UFS Land Data Assimilation (DA) System Demo

By: Gillian Petro

Land DA System

- Based on the NOAH-MP land surface model
- Performs JEDI-based snow data assimilation
- Uses the Local Ensemble Transform Kalman Filter-Optimal Interpolation (LETKF-OI) algorithm
- Runs on Hera & Orion
    - Containers available for other systems

Resources

- Land DA User's Guide
- GitHub Discussions
- Technical Documentation (for NOAH-MP)
- JEDI Documentation (Skylab v7.0)
- Land DA page (EPIC website)
- GitHub Wiki
- Data Bucket

Land DA Forecast Experiment

- Default case:
  - NOAH-MP land component from ufs-weather-model
  - GSWP3 atmospheric forcing data
  - 2 cycles
  - 2000-01-03 00z case
  - spack-stack v1.6.0
  - JEDI Skylab v7.0
- Latest *develop* branch commit* at time of recording: 09f445e
  - *Note that a pull request for updated documentation may be merged by the time of this demo or shortly thereafter.

Cloning Land DA

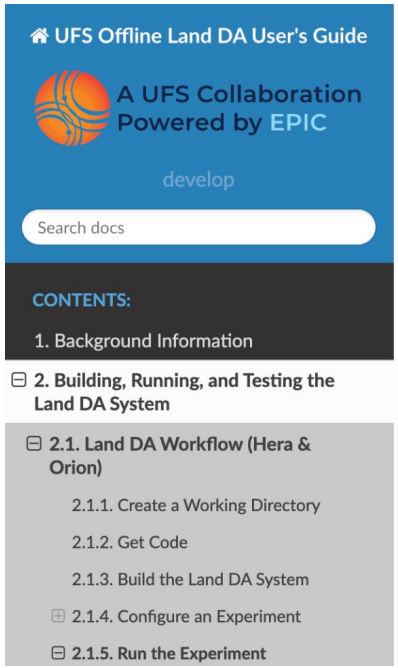
- To clone the **develop** branch, run:

  ```
  git clone -b develop --recursive
  https://github.com/ufs-community/land-DA_workflow.git
  ```

- To clone the **v1.2.0 release** branch, run:

  ```
  git clone -b release/public-v1.2.0 --recursive
  https://github.com/ufs-community/land-DA_workflow.git
  ```

Workflow Tasks



2.1.5. Run the Experiment

2.1.5.1. Workflow Overview

Each Land DA experiment includes multiple tasks that must be run in order to satisfy the dependencies of later tasks. These tasks are housed in the J-job scripts contained in the `jobs` directory.

Table 2.2 J-job Tasks in the Land DA Workflow

J-job Task	Description
JLANDDA_PREP_OBS	Sets up the observation data files
JLANDDA_PRE_ANAL	Transfers the snow data from the restart files to the surface data files
JLANDDA_ANALYSIS	Runs JEDI and adds the increment to the surface data files
JLANDDA_POST_ANAL	Transfers the JEDI result from the surface data files to the restart files
JLANDDA_PLOT_STATS	Plots the JEDI result (scatter/histogram)
JLANDDA_FORECAST	Runs the forecast model

https://land-da-workflow.readthedocs.io/en/develop/BuildingRunningTesting/BuildRunLandDA.html#run-the-experiment

Directory Structure

```
$LANDDAROOT: Base directory
 ├── land-DA_workflow(<CYCLEDIR>): Home directory of the land DA workflow
 └── ptmp (<PTMP>)
       └── <envir> (<OPSROOT>)
             └── com (<COMROOT>)
             │     ├── landda (<NET>)
             │     │     └── vX.Y.Z (<model_ver>)
             │     │           └── landda.YYYYMMDD (<RUN>.<PDY>): Directory containing the output files
             │     │                 ├── hofx
             │     │                 └── plot
             │     └── output
             │           └── logs
             │                 └── run_<forcing> (<LOGDIR>): Directory containing the log files for the Rocoto
             └── tmp (<DATAROOT>)
                   ├── <jobid> (<DATA>): Working directory
                   └── DATA_SHARE
                         ├── YYYYMMDD (<PDY>): Directory containing the intermediate or temporary files
                         ├── hofx: Directory containing the soft links to the results of the analysis task for p
                         └── DATA_RESTART: Directory containing the soft links to the restart files for the next
```

`<forcing>` refers to the type of forcing data used (`gswp3` or `era5`). Each variable in parentheses and angle brackets (e.g., `(<VAR>)`) is the name for the directory defined in the file `land_analysis.yaml`. In the future, this directory structure will be further modified to meet the NCO Implementation Standards.

3.1.3.1. NCO Directory Structure Entities

`HOMElandda:` **(Default: "&EXP_BASEDIR;/land-DA_workflow")**

The location of the land-DA_workflow clone.

`PTMP:` **(Default: "&EXP_BASEDIR;/ptmp")**

User-defined path to the `com` -type directories.

`envir:` **(Default: "test")**

The run environment. Set to "test" during the initial testing phase, "para" when running in parallel (on a schedule), and "prod" in production.

`COMROOT:` **(Default: "&PTMP;/&envir;/com")**

`com` root directory, which contains input/output data on current system.

`NET:` **(Default: "landda")**

Model name (first level of `com` directory structure)

`model_ver:` **(Default: "v1.2.1")**

Version number of package in three digits (second level of `com` directory)

`RUN:` **(Default: "landda")**

Name of model run (third level of com directory structure). In general, same as ${NET}.

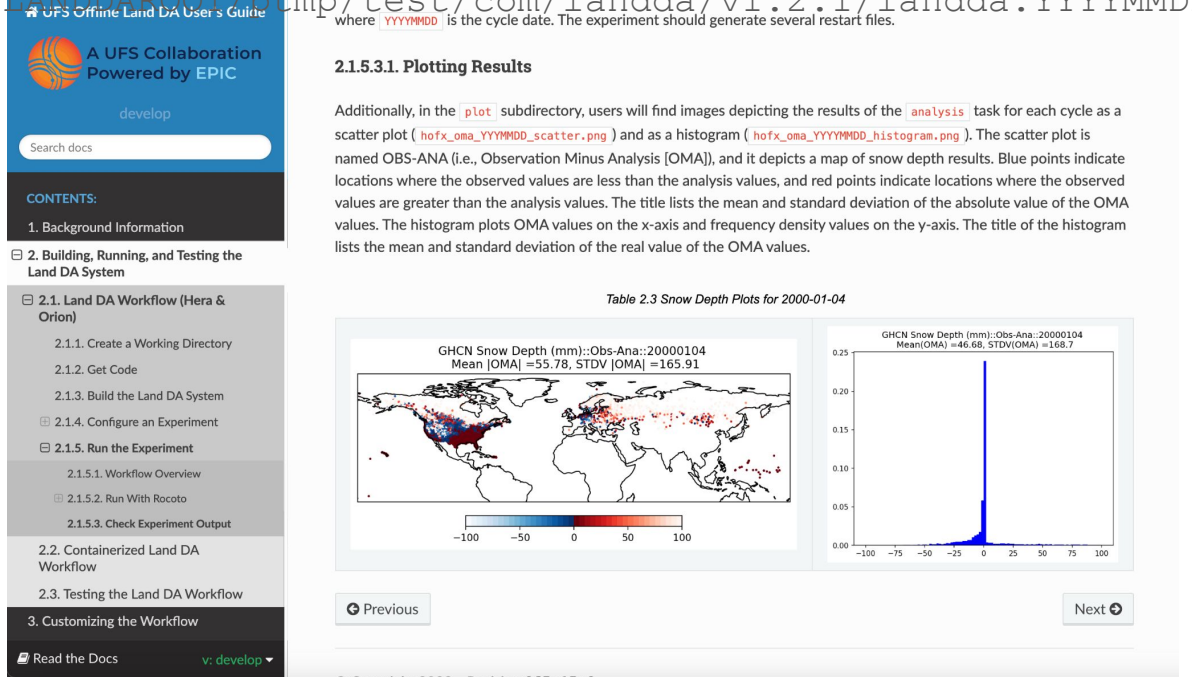
`DATAROOT:` **(Default: "&PTMP;/&envir;/tmp")**

NCO Implementation Standards: https://www.nco.ncep.noaa.gov/idsb/implementation_standards/

Plotting

Default plotting directory:

`$LANDDABOOT/ptmp/test/com/landda/v1.2.1/landda.YYYYMMDD/plot`



UIFCW24
A UFS Collaboration Powered by EPIC

THANK YOU!