CLM5-DART Tutorial: Setting up and running a global assimilation

Brett Raczka, NCAR, Data Assimilation Research Section (DAReS)
Information about DART

Website:  https://dart.ucar.edu
Documentation:  https://docs.dart.ucar.edu

General questions to DART software team:  dart@ucar.edu
Questions related to Land DA and CLM-DART:  bmraczka@ucar.edu

This research was supported by the NASA CMS Program (awards NNX16AP33G and 80NSSC20K0010). CESM is sponsored by the National Science Foundation and the U.S. Department of Energy. We would like to thank the Center for High Performance Computing at the University of Utah. We would also like to acknowledge high-performance computing support from Cheyenne (doi:10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation, through allocation awards UUSL0005 and UUSL0007.
Information about DART

Website: https://dart.ucar.edu

Featured project: NC State, UC San Diego, MIT & KAUST Collaboration

UNDERSTANDING GULF OF MEXICO EDDY DYNAMICS

DATA ASSIMILATION FOR THE ENTIRE EARTH SYSTEM
Use ensemble DA techniques with geophysical models spanning the earth system.

USE DATA FROM ANY SOURCE, TEST MANY ALGORITHMS
Assimilate any suitable observations. Swap out filter and inflation algorithms with ease.

LEARN ON LAPTOPS, RUN ON SUPERCOMPUTERS
Compile without MPI for conceptual models or with MPI for GCMs on supercomputers.
Welcome to the Data Assimilation Research Testbed

The Data Assimilation Research Testbed (DART) is an open-source, freely available community facility for ensemble data assimilation (DA). DART is developed and maintained by the Data Assimilation Research Section (DARES) at the National Center for Atmospheric Research (NCAR).

Ensemble Data Assimilation

Ensemble DA is a technique for combining observations with numerical models to estimate the state of a physical system.

It enables modelers, observational scientists, and geophysicists to:
Information about DART: CLM5-DART

Updated CLM-DART documentation located on tag: ‘clm-swe_pre-release’
Updated CLM-DART documentation located on tag: ‘clm-swe_pre-release’
Information about DART: Tutorials

Prepared tutorials related to DART:  https://dart.ucar.edu/tutorials/

- Jeff Anderson presented

**DART LAB**
An introduction to Data Assimilation using MATLAB
DART_LAB is a MATLAB®-based tutorial to demonstrate the principles of ensemble data assimilation. The DART_LAB tutorial begins at a more introductory level than the materials in the tutorial directory, and includes hands-on exercises. ...  

**The DART tutorial**
The DART Tutorial is intended to aid in the understanding of ensemble data assimilation theory and consists of step-by-step concepts and companion exercises with DART. ...

**CLM-DART Tutorial**
Coming Soon!

**WRF-DART tutorial**
Overview
The WRF-DART tutorial steps through a WRF-DART experiment. The experiment covers the continental United States and uses a 50 member ensemble initialized from NCEP's Global Forecast System (GFS) initial conditions at 2017/04/27 00:00 UTC. ...

Materials of this presentation will go into it.
More information about my work:

Questions related to Land DA and CLM-DART: bmrazcka@ucar.edu

https://www.cgd.ucar.edu/events/seminars/

CESM Workshop 2021: Land Model Working Group

CESM Workshop 2021: Biogeochemistry Working Group
Example of DART workflow

CAM4 DART Reanalysis (80 member ensemble)

Ds199.1 | DOI: 10.5065/38ED-RZ08

Observations

Aboveground Biomass (gC m⁻²)
Liu et al., (2013)

Leaf Area (m² m⁻²)
Zhu et al., (2013)

CLM 5 terrestrial biosphere model

Initial Prior State

Fluxes: NEP + Δ
GPP + Δ
ER + Δ
ET + Δ

Biomass + Δ

Leaf Area + Δ

New Posterior State

Fluxes: NEP + Δ
GPP + Δ
ER + Δ
ET + Δ

Biomass + Δ

Leaf Area + Δ

Soil carbon

Soil water

Soil carbon + Δ

Soil water + Δ

Anderson et al., 2009

Improving CLM5.0 Biomass and Carbon Exchange Across the Western United States Using a Data Assimilation System

Brett Racza, Timothy J. Hoar, Henrique F. Duarte, Andrew M. Fox, Jeffrey L. Anderson, David R. Bowling, John C. Lin,

First published: 19 June 2021 | https://doi.org/10.1029/2020MS002421
CLM is a rapidly-moving target and DART is developed and maintained by a small group of people. Consequently, we have focused on supporting released versions of CLM. This documentation and scripting were tested using the CESM tag release-cesm2.2.0 and CLM tag release-cesm2.2.01 following the download instructions from https://github.com/ESCOMP/CESM.

It is recommended to clone a separate installation of cesm2.2 and specifically use it for CLM-DART simulations.

>> cd <your Cheyenne work directory>
>> git clone https://github.com/escomp/cesm.git cesm_dart

>> cd cesm_dart
>> git tag

>> git checkout release-cesm2.2.01
>> git status
>> ./manageExternals/checkout_external

Optional: If you want to use git to keep track of your personal CLM changes you may want to checkout out a branch to add/commit/track changes:
>> git checkout -b cesm_dart_branch
2) Download & Navigate DART

Creating DART directory within your ‘work’ directory

Downloading DART

DART is available through GitHub. To download the latest version of DART:

```bash
git clone https://github.com/NCAR/DART.git
```

To register for DART and view the terms of use, click on register for DART.

Citing DART

To cite DART, please use the following text:


and update the DART version and year as appropriate.
2) Download & Navigate DART

>> cd <your Cheyenne work directory>/DART/
>> git tag   (what tags are available)

>> git checkout clm-swe_pre-release
>> git status (what branch you are on)
>> git describe –tag (what tag you are on)

Optional but recommended to checkout a local DART branch such that you add/commit/track changes

>> git checkout –b dart_soilmoisture

Quickstart approach to getting local copy of DART:

‘Fork’ the git repository ~/NCAR/DART.git and to set up a remote ‘origin’ and ‘upstream’ branches. For more information see:

https://github.com/NCAR/DART/wiki

For more experienced git users and if you intend to share/develop DART code:

In general setting your remote branches such that the ‘upstream’ points to ~/NCAR/DART.git and ‘origin’ points to ~/<your_git_account/DART.git helps to obtain new DART features, and also if you push your local changes to the ‘origin’ the DART team can view them directly if you are having trouble.
2) Download & Navigate DART

- models/
- observations/
- assimilation_code/
- documentation/
- Others ...
- README/
- CHANGEL OG/
- build_templates/
- diagnostics/
2) Download & Navigate DART

**Important setup scripts**

DART/models/clm/

- model_mod.f90
- model_mod.nml
- model_mod.html

**Various input files**

- shell_scripts/
  - Model-specific scripts. None ‘required’.
- tests/
  - Optional. Not required.
- matlab/
  - Model-specific functions to aid analysis, if any.
- work

*Where all the action happens!*
2) Download & Navigate DART

Key CLM5-DART setup scripts: ~/DART/models/clm/shell_scripts/cesm2_2

1) Initialize model states
2) Generate ensemble spread
3) Assimilate observations

Biomass

~80 member free CLM ensemble

single CLM simulation

~time

<table>
<thead>
<tr>
<th>simple.csh</th>
<th>CLM5_startup_freerun</th>
<th>CLM5_setup_assimilation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DART_params.csh</td>
<td></td>
<td>CESM_DART_config</td>
</tr>
<tr>
<td></td>
<td></td>
<td>assimilate.csh</td>
</tr>
</tbody>
</table>
2) Download & Navigate DART

Key CLM setup scripts: ~/DART/models/clm/shell_scripts/cesm2_2

- **CLM5_setup_assimilation**
  Core setup script for CLM5-DART assimilation. Compiles CLM for multi-instance run, creates CLM case folder similar to normal CLM run.

- **DART_params.csh**
  Resource file used to customize CLM assimilation run. Edit before executing CLM5_setup_assimilation. Almost all edits happen this file.

- **CESM_DART_config**
  Execute in the caseroot directory after CLM compiles. Turns ‘on’ assimilation by bringing in all assimilation scripts and DART executables to caseroot directory.

- **assimilate.csh**
  Core assimilation script that executes DART executables. Enables communication between CLM files and DART.

- ~/DART/models/clm/work

- **input.nml**
  Input namelist file used to customize DART assimilation options.
2) Download & Navigate DART

- Download & Navigate DART
- Output.nc
- Restart.nc
  - Replace indeterminate values
- Analysis.nc
  - Clamp state values
- Forecast.nc
  - Apply mask to indeterminate values
- Preassim.nc
  - Apply Prior Inflation
  - Assimilation Updates State
- Postassim.nc
  - Apply Posterior Inflation
- DART 'filter' stages
CLM5-DART Tutorial Overview

1) Download CLM5
2) Download & Navigate DART
3) Meteorological Forcing
4) Initial Conditions
5) Observations
6) CLM & DART state space
7) Localization
8) Inflation
9) Assimilation Steps
10) Assimilation Diagnostics
11) Soil Moisture Assimilation
3) Meteorological Forcing

- Up to 80 different CAM ensemble members generates spread in CLM simulation

Generates user_nl_datm And datm.streams.txt* files for each ensemble member

Template Stream Files: ~/DART/model/clm/shell_scripts/cesm2_2

datm.streams.txt.CPLHISTForcing.nonSolarFlux_complete
datm.streams.txt.CPLHISTForcing.State3hr_complete
datm.streams.txt.CPLHISTForcing.State1hr_complete
datm.streams.txt.CPLHISTForcing.Solar_complete
3) Meteorological Forcing

Template Stream Files: ~/DART/model/clm/shell_scripts/cesm2_2

datm.streams.txt.CPLHISTForcing.nonSolarFlux_complete

<filePath>
  /glade/collections/rda/data/ds345.0/cpl_unzipped/NINST
</filePath>
<fileNames>
  f.e21.FHIST_BGC.f09_025.CAM6assim.011.cpl_NINST.ha2x3h.2011.nc
</fileNames>
</domainInfo>
<fieldInfo>
  <variableNames>
    a2x3h_Faxa_rainc  rainc
    a2x3h_Faxa_rainl  rainl
    a2x3h_Faxa_snowc  snowc
    a2x3h_Faxa_snowl  snowl
    a2x3h_Faxa_lwdn   lwdn
  </variableNames>
  <filePath>
    /glade/collections/rda/data/ds345.0/cpl_unzipped/NINST
  </filePath>
  1800
  </offset>
</fieldNames>

Note: ‘NINST’ overwritten with:
min: 01
max: 80

Folder of CAM reanalysis

Links CAM met variable names to CLM

CAM reanalysis global files
4) Initial Conditions

# configure settings:
#
# refcase  Name of the existing reference case that this run will start from.
# refyear  The specific date/time-of-day in the reference case that this run will start from. (Also see 'runtime settings' below for start year, start mon, start day and start tod.)
# refmon
# refday
# reftod
#
# stagedir  The directory location of the reference case files.
#
# startdate  The date used as the starting date for the hybrid run.
#
setenv refcase  clm5.0.06_f09_80  \[Resolution: f09_f09_mg17 (0.9x1.25 resolution)\]
setenv refyear  2011  \[Compset: 2000_DATM%GSWP3v1_CLM50%BGC-CROP_SICE_SOCN_MOSART_SGLC_SWAV\]
setenv refmon  01
setenv refday  01
setenv reftod  00000
setenv reftimestamp  ${refyear}-${refmon}-${refday}-${reftod}
setenv stagedir  /glade/p/cisl/dares/RDA_strawman/CESM_ensembles/CLM/CLM5B6C-Crop/ctsm_${reftimestamp}

# In a hybrid configuration, you can set the startdate to whatever you want.
# It does not have to match the reference (although changing the month/day seems bad).
# runtime settings:

setenv start_year  2011  \[Startdate for the assimilation tutorial run\]
setenv start_month  01
setenv start_day  01
setenv start_tod  00000
setenv startdate  ${start_year}-${start_month}-${start_day}  \[Need not align with refcase end-time\]
DART uses observation sequence files to store information about observations that are available for assimilation.

Default names are:

1. `obs_seq.in`  
   Input to `perfect_model_obs` for OSSEs

2. `obs_seq.out`  
   Input to `filter`, (output from `perfect_model_obs`).

3. `obs_seq.final`  
   Output from `filter`.

These files contain metadata describing observations, and may include a number of related values (for instance, the actual observation, the prior ensemble estimates, etc.).

**DART Obs Quality Control Flags** *(obs_seq.final)*

0. Assimilated
1. Evaluated only
2. Assimilated but posterior forward observation operator(s) failed
3. Evaluated only but posterior forward observation operator(s) failed
4. Not used, prior forward observation operator(s) failed
5. Not used because not selected in `obs_kind_nml`
6. Not used, failed prior quality control check
7. Not used, violated outlier threshold

---

1. Blank Template (no obs values, but holds location and time of obs)

2. Filled template (contains obs values)  
   CLM5_setup_pmo script used for this tutorial

3. Contains all diagnostic information of assimilation

---

Upcoming slide shows an overview of how a ‘biomass’ forward operator works.
5) Observations

Scripts that convert raw observation data (netcdf, hdf5, ascii) to obs_seq.out format.

- DART/observations/
  - forward_operators/
    Code to compute forward operators for many types of instruments and for some idealized models.
  - obs_converters/
    Directories containing code and build tools for programs that create observation sequence files from many data sources.
  - utilities/
    Code and build tools for utilities that manipulate observation sequence files. Available for low-order models (oned/) and large models (threed_sphere/).
5) Observations

Observation converters provided by DART

Given a way to compute the expected observation value from the model state, in theory any and all observations can be assimilated by DART through the `obs_seq.out` file. In practice this means a user-defined observation converter is required. DART provides many observation converters to make this process easier for the user. Under the directory `DART/observations/obs_converters` there are multiple subdirectories, each of which has at least one observation converter. The list of these directories is as follows:

<table>
<thead>
<tr>
<th>Observation</th>
<th>Directory</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atmospheric Infrared Sounder satellite retrievals</td>
<td>AIRS</td>
<td>HDF-EOS</td>
</tr>
<tr>
<td>Advanced Microwave Sounding Unit brightness temperatures</td>
<td>AIRS</td>
<td>netCDF</td>
</tr>
<tr>
<td>Aviso: satellite derived sea surface height</td>
<td>Aviso</td>
<td>netCDF</td>
</tr>
<tr>
<td>Level 4 Flux Tower data from AmeriFlux</td>
<td>Ameriflux</td>
<td>Comma-separated text</td>
</tr>
<tr>
<td>Level 2 soil moisture from COSMOS</td>
<td>COSMOS</td>
<td>Fixed-width text</td>
</tr>
</tbody>
</table>

And many, many more available, see web documentation
5) Observations

Contains all diagnostic information of assimilation

<table>
<thead>
<tr>
<th>obs_sequence</th>
<th>obs_kind_definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1 RADIOSONDE_U_WIND_COMPONENT</td>
</tr>
<tr>
<td></td>
<td>2 RADIOSONDE_V_WIND_COMPONENT</td>
</tr>
<tr>
<td></td>
<td>3 RADIOSONDE_SURFACE_PRESSURE</td>
</tr>
<tr>
<td></td>
<td>4 RADIOSONDE_TEMPERATURE</td>
</tr>
<tr>
<td></td>
<td>5 RADIOSONDE_SPECIFIC_HUMIDITY</td>
</tr>
<tr>
<td></td>
<td>6 AIRCRAFT_U_WIND_COMPONENT</td>
</tr>
<tr>
<td></td>
<td>7 AIRCRAFT_V_WIND_COMPONENT</td>
</tr>
<tr>
<td></td>
<td>8 AIRCRAFT_TEMPERATURE</td>
</tr>
<tr>
<td></td>
<td>9 ACARS_U_WIND_COMPONENT</td>
</tr>
<tr>
<td></td>
<td>10 ACARS_V_WIND_COMPONENT</td>
</tr>
<tr>
<td></td>
<td>11 ACARS_TEMPERATURE</td>
</tr>
<tr>
<td></td>
<td>12 MARINE_SFC_U_WIND_COMPONENT</td>
</tr>
<tr>
<td></td>
<td>13 MARINE_SFC_V_WIND_COMPONENT</td>
</tr>
<tr>
<td></td>
<td>14 MARINE_SFC_TEMPERATURE</td>
</tr>
<tr>
<td></td>
<td>15 MARINE_SFC_SPECIFIC_HUMIDITY</td>
</tr>
</tbody>
</table>

OBS 1

<table>
<thead>
<tr>
<th>lon lat</th>
<th>Vertical level, elevation (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>917.000000 917.000000 917.000000 917.000000 917.000000</td>
<td></td>
</tr>
</tbody>
</table>

Observation error variance

NCEP QC index
DART quality control

2 kinds of QC
5) Observations

The forward operator converts model state to the expected observation. Needs to link obs with correct CLM variable.

```
&obs_kind_nml
  assimilate_these_obs_types = 'SOIL_TEMPERATURE',
    'TOWER_NETC_ECO_EXCHANGE',
    'TOWER_LATENT_HEAT_FLUX',
    'TOWER_SENSIBLE_HEAT_FLUX',
    'MODIS_SNOWCOVER_FRAC',
    'MODIS_LEAF_AREA_INDEX',
    'BIOMASS',
    'OCO2_SIF',

evaluate_these_obs_types = 'null'
```

```
&model_nml
  clm_variables = ['leafc', 'QTY_LEAFCarbon',
    'frc_sno', 'QTY_SNOWCOVER_FRAC',
    'snow_depth', 'QTY_SNOW_THICKNESS',
    'H2Osoi_liq', 'QTY_SOIL_liquid_water',
    'H2Osoi_ice', 'QTY_SOIL_ICE',
    'T_soisno', 'QTY_TEMP',
    'livelstec', 'QTY_LIVE_STEM_CARBON',
    'deadstec', 'QTY_DEAD_STEM_CARBON',
    'dzsno', 'QTY_SNOW_THICKNESS',
    'zsno', 'QTY_SNOW_THICKNESS',
    'ZSNO', 'QTY_SNOW_THICKNESS',
    'NEP', 'QTY_NET_CARBON_PRODUCTION',
    'H2Osoi', 'QTY_SOIL_MOISTURE',
    'TLAI', 'QTY_LEAF_AREA_INDEX',
```

CLM variables are summed together and interpolated to location of observation set in obs_seq.out file.

Setting an outlier threshold protects against assimilating observations that are unrealistic or can make CLM crash.

```
&quality_control_nml
  input_qc_threshold = 1.0
  outlier_threshold = 3.0
```

Expected(prior mean - observation) = \sqrt{\sigma_{prior}^2 + \sigma_{obs}^2}.
Reject if (prior mean - observation) > T times expected value.
6) CLM and DART state space

DART state space:
1) Variables to be adjusted by DART
2) Variables required for forward operator

<table>
<thead>
<tr>
<th>&amp;model_nml clm_variables =</th>
<th>CLM name</th>
<th>DART quantity</th>
<th>Clamping values</th>
<th>domain</th>
<th>Overwrite?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>'leafc',</td>
<td>'QTY_LEAFCarbon',</td>
<td>'0.0', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'frac_sno',</td>
<td>'QTY_SNOWCOVER_Frac',</td>
<td>'0.0', '1.', 'restart',</td>
<td>'NO_COPY_BACK',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'SNOW_DEPTH',</td>
<td>'QTY_SNOW_thickness',</td>
<td>'0.0', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'H2OSO1_LIQ',</td>
<td>'QTY_SOIL LIQUID water',</td>
<td>'0.0', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'H2OSO1_ICE',</td>
<td>'QTY_SOIL ICE',</td>
<td>'0.0', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'T_SOISNO',</td>
<td>'QTY_TEMPERATURE',</td>
<td>'0.0', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'livestcm',</td>
<td>'QTY_LIVE STEM CARBON',</td>
<td>'0.0', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'deadstcm',</td>
<td>'QTY_DEAD STEM CARBON',</td>
<td>'0.0', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'DZN0',</td>
<td>'QTY_SNOW_thickness',</td>
<td>'0.0', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'ZSNO',</td>
<td>'QTY_SNOW_thickness',</td>
<td>'NA', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'ZISNO',</td>
<td>'QTY_SNOW_thickness',</td>
<td>'NA', 'NA', 'restart',</td>
<td>'UPDATE',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'NEP',</td>
<td>'QTY_NET Carbon PRODUCTION',</td>
<td>'NA', 'NA', 'history',</td>
<td>'NO_COPY_BACK',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'H2OSO1',</td>
<td>'QTY_SOIL MOISTURE',</td>
<td>'0.0', 'NA', 'history',</td>
<td>'NO_COPY_BACK',</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'TLAI',</td>
<td>'QTY_LEAF AREA INDEX',</td>
<td>'0.0', 'NA', 'vector',</td>
<td>'NO_COPY_BACK',</td>
<td></td>
</tr>
</tbody>
</table>
7) Localization

Spatial Localization Setting

- Reducing the cutoff limits the realm of influence an observation has upon surrounding model state.

- Figure of increments (color). Notice non-zero increments limited to location of observation (x).

```
# cutoff of 0.03 (radians) is about 200km
&assim_tools_nml
  filter_kind = 1
  cutoff = 0.05
```
8) Inflation

- Accounts for systematic errors in obs/model or sampling/regression errors

Settings for tutorial (prior inflation only):

```fortran
&filter_nml

inf_flavor       = 5,
inf_initial_from_restart = .true.,
inf_sd_initial_from_restart = .true.,
inf_deterministic = .true.,
inf_initial      = 1.0,
inf_lower_bound  = 0.0,
inf_upper_bound  = 20.0,
inf_damping      = 0.9,
inf_sd_initial   = 0.6,
inf_sd_lower_bound = 0.6,
inf_sd_max_change = 1.05,
/
```

- 5: Enhanced Spatially-varying state space inflation (inverse gamma)
- 2: Spatially-varying state space inflation (gaussian)

General Guidance:

- Start with no inflation inf_flavor = 0
- Enable prior inflation, no posterior inflation
- If suspect strong sampling/regression error turn on both prior and posterior inflation

Increases ‘apparent’ consistency between prior and observation.
CLM5-DART Tutorial Overview

1) Download CLM5
2) Download & Navigate DART
3) Meteorological Forcing
4) Initial Conditions
5) Observations
6) CLM & DART state space
7) Localization
8) Inflation
9) Assimilation Steps
10) Assimilation Diagnostics
11) Soil Moisture Assimilation
9) Assimilation Steps

a) Compile DART software

1)  >> cd ~/DART/build_templates/
2)  >> vi mkmf_template for Cheyenne environment (mkmf.template.intel.linux)
3)  >> cd ~/DART/models/clm/work/
4)  >> ./quickbuild.csh –mpi

Creates executables for all DART programs

MPIFC = mpif90
MPILD = mpif90
FC = ifort
LD = ifort

INCS = -I$(NETCDF)/include
LIBS = -L$(NETCDF)/lib -lncdf -lnetcdff
FFLAGS = -O -assume buffered_io $(INCS)
LDFLAGS = $(FFLAGS) $(LIBS)
9) Assimilation Steps

b) Modify `DART_params.csh` to match your personal environment

```bash
setenv cesmtag my_cesm Sandbox
setenv resolution f09_f09_mg17
setenv compset 2000_DATM%GSPv31_CLM50%BGC-CROP_SICE_SOCN_MOSART_SGLC_SAV
setenv num_instances 5

if (${num_instances} == 1) then
  setenv CASE clm5_f09_pmo_SIF
else
  setenv CASE clm5_f09_assim_e${num_instances}
endif

setenv use_SourceMods TRUE
setenv SourceModDir ~/SourceMods_release-cesm2.2.01/SourceMods

setenv cesmdata /glade/p/cesmdata/cseg/inputdata
setenv cesmroot /glade/work/${USER}/CESM/${cesmtag}/
setenv caseroot /glade/work/${USER}/cases/${cesmtag}/${CASE}
setenv cime_output_root /glade/scratch/${USER}/${cesmtag}/${CASE}
setenv rundir ${cime_output_root}/run
setenv exeroot ${cime_output_root}/bld
setenv archdir ${cime_output_root}/archive

setenv dartroot /glade/work/${USER}/git/DART_public
setenv baseobsdir /glade/p/cisl/dares/Observations/land

setenv project p ####
setenv machine cheyenne
```

Folder of cesm installation

CLM assimilation case name

Location of run/exe/restart/hist files

Directory of DART installation

Project charge account
9) Assimilation Steps

c) Compile CLM, create assimilation case

1) >> cd ~/DART/models/clm/shell_scripts/cesm2_2/
2) >> ./CLM5_setup_assimilation

Time to check the case.

1) cd /glade/scratch/bmraczka/cesm2.2.0/clm5_SWE_PR/run
   and check the compatibility between the namelists/pointer files
   and the files that were staged.

2) cd /glade/work/bmraczka/cases/cesm2.2.0/clm5_SWE_PR

3) check things

4) run a single job (and send mail), verify that it works without assimilation
   ./case.submit -M all

5) IF NEEDED, compile all the DART executables by
   cd /glade/work/bmraczka/DART/models/clm/work
   ./quickbuild.csh -mpi

6) Modify the case to enable data assimilation and
   run DART by executing
   cd /glade/work/bmraczka/cases/cesm2.2.0/clm5_SWE_PR
   ./CESM_DART_config
   and follow the directions.

7) Make sure the DART-related parts are appropriate.
   Check the input.nml
   Check the assimilate.csh or perfect_model.csh - as appropriate
   ./case.submit -M all

8) If that works
   ./xmlchange CONTINUE_RUN=TRUE
   ./xmlchange RESUBMIT=<number_of_cycles_to_run>

General instructions (1-8)
for a ‘new’ assimilation

For tutorial, skip 1-5, immediately go to step 6 to enable the assimilation.
Continue to next slide.
9) Assimilation Steps

d) Enable assimilation within CLM case

1)  
   >> cd <caseroot>

2)  
   >> ./CESM_DART_config

Check the DART configuration:

1) When you want to run DART, check that the CESM assimilation script is correct and then turn on data assimilation (if you need to). If your job has enough time to run multiple cycles in the same job, you can avoid recomputing for the queue by requesting multiple assimilation cycles in a single job. Each cycle will still use the same 'STOP_OPTION' and 'STOP_N'. This example requests two assimilation cycles instead of the default 1 cycle. You can run as many cycles as you like given limits of the queue and the amount of filesystem you can afford.

   cd /glade/work/bmraczka/cases/cesm2.2.0/clm5_SWE_PR
   ./xmlquery --partial ASSIMILATION
   ./xmlchange DATA_ASSIMILATION_LND=TRUE
   ./xmlchange DATA_ASSIMILATION_CYCLES=2

2) Modify what you need to in the DART namelist file, i.e. /glade/work/bmraczka/

3) If you have recompiled any part of the DART system, 'stage_dart_files' will copy them into the correct places.

4) Submit the CESM job in the normal way.

5) You can use /glade/work/bmraczka/cases/cesm2.2.0/clm5_SWE_PR/stage_cesm_files to stage files to restart a run. Make sure you check the script to specify the correct date to use for the restart. Pay attention to updating the pointer files to use the desired inflation files.
9) Assimilation Steps

e) Review and customize assimilation settings

>> vi input.nml : Below are excerpts of commonly used/modified namelist options:

```
&filter nml
   stages_to_write = 'forecast','preassim','analysis','output'
   inf_flavor = 5,
   inf_initial_from_restart = .true.,
   inf_sd_initial_from_restart = .true.,
```

**Diagnostic stages and inflation type**

**Observation Rejection Threshold**
```
&quality_control_nml
   input_qc_threshold = 1.0
   outlier_threshold = 3.0 /
```

**Fill inflation**
```
&fill_inflation_restart_nml
   write_prior_inf = .true.
   prior_inf_mean = 1.00
   prior_inf_sd = 0.6 /
```

**Spatial Localization Setting**
```
# cutoff of 0.03 (radians) is about 200km
&assim_tools_nml
   filter_kind = 1
   cutoff = 0.05 /
```

**Observation types to assimilate**
```
&obs_kind_nml
   assimilate_these_obs_types = 'SOIL_TEMPERATURE',
                              'TOWER_NETC_ECO_EXCHANGE',
                              'TOWER_LATENT_HEAT_FLUX',
                              'TOWER_SENSIBLE_HEAT_FLUX',
                              'MODIS_SNOWCOVER_FRAC',
                              'MODIS_LEAF_AREA_INDEX',
                              'MODIS_FPAR',
                              'BIO MASS',
                              'OCO2_SIF' /
```

**CLM variables to update or used in forward operator**
```
&model_nml
   clm_variables = 'leafc', 'QTY_LEAF_CARBON', '0.0', 'NA', 'restart', 'UPDATE',
                  'frac_sno', 'QTY_SNOWCOVER_FRAC', '0.0', '1', 'restart', 'ND_COPY_BACK',
                  'SNOW_DEPTH', 'QTY_SNOW_THICKNESS', '0.0', 'NA', 'restart', 'UPDATE',
                  'H2O_DELAY', 'QTY_SOIL_LIQUID_WATER', '0.0', 'NA', 'restart', 'UPDATE',
                  'H2O_DELAY', 'QTY_SOIL_ICR', '0.0', 'NA', 'restart', 'UPDATE',
                  'T_SOIL', 'QTY_TEMPERATURE', '0.0', 'NA', 'restart', 'UPDATE',
                  'livestenc', 'QTY_LIVE STEM_CARBON', '0.0', 'NA', 'restart', 'UPDATE',
                  'deadstenc', 'QTY_DEAD STEM_CARBON', '0.0', 'NA', 'restart', 'UPDATE',
                  'OZONE', 'QTY_SOIL_NITROGEN', '0.0', 'NA', 'restart', 'UPDATE',
                  'OZONE', 'QTY_SOIL_PATCH', '0.0', 'NA', 'restart', 'UPDATE' /
```

9) Assimilation Steps

f) Modify CLM run-time settings

>> cd <caseroot>

Commonly modified run-time settings:
(use ./xmlchange to set new value or ./xmlquery to view the current setting)

• DATA_ASSIMILATION_LND=TRUE
• STOP_OPTION=nhours
• STOP_N= 24 (daily assimilation)
• DATA_ASSIMILATION_CYCLES=1 (How many daily cycles? Review walltime, 30 min)
• RESUBMIT =0 (Resubmit the assimilation case for additional time increment)
• CONTINUE_RUN=FALSE (FALSE if 1st time step, TRUE if a continuation)

g) Submit the assimilation run to Cheyenne

>> ./case.submit

>> qstat -u <user-name>  # Check job status, time, ‘R’, ‘Q’
10) Assimilation Diagnostics

The job just completed – now what?

1) Check to make sure both the CLM and DART ran successfully:

```
>> cd ~/caseroot/
>> cat CaseStatus

Example of successful CLM time step

Example of successful DART step, if unsuccessful will provide location of ‘log’ file
```

```
run command is mpiexec_mpt -p "%g:" -np 360 omplace -tm open64 /glade/scratch/bmraczka/clm2.0/clm5_SWE_PR/bld/cesm.exe >> cesm.log.$SLID 2>&1
   Running /glade/work/bmraczka/cases/clm2.0/clm5_SWE_PR/assimilate.csh
   check for resubmit
dout_s False
mach cheyenne
resubmit_num 0
```
10) Assimilation Diagnostics

The entire job completed successfully, but CLM state variables are not being adjusted -- why?

```
clm_obs_seq.<date>.final

obs_sequence
obs_type_definitions
  1
  2 LPRM_SOIL_MOISTURE
```

Example 1: Obs accepted, model state adjusted

```
0.426458121052355  observation
0.453472528080195  prior ensemble mean
0.451989813949054  posterior ensemble mean
0.0000000000E+000  data product QC
0.0000000000E+000  DART QC
```

Example 2: Obs rejected, no model state change

```
0.273739010095596  observations
0.151536912474349  prior ensemble mean
0.151536912474349  posterior ensemble mean
0.0000000000E+000  data product QC
7.00000000000000  DART QC
```

Example 3: Obs accepted, no model state change

```
0.158023327589035  observations
0.162655747328743  prior ensemble mean
0.162655747328743  posterior ensemble mean
0.0000000000E+000  data product QC
0.0000000000E+000  DART QC
```

https://docs.dart.ucar.edu/en/latest/guide/dart-quality-control.html

The most common reasons assimilated obs have no impact on the model state include:

- Zero spread in ensemble members
- Cutoff value too small (Localization)
- Obs error values too large (less likely)
- No correlation (unlikely)
10) Assimilation Diagnostics

Example of more advanced diagnostics:

>> cd ~/DART/diagnostics/matlab/

1) input.nml &obs_diag_nml
2) ./obs_diag -->obs_diag_output.nc
3) (matlab) plot_rmse_xxx_evolution.m

Observation acceptance, RMSE and spread

Spatial Pattern of Biomass observation acceptance
11) Soil Moisture Assimilation

Use same tutorial settings, but with the following edits:

```plaintext
input.nml

&obs_kind_nml
  assimilate_these_obs_types = 'LPRM_SOIL_MOISTURE',
  evaluate_these_obs_types  = 'null'
/

&model_nml
  clm_variables = 'H2OSOI_LIQ', 'QTY_SOIL LIQUID_WATER', '0.0', 'NA', 'restart', 'UPDATE',
                  'H2OSOI', 'QTY_SOIL_MOISTURE', '0.0', 'NA', 'history', 'UPDATE',
/

CLM5_setup_assimilation
"hist_fincl1 = 'NEP','H2OSOI',

Synthetic soil moisture observations using: CLM5_setup_pmo

DART_params.csh
setenv baseobsdir /glade/scratch/bmraczka/Observations/land
```