H. Pseudo-Code for the Flux Coupler

To understand the more subtle points of the coupler's functionality, it is useful to have a basic understanding of the computer code that is the Flux Coupler. To that end, pseudo-code is shown below. If one wishes to modify the coupler source code, it is strongly recommended that one first study this section in conjunction with studying the source code file, main.f. This should provide a good overview of how the coupler works; a necessary pre-requisite for successful code modification. The actual source code is written almost entirely using standard Fortran 77.

Notice that the Flux Coupler code is configured for a particular time coordination scheme: the ocean model is forced with daily average fluxes, the atmosphere, ice, and land models resolve a diurnal cycle, and the land model is embedded within the atmosphere model. While this configuration is hard-coded within the coupler, it's modular design facilitates code modifications to implement alternate configurations. A variety of time co-ordination schemes can be, and have been, implemented simply by rearranging subroutine calls at the highest level within the main program file, requiring a minimal amount of code modifications or new code.

An explanation of some pseudo-code notation:

- An overbar indicates fluxes averaged over an outer loop synchronization interval: one day. Fluxes without an overbar are fluxes averaged over an inner loop synchronization interval. If an inner loop synchronization interval happens to coincide with one model time step, then we have a degenerate case of time averaging wherein fluxes averaged over an inner loop synchronization interval are actually the time average of only one time sample.
- Subscripts indicate the grid on which a field resides: atmosphere, ice, or ocean.
- x_a, y_a, x_i, y_i, etc. are the component model grids.
- *nstep* is incremented once per outer-loop iteration, once per day, *nspday* is the number of inner loop time steps per day. Models that do not resolve a diurnal cycle (in this example, the ocean model) communicate with the coupler once per day. Models that do resolve a diurnal cycle (the ice and atmosphere/land models), communicate with the coupler *nspday* times per day.
- The coupler stops when $nstep = nstop$.
- The routines map-o2i, map-a2i, etc., map state and flux fields between the various grids. Different mapping techniques are available. For example, when mapping state variables one might want to use an interpolation routine that preserves higher order derivatives, whereas when mapping flux fields one might use a conservative mapping technique (*e.g.,* area averaging).
- The routines mrg-atm, mrg-ocn, etc. merge the various component fluxes into complete sets of input fluxes.
- The routines flx-io, flx-ai, etc. provide the fluxes computed in the coupler between component model pairs. All fluxes that are not model output fluxes are computed by the coupler in these routines.

PROGRAM driver

initial ( ) !— read input namelist —
msg-drvc ( ) !— connect drv to msg-passing environment —

msg-atmi (x_a, y_a, $ndaya$) !— recv initial atm+lnd data: spatial & temporal grid —
msg-icei (x_i, y_i, $ndayi$) !— recv initial ice data: spatial & temporal grid —
msg-ocni (x_o, y_o, $ndayo$) !— recv initial ocn data: spatial & temporal grid —

msg-atmr (Sa_a, Sl_a, $Faia_a$,$Fala_a$,$Faoa_a$) !— recv atm+lnd (initial) state & output fluxes —
msg-icer (Si_i, $Fioi_i$) !— recv ice (initial) state & output fluxes —
msg-ocnr (So_o, $\overline{Fioo_o}$) !— recv ocn (initial) state & output fluxes —

!— acquire drv restart data: note that initial output fluxes from models are over-written —
restart-read ($Faia_a$, $Fala_a$, $Faoa_a$, $Fioi_i$, $Fioo_o$, $\overline{Foxx_o}$)

tcheck ( ) !— verify time coordination —

DO WHILE ($nstepd < nstopd$)

msg-ocns ($\overline{Foxx_o}$) !— send daily avg fluxes to ocn —

tavg-zero ($\sum Foxx_o = 0$) !— initialize daily avg partial sum —

DO $n = 1, nspday$

albd-ice (Si_i) !— impose diurnal cycle on ice albedos —
albd-ocn (So_o) !— compute ocn albedos with diurnal cycle —

!— redistribute radiation from atm based on albedo —
map-i2a (Si_i, Si_a)
map-o2a (So_o, So_a)
flx-cor1 (Si_a, Sl_a, So_a, $Faia_a$, $Fala_a$, $Faoa_a$)

!— compute & map all atm/ice fluxes —
map-a2i (Sa_a, Sa_i)
flx-ai (Sa_i, Si_i, $Faid_i$)
map-i2a ($Faid_i$, $Faid_a$)
map-a2i ($Faia_a$, $Faia_i$)

!— compute & map all atm/ocn fluxes —
map-a2o (Sa_a, Sa_o)
flx-ao (Sa_o, So_o, $Faod_o$)
map-o2a ($Faod_o$, $Faod_a$)
map-a2o ($Faoa_a$, $Faoa_o$)

!— merge atm inputs & send to atm —
mrg-atm ($Faid_a$, $Fall_a$, $Fald_a$, $Faod_a$, $Faxx_a$)

```
            mrg-atm ( Si_a, Sl_a, So_a, Sx_a)
            msg-atms ( Faxx_a, Sx_a)

            !— compute & map all ice/ocn fluxes —
            map-o2i ( Fioo_o, Fioo_i)
            map-i2o ( Fioi_i, Fioi_o)
            map-o2i ( So_o, So_i)
            flx-io ( Si_i, So_i, Fiod_i)
            map-i2o ( Fiod_i, Fiod_o)

            !— merge ice inputs & send to ice model —
            mrg-ice ( Faia_i, Faid_i, \overline{Fioo}_i, Fiod_i, Fixx_i)
            msg-ices ( Fixx_i)

            !— merge ocn inputs, form daily avg partial sum —
            mrg-ocn ( Faoa_o, Faod_o, Fioi_o, Fiod_o, Foxx_o)
            tavg-sum ( Foxx_o, \sum Foxx_o)

            diagnos ( )    !— do diagnostic calculations —
            history ( )    !— write a history file —

            msg-atmr ( Sa_a, Sl_a, Faia_a,Fala_a,Faoa_a)   !— recv atm+lnd outputs —
            msg-icer ( Si_i, Fioi_i)                       !— recv ice outputs —

        END DO

        tavg-done ( \sum Foxx_o, \overline{Foxx}_o)   !— form daily avg ocn fluxes —

        msg-ocnr ( So_o, \overline{Fioo}_o)           !— recv ocn state & output fluxes —

        nstepd = nstepd+1                             !— drv has advanced one day —
        tcheck ( )                                    !— verify time coordination —

        restart-write ( Faia_a, Fala_a, Faoa_a, Fioi_i,\overline{Fioo}_o, \overline{Foxx}_o) !— make drv restart data —
END DO

msg-atmf ( )   !— send final msg to atm —
msg-icef ( )   !— send final msg to ice —
msg-ocnf ( )   !— send final msg to ocn —

msg-drvd ( )   !— disconnect drv from msg-passing environment —

STOP
```