the latest release of GEOS-Chem, v8-02-01. This merged adjoint model is now the standard adjoint code into which all further development efforts will be placed.

The adjoint model is maintained by a group of its users. The Adjoint Model Scientist is Prof. Daven K. Henze at University of Colorado, and the Adjoint Code Support specialist is Yanko Davila. Questions regarding this manual and code in general can be directed to them (daven.henze@colorado.edu; yanko.davila@colorado.edu).

1.2 Recent and ongoing updates

See the wiki for a complete list of features that are implemented and/or in the process of being updated to the GC v8 adjoint.

1.3 Obtaining the adjoint model

Model packages including source code and run directories are available through the GIT repository. This Quick Start Guide goes through the process of how to create an account get the latest version of the code.

More information on GIT is at GIT Documentation and GIT Man Page. Please review this online documentation prior to using GIT. Once you have familiarized yourself with the software you may proceed.

Initial download To download the most current copy of the code, enter the following command

```
git clone git@adjoint.colorado.edu:yanko.davila/gcadj_std.git
```

This will download a project directory *"gcadj_std"* containing the source code and run directories.

Tracking subsequent changes by yourself and others Perhaps you already have a version of model. To determine the status of your existing project vs the current repository version, enter your local copy of your project directory and type

```
git status
```

To determine the difference between your local copy and the current repository copy, type

```
git diff --word-diff=color origin/master
```

The *—word-diff=color* option makes the output with colors so it's easy to read, specifiying *origin/master* takes the difference between your copy and the newest repository copy. Without *origin/master*, you will see the difference between your copy and the version that was in the repository as of the last time you checked out the code.

The commands above are without arguments and will thus apply to all files in the project. To use them for only one specific file, for example *geos_chem_adj_mod.f*, type:

```
git diff --word-diff=color origin/master -- geos_chem_adj_mod.f
```

To replace your local copy of *geos_chem_adj_mod.f* with the newest version from the repository,

```
git checkout origin/master -- geos_chem_adj_mod.f
```

If you have also modified your local copy, you can merge your changes with those made to the repository version

```
git checkout --patch origin/master -- geos_chem_adj_mod.f
```

If you want your updates to be added to the main repository please create a PATCH and contact the Support Team. Yanko Davila will handle incorporating changes into the code repository.

1.4   Additional files for analysis

Some IDL and MATLAB scripts for plotting results of finite difference tests are available here.

1.5   Benchmark simulations

Setting input files   You will need to make the following changes to the input files, as they depend upon your particular filesystem:

- *run* script
  - Set `DRUN` and `DSAVE` in the *run* script, as dictated by your filesystem.
  - Depending upon your computer system, you may or may not need to include coping of you project directory to the $DSAVE filesystem after each iteration. If you won't lose access to the local filesystem where you model is running after it has completed executing, then you don't need to set `DSAVE`
  - If you are running on multiple CPU cores, be mindful of the line
    `export OMP_NUM_THREADS=24`
    and adjust as appropriate for your system.
- Set data folder locations in *input.geos* following standard forward model procedures.
- Set NetCDF locations in *Makefile*
- Set HDF locations in *Makefile* (if your simulation requires these)

Setting source code   The source code is by default to run the *geos5* benchmark. To use the *geos4* benchmark,

- change the preprocessor flag for meteorology from GEOS_5 to GEOS_4 in *code/define.h*
- disable the IN_CLOUD_OD preprocessor flag in *code/define.h*.

- enable the PSEUDO_OBS preprocessor flag in *code/adjoint/define_adj.h*.

Running the tests Now you are ready to execute the run script, with a unix command like:

```
./run > log.my_benchmark_test &
```

If all goes well, the output should finish with:

```
 -------------------------------------------------
 G E O S   C H E M   A D J O I N T   E X I T E D
          N O R M A L L Y
 -------------------------------------------------
```

1.5.1 The geos5 full chemistry finite difference test

The *geos5* benchmark simulation is a full chemistry sensitivity test. It checks the sensitivity of O_x with respect to NO_x using adjoints and finite difference calculations. The results are in the *diagadj/*.fdglob.** file.

To speed up the evaluation of this test, chemistry is only calculated in the `LFD` level.

This simulation will run for 1 complete iteration, and then only the forward part of the 2nd and 3rd iterations, after which the log file will state something like:

```
 Global validation test for values >    1.00000000000000
 MAX of global 2nd order ADJ / FD  =    3.411938
 MIN of global 2nd order ADJ / FD  =   -6.9405900
 Number of places where ratio off by   0.100000000000000    =      88
```

That the min and max of the ADJ / FD ratio is much greater or less than 1.d0 may or may not be significant. In general, it shouldn't be off by more than x10, and the number of places where the ratio is off by more than 10% shouldn't be much more than 100. However, to tell whether ADJ / FD ratios that deviate from 1.0 are owing to errors in the adjoint code or to limitations of finite difference sensitivities, the results will have to be plotted.

The key results are saved to the **.fdglob.** file. The contents can be analyzed using GAMAP. You can load a file and view results such as finite sensitivities, adjoint sensitivities, and the ratio of the two.

```
IDL> gamap, file='gctm.fdglob.20050701.0500'
```

Next generate a data set for a scatter plot of adjoint vs finite difference values using (thanks to Kevin Wecht for providing this IDL script):

```
IDL>plot_fdglob
```

Alternatively, if you prefer to generate figures in MATLAB, scripts for the following scheme are also provided. First output the sensitivities to some text files.

```
IDL> fd_stats
```

and then plot the results in MATLAB using

```
>>fd_vs_adj
```

Sample output is shown here:

```
http://spot.colorado.edu/~henzed/GC_adj/fd_validation.pdf
```

More about setting up and designing your own finite difference validation tests is in Sect. 3.4.

1.5.2 The geos4 tagged CO optimization test

The optimization benchmark simulation attempts to optimize initial concentrations of CO, starting with domain wide linear scaling factors of 0.5 and pseudo observations that were generated with scaling factors of 1.00. The simulation is only 1 day long and should start to converge after a few iterations.

Output from our benchmark runs are in the directory *../OptData/*. To quickly see how the cost function converges,

```
grep '' OptData/cfn*
```

The cost function should have reduced by about an order of magnitude. Note: not all value of the cost function listed here correspond to accepted iterations of the optimization procedure. Some are 'function evaluations' that correspond to the optimization performing searches in various directions before finding the optimal path towards a minimum. To see which values correspond to accepted iterations,

```
grep 'iterate' log
```

To see the inverse modeling solution after the 6th iteration, look at the optimized scaling factors in the *gctm.sf.06* file using gamap,

```
IDL> gamap, file='gctm.sf.06', 'IJ-ICS-$', yrange=[0,1]
```

Scaling factors should be close to 1.0 in locations where CO concentrations are significant.

7

2 Directories and input/output files

2.1 Directory structure

The adjoint code package *gcadj_std* contains the following directories:

code/	Code directory, which contains all unmodified GEOS-Chem files, a Makefile and a few subdirectories relevant to the adjoint code, listed below. This is also where all the object files are placed.
code/modified/	Subdirectory that contains all (forward) GEOS-Chem files that have been modified for the adjoint.
code/adjoint/	Subdirectory that contains all adjoint specific files.
code/obs_operators/	Subdirectory contains all files relevant to observation operators.
code/new/	Subdirectory contains all new files (new to forward AND adjoint).
runs/	Run directory with subdirectories for each met field type.
runs/v8-02-1/geos/*	Run directories with subdirectories for output files.
runs/../geos/adjtmp/*	Adjoint temporary file directory (*gctm.chk.*; gctm.obs.*; gctm.adj.**), the name can be changed in *input.gcadj*.
runs/../geos/tmp/*	Forward model temporary file directory (unzipped met fields).
runs/../geos/OptData/*	Results from each iteration (*gctm.gdt.*; cfn.*; gctm.sf.*, fwd_dat.*.tar; gctm.arr*), the name can be changed in *input.gcadj*.
runs/../geos/diagadj/*	Adjoint diagnostic files (*.fd.*;*.fdglob.*; aero.ave.*; satave.*; jsave.*; gctm.iteration, emis.adj.**), the name can be changed in *input.gcadj*.

2.2 Input files

2.2.1 *input.gcadj*: the main adjoint input file

input.gcadj file contains the following options:

```
01: %%% ADJOINT SIMULATION MENU %%%
02: Do adjoint run    LADJ         : T
03: Select one simulation type     :---
04: Invese problem    L4DVAR       : T
05: Kalman filter     L3DVAR       : F
06: Sensitivity       LSENS        : F
07:  => spot finite diff  FD_SPOT  : F
08:  => global finite diff FD_GLOB : F
```

Description

01: header line	–
02: *LADJ*	Global switch for adjoint option. If set to FALSE, it will overwrite all other options and make the run a forward mode only run.
03: line	If LADJ = T, need to pick one of the following options. 3DVAR not yet supported.
04: *L4DVAR*	Switch for 4d-var runs.
05: *L3DVAR*	Switch for Kalman filter.
06: *LSENS*	Switch for sensitivity runs. If performing a finite difference test, pick one option from below:
07: *FD_SPOT*	Switch for spot finite difference test.
08: *FD_GLOB*	Switc for global finite difference test).

```
01: %%% FORWARD MODEL OPTIONS %%%
02: adjoint chemistry LADJ_CHEM    : T
03: aerosol thermo    LAERO_THEM   : T
04:  => ISORROPIAII              : T
```

Description

01: header line	–
02: *LADJ_CHEM*	Switch for adjoint chemistry option. If set to FALSE, it will turn off adjoint chemistry. Make sure that LCHEM is set to the same value.
03: *LAERO_THEM*	Switch for aerosol thermodynamics option (applies to forward and adjoint).
04: *LISO*	Switch for ISORROPIA II aerosol thermodynamics option (applies to forward and adjoint). Note if LISO is set to FALSE, then RPMARES will be used instead.

```
01: %%% ADJOINT MODEL OPTIONS %%%
02: Include a priori term APSRC    : F
03:  => offdiagonal                : F
04: Compute DFP inverse Hessian    : F
05: Compute BFGS inverse Hessian   : F
06: Include rxn rate sensitivities : F
07: Delete chk files LDEL_CHKPT    : T
08: Scale up and FILL adj transport: F
```

Description

01: header line	–	
02: *APSRC*	Switch for calculating the a priori term in the cost function. Valid option for 4d-var runs.	
03: *LBKCOV_ERR*	Switch for computing non-diagonal background error covariance matrices.	
04: *LINVH*	Switch for computing an approximate inverse Hessian matrix using the DFP method.	
04: *LBFGS*	Switch for computing an approximate inverse Hessian matrix using the L-BFGS method.	
06: Rxn sensitivities	Switch for storing sensitivities wrt reaction rates.	
07: LDEL_CHKPT	Delete checkpoint files after they are used in adj run. Set to F to reuse them for multiple adj runs.	
08: LFILL_ADJ	Scale up adjoints and then use the LFILL option in tpcore for advection.	

```
01: %%% DIRECTORIES %%%
02: Optimization output         : OptData/
03: Temporary adjoint dir adjtmp : /big_scratch/temp/
04: Diagnostics ouptut          : diagadj/
```

Description

01: header line	–
02: OptData.	Specify output directory where essential output will go, typically set to OptData.
03: adjtmp.	Specify output directory where nonessential output will go, typically set to adjtmp.
04: diagadj	Specify output directory where diagnostic output will go, typically set to diagadj.

```
01: %%% CONTROL VARIABLE MENU %%%
02: Initial conditions LICS        : F
03: ... OR emissions    LADJ_EMS   : T
04:  => strat prod/loss LADJ_STRAT : T
05:  => reaction rates LAJ_RRATE   : F
06: >---------------------------<
07:          FOR LICS             :
08: NSOPT: number of tracers opt   : 1
09:   => opt these tracers------>  : TRC# trc_name SF_DEFAULT REG_PARAM ERROR
10: Tracer  #1                     : 1    NOx       1          1         1
11: >---------------------------<
12:          FOR LADJ_EMS          :
13: NNEMS: ems groups implemented  : 33
14: Emission entries ------------> : EMS# ems_name      opt  SF_DEFAULT REG_PARAM ERROR CORR_LX CORR_LY
15: Emission #1                    : 1    IDADJ_ENH3_an  T   1          1         1     100     100
16: Emission #2                    : 2    IDADJ_ENH3_na  T   1          1         1     100     100
17: Emission #3                    : 3    IDADJ_ENH3_bb  T   1          1         1     100     100
18: Emission #4                    : 4    IDADJ_ENH3_bf  T   1          1         1     100     100
19: Emission #5                    : 5    IDADJ_ESO2_an1 T   1          1         1     100     100
20: Emission #6                    : 6    IDADJ_ESO2_an2 T   1          1         1     100     100
21: Emission #7                    : 7    IDADJ_ESO2_bf  T   1          1         1     100     100
22: Emission #8                    : 8    IDADJ_ESO2_bb  T   1          1         1     100     100
23: Emission #9                    : 9    IDADJ_ESO2_sh  T   1          1         1     100     100
24: Emission #10                   : 10   IDADJ_EBCPI_an T   1          1         1     100     100
25: Emission #11                   : 11   IDADJ_EBCPO_an T   1          1         1     100     100
26: Emission #12                   : 12   IDADJ_EOCPI_an T   1          1         1     100     100
27: Emission #13                   : 13   IDADJ_EOCPO_an T   1          1         1     100     100
28: Emission #14                   : 14   IDADJ_EBCPI_bf T   1          1         1     100     100
29: Emission #15                   : 15   IDADJ_EBCPO_bf T   1          1         1     100     100
30: Emission #16                   : 16   IDADJ_EOCPI_bf T   1          1         1     100     100
31: Emission #17                   : 17   IDADJ_EOCPO_bf T   1          1         1     100     100
32: Emission #18                   : 18   IDADJ_EBCPI_bb T   1          1         1     100     100
33: Emission #19                   : 19   IDADJ_EBCPO_bb T   1          1         1     100     100
34: Emission #20                   : 20   IDADJ_EOCPI_bb T   1          1         1     100     100
35: Emission #21                   : 21   IDADJ_EOCPO_bb T   1          1         1     100     100
36: Emission #22                   : 22   IDADJ_ENOX_so  F   1          1         1     100     100
37: Emission #23                   : 23   IDADJ_ENOX_li  F   1          1         1     100     100
38: Emission #24                   : 24   IDADJ_ENOX_ac  F   1          1         1     100     100
39: Emission #25                   : 25   IDADJ_ENOX_an  F   1          1         1     100     100
40: Emission #26                   : 26   IDADJ_ENOX_bf  F   1          1         1     100     100
41: Emission #27                   : 27   IDADJ_ENOX_bb  F   1          1         1     100     100
42: Emission #28                   : 28   IDADJ_ECO_an   F   1          1         1     100     100
43: Emission #29                   : 29   IDADJ_ECO_bf   F   1          1         1     100     100
44: Emission #30                   : 30   IDADJ_ECO_bb   F   1          1         1     100     100
45: Emission #31                   : 31   IDADJ_EISOP_an F   1          1         1     100     100
46: Emission #32                   : 32   IDADJ_EISOP_bf F   1          1         1     100     100
47: Emission #33                   : 33   IDADJ_EISOP_bb F   1          1         1     100     100
48: Number emis time group MMSCL   : 1
49: >---------------------------<
50:          FOR LADJ_STRAT        :
51: NSTPL: strat prod & loss trcs  : 24
52: Read reactions from STR_ID file: T
53: Strat prod & loss trc entries  : ID# trc_name    opt   SF_DEFALUT  REG_PARAM ERROR
55:>---------------------------<
56:          FOR LADJ_RRATE        :
57: NRRATES: num of rxn rates      : 297
58: Read reactions from RXN_ID file: T
59: ...or use these Rxn rates      : ID#  rxn_name  opt SF_DEFAULT REG_PARAM ERROR
```

Description

01: header line	Need to pick one of the four possible sets of control parameters: initial conditions, emissions, stratospheric production and loss rates or reaction rates. Emissions must be turned on to select stratospheric fluxes and reaction rates. NOTE: You can select stratospheric fluxes and reaction rates at the same time or each of them individually.
02: LICS	Tracer initial conditions as control parameters.
03: LADJ_EMS	Emissions as control parameters.
04: LADJ_STRAT	Stratospheric production and loss rates as control parameters.
05: LADJ_RRATE	Reaction Rates as control parameters.
06: spacer line	-
07: FOR LICS	Specify which tracers to allow as control parameters. Note: the range of possible tracers is defined in input.geos. The adjoint will always include adjoints of all tracers. So here we just need to list which of these tracers will be optimized. All and only those tracers listed below will be optimized. This LICS section of the MENU will be ignored if LADJ_EMS = T.
08: NSOPT	Total number of tracers to optimize listed in the submenu below.
09: subheader	-
10: Tracers #1	List the corresponding tracer number (TRC#) and name (trc_name) from input.goes. Here you can also specify a global default scaling factor (SF_DEFAULT) for the first iteration, a regularization parameter (REG_PARAM) and an error (ERROR). The latter two only have an effect if LAPSRC = T.
. . .	add more lines like 08 if you want to make the initial conditions for more than one tracer active. . . .
11: spacer line	-
12: FOR LADJ_EMS	Specify all of the emissions adjoints that are currently implemented. This LADJ_EMS section of the menu will be ingnored if LICS = T.
13: NNEMS	Total number of active emissions groups listed in the submenu below.
14: subheader	-
15: Emission #1	List the emission number (EMS#) and name (ems_name). Names must begin with IDADJ_E.... Select wether this emissions group is to be optimized (opt). Here you can also specify a global default scaling factor (SF_DEFAULT) for the first iteration, a regularization parameter (REG_PARAM), and an error. REG_PARAM and ERROR only have an impact of APSRC = T. The last two columns (CORR_LX and CORR_LY) are the correlation lenght in Km used when calculating non-diagonal background error covariance matrices.
15. . . 46	additional emission group definitions and options.
48: MMSCL	Number of emissions sub-scaling groups. MMSCL > 1 not yet supported.
49: spacer line	-

50: FOR LADJ_STRAT	Specify all of the stratospheric production and loss rate adjoints that are currently implemented. This LADJ_STRAT section of the menu will be only considered if both LADJ_EMS = T and LADJ_STRAT = T.
51: NSTPL	Total number of active stratospheric tracers that have production and loss rates listed in the menu below. NOTE: each tracer has production and loss rates.
52: FI_STRID	Read active stratospheric tracers that have production and loss rates from file.
53: subheader	-
54: Tracer #1	List the corresponding tracer number (ID#) and name (trc_name). Names must end with . . . _p (production) or. . . _l (loss). Select whether this stratospheric flux is to be optimized (opt), although optimization of the stratospheric fluxes has yet to be fully tested. Here you can also specify a global default scaling factor (SF_DEFAULT) for the first iteration, a regularization parameter (REG_PARAM), and an error. For now, only SF_DEFAULT is supported.
55: spacer line	-
56: FOR LADJ_RRATE	Specify all of treaction rate adjoints that are currently implemented. This LADJ_RRATE section of the menu will be only considered if both LADJ_EMS = T and LADJ_RRATE = T.
57: NRRATES	Total number of active reaction rates listed in the menu below.
58: FI_RXNID	Read active reaction rates from file.
59: subheader	-
60. . .	For additional reaction rates and options see Section 4.1

```
01: %%% OBSERVATION MENU %%%
02: %%%  for PSUEDO_OBS  %%%
03: %%%   or LSENSE      %%%
04: Observation frequency OBS_FREQ : 60
05: Limit number of observations?  : F
06:    =>  Forcing time till       : 20050701 050000
07: COST FUNCTION options for LSENS:---
08:    =>  tracer kg/box           : T
09:    =>  tracer ug/m3            : F
10:    =>  tracer ppb              : F
11:    =>  tracer ppm free trop    : F
12:    =>  species ppb w/averaging : F
13:    =>  tracer ug/m3 pop weight : F
14:    =>  tracer ug/m2/hr         : F
15:    =>  deposition based?       : F
16:        =>  dry dep (not kpp)   : F
17:        =>  dry dep (kpp)       : F
18:        =>  tracer wet LS dep   : F
19:        =>  tracer wet CV dep   : F
20:        =>  molec/cm2/s         : F
21:        =>  kgN/ha/yr           : F
22:        =>  eq/ha/yr            : F
23:        =>  kg/s                : F
24:    =>  Regional mask?          : F
25:        =>  binary punch file?  : F
26:          =>  mask name         : usa_mask.geos.4x5
27:        OR  netcdf file    ?    : F
28:          =>  nc mask file name : /home/hyungmin/Class_1/Class1.nc
29:          =>  nc mask var  name : NPS_16
30: >---------------------------<
31: NOBS: number of tracers to obs : 2
32:  => obs these tracers------>  : TRC# tracer_name
33: Tracer  #1                    : 34   BCPI
34: Tracer  #2                    : 35   OCPI
35: >---------------------------<
36: NOBS_CSPEC: # of species to obs: 0
37:  => obs these species------>  :species_name
38: Species #1                    : O3
```

Description

01: header line	The options pretty much pertain only to sensitivity calculations or pseudo observation tests.
02: header line	The only exception to that is if you have an observation operator specific to a chemical species.
03: header line	which is not a tracer (e.g., O_3), in which case you need to specify it in the observed species submenu.
04: *OBS_FREQ*	Frequency (in min) of checking and assimilating observations, both pseudo and real, typically 60.
05 *LMAX_OBS:*	Set this if you wish to limit the number of observations. For example, if you want the cost function to be evaluated only during the final day of your simulation, and OBS_FREQ = 60, then set LMAX_OBS = T and NSPAN = 24. Setting FD_GLOB will trigger LMAX_OBS = T and NSPAN = 1.
06: *NSPAN*	If LMAX_OBS = T, then use this to set the number of hours of adjoint forcing. In this example 1 hour as the Simulation Stop time is 20050701 060000. Setting FD_GLOB will trigger LMAX_OBS = T.
07: subheader	Below are some options for evaluating the cost function during a sensitivity run. Note the distinction between tracers (STT) and species (CSPEC). Some of these options include the WEIGHT array, which allows for spatial masking. Check the respective code segments for details.
08: *LKGBOX*	Evaluate the cost function for tracer concentrations in units of kg/box. Note: FD simulations will default to this option.
09: *LUGM3*	Evaluate the cost function for tracer concentrations in units of ug/m3.
10: *LSTT_PPB*	Evaluate the cost function for tracer concentrations in units of ppb.
11: *LSTT_TROP_PPM*	Evaluate the cost function for tracer concentrations only in the free troposphere in units of ppm.
12: *LCSPEC_PPB*	Evaluate the cost function for species concentrations in units of ppb, averaged over the range NSPAN. There are also hardwired options within CALC_ADJ_FORCE_FOR_SENS that can be used to specify a subdomain over which to average: LMIN, LMAX, JMIN, JMAX, IMIN, IMAX.
13: *LPOP_UGM3*	Domain-wide average population weighted aerosol concentrations
14: *LFLX_UGM2*	Evaluate the cost function for tracer concentrations in units of flux at single level [ug/m2/hr]. Default is L = 1. See adjoint/input_adj_mod.f to change the level.

15:	*LADJ_FDEP*	Evaluate a deposition-based cost function.
16:	*LADJ_DDEP_TRACER*	Tracer dry deposition handled outside KPP.
17:	*LADJ_DDEP_CSPEC*	Species dry deposition handled with KPP.
18:	*LADJ_WDEP_LS*	Large Scale wet deposition.
19:	*LADJ_WDEP_CV*	Convective wet scavenging.
20:	*LMOLECCM2S*	Cost function units are molec/cm2/s (required for FD_TEST with DDEP)
21:	*LKGNHAYR*	Cost function units are kgN/ha/yr (required for FD_TEST with WDEP)
22:	*LEQHAYR*	Cost function units are eq/ha/yr
23:	*LKGS*	Cost function units are kg/yr
24:	*LFORCE_MASK*	Use a regional mask for the cost function.
25:	*LFORCE_MASK_BPCH*	Use a binary mask file.
26:	*FORCING_MASK_FILE*	Name (or PATH) of the regional binary mask file.
27:	*LFORCE_MASK_NC*	Use a netcdf mask file.
28:	*FORCING_MASK_FILE_NC*	Name (or PATH) of the regional netcdf mask file.
29:	*NB_MASK_VAR*	NetCDF mask variable name.
30:	spacer line	This next section is only important if you are using a cost function that involves tracers (STT).
31:	*NOBS*	The number of tracers involved in your cost function. It must match the number of tracers listed below, or if it is zero the section below will be ignored.
32:	subheader	-
33:	Tracer #1	List the ID number of the tracer from input.geos (TRC#) and its name (tracer_name). Make as many entries in this section as necessary.
34-	...:	...
35:	spacer	-
36:	*NOBS_CSPEC*	The number of species involved in your cost function. It must match the number of species listed below, or if it is zero the section below will be ignored.
37:	subheader	-
38:	Species #1	Enter the names of the species to be observed (species_name) They don't need to be ordered or numbered according to their definition in CSPEC, just make sure that the name is exactly as it is listed in globchem.dat so that the code can match the name and find the corresponding an ID index in CSPEC. Make as many entries in this section as necessary.
-...:		...

```
01: %%% FINITE DIFFERENCE MENU %%%
02: fd perturbation        FD_DIFF : 0.1
03: Numerator of derivative to test:---
04:  => longitude degree   LONFD   : 32
05:  => latitude  degree   LATFD   : 21
06:  => OR pick box by grid index? : T
07:    => longidute index  IFD     : 21
08:    => latitude index   JFD     : 31
09:  => altidude index     LFD     : 1
10:  => tracer (STT TRC#)  NFD     : 2
11: Denomenator of deriv.   to test:
12:  => w/LEMS: emis group MFD     : 1
13:  => w/LEMS: sector     EMSFD   : 9
14:  => w/LICS: tracer     ICSFD   : 1
15:  => w/LSTR: tracer     STRFD   : 1
16:  => w/LRRATE: rate     RATFD   : 1
```

Description
Selections in this menu will apply if we are performing a finite difference test, i.e. LSENS = T and either FD_GLOB or FD_SPOT = T in the ADJOINT SIMULATION MENU.

01: header line	–
02: *FD_DIFF*	The size of the finite difference perturbation that is applied to the control parameters (LICS or LADJ_EMS), depending on which ones are selected for the finite difference test.
03: spacer line	The numerator of the derivative to test is selected in the following sub-menu. Note: for debugging, it can be useful to point these indices to troublesome values and then turn on LPRINTFD in the DIAGNOSTICS MENU.
04: *LONFD*	Longitude of the gridbox in the SPOT finite difference test and debugging (converted to index online).
05: *LATFD*	Latitude of the gridbox in the SPOT finite difference test and debugging (converted to index online).
06: specify box	This flag, if set to TRUE, will set the finite difference box as specified by the gridbox indecies, and not lat/lon indecies.
07: *IFD*	Longitude index of the gridbox in the SPOT finite difference test and debugging.
08: *JFD*	Latitude index of the gridbox in the SPOT finite difference test and debugging.
09: *LFD*	Vertical level index of the gridbox in the SPOT and GLOB finite difference test and debugging.
10: *NFD*	Tracer (STT TRC#) index of the SPOT and GLOB finite difference test and in debugging. This value will override whatever tracer is listed in the OBSERVATION MENU.
11: spacer line	The denomenator of the derivative to test (for both SPOT and GLOB) is selected in the following submenu.
12: *MFD*	Emission temporal group index of the finite difference test. Only matters for LADJ_EMS.
13: *EMSFD*	Emission sector index (full chem only) of the finite difference test. Only matters for LADJ_EMS.
14: *ICSFD*	Initial conditions type of the finite difference test. Only matters for LICS.
15: *STRFD*	Stratospheric tracer index (full chem only) of the finite difference test. Only matters for LADJ_STRAT. NOTE: Finite difference test for the stratospheric fluxes is available for one of either production or loss rates. By default it is for loss rates. You can change it to production rate by modifying the code (inverse_mod.f).
16: *RATFD*	Reaction Rates type of the finite difference test. Only matters for LADJ_RRATE.

```
01: %%% DIAGNOSTICS MENU %%%
02: General                      : T
03:  => print debug  LPRINTFD    : T
04:  => jsave, jsave2            : F
05:  => adjoint traj LADJ_TRAJ   : T
06:    => w.r.t. scale factors?  : T
07:  => save iteration diags LITR : T
08:  => sense w.r.t absolute emis : F
09: CO satellite diganostics     : F
10:  => H(model)                 : F
11:  => h(obs)                   : F
12:  => H(model)-h(obs)          : F
13:  => adjoint forcing          : F
14:  => model bias               : F
15:  => observation count        : F
16:  => DOFs                     : F
17: TES NH3 diagnostics          :---
18:  => BLVMR                    : F
19: HDF diagnostics              :---
20:  => Level 2                  : F
21:  => Level 3                  : F
```

Description

01: header line	–
02: general switch	Global diagnostic switch. Needs to be set to TRUE for any of the diagnostics to be saved or printed.
03: *LPRINTFD*	Print (to log file) debugging messages and tracer values in (IFD,JFD,LFD,NFD) gridbox.
04: jsave, jsave2	Switch to save jsave and jsave2 values (debugging/finite difference output).
05: *LADJ_TRAJ*	Switch to store adjoint trajectory. Note: turning this on can generate a lot of output for fullchemistry simulations. You may want to edit the source code (routine MAKE_ADJ_FILE) so that only a subset of the adjoint species are written out, or a subset of vertical levels.
06: *LADJ_TRAJ*	Switch to save adjoint trajectories as sensitivities w.r.t scaling factors.
07: *LITR*	Save iteration diagnostic file *gctm.iteration* to the diagnostic directory. This file contains information related to convergence of the optimization routine.
08: *LEMS_ABS*	Switch to save sensitivities w.r.t. absolute values of emissions (as opposed to scaling factors). These will be in the ems.adj.* files in the diagnostic directory. Currently only supported for SO_2, NH_3, BC and OC emissions.
09: CO diags	Global switch for CO satellite diagnostics. To be used only when MOPITT, AIRS and/or SCIAMACHY observations are used. Corresponding code is very easy to adapt to other observational operators and saves information in the same space for model and observations, making model-data comparisons very easy.
10: *H(model)*	Switch to save model convolved by satellite averaging kernels, currently as a column value.
11: *h(obs)*	Switch to save satellite observations averaged on GEOS-Chem grid (1h resolution).
12: *H(model)-h(obs)*	Switch to save model-satellite differences in satellite/model space.
13: adj forcing	Switch to save adjoint forcing.
14: model bias	Switch to save model bias (model-obs)/obs. This is very useful for computing Relative Residual Error (RRE) values for optimization studies.
15: *OBS_COUNT*	Switch to save the number of observations per gridbox in a given simulation.
16: *DOF*	Switch to save an average degrees of freedom (DOF) per gridbox.
17: TES NH_3 diags	Switches for TES NH_3 assimilation
18: BLVMR	Switch to read in BLVMR values from TES and add GEOS-Chem BLVMR values to the TES data files.
19: HDF diagnostics	Switches for MOPPIT and OMI assimilation
20: *LSAT_HDF_L2*	Switch to write Level 2 Diagnostics
21: *LSAT_HDF_L3*	Switch to write Level 3 Diagnostics

```
01: %%% CRITICAL LOAD MENU %%%
02: Critical Load obs            : F
03:  => N deposition             : T
04:  => Acidity deposition       : F
05: Critical Load file           : Exceedence.nc
06: GEOS-Chem file               : Annual_Deposition.nc
```

Description

01: header line	–
02: *LADJ_CL*	Use a cost function definition based on critical load.
03: *LADJ_CL_NDEP*	Reactive nitrogen critical load.
04: *LADJ_CL_ACID*	Acidification critical load.
05: *CL_FILENAME*	Name (or PATH) of file containing gridded critical load.
06: *GC_FILENAME*	Name (or PATH) of file containing GC annual deposition.

2.2.2 *input.geos*: the main forward model input file

input.geos file contains a new section for the HTAP Experiment:

```
001: %%% HTAP SIM MENU %%%   :
002: Use HTAP v2 Emissions  : F
003: Sector Scaling Factors :---
004:  =>  AIR               : 1.0
005:  =>  SHIPS             : 1.0
006:  =>  ENERGY            : 1.0
007:  =>  INDUSTRY          : 1.0
008:  =>  TRANSPORT         : 1.0
009:  =>  RESIDENTIAL       : 1.0
010:  =>  AGRICULTURE       : 1.0
011: Species Scaling Factors :---
012:  =>  BC                : 1.0
013:  =>  CO                : 1.0
014:  =>  OC                : 1.0
015:  =>  NH3               : 1.0
016:  =>  NOx               : 1.0
017:  =>  SO2               : 1.0
018:  =>  VOCs              : 1.0
019: Source Mask Regions    :---
020:  =>  OCEANS            : F
021:  =>  US + CANADA       : F
022:  =>  EUROPE + TURKEY   : F
```

```
023:  =>   SOUTH ASIA         : F
024:  =>   EAST ASIA          : F
025:  =>   SOUTH EAST ASIA    : F
026:  =>   AUSTRALIA + NEW ZEL: F
027:  =>   NORTH AFRICA       : F
028:  =>   SUB SAHARAN AFRICA : F
029:  =>   MIDDLE EAST        : F
030:  =>   MEXICO + CARIBBEAN : F
031:  =>   SOUTH AMERICA      : F
032:  =>   RUSSIA + UKRAINE   : F
033:  =>   CENTRAL ASIA       : F
034:  =>   ARTIC CIRCLE       : F
035:  =>   ANTARCTIC          : F
036: Receptor Mask           :---
037:  =>   Use receptor mask ?: F
038:  =>   BALTIC SEA         : F
039:  =>   NORTH ATLANTIC OCEA: F
040:  =>   SOUTH ATLANTIC OCEA: F
041:  =>   NORTH PACIFIC OCEAN: F
042:  =>   SOUTH PACIFIC OCEAN: F
043:  =>   INDIAN OCEAN       : F
044:  =>   HUDSON BAY         : F
045:  =>   MEDITERRANEAN SEA  : F
046:  =>   BLACK AND CASPIAN S: F
047:  =>   NORTH EAST US      : F
048:  =>   SOUTH EAST US      : F
049:  =>   NORTH WEST US      : F
050:  =>   SOUTH WEST US      : F
051:  =>   EAST CANADA        : F
052:  =>   W CANADA, ALASKA (U: F
053:  =>   NORTH WEST EUROPE  : F
054:  =>   SOUTH WEST EUROPE  : F
055:  =>   EASTERN EUROPE     : F
056:  =>   GREECE, TURKEY, CYP: F
057:  =>   NORTH INDIA, NEPAL,: F
058:  =>   SOUTH INDIA, SRI LA: F
059:  =>   INDIAN HIMALAYA    : F
060:  =>   NORTH EAST CHINA   : F
061:  =>   SOUTH EAST CHINA   : F
062:  =>   WEST CHINA, MONGOLI: F
063:  =>   NORTH ANS SOUTH KOR: F
```

```
064:  =>  JAPAN             : F
065:  =>  CHINA, TIBET HIMALA: F
066:  =>  INDONESIA, MALAYSIA: F
067:  =>  THAILAND, VIETNAM, : F
068:  =>  PACIFIC           : F
069:  =>  AUSTRALIA         : F
070:  =>  NEW ZELAND        : F
071:  =>  EGYPT             : F
072:  =>  REST OF NORTH AFRIC: F
073:  =>  SAHEL             : F
074:  =>  CONGO,GHANA, GUINEA: F
075:  =>  BURUNDI, KENYA, ETH: F
076:  =>  ANGOLA, MALAWI, SOU: F
077:  =>  LEBANON, ISRAEL, SY: F
078:  =>  OMAN, QUATAR, YEMEN: F
079:  =>  IRAN, IRAK         : F
080:  =>  MEXICO            : F
081:  =>  CENTRAL AMERICA   : F
082:  =>  CARIBBEAN         : F
083:  =>  COLOMBIA, VENEZUELA: F
084:  =>  SOUTH BRAZIL      : F
085:  =>  REST OF BRAZIL    : F
086:  =>  URUGUAY, PARAGUAY, : F
087:  =>  PERU, ECUADOR     : F
088:  =>  RUSSIA WEST       : F
089:  =>  RUSSIA EAST       : F
090:  =>  BELORUSSIA + UKRAIN: F
091:  =>  CENTRAL ASIA      : F
092:  =>  ARTIC CIRCLE, GREEN: F
093:  =>  ANTARCTIC         : F
094:  =>  SOUTHERN OCEAN    : F
```

Description

001: header line	–
002: *LHTAP*	Switch for HTAP emissions.
003: subheader line	–
004...010:	Scaling factor for the different HTAP emissions sectors.
011: subheader line	–
012...018:	Scaling factor for the different HTAP emissions species.
019: subheader line	–
020...035:	Switch to activate a specific source region mask. Note: One can use several source masks at the same time.
036: subheader line	–
037: *LRCPTR_MASK*	Switch for receptor mask.
038...094:	Switch to activate a specific receptor region mask. Note: One can use several receptor masks at the same time.
Note:	For a detailed information on the HTAP regions you can refer to the HTAP Meeting Presentation.

2.2.3 *define_adj.h*: observation selection

The following are observation options, all set in *define_adj.h*. The corresponding files are in *code/obs/*. Some are not yet fully implemented, as indicated by "**Placeholder**".

Note: No observation operators are needed for basic sensitivity runs. Basic sensitivity runs are those where the cost function is calculated directly within subroutine *CALC_ADJ_FORCE_FOR_SENS*.

CO

MOPITT_V3_CO_OBS	Assimilate MOPITT CO (column) v3 observations to perform an optimization problem; data in hdf-eos4 file format.
MOPITT_V4_CO_OBS	Assimilate MOPITT CO (column) v3 observations to perform an optimization problem; data in hdf-eos4 file format.
AIRS_CO_OBS	Assimilate CO (column) observations from AIRS v5, data in hdf-eos4 file format.
SCIA_BRE_CO_OBS	Assimilate CO (column) observations from SCIAMACHY (Bremen retrieval only), data in ASCII format.

Aerosols

TES_NH3_OBS	
SCIA_DAL_SO2_OBS	**Placeholder** for SO_2 from SCIA.
PM_ATTAINMENT	**Placeholder** for aerosol attainment sensitivities; no observations (pseudo or real) required.
IMPROVE_SO4_NIT_OBS	**Placeholder** for aerosol observations from IMPROVE data files.
CASTNET_NH4_OBS	**Placeholder** for aerosol observations from CASTNET data files.

Ozone

SOMO35_ATTAINMENT	**Placeholder** for ozone attainment sensitivities; no observations (pseudo or real) required.
TES_O3_OBS	TES O_3 data in netcdf format from JPL.

NO2

SCIA_KNMI_NO2_OBS	**Placeholder** for SCIAMACHY or GOME NO_2 data from KNMI hdf files.
SCIA_DAL_NO2_OBS	**Placeholder** for . . .

other options

PSEUDO_OBS	Use pseudo observations (generated by the model).
LOG_OPT	Use log-scaling factors.
LIDORT	Compile LIDORT code for radiative forcing caculations.
LBFGS	Calculate inverse Hessian matrix using L-BFGS method.

2.2.4   Other input files

All the files listed below, if needed, should be place in *run/../geos*/* directory.

- *run* script, see Sec. 2.4.

- Observational error file(s): for each dataset used. Currently the code expects it in the following format:
  ```
  RRE_YYYYMMairsGlobal.bpch
  RRE_seasonMay1mopittGlobal.bpch
  RRE_seasonMay1sciabrGlobal.bpch
  ```
  So for example for AIRS data, in May 2004, the file name is RRE_200405airsGlobal.bpch. AIRS data uses monthly errors, while MOPITT and SCIAMACHY (due to scarcity of data), use seasonal errors. Currently there is 1 file per month for AIRS and 1 file each for MOPITTT and SCIA for the whole year. Current setting spans May 1, 2004 to May 1, 2005 NOTE: There is an option to specify a fixed error across all times and gridboxes and not use the files. This is useful for saving model and data to compute Relative Residual Error (RRE) quantities for later use in an inversion.

- *iter.txt*: This input file contains the iteration number of the next execution of the model. It starts with 1 and the code updates it. This way the same executable can be used for multiple runs (different dates and different iterations).

- MOPITT files: *mopitt_v3_apriori.dat* (a priori profile)

- SCIAMACHY files: *ak_co_wfmdscia_V2.dat* (averaging kernels, a priori profile), *SCIA_pressure.dat* (pressure levels for SCIA Bremen)

2.3   Output files

See Sect. 2.1 for locations. All are binary bunch files unless otherwise noted. Note: files that don't have an iteration token in their name will be overwritten or removed at each iteration. Exceptions are *aero.ave** and *satave** files, which are lumped into *fwd_dat.*.tar* files after each iteration, see the run script.

2.3.1   Essential output files

adjtmp/gctm*.chk.*	Checkpoint files. Generated during the forward run; deleted after they are used in the backward run (although the `L_DEL_CHECKPT` flag in *CMN_ADJ* allows you to not delete them if desired). See *checkpt_mod.f* for more details on content.
OptData/gctm.gdt.NN	Gradients of active parameters at each iteration NN (`IJ-GDE-$`). These gradients are semi-normalized. To include fully-normalized gradients (`IJ-GDEN$'` for diagnostic purposes), set `L_WRITE_GDEN = .TRUE.` at the top of the routine `MAKE_GDT_FILE`.
OptData/cfn.NN	Cost function at each iteration NN. ASCII

2.3.2   Nonessential output files

OptData/gctm.ics.NN	Scaling factors at each iteration NN. The scaled emissions themselves at each iteration are also included in this file for diagnostic purposes. Hence, the a priori estimates of emissions will be found in *gctm.ics.01*, optimized emissions found in *gctm.ics.10*, etc. The scaling factors are `IJ-EMS-$` and the emissions themselves are `IJ-EMO-$`.
OptData/gctm.obs.NN	Satellite observations in the CO version. Files contain 3 data structures/arrays. Tracer 1 corresponds to MOPITT obs, 2 to SCIA, 3 is AIRS. The arrays are (IIPAR, JJPAR, NDAYS), where NDAYS is number of simulation days.
OptData/gctm.model.NN	Model columns corresponding to the satellite data in time, space and retrieval processing to observations in gctm.obs.NN. These arrays correspond to the satellite ones and are updated at each iteration of the optimization.
OptData/gctm.costf.NN	These files contain 2 arrays; 1: Cumulative cost function (summed over all observation types) with dimensions (IIPAR,JJPAR,NDAYS), 2: observation count in each gridbox with dimensions (IIPAR,JJPAR).
OptData/gctm.modbias.NN	Contains `(model-obs)/model` at (IIPAR,JJPAR,NDAYS) resolution. Useful for computing Relative Residual Errors (RRE).
OptData/gctm.forcing.NN	Contains `2*(model-obs)/err^2`, also known as adjoint forcing; a diagnostic.
OptData/gctm.gdta.NN	Gradients of all parameters at each iteration NN. Only generated if `CALL MAKE_GDT_ALL_FILE` in *inverse_driver.f*.
OptData/gctm.arr	Integrated reaction rate constant sensitivities.

diagadj/gctm.adj.YYYYMMDD.hhmmss	Adjoint state variables (λ_c). Purely diagnostic; only written if `LADJ_TRAJ = T` (in *input.gcadj*) and `ITS_TIME_FOR_OBS`. The number of these files kept can be controlled using the `REMOVE_ADJ_FILE` routine (defunct, need to fix!) in *geos_chem_adj_mod.f* and the `N_ADJ_KEEP` parameter in *input.gcadj*. Can be saved as sensitivities with respect to concentrations or concentration scaling factors, see `MAKE_ADJ_FILE` in *geos_chem_adj_mod.f*.
adjtmp/gctm.obs.YYYYMMDD.hhmmss	Pseudo observation file.
diagadj/gctm.fd.YYYYMMDD.hhmmss	Diagnostic file. Used for process specific finite difference tests.
diagadj/gctm.fdglob.YYYYMMDD.hhmmss	Diagnostic file. Used for process specific second order finite difference tests. Only generated if FD_GLOB is set to true.
diagadj/aero.ave.YYYYMMDD	Aerosol data and corresponding model predictions. Generated when running with `IMPROVE_OBS` or `PM_ATTAINMENT` options.
diagadj/jsave.YYYYMMDD	Contributions to cost function. Generated when running with `IMPROVE_OBS` or `ATTAINMENT` options.
diagadj/satave.bpch	Satellite data and corresponding model predictions for `NO2_SAT_OBS`.
diagadj/ems.adj.*	Emissions sensitivities on per-kg basis (as opposed to scaling factors). Currently only implemented for SO_2, NH_3, OC and BC emissions. Enabled with the *LEMS_ABS* switch.
diagadj/gctm.iteration	ASCII. Statistics from the optimization procedure. Enabled with the *LITR* switch.
runs/../geos*/FWD_met	Elements of the forward meteorology in the FD cell.
runs/../geos*/BACKWD_met	Elements of the backward meteorology in FD cell. Should agree exactly with contents of *FWD_met*.

29

2.4 The *run* script

Each run of the adjoint can be done with the same executable. Thus the simplest run script is sufficient for 1 iteration. The number in ITER updates *N_CALC_STOP* variable in *inverse_driver.f* and determines the number of iterations in the next execution of the code.

NOTE: The *run* script uses bash like syntax, we expect it to work if you have bash on your system. If you experience problems please write to the development team in other to adress the problem.

To use, you have to set several variables in the script, following the instructions therein.

Change often (almost every run):

```
X
XSTOP
RNAME
TYPE
```

Change rarely (only when migrating to a new filesystem):

```
IFORT_OPT
RECOMPILE
SAVE
DSAVE
ARCHIVE
DARCHIVE
DRUNDIR
DPACK
```

Unlikely to need to be changed from the defaults:

```
DCODE
DRUN
```

The frequently changed variables are `X`, `XSTOP`, `RNAME` and `TYPE`. Set `RNAME` to be a descriptive name of your current calculation (such as `ADJv23_optimize_NH3_emissions`). Set `TYPE` acording to the type of simulation operator, the model supports 4 types of simulation:

```
DEFAULT    : As it says is the default type that most of the people run.
HDF        : Set this type when satellite HDF data is going to be used.
LIDORT     : Set this type when running a LIDORT simulation.
SAT_NETCDF : Set this type when satellite NetCDF data is goin to be used.
```

Set the start (or current) iteration number, `X`. Set `XSTOP` to the final iteration number. For example, if you've already computed five iterations, then set `X=6` and `XSTOP=9` to compute three more. At each iteration, the current value of `X` is assigned to the variable `N_CALC_STOP` in *inverse_driver.f* and the code is executed. `X = 0` is used for generating pseudo observations. For example, if `X=0` and `XSTOP =4`:

Computational flow for each iteration

X=0	X=1	X=2	X=3	X=4
$\sigma=1$	$\sigma=1+\delta\sigma$	read *.01	read *.01	read *.01
DO_GEOS_CHEM	DO_GEOS_CHEM	update σ	update σ	update σ
make *obs*	DO_ADJOINT	DO_GEOS_CHEM	read *.02	read *.02
	write *.01	DO_ADJOINT	update σ	update σ
		write *.02	DO_GEOS_CHEM	read *.03
			DO_ADJOINT	update σ
			write *.03	DO_GEOS_CHEM
				DO_ADJOINT
				write *.04

A reason to do only a single iteration per execution is because of code fragments like this:

```
LOGICAL, SAVE   :: FIRST = .TRUE.
⋮

IF ( FIRST ) THEN
   CALL INIT_ARRAYS
   FIRST = .FALSE.
ENDIF
⋮
```

By exiting the code between each iteration, these logical control switches get automatically reset to `FIRST = .TRUE.`. Doesn't this repeated reading / writing / optimizing take a lot of time? No, not with respect to the expense of the forward and backward model calculations.

For sensitivity calculations, set `X` and `XSTOP` to 1

For global finite difference calculations, set `X=1` and `XSTOP=3`.

For spot finite difference calculations, set `X=1` and `XSTOP=2`.

For generating pseudo observations using *PSEUDO_OBS*, set `X=0` and `XSTOP` equal to the total number of iterations you wish to perform. Subsequent tests using the same set of pseudo observations can start from `X=1`.

The run script will compile the *geos* executable,

```
####################################################################
# Compile geos, move it to the run directory and execute
####################################################################

cd $DRUN/$DPACK/$DCODE
if  [ -n $(grep 'Compute BFGS inverse Hessian: T' $DRUNDIR/input.gcadj) ]; then
       ./objects.sh $TYPE
else
       ./objects.sh $TYPE LBFGS
 fi

 if   [ $TYPE = 'DEFAULT' ]; then
      IFORT_OPT="$IFORT_OPT "
 elif [ $TYPE = 'HDF' ]; then
       IFORT_OPT="$IFORT_OPT HDF=yes"
 elif [ $TYPE = 'SAT_NETCDF' ]; then
       IFORT_OPT="$IFORT_OPT SAT_NETCDF=yes"
 elif [ $TYPE = 'LIDORT' ]; then
       IFORT_OPT="$IFORT_OPT LIDORT=yes"
 fi

 if  [ -n $(grep 'Compute BFGS inverse Hessian: T' $DRUNDIR/input.gcadj) ]; then
     make $IFORT_OPT
else
     make LBFGS='yes' $IFORT_OPT
fi

if [ $RECOMPILE = 'YES' ]; then
    mv -f geos ../$DRUNDIR/
else
     cp -f geos ../$DRUNDIR/
fi

cd ../$DRUNDIR/
time ./geos
```

There are variables on the run script that aren't changed frequently, but require a special atention.

`IFORT_OPT` Define optional flags to the makefile (`DEBUG=yes TRACEBACK=yes IPO=yes`) `RECOMPILE` Define whether or not recompile the code on each iteration. By default is deactivated as it saves time when using the optimization option(`IPO=yes`).

If you are running an interactive job and wish to save a log file that doesn't get overwritten each iteration, than you may replace

```
time ./geos
```

with

```
time ./geos > log.${X}
```

The run script will optionally save a copy of the entire package and move it to `DSAVE`. This may be necessary if you are using `DRUN` on the local storage of a compute node to which you no longer have access after your log has completed. The enable or disable this feature, set the variable `SAVE`.

Lastly, users running parallel jobs with OpenMP on multi-core compute nodes should be mindful to set the following to be appropriate for their system:

```
#  Set number of threads
export OMP_NUM_THREADS=24
```

3 Running the adjoint code

3.1 Selecting the adjoint model operational mode

Selection of adjoint model operation mode is done via an input file *input.gcadj* in the run directory and a preprocessor file *define_adj.h* in the *code/adjoint/* directory.

Active variable selection There are two options for active variables (the ones with respect to which we compute sensitivities): initial conditions (LICS) and emissions (LADJ_EMS), which could include chemical production sources. The active variable tracers (LICS) need to be listed in *input.gcadj* and their order and numbering needs to correspond to the tracers listed in the *input.geos* file. For LADJ_EMS, the emissions also need to be listed in *input.gcadj*.

3.2 Forward model settings

Not all (forward) GEOS-Chem options are accommodated in the adjoint model. Currently, the forward model settings that are supported are as follows:

`resolution:`	4x5 and 2x2.5, nested CO capabilities
`simulation types:`	full chemistry, tagged CO, tagged O3 and offline CO_2.
`meteorology:`	GEOS3, GEOS4, GEOS5
`chemistry:`	via KPP Rosenbrock solver

The forward model needs to be set to one of the supported options, unless we set LADJ to FALSE (in *input.gcadj*), which will run the code in forward model only mode. Check the GEOS-Chem adjoint wiki for up-to-date list of current supported features of the forward model.

3.3 Adjoint code checklist

Generally and ideally all settings should be controlled via the input files (*input.gcadj* and *define_adj.h*). However, a few hardwired options remain and are listed below. Each needs to be checked before a new simulation.

- Subroutine *INIT_WEIGHT* in *adj_arrays_mod.f*:

This subroutine is used to specify the spatial domain over which observations are included in the cost function for the PSEUDO_OBS 4D-Var tests as well as for the following sensitivity run cost function options: LKGBOX, LUGM3, LSTT_PPB, and LSTT_TROP_PPM. If

34

you're computing a sensitivity of specific gridbox or region, you need to make sure that the WEIGHT array is initialized accordingly.

- Subroutines *CALC_ADJ_FORCE_FOR_SENS*

If you are using a cost function option that depends upon species concentrations (i.e., CSPEC), you can adjust the LMIN, LMAX, etc, parameters at the top of this routine to limit the spatial coverage of the cost function.

- *Makefile*

Makefile settings have to correspond to settings in *define_adj.h* file for observational operators, i.e. if we're using hdf or netcdf files, we need to include appropriate libraries. Alternatively, we need to make sure that the files requiring those libraries are not being compiled when they are not needed. For a list of all options supported type `'make help'` inside the `code/` directory.

- subroutines *SET_SF* or *SET_LOG_SF* in *inverse_mod.f.*

The SF_DEFAULT values in *input.gcadj* can be used to set global values for the scaling factors during iteration X=1. Any non-global settings of initial guesses require changing the code in these subroutines.

3.4   Finite difference test checklist

The finite difference test is either done 1 gridbox at a time (*FD_SPOT* is TRUE) or globally at once (*FD_GLOB* is TRUE). Finite difference test compares an adjoint gradient to its finite difference approximation. The finite difference perturbation, as well as the gridbox for the spot test can be set in *input.gcadj*. Current preferred method of testing adjoint code is to perform a global FD test.

Checklist:

- Comment out all observation operators (including pseudo observations) in *define_adj.h*.
- Set *LSENS* to TRUE, and both *L4DVAR* and *L3DVAR* to FALSE.
- Set *FD_GLOB* or *FD_SPOT* to TRUE (not both).
- Set *LADJ_EMS* or *LICS* to TRUE.
- Set finite perturbation, FD_DIFF (e.g. to something like 0.1).
- Select a numerator and denominator for the test in the FD MENU.
- Decide whether to specify finite difference box with index (Specify box is TRUE) or lon/lat values.
- Run from iteration 1 to iteration 3 for a global test . The first iteration performs the base case forward model run and the adjoint run. For iterations 2 and 3, the forward

model will be run for positive and negative perturbations of the control variables being tested.

For a spot test, run from iteration 1 to 2. Both iterations include a forward model run and adjoint model run. The second run is evaluated with a positive perturbation. Finite difference sensitivities are compared to the average of the adjoints from the first and second run.

- A few things to note about FD tests:
  - First guesses settings *SF_DEFAULT* do not apply here.
  - The default cost function setting for *FD_GLOB* is *LKGBOX*. It applies to the entire layer *LFD*.
  - *FD_SPOT* can use any of the STT-based cost function unit options. It applies to grid cell *IFD,JFD,LFD*.
  - If (*FD_GLOB* is TRUE), transport will automatically be turned off, overwriting settings in *input.geos*).
  - Currently *FD_GLOB* is only setup to work with tracers, not yet species.
  - For FD_SPOT, species-based cost functions can be implemented by using the CSPEC_PPB option and filling in the CSPEC OBS MENU. NFD will be ignored.
  - Setting FD_GLOB will trigger LMAX_OBS = T and NSPAN = 1.
  - Setting NFD will override any tracer that is listed in the OBSERVATION MENU.

Relevant diagnostic output:

- *diagadj/gctm.fd.YYYYMMDD.hhmmss* Diagnostic file. Used for process specific finite difference tests. It contains 6 data blocks. The first two are gradients, the third one is the ratio of the gradients.
- *diagadj/gctm.fdglob.YYYYMMDD.hhmmss* Diagnostic file. Used for process specific second order finite difference tests. Only generated if *FD_GLOB* is set to true. It contains 6 data blocks. The first two are gradients, the third one is the ratio of the gradients

3.5 Sensitivity (non-finite difference checklist)

The sensitivity run local and global sensitivity calculation. *LSENS* controls this simulation type. The options here are sensitivity of a predefined gridbox concentration, average concentration etc. (in *CALC_ADJ_FORCING* in *geos_chem_adj_mod.f*).

Note that if *LSENS* is TRUE, both *L4DVAR* and *L3DVAR* have to be FALSE.

- Comment out all observation operators (including pseudo observations) in *define_adj.h*.
- Set *LSENS* to TRUE, and both *L4DVAR* and *L3DVAR* to FALSE.
- Set *FD_GLOB* and *FD_SPOT* to FALSE.

- Set *LADJ_EMS* or *LICS* to TRUE.
- Specify as control variables the tracers (for *LICS*) or emissions *LADJ_EMS* for which you would like to calculate sensitivities. For *LADJ_EMS*, set opt=T, otherwise the gradients get set to zero.
- Specify observations / cost function options.
  - Set *OBS_FREQ* to desired frequency of numerator calculation. Note that even if you want a monthly mean concentration, you still have to set *OBS_FREQ* to something like 60 (min).
  - Use *LMAX_OBS* and NSPAN to evaluate your cost function over a limited time range.
  - Select an option for the cost function evaluation, and recognize whether it depends upon tracers (STT) or species (CSPEC)
  - List the tracers or species to be included in the cost function in *input.gcadj*.
  - Modify the spatial domain of the cost function using *WEIGHT* for tracers or *LMIN, LMAX* etc. for species.
- Run for iteration 1 only. This iteration performs the base case forward model run and the adjoint run.

3.6   4D-var checklist

4D-var run is an inverse model run, which takes pseudo or real observations to estimate initial conditions, emissions, chemical sources and/or reaction rates. Observations must be set in *define_adj.h*. All other flags are in *input.gcadj*. This simulation might require special data libraries (e.g. ncdf, hdf), data, and additional input files (see section 3.3). Supported active/control variables are LICS, LADJ_EMS (someday, both).

For inverse model tests using pseudo observations, we usually set the "initial guess" of scaling factors to be some value other than one (or zero for `LOG_OPT`) and then try to converge to values of one (zero). For real data assimilation problems or attainment studies, we begin with our best estimate, i.e. scaling factors equal to one (zero), and converge to values that lead to best agreement with observations.

Checklist:

- Select desired observations in *define_adj.h*
- Decide whether to optimize scaling factors (default) or their log, also in *define_adj.h*.
- Depending on whether we are using observations requiring additional libraries (e.g, hdf or netcdf), modify the run script to use the appropriate Makefile.
- Set a desired number of iterations in the run script if using a multiple iteration script.
- Set L4DVAR to TRUE., check that LSENS and L3DVAR are FALSE
- Select control parameters LADJ_EMS or LICS.

37

- Set APSRC to TRUE if including a priori cost function term. If TRUE, make sure that *REG_PARAM* and *ERROR* are intentionally defined in the CONTROL VARIABLE MENU.
- Set initial conditions for ICS_SF and EMS_SF. This can be done globally using *SF_DEFAULT*, or on a regionally specific basis by modifying code in *inverse_mod.f*.
- Set observation frequency.
- List optimized species and/or emissions.
- If your observation operator depends upon a chemical species in CSPEC, such as O_3 or NO_2, list this species as an observed species in the OBSERVATION MENU.
- Specify for which gridbox or lat/lon you want debugging print statements in the finite difference menu.
- Select diagnostic output (in the diagnostics menu) such as LITR.

3.7 3D-var checklist

3D-var run is a Kalman filter forward estimation. The code exists outside the v8 framework. Contact Kumaresh Singh if interested in details.

4 Coding, debugging and testing

4.1 Sensitivity with respect to reaction rate coefficients: adding new reactions

Currently, sensitivities with respect to the 297 reaction rate constants are available. The reaction rates are defined in the RXN_ID file inside the run directory.

To have the model calculate sensitivity with respect less than 297 rate constants, the following needs to be updated:

- Update the total number of active reaction rate constants, `NRRATES` in *input.gcadj* and make it match `NCOEFF_RATE` in *gckpp_adj_Util.f90*

- Set `FI_RXNID` to False in *input.gcadj*

- List the subset of reactions in the apropriate section of *input.gcadj*

- For the finite difference test `RATFD` should be equal to the `Rate#` instead of `ID#` (check example)

For example, if we would like to calculate the sensitivity of the cost function with respect to the rate constant for the reaction $2OH \rightarrow O_3 + H_2O$, the should look like this:

```
>------------------------------<
         FOR LADJ_RRATE         :
NRRATES: num of rxn rates       : 1
Read reactions from RXN_ID file: F
 ...or use these Rxn rates      : ID#   rxn_name   opt SF_DEFAULT REG_PARAM ERROR
Rate #1                         : 6     2OH->O3+H2O T  1          1        1
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
%%% FINITE DIFFERENCE MENU %%%
...
Denomenator of deriv.   to test:
...
 => w/LRRATE: rate      RATFD   : 1
--------------------------------------------------------------------------------
```

If more reaction rates are needed the process is similar, just add the entries to *input.gcadj* and remember to update `NRRATES`.

4.2   Troubleshooting and debugging

If you find any incompatible set of switches in *input.gcadj* that crash the code, please add corresponding error checks in `ARE_FLAGS_VALID` in *input_adj_mod.f*.

5   Validating code

[This section needs to be updated]

The adjoint of GEOS-Chem is designed to be validated in several manners.

5.1   Global tests of a subset of the adjoint model

By turning off multidirectional transport related processes, the adjoint model sensitivities can be compared to finite difference sensitivities on a global scale, see Fig. 1 and 3 of *Henze et al.* (2007).

For example, to check the adjoint of the chemistry only in a single vertical level, set the following:

- in *run*:

- X=1
- XSTOP=3
- TYPE=DEFAULT

- in *input.gcadj*:

  - LSENS = T
  - FD_GLOB = T
  - LICS or LADJ_EMS or LADJ_EMS and LADJ_STRAT = T
  - select a dependent species, NFD
  - select a vertical level, LFD
  - select a control parameter, ICSFD or EMSFD and/or STRFD
  - IFD and JFD don't matter as we're doing a domain wide test
  - set the finite difference perturbation ($\delta\sigma$), FD_DIFF = 1.d-1
  - Set the OBS_FREQ to 60, but set LMAX_OBS = T and NSPAN = 1.
  - Set LAERO_THERM = .FALSE.

- in *input.geos*:

  - Turn off convection (LCONV = F)
  - Turn off turbulent mixing (LTURB = F)
  - Turn off wet deposition (LWETD = F)
  - Leave on dry deposition (LDRYD = T)
  - Can leave on transport (LTRAN = T), which will be overridden if FD_GLOB = T.

- in *chemistry_mod.f*, can save time by computing gas-phase chemistry only in certain cells by placing an IF statement around CALL INTEGRATE_ADJ such as:

  ```
  IF ( L == LFD ) THEN

  CALL INTEGRATE_ADJ(...

  ENDIF
  ```

Analysis of global FD tests is described in Sect. 1.5.1.

5.2   Spot tests of full adjoint model

Alternatively, we can compare adjoint gradients to finite difference gradients for control parameters one location at a time, but with all model processes turned on.

- in *run*:

- – X=1

- – XSTOP=2

- – TYPE=DEFAULT

- in *input.gcadj*:

  - – LSENS

  - – Make sure that `FD_GLOB` is set to FALSE and that `FD_SPOT` is set to TRUE.

  - – active variables are `LICS` or `LADJ_EMS` or `LADJ_EMS` and `LADJ_STRAT`

  - – select a dependent species, `NFD`

  - – select particular control parameter `IFD JFD LFD` and `ICSFD` or `EMSFD` and/or `STRFD`

  - – set the finite difference perturbation ($\delta\sigma$), `FD_DIFF = 1.d-1`

  - – probably want to turn on additional diagnostic output, so set `L_PRINTFD = .TRUE.`

  - – Make sure that `OBS_FREQ` is set long enough so that the cost function is only evaluated once during the simulation.

- in *input.geos*:

  - – Turn on all desired processes.

The adjoint and finite difference gradients and the ratio `ADJ / FD` will be written to standard output at the end of the run. The reported adjoint gradient is actually the average of the values at $\sigma = 1$ and $\sigma = 1 + \delta\sigma$. Since such tests involve the continuous adjoint of advection, the ratio `ADJ / FD` can not be expected to be unity. Benchmark tests are given in (*Henze et al.*, 2007).

6   Generating forward and reverse code for GEOS-Chem with the Kinetic PreProcessor (KPP)

(*Damian et al.*, 2002; *Sandu et al.*, 2003; *Daescu et al.*, 2003)

6.1 KPP input files

6.2 Post processing KPP generated code

6.2.1 Interfacing KPP code with GEOS-Chem

6.2.2 Implementing OpenMP Parallelization in KPP generated code

6.3 Performing global benchmarks of new chemical solvers

References

Daescu, D. N., A. Sandu, and G. R. Carmichael (2003), Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: II - numerical validation and applications, *Atmos. Environ.*, *37*(36), 5097–5114.

Damian, V., A. Sandu, M. Damian, F. Potra, and G. R. Carmichael (2002), The kinetic preprocessor KPP - a software environment for solving chemical kinetics, *Comput. Chem. Eng.*, *26*(11), 1567–1579.

Henze, D. K., A. Hakami, and J. H. Seinfeld (2007), Development of the adjoint of GEOS-Chem, *Atmos. Chem. Phys.*, *7*, 2413–2433.

Henze, D. K., J. H. Seinfeld, and D. Shindell (2009), Inverse modeling and mapping U.S. air quality influences of inorganic $PM_{2.5}$ precursor emissions using the adjoint of geos-chem, *Atmos. Chem. Phys.*, *9*, 5877–5903.

Kopacz, M., D. Jacob, D. K. Henze, C. L. Heald, D. G. Streets, and Q. Zhang (2009a), A comparison of analytical and adjoint Bayesian inversion methods for constraining Asian sources of CO using satellite (MOPITT) measurements of CO columns, *J. Geophys. Res.-Atmos.*, *114*, D04305, doi:0.1029/2007JD009264.

Kopacz, M., D. J. Jacob, J. A. Fisher, J. A. Logan, L. Zhang, I. A. Megretskaia, B. M. Yantosca, K. Singh, D. K. Henze, J. P. Burrows, M. Buchwitz, I. Khlystova, W. W. McMillan, J. C. Gille, D. P. Edwards, A. Eldering, V. Thouret, and P. Nedelec (2009b), Global estimates of CO sources with high resolution by adjoint inversion of multiple satellite datasets (MOPITT, AIRS, SCIAMACHY, TES), *Atmos. Chem. Phys. Discuss.*, *submitted*.

Sandu, A., D. N. Daescu, and G. R. Carmichael (2003), Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: Part I - theory and software tools, *Atmos. Environ.*, *37*(36), 5083–5096.

Singh, K., P. Eller, A. Sandu, D. K. Henze, K. Bowman, M. Kopacz, and M. Lee (2009), Towards the construction of a standard geos-chem adjoint model, *ACM High Performance Computing Conference*.

Zhang, L., D. J. Jacob, M. Kopacz, D. K. Henze, K. Singh, and D. A. Jaffe (2009), Intercontinental source attribution of ozone pollution at western US sites using an adjoint method, *Geophys. Res. Lett.*, *36*, L11810, doi:10.1029/2009gl037950.