

Downloading and compiling MPAS-Atmosphere

Michael G. Duda
NCAR/MMM



This material is based upon work supported by the NSF National Center for Atmospheric Research, a major facility sponsored by the U.S. National Science Foundation and managed by the University Corporation for Atmospheric Research. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of NSF.

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, git, grep, awk, sed, cat, ranlib/libtool, etc.

Libraries

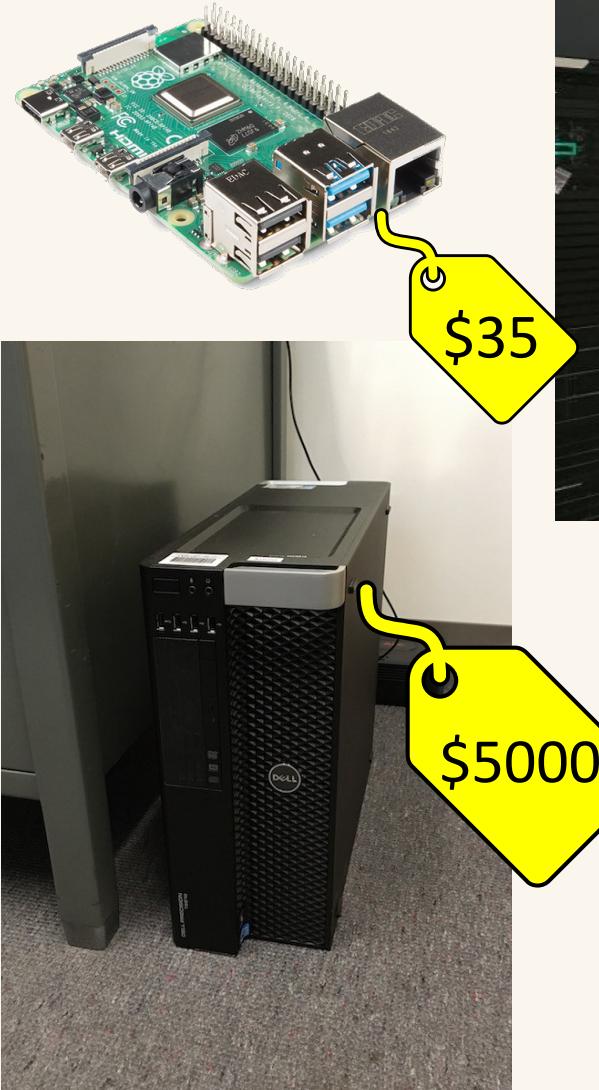
- MPI (OpenMPI, MPICH, etc.)
- Parallel-NetCDF

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

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

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

Example systems that run MPAS-A



Obtaining MPAS-Atmosphere source code

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/>

The screenshot shows the MPAS homepage with a sidebar on the left containing links for Home, Overview, Download, and Resources. A red arrow points to the "MPAS-Atmosphere download" link in the Download section. The main content area features an "MPAS Overview" section with text about the project's purpose and development partners, followed by a large image of the Earth with a hexagonal grid overlay.

MPAS Overview

The Model for Prediction Across Scales (MPAS) is a collaborative project for developing atmosphere, ocean and other earth-system simulation components for use in climate, regional climate and weather studies. The primary development partners are the climate modeling group at Los Alamos National Laboratory ([COSIM](#)) and the [National Center for Atmospheric Research](#). Both primary partners are responsible for the MPAS framework, operators and tools common to the applications; LANL has primary responsibility for the ocean and land ice models, and NCAR has primary responsibility for the atmospheric model.

The defining features of MPAS are the unstructured [Voronoi meshes](#) and [C-grid](#) discretization used as the basis for many of the model components. The unstructured Voronoi meshes, formally Spherical Centriodal Voronoi Tesselations (SCVTs), allow for both quasi-uniform discretization of the sphere and local refinement. The C-grid discretization, for which the normal component of velocity on cell edges is prognosed, is especially well-suited for higher resolution, mesoscale [atmosphere](#) and [ocean](#) simulations. The land ice model takes advantage of the SCVT-dual mesh, which is a triangular Delaunay tessellation appropriate for use with Finite-Element-based discretizations.

The current MPAS release is version 8.2.0. Please refer to each core for changes, and to the GitHub repository for source.

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 Atmosphere Public Releases

MPAS Atmosphere 8.2.0 was released on 27 June 2024.
For information on the GPU-enabled MPAS-Atmosphere model, please refer to [this documentation](#)

[MPAS Atmosphere 8.2.0 release notes](#)

Source code downloads:

- [MPAS v8.2.0](#)
- [GPU-enabled MPAS-Atmosphere v6.x](#)

[MPAS-Atmosphere Users' Guide](#)

[MPAS-Atmosphere tutorial](#)

[MPAS-Atmosphere meshes](#)

[Static geographical datasets](#)

[Monthly climatological aerosol dataset \(QNWFA_QNIFA_SIGMA_MONTHLY.dat\)](#)

[Configurations for idealized test cases](#)

[Sample input files for real-data simulations](#)

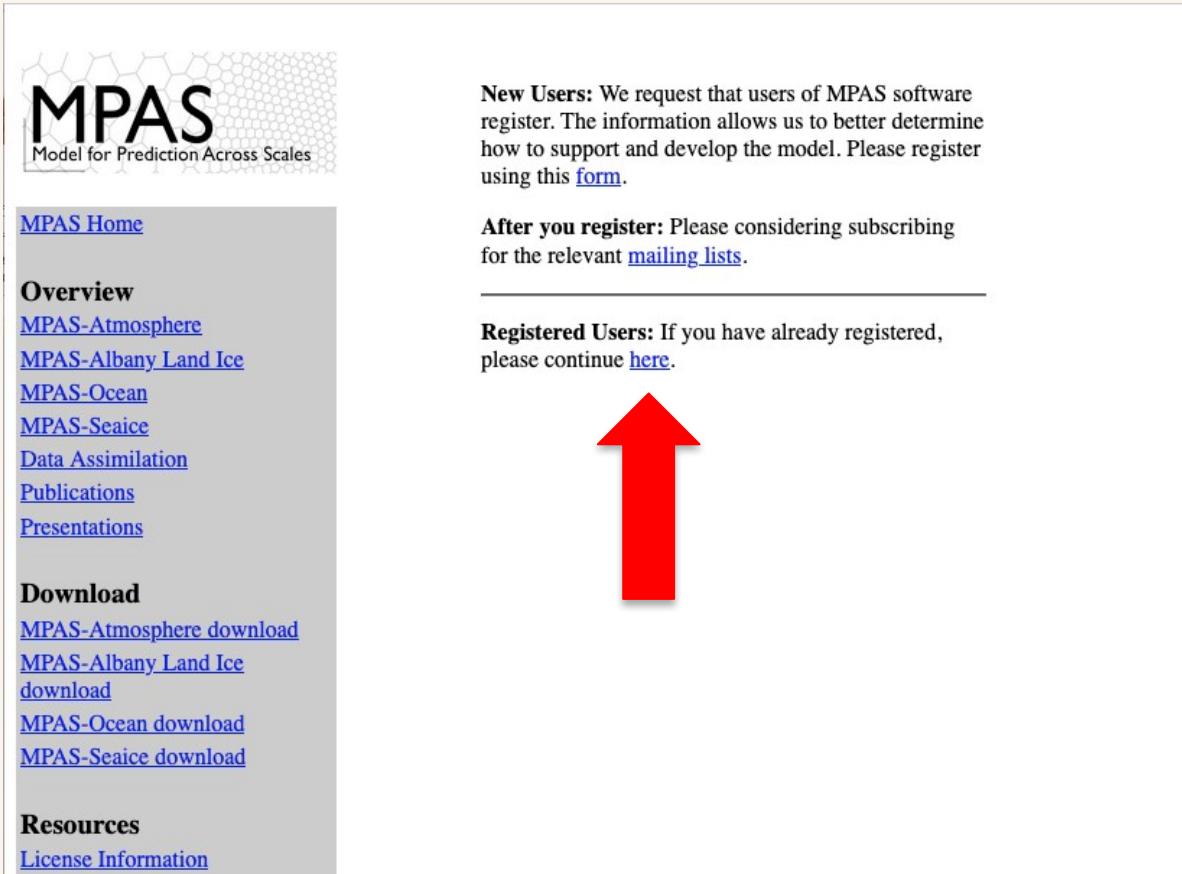
[Visualization and analysis tools](#)



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/>



New Users: We request that users of MPAS software register. The information allows us to better determine how to support and develop the model. Please register using this [form](#).

After you register: Please consider subscribing for the relevant [mailing lists](#).

Registered Users: If you have already registered, please continue [here](#).

Download

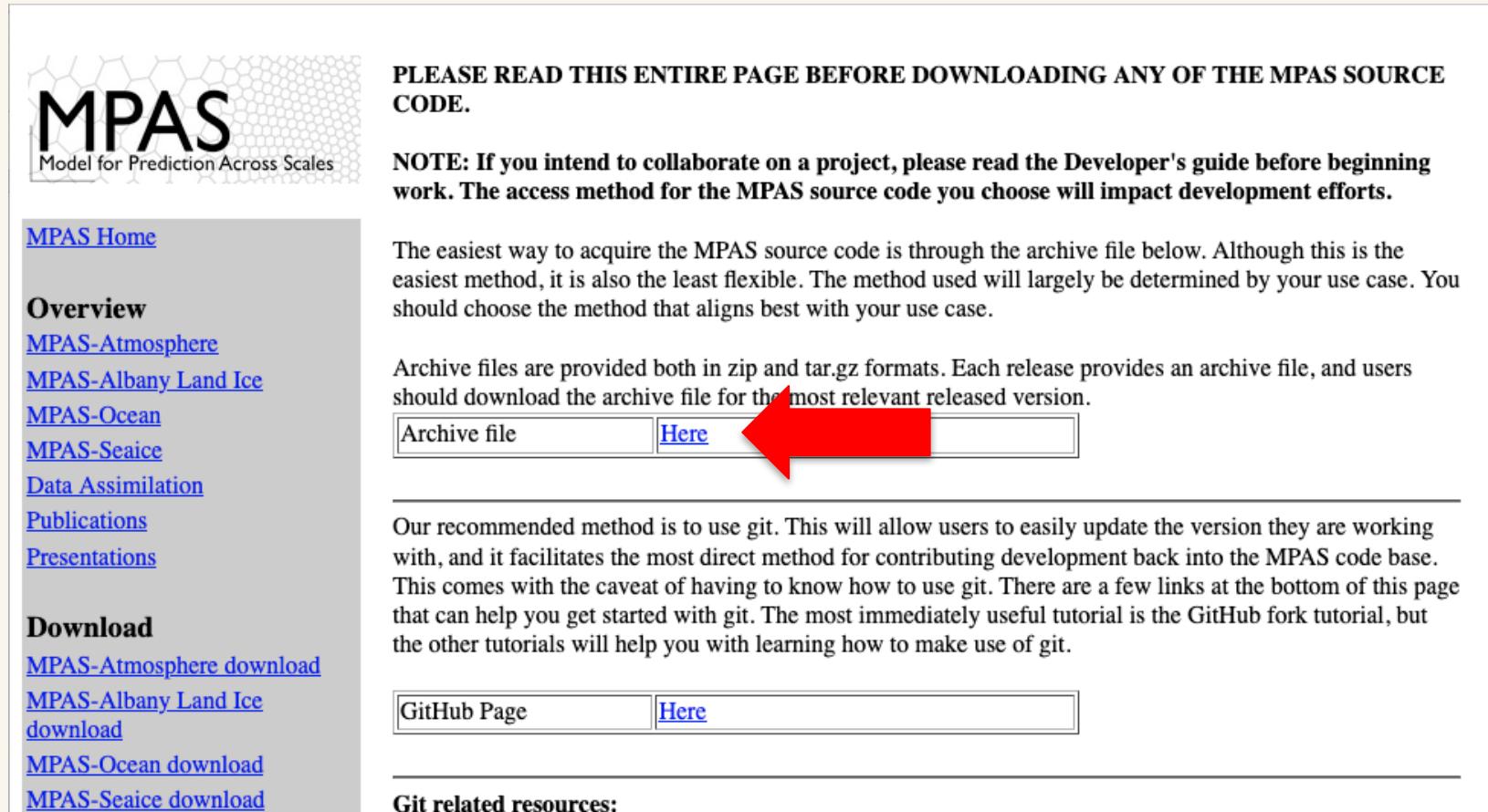
[MPAS-Atmosphere download](#)
[MPAS-Albany Land Ice download](#)
[MPAS-Ocean download](#)
[MPAS-Seaiice download](#)

Resources

[License Information](#)

The “less preferred” method of obtaining code

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



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](#)

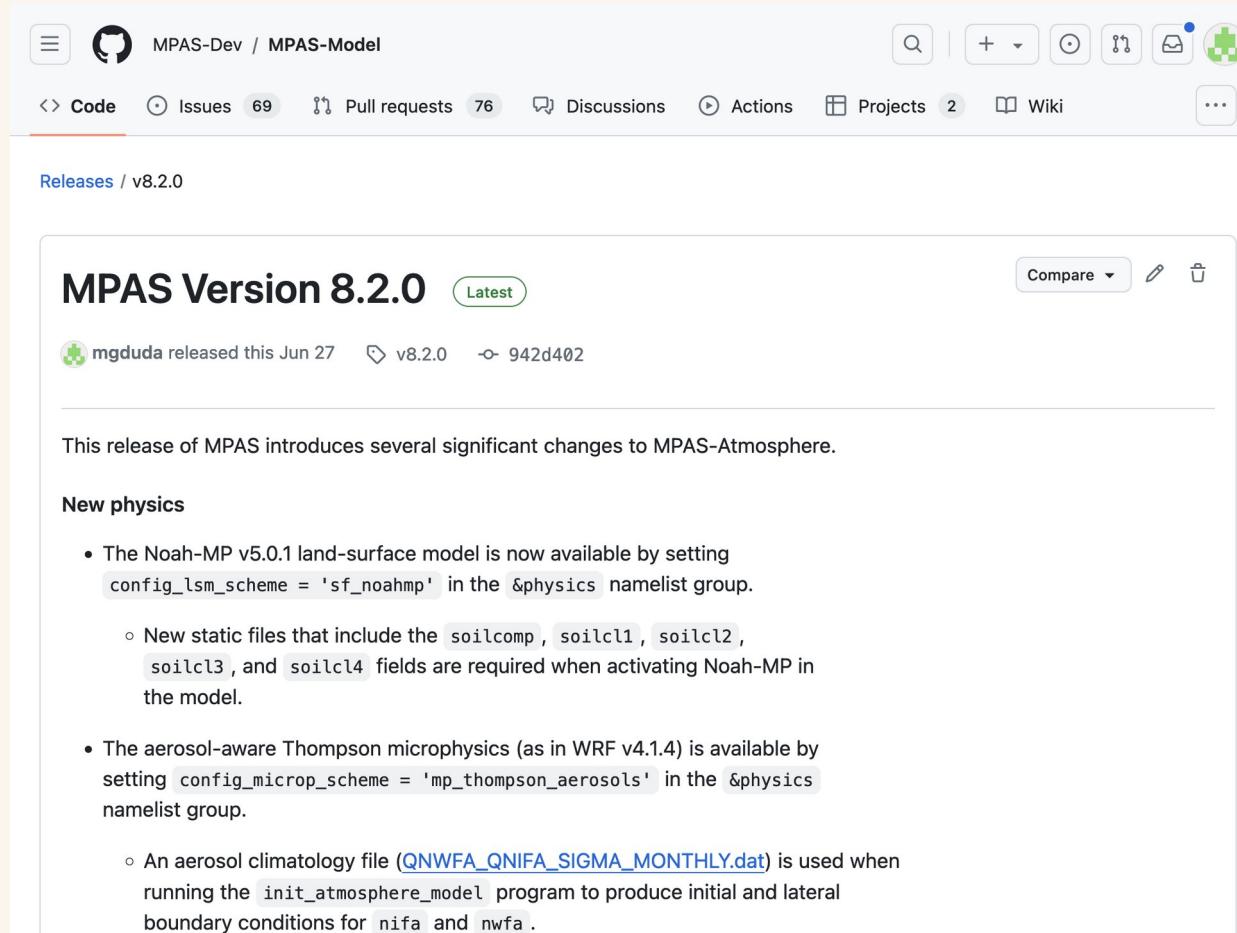
Our recommended method is to use git. This will allow users to easily update the version they are working with, and it facilitates the most direct method for contributing development back into the MPAS code base. This comes with the caveat of having to know how to use git. There are a few links at the 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:

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 screenshot shows the GitHub interface for the MPAS-Dev / MPAS-Model repository. The top navigation bar includes links for Code, Issues (69), Pull requests (76), Discussions, Actions, Projects (2), Wiki, and a three-dot menu. Below the navigation is a search bar and other repository management icons. The main content area is titled "Releases / v8.2.0". A card for "MPAS Version 8.2.0" (Latest) is displayed, showing it was released by mgduda on June 27, 2023, with commit hash 942d402. The card summary states: "This release of MPAS introduces several significant changes to MPAS-Atmosphere." Under the heading "New physics", there is a bulleted list of changes:

- The Noah-MP v5.0.1 land-surface model is now available by setting `config_lsm_scheme = 'sf_noahmp'` in the `&physics` namelist group.
 - New static files that include the `soilcomp`, `soilcl1`, `soilcl2`, `soilcl3`, and `soilcl4` fields are required when activating Noah-MP in the model.
- The aerosol-aware Thompson microphysics (as in WRF v4.1.4) is available by setting `config_microp_scheme = 'mp_thompson_aerosols'` in the `&physics` namelist group.
 - An aerosol climatology file ([QNWFA_QNIFA_SIGMA_MONTHLY.dat](#)) is used when running the `init_atmosphere_model` program to produce initial and lateral boundary conditions for `nifa` and `nwfa`.

The “less preferred” method of obtaining code

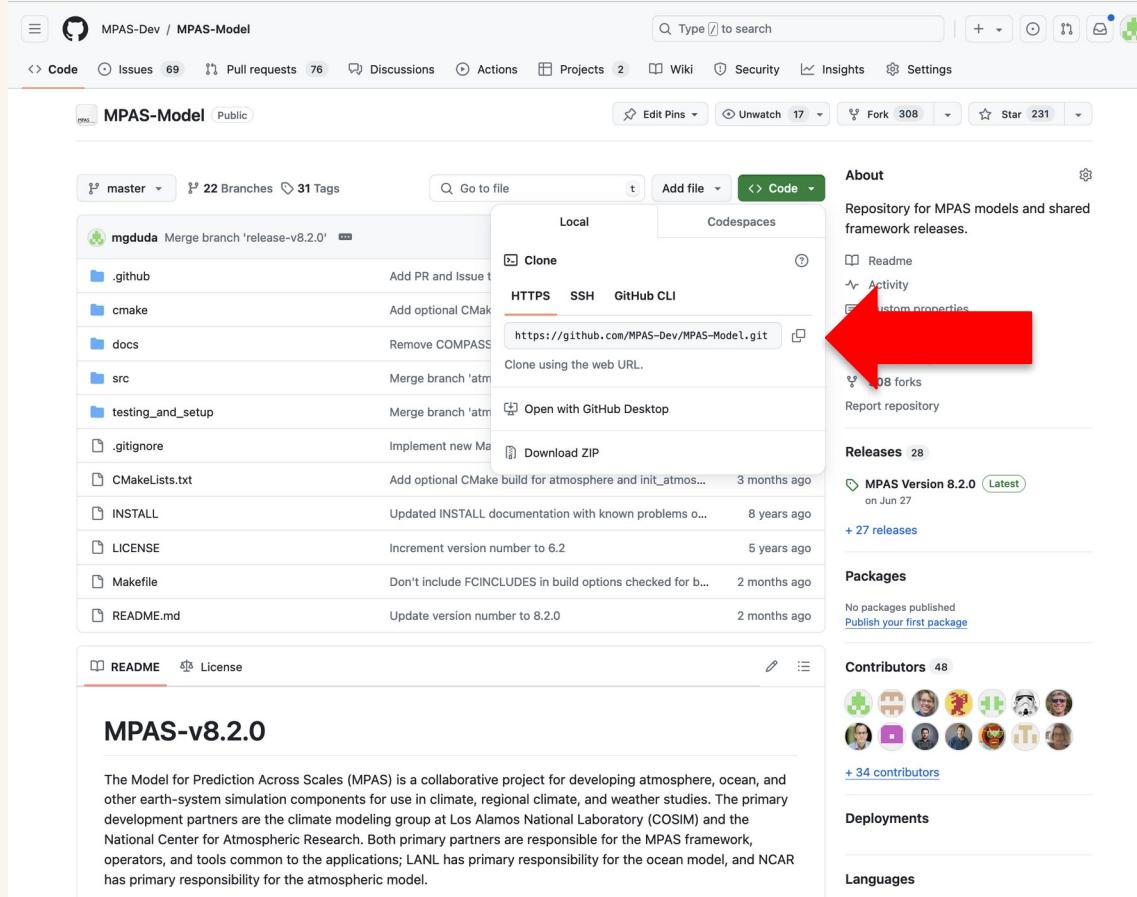
Downloading a `.tar.gz` file of a particular release of the MPAS code certainly works, but it has several disadvantages:

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>



The preferred method of obtaining code

From the command-line, the following should be sufficient:

```
git clone https://github.com/MPAS-Dev/MPAS-Model.git
```

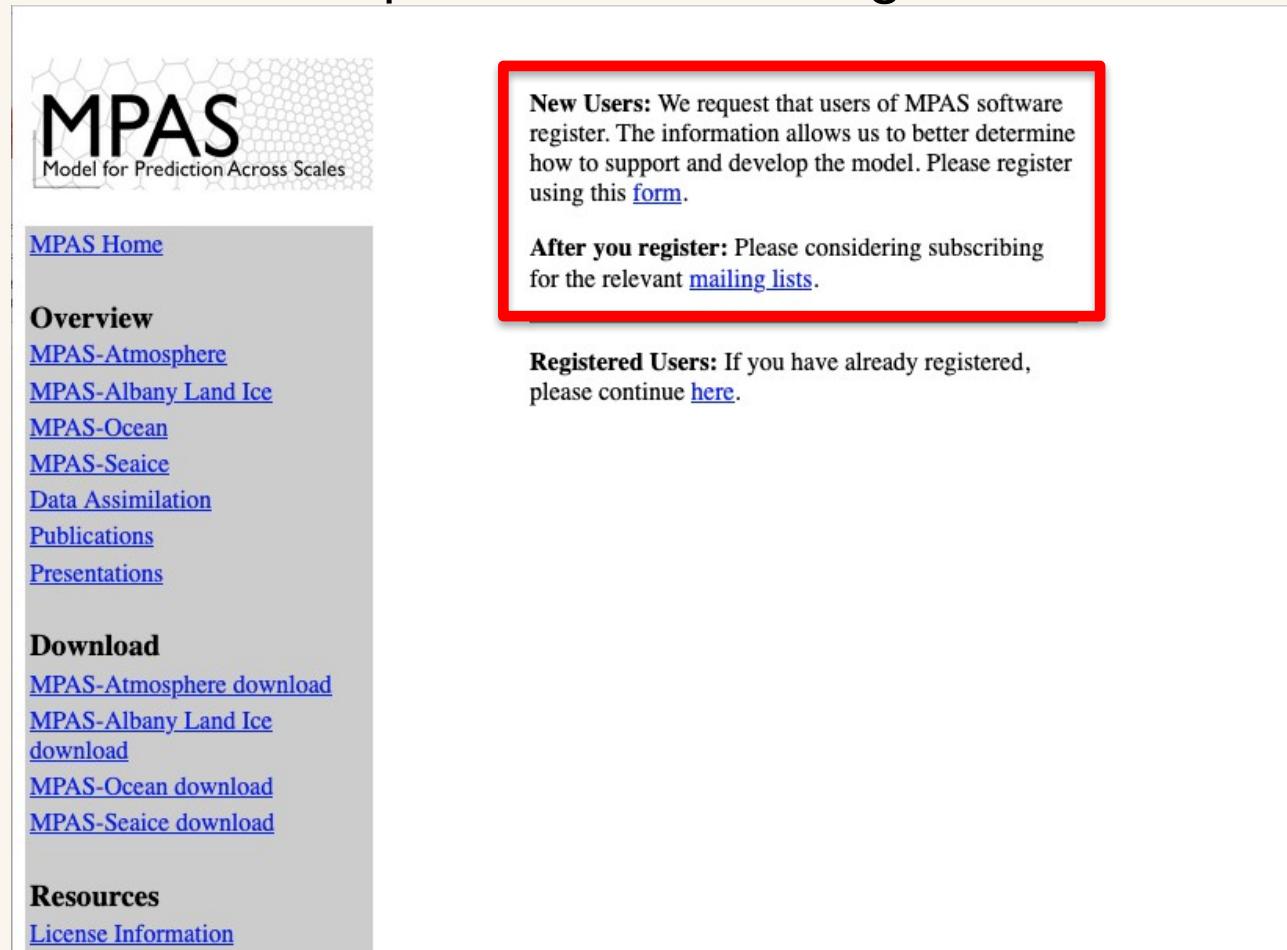
Cloning the repository should take about 10 seconds or less...

- *(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.
```

After obtaining the code for the first time

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



The screenshot shows the MPAS website's registration information. On the left, there is a sidebar with links for MPAS Home, Overview, Download, and Resources. The main content area contains two sections: one for new users and one for registered users. Both sections include a red border.

New Users: We request that users of MPAS software register. The information allows us to better determine how to support and develop the model. Please register using this [form](#).

After you register: Please consider subscribing for the relevant [mailing lists](#).

Registered Users: If you have already registered, please continue [here](#).

Preliminary requirements

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

Preliminary requirements: the easiest route

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

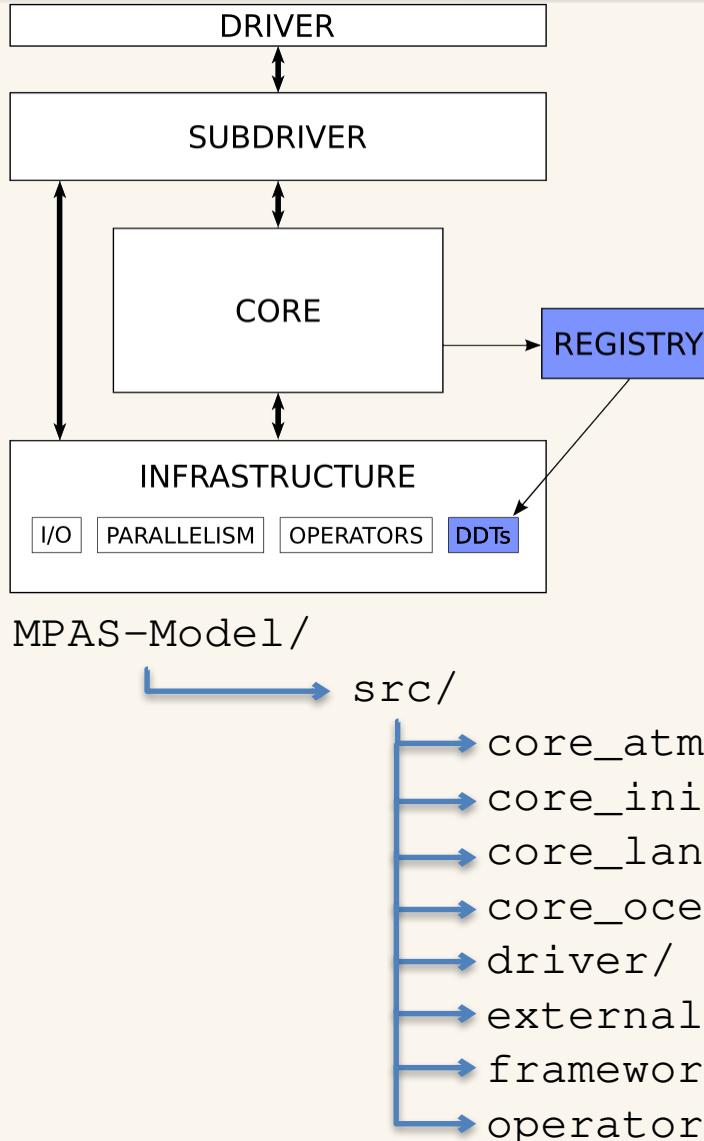
To make this easier, we've prepared a shell script that may be helpful:

```
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

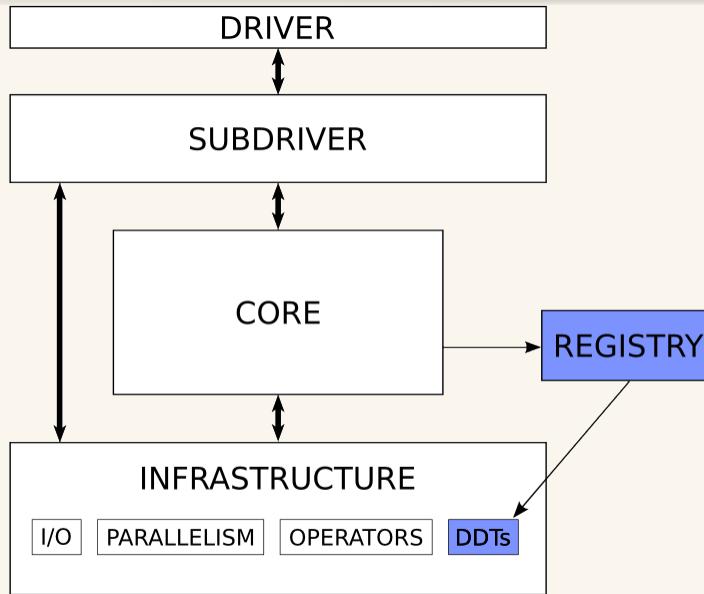
Model Organization



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)

Model Organization



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

Compiling MPAS

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

gnu

intel

nvhpc

llvm

... plus a few others...

For MPAS-Atmosphere, core may be

atmosphere

init_atmosphere

<options> can be any of

DEBUG=true

AUTOCLEAN=true

PRECISION=double

OPENMP=true

Compiling MPAS

Typical build of both the init_atmosphere and atmosphere cores involves:

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

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

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

By default, MPAS cores are built with single-precision

MPAS-Atmosphere can be built in double precision

- Add PRECISION=double to build commands for double-precision executables
- Execution time is slower compared with single-precision
- Output files approximately twice as large

Beginning with MPAS v8.2.0,
an intermediate “clean” step is
no longer needed!

Compiling MPAS v8.2.0+

Typical build of both the init_atmosphere and atmosphere cores involves:

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

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

By default, MPAS cores are built with single-precision floating-point variables

MPAS-Atmosphere can be built in double precision

- Add PRECISION=double to build commands for double-precision executables
- Execution time is slower compared with single-precision
- Output files approximately twice 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
- Ensure that you have modern Fortran, C, and C++ compilers
- 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