

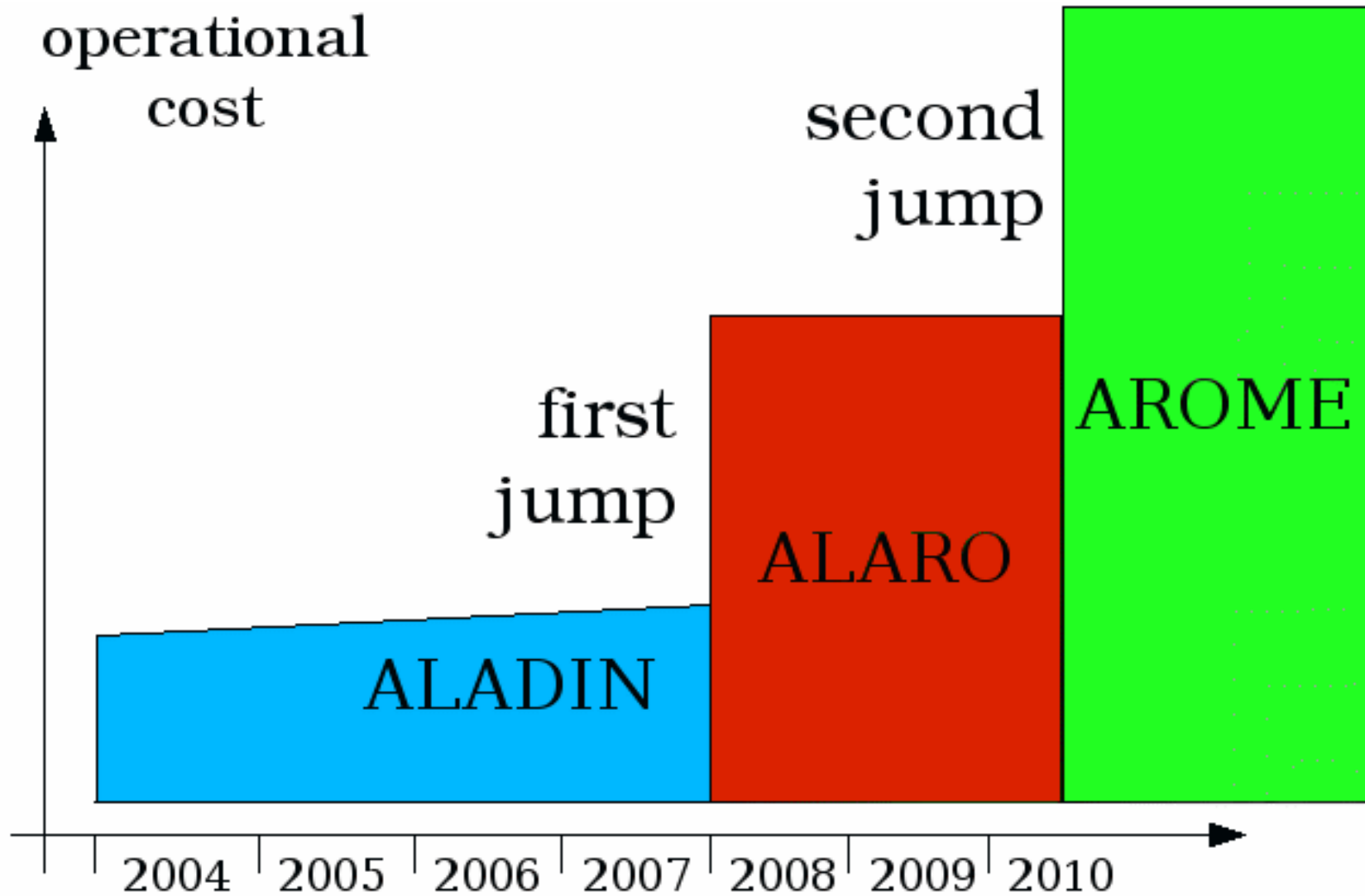
# AROME code management

- a bit of history
- Arome, a member of the IFS/ARPEGE galaxy
- The Méso-NH interaction
- code version management
- code style principles
- validation techniques
- a longer-term view

# Short history of AROME software (1)

- late 80's : birth of the IFS/Arpège project: the ICM, authors: P. Courtier & Mats Hamrud, 1987
  - Integrated Forecast System
  - Action de Recherche Petite Echelle Grande Echelle ('small & large scale research project')
  - InCore Model (i.e. all arrays loaded in memory !)
  - idea: **ECMWF and Météo-France use the same software**, with different switches e.g. stretching in MF. Did not work for physics and analysis [CANARI OI in MF].
- early 90's : birth of the ALADIN project, an extension of IFS/Arpège, lead by J.-F. Geleyn in MF
  - Aire Limitée ADaptation DYNamique (regional dynamical adaptation model)
  - mostly developed by central European visitors in Toulouse in early days
  - idea: as a plug-in library into IFS/Arpège, with minimal specific software
- mid 90's: the great French schism...
  - some aladiners start innovative ALADIN-NH research
  - while the GMME mesoscale group writes its own research model, Méso-NH
- early 90's: ...and the reconciliation
  - French user requirements for a fast kilometric-scale model lead to the creation of AROME, after extensive debate on the possible options (e.g. all ALADIN, or all Méso-NH)
  - **AROME is defined as the merge between some Méso-NH physics and the ALADIN-NH/3DVar software basis**

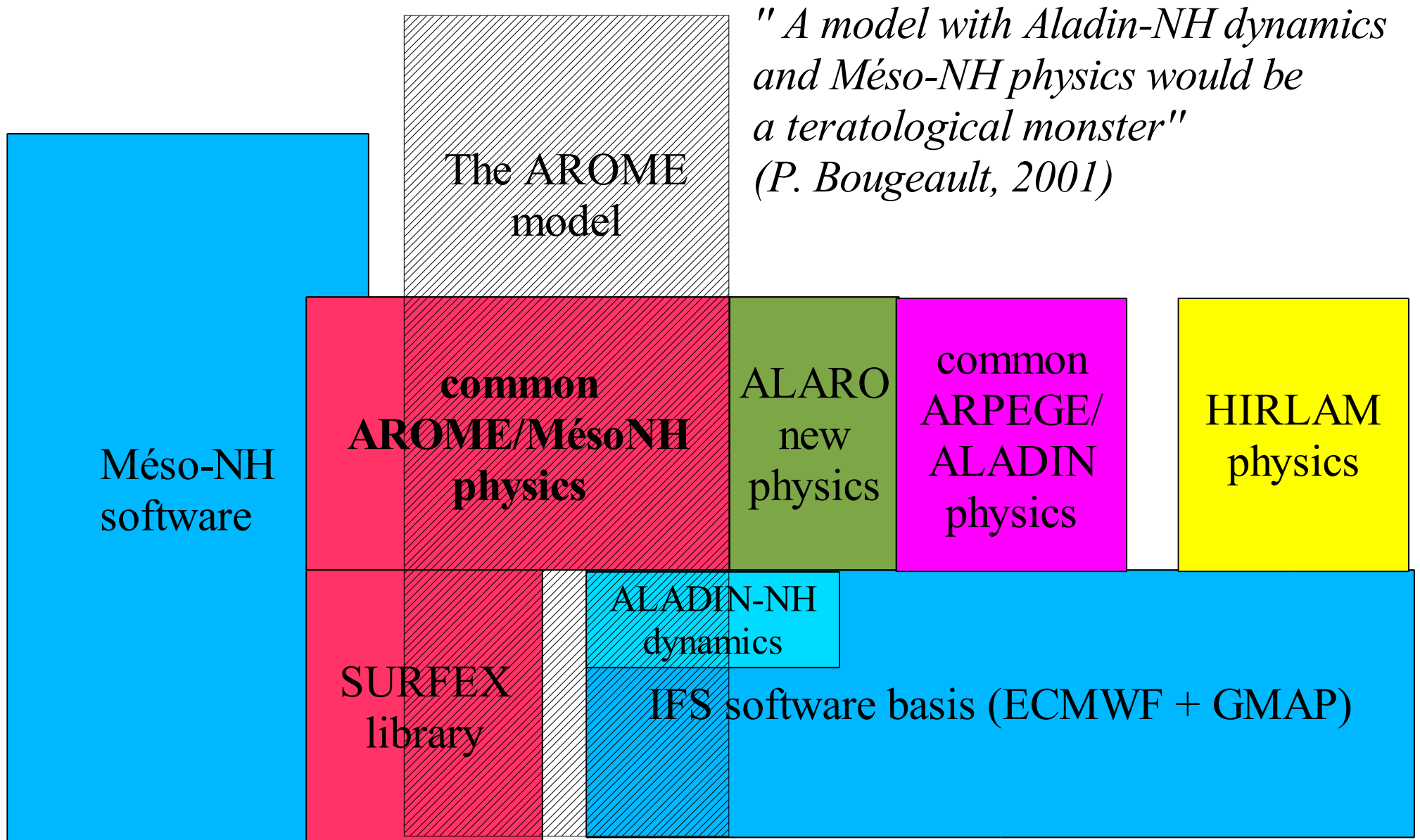
# The Aladin-Arome transition options



# Short history of AROME software (2)

- 2003/2004: birth of the ALARO (model, oops, framework !)
  - idea: AROME will be too expensive for the cohesion of the ALADIN consortium (some institutes cannot afford AROME, others can)
  - ALADIN has to keep improving (with the same physics as ARPEGE)
  - the ALADIN/AROME software architecture must allow a scientific transition between several physics options, including the ALADIN-only and the AROME-only packages
  - development of the INTERFACES project (phys/dyn/diags modular interface) and the ALARO project (new physics parametrizations for the 3-9km resolutions)
- 2005: arrival of HIRLAM
  - after many years of scientific HIRLAM-ALADIN cooperation
  - aim now: software cooperation to develop a NH HIRLAM model
  - based on ALADIN-NH software and 3DVar, but different options (physics, 4D-Var...)
  - extra issue of the future of the HIRLAM synoptic model & assimilation
  - mutual interest for cooperation: recent ALADIN-HIRLAM MoU
- **2003-2005: start of Méso-NH - AROME intergroup activities = continuing development of Méso-NH :**
  - externalization of the SURFEX model,
  - tripartite LA/GMME/GMAP work to upgrade AROME & Méso-NH
  - growing involvement of the French research community in AROME

# The IFS/ARPEGE galaxy

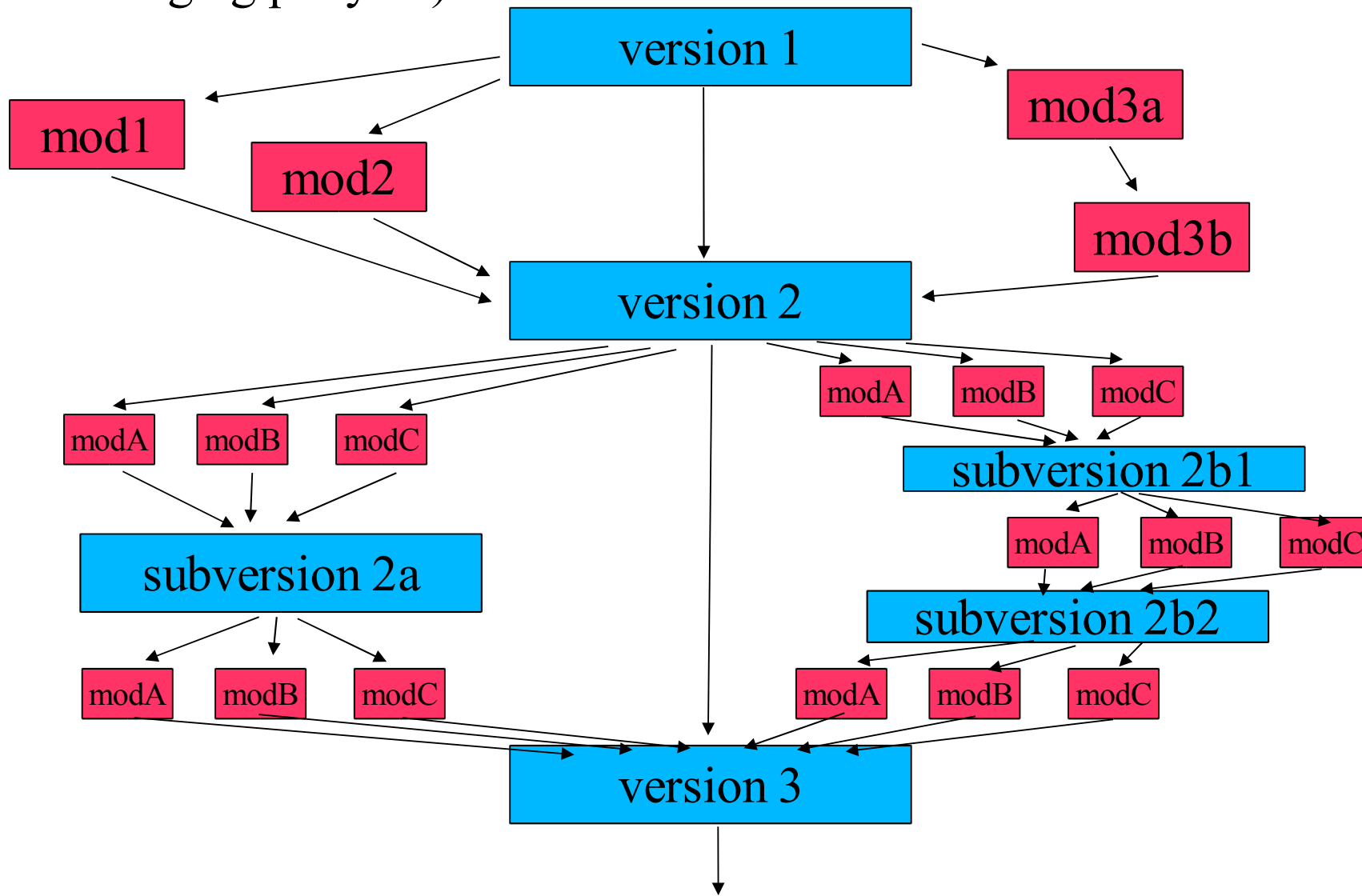


# The Méso-NH interaction

- Méso-NH keeps improving, and the developments are imported into AROME (otherwise AROME would die quickly...). Research labs are better place to do hardcore science than Met. services (who are more concerned with computational & routine meteorological performance)
- AROME and Méso-NH are becoming similar models, but with differences that keep them complementary:
  - AROME is better for real-time and multiple forecasts, because it is computationally cheaper (longer timestep)
  - Méso-NH is better for upstream research, because it has more features to run at very high resolutions (extra physics, interactive gridnesting)
  - AROME is better suited for data assimilation
  - AROME is more operations-friendly
  - Méso-NH is more scientist-friendly
- R&D is being done in both Méso-NH and ALARO, and **AROME will pick what's best, based on operational-style validation**

# Issues of code version management

Basic problem: how to merge modifications developed in parallel by many contributors, in a short time (no more than ~4 months of merging per year)



# Issues of code version management

The solution: a rigorous, intermittent, and intense, "merging" activity where ALL versions are synchronised into official releases

- No One understands the whole code
- Infrequent merges mean painful merges
- Frequent merges mean little time left to improve the software
- The basic merging tasks:
  - set a deadline for contributions
  - gather all contributions into a small merging team
  - identify conflicts, by understanding the mods if possible, or by **extensive validation. Reject or postpone mods if necessary**
  - **resolve conflicts & fix bugs with the participation of the original contributors who are responsible**
- **need for a hierarchy between small and big merges**



# AROME code style principles... & practice

- same style as the rest of ARPEGE/IFS/ALADIN
- rules & advice written down by R El Khatib, M Hamrud, and others(see Aladin Web site)
- (temporary ?) exception for Méso-NH libraries (and Hirlam, probably)
- Need to find a balance between code cleanliness, efficiency and developer-friendliness: *rules are only based on proven practice*
  - make your code **easily readable by others**. Resist 'style anarchy'.
  - **Be modular**, but **do not duplicate existing code** out of laziness - or you will pay for it later
  - never break other's codes: sensitive mods must be **switchable without recompilation**
  - **inactive research** options are ok, but not too much : need to **prune obsolete options** regularly
  - **contribute to code cleaning efforts - or the whole software will die soon**
- The frequency of phasing problems vs. developer productivity vs. operational crashes tells whether we are too tyrannic or too democratic.

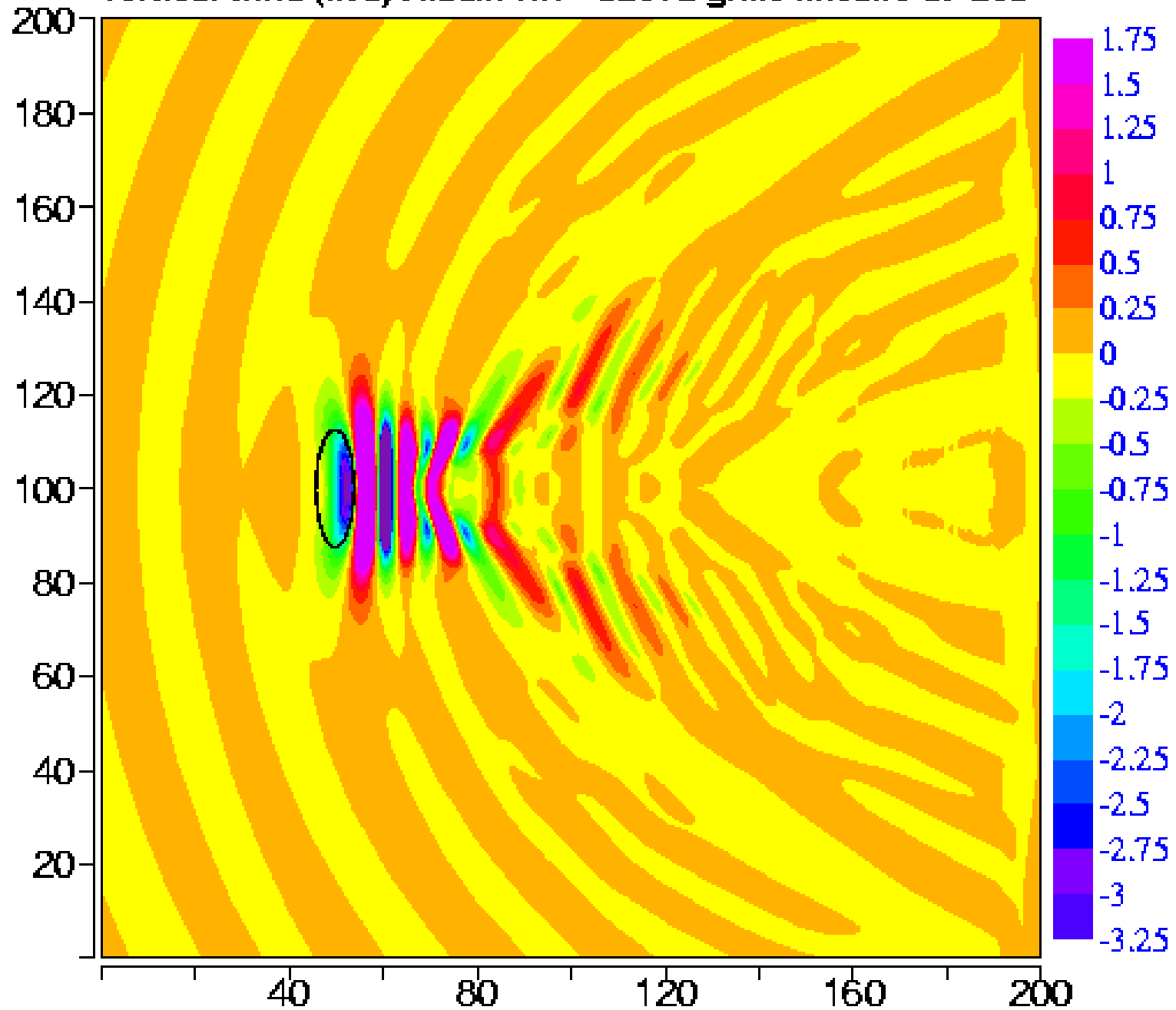
# Validation techniques

- technical tests: non-crashiness, efficiency
- reproduction of earlier results (if possible)
- scientific validation: NWP-style diagnostics: case studies and scores
- new with AROME: more academic validation
- in CNRM/GMAP, 'Mitraillette' procedure to launch hundreds of test jobs



# A dynamical test-case: dry trapped lee waves

vertical wind (m/s) Aladin-NH - SL3TL grille lineaire dt=20s



# A longer-term view

- NWP software is a huge investment. We want to postpone its rewriting as much as possible
- NWP software can no longer be developed (thus, controlled) by a single team. Cooperation is the only way.
- Like a house, it nice to add new things in it, but some cleaning (and even some rebuilding) is regularly needed to keep it going