DART Tutorial Section 16: Diagnostic Output
DART Diagnostic Output Categories:

• State-Space:
  Values of model’s state vector and inflation.
  Output using netCDF format.

• Observation-Space:
  Values of the observations.
  DART-specific obs_sequence format for now.

• Regression confidence factor:
  Values for state vector / observation pairs.
  Output as flat ASCII (soon to be netCDF).

• Program diagnostic output:
  Identification for source code version and namelist values.
  Error, warning, message output from modules.
State-Space Diagnostic Files:

Available in netCDF (a common data format)
http://www.unidata.ucar.edu/software/netcdf

DART outputs up to four state space diagnostic files.

These files are selected by listing their names for the stages_to_write entry in
the &filter_nml.

The stages_to_write namelist entry and resulting netCDF file names are:
- ‘forecast’ forecast.nc
- ‘preassim’ preassim.nc
- ‘postassim’ postassim.nc
- ‘analysis’ analysis.nc

In addition, stages_to_write can also include:
- ‘input’ input.nc Copy of initial conditions, same format as output.nc
- ‘output’ output.nc Output file for restart of subsequent filter steps.
State-Space Diagnostic Files:

Location of each diagnostic file in the filter cycle.

- **forecast.nc**
  - Apply Prior Inflation
  - Ensemble Forecasts

- **analysis.nc**
  - Apply Posterior Inflation

- **preassim.nc**
  - Assimilation Updates State

- **postassim.nc**
State-Space Diagnostic Files:

Contents of state space diagnostic files are controlled by &filter_nml:

```
&filter_nml
...
output_mean            = .true.  (include ensemble mean)
output_sd              = .true.  (include ensemble spread)
num_output_state_members = ##    (include this many individual ensemble members)
output_interval        = N        (only output every N\textsuperscript{th} assimilation time)
...
```

Note: output_interval for true_state.nc is in the &perfect_model_obs_nml namelist.

In input.nml for lorenz_96, make sure all diagnostic files are listed as stages\_to\_write. Run the filter to generate all files.

Try some Matlab diagnostics.
You can change the diagnostic file for a single plot by typing the file at the prompt, or...
You can change the file for all subsequent plots by setting Matlab variable diagn\_file.
For instance, diagn\_file = ‘postassim.nc’;
State-Space Diagnostic Files:

Trying out different diagnostic files:

In `input.nml` for lorenz_96, the default has been to output the `preassim.nc` and `analysis.nc` diagnostic files. You could also add `postassim` and `forecast` to the list in `stages_to_write`.

So far, have only looked at diagnostics for `preassim.nc`.

Two ways to change the diagnostic file in Matlab tools like `plot_total_err`:

1). Change for a single plot by entering diagnostic filename at Matlab prompt:

```
>> plot_total_err
  Input name of ensemble trajectory file: <cr> for preassim.nc
  analysis.nc
  Comparing true_state.nc and
      analysis.nc
```

2). You can change the file for all subsequent plots by setting Matlab variable `diagn_file`.

```
>> diagn_file = ‘analysis.nc’;
```

Try looking at diagnostics for `analysis.nc`, `forecast.nc`, and `postassim.nc`.
Some of these will be the same unless you have both prior and posterior inflation on.
DART State-Space Diagnostic functions

See the DART website section titled: “Configuring Matlab to work with DART”
http://www.image.ucar.edu/DARes/DART/DART2_Starting.php#matlab

ALL the DART Matlab state-space diagnostic functions are in diagnostics/matlab
This **must** be in your `matlabpath`.

Only focus on the files that start with `plot_`

- `plot_bins.m`
- `plot_correl.m`
- `plot_ens_err_spread.m`
- `plot_ens_mean_time_series.m`
- `plot_ens_time_series.m`
- `plot_phase_space.m`
- `plot_reg_factor.m`
- `plot_sawtooth.m`
- `plot_smooother_err.m`
- `plot_total_err.m`
- `plot_var_var_correl.m`
- …

Some, but not all, described here.

All functions have a ‘help’ section available in the standard Matlab way.
Viewing the State-Space netCDF files:

1. Standard DART matlab diagnostics:
   a. `plot_bins`: rank histograms,
Viewing the State-Space netCDF files:

1. Standard DART matlab diagnostics:
   b. `plot_correl`: correlation \( x(t) \) with all other state vars at all times,
Viewing the State-Space netCDF files:

1. Standard DART matlab diagnostics:
   c. `plot_ens_err_spread`: rms error and spread,

   ![Lorenz_96 model Var 1 Ensemble Error Spread for preassim.nc](image)
   - time–mean Ensemble Mean Total Error = 0.22454
   - time–mean Ensemble Spread = 0.27079

   ![Lorenz_96 model Var 13 Ensemble Error Spread for preassim.nc](image)
   - time–mean Ensemble Mean Total Error = 0.30668
   - time–mean Ensemble Spread = 0.31104

   ![Lorenz_96 model Var 27 Ensemble Error Spread for preassim.nc](image)
   - time–mean Ensemble Mean Total Error = 0.21285
   - time–mean Ensemble Spread = 0.26805
Viewing the State-Space netCDF files:

1. Standard DART matlab diagnostics:
   d. plot_ens_mean_time_series: just like the name says,
Viewing the State-Space netCDF files:

1. Standard DART matlab diagnostics:
   e. `plotEnsTimeSeries`: plots the ensemble (as available from `num_output_state_members`),

```
Lorenz_96 state varnum 1 Ensemble Members of preassim.nc
```

```
Lorenz_96 state varnum 13 Ensemble Members of preassim.nc
```

```
Lorenz_96 state varnum 27 Ensemble Members of preassim.nc
```
Viewing the State-Space netCDF files:

1. Standard DART matlab diagnostics:
   f. **plot_phase_space**: 3D phase space time evolution.
1. Standard DART matlab diagnostics:
   g. `plot_sawtooth`: truth, prior and posterior time series.
1. Standard DART matlab diagnostics:
   h. `plot_total_err`: total error for different fields,
Viewing the State-Space netCDF files:

1. Standard DART matlab diagnostics:
   i. `plot_var_var_correl`: x(t) correlation to single variable, all times.
2. **ncview**: a quick and surprisingly useful netCDF viewer.  
   [Link to ncview](http://meteora.ucsd.edu/~pierce/ncview_home_page.html)  
   Displays spatial slices, animations, time series ...
3. Many other graphical/analysis programs can read netCDF. (Note that we use udunits metadata convention.)

4. netCDF Operator (NCO) tools allow operations on netCDF files: (http://nco.sourceforge.net)
Selecting hyperslices of fields,
Differencing netCDF file,
Averaging, etc.

NASA GISS: Panoply
In addition to state variables, the netCDF files also contain time series of the state space inflation mean value and inflation standard deviation if adaptive inflation is on.

These fields are called:

- `state_priorinf_mean`,
- `state_priorinf_sd`,
- `state_postinf_mean`, and
- `state_postinf_sd`
Prior Inflation State-Space Diagnostic Files:

Contents of prior inflation fields for each diagnostic file in the filter cycle.

- **forecast.nc**: Undamped, same as previous time *analysis.nc*
- **analysis.nc**: Same as *postassim.nc*
- **preassim.nc**: Updated by assimilation
- **postassim.nc**: Damped, applied at this time

Ensemble Forecasts

Apply Prior Inflation

Apply Posterior Inflation

Assimilation Updates State
Posterior Inflation State-Space Diagnostic Files:

- **forecast.nc**
- **postassim.nc**
- **preassim.nc**
- **analysis.nc**

Contents of posterior inflation fields for each diagnostic file in the filter cycle.

- **Ensemble Forecasts**
- **Apply Prior Inflation**
- **Updated by observations.**
- **Assimilation Updates State**
- **Undamped, same as previous time analysis.nc**
- **Undamped, same as forecast preassim.nc**
- **Damped, what is applied postassim.nc**

DART Tutorial Section 16: Slide 21
Observation-space files:

Quick recap of ‘standard’ observation sequence file names (all names are actually specified in namelists):

- **obs_seq.in** input to `perfect_model_obs`
- **obs_seq.out** output from `perfect_model_obs`, also input to `filter`
- **obs_seq.final** output from `filter`

Observation sequence file output by `filter` has prior, posterior, observed value (and truth for OSSEs). For an overview, check out the DART webpage section: [http://www.image.ucar.edu/DARES/DART/DART2_Observations.php#obs_seq_overview](http://www.image.ucar.edu/DARES/DART/DART2_Observations.php#obs_seq_overview)

Contents of **obs_seq.final** controlled by `&filter_nml`:

```
&filter_nml
  obs_sequence_in_name = 'obs_seq.out'            Name of input observation sequence file.
  obs_sequence_out_name = 'obs_seq.final'         Name of output observation sequence file.
  num_output_obs_members = ##                   Output this many individual ensemble estimates.
  ...
```

DART Tutorial Section 16: Slide 22
Observation-space diagnostics:

The observation sequence file is not in a particularly user-friendly format. To aid in the evaluation and interpretation, a program named `obs_diag` must be run to produce a netCDF file with results that can be plotted in a manner of your choosing. DART has Matlab functions/scripts that create high-quality graphics.

See tutorial section 18 for full coverage of viewing, diagnosing obs sequences.

Here are a few of the Matlab functions available in `diagnostics/matlab`

- `plot_rank_histogram.m`
- `plot_evolution.m`
- `plot_rmse_XXX_evolution.m`
- `two_experiments_evolution.m` (works with more than two, actually)
- `plot_profile.m`
- `plot_bias_XXX_profile.m`
- `plot_rmse_XXX_profile.m`
- `two_experiments_profile.m` (works with more than two, actually)
Observation-space diagnostics:

SOME of the information in the observation sequence files can be converted to netCDF and easily plotted. A program named `obs_seq_to_netcdf` must be run to produce the netCDF.

Here are a few of the Matlab functions available in `diagnostics/matlab`.

- `link_obs.m`
- `plot_obs_netcdf.m`
- `plot_obs_netcdf_diffs.m`
Regression confidence factor output:

Reminder: \texttt{reg\_factor }\alpha \texttt{ introduced in Tutorial Section 13 – when running the group filter (with more than 1 group!).}

Controlled by \texttt{&reg\_factor\_nml}:

\begin{verbatim}
&reg_factor_nml
    save_reg_diagnostics = .true. Should file be output?
    reg_diagnostics_file = ‘reg_diagnostics’ Name of output file.
...
\end{verbatim}

File size could be (model size) X (number of obs.) X (number of assim times).
Very big, even for small models (only first 4 obs output default).

Normally, modify code in \texttt{reg\_factor\_mod.f90} to control:
Output is at end of select\_regression = 1 code block.
Format is ASCII:
\begin{verbatim}
    time in days, time in seconds, obs\_index, state\_index, \alpha
\end{verbatim}

Plot with Matlab \texttt{plot\_reg\_factor}.
Program Diagnostic Output:

File *dart_log.out*

All DART executables **append** to this file!

Contains:

- registration information
- Program start time,
- version of code for each module used
- Namelist values for each module*
- Names of output files,
- Diagnostic output for modules (through error_handler()),
- Warnings and fatal errors from DART code.

**Fair Warning:** This file is **not** cleared by DART. Can get very longggggggg ...
You should feel free to delete/rename it before starting the next experiment.

*may be in a separate file, depending on &utilities_nml setting
1. Filtering For a One Variable System
2. The DART Directory Tree
3. DART Runtime Control and Documentation
5. Comprehensive Filtering Theory: Non-Identity Observations and the Joint Phase Space
6. Other Updates for An Observed Variable
7. Some Additional Low-Order Models
8. Dealing with Sampling Error
9. More on Dealing with Error; Inflation
10. Regression and Nonlinear Effects
11. Creating DART Executables
12. Adaptive Inflation
13. Hierarchical Group Filters and Localization
14. Quality Control
15. DART Experiments: Control and Design
16. Diagnostic Output
17. Creating Observation Sequences
18. Lost in Phase Space: The Challenge of Not Knowing the Truth
19. DART-Compliant Models and Making Models Compliant
20. Model Parameter Estimation
21. Observation Types and Observing System Design
22. Parallel Algorithm Implementation
23. Location module design (not available)
24. Fixed lag smoother (not available)
25. A simple 1D advection model: Tracer Data Assimilation