

# **HIRLAM**

**Recoding of physics**

**September 2005**

# Items of the presentation

- 1) The document 'Implementation of HIRLAM physics in ALADIN'
- 2) Some physics interface similarities and differences between HIRALD and HIRLAM
- 3) Things to consider in order to get 'completely' identical low level subroutines in the forthcoming HIRLAM physics and HIRALD physics

# HIRLAM physics in ALADIN

- 1) Background to the evolution of HIRLAM physics is given
- 2) Strategy behind present implementation of HIRLAM physics in IFS/ALADIN
  - a) Adapt to existing coding practice in IFS/ALADIN
  - b) Adapt to existing code structures when interfacing HIRLAM physics.
  - c) Adapt to the reference system of equations of Arôme.
  - d) Initial implementation of HIRLAM physics with only a limited part of options and excluding the surface scheme.

3) Initial experiences with the IFS/ALADIN is described

a) e.g., namelist checking in setup routines insufficient to guarantee consistency – a bigger challenge in IFS due the larger amount of source code.

b) Somewhat complicated to compute 12 pseudofluxes in condensation computations, al lot easier and straight forward for turbulence and radiation.

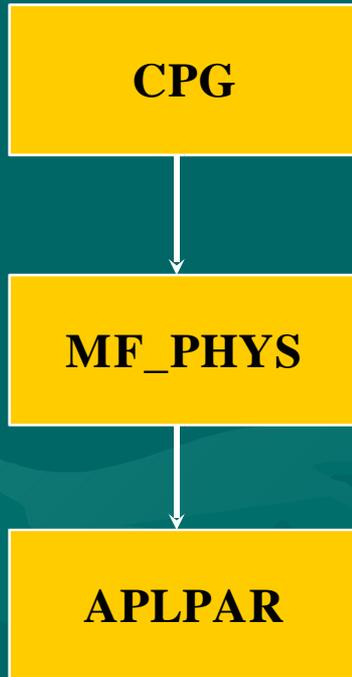
4) Coding issues based on the preliminary work

e.g. nomenclature for constants and parameters

5) Conclusions and Appendix with a brief listing and description of the new and modified subroutines in HIRALD physics.

# Similarities and differences between HIRALD and HIRLAM

## HIRALD



## HIRLAM



# Similarities (1)

- Both systems contain 3 subroutines before calling the low level physics routines.

HIRALD: CPG ---> MF\_PHYS ---> APLPAR

HIRLAM: PHCALL ---> PHTASK ---> PHYS

NB: The actual name of the physics calling routine is APLPARH when HIRLAM physics is used.

# Differences

- CPG has few subroutine arguments because global memory is accessed through USE-statements. In HIRLAM the PHCALL routine contains many arguments which have been passed from other parts of the model.
- The HIRALD system is coded in Fortran 90, and HIRLAM is coded in Fortran 77.
- Additionally, a few specific coding features are seen in IFS/HIRALD environment (see below)

# Similarities (2)

- All interface routines MF\_PHYS, APLPAR, PHTASK , PHYS contain many arguments to become available for the physics low level routine calls

## Things to consider to get identical low level physics routines

Apart from Fortran 90 recoding the following features of the IFS/ALADIN system needs to be adapted to the HIRLAM system:

- `DR\_HOOK`

This subroutine is called in the beginning and at the end of each subroutine and can be used for timing and debugging purposes.

For CBR scheme the call would look like

```
CALL DR_HOOK( ' AVCBR' , 0, ZHOOK_HANDLE )
```

At the subroutine end

```
CALL DR_HOOK( ' AVCBR' , 1, ZHOOK_HANDLE )
```

## Things to consider (2)

2) Use of interface blocks describing interface of that subroutine via an include file. A separate include file is used for every subroutine called from the existing one.

### Example

```
#include "vcbr.intfb.h"
```

This implies that subroutine VCBR is called from the present routine. – The include interface blocks are generated automatically (study needed)

## Additional feature to consider

A mechanism to check the number of arguments in long arguments has been established (recommended)

Example for APLPARH:

Last argument is a character variable 'CDLOCK'

If this does not agree with the last argument 'APLPARH' used in the calling subroutine there is something wrong with the arguments and an ABORT-routine is called.

**THE END**

