

# ***Interactive comment on “An online trajectory module (version 1.0) for the non-hydrostatic numerical weather prediction model COSMO” by A. K. Miltenberger et al.***

**U. Blahak (Referee)**

ulrich.blahak@dwd.de

Received and published: 4 April 2013

## **1 General comments**

This paper describes a new module for calculating forward trajectories online within the COSMO-model. The COSMO-model is the non-hydrostatic fully compressible limited area NWP-model of the German Meteorological Service (DWD).

Lagrangian trajectory calculations and analyses are a longstanding topic in meteorology and computational fluid dynamics, with an existing wealth of literature. All in all, the authors reference an adequate quantity of literature concerning applications in syn-

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Discussion Paper



optic meteorology, which is their field of expertise. There are less citations of the more technical aspects of trajectory calculations, as well as of applications in other fields like lagrangian dispersion modeling. For the scope of this paper, this should however be fine.

Concerning the new module, the paper briefly describes its numerical method for trajectory computations. The numerical method itself (a scheme ascribed to Sverre Pettersen) is a standard second order method and applied in many other trajectory packages. The main technical contribution of the present paper is the implementation and parallelization of the method for online use in the COSMO-model (presumably with some portability to other NWP models which employ a similar domain decomposition strategy and grid staggering as the COSMO-model).

Further, the authors apply the new module to an interesting synoptic situation over Europe, which has been extensively analyzed in the literature before. They investigate the gain in accuracy of online trajectories (i.e., computed every model time step) as compared to offline trajectories as a function of the model resolution (14, 7, and 2.2 km) and the time step of velocity output available for the offline trajectories (1, 3, and 6 h). Offline trajectories are calculated with an existing software package called LAGRANTO, which also employs the Pettersen Scheme. For the comparisons, the 2.2 km online trajectories serve as (presumably) most accurate reference version, which is reasonable. It is clearly shown that, as one would expect, online calculation can lead to substantially different trajectories compared to coarse timestep offline trajectories, especially at higher model resolution where finer and finer details of the 3D flow structure are explicitly resolved by the NWP model (e.g., gravity waves, mountain lee phenomena, deep convection). Online trajectories are expected to be numerically more accurate, and in that sense the differences might be interpreted as error/uncertainty of the offline trajectories.

The authors strongly advocate the Lagrangian perspective also for very high resolution applications. However, diffusive processes are completely neglected in the new mod-

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)

[Discussion Paper](#)



ule. Therefore it is questionable and will depend on the application, if there is really a big gain in “physical” accuracy, if nonetheless trajectories are interpreted as coherent air parcels over several days. Therefore, what I miss is a more thorough discussion of the influence of diffusive processes. Results of this module might be misleading for, e.g., applications in high resolution dispersion modeling. What is the advantage of doing online trajectories when one could use artificial tracers in the model to simulate advection and diffusion simultaneously?

The authors honestly highlight one remaining technical problem of their new online trajectory module, namely that too many trajectories intersect the ground, especially in mountainous regions. Currently this problem is mitigated by the standard method of simply re-initializing ground-hitting trajectories 10 m above the ground. However, the deeper reasons for this behaviour are not very well explained, and this should be improved in the manuscript. This is important for future users of the module to decide whether this fact is acceptable for their application or not. Without being a detailed expert, I would presume that there can be found something in the literature on this topic.

Taking all these things together, the main contribution of this paper is the technical implementation of existing methods into a parallelized module for NWP-Models and the investigation of the error reduction compared to existing offline methods. While this may justify publication in GMD, the comparatively simple numerical implementation (e.g., no turbulent fluctuations on the trajectories) could have some implications for some applications like dispersion modeling or analysis of strongly turbulent flows in convective clouds. This fact should be elaborated more in the paper.

Therefore I would recommend major revisions.

The paper could be also improved concerning a few other aspects described in the next sections.

[Full Screen / Esc](#)[Printer-friendly Version](#)[Interactive Discussion](#)[Discussion Paper](#)

## 2 Specific comments

### Consequences of the neglect of diffusive processes

As mentioned above, please add in section 1 and/or in section 4 (at places where you think it is appropriate) short discussions on the influence of turbulent diffusive processes and the consequences, if these are neglected and if the trajectories are interpreted as coherent air parcels over long times. Discuss which kind of applications, at which spatial/temporal scales, potentially could suffer from this neglect and to what extent.

One example of a paper applying an autoregressive Markov process to represent turbulent fluctuations on trajectories for high-resolution Monte Carlo dispersion modeling would be

G. Gross, H. Vogel, F. Wippermann, 1987: Dispersion over and around a steep obstacle for varying thermal stratification – numerical simulations, Atmospheric Environment, Volume 21, Pages 483-490.

### Section 2

Please specify somewhere here, that the COSMO-model uses a staggered Arakawa C-grid and is formulated in terrain following and rotated spherical coordinates, because this is important for the spatial interpolation procedure and the portability of the module to other NWP or climate models.

At the end of this section, you mention “the trace variables”, but this important part of the module is not explained further. Please add a short paragraph on this aspect and on its applications, starting perhaps on p. 1229, line 8.

### Presentation of the Petterssen Scheme and its implementation in section 2.1

You present the Petterssen scheme in a way which is customary in the literature as a

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Discussion Paper



kind of predictor-corrector-method. I think your term “iterative forward Euler timestep” from page 1230, line 5, is wrong. However, in my view it would be more enlightening to present it slightly differently:

To solve the trajectory equation  $\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}(t), t)$  numerically forward in time from time  $t_i$  to  $t_{i+1}$ , a second order semi-implicit discretization in space and time is used, which reads

$$\mathbf{x}(t_{i+1}) = \mathbf{x}(t_i) + \Delta t \frac{\dot{\mathbf{x}}(\mathbf{x}(t_i), t_i) + \dot{\mathbf{x}}(\mathbf{x}(t_{i+1}), t_{i+1})}{2} = \mathbf{x}(t_i) + \Delta t \frac{\mathbf{u}(\mathbf{x}(t_i), t_i) + \mathbf{u}(\mathbf{x}(t_{i+1}), t_{i+1})}{2}$$

This is an implicit equation for  $\mathbf{x}(t_{i+1})$ , because the last differential on the right-hand side  $\mathbf{u}(\mathbf{x}(t_{i+1}), t_{i+1})$  depends itself on  $\mathbf{x}(t_{i+1})$ . A common method to solve such a problem numerically is a fixpoint iteration. The above equation is already in fixpoint form, so all we need is a starting value  $\mathbf{x}_0(t_{i+1})$  resp.  $\mathbf{u}(\mathbf{x}_0(t_{i+1}), t_{i+1})$ . Then the  $n$ -th iteration step is given by

$$\mathbf{x}_n(t_{i+1}) = \mathbf{x}(t_i) + \Delta t \frac{\mathbf{u}(\mathbf{x}(t_i), t_i) + \mathbf{u}(\mathbf{x}_{n-1}(t_{i+1}), t_{i+1})}{2}$$

which equals your second through  $n$ -th equation on page 1230. The convergence properties of such an iteration depend on the properties of the flow field and the starting value. In your case, the starting value has been simply chosen as  $\mathbf{u}(\mathbf{x}(t_i), t_i)$ , so that your Eq. (1) is recovered. However, you would be basically free to choose different starting values. For example, near the ground,  $\mathbf{u}(\mathbf{x}_0(t_{i+1}), t_{i+1}) = 0$  could help to reduce the problem of terrain hitting trajectories during the first iteration step.

### Detailed specification of the lower boundary condition(s)

I miss a detailed specification of the lower boundary condition(s) for computing the linear spatial interpolations of the velocity components near the ground. These are decisive for the behaviour of trajectories near the ground and should be clearly mentioned, so that future users may decide on the applicability of the module for their specific application. Add a paragraph in section 2.1 or an own subsection, whichever you feel appropriate.

[Full Screen / Esc](#)
[Printer-friendly Version](#)
[Interactive Discussion](#)
[Discussion Paper](#)


Details of these lower boundary conditions could also be responsible for the problem of too many trajectories hitting the ground (see below). Which boundary conditions are assumed in other trajectory schemes from the literature?

### Convergence criterion for the Petterssen Scheme

As described in section 2.1, your scheme uses a fixed number of iterations for the Petterssen Scheme: “The number of iterations required for convergence depends on the flow situation and the time step, but we find that a default value of three iterations gives mostly satisfactory results. However, it is possible to alter this number via the namelist”.

Here, a convergence analysis of the fixpoint iteration would be helpful. We propose to introduce as an alternative a termination criterion into the iteration, so that the iteration stops if the trajectory increment  $|\mathbf{x}_n(t_{i+1}) - \mathbf{x}_{n-1}(t_{i+1})|$  from one step to the next falls below a predefined threshold. Then do a histogram, after how many steps convergence was reached, and mention if some instances did not converge at all (i.e., iteration did not terminate after some maximum step number).

### Case study in section 3

- Although the presented case study in section 3 has been treated extensively in the literature before, a simple weather chart on the case (e.g., 500 hPa geopotential and surface pressure or whatever the authors feel appropriate) would be very helpful for the reader in section 3.1. This chart could be taken, e.g., from the 14-km run and could also contain the outlines of the smaller model domain for the 2.2 km runs. If you do the plot in rotated spherical coordinates, you can easily make the connection to Fig. 2, where you only plot all trajectories starting south of  $8.5^\circ$  rotated North.
- In section 3.2, again mention the fact that the COSMO-model employs rotated spherical coordinates and specify the coordinates of the rotated North pole, per-

[Full Screen / Esc](#)[Printer-friendly Version](#)[Interactive Discussion](#)[Discussion Paper](#)

haps before "We employ. . ." in line 7. Mention also the "true" model resolutions in degrees, not only in km.

Few people know the "standard model setup of the Swiss weather service" (p. 1234, line 7). For the sake of clarity, it would be good to mention at least some main things about these setups which are not mentioned at the moment, e.g., which parameterizations are used/ not used at which resolution, vertical grid spacing (min, mean, max).

- P. 1239, line 18, statement "Therefore it would be desirable . . .": But you did just that! And sticking to the rigorous definition of trajectories as the solution of the trajectory equation, I do not see an immediate alternative other than comparing to the best possible reference solution, which you tried to achieve by online computations.

### Discussion on the properties of the new module and possible enhancements in the future

In section 4, page 1241, starting with line 24, you discuss some challenges/problems associated with online trajectories in general and your implementation in particular. This is a very important topic for any future user of the module. Here, I suggest to add/modify the following points:

- To reproduce all flow features present in the Eulerian NWP model in the trajectories, the time step for trajectory calculations should obey the CFL criterion of the Eulerian model (Seibert, 1990). For your online trajectories, keep in mind that the model time step  $\Delta t$  only guarantees that the horizontal CFL criterion is fulfilled. But the vertical CFL criterion may become an issue when strong updrafts occur close to the ground, where the vertical grid spacing in NWP models is usually strongly reduced (but not necessarily only there). The numerics of most NWP models is robust in that respect. For example, in the COSMO-model a fully im-

[Full Screen / Esc](#)[Printer-friendly Version](#)[Interactive Discussion](#)[Discussion Paper](#)

explicit second order vertical advection scheme is used, which is stable for vertical Courant numbers exceeding 1.

To mitigate this problem for online trajectories, one could control the timestep by a suitable vertical CFL-criterion, and, if necessary, do a local sub-timestepping for some of the trajectory calculations.

Doing a first quick calculation, a suitable criterion could be something like

$$-\frac{\Delta z}{\Delta t} + |\vec{v}_h| \tan \phi \leq w \leq \frac{\Delta z}{\Delta t} + |\vec{v}_h| \tan \phi$$

where  $\Delta z$  is the local vertical grid spacing,  $w$  the vertical velocity,  $\vec{v}_h$  the horizontal wind vector and  $\tan \phi = \nabla H(x, y) \cdot \frac{\vec{v}_h}{|\vec{v}_h|}$  the orography ( $H$ ) gradient in local flow direction.

- Issue with too many trajectories hitting the ground, starting at p. 1242, line 9:

Although I can follow your argument with linear interpolation smoothing the windfield (presumably especially the vertical component) and being responsible for curving down your example trajectories in figure 6 towards the surface, the exact behaviour near the surface should depend on details of what is assumed for the windfield close to the surface and how this interplays with a possible violation of the vertical CFL criterion (see above, in this case with  $\Delta z = \text{vert. distance to the ground}$ ).

However, I do not understand figure 6 because of missing information. What are the axis labels, what is the grid spacing of the Eulerian framework? 1000 X-units as suggested by the crosses on the orography line? What is the online timestep resp. the “typical” online travel distance during 1 timestep? What exactly is the windfield (you only mention in the figure caption, that it is linear. But at which value does it start at the surface? All this information is necessary to understand why the online trajectory can hit the ground. And I do not understand why it can

[Full Screen / Esc](#)[Printer-friendly Version](#)[Interactive Discussion](#)[Discussion Paper](#)



continue below the surface. Should it not end on impact, i.e., when its position first falls below the orography during the Petterssen scheme iteration?

From this I conclude that the surface-normal wind component does not go to 0 towards the ground and that there is an artificial non-zero velocity below the surface, so that the iteration may continue there and eventually may converge to a final position above the surface. Is that true? If yes, how is this calculated? Constant extrapolation from the lowest model level?

You mention that you apply tri-linear interpolation in space (p. 1230, line 23), and this is perfectly fine. I presume this means that the interpolation is done between the 8 neighbours of an interpolation point in a way that “horizontal” interpolations in  $i$  and  $j$  direction are performed along the terrain-following coordinates (which is both efficient and accurate). If such details would have been mentioned in the text, things would be more easy to understand. You could incorporate this into the required new description of the lower boundary conditions (see above).

On p. 1243, starting in line 5, some possibilities for mitigating this problem are shortly mentioned. Perhaps you could elaborate a little more on the possible changes to the integration scheme close to the surface, in addition to adding turbulent fluctuations. Could these be

- Alternative starting value for the Petterssen iteration (see above)?
- Sub-timestepping near the ground so as to keep the vertical CFL  $< 1$  (see above), using suitable time interpolation of  $u$ ?
- Changes to the lower boundary condition(s)?

[Full Screen / Esc](#)[Printer-friendly Version](#)[Interactive Discussion](#)[Discussion Paper](#)

### 3 Technical corrections

- COSMO = COntortium for Small scale MOdeling; the model's correct name is "COSMO-model". Please check everywhere.
- Please cite the COSMO-model when it is first referenced. There is a newer reference  
M. Baldauf et al., 2011: Operational convective-scale numerical weather prediction with the COSMO model: description and sensitivities, Mon. Wea. Rev., DOI: 10.1175/MWR-D-10-05013.1
- P. 1227, line 28: start new paragraph before "More recently ..."
- P. 1229, line 3: Change "Deutscher Wetterdienst" to "German Meteorological Service (DWD)"
- P. 1237, line 11: The sentence "It appears that ..." sounds strange, please change
- P. 1239, line 23: "Because ..." instead of "As in addition ..."?
- Figure 5: Based on this figure, I find it sometimes hard to identify shape differences of the histograms, especially in the lower left figure. This is because the total number of Fohn trajectories (sum over the histogram) seems to be more or less different for the different configurations. Is that true and if yes, why is that so? In any case, it would be better to normalize the histograms by their total number of trajectories and, to be mathematically correct, also by the class width, to obtain a proper estimate of the probability density functions. Then it would be easier to compare the shapes of the distributions.

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)

[Discussion Paper](#)

