1. INTRODUCTION

MITberg is a dynamic-thermodynamic iceberg model simulating the melt and drift of icebergs calved from land-based ice sheets into the ocean. The model was developed at the University of Massachusetts Amherst by Alan Condron and has been coupled to the Massachusetts Institute of Technology general circulation model (MITgcm). MITberg runs efficiently in parallel on a large number of compute nodes, allowing it to be used for a wide variety of simulations both at different timescales and resolutions.

1.2 Dynamics

Icebergs are simulated as lagrangian particles with horizontal motion (acceleration) derived from the equation of motion for an iceberg:

\[
M_i \frac{d\mathbf{V}_i}{dt} = -M_i f \times \mathbf{V}_i + \mathbf{F}_a + \mathbf{F}_w + \mathbf{F}_s + \mathbf{F}_p + \mathbf{F}_r
\]

where \(M_i\) is iceberg mass (kg), \(\mathbf{V}_i\) horizontal velocity of the iceberg (m.s\(^{-1}\)), \(f\) the Coriolis parameter, \(\mathbf{F}_a\) the wind drag, \(\mathbf{F}_w\) the water drag, \(\mathbf{F}_s\) the sea ice drag, \(\mathbf{F}_p\) the horizontal pressure gradient, and \(\mathbf{F}_r\) wave drag. Ocean drag is calculated using either the ocean model surface velocities (model level 1) – which is typical of most current iceberg models – or by considering drag at each vertical ocean model level each iceberg penetrates. By default, icebergs are assumed to be rectangular and uniform in size with depth, although MITberg has a keel model to prescribe a more authentic (conical) shape both above and below the waterline. The keel model uses a 2\(^{nd}\) order polynomial fit data to create a keel shape consistent with data from sonar profiles of icebergs observed along the coast of Labrador (e.g. Savage 2001; Barker et al. 2004). The keel model is applied to both iceberg length and width and also to the subaerial portion of the iceberg to create a sail model. Figure 1 provides a graphical illustration of what the shape of a 6 level iceberg would look like:
Figure 1: An illustration of the multilevel drag scheme in MITberg and the change in shape produced by using the keel model. The iceberg shape is more consistent with icebergs observed off the coast of Labrador, Canada. In this example the iceberg penetrates 6 vertical levels in the ocean model resulting in the widest part of the iceberg being at the third level which in this configuration is at 30 m depth. The total drag exerted on the iceberg is a sum of the ocean drag forces exerted on each layer of the keel model (here 1 through 6), and the wind on the sail. The gray dashed line represents the original rectangular shape of the iceberg prior to being modified by the keel model. Blue (orange) arrows represent an idealized change in ocean velocity with depth, and the orange arrow a hypothetical wind force.

The total drag force due to the ocean is based on the classic drag equation and (if using the multilevel drag scheme) given by summing the drag force at each vertical level the iceberg penetrates:  

\[ \vec{F}_w = \frac{1}{2} \rho_w C_{w,v} \sum_{k=1}^{n} D(k)(T(k) - T_s)|\vec{V}_w(k) - \vec{V}_i|(|\vec{V}_w(k) - \vec{V}_i|) \]

where \( P_w \) is water density, \( C_{w,v} \) is the vertical form drag coefficient for water, \( D(k) \) is the horizontal length of the iceberg normal to the stressing force (i.e. L or W), \( T(k) \) the vertical thickness of each level, \( \vec{V}_w(k) \) the horizontal velocity of the ocean at each level, and \( T_s \) the vertical thickness of sea ice (Table 1 gives a list of the values of the various coefficients). Sea ice thickness is only considered for the surface level (\( k=1 \)) and only when the sea ice concentration \( \geq 0.15 \). At the base of the iceberg (\( k=n \)) water drag acting in the horizontal direction (so called skin friction) is also accounted for using the horizontal drag coefficient, \( C_{w,h} \). The atmospheric drag created by wind stress on both the vertical side wall above the waterline and along the top surface is given by:

\[ \vec{F}_a = \frac{1}{2} \rho_a (C_{a,v} A_h + C_{a,h} W L) |\vec{V}_a - \vec{V}_i|(|\vec{V}_a - \vec{V}_i|) \]

where \( C_{a,v} \) and \( C_{a,h} \) are the vertical and horizontal form drag coefficients for air, \( A_h \) is the area of the vertical sidewall of the iceberg normal to the stressing force above the waterline, \( \vec{V}_a \) is the horizontal 10 meter wind velocity. In a similar fashion, the drag force exerted by sea ice on the iceberg is given by:

\[ \vec{F}_s = \frac{1}{2} \rho_s C_{s,v} D T_s |\vec{V}_s - \vec{V}_i|(|\vec{V}_s - \vec{V}_i|) \]
where D is the horizontal dimension normal to the stressing force in the surface level of the keel model (W or L at k=1). The drag force from sea ice is only considered when the concentration of sea ice ≥15%, while icebergs surrounded by high (>90%) concentrations of sea ice drift entirely with the pack ice (i.e. with same U and V velocities as the sea ice model) until the ice concentration sufficiently reduces. Lastly a pressure gradient force is calculated directly from the sea surface height simulated by the ocean models nonlinear free surface:

\[ \vec{F}_p = -M_1 g \vec{\eta} \]

At this point in the development of MITberg, the model will not work without a nonlinear free surface.

Table 1: A list of the main coefficients used in the iceberg dynamics model.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Description</th>
<th>Units</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho_i )</td>
<td>density of iceberg</td>
<td>kg.m(^{-3} )</td>
<td>917</td>
</tr>
<tr>
<td>( \rho_w )</td>
<td>density of water</td>
<td>kg.m(^{-3} )</td>
<td>1025</td>
</tr>
<tr>
<td>( \rho_a )</td>
<td>density of air</td>
<td>kg.m(^{-3} )</td>
<td>1.2</td>
</tr>
<tr>
<td>( \rho_s )</td>
<td>density of sea ice</td>
<td>kg.m(^{-3} )</td>
<td>910</td>
</tr>
<tr>
<td>( C_{w,v} )</td>
<td>vertical drag coefficient for water</td>
<td>Dimensionless</td>
<td>1.0</td>
</tr>
<tr>
<td>( C_{a,v} )</td>
<td>vertical drag coefficient for air</td>
<td>Dimensionless</td>
<td>0.8</td>
</tr>
<tr>
<td>( C_{s,v} )</td>
<td>vertical drag coefficient for sea ice</td>
<td>Dimensionless</td>
<td>1.0</td>
</tr>
<tr>
<td>( C_{w,h} )</td>
<td>horizontal drag coefficient for water</td>
<td>Dimensionless</td>
<td>0.0012</td>
</tr>
<tr>
<td>( C_{a,h} )</td>
<td>horizontal drag coefficient for air</td>
<td>Dimensionless</td>
<td>0.0055</td>
</tr>
<tr>
<td>( g )</td>
<td>Gravity</td>
<td>m.s(^{-2} )</td>
<td>9.8</td>
</tr>
</tbody>
</table>

1.3. Thermodynamics

Melting of icebergs occurs both above (subaerial) and below (subaqueous) the waterline. All the melt rates given here are in units of meters per second (m s\(^{-1} \)). In the model, subaerial melt occurs due to radiative heating, \( M_{sr} \), and sensible heating (i.e. forced convection), \( M_{f,a} \), while melt below the waterline occurs due to forced convection, \( M_{f,w} \), buoyant vertical convection, \( M_b \), and wave erosion, \( M_w \) (Savage et al., 2001; Kubat et al., 2007). Radiative heating reduces the vertical thickness (T) of the iceberg:

\[ M_{sr} = \frac{F_{sol}}{\rho_i \Gamma_i} (1 - \alpha) \]

where \( F_{sol} \) is the solar radiation flux (W m\(^{-2} \)) taken from the atmospheric forcing fields, \( \Gamma_i \) is the latent heat of fusion of ice (J kg\(^{-1} \)) and \( \alpha \) is the iceberg albedo. Subaerial melt from sensible heating (also known as forced convection) is generated by turbulence created by the relative motion of the air passing the iceberg (Savage et al., 2001; Kubat et al., 2007). Forced convection leads to both a reduction in waterline length and vertical thickness. When 2-m air temperature is above freezing (T>0°C) the melt rate is given by

\[ M_{f,a} = \frac{q_f}{\rho_i \Gamma_i} \]
where $q_f$ is the heat flux per unit surface area ($J\ s^{-1}\ m^{-2}$). Melt from forced convection is also generated below the waterline, $M_{f,w}$, by turbulence created by the relative motion of the water passing the iceberg keel. Below the waterline, melt from buoyant vertical convection occurs along the side walls of an iceberg due to the temperature difference between ice and water and reduces iceberg length (by default iceberg temperature is set as $-4^\circ C$). The equation used in MITberg is derived from the CIS iceberg model (Kubat et al., 2007) and takes into account changes in the freezing point of seawater due to salinity. Melt from wave erosion is assumed to reduce iceberg length and based on the CIS wave erosion equation:

$$M_w = 0.000146 \left( \frac{R}{W_h} \right)^{0.2} \left( \frac{W_h}{W_p} \right) \Delta T$$

where $R$ is the roughness height of the iceberg, $W_p$ the wave period (based El-Tahan et al., 1987). Wave height, $W_h$, is dampened by the presence of sea ice and capped at the iceberg freeboard height, $F$, to avoid producing unrealistically high melt rates. Erosion at the waterline creates a notch that progressively deepens to create overhanging slabs of ice which eventually become unstable and break off into the ocean. Finally, precipitation falling on the iceberg as snow (when $T_a \leq 0^\circ C$) increases vertical ice thickness, based on the ice equivalent of water.

Table 2: A list of the main coefficients used in the iceberg thermodynamics model.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Description</th>
<th>Units</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_i$</td>
<td>latent heat of fusion of ice</td>
<td>J kg$^{-1}$</td>
<td>3.33x10$^5$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Iceberg albedo</td>
<td>dimensionless</td>
<td>0.7</td>
</tr>
<tr>
<td>$k_a$</td>
<td>Thermal conductivity of air (at 10°C)</td>
<td>J s$^{-1}$ m$^{-1}$ K$^{-1}$</td>
<td>0.0249</td>
</tr>
<tr>
<td>$k_w$</td>
<td>Thermal conductivity of water (at 0°C)</td>
<td>J s$^{-1}$ m$^{-1}$ K$^{-1}$</td>
<td>0.563</td>
</tr>
<tr>
<td>$v_a$</td>
<td>kinematic viscosity of air (at 10°C)</td>
<td>m$^2$ s$^{-1}$</td>
<td>1.46 x 10$^{-5}$</td>
</tr>
<tr>
<td>$v_w$</td>
<td>kinematic viscosity of water (at 0°C)</td>
<td>m$^2$ s$^{-1}$</td>
<td>1.83 x 10$^{-6}$</td>
</tr>
<tr>
<td>$D_a$</td>
<td>thermal diffusivity air (at 0°C)</td>
<td>m$^2$ s$^{-1}$</td>
<td>2.16 x 10$^{-5}$</td>
</tr>
<tr>
<td>$D_w$</td>
<td>thermal diffusivity water (at 0°C)</td>
<td>m$^2$ s$^{-1}$</td>
<td>1.37 x 10$^{-7}$</td>
</tr>
<tr>
<td>$R$</td>
<td>Roughness height of the iceberg</td>
<td>m</td>
<td>0.01</td>
</tr>
<tr>
<td>$W_p$</td>
<td>Wave period</td>
<td>s</td>
<td>10</td>
</tr>
</tbody>
</table>

Iceberg erosion from melting tends to occur laterally much faster than vertically, creating ‘tall and skinny’ such that icebergs become unstable and roll-over. Consistent with Mugford and Doweswell (2010) icebergs are assumed to be stable and roll onto their side when the length to thickness ratio is greater than 0.7, (e.g. $L/T > 0.7$). In this case, L and T are instantaneously swapped. Finally, icebergs that ground on the sea floor (when keel depth $> sea floor depth$) remain stationary until they melt sufficiently to begin drifting again.

Fresh water from melting icebergs is released into the surface layer of the MITgcm ocean model at a temperature of 0°C and 0 psu (i.e completely fresh), thus having a cooling and freshening effect on the ocean. Note that both the temperature and salinity of iceberg meltwater is set in data.iceberg.

2. SET-UP INSTRUCTIONS
As MITberg is not officially checked-in as a package for MITgcm you will need to add the following code to several scripts in ‘/MITgcm/model/src/’.
do_oceanic_phys.F
#ifdef ALLOW_ICEBERG
  IF (useICEBERG) THEN
    CALL TIMER_START('ICEBERG [DO_OCEANIC_PHYS]', myThid)
    CALL ICEBERG_MODEL ( myTime, myIter, myThid)
    CALL TIMER_STOP ('ICEBERG [DO_OCEANIC_PHYS]', myThid)
  ENDIF
#endif
I added this after the call to SHELFICE

do_the_model_io.F
#ifdef ALLOW_ICEBERG
  IF (useICEBERG) THEN
    CALL ICEBERG_OUTPUT( myTime, myIter, myThid )
    CALL ICEBERG_WRITE_PICKUP ( myTime, myIter, myThid)
  ENDIF
#endif /* ALLOW_ICEBERG */
This allows MITberg to write diagnostics and pickup files. I added this code after the call to SHELFICE.
external_forcing_surf.F
#ifdef ALLOW_ICEBERG
  #include "ICEBERG.h"
#endif /* ALLOW_ICEBERG */
The above code goes near the top of the script under the “Global variables”. Then you will need to add:
#ifdef ALLOW_ICEBERG
  IF (useICEBERG) THEN
    CALL ICEBERG_FORCING_SURF(
        &    bi, bj, iMin, iMax, jMin, jMax,
        &    myTime, myIter, myThid)
  ENDIF
#endif /* ALLOW_ICEBERG */
This code goes after #ifdef ALLOW_SALT_PLUME and will let the freshwater from the icebergs alter T & S of the ocean model.

packages_boot.F
Under “NAMELIST /PACKAGES/”

add:
    &    useICEBERG,
Under “C-- Default package configuration” add

```fortran
useICEBERG = .FALSE.
```

to the list of packages

```fortran
packages_init_fixed.F
#ifdef ALLOW_ICEBERG
   IF (useICEBERG) CALL ICEBERG_INIT_FIXED( myThid )
#endif
```

I added this code after the call to the #ifdef ALLOW_FLT ).

```fortran
packages_init_variables.F
#ifdef ALLOW_ICEBERG
   IF (useICEBERG ) THEN
      CALL ICEBERG_INIT_VARIA( myThid )
   ENDIF
#endif /* ALLOW_ICEBERG */
```

Again, I added this code after the call to #ifdef ALLOW_SHELFICE).

```fortran
packages_readparms.F
#ifdef ALLOW_ICEBERG
   IF (useICEBERG) CALL ICEBERG_READPARMS( myThid )
#endif /* ALLOW_ICEBERG */
```

This code was added after call to #ifdef ALLOW_SHELFICE

Once you’ve added the above bits of code you’ll need to create a new folder in MITgcm/pkg called ‘iceberg’. Copy the file iceberg.tar to this folder and untar it. You then need to copy file ICEBERG_OPTIONS.h to your verification/\{EXPname}/code/ folder, i.e. the one where you’ll be running your experiments. You can edit this file to turn on/off various aspects of the iceberg model such as multilevel drag, keel model, calving scheme etc. The file should be fairly self-explanatory.

Now add the line “iceberg” to your list of packages in code/packages.conf

```fortran
In data.pkg you will need to write:
```

```fortran
useICEBERG = .TRUE.,
```

This will allow you to turn on/off MITberg once the model has compiled using “make”

Finally copy data.iceberg to the folder where you’ll run your experiments.

You can now try compiling MITgcm.

3. RUNNING THE MODEL
First you must create a file with the initial positions, size, and velocity of the icebergs you wish to simulate. In my example, ‘iceberg_locations.txt’ there are 3 icebergs. You can have up to 15000 listed here. The file format is currently set as I6, 6F9.2 with the 7 input fields as follows:

<table>
<thead>
<tr>
<th>ID number</th>
<th>i-location</th>
<th>j-location</th>
<th>Width (m)</th>
<th>Thickness (m)</th>
<th>u-velocity (m.s⁻¹)</th>
<th>v-velocity (m.s⁻¹)</th>
</tr>
</thead>
</table>

The i and j positions refer to the location of your icebergs on the ocean model domain (the model then works out which tile each iceberg is located on if you’re running in parallel using MPI). To enable MITberg to read this file set niter0=0 in ‘data’ with the name of this file specified in data.iceberg using ‘IcebergLocationFile’. Make sure calving is turned off in code/ICEBERG_OPTIONS.h otherwise the iceberg model will look for the Calving file and ignore this one. If you wish to start your model runs from an ocean pickup rather than initial conditions but don’t have an iceberg pickup file then you can set useIcebergPickup = .FALSE. so that the model will read in the IcebergLocationFile rather than an iceberg pickup.

3.1 data.iceberg
Many of the variables in data.iceberg deal with calving, and initially they can be commented out (as I’ve done in my example). Most of the parameters should be quite self-explanatory as they deal with density and iceberg drag coefficients. There are a couple of things in here to stop the model becoming unstable, one of which is ‘min_size’ which sets the minimum size an iceberg can be before it is assumed to melt. This is included to stop the model becoming unstable with very small icebergs. The other parameter is ‘ibMaxV’ which sets the maximum absolute velocity (ms⁻¹) that an iceberg can move at but this is more important in areas of high ocean velocity when using coarser grids and longer time steps. At the bottom of data.iceberg you will see a list a scalable parameters. Again these are probably not going to be all that useful but they can be used, for example, to locally increase wind speeds, air temperature etc at the iceberg location.

3.2 Iceberg model time step
The iceberg model timestep is set using ‘deltaT_ice’ in data.iceberg. Rather than being the actual timestep in seconds it is the number of times the iceberg code will loop through the dynamics routine (iceberg_adv.F) for every 1 ocean model time step. For example, if your ocean model time step is 300s, then setting ‘deltaT_ice’ to 10 would make the iceberg advection loop 10 times for every 1 model time step, so essentially the advection time step is 30 seconds (300/10), setting to 20 would make the iceberg advection time step 15 seconds, etc. Note that the iceberg thermodynamics routine has the same time step as the ocean model.

3.3 Output data
MITberg writes two output files in binary. The main one is ICEBERG.XXXXXX (where X's refer to the model time). This file provides a snapshot of where all the icebergs in your domain are at any one time, how big they are, how fast they are moving etc. The file is written out at a user defined frequency (in seconds) by setting ‘ib_write_freq’ in data.iceberg. The file is binary, REAL8, and 15000 rows by 12 columns. The columns are:
The other main output file is ibFWflx.XXXXXXXXXX.data. This is a 2-dimensional time average of the volume of fresh water released to the ocean model from melting icebergs (units: kg m\(^{-2}\) s\(^{-1}\)). The file structure is binary (real*4) and has the same dimensions as the ocean model diagnostics written out (i.e. dimensions of the ocean model domain). The output frequency is set by changing ‘IcebergTaveFreq’ in data.iceberg.

### 3.4 Reading output files

Two simple example matlab routines are included to read in a single ICEBERG file and ibFWflx field, and called read_iceberg_data.M and read_FWforcing.F, respectively.

### 3.5 Pickup Files

Iceberg model pickup files are written out at the same interval as the ocean model (set as pChkptFreq in data). There are two pickup files written out at each interval: pickup_iceberg.XXXXXXXXXX.data and pickup_calving.XXXXXXXXXX.data, the latter only being written if the iceberg calving scheme is turned on. The files are both binary and real*8. Files labeled ‘pickup_iceberg’ are 15000 rows by 13 columns and are therefore quite similar in structure to the ICEBERG.XXXXXXXXX files:

<table>
<thead>
<tr>
<th>ID</th>
<th>Tile</th>
<th>Face</th>
<th>calve_slab_counter</th>
<th>i-pos</th>
<th>j-pos</th>
<th>Width</th>
<th>thickness</th>
<th>u-vel</th>
<th>v-vel</th>
<th>Flag</th>
<th>scale</th>
<th>source</th>
</tr>
</thead>
</table>

where calve_slab_counter is the number of seconds since a slab of overhanging ice was calved from an iceberg.

### 3.6 Finally, some things to be consider:

- Iceberg length (L) is always a multiple of iceberg width (W), and is user defined in data.iceberg. By default L = 1.62 x W.
- If iceberg keels touch bottom then iceberg velocity = 0
- Icebergs are not aware of each other. They will not collide and 2 icebergs could, in theory, occupy the same space.
- There is some code to allow icebergs to move along the coasts if they hit land (but are too small to ground on the sea floor). In this case the model will only allow the iceberg to move parallel or away from the...
coast. Turn this on in data.iceberg.

- If sea ice cover > 0.9 then icebergs drift with sea ice model.
- Meltwater is released to the surface layer of the ocean model.

4. MITBERG: MODEL DEVELOPMENT FOR THE FUTURE

- Allow icebergs to collide with fjord walls and each other and ‘bounce off’.
- Release meltwater at different vertical levels in the ocean model.
- Allow user defined iceberg shapes based on observed iceberg profiles to be read in at niter0=0
- Add a dye tracer to iceberg meltwater to track the fresh water released from individual icebergs.