1 Introduction

The NASA GMAO coupled model framework GEOS contains a surface module which computes energy balance in a semi-implicit way. The surface module has its own internal state and resides on an unstructured grid. Due to this constraint, coupling to CICE6/ICEPACK can not be done in the standard implicit manner where surface temperature and flux are updated together with the ice column temperatures. The BL99 thermo solver in CICE4/5 and ICEPACK still supports the explicit coupling via a run-time option *calc_Tsfc*. However, in its current form, the BL99 explicit solver produces oscillating temperature and nonconvergent behavior under certain forcing conditions. The resultant imbalance between conductive flux and internal ice energy change is unacceptable for climate simulations. Here some approaches are sought to remedy the problematic solutions.

2 Description of the problem in the current explicit coupling and a fix

The relevant equation for the energy balance of the top layer snow or ice is in the form of (1)

$$\rho_1 c_1 \frac{\left(T_1^{m+1} - T_1^m\right)}{\Delta t} = \frac{1}{\Delta h_1}\left[K_1^*\left(T_0^{m+1} - T_1^{m+1}\right) - K_2\left(T_1^{m+1} - T_2^{m+1}\right)\right] \tag{1}$$

In the explicit coupling mode, (1) becomes

$$\rho_1 c_1 \frac{\left(T_1^{m+1} - T_1^m\right)}{\Delta t} = \frac{1}{\Delta h_1}\left[F_{ct} - K_2\left(T_1^{m+1} - T_2^{m+1}\right)\right] \tag{2}$$

The first term on the r.h.s. of (2) is a form of 2nd-type (Neumann) boundary conditions for the 2nd-order heat diffusion equation. Although enforcing the flux at the top surface works well under most circumstances, occasionally the solver failed to converge when the top conductive flux is large and positive downward. In these cases, the top layer snow/ice temperature T_1 was near at melting point and the flux forces them to be above 0^oC. Since T_1 has to be capped at 0^oC, the solver failed to converge because the energy change is not consistent with the prescribed flux.

There are a few measures in the literature that could prevent the problem from happening. One is to limit the effective conductivity such that the CFL condition is not violated; the other one is to limit conductive flux itself so the solution does not overshoot. Nevertheless, both methods suffer from inaccurate temperature and fluxes relative to the implicit solver solution.

Instead of imposing F_{ct} in (2), the 1st-type (Dirichlet) boundary condition can also be applied, in the form of (3)

$$\rho_1 c_1 \frac{\left(T_1^{m+1} - T_1^m\right)}{\Delta t} = \frac{1}{\Delta h_1}\left[K_1^*\left(T_{sf}^{fix} - T_1^{m+1}\right) - K_2\left(T_1^{m+1} - T_2^{m+1}\right)\right] \tag{3}$$

(3) is especially useful when surface module determines T_{sf}^{fix} is at the melting point, hence $T_{sf}^{fix} = 0$. In the case that surface is still cold, $T_{sf}^{fix} < 0$ can still be prescribed. In both cases, F_{ct} needs to be recomputed during iterations (which helps solver to converge). When surface is cold, the condition $F_{ct} = F_{surf}$ has to be respected and the difference $F_{surf} - F_{ct}$ can be partitioned to ice-ocean heat flux to conserve energy; otherwise, the thermo solver proceeds as usual.

3 Proposed changes to icepack_therm_bl99.F90

At the top level, we introduce the following parameters to facilitate implementing the new boundary condition.

```
integer (kind=int_kind), parameter :: &
   DIRICHLET = 1,                     &
   NEUMANN   = 2

character (char_len)  :: &
```

```
top_bc  = 'flux' ! top boundary condition type in
                 ! explicit coupling mode (.calc_Tsfc = .false.)
                 ! default is 'flux', indicating the standard Neumann
                 ! type BC where Fcond is prescribed throughout
                 ! 'mixed' is a combination of Neumann and Dirichlet type BC
                 ! depending on the surface conditions:
                 ! when surface is melting or surface is cold but the solver
                 ! did not converge within 10 iterations, switch to Dirichlet
                 ! otherwise, Neumann is used
```

Parameter *top_bc* is used to turn on or off the new boundary conditions. When it is '*flux*', the current Neumann-type BC is enforced, recovering the old behavior. Setting it to '*mixed*' will enable Dirichlet-type BCs under 2 conditions: 1) surface is melting or 2) surface is cold but the solver did not converge within 10 iterations.

We introduce a few additional parameters into the interface of *get_matrix_elements_know_Tsfc* subroutine

```
subroutine get_matrix_elements_know_Tsfc (nilyr, nslyr, &
                                          l_snow,    Tbot,        &
                                          Tin_init,  Tsn_init,    &
                                          kh,        Sswabs,      &
                                          Iswabs,                 &
                                          etai,      etas,        &
                                          sbdiag,    diag,        &
                                          spdiag,    rhs,         &
                                          bnd_type,  Tsf,         &
                                          fcondtopn)


  integer (kind=int_kind), dimension (icells), &
       intent(in), optional ::                 &
       bnd_type          ! top boundary condition type
                         ! 1: Dirichlet
                         ! 2: Neumann (default)

  real (kind=dbl_kind), dimension (icells), intent(in),  &
       optional :: &
       Tsf             ! ice/snow top surface temp (deg C)
```

In the iterative temperature solver section of subroutine *temperature_changes*, a few checks are to be added to turn on Dirichlet BC via *bnd_type* when *top_bc* is set to '*mixed*' and the two conditions above are satisfied. The $fcondtopn$ will also need to be recomputed such that energy conservation check is self-consistent.

Finally, a modification to the top melt calculation in subroutine *thickness_changes* is also required. Currently the top melt is computed as below

```
wk1 = (fsurfn - fcondtopn) * dt
etop_mlt = max(wk1, c0)              ! etop_mlt > 0
```

When our Dirichlet BC is on and surface is cold, we still have $(fsurfn - fcondtopn) > 0$. However, surface melting can only happen when $T\_sfc = 0$. So a conditional check is necessary here to make sure *etop_mlt* is zero if Dirichlet BC is on due to condition 2).

4 Testing

The new Dirichlet BC in explicit coupling requires extensive testing to make sure it produces comparable sea ice and polar climate to implicit solver. This can be done in GEOS couple model framework which uses CICE4. The implicit coupling has been implemented and so the two approaches can be compared directly.

Ideally the test can also be conducted in a stand-alone mode within ICEPACK. Depending on the availability, a small external interface module may need to be built to run ICEPACK in its explicit coupling mode.

5 Summary

The code change as proposed enables explicit coupling of CICE6/ICEPACK to climate models which maintains a surface energy balance module on its own grid. It achieves the goal without resorting to such measures as limiting effective conductivity or conductive fluxes. The resultant high fidelity in temperature and fluxes makes it a viable approach for more coupled models to adopt CICE6 as a component.