

MITgcm.jl: a Julia Interface to the MITgcm

Gaël Forget  ¹

¹ Massachusetts Institute of Technology, Cambridge, MA, USA

DOI: [10.21105/joss.06710](https://doi.org/10.21105/joss.06710)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Anjali Sandip](#)  

Reviewers:

- [@simone-silvestri](#)
- [@christophernhill](#)
- [@Datseris](#)
- [@swilliamson7](#)

Submitted: 03 April 2024

Published: 21 October 2024

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

General circulation models are used to study climate, ocean and atmosphere dynamics, biogeochemistry, ecology, and more. The MITgcm, written in Fortran, is one of the most widely-used models of this class. We present MITgcm.jl, a two-way interface to MITgcm written in Julia that can be used to not only analyze model results but also drive model simulations. With MITgcm.jl users can setup, build, and launch MITgcm without having to know shell scripting or having to edit text files manually. MITgcm.jl provides support in Julia for the various input and output formats used in MITgcm. It implements the ClimateModels.jl interface, and opens up a whole new way of using MITgcm interactively from Jupyter and Pluto notebooks. MITgcm.jl in turn brings full-featured, reliable ocean modeling to Julia.

Statement of Need

The cutting-edge of climate modeling, and much of its legacy, is based on numerical models written in compiled languages like Fortran or C. The MIT general circulation model (MITgcm) for example now runs globally at the kilometer scale to provide an unprecedented view of ocean dynamics (Fig. 1, Campin et al. (2004), Marshall et al. (1997), Gallmeier et al. (2023)). With its unrivaled adjoint modeling capabilities, MITgcm is also the computational engine that powers the ECCO ocean reanalysis, a widely-used data-assimilating product in climate science (Fig. 1, Heimbach et al. (2002), G. Forget et al. (2015), Gaël Forget (2024a)). MITgcm additionally provides unique modeling capabilities for ocean biogeochemistry, ecology, and optics (Dutkiewicz et al. (2015), Gaël Forget (2019)). While a new generation of models, written in languages like C++ and Julia, is poised to better exploit GPUs (Ramadhan et al. (2020), Wu & Forget (2022), E3SM Project (2024)), Fortran-based models are expected to remain popular on other computer architectures for the foreseeable future. They also provide a crucial reference point to evaluate next generation models.

Models like MITgcm benefit from decades of accumulated expertise, careful inspection, and continuous integration. Running such a model or exploiting its results does not require knowing the computer language being used internally. However, Fortran-based models can sometime appear complicated or inconvenient to operate due to technical hurdles like having to use a compiler directly, to edit text files manually, or to deal with shell-scripting. Fortunately, such issues are easily alleviated by providing a user-friendly interface (e.g., Fig. 2) written in a high-level language (e.g., Julia) to interact with climate models written in lower-level languages as done here for MITgcm.

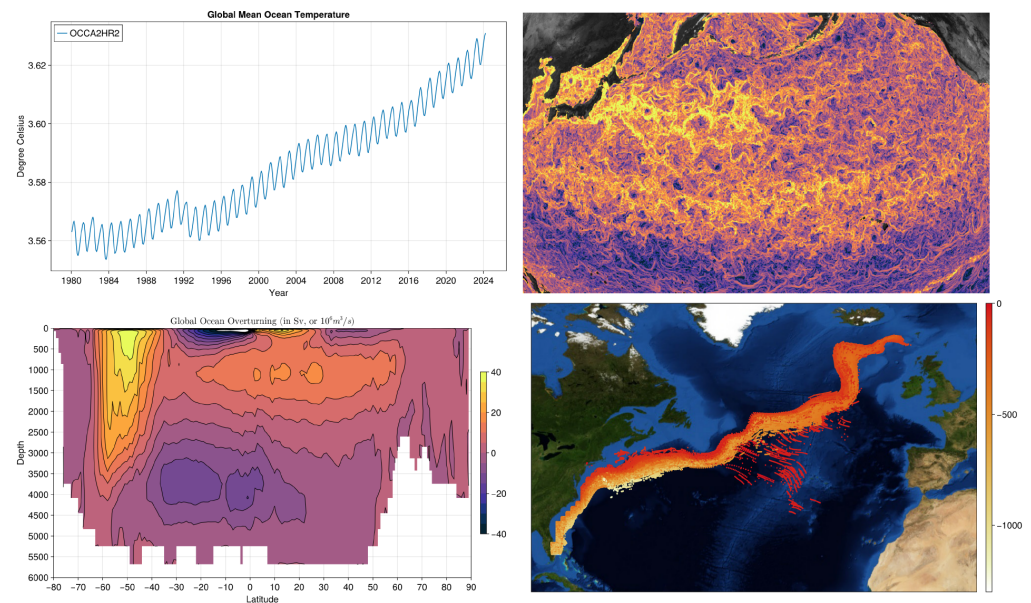


Figure 1: Examples of MITgcm output created, read, processed, plotted, and analyzed via MITgcm.jl in Julia. Top left : global mean ocean warming over 1980-2023 as simulated in OCCA2HR2 using MITgcm.jl (Gaël Forget, 2024a). Bottom Left : climatological mean view of the global ocean conveyor belt (Rousselet et al., 2021) produced in Climatology.jl. Bottom right : particle tracking of seawater pathways along the Gulf Stream, computed in three dimensions by IndividualDisplacements.jl (Gaël Forget, 2021). Top Right : temperature fronts in a global MITgcm simulation, which ran on a 4km resolution Lat-Lon-Cap grid (G. Forget et al., 2015, p. Gallmeier2023), diagnosed and visualized in Julia using MeshArrays.jl.

MITgcm.jl can read the various types of output that MITgcm generates. This feature enables not only common analyses of model results (e.g., mapping and plotting), but also accurate computations of quantities such as ocean heat transport and global warming (Fig. 1, G. Forget et al. (2015), Gaël Forget & Ferreira (2019), Gaël Forget (2024a)). MITgcm.jl also makes it easy to deploy and run any configuration of MITgcm on laptops, HPC clusters, and in the cloud. MITgcm.jl interacts with MITgcm's run-time model parameters in Julia via ordered dictionaries. It can re-export them to the standard TOML file format, or to the native MITgcm format. Owing to this two-way interface, MITgcm can now be used from Jupyter or Pluto notebooks interactively via MITgcm.jl. The package includes a series of examples and tutorials that demonstrate the interface (e.g., Fig. 2).

In the code example below, MITgcm_config defines the MC data structure. The run command is equivalent to the sequence of `setup(MC)`; `build(MC)`; `launch(MC)`. And `readdir` returns a list of files found in the folder where MITgcm ran (a temporary folder by default).

```
using MITgcm
MC=MITgcm_config(configuration="tutorial_held_suarez_cs")
run(MC)
readdir(MC,"run")
```

We can then modify parameters in julia, call `write_all_namelist`s to update the MITgcm run-time parameter files accordingly, and rerun a new model simulation in the same folder. In the example below, we thus extend the simulation to 64 time steps.

```
MC.inputs[:main][:PARM03][:nTimeSteps]=64
write_all_namelist(MC.inputs,joinpath(MC,"run"))
launch(MC)
```

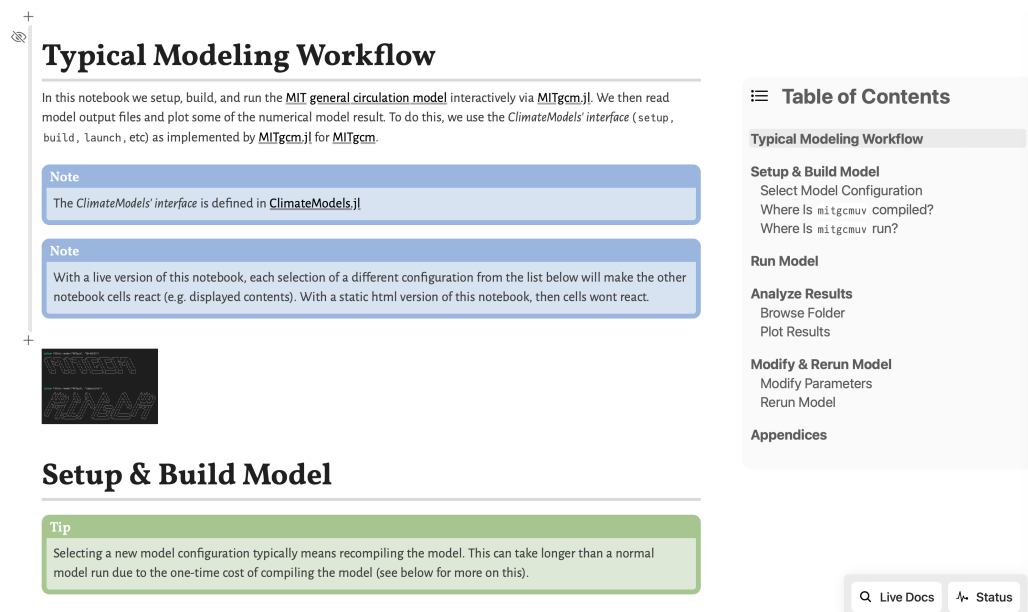


Figure 2: Notebook that operates MITgcm interactively, and let's user visualize model results without having to write code. Both Jupyter and Pluto notebooks are supported.

MITgcm.jl brings all of MITgcm's modeling capabilities to a new category of users, who may not be trained in Fortran or shell-scripting. Furthermore, MITgcm.jl implements the ClimateModels.jl interface (Gaël Forget, 2024b), which (1) streamlines the handling of file folders, (2) makes it easier to run and rerun simulations, and (3) supports an extensive lineup of complementary models written in various languages. MITgcm.jl can also be used to build integrated cyberinfrastructure solutions as demonstrated in Duckworth et al. (2023).

MITgcm.jl in turn provides the vast MITgcm user community with a bridge to new tools for machine learning, artificial intelligence, differential equations, visualization, etc from the Julia software stack. Examples include the use of MITgcm output in offline mode to estimate sea water pathways and ocean transports (Fig. 1, Gaël Forget (2021)), or simulate the behavior of marine ecosystems using an agent-based modeling approach (Wu & Forget, 2022).

Video Presentations

Video presentations listed below further demonstrate key features of MITgcm.jl.

- [MITgcm demo](#) that uses the ClimateModels.jl interface to run MITgcm via MITgcm.jl.
- [JuliaCon2021](#) ClimateModels.jl was presented at JuliaCon in 2021.
- [JuliaCon2023](#) ClimateModels.jl and MITgcm.jl were further presented at JuliaCon in 2023.

Acknowledgements

G.F. is supported by NASA Awards 80NSSC20K0796, 80NSSC23K0355, 80NSSC22K1697, 1676067, and 1686358, as well as Simons Foundation awards 329108 (SCOPE) and 549931 (CBIOMES).

References

- Campin, J. M., Adcroft, A., Hill, C., & Marshall, J. (2004). Conservation of properties in a free surface model. *Ocean Modelling*, 6, 221–244. [https://doi.org/10.1016/S1463-5003\(03\)00009-X](https://doi.org/10.1016/S1463-5003(03)00009-X)
- Duckworth, B., Dutkiewicz, S., Jahn, O., Hill, C. N., & Follett, C. L. (2023). DAR1 - an easy to use, single-box, configuration of a global scale biogeochemical ecosystem model. *bioRxiv*. <https://doi.org/10.1101/2023.11.23.568480>
- Dutkiewicz, S., Hickman, A. E., Jahn, O., Gregg, W. W., Mouw, C. B., & Follows, M. J. (2015). Capturing optically important constituents and properties in a marine biogeochemical and ecosystem model. *Biogeosciences*, 12(14), 4447–4481. <https://doi.org/10.5194/bg-12-4447-2015>
- E3SM Project. (2024). *Energy Exascale Earth System Model (E3SM)*. [Computer Software] <https://dx.doi.org/10.11578/E3SM/dc.20240301.3>. <https://doi.org/10.11578/E3SM/dc.20240301.3>
- Forget, Gaël. (2019). *CBIOMES-global: A global ocean state estimate for marine ecosystems*. (Version v0.1.0). Zenodo. <https://doi.org/10.5281/zenodo.2653669>
- Forget, Gaël. (2021). IndividualDisplacements.jl: A julia package to simulate and study particle displacements within the climate system. *Journal of Open Source Software*, 6(60), 2813. <https://doi.org/10.21105/joss.02813>
- Forget, Gaël. (2024a). *Energy imbalance in the sunlit ocean layer*. (in prep). <https://doi.org/10.21203/rs.3.rs-3979671/v1>
- Forget, Gaël. (2024b). *ClimateModels.jl: Julia interface to climate models, and tracked modeling workflow framework*. (Version v0.3.2). Zenodo. <https://doi.org/10.5281/zenodo.10903961>
- Forget, G., Campin, J.-M., Heimbach, P., Hill, C. N., Ponte, R. M., & Wunsch, C. (2015). ECCO version 4: an integrated framework for non-linear inverse modeling and global ocean state estimation. *Geoscientific Model Development*, 8(10), 3071–3104. <https://doi.org/10.5194/gmd-8-3071-2015>
- Forget, Gaël, & Ferreira, D. (2019). Global ocean heat transport dominated by heat export from the tropical Pacific. *Nature Geoscience*. <https://doi.org/10.1038/s41561-019-0333-7>
- Gallmeier, K., Prochaska, J. X., Cornillon, P., Menemenlis, D., & Kelm, M. (2023). An evaluation of the LLC4320 global-ocean simulation based on the submesoscale structure of modeled sea surface temperature fields. *Geoscientific Model Development*, 16(23), 7143–7170. <https://doi.org/10.5194/gmd-16-7143-2023>
- Heimbach, P., Hill, C., & Giering, R. (2002). Automatic generation of efficient adjoint code for a parallel Navier-Stokes solver. In *Computational science—ICCS 2002* (pp. 1019–1028). Springer. https://doi.org/10.1007/3-540-46080-2_107
- Marshall, J., Adcroft, A., Hill, C., Perelman, L., & Heisey, C. (1997). A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers. *J. Geophys. Res*, 102(C3), 5753–5766. <https://doi.org/10.1029/96JC02775>
- Ramadhan, A., Wagner, G. L., Hill, C., Campin, J.-M., Churavy, V., Besard, T., Souza, A., Edelman, A., Ferrari, R., & Marshall, J. (2020). Oceananigans.jl: Fast and friendly geophysical fluid dynamics on GPUs. *Journal of Open Source Software*, 5(53), 2018. <https://doi.org/10.21105/joss.02018>
- Rousselet, L., Cessi, P., & Forget, G. (2021). Coupling of the mid-depth and abyssal components of the global overturning circulation according to a state estimate. *Science Advances*,

7(21), eabf5478. <https://doi.org/10.1126/sciadv.abf5478>

Wu, Z., & Forget, G. (2022). PlanktonIndividuals.jl: A GPU supported individual-based phytoplankton life cycle model. *Journal of Open Source Software*, 7(73), 4207. <https://doi.org/10.21105/joss.04207>