Spatially and temporally varying heat flux without physics WRF v3.2.1

1 Namelist

Set `isfflx` = 0 (with `tke_drag_coefficient` and `tke_heat_flux`) or 2 (only with `tke_heat_flux`, model calculates surface drag), `sf_sfclay_physics` = 0, `sf_surface_physics` = 0, and `bl_pbl_physics` = 0 (works only for LES and `diff_opt` = 2). Set `tke_heat_flux` to maximum heat flux value (K m s^{-1}) and set `tke_drag_coefficient`.

2 Modifications to subroutine vertical_diffusion_2

List of variables:

- `rt_tendf`: θ physics tendency
- `heat_flux`: namelist defined maximum heat flux (parameter `tke_heat_flux`)
- `heat_flux_min`: minimum sensible heat flux (representing "diffuse radiation", 5% of maximum value)
- `heat_flux_ij`: sensible heat flux at a specific grid point
- `slope`: slope angle
- `slp_azi`: slope azimuth
- `cosi`: cosine of the incidence angle
- `coszen`: cosine of the sun's zenith angle
- `sinzen`: sine of the sun's zenith angle
- `sazim`: sun's azimuth angle
- `xtime`: time since simulation start in minutes
- `mu`: total air mass of column
- `rdzw`: inverse of vertical grid point distance

`slope, slp_azi`, and `xtime` in `vertical_diffusion_2` are passed from subroutine `first_rk_step_part2` (added `grid%slope, grid%slp_azi, grid%xtime` to arguments in call to subroutine `vertical_diffusion_2` in subroutine `first_rk_step_part2`).

Added declarations:

```
REAL, DIMENSION(ims:ime, jms:jme), INTENT(IN) :: slope, slp_azi
REAL, INTENT(IN) :: xtime
REAL :: heat_flux_min, heat_flux_ij, cosi, coszen, sinzen, sazim
```

Add calculation of minimum heat flux below `hflux: SELECT CASE( config_flags%isfflx )` for `CASE (0,2)`

```
heat_flux_min = 0.05 * heat_flux
```

before the loop over `i` and `j`.

For a heat flux that is constant in time but varies with slope angle and orientation change calculation of `rt_tendf` in loop over `i` and `j`. `coszen` and `sazim` determine the slope angle and direction that receives maximum irradiation (the example below uses 60 and 90^o) similar to the position of the sun with respect to the topography.

```
coszen = 0.5  ! 60 deg
sazim = 90. * degrad

sinzen = sqrt( 1. - coszen**2 )
cosi = cos(slope(i,j)) * coszen + sin(slope(i,j)) * sinzen * cos(sazim - slp_azi(i,j))

heat_flux_ij = heat_flux_min + (heat_flux - heat_flux_min) * cosi / coszen
rt_tendf(i,kts,j)=rt_tendf(i,kts,j)  &
                 +mu(i,j)*heat_flux_ij*rdzw(i,kts,j)
```

For a time-varying sensible heat flux replace

```
heat_flux = config_flags%tke_heat_flux
```

with your time-varying definition (example below uses sine curve over 12 h with maximum value = `config_flags%tke_heat_flux` starting after 60 min)

```
IF (xtime .lt. 60) heat_flux = 0.
IF (xtime .ge. 60) THEN
  heat_flux = sin( (xtime-60.) / (60.*12.) * 3.1415926 ) * config_flags%tke_heat_flux
ENDIF
```

For a spatially varying maximum sensible heat flux ("varying sun's azimuth and zenith angle") replace azimuth angle `sazim` and `coszen` with time-varying funtion, e.g.

```
sazim = (90. + 0.25 * xtime) * degrad
```