Exercises

Day 4

Exercise #4a

- Test the example of dangling pointer:

```
PROGRAM main
IMPLICIT NONE
REAL, DIMENSION(:), POINTER :: p
REAL, DIMENSION(:), ALLOCATABLE, TARGET :: t
INTEGER :: info,Nt
Nt = 100000 ! 10000 ! does it crash for all Nt values ?
ALLOCATE(t(Nt))
CALL RANDOM_NUMBER(t) ! assign some value to the target
p => t ! let p point to t
PRINT*,' p(3) = ',p(3) ! Print some value of p
DEALLOCATE(t,STAT=info) ! Deallocate t
ALLOCATE(t(1000))
p => t ! is this required ?
CALL RANDOM_NUMBER(t)
! NULLIFY(p) !
IF (ASSOCIATED(p)) THEN ! Is p associated if t is deallocated ?
   PRINT*,'p is associated'
   PRINT*,' p(3) = ',p(3) ! Does this always crash ?
ENDIF
END PROGRAM main
```

Exercise #4b

- Extend the program written in the previous exercises:
  - Use IMPLICIT NONE in all subprograms.
  - Update your USE of modules with the ONLY option.
  - Print the date and time at the beginning and at the end of the simulation.
  - Measure the CPU and wall clock time of the central/main DO loops.
  - Create a diagnostic routine that measures the minimum value of the field array as function of time. Print it to a file (fx. ‘diag.dat’). Call this routine every 10 step and open the output file the first time the routine is called (so do NOT save the diagnostics in memory but on disk).