

DMStag: Staggered, Structured Grids for PETSc

Patrick Sanan ^{1,2}, Dave A. May  ³, Richard T. Mills  ², and Boris J. P. Kaus  ⁴

1 ETH Zurich, Switzerland 2 Argonne National Laboratory, USA 3 University of California, San Diego 4 Johannes Gutenberg University Mainz, Germany

DOI: [10.21105/joss.04531](https://doi.org/10.21105/joss.04531)

Software

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

Editor: [Kyle Niemeyer](#)  

Reviewers:

- [@kris-rowe](#)
- [@elauksap](#)
- [@mbarzegary](#)

Submitted: 18 June 2022

Published: 15 November 2022

License

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

Summary

“Staggered grid” codes arise in many applications, where a physical problem is discretized and solved in terms of unknowns located at different locations on a structured grid. For example, the Navier-Stokes and Stokes equations can be solved on a staggered grid with the classical Marker and Cell (MAC) method ([Harlow & Welch, 1965](#)), storing pressure unknowns in element centers and velocities on element boundaries. This gives a simple and compact method which exhibits superior numerical properties to colocated grid methods which consider all unknowns to be located on the same grid of points.

DMStag is a component of PETSc ([Balay et al., 2022a, 2022b](#)), the Portable, Extensible Toolkit for Scientific Computation, an MPI-parallel C library which is widely used to perform very large simulations in computational science and engineering. DMStag is a new implementation of the DM (Domain Management) to provide a native abstraction for defining and solving partial differential equations (PDEs) on structured, staggered grids, in parallel. With DMStag, as with other DM implementations in PETSc, one can write code which will run on any number of MPI ranks and rely on the DM API to work with global vectors representing the full discretized problem, and with regularly-blocked local vectors on the overlapping local patch upon which lower-level computations are performed. This allows users to scale codes to thousands of processors, and provides a platform upon which to compose the advanced, scalable PDE solvers, and to leverage various computational backends, notably including GPUs ([Mills et al., 2021](#)).

DMStag provides a simple API, similar to that for the DMDA class, to allow a user to create and interact with objects which (like all DM objects) represent

1. A discrete topological space (here, a structured cell complex)
2. An *atlas* of overlapping local patches, assigned to MPI ranks
3. A *field* assigning sets of scalar unknowns to each point in the topological space (here, a constant number of unknown “DOFs” for each point in a given *stratum*, defined as cells of a given dimension)
4. A special field for coordinates of each point

DMStag is included with and requires PETSc. As of PETSc 3.17, the vast majority of the source code to implement DMStag can be found in `src/dm/impls/stag/`, `include/petscdmstag.h` and `include/petsc/private/dmstagimpl.h`. For more information, please consult the [PETSc manual chapter on DMStag](#) and the [DMStag manual pages](#).

Statement of need

The discretization of PDEs on structured (or “regular”) grids of cells, associated unknown quantities with several different geometric entities (for instance, cell centers and cell faces), has applications in many fields. The highly structured nature of this discretization and relative simplicity of many of the stencils can lead to highly efficient and maintainable application

codes which solve important problems in fluid simulation (including weather and climate simulation and geophysical flows) and electromagnetic and plasma simulations. Users have commonly implemented these methods with PETSc, using the DM^A object. However, since this represents a structured grid with an equal number of unknowns attached to each vertex (or cell center, if interpreted that way), a staggered grid must either be represented as a collection of DM^A objects, or by introducing a convention to store staggered data at a nearby non-staggered location, ignoring unused points on the boundary. This is inconvenient and can also have performance consequences. The choice of approach commits one to interlace or segregate data from different physical fields, and the presence of unused “dummy” points in the global representation may induce additional complication when attempting to use grid-aware solvers like PETSc’s geometric or algebraic multigrid solvers, or block-based preconditioners (PCFIELDSPLIT).

DMStag is currently used in research simulating magmatic flow in the Earth’s mantle ([Pusok, Katz, et al., 2022](#)) as part of the FD-PDE framework ([Pusok, May, et al., 2022](#)), and MHD simulations for tokamak simulation. Its base functionality was introduced into PETSc with version 3.11 (March 2019).

We are not aware of any directly-comparable general-purpose frameworks. In many application codes, the relatively simple data structures involved can be implemented directly. This approach has the obvious advantage of avoiding a heavy dependency like PETSc, but could incur larger investment of implementation effort to experiment with new, more-scalable solvers or with GPU-backed data structures, both of which PETSc’s focus on composability and portability aim to make more accessible. Projects like [GridTools](#) and [STELLA](#) ([Gysi et al., 2015](#)) provide high-performance stencil operations for particular domains, in this case climate and weather modelling.

Examples

A few example codes are available with PETSc, currently in `src/dm/impls/stag/tutorials`. Additional codes in `src/dm/impls/stag/tests` are used for testing.

Example ex6 modifies a simple velocity-stress formulation for seismic wave propagation ([Virieux, 1986](#)) to operate in 3D: this shows usage of much of the basic API of DMStag to perform explicit time-stepping of fields associated with multiple, compatible DMStag objects. [Figure 1](#) shows one component of the velocity from one timestep of a 2D simulation with this code. This was generated with Paraview, loading output from frame 485 generated with

```
# Configure and build PETSC, set PETSC_DIR and PETSC_ARCH to working values
cd $PETSC_DIR/src/dm/impls/stag/tutorials
make ex6
./ex6 -nsteps 1000 -stag_grid_x 1000 -stag_grid_y 1000 -dt 4e-4
```

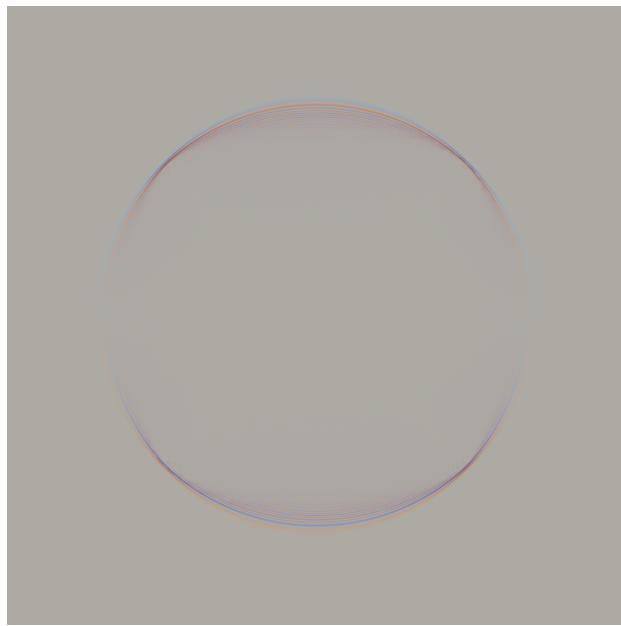


Figure 1: y velocity for a single timestep of a seismic wave simulation generated with DMStag ex6 on a 1000^2 element discretization of the unit square.

Example ex4 solves the variable-viscosity, stationary Stokes equations with finite differences (see e.g. Gerya (2019)). This demonstrates assembling and solving an explicit linear system. Figure 2 shows a flow field generated with this code, by using Paraview to visualize .vtr files generated with

```
# Configure and build PETSC, set PETSC_DIR and PETSC_ARCH to working values
cd $PETSC_DIR/src/dm/impls/stag/tutorials
make ex4
./ex4 -dim 3 -s 64 -coefficients sinker_box -isoviscous -nondimensional \
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type schur \
-pc_fieldsplit_schur_fact_type upper \
-fieldsplit_element_ksp_type preonly -fieldsplit_element_pc_type none \
-fieldsplit_face_pc_type mg -fieldsplit_face_pc_mg_levels 5 \
-fieldsplit_face_pc_mg_galerkin -fieldsplit_face_mg_levels_ksp_max_it 6 \
-dump_solution
```

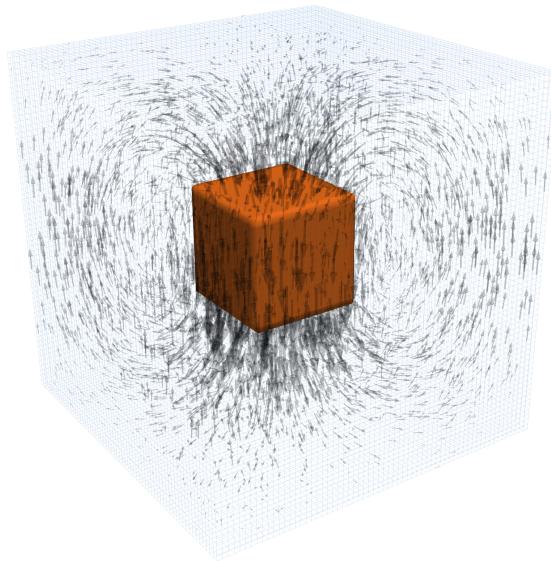


Figure 2: Flow field for stationary Stokes flow around a cubic inclusion, inspired by a benchmark setup from Furuichi et al. (2011) and computed with DMStag tutorial ex4. This solves the isoviscous Stokes equations, driven by a density anomaly in the inclusion, on a 64^3 element regular decomposition of the unit cube.

Acknowledgements

We thank the many members of the PETSc developer and user community who have contributed in ways large and small to this code's development and maintenance.

P. S. acknowledges financial support from the Swiss University Conference and the Swiss Council of Federal Institutes of Technology through the Platform for Advanced Scientific Computing (PASC) program, as part of the StagBL project.

References

- Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., ... Zhang, J. (2022a). *PETSc Web page*. <https://petsc.org/>
- Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., ... Zhang, J. (2022b). *PETSc/TAO users manual* (ANL-21/39 - Revision 3.17). Argonne National Laboratory.
- Furuichi, M., May, D. A., & Tackley, P. J. (2011). Development of a Stokes flow solver robust to large viscosity jumps using a Schur complement approach with mixed precision arithmetic. *Journal of Computational Physics*, 230(24), 8835–8851. <https://doi.org/10.1016/j.jcp.2011.09.007>
- Gerya, T. (2019). *Introduction to numerical geodynamic modelling* (2nd ed.). Cambridge University Press.
- Gysi, T., Osuna, C., Fuhrer, O., Bianco, M., & Schulthess, T. C. (2015). STELLA: A domain-

- specific tool for structured grid methods in weather and climate models. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–12.
- Harlow, F. H., & Welch, J. E. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12), 2182–2189. <https://doi.org/10.1063/1.1761178>
- Mills, R. T., Adams, M. F., Balay, S., Brown, J., Dener, A., Knepley, M., Kruger, S. E., Morgan, H., Munson, T., Rupp, K., Smith, B. F., Zampini, S., Zhang, H., & Zhang, J. (2021). Toward performance-portable PETSc for GPU-based exascale systems. *Parallel Computing*, 108, 102831. <https://doi.org/10.1016/j.parco.2021.102831>
- Pusok, A. E., Katz, R. F., May, D. A., & Li, Y. (2022). Chemical heterogeneity, convection and asymmetry beneath mid-ocean ridges. *Geophysical Journal International*, 231(3), 2055–2078. <https://doi.org/10.1093/gji/ggac309>
- Pusok, A. E., May, D., Li, Yuan, & Katz, R. (2022). FD-PDE. In *GitHub repository*. GitHub. <https://github.com/apusok/FD-PDE>
- Virieux, J. (1986). P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics*, 51(4), 889–901. <https://doi.org/10.1190/1.1442147>