Running SHiELD with GFDL's FMS full coupler infrastructure

Joseph Mouallem
Cooperative Institute for Modeling the Earth System,
Program in Oceanic and Atmospheric Sciences, Princeton University

November 2024

GFDL Weather and Climate Dynamics Division
Technical Memorandum GFDL202402

Abstract

This document outlines the technical requirements and upgrades implemented across various code repositories to enable the FV3-based atmospheric model, SHiELD, to run with the FMS full coupler infrastructure. It provides an overview of the build environment and the simplified SHiELD coupler. It details the necessary modifications for transitioning to the FMS full coupler infrastructure and offers recommendations for future work to ensure the seamless integration of SHiELD into GFDL's modeling suite.

Note on nomenclature

In what follows, words in `verbatim` are associated with official GitHub repositories and code files under NOAA's official GitHub account https://github.com/NOAA-GFDL. For example, `FMS/mpp/mpp.F90` will stand for the file `mpp.F90` located at https://github.com/NOAA-GFDL/FMS/blob/main/mpp/mpp.F90

Code version

In what follows, when a reference to GitHub file or directory is made, we refer to the following repository tags/branches:

Repository	**Tag**
https://github.com/NOAA-GFDL/SHiELD_build/	FV3-202411-public
https://github.com/NOAA-GFDL/GFDL_atmos_cubed_sphere	FV3-202411-public
https://github.com/NOAA-GFDL/atmos_drivers	FV3-202411-public
https://github.com/NOAA-GFDL/FMS	2024.03
https://github.com/NOAA-GFDL/FMScoupler	2024.03.01
https://github.com/NOAA-GFDL/SHiELD_physics	FV3-202411-public
https://github.com/NOAA-GFDL/ocean_null	main
https://github.com/NOAA-GFDL/ice_null	main
https://github.com/NOAA-GFDL/land_null	main
https://github.com/NOAA-GFDL/ice_param	2024.02

Table 1: Relevant repositories and their associated tags. Each hyperlink leads directly to the NOAA's official GitHub repository.

1 Introduction

The System for High-resolution prediction on Earth-to-Local Domain (SHiELD)[1] is a high-resolution atmospheric model developed at GFDL/NOAA using the FV3 dynamical core. Unlike more complex GFDL models, such as AM4 and SPEAR, SHiELD was initially built using a simplified framework that relies on a customized simple coupler. The original SHiELD workflow involved a separate main driver under the `FMScoupler/SHiELD/` directory, which coordinated the interaction between atmospheric components and other model elements, without fully integrating the standard GFDL coupler infrastructure.

To advance SHiELD's capabilities and enhance its compatibility with GFDL's modeling suite, the model infrastructure has been upgraded to utilize the FMS full coupler. This transition aligns SHiELD with other GFDL models, enabling more comprehensive and flexible coupling of atmospheric components with ocean, ice, wave and land models. The integration process involved significant modifications to the build system and driver components to support the FMS full coupler infrastructure.

This document outlines the technical details of the infrastructure upgrade, describing the changes made to SHiELD's build and driver systems. It provides a short guide to download and compile the updated SHiELD model using the FMS full coupler and offers an outlook for future developments to further integrate SHiELD into GFDL's coupled modeling framework.

2 Original infrastructure

In this section, we will go over SHiELD's original infrastructure and workflow. The changes to this workflow will be shown in the next section. The user or developer can check out and build the code executable from the `SHiELD_build` directory. To check out SHiELD, simply run

```
$ ./CHECKOUT_code shield
```

this command will check out the following repositories: `FMScoupler`, `FMS`, `GFDL_atmos_cubed_sphere`, `atmos_drivers`, `SHiELD_physics`. To compile SHiELD, follow the instructions in the read-me file, which will eventually lead to changing directory to `SHiELD_build/Build` then run:

```
$ ./COMPILE shield
```

with the appropriate arguments. By default, the compilation script `SHiELD_build/Build/COMPILE` will compile with `nh prod 32bit intel noyaml nopic`. This process employs `mkmf` to build libraries, namely FMS library `libfms.a`, FV3 library `libfv3.a`, physics library `libgfs.a` as well as NCEP libraries `libbacio.a, libsp_d.a, libw3emc.a, libw3nco_d.a`. These libraries are then linked together through the FMScoupler to generate the final executable. The corresponding code for this process can be found in `SHiELD_build/Build/BUILDlibfms`, `SHiELD_build/Build/BUILDnceplibs` and `SHiELD_build/Build/mk_scripts/mk_makefile`. To link the necessary files from each repository, the list of paths can be found at `SHiELD_build/Build/mk_scripts/mk_paths`. Considering the original SHiELD infrastructure configuration, the files needed to compile the FV3 and physics libraries are:

```
list_paths -o ${BUILD_ROOT}/Build/exec/${CONFIG}_${HYDRO}.${COMP}.${BIT}.${COMPILER}/pathnames_gfs \
    SHiELD_physics/gsmphys/  \
    SHiELD_physics/GFS_layer/ \
    SHiELD_physics/IPD_layer/
```

```
list_paths -o ${BUILD_ROOT}/Build/exec/${CONFIG}_${HYDRO}.${COMP}.${BIT}.${COMPILER}/pathnames_fv3 \
    SHiELD_physics/FV3GFS/ \
    atmos_drivers/SHiELD/atmos_model.F90 \
    GFDL_atmos_cubed_sphere/driver/SHiELD/atmosphere.F90 \
    GFDL_atmos_cubed_sphere/tools/ \
    GFDL_atmos_cubed_sphere/model/

list_paths -o ${BUILD_ROOT}/Build/exec/${CONFIG}_${HYDRO}.${COMP}.${BIT}.${COMPILER}/pathnames_driver \
    FMSCoupler/SHiELD/coupler_main.F90
```

Figure 1 illustrates the repositories required for this workflow. As mentioned previously, SHiELD's workflow has been developed independently compared to other GFDL models. Specifically, there is a dedicated main driver located in `FMScoupler/SHiELD/coupler_main.F90` that drives the model execution. As shown in the code snippet above, this driver is linked to `atmos_drivers/SHiELD/atmos_model.F90` and subsequently to `GFDL_atmos_cubed_sphere/driver/SHiELD/atmosphere.F90`

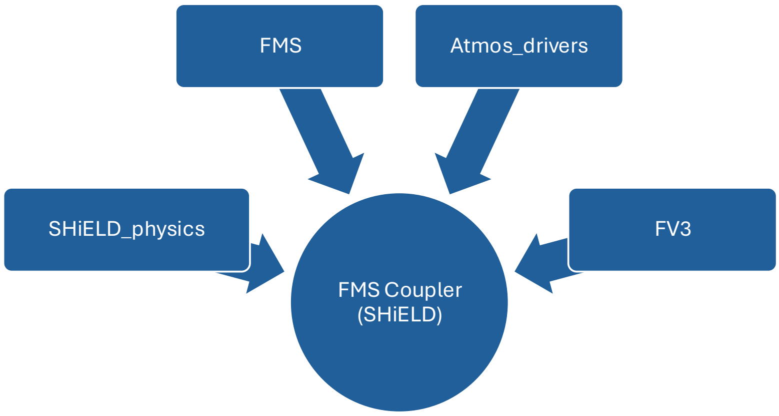

Figure 1: Schematic of SHiELD's original infrastructure.

3 Transition to the FMS full coupler

Transitioning the workflow to utilize the full coupler infrastructure involves changing the main program driver from

`FMScoupler/SHiELD/coupler_main.F90`

to

`FMScoupler/full/coupler_main.F90`

The schematic for the updated infrastructure is shown in figure 2. The main program `FMScoupler/full/coupler_main.F90` will require entries for different model components, including the atmosphere, ocean, ice and land (with the wave component planned for future integration). Since the ocean, ice and land models are run through `SHiELD_physics`, we use null modules of the ocean, ice and land: `ocean_null`, `ice_null` and `land_null` to represent these components in the full coupler.

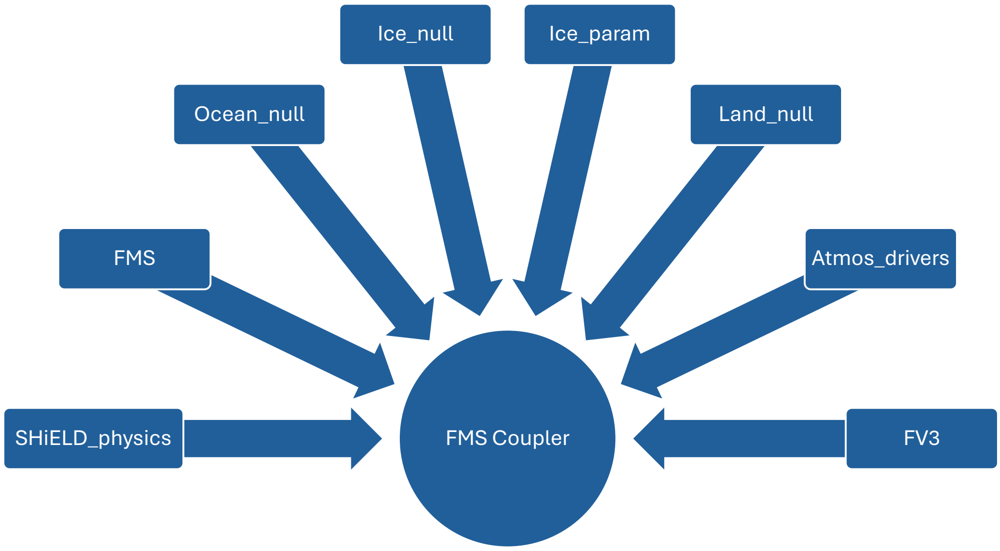

Figure 2: Schematic of SHiELD's upgraded infrastructure utilizing the FMS full coupler

3.1 Upgrades to the build system

We have updated the build system following the original logic to accommodate running the full coupler infrastructure. Similar to the original configuration, the user or developer can check out and compile the code for the full coupler infrastructure using the same scripts `CHECKOUT_code` and `COMPILE` but using `shieldfull` as the script argument instead of `shield`.

The check out script will clone all repositories required for a successful compilation including `FMScoupler`, `FMS`, `GFDL_atmos_cubed_sphere`, `atmos_drivers`, `SHiELD_physics`, `ocean_null`, `land_null`, `ice_null` and `ice_param`. While the libraries build workflow remains unchanged, the list of files and directories to compile will differ. Below are the changes added to `SHiELD_build/Build/mk_scripts/mk_paths`:

```
list_paths -o ${BUILD_ROOT}/Build/exec/${CONFIG}_${HYDRO}.${COMP}.${BIT}.${COMPILER}/pathnames_gfs \
    SHiELD_physics/gsmphys/  \
    SHiELD_physics/GFS_layer/ \
    SHiELD_physics/IPD_layer/

list_paths -o ${BUILD_ROOT}/Build/exec/${CONFIG}_${HYDRO}.${COMP}.${BIT}.${COMPILER}/pathnames_fv3 \
    SHiELD_physics/FV3GFS/ \
    GFDL_atmos_cubed_sphere/tools/ \
    GFDL_atmos_cubed_sphere/model/ \
    SHiELD_physics/atmos_shared/  \
    GFDL_atmos_cubed_sphere/driver/SHiELDFULL/atmosphere.F90

list_paths -o ${BUILD_ROOT}/Build/exec/${CONFIG}_${HYDRO}.${COMP}.${BIT}.${COMPILER}/pathnames_driver \
    ocean_null/ \
    ice_null/ \
    ice_param/ \
    land_null/ \
    atmos_drivers/SHiELDFULL/ \
    FMSCoupler/full/ \
    FMSCoupler/shared/
```

As shown here, the new driver and its corresponding files for the flux exchange and exchange grid are linked under `FMSCoupler/full/` and `FMSCoupler/shared/`. The updated atmosphere driver is located at atmos_drivers/SHiELDFULL/atmos_model.F90 and calls the new GFDL_atmos_cubed_sphere/driver/SHiELDFULL/atmosphere.F90. The driver updates are discussed in the next section. Additionally, the ocean, ice, land null components are also linked here to meet the full coupler requirements. It is worth noting that mixed precision mode is supported with this configuration, allowing FV3 to be built in either 32-bit or 64-bit precision, while the other model components remain in 64-bit precision.

3.2 Upgrades to the drivers

A new driver file is added at atmos_drivers/SHiELDFULL/atmos_model.F90. This file is an upgrade of atmos_drivers/SHiELD/atmos_model.F90 following the subroutine name and logic from atmos_drivers/coupled/atmos_model.F90. Below are some of the main changes made in SHiELDFULL/atmos_model.F90 starting from SHiELD/atmos_model.F90:

- Atmosphere fluxes and lowest atmosphere layer variables at the surface (for the flux exchange routines) are added under atmos_data_type. For example, we add the net short wave and long wave fluxes at the surface flux_sw, flux_lw, the zonal wind component at lowest model level u_bot, the surface pressure p_surf and many more that we don't list here for brevity.
- We add new data types required by the full coupler routines:
  - land_ice_atmos_boundary_type
  - land_atmos_boundary_type
  - ice_atmos_boundary_type
  - surf_diff_type
- New subroutines have been added, and the names of some existing routines have been updated; all routines are public unless specified otherwise. Additionally, all routines perform specific functionalities, except where indicated as no-op (no operation):
  - update_atmos_radiation_physics changed to update_atmos_model_radiation
  - update_atmos_model_up (no-op)
  - update_atmos_model_down (no-op)
  - lnd_ice_atm_bnd_type_chksum
  - atmos_data_type_chksum
  - lnd_atm_bnd_type_chksum
  - ice_atm_bnd_type_chksum
  - atm_stock_pe
  - alloc_atmos_data_surfdiff_type (local)

Additional Notes (at the time of this write-up):

- All SHiELD's physics are invoked in update_atmos_model_radiation which continues to utilize the IPD infrastructure to store physics tendencies.

- Physics subroutines from the original configuration `update_atmos_pre_radiation`, `update_atmos_radiation` and `update_atmos_physics` have been consolidated into the new routine `update_atmos_model_radiation`.
- Local routines `alloc_atmos_data_type`, `dealloc_atmos_data_type` have been updated to include the new variables previously mentioned.

3.3 Extra files

Additional atmosphere physics routines required by the full coupler are put under `SHiELD_physics/atmos_shared` to replicate the structure of GFDL's AMx `atmos_phys/`, although they are not currently executed by SHiELD.

4 Runscript modifications - What users should know

When running SHiELD with the FMS full coupler infrastructure, all users are required to make changes to the coupler namelist `&coupler_nml`; however, these changes do not affect their workflow or the code performance in any way. The changes require commenting out the ocean timestep dt_ocean and adding the following additional parameters as shown below. Setting `do_flux=.F.` will bypass all the exchange grid and flux initialization and calculations in the full coupler, thereby relieving this framework from unnecessary computational overhead and the requirement for additional mosaic input files. All SHiELD configurations including global, global-nest, regional, and idealized runs are bit-for-bit reproducible. The new namelist should look like:

```
&coupler_nml
  months = $months
  days   = $days
  hours = $hours
  dt_atmos = $dt_atmos
  current_date =  $curr_date
  calendar = 'julian'
  atmos_nthreads = $nthreads
  use_hyper_thread = $hyperthread

  !dt_ocean = $dt_atmos    ! commented
  ice_npes = -1            ! added
  land_npes = -1           ! added
  do_ocean=.False.         ! added
  dt_cpld = $dt_atmos      ! added
  do_flux=.False.          ! added
  do_land=.False.          ! added
  do_ice=.False.           ! added
\
```

`do_ocean`, `do_land`, `do_ice` are turned off to bypass all corresponding calculations for the ocean, land, and ice components given the usage of their null modules. `dt_cpld` is the coupling timestep and is set to the atmosphere timestep `dt_atmos`. `do_flux` was previously discussed and will bypass the exchange grid and flux related operations. `ice_npes` and `land_npes` correspond to the number of processors allocated for the ice and land components; by default, they are set to zero and should be set here to -1 to allocate all available processors to the atmosphere.

5 Conclusions

The transition to the FMS full coupler infrastructure represents a significant advancement in integrating SHiELD within the broader GFDL modeling framework. This upgrade streamlines the process of incorporating multiple model components, such as atmosphere, ocean, ice, and land, under a unified coupling architecture, thus enhancing the modularity and flexibility of the model. The adoption of the full coupler not only simplifies the workflow but also ensures compatibility with other complex models within the GFDL suite, facilitating future collaborative development efforts.

The modifications to the build system and driver files, along with the introduction of new data types and subroutines, were essential to accommodate the full coupler requirements. These changes have been carefully implemented to maintain the model's reproducibility, performance, and stability while enabling new capabilities that could be implemented for SHiELD such as using the current infrastructure and exchange grid to be coupled to other model components.

Looking ahead, the ongoing refinement and comprehensive testing of these modifications will be essential to fully leverage the advantages offered by the full coupler. Subsequent research should prioritize the enhancement of the model's capabilities, broadening its applicability to a diverse array of configurations, and integrating it with additional model components, including ocean, ice, land, and wave models, to augment SHiELD's functionalities. This advancement establishes a solid foundation for the progression of SHiELD's role in high-resolution atmospheric modeling and significantly contributes to the overall enhancement of GFDL's modeling infrastructure.

Acknowledgment

Thanks to Rusty Benson, Lauren Chilutti and Lucas Harris for many insightful discussions and valuable assistance during this implementation. Thanks to Niki Zadeh, Kai-Yuan Cheng and Lauren Chilutti for their thorough review and suggestions that made this technical note clearer.

References

[1] L. Harris, L. Zhou, S. J. Lin, J. H. Chen, X. Chen, K. Gao, M. Morin, S. Rees, Y. Sun, M. Tong, B. Xiang, M. Bender, R. Benson, K. Y. Cheng, S. Clark, O. D. Elbert, A. Hazelton, J. J. Huff, A. Kaltenbaugh, Z. Liang, T. Marchok, H. H. Shin, and W. Stern, "Gfdl shield: A unified system for weather-to-seasonal prediction," *Journal of Advances in Modeling Earth Systems*, vol. 12, pp. 1–25, 2020.