Science merge of CABLE into JULES: Working document

Ian Harman, September 2018

The following is working document with the aim of providing a science-based structure for the merge of CABLE into the JULES repository. It will also, inevitably, discuss technical issues. This has been informed by discussions by staff at the UK Met Office and the current code base to be used in CMIP6.

In positioning this document several critical issues have been considered:

i. While CABLE-JULES is primarily aimed for use within a coupled climate system model, ACCESS, code management requires that CABLE also (and firstly) operates within the stand alone [surface-only] JULES model.

ii. CABLE does not provide all components of surface-atmosphere exchange as required by the atmospheric model, the UM. The initial intent is that CABLE will co-opt the JULES or UM representations for those components – and there are associated technical issues that accompany that aim. The key examples are **surface exchange over sea and sea-ice, river routing, dust** and **aerosols.** (**UKCA** is also involved).

iii. CABLE and JULES have inherently different structures, numerical architectures and time stepping methods. This may pose challenges for the combined model structure.

iv. CABLE needs to satisfy numerical stability issues of the UM (and those JULES representations that CABLE is co-opting).

v. CABLE needs to interface with the UM input, output and memory structures – both at the simulation scale (e.g. ancillaries, restart, namelists and STASH) but also at the module/subroutine level. Variables may need to be quantified by CABLE even if there is no intent of using the associated science because of compilation requirements.

vi. In the first instance the interface between CABLE and JULES should be simple – occurring in a few locations. **Only code for those components of JULES that are to be used by CABLE should occur below the CABLE-JULES interfaces.** This is NOT the case currently**. A review is urgently needed to definitively determine which options need to be supported and which not.**

vii. It is the strong preference of the CABLE community that the science within CABLE remains self-contained and as untouched as possible. Numerically more expensive solutions (e.g. complex CABLE-JULES interfaces) are preferred over splitting the code base or creating new science. This is to facilitate code management and reduce the amount of additional (and unsupported) effort required to get advances in the stand alone CABLE into the coupled code.

viii. It is assumed that a staging of the merge effort, with physical climate first then Earth system components, is feasible.

Overarching structure and time stepping

The CABLE-JULES merge rather inevitably leads itself to follow the existing structure of JULES (and that of ACCESS). There are six key areas of intersection

1. Control level architecture - including memory allocations and input data reading (ancillaries, namelists boundary conditions and restart files).
2. From the radiation code (surf_couple_radiation)
3. From the boundary-layer scaling code (surf_couple_explicit)
4. From the boundary-layer solver code (surf_couple_implicit)
5. From the surface processes code (surf_couple_extra)

The sixth area of importance concerns the assorted diagnostics routines. In addition, many modules in the UM and JULES code base may need to be modified to assist with data passing. The control level architecture (1.) is being handled separately (Eisenberg) and is largely separate from the science considerations (and assumed successful).

In the remainder of this note, items headed by a **bold capital letter** indicate either self-contained Tickets have been identified or pre-tasks need to be undertaken to determine the most appropriate (permissible) way forward.

The current coupled model code base can be retrieved/inspected at
https://code.metoffice.gov.uk/svn/jules/main/branches/dev/Share/vn10.6_CABLE/src/

The current code base for development work within the offline model can be retrieved/inspected at
https://code.metoffice.gov.uk/svn/jules/main/branches/dev/dannyeisenberg/

Both branches include code to facilitate the switch between land surface models – the lsm_id swtich.

> **A:** Why are we converting from integer lsm_id (namelist/metadata) to character lsm_name (code)? CASE statements can work with integers or strings and to assist the metadata/namelist exercise integers are preferred.

Science outline: JULES

Within JULES the four science interfaces handle:

1. Evaluating surface albedo as needed by the UM radiation scheme. (Ideally this would also handle the emissivity information)
2. Providing a first estimate of the surface energy balance and evaluating other surface exchange related exchange variables and fluxes for use within the boundary-layer scaling.
3. Evaluating the surface energy balance at the end of time step as needed for the **co-evolution** of the surface energy balance and boundary-layer states (the fast processes in surface exchange)
4. Evaluating the evolution of the soil temperature and moisture, river flow and other 'slow' parts of the surface physics.

The split between 2. and 3. is possible (and necessary) as the energy balance component of JULES is inherently structured to evaluate both

- the energy balance on surface tiles,
- and a linearization (or equivalently sensitivities) of the tile and grid box mean energy balance to changes in reference level air temperature and humidity.

The advection-diffusion equation of the boundary-layer evolution can then be solved using a numerical scheme that is fully numerically stable and consistent with the energy balance over the time step.

CABLE-JULES differences

CABLE is however structured differently which makes an equivalent posing difficult (if not impossible).

1. CABLE is a multi-tile, dual source model – at this time JULES (remembering that CABLE needs to retain the JULES sea and sea-ice representations) cannot handle dual sources in its numerical methods.
2. The linearization/sensitivity terms for CABLE are not currently available. CABLE is inherently more nonlinear with respect to surface temperature, air temperature and vapour pressure deficit than JULES. These terms may not ever be possible to frame analytically; furthermore the linear tangent approach may be substantially less accurate with CABLE.
3. The energy balance within CABLE is fundamentally, and iteratively, interlinked with the evaluation of GPP (not just the stomatal response). For consistency any linearization should also involve and apply to the fast components of the carbon cycle.
4. Unlike JULES (which applies an implicit correction to the energy balance with respect to air temperature and humidity), CABLE applies an implicit correction to the energy balance with respect to soil temperature, i.e. CABLE cannot evaluate the full energy balance until the soil temperature is evolved.

Implementations within ACCESS

As a consequence of these differences an alternate approach was adopted for ACCESS1.3 in an attempt to maintain both numerical stability and the intent of the sea and sea-ice parts of the JULES code. This involves an iterative approach applied to the land energy balance, in partnership with the existing JULES sea and sea-ice exchange –specifically

- in 2. CABLE determines the energy balance and fast components of the carbon cycle on each land tile and for the grid box.
- in 3. CABLE determines the air temperature and humidity that would have been determined if the grid box energy balance had been maintained over the time step.
- in 3. CABLE re-evaluates the entire land surface energy balance and fast carbon cycle given the updated air temperature and humidity.
- in 3. CABLE retains the (implicit) sea and sea-ice calculations – and hence works over coastal grid cells.
- Section 3 gets called twice due to the ENDGAME reset of the fast physics (double call to atmos_phys2) – this includes the estimation of the updated air temperature and humidity.
- in 3. CABLE also evolves the soil temperature and moisture - and resets this between calls.

The intent is to retain this 'iterative' approach as no alternate is obvious without significant development within CABLE. However, there are proposed developments within UM-JULES (extending the predictor-corrector boundary layer solver to the surface energy balance) that may require this to be reconsidered at some point into the future.

This iterative approach aims to constrain numerical issues caused by disconnecting the surface energy balance and boundary layer but is at the expense of additional computation. There are also some immediate technical challenges posed

> **B:** the fast components of the carbon cycle (e.g. GPP, soil respiration) are updated within the implicit section of the code. Their re-evaluation needs to be passed through the interfaces to the boundary-layer diagnostics routines (affecting surf_couple_implicit, imp_solver and ni_imp_ctl). Similarly additional forcing variables need to be passed to CABLE through the interface (see F)
>
> **C:** the soil temperature and moisture variables are time stepped in the implicit section of the code. Their values need to be passed to the 'extra' part of the code to work with the diagnostics routines.

Current structure

There are likely many technical designs that would facilitate the exchange of information necessary: The current implementation takes the following overarching structure

- UM-JULES interfaces (surf_couple*)
- JULES science used by CABLE
- CABLE-JULES interfaces (cable_*_main and cable_*_driver)
- CABLE core code

These different layers can be broadly classified by the code. UM-JULES interfaces and JULES science code would, ideally, be entirely unchanged the inclusion of CABLE – however currently this is not entirely true. Both layers only utilise JULES variables and follow the JULES structure. The CABLE variant of the JULES science layer involves selective retaining/removing JULES code as appropriate. The CABLE-JULES interface layer utilises both JULES and CABLE variables and primarily acts to pass values between JULES and CABLE variables and vice-versa. The CABLE core is entirely separate code wise from JULES.

As noted earlier, one source of technical difficulties during this process is that some JULES variables that are irrelevant to CABLE nevertheless need to be populated (as they would be within JULES). Many of these variables are not initialised to default variables within the current code base – so it can be difficult to ascertain whether evaluation is necessary without running the model. For convenience/safety it has often been simpler to co-opt JULES code/routines for this purpose.

At the moment there are no instances of pre-compiler flags (#ifdef) specific to the use of CABLE. This may not be sensible (or desirable) – certainly being able to use this facility would assist in the merge.

Revised Structure

Following discussions with the UK Met Office's Land and UM systems staff a joint development process is proposed aimed at establishing a combined code base for JULES and CABLE science yet with clear lines of demarcation. The interface between the models will reside within the four main science coupling points – with everything below that point designated as JULES or CABLE. Routines below the science coupling points, however, may be shared provided they remain entirely common. CABLE/JULES conditions should not be included outside of the science coupling points without consideration of alternate approaches and the agreement of both groups. This is a change from the current (Australian) intent which was to retain a single shared science layer (down to the CABLE-JULES interfaces) but include multiple if (CABLE) statements. **There are resourcing implications for this change in intent.** In relation to the above the new structure is

- UM-JULES interfaces (surf_couple*)
    - JULES science
    - CABLE science
        - JULES modules co-opted but owned by CABLE
        - CABLE core
    - diagnostics

To facilitate the co-hosting of code, the aim is to split the JULES science into land, sea, sea-ice and other (mainly grid box averaging) sections as self-contained code units. A JULES Ticket [#834](#834) has been opened to collate discussions on the re-organisation.

surf_couple_radiation

This is a relatively trivial interface – and is almost in existence. Both CABLE and JULES are required to provide the albedo over sea, sea-ice and for each land tiles. After some development simple CASE statement will facilitate the branch point; in the first instance a minor complication exists and a more complicated approach is needed.

```
Additional CABLE related USE statements pertaining to existing UM_JULES modules

first_call = .TRUE.
select CASE(lsm_id)
    CASE('jules')
        ftsa
        ...              [soot and zenith angle code]
        tile_albedo
    CASE('cable')
        ftsa
        if (first_call) then
            ...
            tile_albedo
            first_call = .FALSE.
        else
            dummy initialise surf_sw_down_cable
            initialise albobs_sc_ij
            cable_rad_main
        end if
    CASE DEFAULT
```

The longer-term aim is to completely remove the (first_call) section in the CABLE branch as this will facilitate restart reproducibility. Some developments are needed within CABLE's core code and the ancillary reading section before that can happen. Specifically these are to

i. dummy initialise the 4-band downwards shortwave. This is needed to determine the current value for the direct-diffuse weighted albedo inside CABLE (land_alb). NB this is a diagnostic only and recalculated later in the explicit branch. However the radiation diagnostics may occur before recalculation – CAUTION.
ii. initialise output variable albobs_sc_ij to a default value.
iii. change a mask condition from being based on non-zero shortwave, to non-negative cos-zenith angle.
iv. ensure that additional ancillary variables needed by CABLE are available by the first call to surf_couple_radiation.

D: Establish a variant of cable's albedo code that does not depend numerically on values for the 4-band shortwave.

On iv: This should be resolved by parallel efforts around the input requirements for CABLE. There may also need to be effort to move some CABLE interface routines from the explicit branch to 'somewhere else' – this is part of the discussions around the D1 array, CABLE's data modules and information passing (see later).

There are four output variables required to be passed through the surf_couple_radiation argument list and one passed via the atm_field_real module. These are

- alb_surft: 4-band albedo for each land point and tile
- albobs_sc_ij: VIS and NIR scaling factor between albedos and observations (UM diagnostic only triggered by l_albedo
- land_albedo_ij: grid box mean 4-band albedos (indexed by pfield)
- land_alb (via atm_fields_real): grid box mean net albedo (i.e. VIS-NIR weighted, direct-diffuse weighted, indexed by grid indexes i and j)

The additional output variable land_alb is not determined by JULES – but does exist in the UM control level code. This may pose a technical challenge to the stand alone code and further edits needed accordingly.

CABLE's radiation code, including the 4-band albedo calculation, is recalculated within the explicit and implicit branches. This is principally for consistency with CABLE's stand alone code. Care is needed to ensure that the inputs are precisely the same in all three calls.

surf_couple_explicit

This is a relatively trivial interface – and is almost in existence. JULES and CABLE are tasked to provide an estimate of the surface fluxes at the start of the time step and evaluate other boundary-layer and surface exchange parameters for use later. The science layers below the control layer however are complicated and need consideration. While it is not possible to be entirely definitive prior to that exercise, the broad structure will be probably be close to

```
Additional CABLE related USE statements pertaining to existing UM_JULES modules

select CASE(lsm_id)
    CASE('jules')
        explicit_jules_land
        explicit_jules_seaseaice
        explicit_jules_finalise
        sf_aero
    CASE('cable')
        explicit_cable_land
        explicit_jules_seaseaice
        explicit_jules_finalise        [or possibly explicit_cable_finalise]
        sf_aero
    CASE DEFAULT
```

explicit_cable_land covers the land based science in sf_expl and sf_exch. The broad structure of that routine is likely to be

```
tile_pts                  [?needed: shared]
windshear                 [shared: new, maybe land specific or in surf_couple_explicit]
cable initialisations
cable_explicit_main
sf_rib + other existing boundary-layer scaling code [shared]
aerosol exchange          [possibly shared]
```

Specific notes:

CABLE requires additional forcing information. Much of this information is currently supplied through existing UM_JULES modules – this may present a technical challenge for the stand alone code. In addition CABLE requires 6 extra forcing variables for which a new data module has been created to short cut the exchange of information directly from the relevant UM routines to CABLE. With some effort inside CABLE it may be possible to remove the need (in surf_couple_explicit) for some of these variables, however the four band shortwave radiation and precipitation information is needed. For full acceptance these may need to be passed via the argument lists – in the case of the radiation information this has to come from rad_ctl!

There are close to 100 separate outputs from surf_couple_explicit. Not all are land based and not all land based outputs will need to be populated by CABLE (e.g. those that are used in the related sections in the implicit and extras branches of JULES). Whether variable needs evaluating, a default value or can be left unevaluated is a task to be tackled at some point (and guidance will be needed from the JULES community).

The land related output variables are (*carbon cycle variables in italics*):

indexing information:	surft_index, surft_pts
tile information:	tile_frac, flake
tile characteristics:	emis_surft, gc_surft, fsmc, fraca, z0h_surft, z0m_surft, canhc_surft, alpha1
grid box information:	smc_gb, z0m_eff_ij, z0m_gb, z0h_eff, rhostar, hcons
tile fluxes:	ftl_surft, fqw_surft, epot_surft, *g_leaf_acc_pft, npp_acc_pft, resp_w_acc_pft, g_leaf_pft, gpp_pft, npp_pft, resp_p_pft, resp_w_pft*
grid box fluxes:	ftl_1, fqw_1, *resp_s_acc_gb_um, gpp_gb, npp_gb, resp_p_gb, resp_s_gb_um, resp_s_tot*
boundary-layer scaling:	fb_surf, u_s, t1_sd, q1_sd, vshr_land_ij, vshr
tile flux information:	rib_surft, ashtf_prime_surft, resfs, resft, rhokh_surft, dtstar_surft, wt_ext_surft
grid box flux info:	rhokm_1, rhokm_land, rhokh, wt_ext
screen level diagnostics:	cdr10m, cdr10m_n, cd10m_n, chr1p5m
dust on tiles:	rho_aresist_surft, aresist_surft, resist_b_surft, u_s_std_surft
dust at grid box:	rho_aresist_ij, aresist_ij, resist_b_ij, r_b_dust_ij, cd_std_dust_ij

All these variables, with the possible exception of some of the 'flux information' variables, are required within ACCESS. Where not available directly from CABLE they are determined from CABLE variables using JULES formulae and code.

Other output variables are determined by CABLE but passed into CABLE data arrays for use elsewhere within CABLE.

In the short-medium term it is recommended that CABLE revisits whether it is appropriate or not to co-opt the science (i.e. JULES formulae) for those output variables bits of JULES that we are not directly providing (e.g. for dust, 10m winds, orographic effects). The recommendation from the UKMO is that we develop our own, consistent, code for this.

Of particular importance for the merge are decisions around the use of the JULES canmodel in setting the dust and aerosol exchange coefficients. The current code base does not appear to be internally inconsistent given the dramatically different representation of turbulent exchange within canopies between JULES and CABLE!

> **I1:** Investigate/understand the science formulation of the various exchange coefficients and explore whether correspondence exists (rather than simple utilisation of the JULES code).

Similarly, CABLE has no representation of sub-grid scale orographic effects and utilises JULES-UM representations.

> **I2:** Investigate/understand the science formulation of the orographic drag/effective roughness and explore whether this is appropriate to use with CABLE.

surf_couple_implicit

This is a relatively trivial interface – and is almost in existence. Both JULES and CABLE are tasked to provide the surface fluxes at the end of the time step. Critically this routine has to function within the predictor-corrector method for the boundary-layer solver. The broad structure will be probably be close to

```
Additional CABLE related USE statements pertaining to existing UM_JULES modules

select CASE(lsm_id)
    CASE('jules')
        ice_fraction code
        im_sf_pt2
        implicit_jules_land
        imlicit_jules_seaseaice
        implicit_jules_finalise    (may not be needed)
        screen_tq*
        winds (corrector step)
    CASE('cable')
        ice_fraction code          [copied]
        im_sf_pt2_cable
        implicit_cable_land
            cable_implicit_main
            land diagnostics
        implicit_jules_seaseaice        [shared]
        implicit_jules_finalise         (or possibly implicit_cable_finalise)
        screen_tq_cable*
        winds (corrector step)          [shared]
    CASE DEFAULT
```

Specific notes:

CABLE requires additional forcing information and requires additional outputs. Much of the forcing information is supplied through existing UM_JULES modules – this may present a technical challenge for the stand alone code. CABLE requires all 6 extra forcing variables for which the data module exists to short cut the exchange information directly from the relevant UM routines to CABLE.

Since CABLE updates its full surface exchange in the implicit branch, the fast components of the carbon cycle are also recalculated. The revised values need to available for output otherwise the diagnostics will be inconsistent with CABLE's evolution. These variables need to be exchanged with both the UM (coupled) and the stand alone outputs.

It may be more scientifically credible to place implicit_cable_land ahead of the call to im_sf_pt2_cable for the purposes of coastal grid cells. The sea component of the coastal tiles should be impacted by the incremented land fluxes; similarly the air temperature used to force the CABLE on the implicit branch should not depend on the increment to the sea fluxes. Both are occurring in the current implementation.

> **H:** Reconsider the sequencing of im_sf_pt2 and cable_implicit_main and any necessary associated changes, especially with regards to the predictor-corrector algorithm.

*It may be appropriate to also split the screen level routines into land, sea and sea-ice components. This may depend on the options that CABLE wishes to retain for the sea and sea-ice calculations. Coastal grid cells will, of course, be complicated.

There are close to 70 separate outputs from surf_couple_implicit. Not all land based outputs will need to be populated by CABLE (e.g. those that are used in the related sections in the extras branch of JULES). Whether variable needs evaluating, a default value or can be left unevaluated is a task to be tackled at some point (and guidance will be needed from the JULES community).

The land related output variables are (*carbon cycle variables in italics*):

tile states:	tstar_surft, tscrndcl_surft, dtstar_surft
tile fluxes:	fqw_surft, ftl_surft, epot_surft, le_surft, radnet_surft, ecan_surft, ei_surft, esoil_surft, surf_htf_surft
grid states:	tstar_land_ij, tstbtrans, tstar
grid fluxes:	taux_land_ij, tauy_land_ij, taux_land_star, tauy_land_star, taux_1, tauy_1, surf_ht_flux_land_ij, surf_ht_flux, fqw_1, ftl_1, olr, ei, esoil, ext*, snowmelt, ecan
others:	rhokh_mix
CABLE:	gs, *gpp, gpp_ft, npp, npp_ft, resp_p, resp_p_ft, resp_s, resp_s_tot, g_leaf*

All of these variables are needed, and determined by default (except tstbtrans and rhokh_mix), within CABLE.

It is important to note that many other CABLE state variables (e.g. soil temperature and moisture) are updated within the implicit branch. These are not made available until the extras branch and communicated between the branches via internal CABLE memory.

tstbtrans is the time since transition and is simple prognostic variable that impacts the calculation of screen level temperatures (including over sea and sea-ice) under some conditions.

*ext is the flux of water extracted by transpiration from each soil layer. In the current implementation CABLE sets this variable to zero. The soil is tiled by default in CABLE and while grid-averaged state variables are determined, this has not be extended to the within-soil fluxes.

It is important to note that many other CABLE state variables (e.g. soil temperature and moisture) are updated within the implicit branch. These are not made available until the extras branch and communicated between the branches via internal CABLE memory.

surf_couple_extra

This is possibly the most complicated interface routine to address – but not the most problematic from a science perspective. The complication arises because the interface handles many different science modules and diagnostics routines, only some of which CABLE wishes to co-opt. The proposal is to only duplicate those routines that each case wishes to use – as per the overarching principal – not handle via namelists. The broad structure is likely to be

```
Additional CABLE related USE statements pertaining to existing UM_JULES modules
Additional working variable – gridded water imbalance = 0.

select CASE(lsm_id)
    CASE('jules')
        …
        revised rivers (takes in gridded water imbalance)
        …
    CASE('cable')
        hydrol_cable            [layer may not be needed]
            hydrology initialisations
            cable_hyd_main      [includes setting gridded water imbalance]
            soilmc              [shared]
        revised rivers
        diagnostics_hyd         [shared?]
        diagnostics_riv         [shared]
        diagnostics_veg         [shared?]
        diagnostics_cablecasa
    CASE DEFAULT
```

Does CABLE co-opt of any JULES vegetation, phenology or crop routines? If yes, then the surf_couple_extra proposed structure will need revision.

Detangling TRIFFID code from the surf_couple routines and the hydrology code is a potential obstacle to this effort as code appears in multiple locations, and is often unmarked.

Despite being the tail of the chain of routines there are still multiple outputs required from surf_couple_extra. Many of these are the updated state variables that need to be passed to the D1 array. With CABLE the primary need to pass arguments into these variables is for restart purposes and to provide information for other parts of the UM that require JULES-like surface variables (e.g. the dust module uses JULES soil moisture information, smcl). Both the evolution and passing between time steps of the land surface state variables is done within the CABLE data structures.

The land related output variables are (*carbon cycle variables in italics*):

tile fluxes:	melt_surft, snomlt_sub_htf, snow_soil_htf, infil_surft
grid fluxes:	snowmelt, dhf_surf_minus_soil, sub_surf_roff, surf_roff, tot_tfall, snomlt_surf_htf , snow_melt, surf_ht_flux_ld
tile state:	catch_snow_surft, t_soil_soilt, tsurf_elev_surft, smcl_soilt*, sthf_soilt*, rgrain_surft, snow_grnd_surft, infil_surft, catch_surft, sthu_soilt*, canopy_surft, fsat_soilt*, fwetl_soilt*, zw_soilt*, sthzw_soilt*
grid state:	hcons, smcl, sthf, sthu, snow_depth, smc_gb, canopy_gb
rivers:	tot_surf_runoff, tot_sub_runoff, acc_lake_evap, twatstor, inlandout_atm_gb, surf_ht_flux_ld, rrun, rflow
veg/carbon:	asteps_since_triffid, z0_surft, z0h_bare_surft, *g_leaf_acc_pft, g_leaf_phen_acc_pft, npp_acc_pft, resp_s_acc_gb_um, resp_w_acc_pft, cs_pool_gb_um*, frac_surft, *lai_pft, canht_pft*

Most of these variables are determined by default within CABLE – it is simply a matter of mapping from one to the other – or, for rivers, letting the existing code use CABLE's equivalent forcing. Some the water cycle fluxes, especially snow related, have no direct equivalent in CABLE. For example, JULES variable hcons, was retained by CABLE in the implementation of ACCESS1.3 and ACCESS-CM2 (evaluated using the JULES routine) – but this was for numerical not scientific reasons.

*tiled soil state variables are included within stand alone JULES but not yet active within the UM. CABLE has implemented a 'temporary' methodology to facilitate its tiled soil variables within the coupled model. This may present a technical difficulty in implementing CABLE into JULES stand alone and coupled using the same technical approach.

others: rad_ctl

This is a potential show stopper!

There is a second location for interface between CABLE and the radiation scheme (separate to surf_couple_radiation). CABLE utilises the four band shortwave radiation (visible-near infra-red, direct beam and diffuse) directly in the energy balance (JULES does not yet). To get that information into CABLE a simple interface module has been included within rad_ctl.

> **E:** Investigate how best to do this exchange of information. It may be that to 'future proof' JULES this capability should be implemented into UM_JULES and hence a common approach adopted.

others: imp_solver, ni_imp_ctl and atmos_phys2

This is a potential show stopper!

On first inspection, the impression was that only modest changes would be needed above the surf_couple layer i.e. inside the UM, to accommodate CABLE. For example, in the current implementation a small number of additional arguments need to be passed between the boundary-layer and surface and a small number of existing arguments need to be changed from INTENT(IN) [JULES] to INTENT(INOUT) [CABLE].

However, more substantive changes than immediately obvious (include the use of new modules) are needed within atmos_phys2. These are principally needed to pass meteorological forcing data used by CABLE directly from atmos_phys2 to the explicit and implicit branches and to get output data from the implicit back out to the boundary layer code for diagnostics purposes.

> **F:** Investigate whether it is better to significantly increase (~20) the number of variables passed through the interfaces (involves changes all the way up to atm_step) or whether modules are permitted to be used *and* would be acceptable (may be via #ifdef statements)

others: river routing scheme

While not strictly at the interface it is worthwhile to note a separate issue/ticket that needs to be facilitated alongside this merge process. In order to globally conserve water, CABLE-in-ACCESS needs to rescale river flow into oceans to account for i) water added to maintain lakes and ii) water lost to inland water basins. ACCESS uses the (deprecated) TRIP river model and has implemented changes into the riv_intcl-riv_1ca and riv_rout-river1a routines. There are no changes to the interfaces and the code has been developed so as to maintain the JULES intent. However the new code does reference an additional module and CABLE 'typed' variables (i.e. this violates the intent that only JULES and JULES-like code should be used in the shared layer). Discussions with the UKMO indicate that the same intent could be implemented via changes in the argument list and a short section of code, and that this would be the preferable method.

> **G:** CABLE's global river flow scaling edits need to be interfaced with the existing JULES routines. Change the variable passing method from a USE module statement to a 'via the argument list' from cable_hydrology

The same (but edited) river code can then be utilised within both JULES and CABLE cases (provided that the namelists prevent some rivers options being used – see **O**)

others: diagnostics_bl and diagnostics_hyd

The fundamental issue here is that CABLE **cannot** utilise STASH codes unless there is a direct equivalence – this is currently the case. This will be prevented by the UM systems team. Examples include STASH 173 (tiled soil respiration) and the addition of soil diagnostics to diagnostics_bl. Through discussion we may be able to change some of the dimensions of the STASH output (e.g. from npft to ntile) as requested by the Australian community.

J: Investigate the recent expansion to tiled soils to see whether this can facilitate the output of tiled diagnostics. New/expanded module to port tiled soil information across to diagnostics_hyd.

K: Investigate the creation of a CABLE section of STASH and creation of a separate diagnostics_CABLE routine.

other issues

L: The dust routine dust_srce needs to be corrected to permit different surface soil layer thickness. Further development to permit the use of tiled soil moisture may be necessary.

M: UKCA assumes and hard-wires a tile ordering that is different to CABLE. CABLE also carries a dedicated tundra tile whereas JULES does not. Consequently additional code has been necessary to manipulate the UKCA surface emissions for use with CABLE.

At this time it is proposed that this code (new subroutines and edited UKCA routines) remains outside the UM-CABLE-JULES merged code as a) further development is likely from the CSIRO perspective, b) it is a relatively small amount of code and separate from the main code and b) work is in train within the UK community to reorganise how surface chemistry is handled.

N: The existing CABLE-in-ACCESS code would ideally be rationalised further to consist of three sections

1. Variable declarations (once per run, common between runs)
2. Ancillary reading, restart reading and other variable default initialisations (once per run, different between runs)
3. Mapping of UM forcing variables and other associated ACCESS time step calculations (every time step)

Parts 1 and 2 need to be managed through the control section of JULES; part 3 has to be within called from within the cable_main routines.

O: Meta-data and rose-stem tests. As noted at the outset **a review is urgently needed to definitively determine which JULES options need to be supported within the CABLE branch.** Once complete the associated JULES namelists and meta-data can be determined. This effort can occur in parallel with (or even ahead of) the code merge and is critical to the establishment of a CABLE rose-stem test. As this is the primary means by which UK Met Office and JULES community researchers/code developers will intersect with CABLE this is a critical task.

P: The recommendation from the UKMO is that drhook diagnostics routines are included wherever sensible. At this time it is proposed that CABLE code that interacts with JULES or the UM have these included, but that the CABLE core (cable_driver and below) does not as we wish to retain functionality of the code away from the JULES-UM systems.

Two further options to explore at some point directly affect the outline of the coupling:

1) Can CABLE be written in the same form as JULES (i.e. a central estimate and linear tangent in T and q)? This would assist in the correspondence and numerical stability of the model.
2) How important are the implicit corrections to the energy balance with respect to soil temperature? Can these terms be ignored – as they are in offline studies – while maintaining water conservation? If so we can run soilsnow directly from the extras branch (like JULES) and remove the %fes_cor issue around restart reproducibility [but SLI is still problematic!].

Potential Gotchas

The usual suspects:

- information not available at the correct level - or on the wrong grid
- variables changing meaning, name, physical units and/or computational dimensions
- misunderstanding of the UM or JULES algorithms
- coding standards
- the moving targets that are the UM and JULES

plus agreement on

- managing tiled soil T, q and CABLE's C-N-P variables - especially in the D1 array, restart and diagnostics.
- CABLE's data arrays, location(s) of initialisation and use of MODULEs more generally.

Current JULES switches

The following is a partial (relevant) list JULES switches/options as set within a work-horse, single column, version of ACCESS. This provides some idea of those elements of JULES that CABLE is co-opting.

l_ctile	true	
l_hydrology	true	
l_pdm	false	
l_top	true	(? not used as far as I can tell)
l_var_rainfrac	false	(ok as CABLE does not address this yet)
l_elev_absolute_height	false	
l_elev_lw_down	false	
l_dolr_land_black	true	
l_aggregate	false	
l_epot_corr	true	
l_triffid	false	

cor_mor_iter	on for low winds
orog_form_drag	set to effective roughness (orog_parm = 0.15)
iscrntdiag == 2	transient radiative effects are included in setting screen level temperature for sea/sea-ice
isrf_cnvgust	on

l_snowdep_surf	true
cansnowpft	true under 1 PFT, false otherwise (!?)
canmodel == 4	
canradmodel == 4	(ilayers == 10)
l_q10	set to temperature
l_dpsids_dsdz	true

Sequencing

Implementing the proposed structure is clearly a challenge. The challenge is made substantially harder as it appears that an incremental implementation while maintaining backwards will be impossible. Nevertheless a sequence of sub-tasks is needed to guide the implementation. The following is a first attempt at identifying 'self-contained' elements of the implementation and a rough sequence:

- Dust bug
- Revise the existing rivers code to accommodate gridded water imbalance associated with unconstrained lake evaporation and river flow into inland water basins. Revise the existing CABLE methodology to populate the gridded water imbalance variable (i.e. move the call from rivers to hydrol)
- Address CABLE's albedo code to be fit for purpose. [consideration of stand-alone code needed]
- Identify 'plumbing' issues for which the technical solutions need to be agreed upon between CABLE, JULES and the UM. Identify issues around diagnostic outputs [consideration of stand-alone code needed]
- Implement coding standards on the CABLE core code.
- Implement CASE statements into the four science coupling routines, with default values for all output variables under the CABLE case.
- Implement the revised structure for JULES
- Implement any additional argument list information needed into/out of the surf_couple routines as needed by CABLE [consideration of stand-alone code needed]
- Develop CABLE equivalents of JULES structure – as informed by revised code and the outline above.
- Implement the broad structure - down to the CABLE-JULES interface routines – for the CABLE case. Implement drhook accordingly. Move the default values for land variables accordingly.
- [following progress on the ancillaries, data i/o, D1 and restart] Implement CABLE data arrays.
- Implement CABLE science routines [stand alone testing can occur]
- Implement any UM (interface) changes necessary to accommodate CABLE.
- Implement necessary changes to the diagnostics routines.

Conversations will be required with the UM systems teams early in the process regarding CABLE's data constructs and use of modules to pass information around. Similarly issues around diagnostic output need to be identified early and the appropriate solution identified. It may be that a dedicated CABLE STASH section is required.

Other areas (within CABLE) which can be deferred but should not be forgotten about:

- Partitioning of the data read sections into parts that are read once per simulation (ancillaries, restart) and parts that are read every time step (UM to CABLE variable exchange).
- Consolidation of the canopy water content science and its location within the code (it is the sole remaining prognostic variable that is incremented in the explicit branch).
- The co-opting of JULES dust/aerosol code. The underlying science of land-surface exchange is different between JULES and CABLE yet the co-opting of the JULES formulations assumes a direct correspondence.
- The use of and current implementation of the correction terms to CABLE's energy balance. There is a cross time-step increment to soil evaporation arising from the correction terms. This will prevent restart reproducibility unless an additional variable is passed into the D1 array/restart.
- Screen level diagnostics routine: This is a single routine that applies on all grid cells. Under some JULES configurations, additional science (decoupling, radiative transitions etc.) are applied – it is not obvious that this science is compatible with CABLE. Nevertheless it is likely that CABLE would wish to retain the full set of options for the sea and sea-ice diagnostics.
- SLI, POP and other 'in development' science requiring further consideration of the UM-JULES interface.
- Writing CABLE in a tangent linear form analogous to JULES.