This material is based upon work supported by the NSF National Center for Atmospheric Research, which is a major facility operated by the U.S. National Science Foundation under Cooperative Agreement No. 1852977.

# MPAS-JEDI Overview and Introduction to Practical Sessions

**Jake Liu (liuz@ucar.edu)**

*Mesoscale & Microscale Meteorology Laboratory*
*NSF National Center for Atmospheric Research*

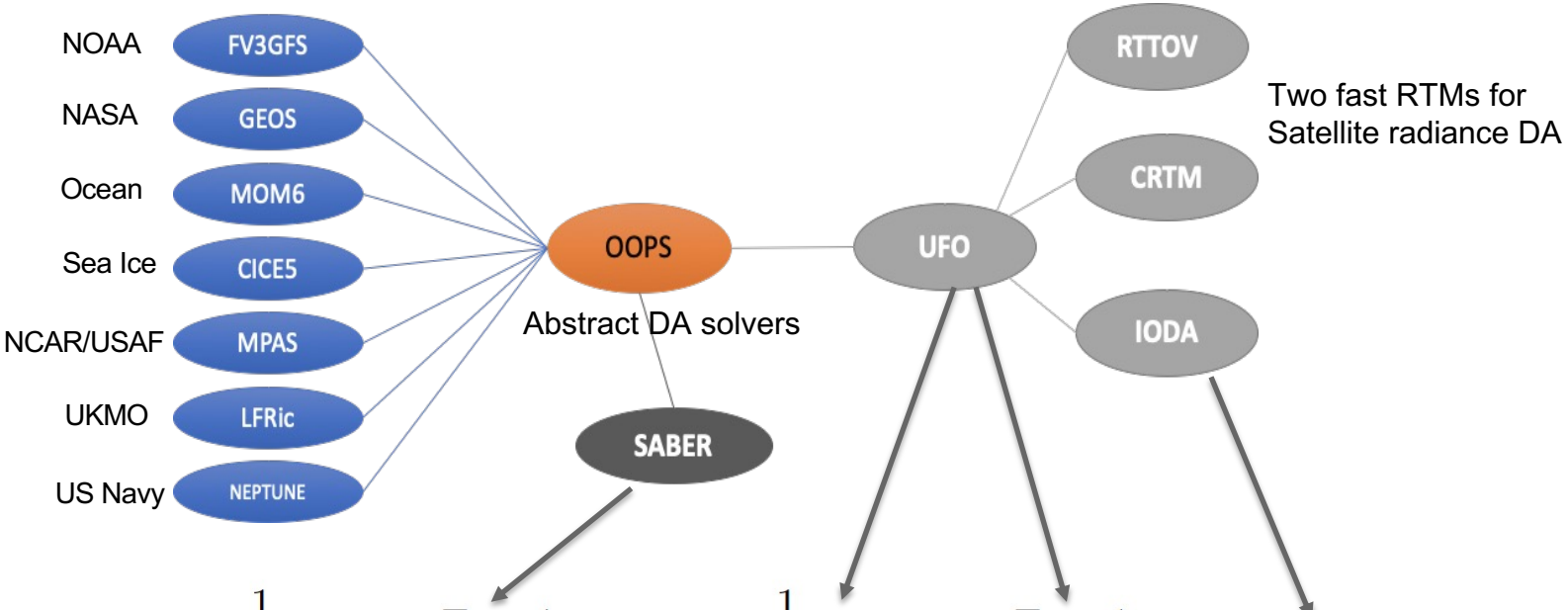




**MPAS-JEDI Tutorial, St Andrews, UK**
**25-26 June, 2025**

# Joint Effort for Data assimilation Integration (JEDI)

## led by Joint Center for Satellite Data Assimilation (JCSDA)




$$J(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_b)^{\mathbf{T}}\mathbf{B}^{-1}(\boldsymbol{x}-\boldsymbol{x}_b) + \frac{1}{2}[H(\boldsymbol{x})-\boldsymbol{y}]^{\mathbf{T}}\mathbf{R}^{-1}[H(\boldsymbol{x})-\boldsymbol{y}]$$

JCSDA and all partner groups contributing to JEDI's development

# 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
- 2nd publication, Guerrette et al. (2023), Ensemble of EnVar
- 1st tutorial, September 2023 at NCAR
- 3rd publication, Jung et al. (2024), Multivariate B, 3DVar, hybrid
- 3rd release, September 2024

# MPAS-JEDI publications


Liu Z et al., 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, Geosci. Model Dev., *15,* 7859–7878.

Guerrette, J. J. et al., 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, Geosci. Model Dev., 16, 7123–7142.

Jung et al.. 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), Geosci. Model Dev., 17, 3879–3895.

Ha, S. et al., 2024: Incremental analysis update (IAU) in the Model for Prediction Across Scales coupled with the Joint Effort for Data assimilation Integration (MPAS–JEDI 2.0.0), Geosci. Model Dev., 17, 4199–4211.

Sun, T. et al., 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, EGUsphere [preprint].

Schwartz C.S. et al., 2025: A first step toward global ensemble-based data assimilation at convection-allowing scales using MPAS and JEDI, Monthly Weather Review, Accepted.

# Latest stable version: MPAS-JEDI-3.0.2, Nov. 2024

- MPAS-JEDI-3.0.2 code accessible from
  - https://github.com/JCSDA/mpas-bundle/tree/3.0.2
- Based on the latest version of MPAS-A model and its interface to JEDI
  - https://github.com/MPAS-Dev/MPAS-Model/tree/v8.2.2
  - https://github.com/JCSDA/mpas-jedi/tree/release/3.0.2
    - Fixed a bug related to vertical height coordinate input to GNSS RO operator

Code of other repos as of September 2024 when making the 3.0.0 release:

https://github.com/JCSDA/oops
https://github.com/JCSDA/saber
https://github.com/JCSDA/ufo
https://github.com/JCSDA/ioda

**Other related tools:**

- Python-based Diagnostic/Verification package included in: https://github.com/JCSDA/mpas-jedi/tree/release/3.0.2/graphics
- Data assimilation cycling Workflow based on cylc: https://github.com/NCAR/MPAS-Workflow
- Observation processing, format conversion: https://github.com/NCAR/obs2ioda

# Main features with MPAS-JEDI

- Deterministic analysis:
  - **3DVar, 3D/4DEnVar**, and **hybrid-3D/4DEnVar with dual-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

# Instructions for practical exercises

https://www2.mmm.ucar.edu/projects/mpas-jedi/tutorial/202506NCAS

# 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: **UMMM0012** and the ‘**main**’ queue

# Copy the “mpas_jedi_tutorial” folder

**cd /glade/derecho/scratch/${USER}**

**cp -r /glade/derecho/scratch/liuz/mpasjedi_tutorial202506NCAS ./mpas_jedi_tutorial**

**ls -l mpas_jedi_tutorial**, *you will see*

```
drwxr-xr-x 3 liuz ncar 16384 Jun 18 15:08 background
drwxr-xr-x 3 liuz ncar 16384 Jun 18 15:07 background_120km
drwxr-xr-x 3 liuz ncar 16384 Jun 18 22:44 Bflow_global120km
drwxr-xr-x 2 liuz ncar 16384 Jun 18 21:55 Bflow_preprocessing
drwxr-xr-x 5 liuz ncar 16384 Jun 18 15:08 B_Matrix
drwxr-xr-x 4 liuz ncar 16384 Jun 18 21:04 conus15km
drwxr-xr-x 2 liuz ncar 16384 Jun 18 15:07 crtm_coeffs_v3
drwxr-xr-x 5 liuz ncar 16384 Jun 19 11:12 cyclingDA
drwxr-xr-x 3 liuz ncar 16384 Jun 18 15:07 ensemble
drwxr-xr-x 2 liuz ncar 16384 Jun 18 21:14 MPAS_JEDI_yamls_scripts
drwxr-xr-x 2 liuz ncar 16384 Jun 18 15:08 MPAS_namelist_stream_physics_files
drwxr-xr-x 3 liuz ncar 16384 Jun 18 15:07 obs_bufr
drwxr-xr-x 4 liuz ncar 16384 Jun 18 15:07 omboma_from2experiments
```

# 1st Practice: build and test MPAS-JEDI

1. *loading spack-stack build environment*

2. *cmake step*

3. *make step*

4. *ctest step*

# 1. loading spack-stack-1.6.0

- cd mpas-jedi-tutorial
- mkdir mpas_bundle_v3
- cd mpas_bundle_v3
- git clone –b release/3.0.2 https://github.com/JCSDA/mpas-bundle code
- **source ./code/env-setup/gnu-derecho.sh**
- module list

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

```
Currently Loaded Modules:
  1) ecflow/5.8.4                 35) base-env/1.0.0          69) json-schema-validator/2.1.0
  2) mysql/8.0.33                 36) boost/1.83.0            70) odc/1.4.6
  3) ncarenv/23.09          (S)   37) openblas/0.3.24          71) py-attrs/21.4.0
  4) gcc/12.2.0                   38) py-setuptools/63.4.3    72) py-pycparser/2.21
  5) stack-gcc/12.2.0             39) py-numpy/1.22.3         73) py-cffi/1.15.1
  6) craype/2.7.20                40) bufr/12.0.1             74) py-findlibs/0.0.2
  7) cray-mpich/8.1.25            41) ecbuild/3.7.2           75) py-eccodes/1.5.0
  8) libfabric/1.15.2.0           42) libpng/1.6.37           76) py-f90nml/1.4.3
  9) cray-pals/1.2.11             43) openjpeg/2.3.1          77) py-h5py/3.7.0
 10) stack-cray-mpich/8.1.25      44) eccodes/2.32.0          78) py-cftime/1.0.3.4
 11) tar/1.34                     45) eigen/3.4.0             79) py-netcdf4/1.5.8
 12) gettext/0.21.1               46) eckit/1.24.5            80) py-bottleneck/1.3.7
 13) libxcrypt/4.4.35             47) fftw/3.3.10             81) py-numexpr/2.8.4
 14) zlib/1.2.13                  48) fckit/0.11.0            82) py-et-xmlfile/1.0.1
 15) sqlite/3.43.2                49) fiat/1.2.0              83) py-openpyxl/3.1.2
 16) util-linux-uuid/2.38.1       50) ectrans/1.2.0           84) py-six/1.16.0
 17) python/3.10.13               51) qhull/2020.2            85) py-python-dateutil/2.8.2
 18) stack-python/3.10.13         52) atlas/0.35.1            86) py-pytz/2023.3
 19) nghttp2/1.57.0               53) git-lfs/3.3.0           87) py-pyxlsb/1.0.10
 20) curl/8.4.0                   54) gsibec/1.1.3            88) py-xlrd/2.0.1
 21) cmake/3.23.1                 55) gsl-lite/0.37.0         89) py-xlsxwriter/3.1.7
 22) git/2.41.0                   56) libjpeg/2.1.0           90) py-xlwt/1.3.0
 23) pkg-config/0.29.2            57) krb5/1.19.2             91) py-pandas/1.5.3
 24) hdf5/1.14.0                  58) libtirpc/1.3.3          92) py-pybind11/2.11.0
 25) snappy/1.1.10                59) hdf/4.2.15              93) py-pycodestyle/2.11.0
 26) zstd/1.5.2                   60) jedi-cmake/1.4.0        94) py-pyhdf/0.10.4
 27) c-blosc/1.21.5               61) libxt/1.1.5             95) libyaml/0.2.5
 28) netcdf-c/4.9.2               62) libxmu/1.1.4            96) py-pyyaml/6.0
 29) nccmp/1.9.0.1                63) libxpm/3.5.12           97) py-scipy/1.11.3
 30) netcdf-fortran/4.6.1         64) libxaw/1.0.13           98) py-packaging/23.1
 31) parallel-netcdf/1.12.2       65) udunits/2.2.28          99) py-xarray/2023.7.0
 32) parallelio/2.5.10            66) ncview/2.1.9           100) sp/2.5.0
 33) py-pip/23.1.2                67) netcdf-cxx4/4.3.1      101) jedi-base-env/1.0.0
 34) wget/1.20.3                  68) json/3.10.5            102) jedi-mpas-env/1.0.0

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

# 2. cmake step

- mkdir build; cd build
- git lfs install
- **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   oops
crtm             README.md
env-setup        saber
ioda             scripts
ioda-data        test-data-release
LICENSE          ufo
MPAS             ufo-data
mpas-jedi        vader
mpas-jedi-data
```

after doing cmake
under ~build

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

# 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 d772173 )
ecbuild_bundle( PROJECT vader     GIT "https://github.com/JCSDA/vader.git"        TAG 6d56a1e )
ecbuild_bundle( PROJECT saber     GIT "https://github.com/JCSDA/saber.git"        TAG bba6f7e )
ecbuild_bundle( PROJECT crtm      GIT "https://github.com/JCSDA/CRTMv3.git"       TAG 73102a2 )

ecbuild_bundle( PROJECT ioda-data GIT "https://github.com/JCSDA-internal/ioda-data.git"   TAG bcb0754 )
ecbuild_bundle( PROJECT ioda      GIT "https://github.com/JCSDA/ioda.git"                 TAG d49ed17 )

ecbuild_bundle( PROJECT ufo-data  GIT "https://github.com/JCSDA-internal/ufo-data.git"    TAG 9e4eb40 )
ecbuild_bundle( PROJECT ufo       GIT "https://github.com/JCSDA/ufo.git"                  TAG 94d50d6 )
```

MPAS-specific components of MPAS-JEDI

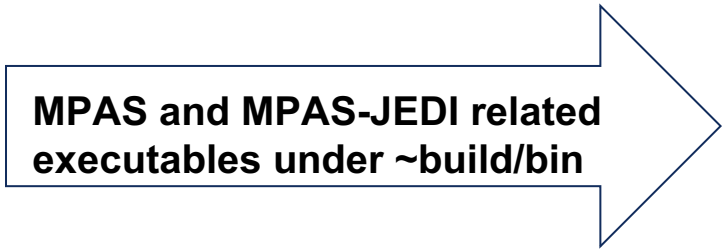
```
ecbuild_bundle( PROJECT MPAS      GIT "https://github.com/MPAS-Dev/MPAS-Model" TAG v8.2.2 )
option(ENABLE_MPAS_JEDI_DATA "Obtain mpas-jedi test data from mpas-jedi-data repository (vs tarball)" ON)
ecbuild_bundle( PROJECT mpas-jedi-data  GIT "https://github.com/JCSDA-internal/mpas-jedi-data.git"  TAG 12cdc56 )
ecbuild_bundle( PROJECT mpas-jedi GIT "https://github.com/JCSDA/mpas-jedi"     TAG 3.0.2.mmm )
```

# 3. make step under an interactive job

- `qsub -A ummm0012 -N build-bundle -q main -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 –j20




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

# 4. ctest step, also under an interactive job

export LD_LIBRARY_PATH=/glade/derecho/scratch/${USER}/mpas_jedi_tutorial/mpas_bundle_v3/build/lib:$LD_LIBRARY_PATH
cd mpas-jedi

**ctest**

```
.....

      Start 58: test_mpasjedi_3dvar_2pe
58/59 Test #58: test_mpasjedi_3dvar_2pe ........................   Passed    4.28 sec
      Start 59: test_mpasjedi_3dhybrid_bumpcov_bumploc_2pe
59/59 Test #59: test_mpasjedi_3dhybrid_bumpcov_bumploc_2pe .....   Passed    3.69 sec

100% tests passed, 0 tests failed out of 59

Label Time Summary:
executable    =  33.28 sec*proc (13 tests)
mpasjedi      = 319.85 sec*proc (59 tests)
mpi           = 314.23 sec*proc (58 tests)
script        = 286.58 sec*proc (46 tests)

Total Test time (real) = 319.94 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_2stream_bumploc.yaml
3denvar_amsua_allsky.yaml
3denvar_amsua_bc.yaml
3denvar_bumploc.yaml
3denvar_dual_resolution.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_append_obs.yaml
4dfgat.yaml
4dhybrid_bumpcov_bumploc.yaml
convertstate.yaml
converttostructuredgrid_latlon.yaml
dirac_bumpcov.yaml
dirac_bumploc.yaml
dirac_noloc.yaml
dirac_spectral_1.yaml
dirac_spectral_no_wind.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.yaml
process_perts_spectral_no_wind.yaml
rtpp.yaml
state.yaml
unsinterp.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

- For other practical sessions, you will use pre-compiled executables of mpas-bundle with the single-precision mode
  - **/glade/derecho/scratch/liuz/mpas_bundle_v3.0.2_public_gnuSP**

```
diff --git a/CMakeLists.txt b/CMakeLists.txt
index a4c8cb4..88a3ba4 100644
--- a/CMakeLists.txt
+++ b/CMakeLists.txt
@@ -64,7 +64,7 @@ ecbuild_bundle( PROJECT ufo       GIT "https://github.com/JCSDA/ufo.git"
 # Find external ESMF for mpas-model (optional)
 find_package(ESMF 8.3.0 MODULE)

-set(MPAS_DOUBLE_PRECISION "ON" CACHE STRING "MPAS-Model: Use double precision 64-bit Floating point.")
+set(MPAS_DOUBLE_PRECISION "OFF" CACHE STRING "MPAS-Model: Use double precision 64-bit Floating point.")
 set(MPAS_CORES init_atmosphere atmosphere CACHE STRING "MPAS-Model: cores to build.")
```

Just need a one-line change in building MPAS model

Note: ctest reference files are produced with the double-prevision build,
thus some of ctest cases with single-precision build will fail due to difference larger than tolerance