DALES split-flux instructions

The DALES split-flux functionality allows for a prescribed surface flux to be split between several scalars following . Here, Fn is the surface flux for scalar n, Cn is the concentration for scalar n, Ctot is the total concentration of the to-be-combined scalars (discussed later) and Fprescribed is the prescribed total surface flux.

This functionality is designed to study ammonia (NH3), in a setup where one can distinguishes between the background concentration and concentration from emission source(s). Here, each emission source, including a background concentration, is described by a separate scalar, e.g. three scalars named nh3\_a0 for the background, nh3\_a1 for the first NH3 source and nh3\_a2 for the second NH­3 emission source. The sum of these three scalars is considered the total amount of NH3 in the atmosphere: CNH3 = Cnh3\_A0 + Cnh3\_A1 + Cnh3\_A2. When one wishes to prescribe a constant surface flux for ammonia, it requires the flux to be divided between the three NH3 scalars (nh3\_a0, nh3\_a1 and nh3\_a2). Hence, the split-flux functionality.

# Methodology

The to-be combined scalars are defined in a matrix with default values of -1: sf\_scalars. An example is shown below. This matrix is filled with scalar indices, in the order in which they are defined in scalar.inp.

Each row in the matrix describes the scalar numbers which form a set and over which the surface flux will be divided. The prescribed flux of the scalar in the first column, the base scalar, is the Fprescribed of the earlier equation. The prescribed fluxes of the additional scalars in the set should be equal to the flux of the base scalar. If they are not equal, they will not be used for the split-flux functionality. This is the case for heterogeneous surface patches, like an emission source. All values of -1 will be ignored as well. The sf\_scalar matrix is filled the same as the land\_use input.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sf\_scalars | Base scalar index | Additional scalar index/indices | | |
| scalar sets | **1** | **2** | **3** | -1 |
| **4** | **5** | **6** | -1 |
| **7** | **8** | -1 | -1 |
| **9** | **10** | -1 | -1 |
| -1 | -1 | -1 | -1 |

Exmple for row 1:

Exmple for row 3:

The size of sf\_scalars can be changed in modsurfdata.f90 by changing sf\_dim1 & sf\_dim2.

# Inputs:

IMPORTANT: Make sure to check if sf\_dim1 & sf\_dim2 in modsurfdata.f90 are still sufficiently large! These variables determine the size of sf\_scalars. If it is changed, build a new dales4 executable!

**Namoptions:**

* lsplitflux:  
  .true. if you want to use the functionality
* sf\_scalars:  
  Matrix with the scalar indices for the sets of scalars.
  + In the horizontal (second dimension) are the indices of the scalars that form a set. E.g. one ammonia background concentration scalar (nh3\_a0) and one ammonia emission scalar (nh3\_a1) together form the total ammonia concentration (nh3 = nh3\_0 + nh3\_1)
  + In the vertical (first dimension) are the number of scalar sets.
* Example with 4 scalar sets and the first scalar (#1) is independent.:  
  &NAMSURFACE   
  lhetero = .true.   
  lsplitflux = .true.  
  sf\_scalars(1,1) = 2 (scalar index #2 -> scalar name nh3\_a0)  
  sf\_scalars(1,2) = 3 (scalar index #3 -> scalar name nh3\_a1)   
  sf\_scalars (2,1) = 4 (scalar index #4 -> scalar name nh3\_b0)  
  sf\_scalars (2,2) = 5 (scalar index #5 -> scalar name nh3\_b1)  
  sf\_scalars (3,1) = 6 (scalar index #6 -> scalar name nh3\_c0)   
  sf\_scalars (3,2) = 7 (scalar index #7 -> scalar name nh3\_c1)   
  sf\_scalars (3,3) = 8 (scalar index #8 -> scalar name nh3\_c2)   
  sf\_scalars (4,1) = 9 (scalar index #9 -> scalar name nh3\_d0)   
  sf\_scalars (4,2) = 10 (scalar index #10 -> scalar name nh3\_d1)   
  sf\_scalars (4,3) = 11 (scalar index #11 -> scalar name nh3\_d2)
* **Requirements:**
  + lhetero = .true.
  + lsplitflux = .true.
  + sf\_scalars: NO DUPLICATE scalar indices
  + Variables lsplitflux & sf\_scalars are defines in NAMOPTIONS under &NAMSURFACE

**surface.interactive.inp**

* wsvsurf(n):
  + scalars with the same wsvsurf as the “base scalar” will be have the prescribed surface flux divided between their set of scalars:   
    wsvsurf( *base scalar index* ) = wsvsurf( *additional scalar index* 1 ) =   
    wsvsurf( *additional scalar index* 2 ) = ... = wsvsurf( *additional scalar index* M )
  + Scalars with a different wsvsurf from the “base scalar” will not have the prescribed surface flux between their set of scalars. This is useful for having an emission source.
* Example with 3 scalars: 1 background scalar (no sources) and two emission scalars:  
  #typenr name ... svsurf(01) svsurf(02) svsurf(03)  
  0 “standard” ... -0.1 -0.1 -0.1  
  1 “emission1” ... -0.1 1.0 -0.1  
  2 “emission2” ... -0.1 -0.1 1.5  
  For land\_use = 0: deposition flux is split between scalars n=1, n=2 & n=3  
  For land\_use = 1: deposition flux is split between scalars n=1 & n=3, scalar n=2 gives emission  
  For land\_use = 2: deposition flux is split between scalars n=1 & n=2, scalar n=3 gives emission