Data associated with Hughes et al. (in prep)

Ken Hughes, October 2021

Pathways, form drag, and turbulence in simulations of an ocean flowing through through an ice melange

To be submitted to *J. Geophys. Res. Oceans*

Summary file

Within the paper, many of the simulations are summarized as single data points for either form drag or turbulent dissipation. A csv file ('melange_flow_simulations_summary.csv') contains just these summary data as integrated quantities. That is, form drag is area integrated to give a quantity in Newtons and turbulent dissipation is volume integrated to give a quantity in Watts. Note that, within this file, $\lambda = 0.0$ denotes the single isolated obstacle and $\lambda = 1.0$ denotes the channel-wide obstacle.

Individual output files

For each simulation, we have also included the output in its own netCDF file.

File naming conventions

As noted in the paper, the main set of simulations involved five velocities U and fourteen fractional coverages λ:

$U = 0.02$, 0.05, 0.12, 0.25, and 0.40 $\mathrm{m\,s^{-1}}$

$\lambda = 0.01$, 0.02, 0.03, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50 and 0.55

For these 5×14 runs, each file is named

`run.lambda_lll.U_uuu.nc` where `lll` is 1000λ and `uuu` is $1000U$.

Hence, as an example with $U = 0.05\,\mathrm{m\,s^{-1}}$ and $\lambda = 0.15$, the file would be `run.lambda_050.U0_150.nc`.

Two sets of additional runs involved either a single obstacle or a channel-wide obstacle. These are named

`run.single_obstacle.U_uuu.nc`

`run.channel_wide_obstacle.U_uuu.nc`

netCDF output details

Each file originates at the output from MITgcm's diagnostic package. However, to make the output more user-friendly and to save space, we made a few changes. First, we shortened some variable names (`UE_VEL_C` to `U`, `VN_VEL_C` to `V`, `WVEL` to `W`, and `T` to `t`). Second, we removed the dynamically unimportant lower half of each simulation as well as the 8km restoring regions at the end of the channel. Third, we converted the output files from double precision (flaot-64) to single precision (float-32) and compressed the output using zlib compression implemented in xarray's `to_netcdf` function.

Note that density in the simulations is linearly proportional to temperature, which is whey there is a `THETA` quantity in the output files. To get density, use $\rho = \rho_0 \alpha T$ where $\rho_0 = 1000\,\mathrm{kg\,m^{-3}}$ and $\alpha = -2 \times 10^{-4}\,\mathrm{K^{-1}}$.

As noted in the paper, there are issues with ϵ values (named `KLeps`) in the output. The affected values have been set to 0.