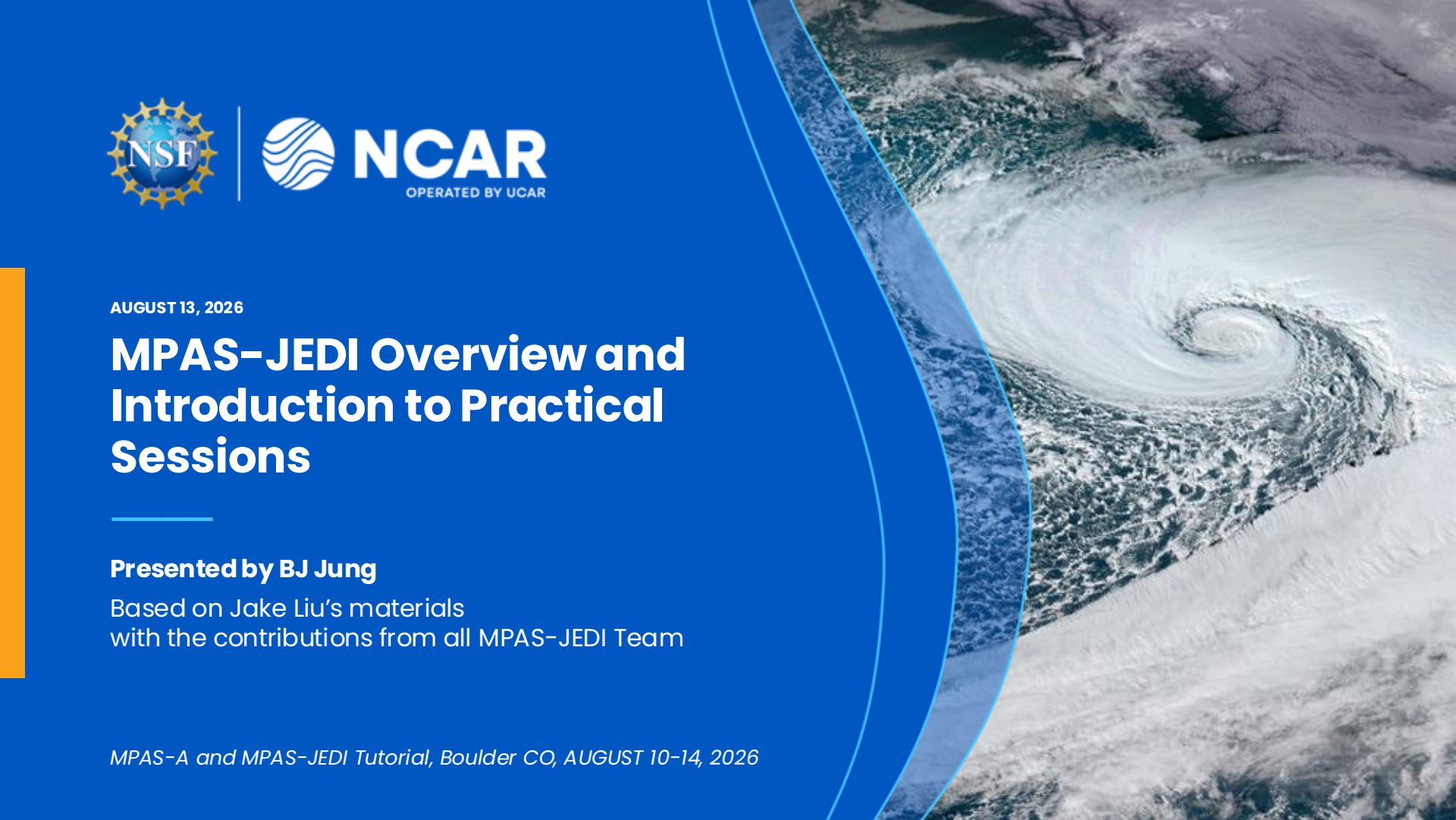

AUGUST 13, 2026

MPAS-JEDI Overview and Introduction to Practical Sessions

Presented by BJ Jung

Based on Jake Liu's materials
with the contributions from all MPAS-JEDI Team

MPAS-A and MPAS-JEDI Tutorial, Boulder CO, AUGUST 10-14, 2026

History of MPAS-JEDI

- Started from early 2018
- 1st release, September 2021
- 1st publication, Liu et al. (2022), EnVar and all-sky AMSU-A DA
- 2nd release, June 2023
- 1st tutorial, September 2023 at NCAR
- 3rd release, September 2024
- 4th release, June 2026

Latest release: MPAS-BUNDLE-4.0.1, July 2026

- MPAS-Bundle-4.0.1 code accessible from
  - https://github.com/JCSDA/mpas-bundle/releases/tag/4.0.1
- Based on the latest version of MPAS-A model and its interface to JEDI
  - https://github.com/JCSDA/mpas-jedi/releases/tag/4.0.1
  - https://github.com/MPAS-Dev/MPAS-Model/releases/tag/v8.4.1

- This is based on JEDI 2026.1 Release (June 2026), which includes

  https://github.com/JCSDA/oops
  https://github.com/JCSDA/saber
  https://github.com/JCSDA/ufo
  https://github.com/JCSDA/ioda
  https://github.com/JCSDA-internal/ioda-converters
  and more model-agnostic JEDI repositories

- **Other related tools:**

  - Python-based Diagnostic/Verification package included in:

  https://github.com/JCSDA/mpas-jedi/tree/release/4.0.1/graphics

  - Data assimilation cycling Workflow based on cylc:

  https://github.com/NCAR/MPAS-Workflow

What's new for v4.0.0 (and v4.0.1) ?

MPAS-Bundle V4.0 (full release note)

1. MPAS-Bundle v4 is based on MPAS v8.4.1 and the JEDI 2026.1 release.
2. Introducing the generic model-variable naming conventions in MPAS-JEDI
   - "uReconstructZonal" → "eastward_wind"
3. Improved MPAS-JEDI ctest suite
4. NCAR's obs2ioda into ioda-converters
5. Improved computational efficiency

MPAS-Bundle V4.0.1 (full release note)

1. MPAS-Bundle V4.0.1 is based on MPAS-Bundle v4.0.0, with the following additions
2. ufo is updated to include
   2.a. Add Harnisch et al. (2016) GEOIR observation error and cloud predictors for VarBC
   2.b. Adding code to thin single observations, and change default min_spacing
   2.c. Add Parameterized Polarimetric Radar Operator (P-PRO).
3. mpas-jedi is updated to include
   3.a. Add YAML data sidecars to all graphics analysis types

Main features with MPAS-JEDI

- Deterministic analysis:
  - 3DVar, 3D/4DEnVar, and hybrid-3D/4DEnVar with multi-resolution capability
- Ensemble analysis:
  - Ensemble of EnVar (**EDA**) with perturbed observations
  - **LETKF** and Gain form of LETKF (**LGETKF**)

- Analysis directly done on **MPAS unstructured grid** for uniform or variable-resolution mesh, global or regional mesh
  - **Analysis Variables**: u/v wind, Temperature, Specific Humidity, and Surface Pressure

- Allow to assimilate cloud-/precipitation-affected MW/IR satellite radiance data
  - Using **CRTM** (Community Radiative Transfer Model)
  - With mixing ratios of **hydrometeors** as part of analysis variables

- Radar DA: reflectivity and radial velocity
- Surface DA: Ps, t2m, q2m, u10, v10
- GNSS RO DA with multiple choices of RO operators

MPAS-JEDI status

Snyder et al: Data Assimilation with MPAS-JEDI: 2026 Annual update", Joint MPAS/WRF Users Workshop 2026 (link)

Recent progress with MPAS-JEDI

Improved baseline configuration

- Global 15-km EnVar, 80-member (G)LETKF, extensive suite of satellite obs

DA with 70-km MPAS top

- *Jung et al., talk 39A, Wed 11:15*

Enhancements aimed at higher-resolution DA

- Refinements for assimilation of geostationary infrared observations
- First tests with satellite-borne active sensors (GPM DPR, EarthCare CPR) [*Ban et al., poster P13, Wed PM*]
- Forward operator for dual-polarization radar obs
- Hourly cycling

Other improvements

- Use of commerical MW observations [*Baños et al. and Lin et al., talks 33A and 34A, 9:15 and 9:30 Wed*]
- Use of airborne GNSSRO [*Baños et al., JAMES 2026, submitted*]
- RTTOV v14 [*Kim et al., poster P14, Wed PM*]
- Test machine-learning emulator for propagating ensemble covariances [*Toedtli et al., talk 36A, 10:00 Wed*]

MPAS-JEDI publications

Liu, Z., and Coauthors, 2022: Data assimilation for the Model for Prediction Across Scales – Atmosphere with the Joint Effort for Data assimilation Integration (JEDI-MPAS 1.0.0): EnVar implementation and evaluation. Geoscientific Model Development, 15, 7859–7878, https://doi.org/10.5194/gmd-15-7859-2022 .

Guerrette, J. J., and Coauthors, 2023: Data assimilation for the Model for Prediction Across Scales – Atmosphere with the Joint Effort for Data assimilation Integration (JEDI-MPAS 2.0.0-beta): ensemble of 3D ensemble-variational (En-3DEnVar) assimilations. Geoscientific Model Development, 16, 7123–7142, https://doi.org/10.5194/gmd-16-7123-2023 .

Jung, B.-J., and Coauthors, 2024: Three-dimensional variational assimilation with a multivariate background error covariance for the Model for Prediction Across Scales – Atmosphere with the Joint Effort for Data assimilation Integration (JEDI-MPAS 2.0.0-beta). Geoscientific Model Development, 17, 3879–3895, https://doi.org/10.5194/gmd-17-3879-2024 .

Nystrom, R. G., C. Snyder, Z. Liu, B. J. Jung, J. Ban, and I. H. Banos, 2025: A Hybrid Four-Dimensional Variational Data Assimilation System for the Model for Prediction Across Scales (MPAS-Atmosphere): Leveraging the Joint Effort for Data Assimilation Integration (JEDI). Journal of Advances in Modeling Earth Systems, 17, e2025MS005183, https://doi.org/10.1029/2025MS005183 .

Schwartz, C. S., and Coauthors, 2025: A First Step toward Global Ensemble-Based Data Assimilation at Convection-Allowing Scales Using MPAS and JEDI. Monthly Weather Review, 153, 2139–2166, https://doi.org/10.1175/MWR-D-24-0155.1 .

Sun, T., J. J. Guerrette, Z. Liu, J. Ban, B.-J. Jung, I. Hernandez Banos, and C. Snyder, 2025: All-sky AMSU-A radiance data assimilation using the gain-form of Local Ensemble Transform Kalman filter within MPAS-JEDI-2.1.0: implementation, tuning, and evaluation. Geoscientific Model Development, 18, 8569–8587, https://doi.org/10.5194/gmd-18-8569-2025 .

Ban, J., Z. Liu, B.-J. Jung, I. H. Banos, B. Ruston, and A. Collard, 2026: All-sky ATMS radiance data assimilation with MPAS-JEDI. EGUsphere, 1–24, https://doi.org/10.5194/egusphere-2026-1047 .

Baños, I. H., Haase, J. S., Hordyniec, P., Cao, B., Liu, Z., Do, P.-N., 2026: Assimilating GNSS airborne radio occultation bending angles with MPAS-JEDI during a sequence of atmospheric rivers. Journal of Advances in Modeling Earth Systems, accepted.

Kong, R., Liu, Z., Zhang, G., Sun, T., Xie, H., Ban, J., and Wu, T.-C., 2026: Assimilation of Radar Radial Velocity and Polarimetric Observations Using LETKF within MPAS-JEDI: A Case Study of an Afternoon Thunderstorm in Taiwan, Monthly Weather Review, submitted.

Sun, T., I.-H. Chen, C. S. Schwartz, Z. Liu, H. Zhang, and D. Barker, 2026: Direct Radar Reflectivity Assimilation within MPAS-JEDI using Reflectivity Analysis Variable and Multivariate Background Error Covariance. EGUsphere, 1–29, https://doi.org/10.5194/egusphere-2026-2476

Instructions for practical exercises

https://www2.mmm.ucar.edu/projects/mpas-jedi/tutorial/202608NCAR

Basic info for working on NCAR HPC Derecho

ssh –x username@derecho.hpc.ucar.edu (Mac users may need to use 'ssh –Y')

You should be under bash after login: "echo $SHELL"

Submitting Jobs with PBS

qsub script	Submit a job script
qstat -u $USER	Check the status of your pending and running jobs
qdel job-id	Delete a queued or running job

This tutorial uses an account number: **UMMM0016** and the '**tutorial**' queue

NOTE:
Only during the dedicated practice time.
For other times, please use '**main**' queue instead.

Copy the “mpas_jedi_tutorial” folder

- **cd /glade/derecho/scratch/${USER}**
- **cp -r /glade/derecho/scratch/bjung/mpasjedi_tutorial202608NCAR ./mpas_jedi_tutorial**
- **ls -l mpas_jedi_tutorial**, *you will see*

```
drwxr-xr-x  2 bjung ncar 16384 Aug  6 13:42 abias/
drwxr-xr-x  3 bjung ncar 16384 Aug  6 13:38 background/
drwxr-xr-x  3 bjung ncar 16384 Aug  6 13:35 background_120km/
drwxr-xr-x  3 bjung ncar 16384 Aug 11 11:06 Bflow_global240km/
drwxr-xr-x  2 bjung ncar 16384 Aug 11 11:05 Bflow_preprocessing/
drwxr-xr-x  5 bjung ncar 16384 Aug  6 13:35 B_Matrix/
drwxr-xr-x  4 bjung ncar 16384 Aug 11 11:06 conus15km/
drwxr-xr-x  2 bjung ncar 16384 Aug  6 13:42 crtm_coeffs_v3/
drwxr-xr-x  3 bjung ncar 16384 Aug 11 11:05 cyclingDA/
drwxr-xr-x  3 bjung ncar 16384 Aug  6 13:34 ensemble/
drwxr-xr-x  2 bjung ncar 16384 Aug 11 11:07 MPAS_JEDI_yamls_scripts/
drwxr-xr-x  2 bjung ncar 16384 Aug  6 13:42 MPAS_namelist_stream_physics_files/
drwxr-xr-x  2 bjung ncar 16384 Aug  6 13:42 ncl_scripts/
drwxr-xr-x  3 bjung ncar 16384 Aug  6 13:35 obs_bufr/
drwxr-xr-x  4 bjung ncar 16384 Aug  6 13:53 omboma_from2experiments/
```

1st Practice: build and test MPAS-JEDI

1. *loading spack-stack build environment*

2. *cmake step*
   (Generates build configuration files)

3. *make step*
   (Executes build commands to compile code)

4. *ctest step*
   (runs the unit tests to ensure the compiled code behaves correctly)

1. loading spack-stack-1.9.3

- cd mpas_jedi_tutorial
- mkdir mpas_bundle_v4
- cd mpas_bundle_v4
- git clone –b release/4.0.1 https://github.com/JCSDA/mpas-bundle code
- **source ./code/env-setup/gnu-derecho.sh**
- module list

```
Currently Loaded Modules:
  1) ecflow/5.8.4                 18) python-venv/1.0          35) snappy/1.2.1
  2) mysql/8.0.33                 19) py-setuptools/69.2.0     36) zstd/1.5.6
  3) stack-gcc/12.4.0             20) py-pycodestyle/2.11.0    37) c-blosc/1.21.5
  4) craype/2.7.31                21) boost/1.84.0             38) hdf5/1.14.3
  5) ncarenv/24.12          (S)   22) eigen/3.4.0              39) netcdf-c/4.9.2
  6) gcc/12.4.0                   23) openblas/0.3.24          40) netcdf-cxx4/4.3.1
  7) libfabric/1.15.2.0           24) eckit/1.28.3             41) netcdf-fortran/4.6.1
  8) cray-mpich/8.1.29            25) fftw/3.3.10              42) parallel-netcdf/1.12.3
  9) stack-cray-mpich/8.1.29      26) fckit/0.13.2             43) parallelio/2.6.2
 10) glibc/2.38                   27) fiat/1.4.1               44) gsl-lite/0.37.0
 11) tar/1.34                     28) ectrans/1.5.0            45) nccmp/1.9.0.1
 12) gettext/0.22.5               29) qhull/2020.2             46) udunits/2.2.28
 13) zlib/1.2.11                  30) atlas/0.40.0             47) py-numpy/1.26.4
 14) sqlite/3.46.0                31) nghttp2/1.63.0           48) bufr/12.1.0
 15) util-linux-uuid/2.40.2       32) curl/8.10.1              49) py-pybind11/2.13.5
 16) python/3.11.7                33) cmake/3.27.9
 17) stack-python/3.11.7          34) ecbuild/3.7.2

  Where:
   S:  Module is Sticky, requires --force to unload or purge
```

Install spack-stack on your own machine will be covered Tomorrow afternoon.

2. CMake step

- git lfs install
- mkdir build; cd build
- **cmake ../code**

  # cmake will look for ../code/CMakeLists.txt file, this will take ~15-20min

before doing cmake under ~code

```
CMakeLists.txt
env-setup
LICENSE
README.md
scripts
```

after doing cmake under ~code

```
CMakeLists.txt   mpas-jedi-data
crtm             oops
env-setup        README.md
ioda             saber
iodaconv         scripts
ioda-data        test-data-release
jedicmake        ufo
LICENSE          ufo-data
MPAS             vader
mpas-jedi
```

after doing cmake under ~build

```
bin                       lib
CMakeCache.txt            lib64
CMakeFiles                Makefile
cmake_install.cmake       module
CPackConfig.cmake         MPAS
CPackSourceConfig.cmake   mpas-bundle-config.cmake
crtm                      mpas-bundle-config-version.cmake
CTestTestfile.cmake       mpas-jedi
DartConfiguration.tcl     mpas-jedi-data
_deps                     oops
ecbuild-cache.cmake       saber
ecbuild.log               share
ecbuild_tmp               Testing
ioda                      ufo
ioda-data                 ufo-data
jedicmake                 vader
```

Portions of lines from “vi code/CMakeLists.txt”

ECMWF software packages pre-built into spack-stack, thus not appear under ~code

```
ecbuild_bundle( PROJECT eckit     GIT "https://github.com/ecmwf/eckit.git" TAG 1.24.4 )
ecbuild_bundle( PROJECT fckit     GIT "https://github.com/ecmwf/fckit.git" TAG 0.11.0 )
ecbuild_bundle( PROJECT atlas     GIT "https://github.com/ecmwf/atlas.git" TAG 0.34.0 )
```

Model-agnostic components of JEDI

```
ecbuild_bundle( PROJECT oops      GIT "https://github.com/JCSDA/oops.git"         TAG 1.13.0 UPDATE )
ecbuild_bundle( PROJECT vader     GIT "https://github.com/JCSDA/vader.git"        TAG 1.10.0 UPDATE )
ecbuild_bundle( PROJECT saber     GIT "https://github.com/JCSDA/saber.git"        TAG 1.13.0 UPDATE )
ecbuild_bundle( PROJECT crtm      GIT "https://github.com/JCSDA/CRTMv3.git"       TAG v3.1.4 UPDATE )

ecbuild_bundle( PROJECT ioda-data GIT "https://github.com/JCSDA-internal/ioda-data.git"   TAG 2.12.0 UPDATE )
ecbuild_bundle( PROJECT ioda      GIT "https://github.com/JCSDA/ioda.git"         TAG 2.12.0 UPDATE )

ecbuild_bundle( PROJECT ufo-data  GIT "https://github.com/JCSDA-internal/ufo-data.git"    TAG 1.13.0 UPDATE )
ecbuild_bundle( PROJECT ufo       GIT "https://github.com/JCSDA/ufo.git"                  TAG 28e9c14 UPDATE )
if(BUILD_IODA_CONVERTERS)
    ecbuild_bundle( PROJECT iodaconv GIT "https://github.com/jcsda-internal/ioda-converters.git" TAG 1.0.0 )
endif()
```

MPAS-specific components of MPAS-JEDI

```
ecbuild_bundle( PROJECT MPAS      GIT "https://github.com/MPAS-Dev/MPAS-Model" TAG v8.4.1 )
option(ENABLE_MPAS_JEDI_DATA "Obtain mpas-jedi test data from mpas-jedi-data repository (vs tarball)" ON)
ecbuild_bundle( PROJECT mpas-jedi-data  GIT "https://github.com/JCSDA-internal/mpas-jedi-data.git"  TAG 4.0.0 UPDATE )
ecbuild_bundle( PROJECT mpas-jedi GIT "https://github.com/JCSDA/mpas-jedi"      BRANCH release/4.0.1 UPDATE )
```

3. make step under an interactive job

- qsub -A ummm0016 -N build-bundle -q tutorial -l job_priority=premium \

  -l walltime=03:00:00 -l select=1:ncpus=128:mem=235GB **–I**

- source ../code/env-setup/gnu-derecho.sh
- make –j32

- MPAS and MPAS-JEDI related executables under **~build/bin**

```
mpas_atmosphere                        mpasjedi_ens_mean_variance.x          mpasjedi_saca.x
mpas_atmosphere_build_tables           mpasjedi_error_covariance_toolbox.x   mpasjedi_variational.x
mpas_init_atmosphere                   mpasjedi_forecast.x                   mpas_namelist_gen
mpasjedi_convertstate.x                mpasjedi_gen_ens_pert_B.x             mpas_parse_atmosphere
mpasjedi_converttostructuredgrid.x     mpasjedi_hofx3d.x                     mpas_parse_init_atmosphere
mpasjedi_eda.x                         mpasjedi_hofx.x                       mpas_streams_gen
mpasjedi_enkf.x                        mpasjedi_process_perts.x
mpasjedi_enshofx.x                     mpasjedi_rtpp.x
```

4. ctest step, also under an interactive job

- cd mpas-jedi
- **ctest**

```
      Start  1: mpasjedi_coding_norms
 1/62 Test  #1: mpasjedi_coding_norms ........................   Passed    3.29 sec
      Start  2: mpasjedi_geometry
 2/62 Test  #2: mpasjedi_geometry ............................   Passed    7.94 sec
      Start  3: mpasjedi_state
 3/62 Test  #3: mpasjedi_state ...............................   Passed    1.06 sec
      Start  4: mpasjedi_model
 4/62 Test  #4: mpasjedi_model ...............................   Passed    3.23 sec
      Start  5: mpasjedi_increment
 5/62 Test  #5: mpasjedi_increment ...........................   Passed    0.75 sec

 ......

      Start 62: mpasjedi_3dhybrid_bumpcov_bumploc_2pe
62/62 Test #62: mpasjedi_3dhybrid_bumpcov_bumploc_2pe ........   Passed    2.95 sec

100% tests passed, 0 tests failed out of 62

Label Time Summary:
ci_oneapi_disable    =   6.05 sec*proc (1 test)
executable           =  19.89 sec*proc (12 tests)
mpasjedi             = 246.61 sec*proc (62 tests)
mpi                  = 243.32 sec*proc (61 tests)
script               = 226.72 sec*proc (50 tests)
tier2                = 113.22 sec*proc (15 tests)

Total Test time (real) = 246.74 sec
```

What a ctest case “Passed” means?

Each test run will produce text log files
(Under ~/build/mpas-jedi/test/testoutput)

```
4denvar_bumploc.ref
4denvar_bumploc.run
4denvar_bumploc.run.ref
4denvar_ID.ref      → existing reference file
4denvar_ID.run      → full text log file for the present test
4denvar_ID.run.ref  → shortened reference file
                      (part of the 4denvar_ID.run)
convertstate_bumpi
convertstate_bumpinterp.run
convertstate_bumpinterp.run.ref
convertstate_unsinterp.ref
```

- ➔ **4denvar_ID.run.ref** is compared with the existing **4denvar_ID.ref**.
- ➔ The test is deemed as **“Passed”** if numerical values between the two files are identical or within a **tolerance**.

Yaml configuration files under ~mpas-jedi/test/testinput

Useful for learning how to run various mpas-jedi applications

```
3denvar_amsua_allsky.yaml
3denvar_amsua_bc.yaml
3denvar_bumploc.yaml
3denvar_multi_resolution_regional.yaml
3denvar_multi_resolution.yaml
3dfgat_cda.yaml
3dfgat_pseudo.yaml
3dfgat.yaml
3dhybrid_bumpcov_bumploc.yaml
3dvar_bumpcov_nbam.yaml
3dvar_bumpcov_ropp.yaml
3dvar_bumpcov_rttovcpp.yaml
3dvar_bumpcov.yaml
3dvar.yaml
4denvar_bumploc.yaml
4denvar_ID.yaml
4denvar_VarBC_nonpar.yaml
4denvar_VarBC.yaml
4dfgat_cda.yaml
4dfgat.yaml
4dhybrid_bumpcov_bumploc.yaml
convertstate.yaml
converttostructuredgrid_latlon.yaml
dirac_bumpcov.yaml
dirac_bumploc.yaml
dirac_diffusion_duplicated.yaml
dirac_diffusion_univariate.yaml
dirac_noloc.yaml
dirac_spectral_loc.yaml
eda_3dhybrid_1.yaml
eda_3dhybrid_2.yaml
eda_3dhybrid_3.yaml
eda_3dhybrid_4.yaml
eda_3dhybrid.yaml
enshofx_1.yaml
enshofx_2.yaml
enshofx_3.yaml
enshofx_4.yaml
enshofx_5.yaml
enshofx.yaml
ens_mean_variance.yaml
errorcovariance.yaml
forecast.yaml
gen_ens_pert_B.yaml
geometry_iterator_2d.yaml
geometry_iterator_3d.yaml
geometry.yaml
getvalues.yaml
hofx3d_nbam.yaml
hofx3d_ropp.yaml
hofx3d_rttovcpp.yaml
hofx3d.yaml
hofx4d_pseudo.yaml
hofx4d.yaml
increment.yaml
letkf_3dloc.yaml
lgetkf_height_vloc.yaml
lgetkf.yaml
linvarcha.yaml
model.yaml
obslocalizations.yaml
obslocalization_vertical.yaml
obslocalization.yaml
obsop_name_map.yaml
parameters_bumpcov.yaml
parameters_bumploc_regional.yaml
parameters_bumploc.yaml
parameters_diffusion.yaml
process_perts_spectral.yaml
rtpp.yaml
state.yaml
```

Further reading about ctest

https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/inside/testing/index.html

- JEDI Testing
  - Running ctest
  - Manual Execution
  - The JEDI test suite
  - Tests as Applications
  - Initialization and Execution of Unit Tests
  - Anatomy of a Unit Test
  - Integration and System (Application) Testing
  - JEDI Testing Framework
- Adding a New Test
  - Step 1: Create a File for your Test Application
  - Step 2: Define A Test Fixture
  - Step 3: Define Your Unit Tests
  - Step 4: Register your Unit Tests with eckit
  - Step 6: Create an Executable
  - Step 7: Create a Configuration File
  - Step 8: Register all files with CMake and CTest
  - Adding an Application Test

Single vs. Double precision build of mpas-bundle

- *By default*, MPAS Model is built with a **double** precision in mpas-bundle.
- See ~/code/CMakeLists.txt

```
set(MPAS_DOUBLE_PRECISION "ON" CACHE STRING "MPAS-Model: Use double precision 64-bit Floating point.")
```

- For the large experiments, MPAS Model can be built with a single precision, so the memory requirement can be much reduced.
- In the CMake step,
  - **cmake -DMPAS_DOUBLE_PRECISION=OFF ../code**

NOTE
- Ctest reference files are generated with a double-precision build.
- Thus, some ctest with a single precision build will fail due to difference larger than tolerance.

Single vs. Double precision build of mpas-bundle

For example:

15km-30km Hybrid-3DEnVar

Assimilating the baseline obs (including clear sky AMSU-A and clear sky MHS)

8 nodes with total 512 PEs on Derecho HPC

Double precision

OOPS_STATS Run end - Runtime: **1173.92** sec, Memory: total: **1372.84** Gb, per task: min = 2.54 Gb, max = 7.71 Gb

Single precision

OOPS_STATS Run end - Runtime: **975.47** sec, Memory: total: **1136.91** Gb, per task: min = 2.18 Gb, max = 7.26 Gb

~17% saving with a single precision build

NOTE

- The MPAS precision only affects the mpas-jedi's state or increment.
- Many other JEDI components still use a double precision.