

Instructions for running simulations

The code for running ice-ocean simulations using the Firedrake framework is available at this github site: <https://github.com/thwaitesproject/thwaites.git>

The Zenodo repository also contains a copy of the repository on 30th January 2023.

Instructions for downloading Firedrake can be found here:

<https://www.firedrakeproject.org/download.html>

Once Firedrake is installed and you have cloned the thwaites git repository to install thwaites in the Firedrake venv:

- `source ~/firedrake/bin/activate`
- `pip install -e thwaites`
- `cd thwaites`
- `pip install -r requirements.txt`

Section 3.1 MMS test

The run script for the MMS test in section 3.1 is:

`thwaites/tests/mms_groundingzone_TS/mms_groundingzone.py`

With the Firedrake venv activated it can be run with

- `cd thwaites/tests/mms_groundingzone_TS`
- `python mms_groundingzone.py`

Section 3.2 3d ISOMIP+ Ocean0

The run script for the Ocean0 ISOMIP+ experiment in section 3.2 is:

`thwaites/run_scripts_testing/ice_shelf_cavity/isomip_plus/isomip_plus_3d_extruded_layered.py`

With the Firedrake venv activated it can be run with:

- `cd thwaites/run_scripts_testing/ice_shelf_cavity/isomip_plus/`
- `python isomip_plus_3d_extruded_layered.py 24.01.23 4000 30 6 1e-3 1 5e-5 900 864000 8640000`

where the arguments are: DATE, dx, nz (number of layers), mu_h (horizontal viscosity), mu_v, kappa_h, kappa_v, dt, output_dt, Tend

nz - number of layers

mu_h (kappa_h) – horizontal viscosity (diffusivity)

mu_v (kappa_v) – vertical viscosity (diffusivity)

The meshes used for different horizontal resolution are:

2km - `isomip_outline_mesh_res_alignedinterioricefront_2km.msh`

4km - `isomip_outline_mesh_res_alignedinterioricefront.msh`

8km - `isomip_outline_mesh_res_alignedinterioricefront_8km.msh`

Readme: Towards a fully unstructured ocean model for ice shelf cavity environments:
30.01.23 - William Scott

Non-uniform refined GL and western boundary -
isomip_outline_mesh_res_alignedinterioricefront_westerngl2km_8km.msh

Note these simulations were run on 32 cores with 124Gb allocated memory.

MITgcm

The MITgcm setup files can be found in the folder "MITgcm_run_isomip_2km.zip"

Notes on compiling MITgcm

Go to the directory where you want to build mitgcm

If not within mitgcm directory you need to do:

```
~/MITgcm/tools/genmake2 -rootdir=/home/wis15admin/MITgcm -mods ~/MITgcm/ben_frisp/code  
-optfile=/home/wis15admin/MITgcm/tools/build_options/linux_amd64_gfortran
```

(Note the file paths need to be updated for your specific computer. We have also included a copy of the 'MITgcm/ben_frisp/code' in the zenodo repository "MITgcmbuild.zip". This is used when compiling MITgcm for a specific grid domain.)

```
make depend
```

```
make -j 4
```

Then copy "mitgcmuv" executable to where you want to run MITgcm with the grid specific setup files in "MITgcm_run_isomip_2km.zip"

```
./mitgcmuv
```

For parallel (mpi)

Same but with -mpi after code

```
~/MITgcm/tools/genmake2 -rootdir=/home/wis15admin/MITgcm -mods ~/MITgcm/ben_frisp/code  
-mpi -optfile=/home/wis15admin/MITgcm/tools/build_options/linux_amd64_gfortran
```

```
make depend
```

```
make -j 4
```

(As above, but:

```
mpirun -np X mitgcmuv
```

where X is the number of processors allocated during compilation.)

More information about MITgcm can be found here: <https://mitgcm.readthedocs.io/en/latest/>

Section 3.3 Preliminary adjoint sensitivity calculations

The run script for the adjoint sensitivity calculations is here:

```
thwaites/run_scripts_testing/ice_shelf_adjoint/ice_shelf_coarse.py
```

With the Firedrake venv activated it can be run with:

- cd thwaites/run_scripts_testing/ice_shelf_adjoint/
- python ice_shelf_coarse.py

Readme: Towards a fully unstructured ocean model for ice shelf cavity environments:
30.01.23 - William Scott

The mesh is in the same folder:
thwaites/run_scripts_testing/ice_shelf_adjoint/coarse.msh

The dump file from 50 days of the forward run is in the same folder

thwaites/run_scripts_testing/ice_shelf_adjoint/ 50day_Kh0.25Kv1e-3_dx500m_dy2m_dump.h5

Note this patch is needed in Firedrake:

```
diff --git a/firedrake/adjoint/blocks.py b/firedrake/adjoint/blocks.py
```

```
index 6a6a254e..c07744e9 100644
```

```
--- a/firedrake/adjoint/blocks.py
```

```
+++ b/firedrake/adjoint/blocks.py
```

```
@ @ -194,6 +194,7 @@ class NonlinearVariationalSolveBlock(GenericSolveBlock):
```

```
    tmp_bc = self.compat.create_bc(c, value=self.compat.extract_subfunction(adj_sol_bdy, c.function_space()))
```

```
    return [tmp_bc]
```

```
elif isinstance(c, self.compat.MeshType):
```

```
+     return None
```

```
    # Using CoordianteDerivative requires us to do action before
```

```
    # differentiating, might change in the future.
```

```
    F_form_tmp = firedrake.action(F_form, adj_sol)
```

Any questions I am very happy to help (Will Scott - wis15@ic.ac.uk) .