*Regional Cooperation for Limited Area Modelling in Central Europe*

# RC LACE Stay Report

Topic: Debugging and testing TOUCANS module for ALARO-1

Prague 18th March - 30th March 2018

Peter Smerkol

## 1 Introduction

TOUCANS (Third Order moments Unified Condesation Accounting and N-dependent Solver for turbulence and diffusion) is a compact turbulence parameterization, used in the ALARO-1 physical package. TOUCANS integrates several ideas in turbulence parameterization: no existence of critical Richardson number, anisotropy of turbulence, prognostic treatment of mixing length, third order moments parameterization, parameterization of moisture influence and the possibility of 3D parameterization [1].

## 2 Previous work

This stay continued the work of the previous stays [2], [3], which focused on the `ACDIFV3` routine in the TOUCANS parameterization. This routine calculates the TOMs (third order moments) contributions to the turbulent heat and moisture fluxes. In the previous stays, the code was first logically reorganized and cleaned, with negligible numeric effect to the results. This version of code was backphased to cycle 40t1bf6.

Next, the list of known bugs in the routine from the TOUCANS documentation [1] was checked by checking all the code from the `ACDIFV3` routine from the beginning, and in this process, one more bug was found. The bugs were corrected as proposed in the documentation (without taking into account the stability considerations), and all corrections but one was found to be correct. The incorrect one concerns the so-called `ZZZ` bug (point 1. in the list of bugs in [1]). It is incorrect because the code becomes numerically unstable when corrected.

Because of the ZZZ bug, in the next two months, the work continued from home. Since the bug influences the calculation (solver) procedure of the diffusion-like calculations for the relevant TOMSs equations, these equations were theoretically rederived, using [1] as a guideline. During this, some errors in documentation were found, and a correct solver equation was rederived. In the code, this correct equation is already implemented, so the errors in the documentation did not influence the results.

Next, the whole code of the ACDIFV3 routine was converted to mathematical expressions, in order to be able to keep track of the potentially wrong expression for the ZZZ variable (only for the code describing the heat flux, as the code for the moisture flux is directly analogous). The algorithm which uses the Thomas algorithm to solve the equation was rewritten in mathematical terms and compared to the code in the ACDIFV3 routine, in order to find discrepancies from the ZZZ variable.

During this procedure, three adittional bugs were found in the code.

## 3 Current work

The additional bugs were corrected, and the discrepancy between theory and code for the ZZZ variable was checked (see [3]).

The mentioned 'bug' in the ZZZ variable is, that it is not known exactly if ZZZ should be divided by TSPHY or not. ZZZ is an auxiliary variable that enters in solver equations for both turbulent fluxes, on LHS and RHS sides of the equations. After comparing all terms of equations with the derived theoretical expressions, it was found out that the proposed correction from the documentation [1] was correct. However, with this correction, the code becomes numerically unstable.

After some considerations, we (with R. Brožkova and J. Mašek) determined that the reason for this is almost certainly because the ZZZ variable enters into expressions for the ZXSTAM and ZXSTAP variables, which are then put through an algorithm which protects them from non-linear instabilities. This algorithm was written with the previous (wrong) expression in mind, and should therefore be changed to make the code numerically stable again.

This algorithm was coded by J.-F. Geleyn and is not completely understood, as there is no documentation that describes it. So, an effort was made to better understand the algorithm, and analyse the significance of all variables that enter the algorithm on the numerical stability of the solver system.

First, two quick ways were devised (based on correcting the variables `ZXSTAP` and `ZXSTAM` only in the protection part of the code and then correcting them back),but neither improved the solver stability.

Next, we found that the `ZZZ`, `ZKTROV2` and `ZKTROV2Q` variables were incorrectly calculated in the lowest vertical level, if the boundary conditions imposed by the derived equations are correct. After correcting this, the stability of the solver algorithm greatly improved, however the system still crashed after five time steps.

Because of lack of time, we agreed upon further course of action to be done from home. The instability analysis should continue, and if unsuccessful, a new instability protection algorithm should be devised, most likely based on the deep convection solver algorithms from the `ACCVUD` routine.

## References

[1] Ivan Bašták-Ďurán, *TOUCANS documentation*, 15. July 2015.

[2] Peter Smerkol, *RC-LACE Stay report*, Prague, 6th March - 19th March 2017.

[3] Peter Smerkol, *RC-LACE Stay report*, Prague, 26th February - 11th March 2016.