DART-GITM Interface Manual

Alexey Morozov*

April 12 2013

Contents

1 The conventions used in this document

We mostly stick to DART's tutorial conventions:

- external clickable links look like this – (blue, no underline)
- `all filenames look like this -- (typewriter font, red)`
- *program names look like this – (italicized font, green)*
- `user input looks like this -- (typewriter font, magenta)`
- `command line commands look like this -- (light green box)`
- And the contents of a file are (sometimes) colorcoded and enclosed in a box:

```
1 subroutine write_restart(dir)
2   use ModGITM
3   use ModInputs !alexey uses useDART and f107 variable from ModInputs as
        estimate updated by DART, f107a is assumed = f107
```

Code 1: Sample code from `GITM2/src/restart.f90`

*alexeymor at gmail

2 Downloading and building DART

1. Install all the prerequisites for DART as described here. I recommend using University of Michigan NYX cluster or your local cluster as it is likely to have all the required software installed. Installing NetCDF on Mac is a pain, so I would strongly advise you against it, but if you have to, see Tim Hoar's site.

2. Download DART by using (I put it in home directory and all my pbs_files use that fact)

   ```
   mkdir ~/HEAD
   ```

   ```
   cd ~/HEAD
   ```

   ```
   svn checkout https://proxy.subversion.ucar.edu/DAReS/DART/branches/gitm DART
   ```

   Particularly note that the last line is different than what you get when you register at the DART's website (please register!) because currently GITM is not part of main DART branch (link to the main branch is given at the registration).

3. Download a GITM patchfile via

   ```
   wget http://www-personal.umich.edu/~morozova/DART/GITMpatch1
   ```

4. Go into the DART directory

   ```
   cd DART
   ```

5. Apply the patch via

   ```
   patch -p0 < ../GITMpatch1
   ```

   (note that the correct way to undo the patch is via `patch -p0 -R < ../GITMpatch1`, which interestingly enough will not delete the files it created, but will just erase their contents. Also note that applying the patch multiple times will add the lines to the files multiple times, so if in doubt, start over.)

6. Customize a couple of files (here is what we use at the University of Michigan on NYX cluster:

   (a) on NYX create a file `~/privatemodules/default` containing the following lines (do `emacs ~/privatemodules/default` if you don't know how to do this and google 'emacs tutorial'). this file is used to load the correct compilers and libraries at login, so you might need to re-login

   ```
   module load intel-comp/12.1
   module load openmpi/1.6.0/intel/12.1
   module load netcdf/4.2.0
   module load python
   module load matplotlib
   ```

   (b) modify part of `DART/mkmf/mkmf.template` as shown next. (this is already complete if you applied the patch above. The purple entries is what is different from the original mkmf.template and specific to UMichigan's setup - more detail on customizing it)

```
MPIFC = mpif90
MPILD = mpif90
FC =  ifort
LD =  ifort
NETCDF =  /home/software/rhel5/netcdf/4.0-intel
INCS =  -I$(NETCDF)/include
LIBS = -L$(NETCDF)/lib -lnetcdf
FFLAGS  =  -O2 -ftz -vec-report0 $(INCS)
LDFLAGS =  $(FFLAGS) $(LIBS)

# for development or debugging, use this instead:
# FFLAGS = -C -check noarg_temp_created -ftz -vec-report0 -fp-model precise -O0 -g
#        -warn argument_checking,declarations,uncalled,uninitialized,unused $(INCS)
# FFLAGS = -ftz -vec-report0 -O3 -g $(INCS)
# Some optimized (BLAS, LAPACK) libraries may be available with:
# LIBS = -L$(NETCDF)/lib -lnetcdf -lmkl -lmkl_lapack -lguide -lpthread
```

7. Build and run the Lorenz 63 model (see here for detailed troubleshooting or DART/README for plain-text version of the quickstart) - this a good intro to building DART models.

8. Now we can move on to building GITM:

   ```
   cd ~/HEAD/DART/models/gitm/GITM2
   ```

   ```
   chmod -R 775 ../../
   ```

   ```
   svn rm --force src/ModSize.f90; svn rm --force src/ModKind.f90
   ```

   ```
   ./Config.pl -install -compiler=ifortmpif90 -earth
   ```

   (which assumes you are using intel fortran compiler along with openMPI. For gfortran-openMPI combination use -compiler=gfortran, otherwise read Config.pl)

   ```
   make
   ```

9. If 'make' didn't return any errors, you can move on to building the DART side of the interface:

   ```
   cd ../work
   ```

   ```
   ./quickbuild.csh
   ```

10. If that also compiled without errors, you should be ready to run. Change the ACCOUNT, QUE, USERNAME and EXAMPLE in work/pbs_file.sh:lines 5-8, 14 to reflect the account and que you want to use (lines 5-7, for guidelines, see this), email where you want to receive job start and finish notifications (lines 5-8) and your login name on NYX (line 14).

11. Save the changes and exit back to the shell and you are almostready to submit your first job. So the last thing you need to do is get a binary file (the patchfile you downloaded doesn't contain any binary files for now):

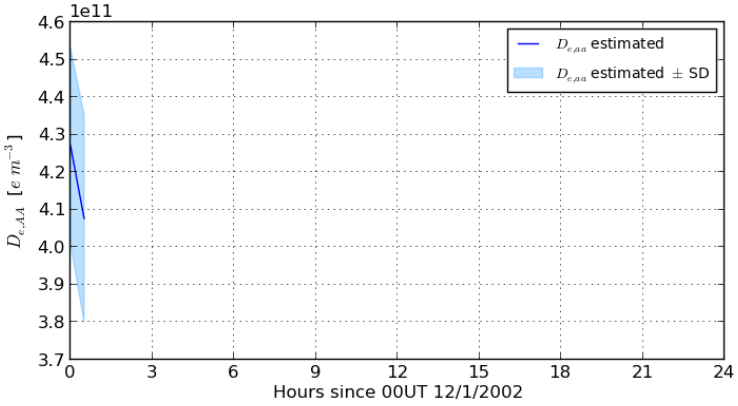


Figure 1: what /nobackup/<YOUR USERNAME>/dart_test_tec/gitm/python/daa00.png should look like. Nothing exciting, but you see that electron density above Ann Arbor, MI changed quite a bit during that half hour.

```
wget http://www-personal.umich.edu/~morozova/DART/hwm071308e.dat
```

```
mv hwm071308e.dat ../GITM2/srcData
```

and now you are ready to submit your first job:

```
qsub pbs_file.sh
```

12. Job should finish without errors in about 13 minutes on 21 processors (it simulates only 30 virtual minutes) and should create a folder /nobackup/<YOUR USERNAME>/dart_test_tec/gitm/work/advance_temp_e1 along with a bunch of other folders. If it takes the full requested 60 minutes, likely something went wrong (see the troubleshooting sections below).

13. If it finished without errors, you can even go as far as to creating a couple cool plots. For that change the USERNAME and EXAMPLE in ../python/pbs_py.sh:line5-8,13 to reflect the que you have access to, email where you want to receive job start and finish notifications and your account name on NYX (line 13).

    ```
    qsub pbs_py.sh
    ```

    which will hopefully will produce /nobackup/<YOUR USERNAME>/dart_test_tec/gitm/python/daa00.png which you can `scp` to your home computer and compare to the one shown in Figure 1.

3 More detailed big picture of interface operation

4 DART source code modifications

Accordingly, using

```
a=0.9965;
b=0.0012;
c=-0.0067;
G=tf(b,[1 -a],1);
h=impulse(G); %one way
[A,B,C,D]=ssdata(G);
H(1)=D; for i=2:10; H(i)=C*A^(i-2)*B; end %another way
```

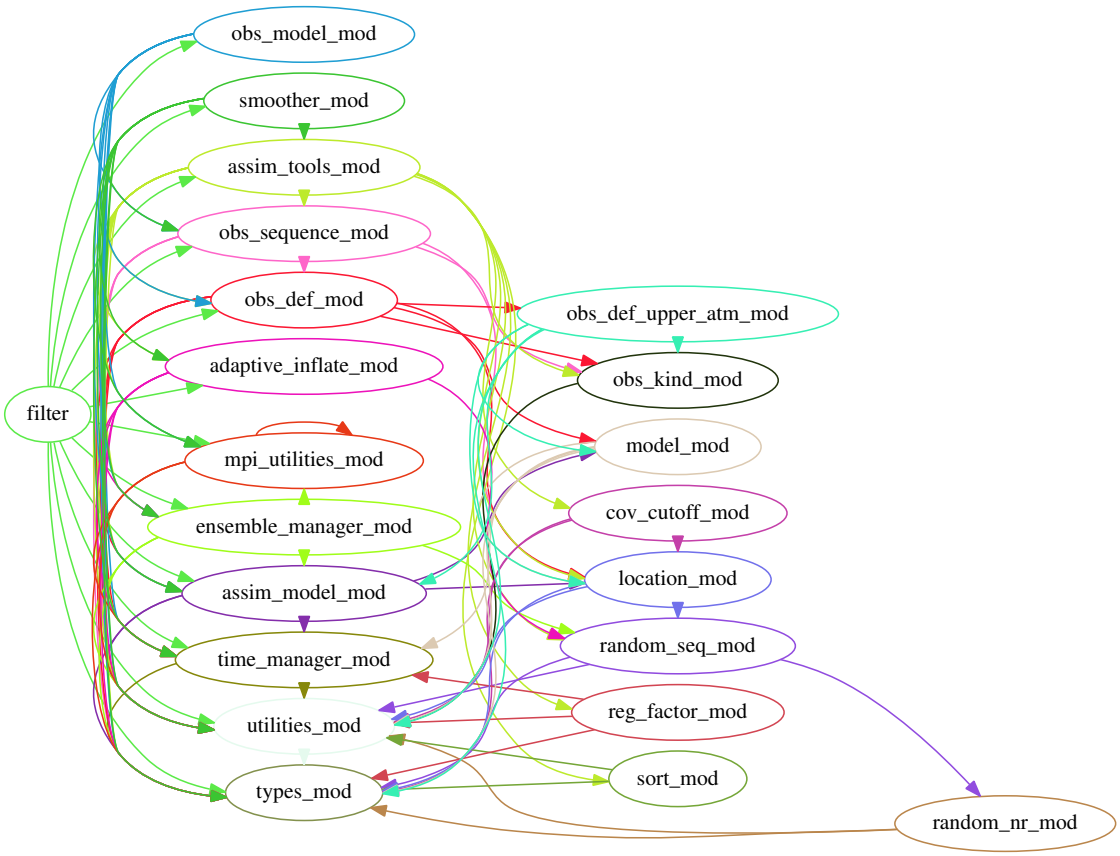
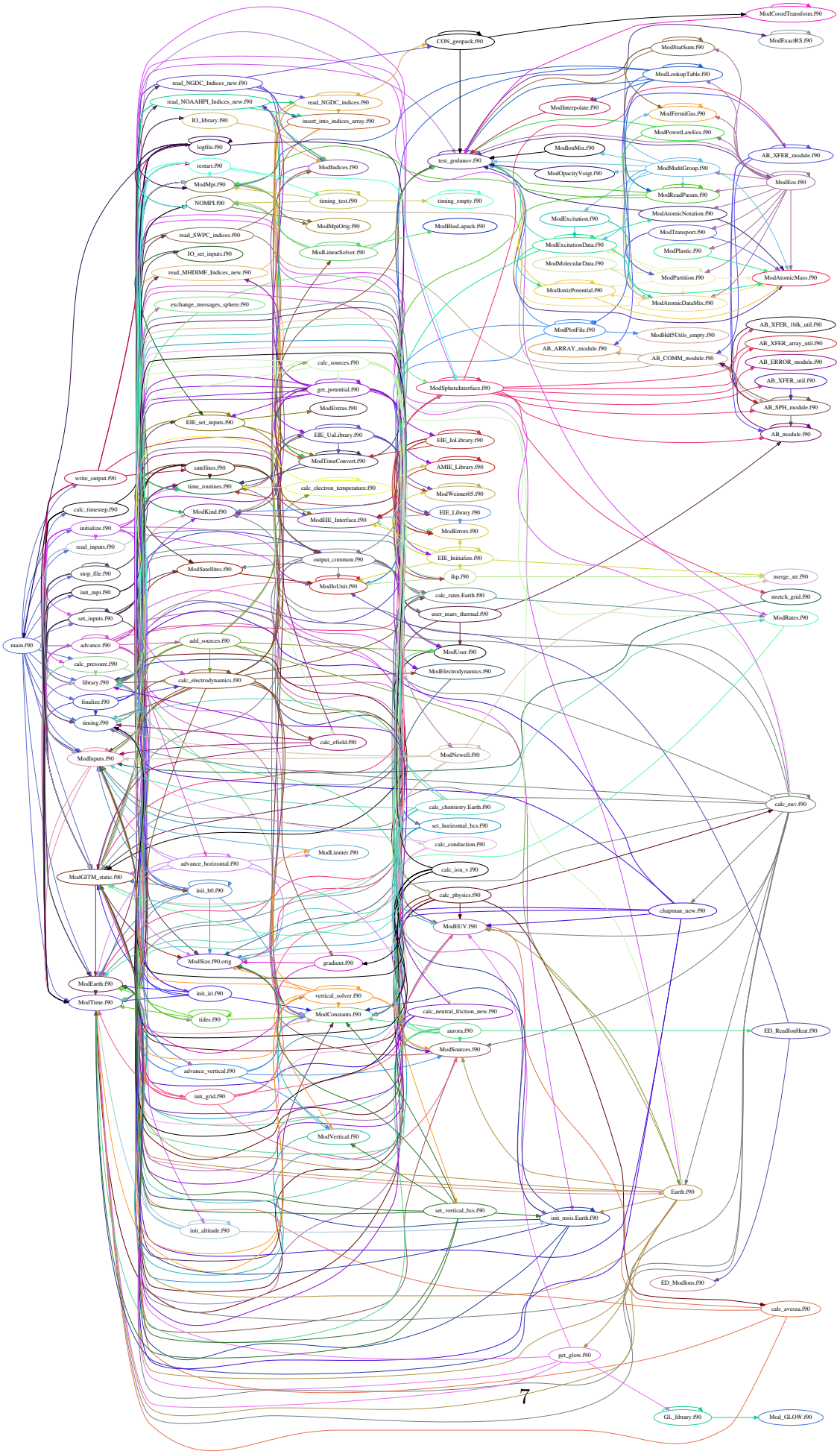

obs_model_mod
smoother_mod
assim_tools_mod
obs_sequence_mod
obs_def_mod
adaptive_inflate_mod
filter
mpi_utilities_mod
ensemble_manager_mod
assim_model_mod
time_manager_mod
utilities_mod
types_mod
obs_def_upper_atm_mod
obs_kind_mod
model_mod
cov_cutoff_mod
location_mod
random_seq_mod
reg_factor_mod
sort_mod
random_nr_mod

Figure 2: hi

would result in `h(2) = 0.0012;` and `H(2) = 0.0012`, which is also equal to `b`, as we would expect. Part of me wants to say just ignore the bias altogether as it is not strictly linear, so I would say use `H(2) = 0.0012` as `T` in RCAC and see if it works.

5 GITM source code modifications


ModCoordTransform.f90
CON_geopack.f90
ModExactRS.f90
ModStatSum.f90
ModLookupTable.f90
read_NGDC_Indices_new.f90
read_NOAAHPI_Indices_new.f90
read_NGDC_indices.f90
insert_into_indices_array.f90
IO_library.f90
ModInterpolate.f90
ModFermiGas.f90
ModPowerLawEos.f90
logfile.f90
ModIonMix.f90
AB_XFER_module.f90
restart.f90
ModIndices.f90
test_godunov.f90
ModOpacityVoigt.f90
ModMultiGroup.f90
ModEos.f90
ModMpi.f90
timing_test.f90
timing_empty.f90
ModReadParam.f90
NOMPI.f90
ModMpiOrig.f90
ModBlasLapack.f90
ModExcitation.f90
ModAtomicNotation.f90
read_SWPC_indices.f90
ModLinearSolver.f90
ModExcitationData.f90
ModTransport.f90
IO_set_inputs.f90
ModMolecularData.f90
ModPlastic.f90
read_MHDIMF_Indices_new.f90
ModPartition.f90
ModAtomicMass.f90
exchange_messages_sphere.f90
ModIonizPotential.f90
ModAtomicDataMix.f90
ModPlotFile.f90
ModHdf5Utils_empty.f90
AB_XFER_1blk_util.f90
AB_ARRAY_module.f90
AB_COMM_module.f90
AB_XFER_array_util.f90
AB_ERROR_module.f90
calc_sources.f90
AB_XFER_util.f90
get_potential.f90
ModSphereInterface.f90
ModExtras.f90
AB_SPH_module.f90
EIE_set_inputs.f90
EIE_UaLibrary.f90
EIE_IoLibrary.f90
AB_module.f90
ModTimeConvert.f90
write_output.f90
satellites.f90
AMIE_Library.f90
time_routines.f90
calc_electron_temperature.f90
ModWeimer05.f90
calc_timestep.f90
ModKind.f90
ModEIE_Interface.f90
EIE_Library.f90
initialize.f90
ModErrors.f90
read_inputs.f90
output_common.f90
EIE_Initialize.f90
stop_file.f90
ModSatellites.f90
ihp.f90
merge_str.f90
ModIoUnit.f90
init_mpi.f90
calc_rates.Earth.f90
stretch_grid.f90
user_mars_thermal.f90
set_inputs.f90
ModRates.f90
main.f90
advance.f90
add_sources.f90
ModUser.f90
calc_pressure.f90
ModElectrodynamics.f90
library.f90
calc_electrodynamics.f90
finalize.f90
timing.f90
calc_efield.f90
ModNewell.f90
ModInputs.f90
calc_euv.f90
calc_chemistry.Earth.f90
set_horizontal_bcs.f90
calc_conduction.f90
ModLimiter.f90
advance_horizontal.f90
calc_ion_v.f90
ModGITM_static.f90
init_h0.f90
calc_physics.f90
chapman_new.f90
ModEUV.f90
ModSize.f90.orig
gradient.f90
ModEarth.f90
init_iri.f90
vertical_solver.f90
ModTime.f90
ModConstants.f90
calc_neutral_friction_new.f90
tides.f90
aurora.f90
ED_ReadIonHeat.f90
ModSources.f90
advance_vertical.f90
init_grid.f90
ModVertical.f90
Earth.f90
set_vertical_bcs.f90
init_msis.Earth.f90
init_altitude.f90
ED_ModIons.f90
calc_avesza.f90
get_glow.f90
GL_library.f90
Mod_GLOW.f90