# MPAS-A System Requirements and Installation

Michael G. Duda
NSF 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.

# Motivation

Can I run MPAS-A on my computer?

Left: A Cray-1 supercomputer.

(Trivia: In 1977, the National Center for Atmospheric Research acquired a Cray-1, serial number 3.)

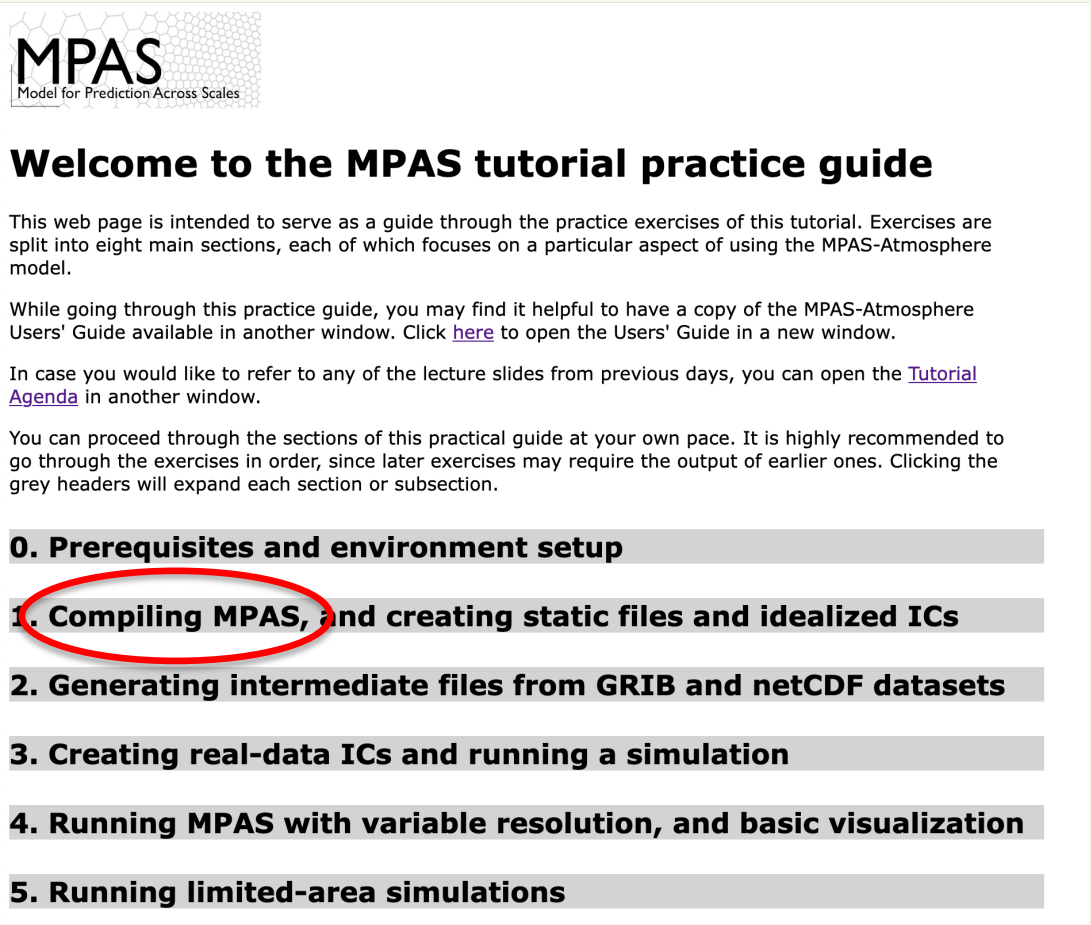


If so, how do I install MPAS-A on it?

# Motivation

MPAS-Atmosphere is distributed as *source code*

- No pre-compiled model or binary distribution!



MPAS
Model for Prediction Across Scales

**Welcome to the MPAS tutorial practice guide**

This web page is intended to serve as a guide through the practice exercises of this tutorial. Exercises are split into eight main sections, each of which focuses on a particular aspect of using the MPAS-Atmosphere model.

While going through this practice guide, you may find it helpful to have a copy of the MPAS-Atmosphere Users' Guide available in another window. Click here to open the Users' Guide in a new window.

In case you would like to refer to any of the lecture slides from previous days, you can open the Tutorial Agenda in another window.

You can proceed through the sections of this practical guide at your own pace. It is highly recommended to go through the exercises in order, since later exercises may require the output of earlier ones. Clicking the grey headers will expand each section or subsection.

**0. Prerequisites and environment setup**

**1. Compiling MPAS, and creating static files and idealized ICs**

**2. Generating intermediate files from GRIB and netCDF datasets**

**3. Creating real-data ICs and running a simulation**

**4. Running MPAS with variable resolution, and basic visualization**

**5. Running limited-area simulations**

The first practical exercise will involve compiling your own copy of MPAS-A on Derecho.

Compiling MPAS-A is probably also the first step on your own machine!

# System Requirements

What system attributes need to be considered?

## Hardware

- Processor(s)
- Memory (RAM)

## Software

- Operating system
- Compilers
- Libraries

Image credit: Irid Escent, CC BY-SA 2.0, via Wikimedia Commons

# Hardware Requirements: Processors

MPAS-A itself does not require any particular kind of processors

Consider this (not actual MPAS-A) Fortran code:

```
! Compute temperature in K from temperature in deg. C
do i = 1, n
    tk(i) = tc(i) + 273.15
end do
```

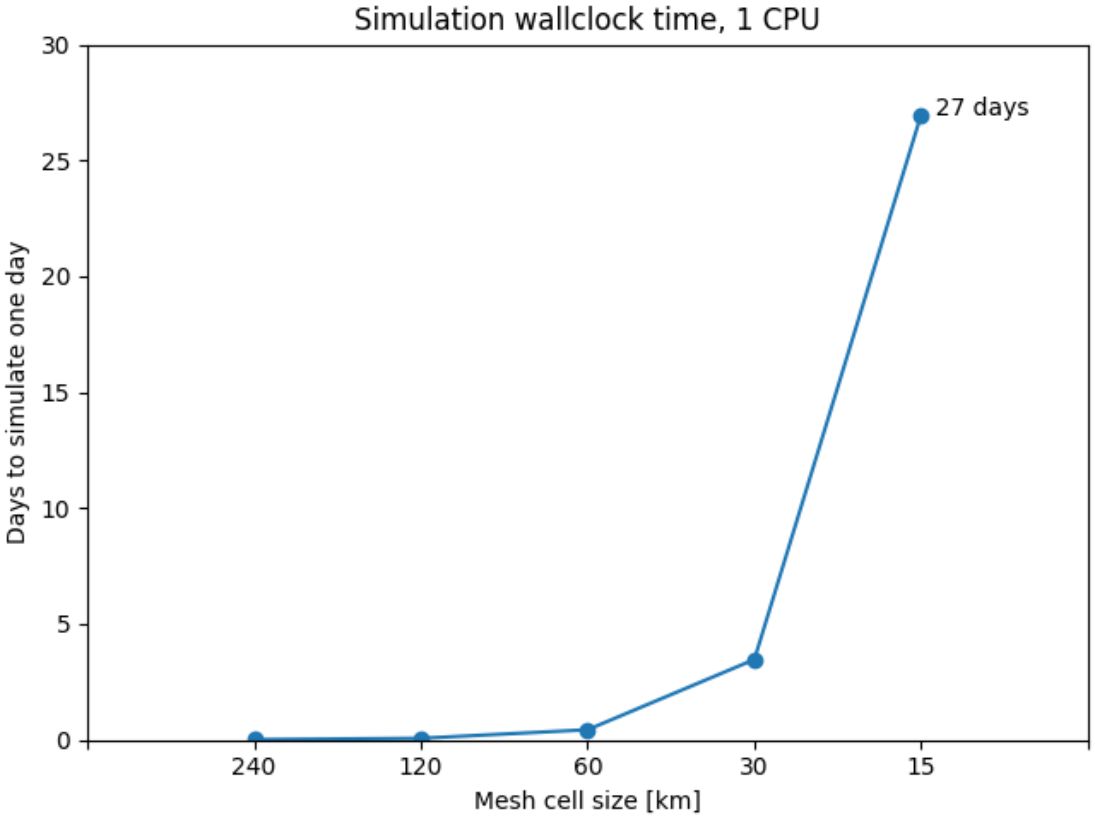
Nothing here is processor-specific!

Any of the following processor architectures should be fine:

- ARM
- x86_64
- POWER
- RISC-V
- Various GPU architectures!

# Hardware Requirements: Processors

If you are patient enough, *a single processor is all that you need.*




Left: Time to simulate one day on a global mesh for different mesh sizes using a single processor

In practice, using multiple processors is critical for achieving useful simulation rates

MPAS
Model for Prediction Across Scales

# Hardware Requirements: Memory

The memory requirement is a function of:

1. The number of grid columns in the mesh (simulation domain)
2. Choice of physics schemes / suite
3. The number of MPI tasks
4. The number of I/O tasks

A reasonable starting point:

0.4 MB / grid column

(for the default MPAS-A configuration with 55 layers, 'mesoscale_reference' physics suite, single precision).

MPAS
Model for Prediction Across Scales

# Software requirements

Generally, a UNIX-like OS is needed

- *E.g.*, Linux, *BSD, macOS, AIX
- Also need make, grep, awk, sed, cat, ranlib/libtool, **git**, etc.

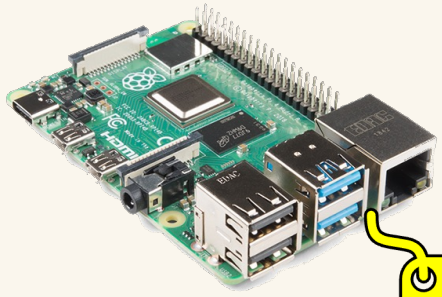
Fortran 2008, C, and C++ compilers

- *E.g.*, GCC, Intel oneAPI, NVIDIA NVHPC, Cray, NAG

Libraries

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

# Example systems that run MPAS-A



$35


$35M

$5000


$200M
(?)

# Obtaining MPAS-Atmosphere source code

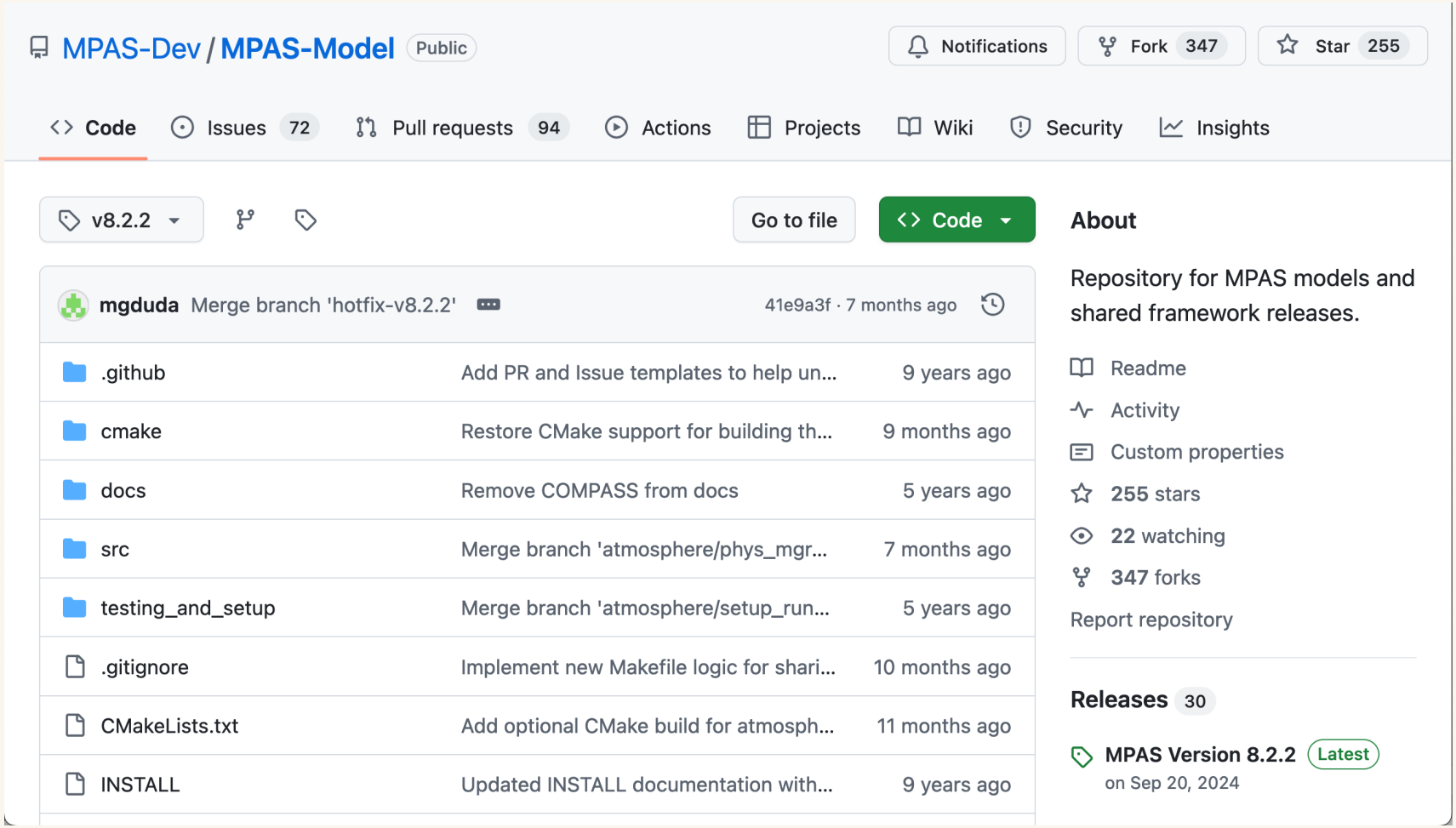
Working under the assumption that we have a system that meets the basic requirements, we can then ask:

How do I get a copy of the source code?

*If you are working on a machine with the git command-line tools and internet access, this can be done in under 10 seconds!*

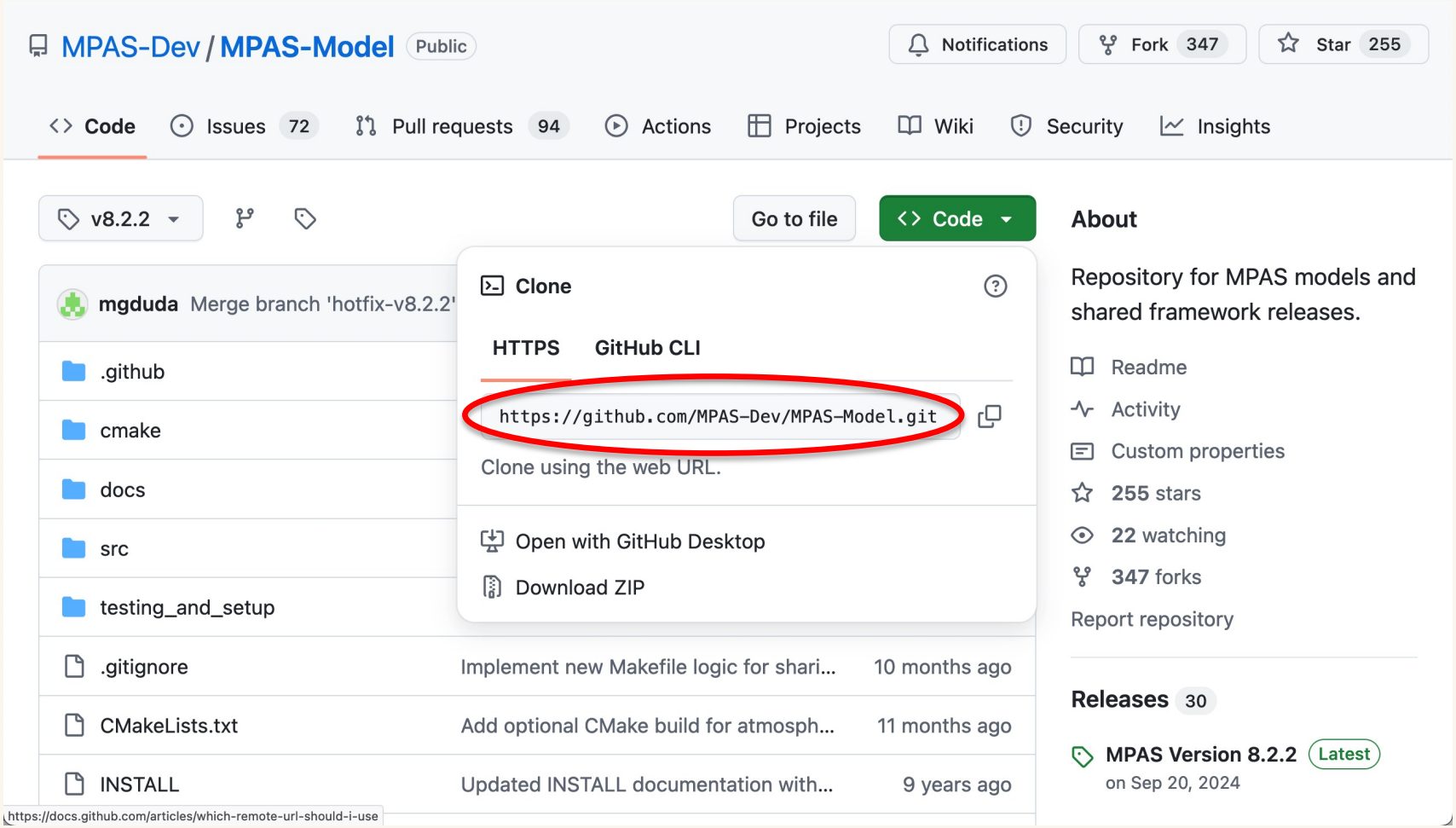
# Obtaining MPAS-Atmosphere source code

The MPAS-Atmosphere code is developed on and distributed through a GitHub repository: https://github.com/MPAS-Dev/MPAS-Model



# Obtaining MPAS-Atmosphere source code

The MPAS-Atmosphere code is developed on and distributed through a GitHub repository: https://github.com/MPAS-Dev/MPAS-Model



# Obtaining MPAS-Atmosphere source 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 5.65 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.
```

# Optional: Registration and Mailing Lists

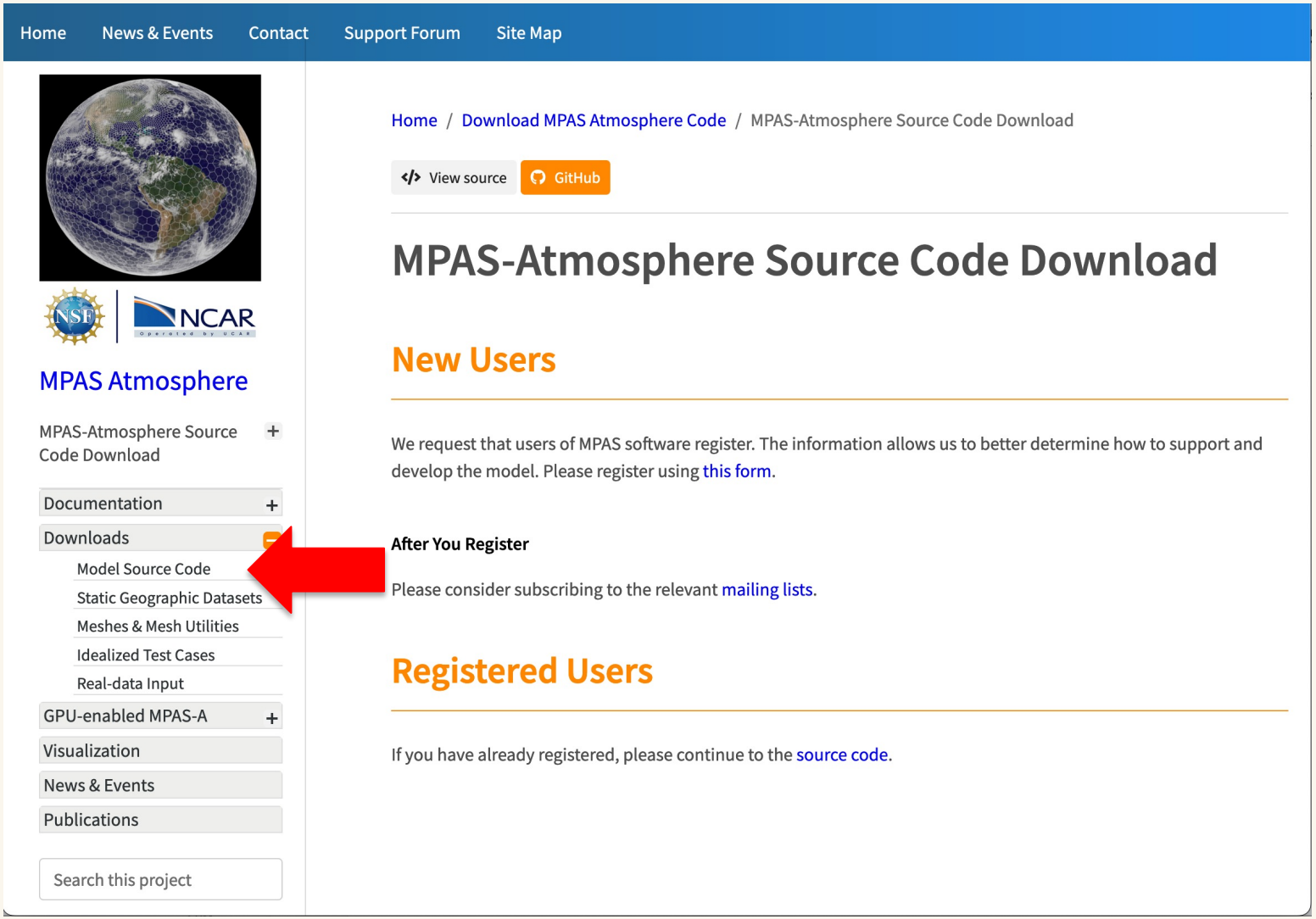
As a new user, you may like to:

- **Register** – this is completely optional, but helpful for us to gauge how widely used MPAS-A is, justify development and support resources, etc.
- Join the **mpas-atmosphere-users mailing list** – Very low traffic, mostly announcements of tutorials, workshops, and code releases

Links for the above are available from the MPAS-A website:

https://www2.mmm.ucar.edu/projects/mpas/site/

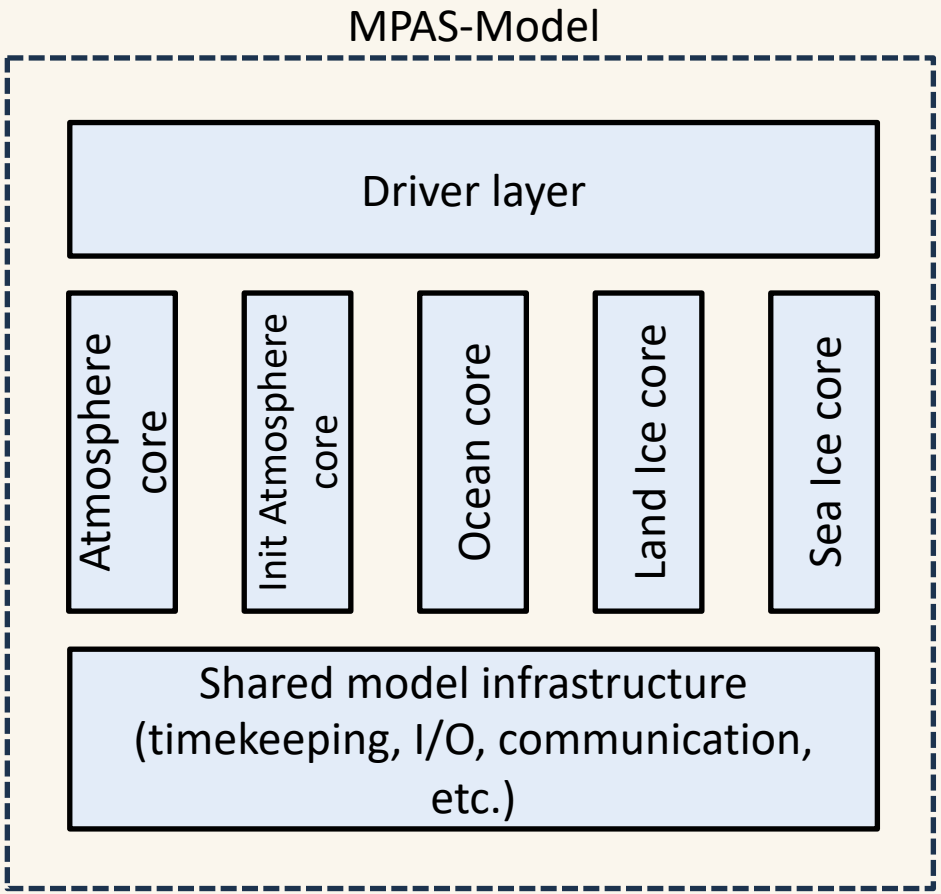
# Optional: Registration and Mailing Lists



# Model Organization

*Cloning* the MPAS-Model repository provides all MPAS models, not just MPAS-Atmosphere

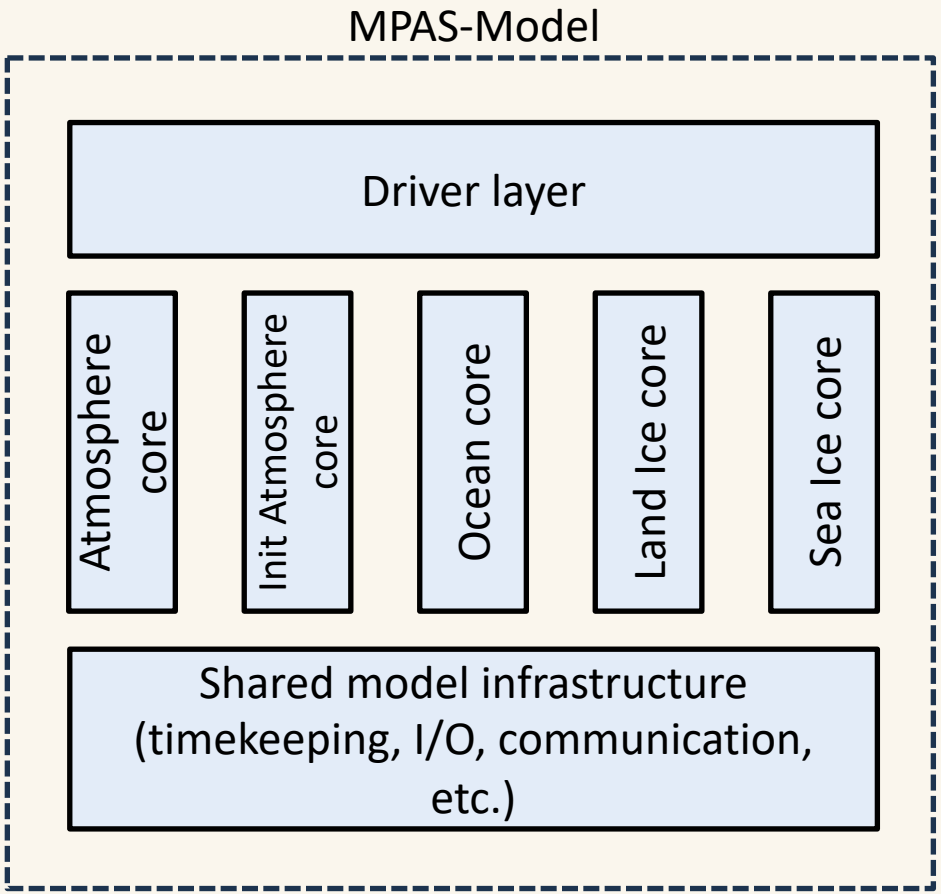
- All models share a common set of infrastructure modules
- Each MPAS model is implemented as a "core" that lives in its own sub-directory
- The MPAS models *are not coupled in this repository!*
- You must select which "core" to compile



# 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



MPAS
Model for Prediction Across Scales

# Compiling MPAS

There is no "configuration" step for MPAS

- 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
OPENACC=true
```

# Compiling MPAS

Historically, building both the init_atmosphere and atmosphere cores involved:

`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`*)*

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

By default, MPAS cores are built with single-precisi

MPAS-Atmosphere can be built in double precision

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

# Compiling MPAS v8.2.0+

Typical build of both the init_atmosphere and atmosphere cores involves:

`make -j8 gnu CORE=init_atmosphere` *(build* `init_atmosphere_model`*)*

`make -j8 gnu CORE=atmosphere` *(build* `atmosphere_model`*)*

The `-j8` option causes make to use up to 8 jobs (simultaneous commands) to compile the code and can reduce build times; values other than 8 can be used.

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-precisoin 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)
- Obtain the MPAS-Model source code with *git clone*
- Compile the *init_atmosphere* and *atmosphere* cores