ALADIN/HIRLAM Strategy Meeting
26–28 April 2016, Toulouse

Code design aspects

Daan Degrauwe

RMI Belgium

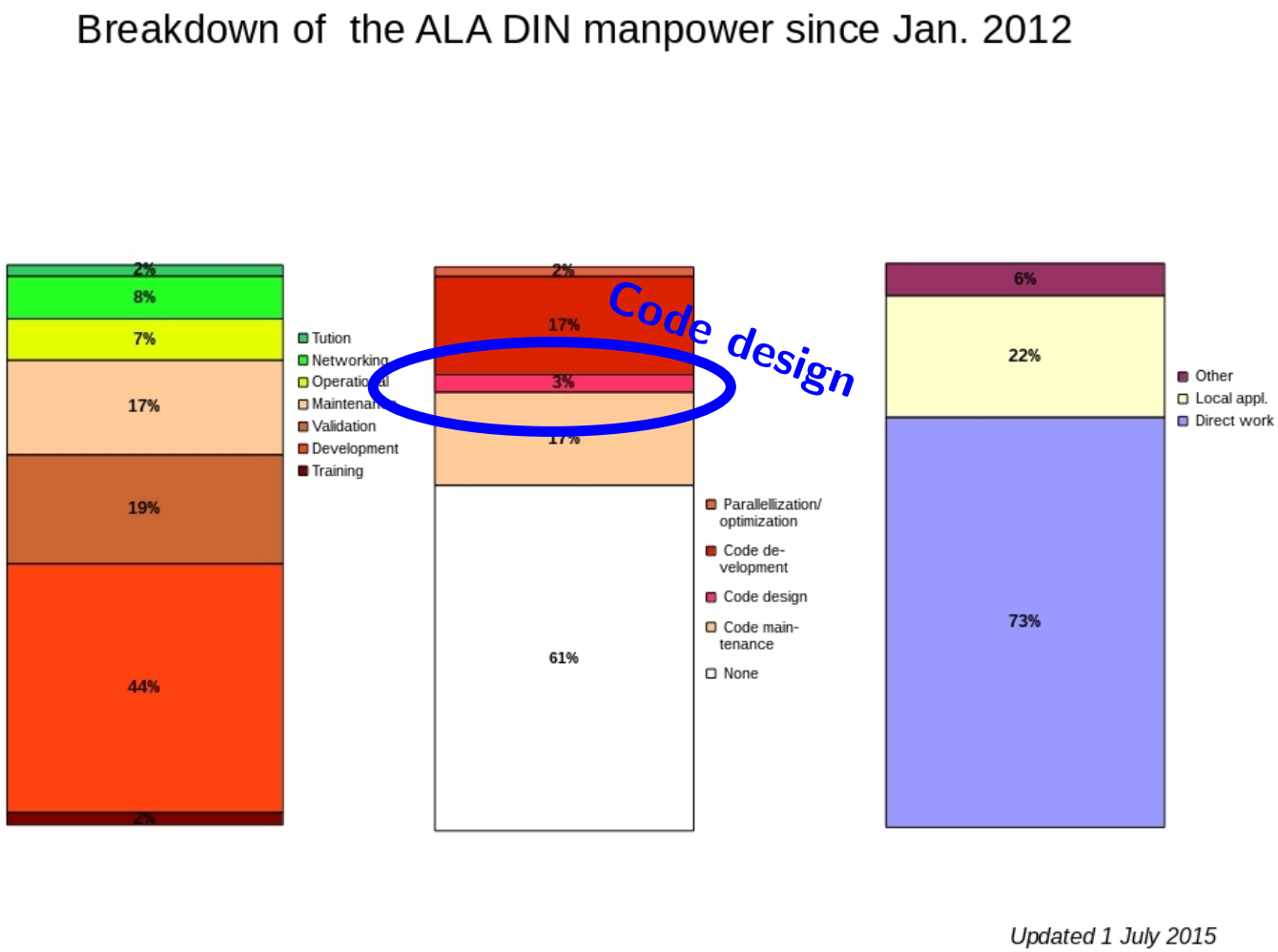
Content

- Motivation
- Example 1: Physics parameterizations
- Example 2: Scalability
- Discussion

Motivation

- Code development is definitively a core program of the ALADIN/HIRLAM cooperation!
- Cooperation means that you'll have to find a way
  - to integrate your own research in the existing code...
  - ...without breaking other people's work.
- The enormous complexity of our code is starting to obstruct new developments. Some actions have been started to remediate this:
  - Data assimilation: OOPS
  - Physics: Convergence actions
  - Scalability: ESCAPE

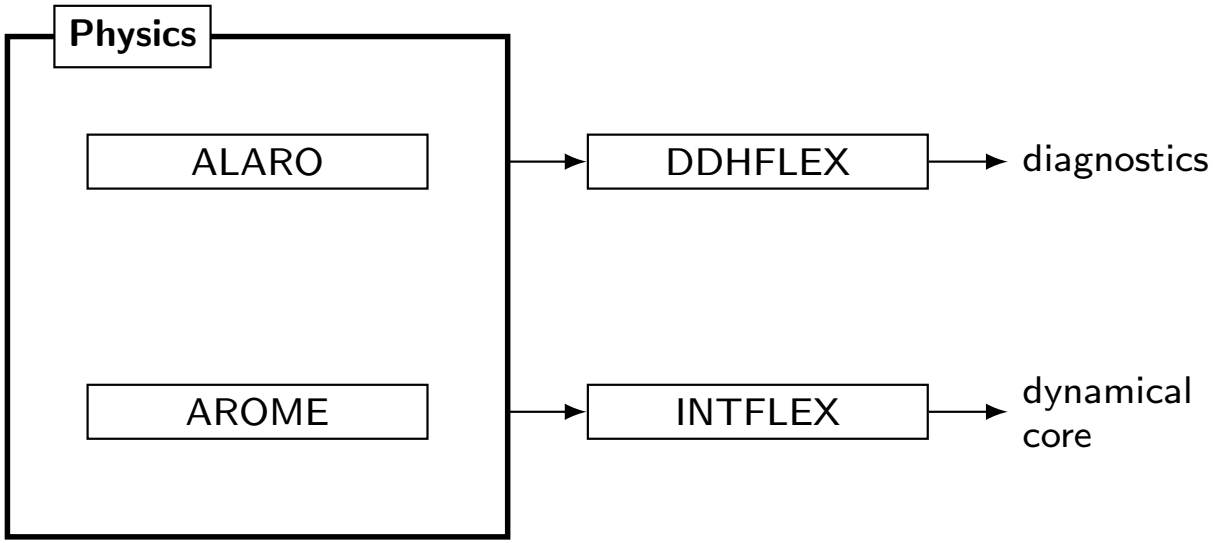
- These actions are related to the *design* of the code, an aspect that seems to have got a bit neglected:



Physics parameterizations

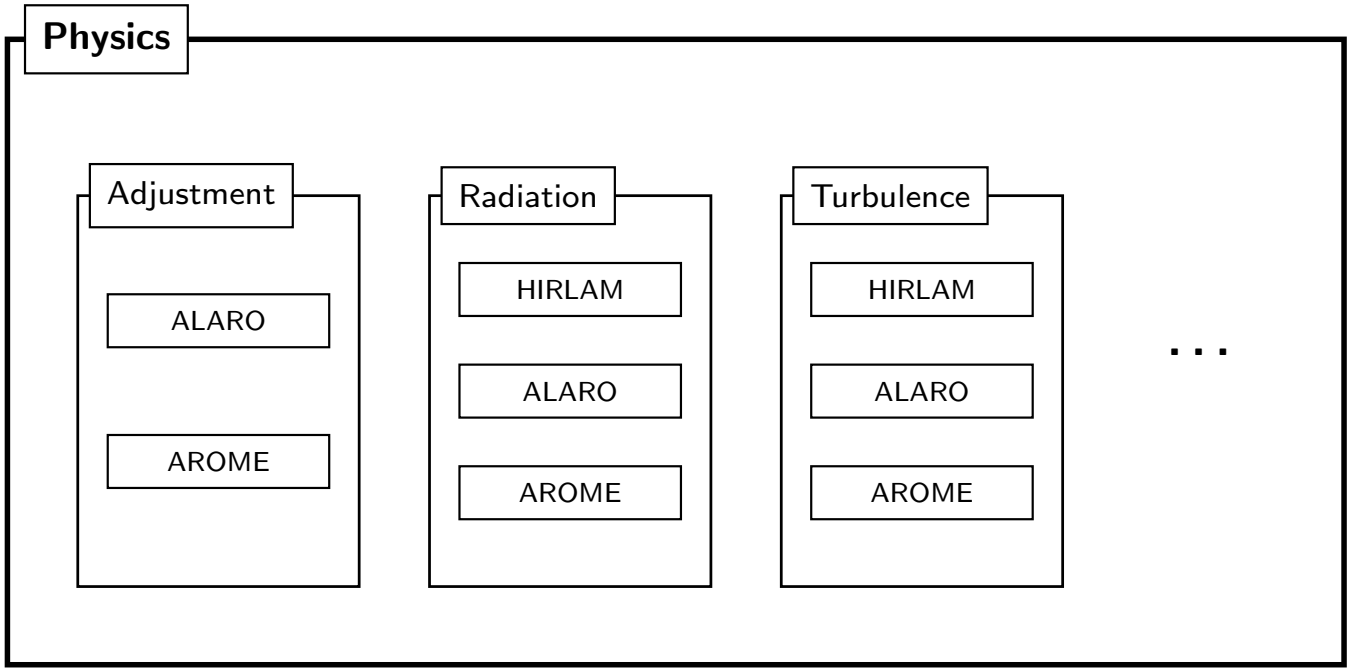
- Efforts have been made to bring the physics packages ALARO and AROME closer together.

  The situation now schematically looks like this:



Physics parameterizations

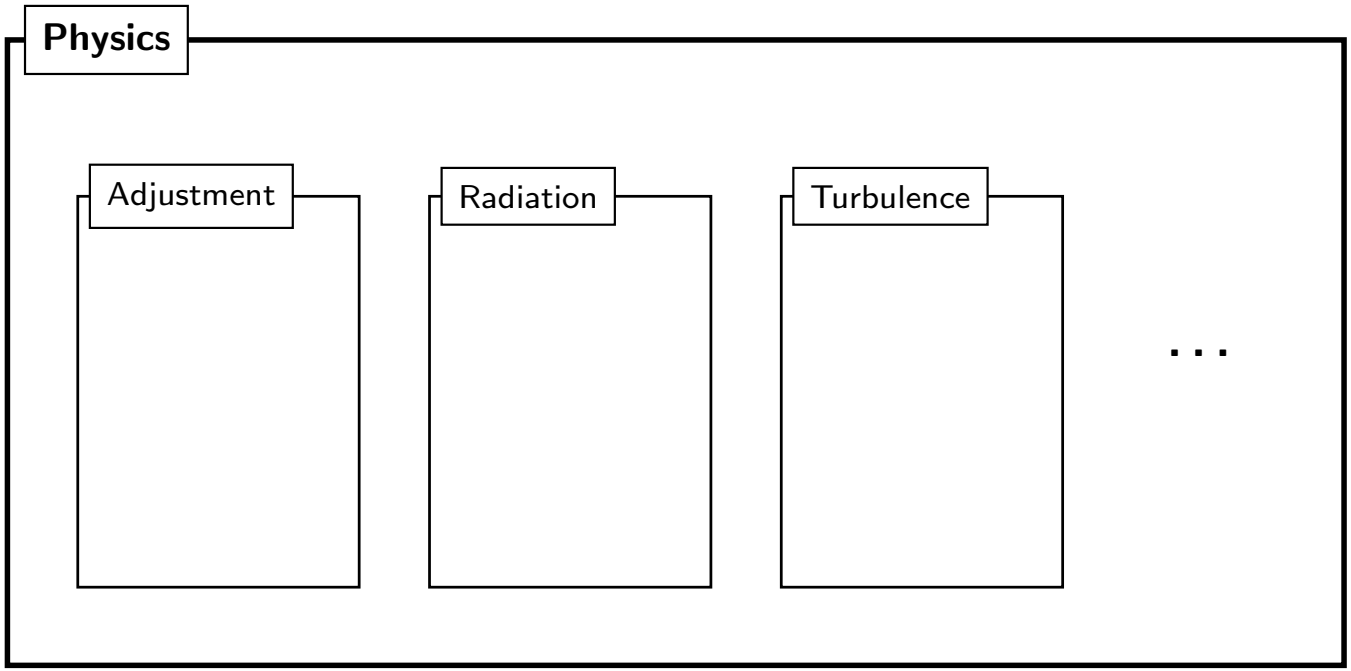
- Efforts have been made to bring the physics packages ALARO and AROME closer together.
- A next step in the convergence between AROME and ALARO would be at the level of the individual parameterizations



- . . . but this requires an analysis of the interactions between parameterizations.

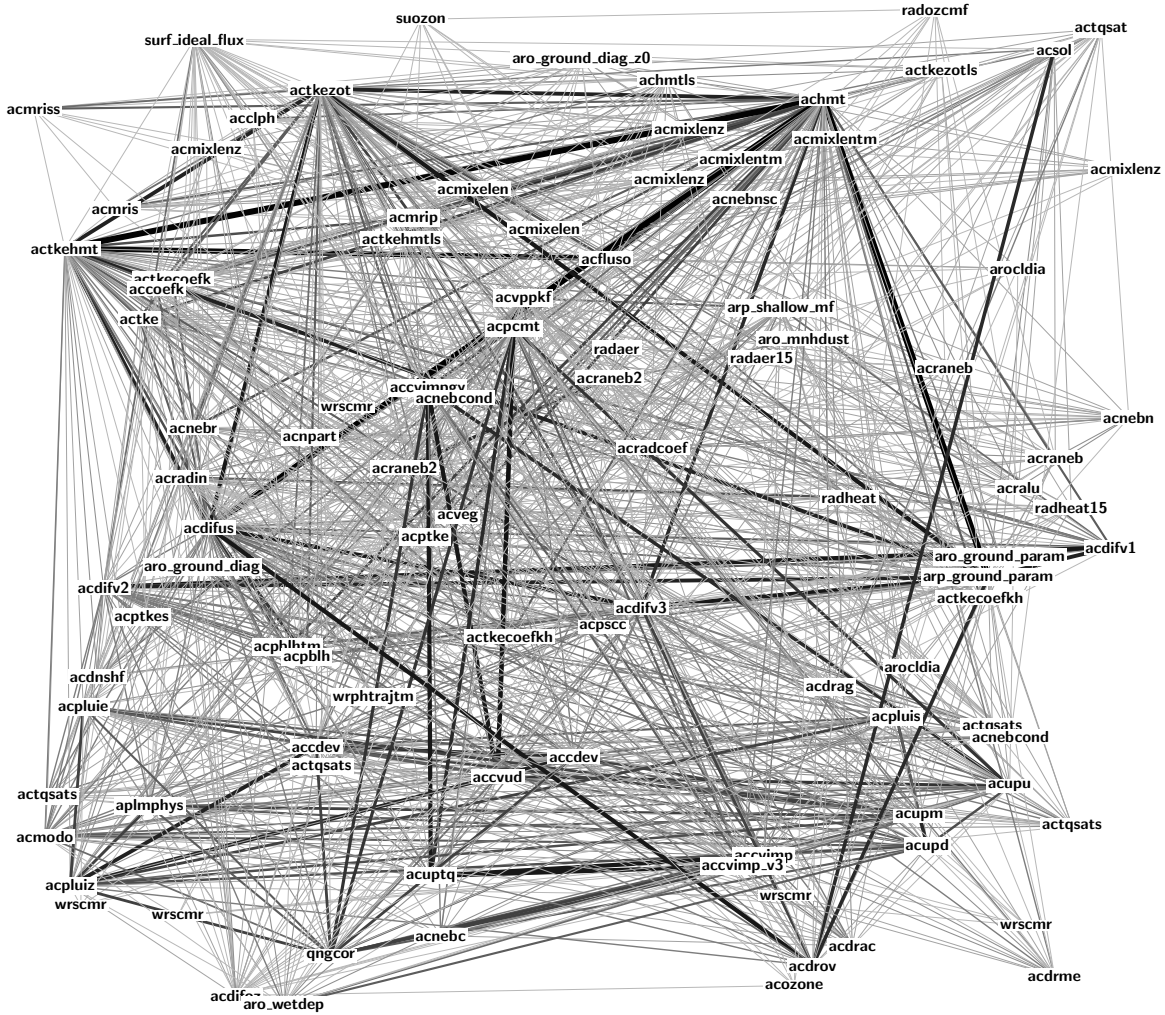
Physics parameterizations

- What you expect to see:



Physics parameterizations

- What you expect to see . . .
- What the code (`aplpar` routine) actually looks like:



Physics parameterizations

- It is clear that this situation hinders developments
- Maintenance is not the only motivation to target a more modular organization:
  - Clean scientific comparison between parameterizations
  - E.g. useful for richer multiphysics EPS systems
- The position of Aladin Code Architect was created to deal with this issue

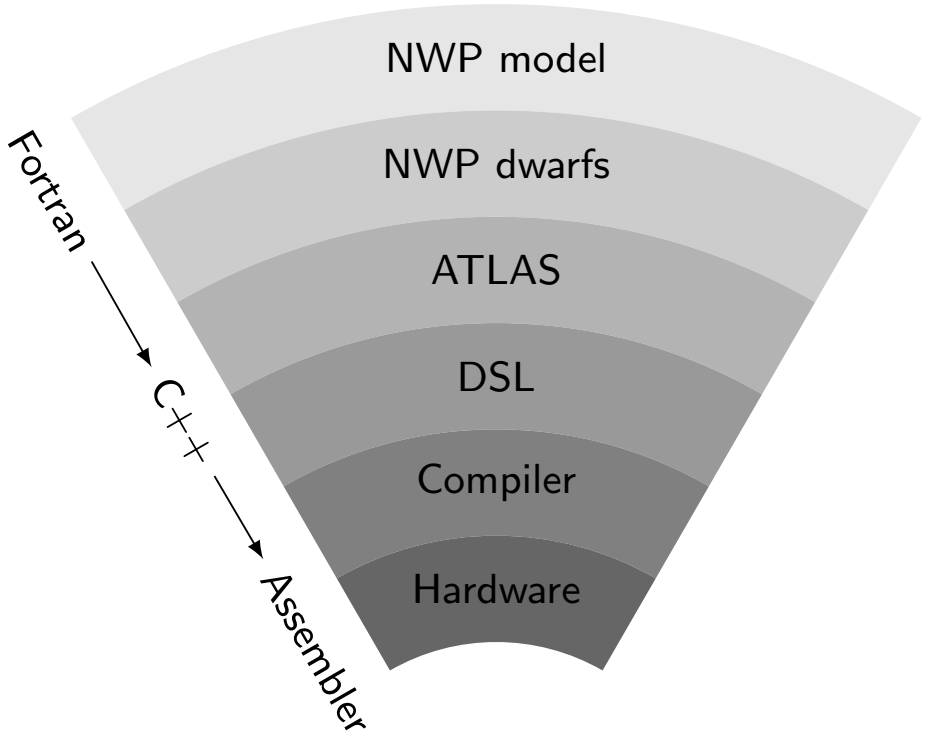
Scalability

- Our spectral semi-implicit semi-Lagrangian dynamical core is quite efficient and accurate.
- However, considering alternatives is quite hard because of
  - the complexity of the code
  - the close relation with the specific hardware platform
- The ESCAPE project, led by ECWMF, tries to tackle this issue by breaking down the atmospheric model in 'NWP dwarfs':
  - spectral transforms
  - advection schemes
  - column physics
  - sparse solvers
  - ...

- These dwarfs are easier to port, optimize and benchmark on emerging hybrid hardware platforms

Scalability

- Part of ESCAPE is about separating science from (hardware-dependend) technicalities.
- This leads to a layered design



ATLAS is a framework for parallel data structures

A Domain-Specific Language (DSL) is a metalanguage which is converted to optimal code depending on the hardware.

- ESCAPE is a research project, where we get the opportunity to get acquainted with new ways of code development: several abstraction layers, extensive modularity, GIT-based cooperation.

Discussion

- Model developments on OOPS, physics and scalability are motivated by a common concern: to deal with the increasing complexity of the code.
- Putting efforts in the design of the code is crucial for this.
- Modularity, i.e. dividing the code in manageable pieces with well-defined interfaces, is the key to a proper code design.

Thank you