PROJECT PROPOSAL: TOPOGRAPHY IN LARGE-EDDY SIMULATION

Victoria Whitley

Advisor: Jacob Wenegrat

PROBLEM FORMULATION

As the only computationally feasible technique for modeling ocean flows at scales that resolve both turbulence and large-scale dynamics, how can we include topography into large-eddy simulation, through either immerse boundary method or adding finite volume topography, to assist the state of the art solver, Oceananigans.jl, in answering the many research questions involving flow along and over topography?

Ramadhan, et al. (2020)

OUTLINE

- Background
- Motivation
- Methodology
- Project Goals
- Implementation
- Milestones
- Final Product

BACKGROUND: TURBULENCE

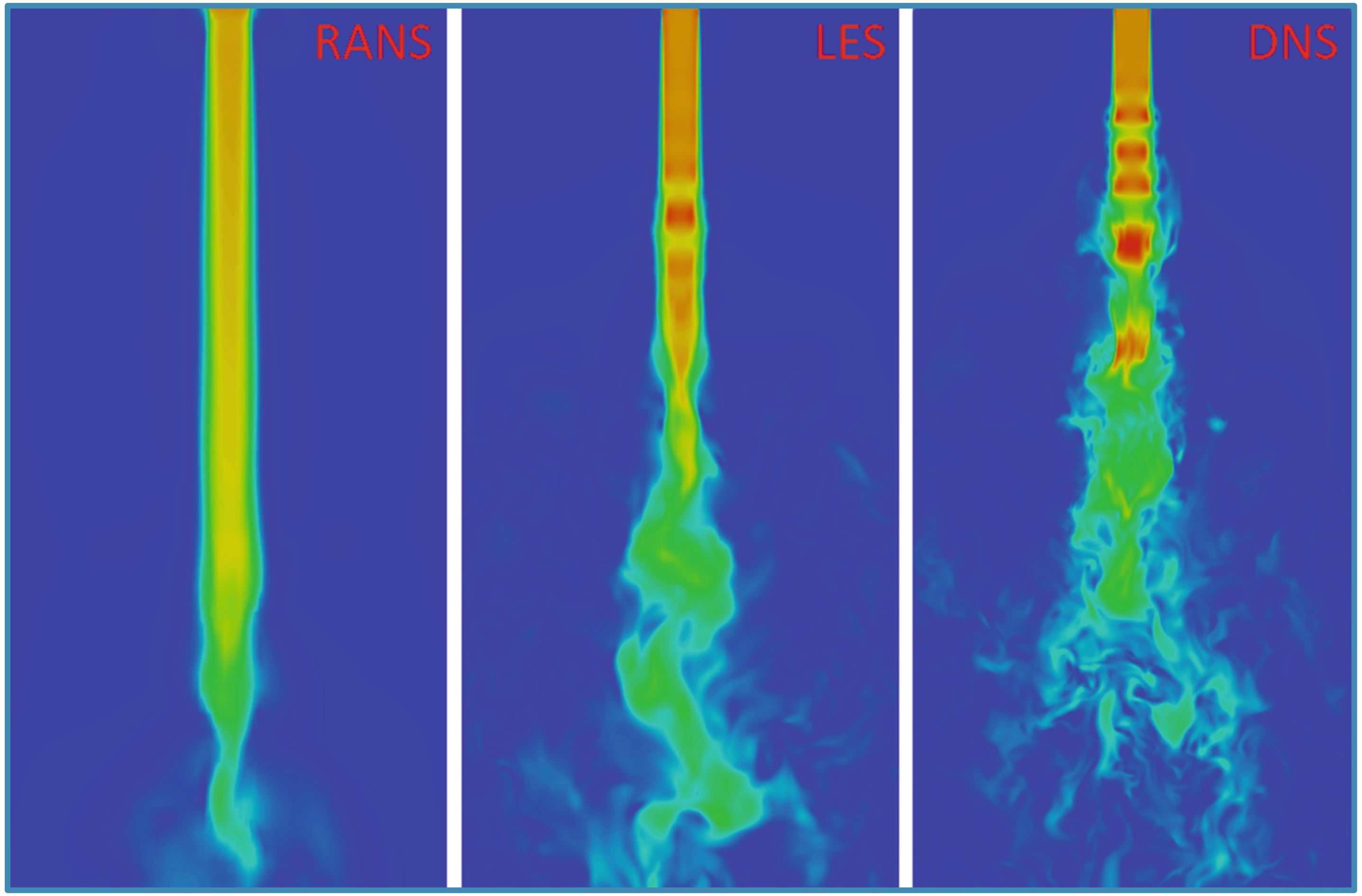
- Characteristics
  - Irregular and unsteady
  - Fluctuating in time and space
- Examples
  - Flow over a gold ball
  - Steam above a coffee cup
  - Mixed layers of the atmosphere or ocean
- Scales
  - Large eddies on scale with ocean mixed layer depth, ~100m, energy often added from boundary fluxes
  - Molecular level, where energy is dissipated by viscosity

Nuclear Power, (2020), Zhiyin, (2015)

BACKGROUND: LARGE-EDDY SIMULATION (LES)

- Resolves largest eddies
- Models smaller scales (inertial sub-range) based on physically motivated parametrizations
- Computationally feasible and accurate at desired resolution for turbulence
- Still constrained to ~10km x 10 km ocean domains
- Hard to deal with complex geometries in flow problems

Ramadhan, et al. (2020)



BACKGROUND: COMPARING SOLVERS

REYNOLDS-AVERAGED NAVIER-STOKES (RANS)

LES

DIRECT NUMERICAL SIMULATION (DNS)

Rodriguez. (2019)

BACKGROUND: OCEANANIGANS (O.JL)

- Climate Modeling Alliance (CliMA) to build climate model from scratch
- Oceananigans.jl is part of the model, to solve incompressible fluid problems for ocean
  - "fast and friendly"
  - Updated parametrizations and numerical methods
  - Supports LES and DNS for CPUs and GPUs
  - Written in Julia

Ramadhan, et al. (2020)

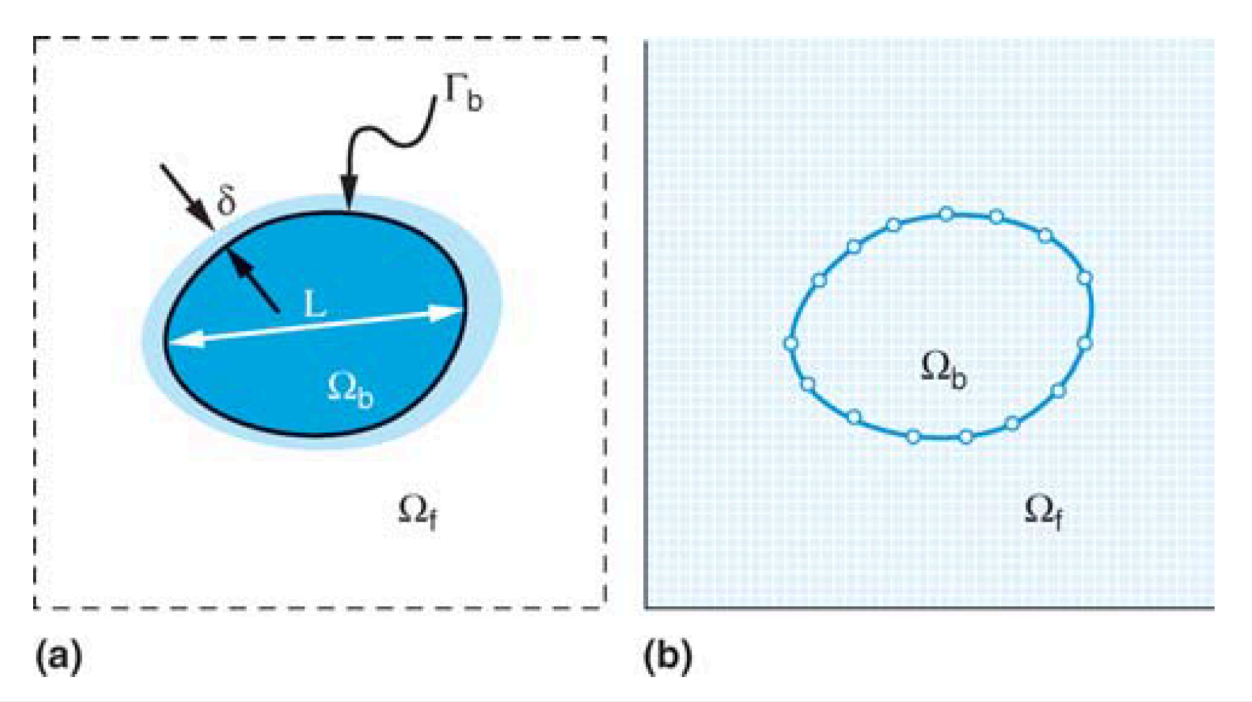
MOTIVATION: IMPORTANCE OF TOPOGRAPHY IN CLIMATE MODELS

- Topography can extract energy from geostrophic flows (10-100 km), creating sub-mesoscale turbulence (0.1 – 10 km)
- Complex bottom boundary layer can cause flow to separate from wall and mix with interior
- Undiscovered differences between sea surface layer and bottom boundary layer (BBL)
- Work around LES resolution limits with moving topography

Gula, (2016)
McWilliams, (2016
Wenegrat, (2020),(2018

METHODOLOGY: IMMERSED BOUNDARY METHOD (IBM)

- Instead of conforming the mesh to the fluid domain, Ω_f, a cartesian grid is generated without regard to the solid body, Ω_b
- Incorporate the BCs by modifying the equations near the boundary
- Allows discretization of complex domains without coordinate transformations or complicated discretization operators

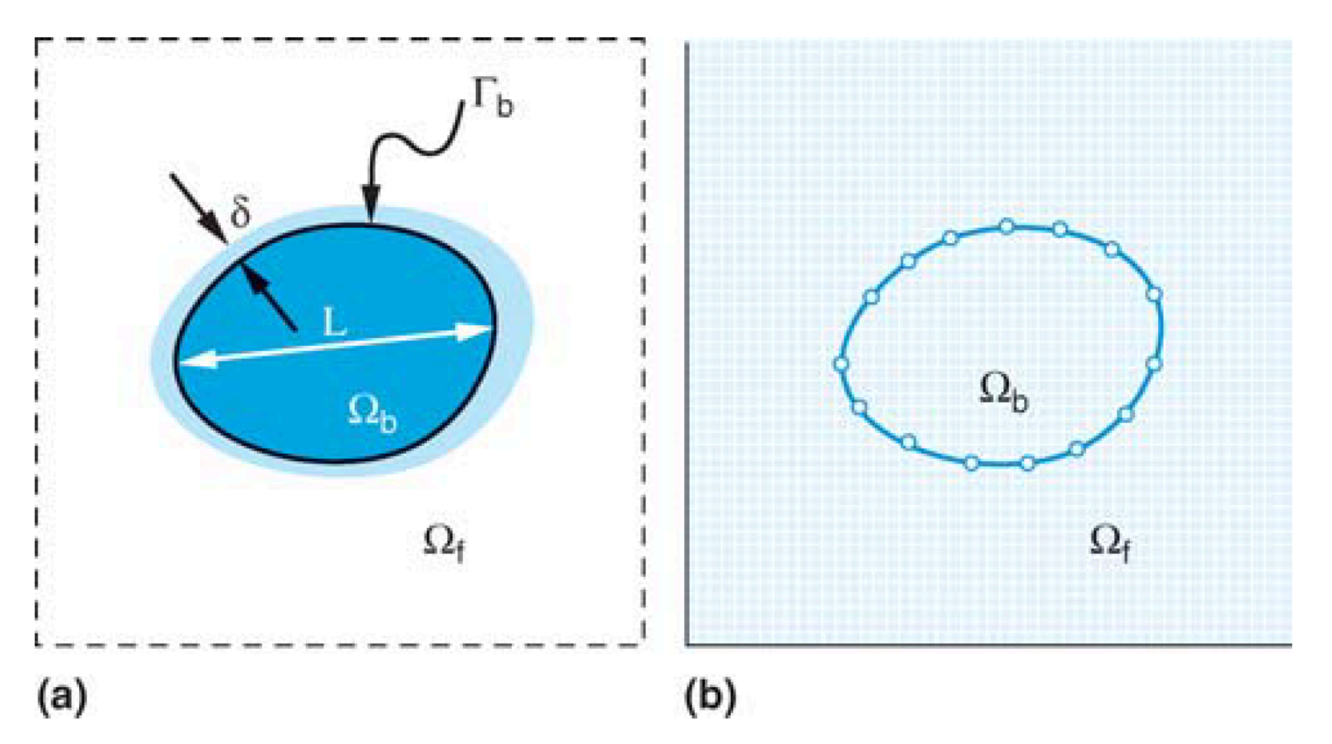


Mittal and Iaccarino, (2005)

METHODOLOGY: IMPOSING BOUNDARY CONDITIONS

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho} \nabla p - \frac{1}{Re} \nabla^2 \boldsymbol{u} = 0$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \in \Omega_f$$



With the boundary,

$$\boldsymbol{u} = \boldsymbol{u}_\Gamma \qquad \text{on } \Gamma_b$$

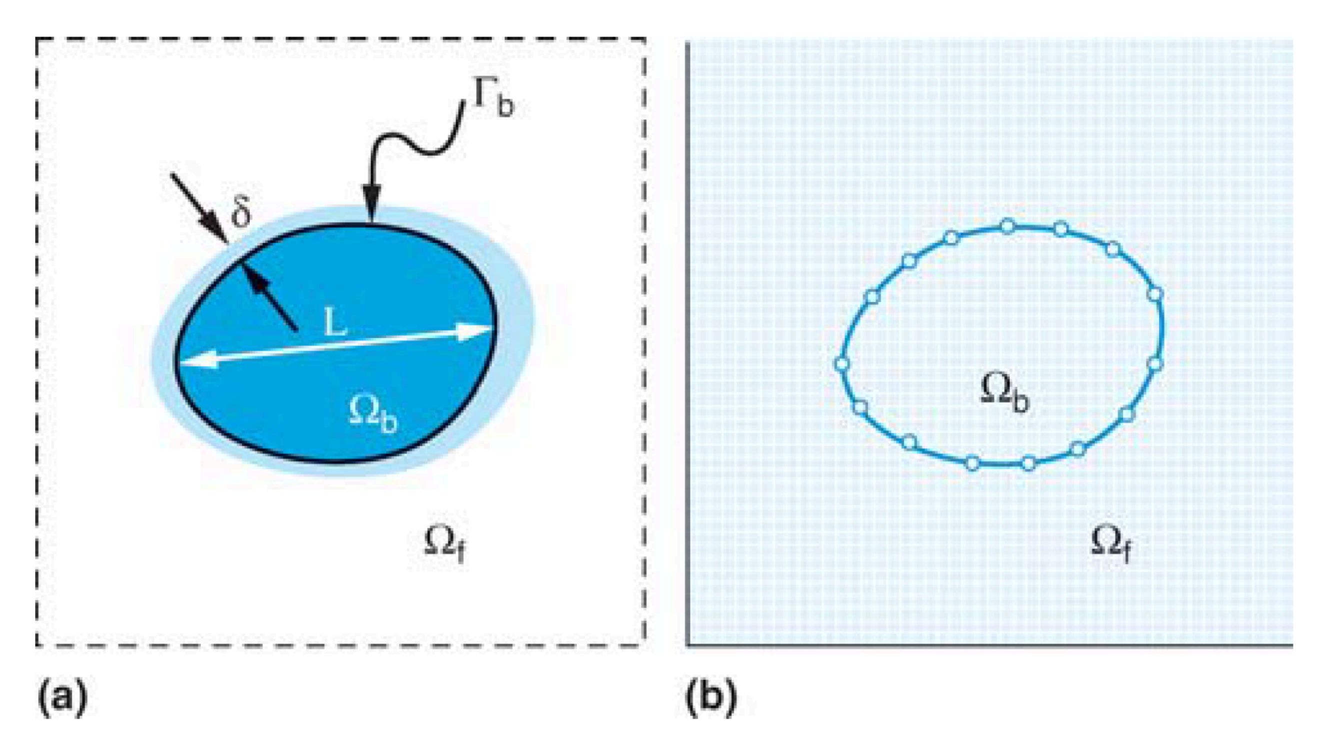
Mittal and Iaccarino, (2005)

METHODOLOGY: IMPOSING BOUNDARY CONDITIONS

$$\mathcal{L}(U) = 0 \qquad \in \Omega_{\mathrm{f}}$$

$$U = U_{\Gamma} \qquad \in \Gamma_b$$

where $U = (\boldsymbol{u}, p)$ and $\mathcal{L}$ is the Navier-Stokes equations operator.



Mittal and Iaccarino, (2005)

METHODOLOGY: CONTINUOUS FORCING

Modify continuous equations by adding in a forcing term $f_b = (\boldsymbol{f}_m, f_p)$ to both momentum and pressure equations

$$\mathcal{L}(U) = f_b \quad \in (\Omega_b \cup \Omega_f)$$

Then discretized on a Cartesian grid on the whole domain without regard to the original boundary

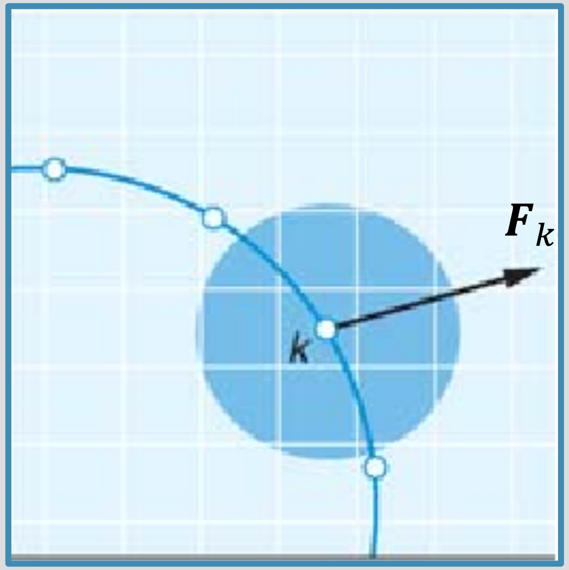
$$[\mathrm{L}]\{U\} = \{f_b\}$$

Mittal and Iaccarino, (2005)

METHODOLOGY: CONTINUOUS FORCING

IB represented by massless elastic fibers with Lagrangian points moving at local fluid velocity

$$\frac{\partial \boldsymbol{X_k}}{\partial t}(s,t) = \boldsymbol{u}(\boldsymbol{X_k}(s,t))$$



Force $f(x,t)$is then defined

$$\boldsymbol{f}(\boldsymbol{x},t) = \sum_k \boldsymbol{F}_k(t)\delta(|\boldsymbol{x} - \boldsymbol{X_k}|)$$

Where $\boldsymbol{F_k}$ is the stress of the fibers characterized by Hooke's law

Note: δ is usuaully implemented as a distribution function

Mittal and Iaccarino, (2005)

METHODOLOGY: DISCRETE FORCING

Discretize original system onto Cartesian grid without regard to boundary

$$[\mathrm{L}]\{U\} = 0$$

Adjust the discretization near the boundary through a modified discretized system

$$[\mathrm{L}']\{U\} = \{r\} \qquad \text{or} \qquad [\mathrm{L}]\{U\} = \{f_b{}'\}$$

Where $f_b' = \{r\} + [\mathrm{L}]\{U\} - [\mathrm{L}']\{U\}$

Mittal and Iaccarino, (2005)

METHODOLOGY: DISCRETE FORCING

Discretize original system onto Cartesian grid without regard to boundary as a time stepping scheme

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = RHS_i$$

Solve without forcing for $u^{*(n+1)}$ (does not satisfy IBC)

Compute f^{n+1} by adding it into the equation $\frac{u_i^{*(n+1)} - u_i^n}{\Delta t} = RHS_i + f^{n+1}$

Recompute the true solution $u^{(n+1)}$ with the new forcing term
$[\mathrm{L}]\{U^{(n+1)}\} = f^{n+1}$ (satisfies IBC)

Update velocity and pressure and continue for next time step

Mittal and Iaccarino, (2005)

METHODOLOGY: CONTINUOUS VS DISCRETE

	Pros	Cons
Continuous	• Easy to implement on elastic or moving boundaries • Independent of mesh	• Stability constraints for rigid bodies • Smoothing functions make sharp IB difficult
Discrete	• No extra stability constraints • Can create sharp IB	• Forcing procedure less practical • Difficult to include boundary motion • Dependent on discretization

Mittal and Iaccarino, (2005)

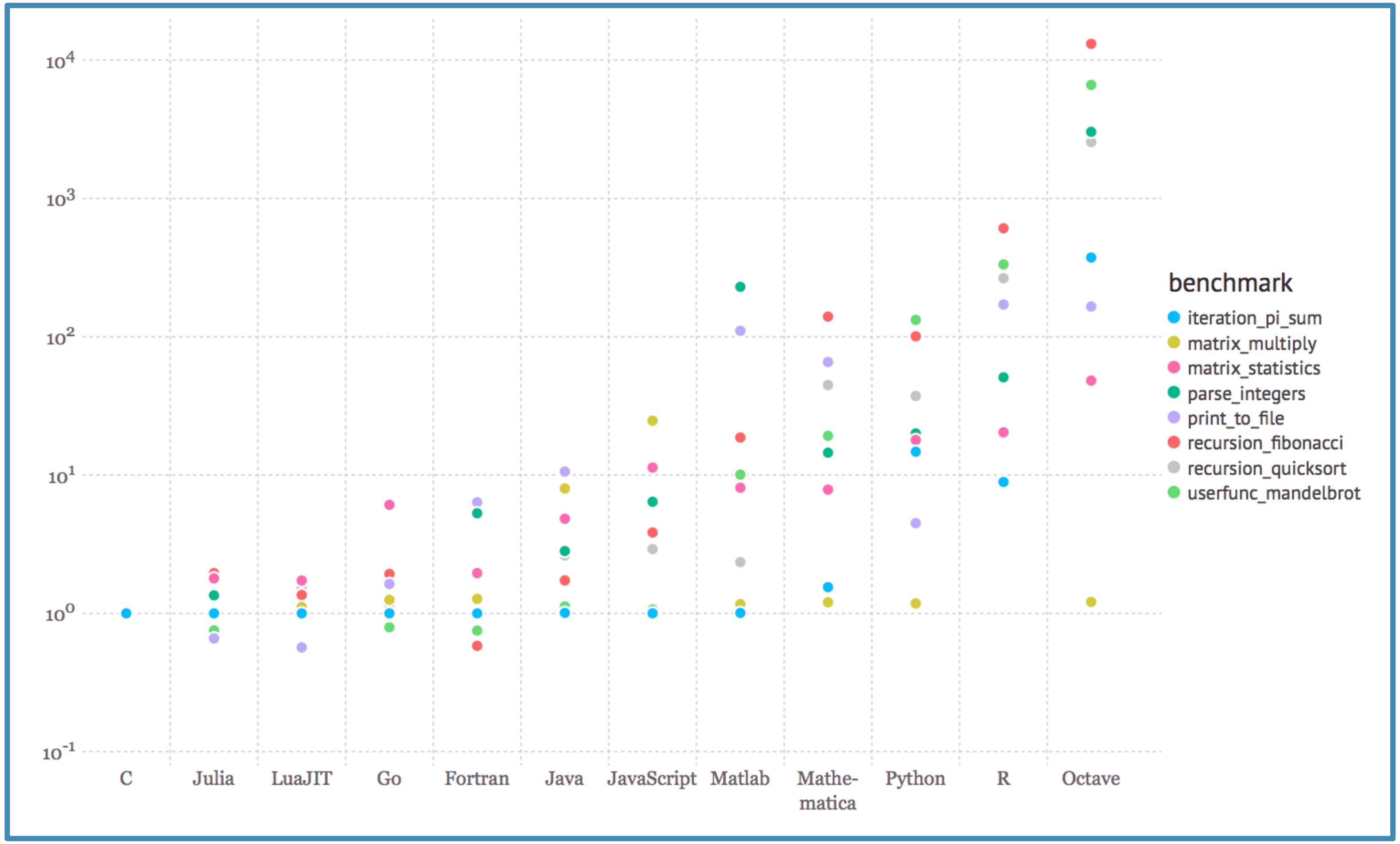
PROJECT GOALS

1. Finding best way to implement topography into LES
   - IBM vs Finite Volume Topography
   - Discrete vs Continuous Forcing
   - Dealing with different types of boundary conditions
2. Implement topography in O.jl
   - Compatibility with pressure solver
   - Keeping added computational overhead low
   - Following their standards of validation
3. Implement logarithmic wall model for modeling boundary stresses rather than direct no slip condition
4. (Possibly) Consider BBL shedding off topography as an application to code

IMPLEMENTATION: THE LANGUAGE

- Programming Language: Julia
  - Aims to combine efficiency and ease of use
  - Already has many CFD tools
  - Good for GPU compatibility and parallel computing
  - Can call Python, C, and Fortran libraries
  - "Walks like Python, runs like C."
  - Free and opensource!

Zhou, (2020),
Ramadhan, et al., (2020)
Edelman, et al., (2016)



IMPLEMENTATION: JULIA IN COMPARISON

Edelman, et al., (2016)

IMPLEMENTATION: TYPICAL EQUATIONS

- Spatially-filtered, incompressible Boussinesq equations, with tracer equations

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + (f - \nabla \times \boldsymbol{u}^S) \times \boldsymbol{u} = -\nabla\phi + b\,\hat{\mathbf{z}} - \nabla \cdot \boldsymbol{\tau} - \partial_t \boldsymbol{u}^S + \boldsymbol{F_u}$$

$$\partial_t \theta + \boldsymbol{u} \cdot \nabla\theta = -\nabla \cdot \boldsymbol{q} + \mathbf{F}_{\boldsymbol{\theta}}$$

$$\nabla \cdot \boldsymbol{u} = 0$$

- Linear buoyancy relationship

$$b = -\alpha\, g\, (\theta_0 - \theta)$$

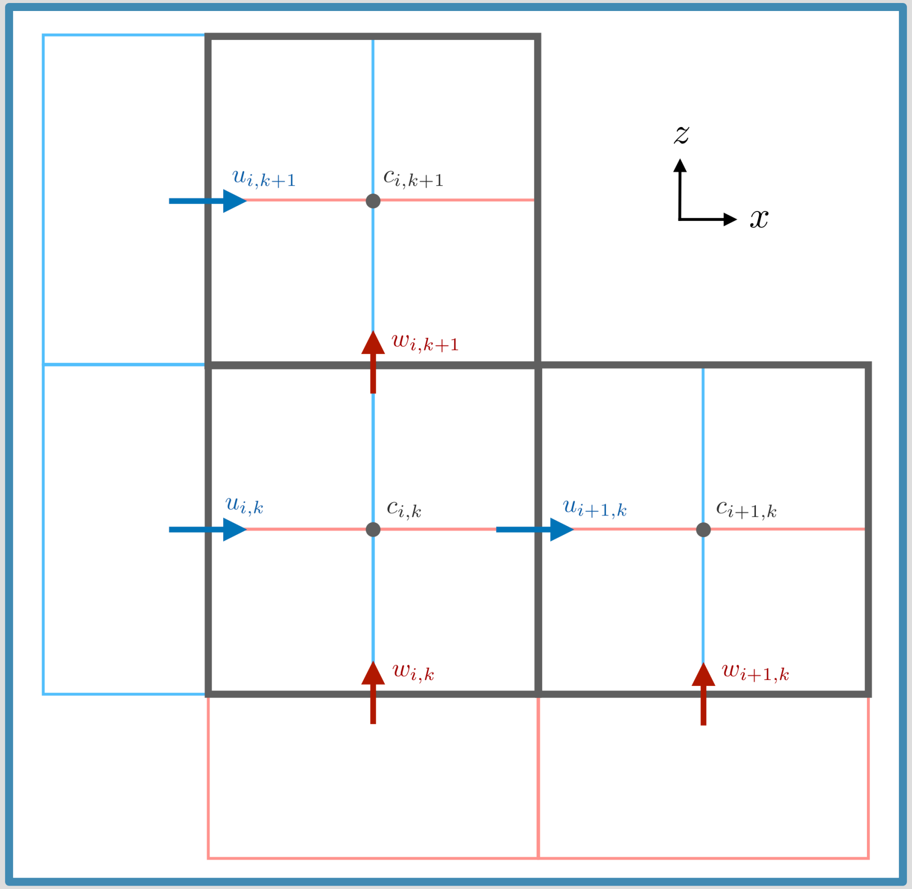
Souza, (2020)

IMPLEMENTATION: SPACE DISCRETIZATION

- Finite volume method

$$c_{i,j,k} = \frac{1}{V_{i,j,k}} \int c(\boldsymbol{x}) dV_{i,j,k}$$

- Staggered Arakawa C-grid, interpolated to perform operations
- Centered second-order differences for advection and diffusion terms



Souza, (2020)
Eymard and Gallouã, (2010)
Marshall, et al., (1997)

IMPLEMENTATION: TEMPORAL DISCRETIZATION

- Time integral of momentum equation with pressure decomposition

$$\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n} = \int_{t_n}^{t_{n+1}} \left[-\boldsymbol{\nabla}\phi_{non} - \boldsymbol{\nabla}_h \phi_{hyd} - (\boldsymbol{u} \cdot \boldsymbol{\nabla})\mathbf{u} \ - \mathbf{f} \times \boldsymbol{u} + \boldsymbol{\nabla} \cdot \boldsymbol{\tau} + \boldsymbol{F_u} \right] dt$$

$$\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n} \approx -\Delta t \, \boldsymbol{\nabla}\phi_{non}^{n+1} + \int_{t_n}^{t_{n+1}} \boldsymbol{G_u} \ dt$$

- 2nd order explicit Adams-Bashforth

$$\int_{t_n}^{t_{n+1}} G \, dt \approx \Delta t [\left(\frac{3}{2} + \mathcal{X}\right) G^{n} - \left(\frac{1}{2} + \mathcal{X}\right) G^{n-1}]$$

Souza, (2020)
Yinnian and Li, (2009)

IMPLEMENTATION: PRESSURE SOLVER

- Pressure projection method with Poisson pressure equation

$$\phi(\boldsymbol{x}, t) = \phi_{hyd}(\boldsymbol{x}, t) + \phi_{non}(\boldsymbol{x}, t)$$

$$\nabla^2 \phi_{non} = \frac{\nabla \cdot \boldsymbol{u}^{\boldsymbol{n}}}{\Delta t} + \nabla \cdot \boldsymbol{G}_u = \mathcal{F}$$

- Ensures incompressibility of u while decoupling u and p
- FFT eigenfunction expansion of 2nd order operator

$$\widehat{\phi_{non}}\,(i, j, k) = -\frac{\hat{\mathcal{F}}}{\lambda_i^x + \lambda_j^y + \lambda_k^z}$$

Souza, (2020)
Guermond, et al, (2006)
Ulrich and Sweet, (1988)

IMPLEMENTATION: HARDWARE

- Deepthought 2 HPC cluster
  - Primary HPC cluster maintained by the Division of IT at UMD
  - Being upgraded to RedHat Linux version 8 (RHEL8)
  - Includes 40 GPU nodes

DIT, (2017)

IMPLEMENTATION: VALIDATION METHODS

1. Accuracy
   - Classic channel flow, but create wall with IB
   - Griffith, B. and Luo, X. (2017) and Kallemov, B., et al. (2016) "benchmark problems" with rigid, elastic, and contracting structures: (un)steady flow past stationary cylinder(s) or sphere(s), steady-state flow through a nozzle
2. Computational Cost
   - Compare to current O.jl solver with regular BC implementation
3. Oceananigans.jl Validation Methods
   - Convergence tests including 2D integration tests with non-trivial pressure fields, advection, diffusion
   - Lid-driven cavity test:
   - Stratified Couette flow

MILESTONES

- **October**: Familiarize with Oceananigans and determine IB method
- **November**: Implement topography for simpler case of Dirichlet BC
- **December**: Determine how to implement topography for Neumann BC
- **January**: Solidify and implement topography for various cases in O.jl
- **February**: Test accuracy for various boundary conditions and fix bugs
- **March**: Add wall model and reduce computational costs using O.jl methods
- **April**: Consider scientific question of bottom boundary layer dynamics
- **May**: Get code and method approved for inclusion in next O.jl release

FINAL PRODUCT

- Fast and easy method of implementing topography in Oceananigan.jl's LES code
- Code approved and added to next Oceananigans.jl release
- A tool to use in future research simulating the separation of ocean boundary layers over rough topography

SOURCES

- Arthur, Robert S., et al. "Evaluating Implementations of the Immersed Boundary Method in the Weather Research and Forecasting Model." *Monthly Weather Review*, vol. 148, no. 5, 2020, pp. 2087–2109., doi:10.1175/mwr-d-19-0219.1.
- Edelman, Alan, et al. "Julia 1.5 Documentation." *Julia Documentation · The Julia Language*, 2016, docs.julialang.org/en/v1/.
- E. Griffith, Boyce, and Xiaoyu Luo. "Hybrid Finite Difference/Finite Element Immersed Boundary Method." *International Journal for Numerical Methods in Biomedical Engineering*, vol. 33, no. 12, 2017, doi:10.1002/cnm.2888.
- "Examples of Turbulent Flow.", www.nuclear-power.net/nuclear-engineering/fluid-dynamics/turbulent-flow/examples-of-turbulent-flow/.
- Eymard, Robert, and Thierry Gallouã. "Finite Volume Method." *Scholarpedia*, vol. 5, no. 6, 2010, p. 9835., doi:10.4249/scholarpedia.9835.
- Fadlun, E.A., et al. "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations." *Journal of Computational Physics*, vol. 161, no. 1, 2000, pp. 35–60., doi:10.1006/jcph.2000.6484.
- Guermond, J.l., et al. "An Overview of Projection Methods for Incompressible Flows." *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 44-47, 2006, pp. 6011–6045., doi:10.1016/j.cma.2005.10.010.
- Gula, Jonathan, et al. "Topographic Generation of Submesoscale Centrifugal Instability and Energy Dissipation." *Nature Communications*, vol. 7, no. 1, 2016, doi:10.1038/ncomms12811.

SOURCES

- He, Yinnian, and Jian Li. "Numerical Implementation of the Crank-Nicolson/Adams-Bashforth Scheme for the Time-Dependent Navier-Stokes Equations." *International Journal for Numerical Methods in Fluids*, 2009, doi:10.1002/fld.2035.
- Iaccarino, Gianluca, and Roberto Verzicco. "Immersed Boundary Technique for Turbulent Flow Simulations." *Applied Mechanics Reviews*, vol. 56, no. 3, 2003, pp. 331–347., doi:10.1115/1.1563627. .
- Kallemov, Bakytzhan, et al. "An Immersed Boundary Method for Rigid Bodies." *Communications in Applied Mathematics and Computational Science*, vol. 11, no. 1, 2016, pp. 79–141., doi:10.2140/camcos.2016.11.79.
- Marshall, John, et al. "A Finite-Volume, Incompressible Navier Stokes Model for Studies of the Ocean on Parallel Computers." *Journal of Geophysical Research: Oceans*, vol.102, no. C3, 1997, pp. 5753–5766., doi:10.1029/96jc02775.
- Mcwilliams, James C. "Submesoscale Currents in the Ocean." *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 472, no. 2189, 2016, p. 20160117., doi:10.1098/rspa.2016.0117.
- Mittal, Rajat, and Gianluca Iaccarino. "Immersed Boundary Methods." *Annual Review of Fluid Mechanics*, vol. 37, no. 1, 2005, pp. 239–261., doi:10.1146/annurev.fluid.37.061903.175743.
- Ramadhan, Ali, et al. "Oceananigans.jl." *CliMA/Oceananigans.jl*, GitHub, 2020, github.com/CliMA/Oceananigans.jl.
- Rodriguez, Sal. "LES and DNS Turbulence Modeling." *Applied Computational Fluid Dynamics and Turbulence Modeling: Practical Tools, Tips and Techniques*, by Sal Rodriguez, Springer, 2019, pp. 197–223.

SOURCES

- Schumann, Ulrich, and Roland A Sweet. "Fast Fourier Transforms for Direct Solution of Poisson's Equation with Staggered Boundary Conditions." *Journal of Computational Physics*, vol. 75, no. 1, 1988, pp. 123–137., doi:10.1016/0021-9991(88)90102-7.
- Souza, Andre Nogueira, et al. "Uncertainty Quantification of Ocean Parameterizations: Application to the K-Profile-Parameterization for Penetrative Convection." *Journal of Advances in Modeling Earth Systems*, 2020, doi:10.1002/essoar.10502546.1.
- Verzicco, Roberto, et al. "Large Eddy Simulation in Complex Geometric Configurations Using Boundary Body Forces." *AIAA Journal*, vol. 38, 2000, pp. 427–433., doi:10.2514/3.14430.
- Wagner, Gregory, et al. "Near-Inertial Waves and Turbulence Driven by the Growth of Swell." *Journal of Physical Oceanography*, 30 July 2020, doi:10.1002/essoar.10503838.1.
- Wenegrat, Jacob O., and Leif N. Thomas. "Centrifugal and Symmetric Instability during Ekman Adjustment of the Bottom Boundary Layer." *Journal of Physical Oceanography*, vol. 50, no. 6, 2020, pp. 1793–1812., doi:10.1175/jpo-d-20-0027.1.
- Wenegrat, Jacob O., et al. "Submesoscale Baroclinic Instability in the Bottom Boundary Layer." *Journal of Physical Oceanography*, vol. 48, no. 11, 2018, pp. 2571–2592., doi:10.1175/jpo-d-17-0264.1.
- Zhiyin, Yang. "Large-Eddy Simulation: Past, Present and the Future." *Chinese Journal of Aeronautics*, vol. 28, no. 1, 2015, pp. 11–24., doi:10.1016/j.cja.2014.12.007.
- Zhou, Hua. "Introduction to Julia." *Juliaintro*, 2020, hua-zhou.github.io/teaching/biostatm280-2018spring/slides/02-