Downloading and compiling MPAS-Atmosphere

Michael G. Duda
NCAR/MMM

Basic system requirements

Before getting in to the details of how to download and compile the code, let's first answer the question:

On which systems can MPAS-Atmosphere be run?

There are essentially two considerations:

- Hardware

- Software

Hardware requirements

Generally speaking, no requirement regarding number of CPUs or type of CPU

- But you must have relatively modern Fortran and C/C++ compilers that target your CPU architecture

However, the amount of memory (RAM) dictates the maximum domain size

- Roughly 0.175 MB/memory per grid column (assuming 55 vertical levels and 'mesoscale_reference' physics suite)

As long as you're patient enough, and as long as your simulation fits in memory, a single processor is all you need!

- *On the upper end of CPU count, model generally scales well to around 100 grid columns per MPI task*

Software requirements

Generally, a UNIX-like OS is needed
- Linux, *BSD, macOS, AIX, etc.
- Untested: Windows Subsystem for Linux
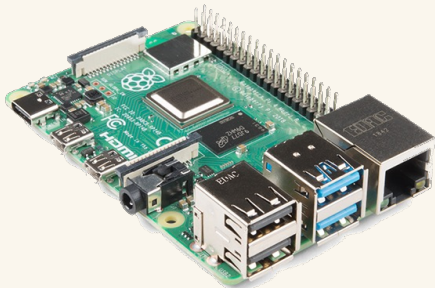
A POSIX shell and the usual programs.
- GNU Make, grep, awk, sed, cat, ranlib/libtool, etc.

Libraries
- MPI (OpenMPI, MPICH, etc.)
- Parallel-NetCDF

- zlib
- HDF5
- NetCDF4 (C + Fortran)
- PIO

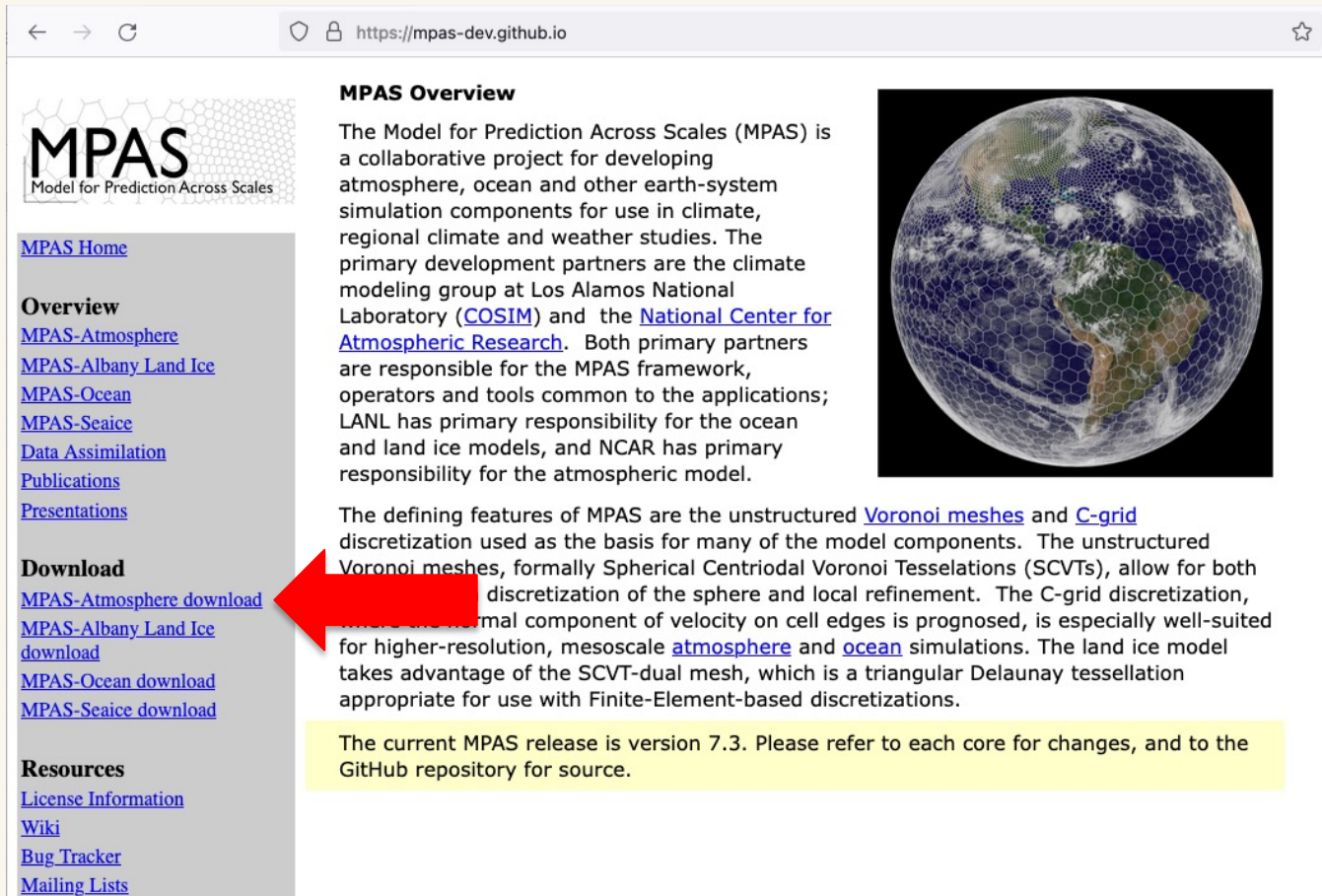If using MPAS v8.0+, the Simple MPAS I/O Layer (SMIOL) requires just the

If using MPAS v7.3 and earlier, or if using the PIO library.

Example systems that run MPAS-A

Working under the assumption that we have a system that meets the basic requirements, we can then ask:

> How do I get a copy of the source code?

There are essentially two options:

1) The "traditional", but not necessarily encouraged, method
   - Download a `.tar.gz` or `.zip` file

2) The preferred method
   - Make a *clone* of the MPAS-Model repository

The "less preferred" method of obtaining code

One can navigate to a download link from the MPAS homepage at https://mpas-dev.github.io/

MPAS
Model for Prediction Across Scales

One can navigate to a download link from the MPAS homepage at https://mpas-dev.github.io/

A variable resolution MPAS Voronoi mesh

The "less preferred" method of obtaining code

One can navigate to a download link from the MPAS homepage at https://mpas-dev.github.io/

The "less preferred" method of obtaining code

MPAS
Model for Prediction Across Scales

One can navigate to a download link from the MPAS homepage at https://mpas-dev.github.io/

MPAS
Model for Prediction Across Scales

MPAS Home

Overview
MPAS-Atmosphere
MPAS-Albany Land Ice
MPAS-Ocean
MPAS-Seaice
Data Assimilation
Publications
Presentations

Download
MPAS-Atmosphere download
MPAS-Albany Land Ice download
MPAS-Ocean download
MPAS-Seaice download

PLEASE READ THIS ENTIRE PAGE BEFORE DOWNLOADING ANY OF THE MPAS SOURCE CODE.

NOTE: If you intend to collaborate on a project, please read the Developer's guide before beginning work. The access method for the MPAS source code you choose will impact development efforts.

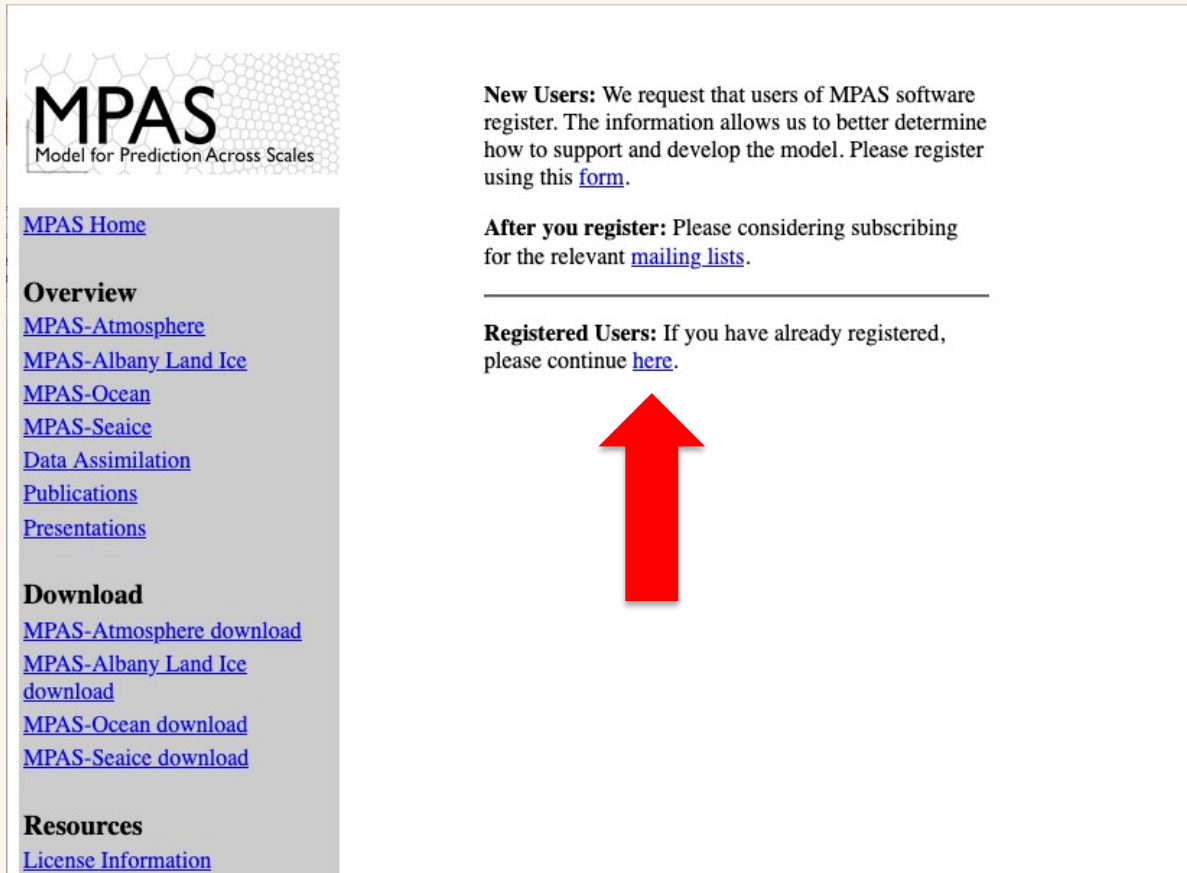The easiest way to acquire the MPAS source code is through the archive file below. Although this is the easiest method, it is also the least flexible. The method used will largely be determined by your use case. You should choose the method that aligns best with your use case.

Archive files are provided both in zip and tar.gz formats. Each release provides an archive file, and users should download the archive file for the most relevant released version.
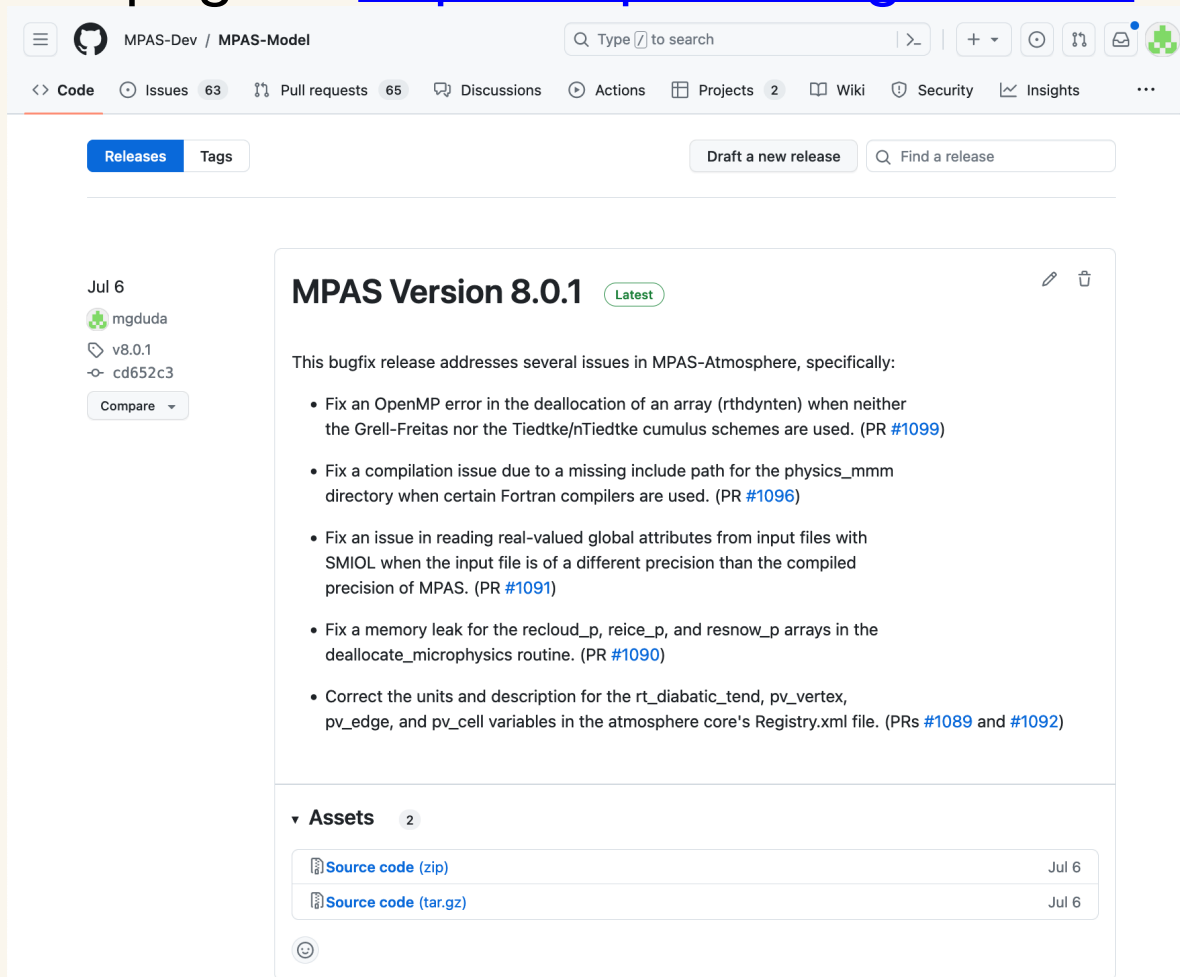
Archive file	Here

Our recommended method is to use git. This will allow users to easily update the version they are working with, and it facilitates the most direct method for contributing development back into the MPAS code base. This comes with the caveat of having to know how to use git. There are a few links at the bottom of this page that can help you get started with git. The most immediately useful tutorial is the GitHub fork tutorial, but the other tutorials will help you with learning how to make use of git.

GitHub Page	Here

Git related resources:

MPAS
Model for Prediction Across Scales

One can navigate to a download link from the MPAS homepage at https://mpas-dev.github.io/

Downloading a `.tar.gz` file of a particular release of the MPAS code certainly works, but it has several disadvantages:

1. You'll only obtain a specific release of the code

2. There's no direct way to see the history of changes to parts of the code

3. It's more difficult to see what local modifications have been made to the code

4. There's no easy route to updating to a newer release while preserving your local code modifications

The preferred method of obtaining code

A much better option is to *clone* the MPAS-Model repository
- The repository URL can be found from the MPAS GitHub page at https://github.com/MPAS-Dev/MPAS-Model

MPAS
Model for Prediction Across Scales

From the command-line, the following should be sufficient:
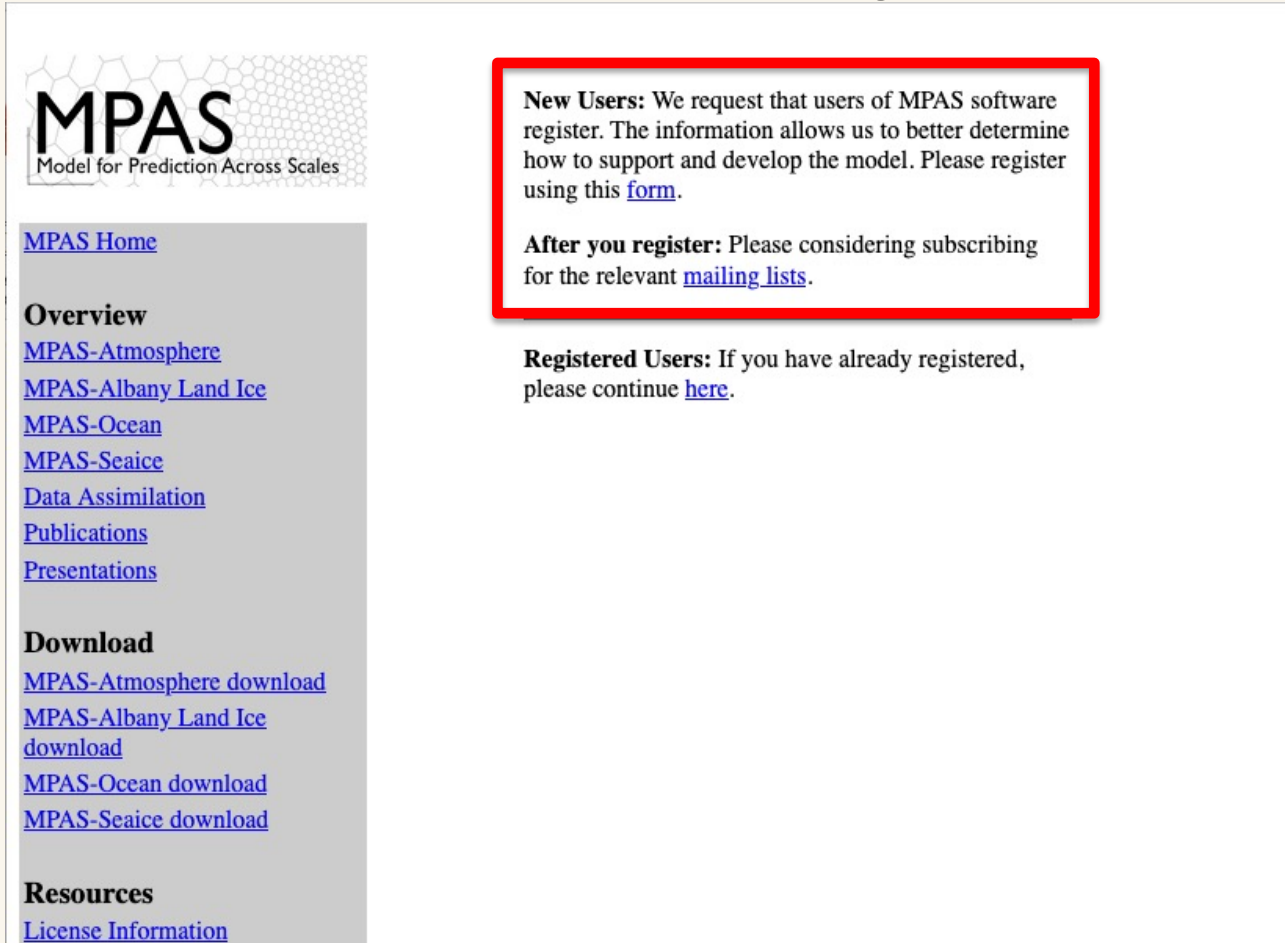
```
git clone https://github.com/MPAS-Dev/MPAS-Model.git
```

Cloning the repository should take about 10 seconds or less…
- *(I timed this at home, and it took 5.16 s)*

```
$ git clone https://github.com/MPAS-Dev/MPAS-Model.git
Cloning into 'MPAS-Model'...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (47/47), done.
remote: Total 46608 (delta 38), reused 42 (delta 23), pack-reused
46537
Receiving objects: 100% (46608/46608), 19.65 MiB | 2.57 MiB/s,
done.
Resolving deltas: 100% (35848/35848), done.
```

You may also like to register as an MPAS user and join the MPAS-Atmosphere Users mailing list

In order to compile MPAS and its required libraries, relatively modern Fortran, C, and C++ compilers are necessary

- The Fortran compiler should be recent enough to support the ISO_C_BINDING module from the Fortran 2003 standard, procedure pointer components of derived types, and a few bits of the Fortran 2008 standard

- Most versions of common compilers from the last couple of years should be fine

Building MPAS requires *at least* the following libraries:

- Any implementation of MPI-2, e.g., MPICH, MVAPICH, OpenMPI

    - Ensure that `mpif90` and `mpicc` commands are in your path

- Parallel-NetCDF (http://trac.mcs.anl.gov/projects/parallel-netcdf/)

    - Set `PNETCDF` environment variable to base installation directory

Installing the full set of I/O libraries required by MPAS can be tedious.

To make this easier, we've prepared a shell script that may be helpful:

```
https://www2.mmm.ucar.edu/projects/mpas/scripts/mpas_lib_install.sh
```

Notes:

- Before running or following the above script, you will need to have downloaded the library sources from
  ```
  https://www2.mmm.ucar.edu/projects/mpas/scripts
  ```

- If you already have an MPI library, skip the MPICH installation

- After editing paths and compiler names at the top, you *may* be able to run the script, but in general the script may be best used as a guide

Model Organization

MPAS-Model/

Checking out the MPAS code provides all MPAS models, not just MPAS-Atmosphere

- All models share a common set of infrastructure modules

- Each MPAS model is implemented as a "core" that lives in its own directory

- User must select which "core" to compile

- Each "core" is associated with a source code subdirectory under src/ and has a Registry file (similar to WRF)
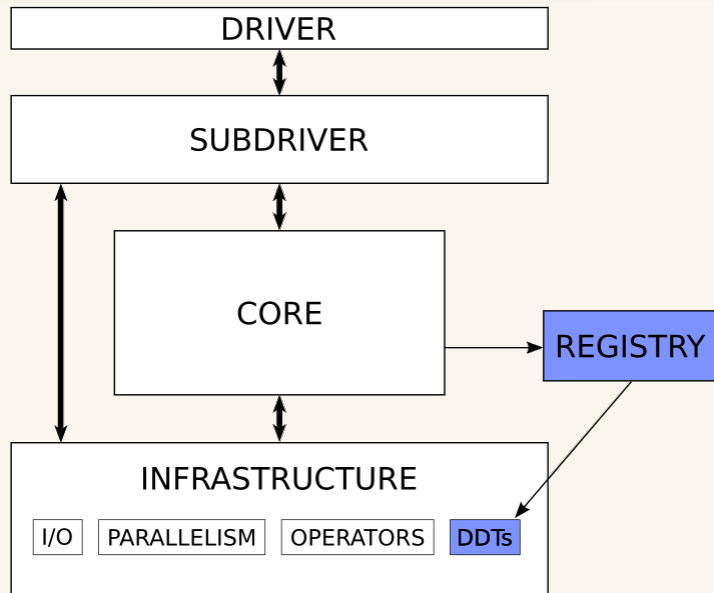
```
src/
    core_atmosphere/
    core_init_atmosphere/
    core_landice/
    core_ocean/
    driver/
    external/
    framework/
    operators/
```

NCAR | UCAR
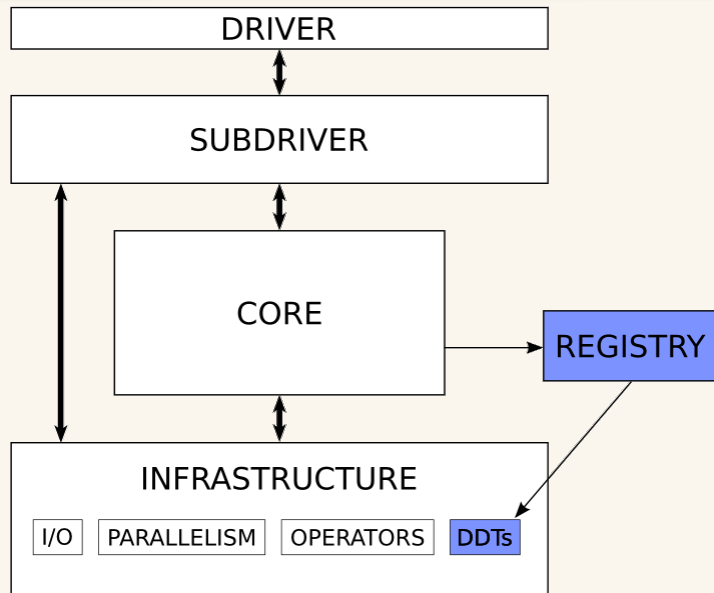
Running MPAS-Atmosphere involves two "cores":

- The **init_atmosphere** core is responsible for
  - Interpolating static fields to the mesh
  - Generating a vertical grid
  - Horizontally and vertically interpolating meteorological data to the 3-d grid
- The **atmosphere** core is the model itself; integrates forward in time from initial state

There is no "configuration" step for MPAS, unlike, e.g., for the WRF model

- All build flags are either set in the top-level `Makefile` or on the command-line

General MPAS build command:

$ make target CORE=core <options>

target can be either

    clean

or

    gfortran
    ifort
    nvhpc
    llvm
    ... plus a few others...

For MPAS-Atmosphere, core may be

    atmosphere
    init_atmosphere

<options> can be zero or more of

    DEBUG=true
    AUTOCLEAN=true
    PRECISION=single
    OPENMP=true

Typical build of both the init_atmosphere and atmosphere cores involves:

```
make gfortran CORE=init_atmosphere (build init_atmosphere_model)
```

```
make clean CORE=atmosphere (clean any infrastructure files used by both
                                      init_atmosphere and atmosphere)
```

```
make gfortran CORE=atmosphere (build atmosphere_model)
```

By default, MPAS cores are built with double-precision reals

MPAS-Atmosphere can be built in single precision

- Add `PRECISION=single` to build commands for single-precision executables
- execution time ~35% less compared with double-precision
- output files approximately half as large

Summary

- Ensure that you have a system running a UNIX-like operating system with the usual commands: make, awk, sed, grep, *git*, etc.

    - Available memory dictates maximum number of grid cells

- Install supporting libraries (MPI, Parallel-NetCDF; or if you need PIO: HDF5, NetCDF4, Parallel-NetCDF, PIO)

    - Can use `mpas_lib_install.sh` script as a guide

- Obtain the MPAS-Model source code with *git clone*

- Compile the *init_atmosphere* and *atmosphere* cores