Report from stay in Brussels :
8 December - 21 December 2014

executed by: Ivan Bašták Ďurán

co-author: Daan Degrauwe

supported by: Rafiq Hamdi

5.2.2015

LACE

nwp central europe

Contents

Abstract

New code organization of ALARO physics package concerning the calls to surface parametrization, turbulence parametrization and radiation parametrization was proposed.

The aim of the reorganization is to provide a maintenance friendly code via stratification, i.e introduction of more levels in subroutine calls.

This modification should also improve modularity of the code and remove redundant allocations and computations in the main ALARO physics subroutine - APLPAR.

1 Introduction

Main subroutine of ARPEGE and ALARO physics in cycle CY38t1 - APLPAR - contains over 4500 lines. This subroutine is a mixture of initializations, preparatory and complementary computations and subroutine calls. Such code structure makes the implementation of new developments difficult.

The aim of this stay is to propose a code reorganization of surface parametrization, turbulence parametrization and radiation parametrization, which would stratify the code in more subroutine levels. This modification would enable better orientation in the code and simplify further development. Modularity of the code would also increase, which would lead to more straightforward exchangebility of schemes for parameterizations. Additionally redundancy in allocations and computations in APLPAR subroutine could be reduced.

Since the code reorganization in APLPAR concerns both ARPEGE and ALARO physics an agreement from both sides is required in order to unify (on the highest level) the calls to physics packages. Also with the same aim, the new code organisation should take in to account organisation of code in AROME physics - APL_AROME subroutine.

2 Proposal

2.1 Implementation

In order to simple introduce our new code reorganization we put the new code under a switch - LREORG - in APLPAR. In case it is set to .FALSE. old code will be run. This way we can simply present our code proposal and test it versus the old code structure.

This arrangement is only temporary and LREORG will be used only in the first stages of the code reorganization.

2.2 Proposed structure

In the chosen code section (surface, turbulence and radiation parametrisation) we propose to split the code in to four blocks, which will be our fist level of subroutine organisation:

1. **Preparations**
2. **Radiation parametrization**
3. **Shallow convection parametrization**
4. **Turbulence parametrization**

Such structure seems to be simple, but there exists a multitude of schemes, which can be used in ALARO physics. Also possible unification with AROME physics will introduce more options. So in the reorganization it is crucial that all required dependencies will be satisfied. Then we propose following code structure (not all details are displayed):

1. **Preparations**
   (a) **Surface parameters that are not turbulent-specific**
   (b) **Calculation of Scc (Shallow convection cloudiness)**
   ACMRISS, ACMRIS, (first part of) ACMRIP, (ACMIXELEN, ACMIXLENZ for moist AF scheme)
2. **Radiation parametrization**
3. **Shallow convection parametrization**
4. **Turbulence parametrization**

(a) **Diffusion Coefficients**

   i. Calculation of mixing lengths
      ACMIXELEN, ACMIXLENZ, ACMIXLENTM

   ii. Horizontal diffusion
      ACTKECOEFKH

   iii. Calculation of drag coefficients
      (remaining part of) ACTKEHMTLS, ACHMT, ACHTMLS and ACTKEHMT

   iv. Diagnostics
      ACFLUSO

   v. Calculation of stability functions
      (remaining part of) ACMRIP

   vi. Calculation of exchange coefficients
      ACCOEFK, ACTKECOEFK

   vii. TKE Solver and update of Km, Kh
      ACPTKE

(b) **Turbulent fluxes**

   i. Vegetation calculations
      ACVEG

   ii. Old diffusion
      ACDIFUS

   iii. Downward sweep
      ACDIFV1

   iv. Surface scheme
      ARP_GROUND_PARAM, ARO_GROUND_DIAG,
      ARO_GROUND_PARAM, SURF_IDEAL_FLUX

   v. Upward sweep
      ACDIFV2

   vi. TOM's and other stuff
      ACDIFV3, ACDNSHF

3 Example of ACHMT

ACHMT and its TOUCANS counterpart ACTKEHMT (both called from APLPAR according to the value of LCOEFKTKE switch: LCOEFKTKE=.F. ⇒ ACHMT; LCOEFKTKE=.T. ⇒ ACTKEHMT) is a good example for subroutine, which can be stratified. These ARPEGE/ALARO subroutines are used for computation of surface parameters used in turbulence parametrisation. This can be split in to two main parts: roughness lengths - z_0,z_{0h} computation and computation of drag coefficients - C_D, C_H (influenced by stratification via Richardson number - Ri). Additionally at the end of AC(TKE)HMT calls the diagnostics subroutine ACNTCLS (drag coefficients required), which interpolates temperature, humidity and wind components in to heights of standard measurements.

For the purpose of stratification it would be convenient to separate AC(TKE)HMT in two above parts, let's call the corresponding subroutines ACZOT and ACHMTCD, respectively. This would enable flexibility in placement of these two blocks.

In case of activation of TOUCANS (LCOEFKTKE switch) this would mean that the common part (TOUCANS does have the same computation of roughness length) - ACZOT could be run independent from the switch, but ACHMTCD (the differing part) would be chosen by the switch (the name for TOU-

CANS would then be ACTKEHMTCD). Since ACZOT is common for both branches it should be coded only once to avoid inconsistencies in development (in current state branch for TOUCANS can stay in old version while the other branch uses new code).

Also by considering usage of SURFEX, such split could enable computation of roughness lengths in SURFEX (counterpart of ACZOT), but compute the drag coefficients - C_D, C_H by using TOUCANS stability functions (ACTKEHMTCD).

In order to preserve the ’old’ interface of AC(TKE)HMT, so it can be used in the same form as before the split (e.g. ACHMT is used in CANARI) a wrapper subroutine will be prepared, let’s call it ACHMT2. ACHMT2 will simply call the two blocks ACZOT and AC(TKE)HMTCD. If some new development requires additional variables for communication between ACZOT and ACHMTCD, they will be added in ACHMT2 as local variables and put to interfaces of ACZOT and ACHMTCD. This way the ACHMT2 subroutine keeps the same interface as ACHMT.

It is obvious that the two blocks ACZOT and AC(TKE)HMTCD can be used also outside of ACHMT2, which can be used in the new(LREORG=.T.) stratified structure of the physics.

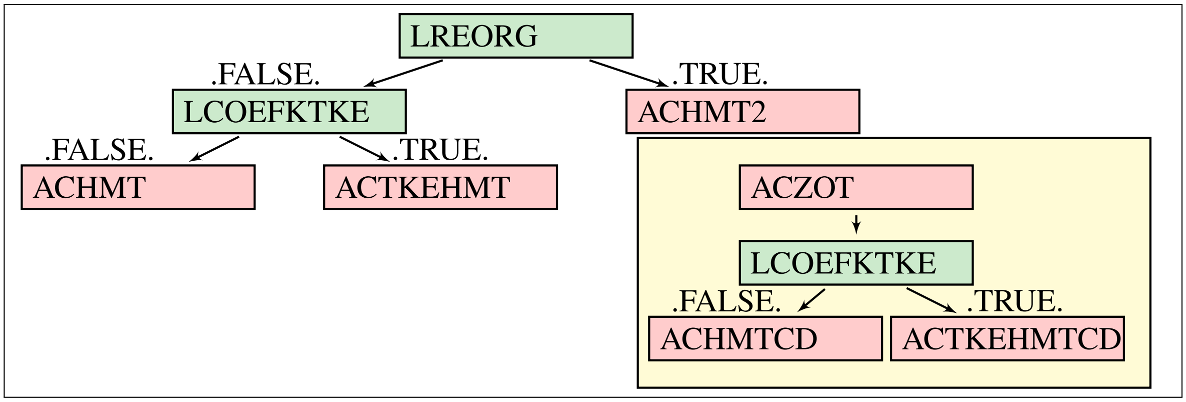

Figure 1: Scheme of code reorganization for AC(TKE)HMT.

4 Planned modifications

In the first stage of the code reorganization we would like to work on the turbulence parametrisation block. At start we would analyze the turbulence subroutines and identify the separable blocks of code (following example of ACHMT) with consideration of possible future reorganization. Then the splits and wrapper functions would be prepared. The parallel code (under LREORG) with unchanged sequence of calls would be verified if it delivers the same results as the original code.

In the next stages, the remaining parametrization should follow this procedure. After that, the four main proposed blocks could be introduced in the APLPAR subroutine and the actual reorganization of the code could begin.