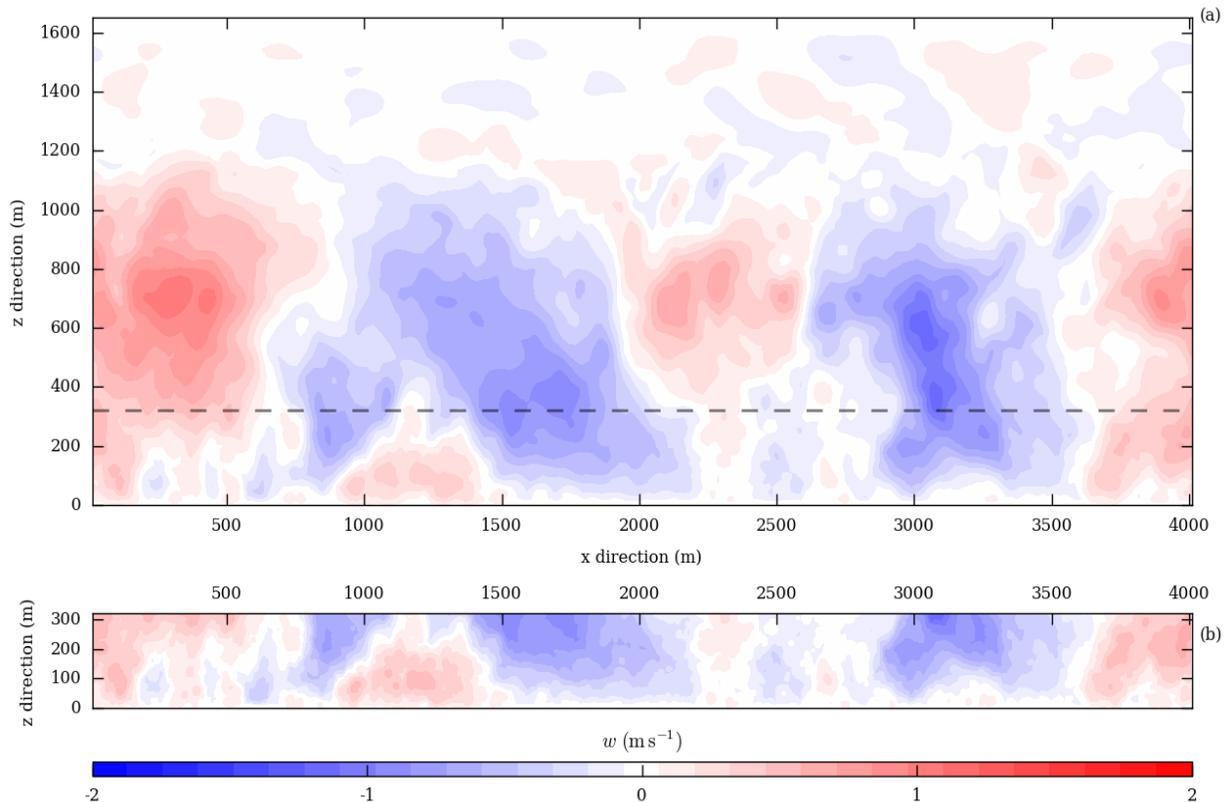


Response to Referee Comment 1 (Dr. Lucas Harris)

It is often hard to see artifacts in a snapshot such as Fig. 4. Would a time-averaged spatial plot show more artifacts?

We agree that the artifacts will not be clearly visible in a snapshot. However, we included the snapshot to qualitatively show that the flow structures can propagate across the grid interface without any distortion and also show that smaller structures are better resolved in the FG. In the figure below, we show a vertical velocity cross-section time-averaged over 1800 s at the end of the 10800 s simulation. There are no visible artifacts.

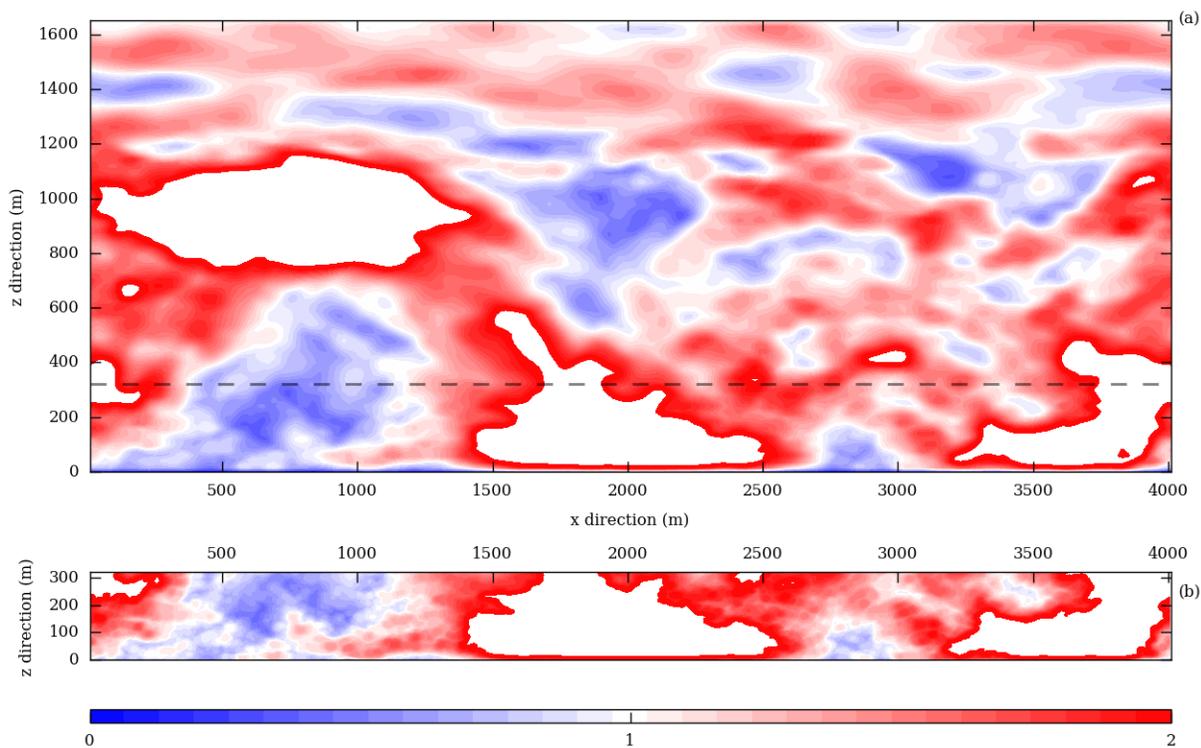


As seen in Fig. 6 there are some artifacts seen in the averaged profiles of the velocity variances, especially in the v variance. Why might v have larger shift in the variance between the two grids? Would these appear if a time-average of the variance were plotted as in Fig.4?

We realized that in the initial runs we had failed to include the compiler flag (`-fpmodel strict`) necessary to make every realization deterministic. As a result, the plots in our initial submission have compiler optimization related random effects, that were noticeable in the variance profile. The results of both the nested and the standalone simulation were affected. We have now ensured that the simulations are deterministic, which improves their comparability. The plots and their description have been updated accordingly.

In the updated spatial velocity variance profile, we observe a marginal shift in u . For, clarity, the profiles

of the variance in the manuscript are based on the spatial variance. We cannot use the same spatial variance for visualizing the variability over an XY cross-section of the variance because the local information is already averaged out to obtain the spatial variance. However, we used a custom user code to output the 30 min averaged temporal variance of u shown below, and we can plot the spatial variability of this quantity. The shifts that are observed in the variance profiles are not noticeable in the contour plot. Since the custom user code has an influence on the computational performance an additional run with identical set-up described in the manuscript was performed to produce this output.



Temporal variance of u averaged over 30 mins.

The issue of reflections of vertically-propagating gravity waves at the top boundary of a vertically-nested grid was considered by Clark and Farley (1984, JAS). In this model the nested upper boundary condition is relatively simple, which is OK for the Boussinesq LES problem presented here in which there are no sound waves and any vertically-propagating gravity waves would be very well-resolved. Do the authors expect that at coarser resolutions (~ 1 km) or if compressible equations are used that the form of the upper boundary condition used here would still yield acceptable results?

In our implementation of the nesting method, we assume that most of the TKE is resolved well down to the inertial subrange, except for the lowest few grid layers. This allows us to use zero-gradient Neumann boundary condition for TKE at top of the nested grid. This assumption will not be valid at coarser resolutions (~ 1 km) and therefore such simulations will not be possible with our method. Furthermore, an advanced sponge layer should also be implemented. We have updated the TKE boundary condition in the manuscript:

“Furthermore, in our implementation of the nesting method, we assume that most of the TKE is

resolved well down to the inertial subrange, except for the lowest few grid layers. This allows us to use zero-gradient Neumann boundary condition for TKE at the FG top boundary.”

I found it strange that the two grids use the same timestep, which could introduce a significant computational burden. Furthermore the communication is done every single timestep, which also introduces substantial overhead due to the amount of message passing needed for the antepolation. Has any consideration been given to a longer communication timestep, or to use a longer timestep on the coarse grid?

Initially we had considered allowing a longer timestep on the coarse grid. However, we did not implement the idea due to parallelization restrictions. We have used a simple parallelization approach where the CG and FG are each assigned a dedicated group of processors. Allowing for a longer timestep on coarse grid also leads to an increased CPU idle time on the CG PE. Therefore, we restricted both the grids to have same timestep. Communicating at every single timestep indeed increases the computational cost. We have not considered limiting the antepolation to every few time integration steps.

Other comments:

- Sec 2.2: Quadratic interpolation is used for scalars. Does this introduce new extrema or negative values into the interpolated fields?

We agree that quadratic interpolation could introduce new local extrema. In our tests we found that the extrema of the interpolated FG scalars had less than 0.005 % difference compared to the CG scalars at the nesting height. However, the horizontal mean of the CG and the FG is equal. We chose the quadratic interpolation of Clark and Farley (1984) because in the interpolation formulation, the coefficient alpha is chosen such the conservation condition of Kurihara et al. (1979) is satisfied and consequently the interpolation is reversible. We have updated the manuscript as:

“The same interpolation formulation is also used to initialize all vertical levels of the fine grid domain at the beginning of the nested simulation. The interpolation is reversible as it satisfies the conservation condition of Kurihara et al. (1979):”

$$\langle \phi \rangle = \langle \Phi \rangle$$

(9)

- Table 2: It seems strange that the SA-F run is more than 2000x more expensive than SA-C despite having only 125x more grid cells. Is this correct?

The SA-F has more grid cells and consequently needs more cores, the number of grid points per core in SA-F is more than 6 times higher than SA-C. Furthermore, the higher resolution in SA-F demands smaller time steps than SA-C. The total number of time steps in SA-F is 5 times more than SA-C. Therefore, the SA-F run is indeed 1000 times more expensive.

We have updated the grid configuration Table 3 to include the grid points per core and the time steps needed in each simulation.

- Sec 3.1: The potential use of a sponge layer is briefly discussed. Do the authors plan to look more into this in future work to alleviate some of the artifacts at the upper

boundary?

Our simplified approach provides reasonable results. Nevertheless, we agree that a thorough investigation of a sponge layer certainly needs more attention to effectively alleviate the artifacts at the upper boundary. We have not planned a thorough investigation of the sponge layer in the near future. However, the main developers of PALM are developing 3D nesting in PALM (similar to our method) and we expect that the artifacts at the boundary will be investigated further.

- The lines in Figures 5–7 are difficult to distinguish because they overlap so much. Perhaps thicker background lines for the SA simulations overplotted by thinner lines for the two grids of the nested grid would work better.

The Figures 5–9 have been updated. As suggested, thicker dashed lines for the standalone simulations and thinner lines for the two nested grids improve the visibility of the overlapping lines.

- Sec 3.4: The authors recommend an odd refinement ratio. Why would this be? The sort of averaging interpolation used should be able to handle even refinements as well.

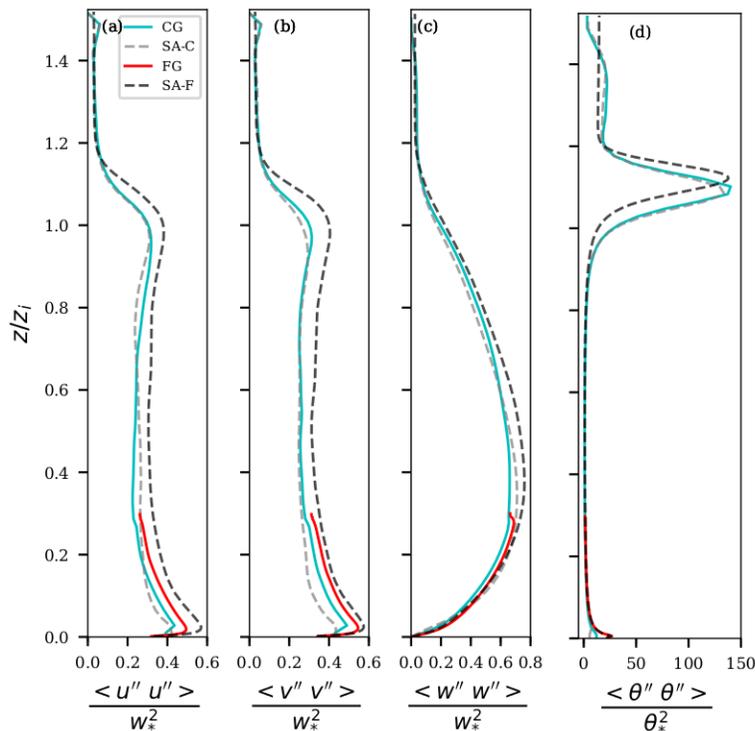
The averaging interpolation can indeed handle both odd and even refinements. However, if the grid nesting ratio is odd, there will be one FG cell of which the center is exactly at the same position as the center of the coarse cell as shown in the updated Figure 1 in the manuscript, due to the Arakawa-C grid in PALM. Such a nested grid set-up is expected to increase the numerical accuracy of the interpolation operation.

Also it is said that the first five gridpoints in an LES are unreliable; why is this, and in which direction?

We meant the first five grid layers in the vertical direction from the surface are not reliable because a lot of turbulence is still sub-grid. This is only based on our experience, due to lack of literature to support this point we have removed this sentence.

Response to Referee Comment 2

Page 11, line 5. You normalized all the profiles using scaling quantity values from SA-F only, rather than values from the respective simulations. Are there any surprises or interesting features when scaling each profile with data obtained from their respective simulations?



The variances of velocity and temperature normalized by the values from the respective simulations are shown below. Since the boundary layer height differs only about 30 m, the difference in the surface potential temperature is less than 1 K and the surface heat flux is constant between the simulations there is no considerable difference in the profiles normalized by SA-F values and the profiles normalized by the values from the respective simulations.

A general comment for all of the vertical profile figures that is relevant here is to use different line styles, in addition to the different colors, to better differentiate profiles that are nearly on top of each other. With this strategy, you should be able to plot additional data without making the plots unwieldy to decipher.

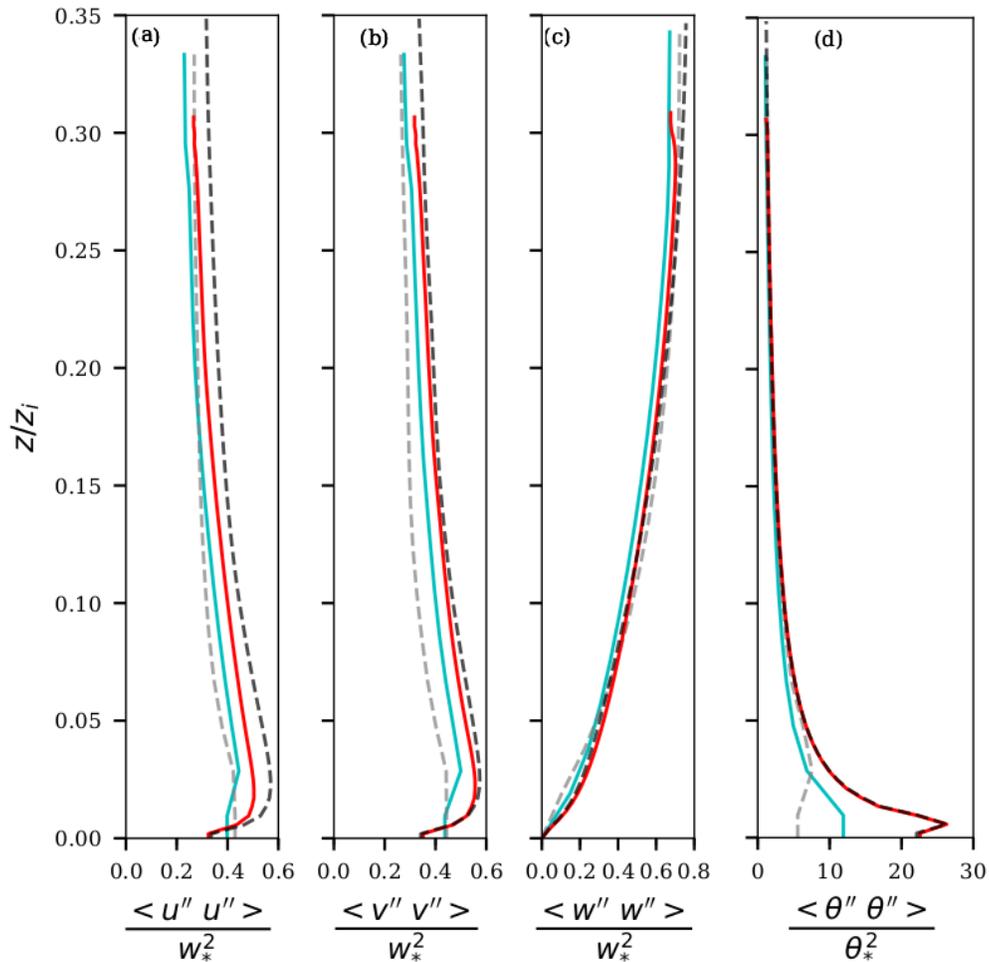
The Figures 5–9 have been updated. As suggested, different lines styles are now used, marginally thicker dashed lines for the standalone simulations and thinner lines for the two nested grids improve the visibility of the overlapping lines.

Page 11, line 9: I think it would be interesting to see the anterpolated values, just to see how the algorithm is working behind the scenes. The same comment as above

regarding plotting these additional data within the same plot applies here.

The variances of velocity and temperature close the surface are shown below. Compared to SA-C and SA-F, the CG profile in the anteroplated region is improved close to the surface.

We have added a profile of the sub-grid scale TKE to the manuscript, where the effect of maintaining the Germano identity during anterpolation is clearly noticeable. The anterpolated values are also shown in the vertical velocity skewness plot.



Page 12, lines 5-10 & Fig. 6. **Please explain more thoroughly the discontinuities in all profiles between CG and FG near the FG top.** Do the plotted profiles utilize the sponge layer that you describe, or not? Perhaps you could show the results with and without the sponge layer, using different linestyles and colors, as described above.

The plotted profiles use our simplified sponge layer approach, i.e. limiting the anterpolation to one CG cell less than the nested height. In our implementation, it is currently not possible to disable the simplified sponge layer as the data exchange from FG to CG for anterpolation are by default limited to one CG cell less than the nested height. We have expanded the explanation as:

“The v and w FG profiles have a better agreement with the SA-F than the u variance. The u and v variance in Fig. 7 (d and e) lie between SA-C and SA-F indicating that the resolved variances are improved compared to the SA-C but not sufficiently resolved to match SA-F. At the nesting height the variances deviate more from the SA-F and approach the CG values. Due to conservation of total kinetic energy across the nest boundary more CG TKE is contained in the sub-grid scale. Consequently, the resolved CG variances could have an undershoot as compared to SA-F, resulting in an undershoot of the FG variances too at the nesting height. Above the nesting height, the variance of u, v and w in CG are similar to SA-C.”

While you show mean profiles of various quantities, it would be nice to also see if there is any impact of nesting on the structures resolved within the CG above the FG in the nested simulations, relative to the SA-C (or within the nested domain relative to SA-F, although this is not as relevant). **Perhaps comparing spectra of streamwise velocity and/or w at a few heights would provide some useful information on this issue.** If the nested FG in the surface layer is able to improve the instantaneous structures resolved within the CG above, that would be another noteworthy advantage of the vertical nesting capability.

We have added new figure to compare the spectra of SGS TKE and vertical velocity at two heights, one within the nested grid and one above the nested grid. The w spectra of CG above the nested height follows SA-C and improvement due to higher resolution at the surface layer is not noticeable. We added the following text to the manuscript:

“The horizontal spectra of SGS turbulent kinetic energy and vertical velocity are plotted in Fig. 9 at two levels, one within the nested grid and one above the nested grid height. The FG TKE spectra in Fig. 9 (c) perfectly overlaps the SA-F spectra. The CG spectra has higher energy than the SA-C, this corresponds to the higher CG TKE values observed in Fig. 8 (c). As the limit of the grid resolution is reached at high wavenumber, the drop in the CG spectra is marginally delayed compared to SA-C. This improvement at high wavenumber is due to feedback from the FG. Similarly, the vertical velocity spectra in Fig. 8 (d) shows marginal improvement at high wavenumber for the CG. While the FG agrees with SA-C at high-wave number and at the spectra peak, at low wavenumber FG follows the CG spectra. At the level above the nested grid, the CG spectra agrees with SA-C for both TKE and the vertical velocity.”

Response to Referee Comment 3

I miss a thorough discussion on how the subgrid fluxes are handled at the interface between the course grid domain and the fine grid domain. I could imagine that the subgrid fluxes at the boundary between CG and FG would need to be interpolated.

The PALM model employs a 1.5 order turbulence closure parameterization. Therefore, at each time step, all the sub-grid fluxes are derived from the turbulent kinetic energy and the resolved gradients. Therefore, it suffices that the prognostic variables are communicated correctly. In our implementation of the nesting method, we assume that most of the TKE is resolved well down to the inertial subrange, except for the few lowest grid-layers. This allows us to use zero-gradient Neumann boundary condition for TKE at top of the nested grid.

We have added the following sentences to the manuscript:

“In the 1.5 order turbulence closure parameterization, all the sub-grid fluxes are derived from the turbulent kinetic energy and the resolved gradients at each time step. Therefore, the sub-grid fluxes do not have to be interpolated from CG to FG at the top boundary. Furthermore, in our implementation of the nesting method, we assume that most of the TKE is resolved well down to the inertial subrange, except for the lowest few grid layers. This allows us to use zero-gradient Neumann boundary condition for TKE at the FG top boundary.”

Or continuity of the

subgrid fluxes at that interface could be ensured by the subgrid models on either side of the interface. However, I do not see how the subgrid flux between CG and FG are handled in a conserving way: what leaves the CG should enter the FG and the other way around.

The sub-grid fluxes do not need to be continuous, only the sum of resolved and sub-grid fluxes should be. In the FG, there is a larger proportion of resolved flux (due to the smaller grid spacing) and less sub-grid flux.

However, we agree with the reviewer that for LES that employ higher order closure models (where the fluxes are independent variables) the sub-grid fluxes should be communicated as well.

b. The validation of the results of the nested simulation (characteristics of turbulent fields) is rather superficial (fluxes and variances, no spectral analysis or higher order moments; also little consideration for subgrid contributions).

We have included spectral analysis and vertical profiles of vertical velocity skewness and SGS TKE to expand our analysis. More information is provided in the detailed comments below.

c. The analysis of the timing of the simulations (scaling, overhead, net gain etc.) is limited.

We demonstrate the linear scalability of the nested simulations on more than fourteen thousand CPUs. However, in our benchmark runs, we did not profile the time taken by each nesting routine, and therefore a detailed analysis of the overhead is not possible. To overcome this limitation in our analysis, we executed standalone simulations with the same number of grid points as in the FG domain. The difference between the nested and the standalone in the scalability plot provides a rough estimate of the overhead. The analysis is expanded with the strong scaling efficiency suggested in the detailed comments.

d. The application of boundary conditions to the nested grid is insufficiently clearly described:

- Is the Dirichlet condition for horizontal wind components and scalars applied to a point just above the fine grid domain, or to the highest point just inside the fine grid?

We define 'top of the FG' as the highest point in the FG. This is the boundary point which is excluded from the CFD calculations. We added the following sentence in section 2.3.1:

“We define the top of the FG as the boundary level just above the prognostic level of each quantity.”

- The equations given for the interpolation algorithm lack explanation.

The explanations to interpolation equations is added to the text. More information is provided in the detailed comments below.

e. The structure of the introduction could be improved. After the overview of the history of LES, I would expect a clear definition of the problem (we need high resolution where it matters: close to the surface (and in the entrainment zone), an overview of how people have solved this until now, what is that we still not know/can/have?, and how are you going to solve it. Also the structure of section 2 could be improved to more clearly separate the different aspects of the new model.

We have improved the structure of the introduction and re-structured section 2 as suggested. The changes are listed in the relevant detailed comments.

Detailed comments

1. 2-27: You immediately make the jump to grid-nesting. However, the main point is that you need increased resolution. And if you cannot afford to increase the resolution in the entire domain, you want to do it locally. One way of doing that is by grid-nesting. But there are other ways: if one does not insist to stick to a structured grid, local grid refinement (without nesting) is feasible. This grid refinement can even be made dependent on the flow itself (see van Hooft et al., 2018). So: grid-nesting is just one of the ways to locally increase resolution.

We have re-structured the paragraph to clearly define the problem and then introduce solutions other than nesting and then introduce a summary of the nesting literature. We updated the manuscript as:

“Still, especially in heterogeneous terrain, near topographic elements, buildings or close to the surface, the required higher resolution is not always attainable due to computational constraints. In spite of the radical increase in the available computing power over the last decade, large-eddy simulations of high Reynolds number atmospheric flows with very high-resolution in the surface-layer remain a challenge. Considering the size of the domain required to reproduce boundary-layer scale structures, it is computationally demanding to generate a single fixed grid that could resolve all relevant scales satisfactorily. Alternatively, local grid refinement is possible in the Finite-Volume codes that are not restricted to structured grids. Flores et al. (2013) developed a solver for the OpenFOAM modelling framework to simulate atmospheric flows over complex geometries using an unstructured mesh approach. Van Hooft et al. (2018) demonstrated the potential of adaptive mesh refinement technique where the tree-based Cartesian grid is refined or coarsened dynamically, based on the flow structures.”

2. 3-7 to 16: here you explain why vertical nesting is needed. But you started that argument

already in line 2-28 to 30. Please restructure your argumentation (either move 3-7 to 16 to the point where you introduce vertical nesting (and then talk about horizontal nesting to show what we know from that), or first introduce horizontal nesting and then make the step to vertical nesting (or ignore horizontal nesting altogether, since vertical nesting is in itself not new, just your implementation in PALM is new).

We restructured the text to first introduce horizontal nesting and then focus our discussion on the vertical nesting. We would like to retain the discussion on horizontal nesting as our vertical nesting is motivated by literature in horizontal nesting. The vertical nesting discussion is restructured as:

“For our purposes, we will focus on vertical nesting, i.e. we consider a Fine Grid nested domain (FG) near the lower boundary of the domain, and a Coarse Grid parent domain (CG) in the entire of the boundary layer. While the latter's resolution is sufficient to study processes in the outer region where the dominant eddies are large and inertial effects dominate, such coarse resolution is not sufficient where fine-scale turbulence in the surface layer region is concerned.

The higher resolution achieved by the vertical nesting will then allow a more accurate representation of the turbulence in the surface layer region, by resolving its dominant eddies. For studies that require very high resolution near the surface (e.g. virtual tower measurements, wakes behind obstacles, dispersion within street canyons for large cities) a nesting approach is an attractive solution due to the reduced memory requirement. The challenge of a vertically nested grid is that the FG upper boundary conditions need to be correctly prescribed by the CG. Though vertical nesting is less common than horizontal nesting, it has been implemented in some LES models. A non-parallelized vertical nesting was explored by Sullivan et al. (1996) but this code is not in public domain and we could not find any record of further development or application of this code in publications. An LES-within-LES vertical nesting is implemented by Zhou et al. (2018) in the Advanced Regional Prediction System (ARPS) model.”

3. 3-17: it seems that Clark and Hall (1991) deals with horizontal nesting. To what extent is it still relevant for this paper?

Though Clark and Hall (1991) deals with horizontal nesting, their error analysis of the nesting procedures is relevant in understanding the 'post-insertion' and 'pressure defect correction'. Their work has also provided motivation for other vertical nesting development (Sullivan et al. (1996). Updated the manuscript as:

“Clark and Hall (1991) studied two different approaches for updating the CG values, namely "post-insertion" and "pressure defect correction".

The two approaches were also investigated by Sullivan et al. (1996) in their vertical nesting implementation.

4. 3-23: ‘...superior when the waves ...’: doe you mean ‘when’ or ‘if’? And what happens if/when the waves are not well resolved? In what way is this relevant for the present paper on the simulation of turbulence?

Harris and Durran (2010) observed that only for moderately well resolved waves, the two-way interaction performed better than the one way interaction. Modified the text to introduce the concept of 'sponge boundary condition':

“Harris and Durran (2010) used a linear 1D shallow-water equation to study the influence of the nesting method on the solution and found the two-way interaction to be superior if the waves are well resolved. They introduce a filtered sponge boundary condition to reduce the amplitude of the reflected

wave at the nested grid boundary.”

5. 3-28: ‘... both the resolved and SGS fluxes...’: does this also hold for the finite difference code used here? In what way would/does it increase coding complexity?

We had written:

“Sullivan et al. (1996) report that in the case of their Pseudo-Spectral LES, both the resolved and SGS fluxes need to be anterpolated to the CG and such a procedure increases coding complexity.”

Since no explanation for the increase in the coding complexity is found in the literature and also because this does not hold for the finite difference code, we have removed this statement.

6. 4-17: please explain the variables used in the equations. In particular the notation for resolved variables and subgrid variables is important. Furthermore, I assume you include the tendency equation for potential temperature because the potential temperature plays a role in the SGS-TKE equation and in the momentum equation. But then you should also include the moisture tendency in order to be able to determine the tendency of the virtual potential temperature (which then also should be used in the buoyancy terms). Finally, the heat flux that appears in equation (4) is the subgrid heat flux: (1) apparently you denote subgrid variations by a single prime and the filtering operation by an overbar and (2) in the model the subgrid heat flux is parameterized using a gradient hypothesis (also the next term, the transport term, is parameterized).

We have updated the equation adopting the convention followed by Maronga et al. (2015). The moisture tendency equation is added and the virtual potential temperature is included in the buoyancy term. We now denote the sub-grid heat flux with double prime and have also added the parameterization by gradient hypothesis. All the symbols are listed in Table 1.

7. 4-20: ‘guarantees a stable’: how does the choice of the time integration method guarantee a stable solution. The magnitude of the time step would still play a role (and it does, as later on you invoke the CFL criterion). So why mention stability here?

The combination of Runge-Kutta-2 integration and the 5th order advection scheme is known to be conditionally unstable. The default time integration and advection scheme in PALM are RK3 and 5th order upwind discretization according to Wicker and Skamarock, respectively.

Modified the sentence as:

“The low storage RK3 scheme with three sub-steps proposed by Williamson (1980) guarantees a stable numerical solution in combination with both the advection schemes”.

8. 4-23: I assume that you refer the vertical zero pressure gradient here.

Yes, we refer to vertical zero pressure gradient. Updated the text as:

“A vertical zero pressure gradient at the surface guarantees the vertical velocity to be zero.”

9. 5-9: apart from updating the ghostpoint, there is also global communication needed in the Poisson solver. This involves way more communication than the ghostpoint update.

We agree that the global communications for the Poisson solver need more communication than the ghost point update. Updated the sentence as:

“The data exchange between PEs needed by the Poisson solver and to update the ghost points are performed via the Message Passing Interface (MPI) communication routines.”

10. 5-10: regarding the structure of the rest of section 2: I would suggest to restructure this section as follows:

2.2 Model structure

2.2.1 Grid configuration (now 2.2, up to line 5-29)

2.2.2 Nesting algorithm

2.3 Translation between grids (line 5-9 until 7-4)

2.3.1 Anterpolation

2.3.2 Interpolation

2.4 Parallel inter grid communication

(after 2.2.2 it is clear where and why anterpolation and interpolation are needed).

The section 2 is re-structured as suggested.

11. 5-30: only the vertical velocity really has a boundary at the top of the FG. For the other velocity components and scalars it is unclear whether the boundary condition (interpolation from CG) is applied to a ghost point (just above the FG) or to the first point just below the boundary of the FG.

The boundary condition is applied at the boundary level. This level is excluded from the CFD computations and only acts as a boundary constraint in the CFD equations for the neighbouring prognostic grid cells. We prefer to call it as boundary point instead of ghost point and keep the terminology of ghost points for the grid points which are constraints on one parallel processing element but CFD point on another processing element. The (vertical) boundary level is a boundary level on all processing elements. We added the following sentence in section 2.3.1:

“We define the top of the FG as the boundary level just above the prognostic level of each quantity.”

12. 5-32: what is the ‘logical’ direction? If figure 1 would be upgraded (see below), this ‘logical’ linear interpolation would probably become clear.

The logical direction is the dimension corresponding to the velocity component. We replace it by “in its own dimension” in the manuscript:

“For the velocity components, the interpolation is linear in its own dimension, and quadratic in the other two directions.”

13. Figure 1: the current figure is not very informative. I would suggest to replace it by a figure in which you show a few CG cells as well as the FG cells within one or two of them (preferably with a grid ratio of 3, not more). Then clearly show how the interpolation of vertical velocities, as well as horizontal velocities and scalars works (in order to support the interpretation of equations (5) as well as the notion that the velocities are interpolated in a ‘logical’ direction. The connection to equations (5) could also clarify the meaning of the various indices (lowercase and uppercase).

As suggested we have included a schematic of a nested grid with a nesting ratio of 3. However, we would also like to retain the original figure as it could be informative to readers not familiar with the nesting procedure. The explanation to the equations have also been expanded.

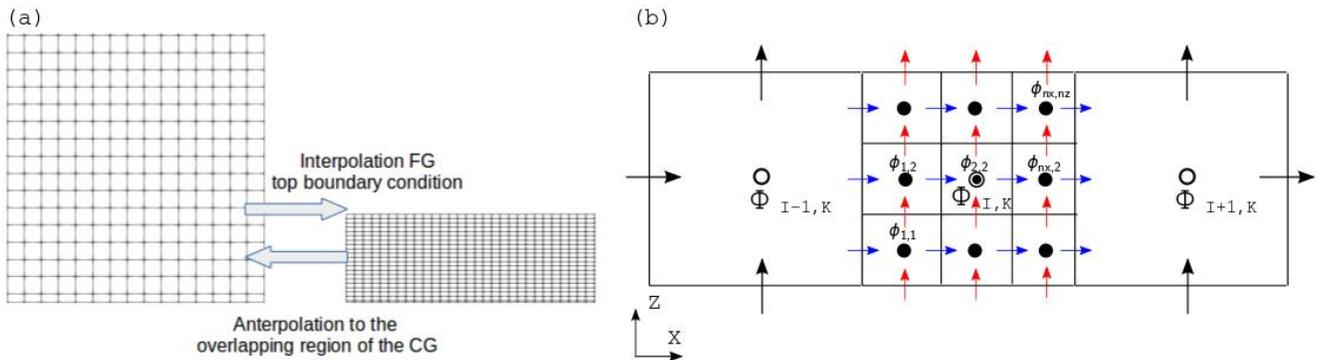


Figure 1. (a) Schematic of the interpolation and antinterpolation between grids. The FG top boundary condition is interpolated from the CG. The CG prognostic quantities in the overlapping region are antinterpolated from the FG. (b) Schematic of Arakawa C grid for two grids with nesting ratio of three. The black arrows and circles are CG velocity and pressure, respectively. The blue and red arrows are horizontal and vertical velocity, respectively, in the FG. The filled black circle is the FG pressure. The symbols Φ and ϕ represent CG and FG scalar quantities. Where I and K are CG indices and nx and nz are the nesting ratio in x and z , respectively.

14. Please completely rework the equations and add explanations:

- Make clear that the first equation is the actual interpolation, and all the other equations just define the various parameters occurring therein.

We have re-numbered the equations to make the distinction. We kept the indices for the antinterpolation equations, but we changed the indices in the interpolation, to make it less confusing.

- In which coordinate direction does i vary: only in the x -direction, or also in other directions. Or are we actually looking at a 2D or 3D stencil of which only one dimension is shown?

i varies only in x . Yes, we are only looking at one dimension of the stencil (added in the text). For the interpolation, we renamed this index to “ m ” because it has a different flavour than the index “ i ” in the antinterpolation.

- The capital indices I , J , and K are counting through the entire domain, I assume. But how about the lower case indices: do they start counting at 1 (or zero) within each CG cell, or do they also count globally?

We understand that the reviewer is referring to the antinterpolation equation. The lower case indices only count over the fine grid cells that belong to that particular coarse cell. So for each (I,J,K) tuple it is a restricted set of (i,j,k) . However, due to the grid conventions in PALM the (i,j,k) have global numbers. Due to the setup of the grid indices in PALM, this is also true for the parallelization with MPI. However, whether the i,j,k are defined locally or globally is a matter of how the nesting is applied practically, and it does not influence the nesting philosophy.

In the text: “The lower case indices only count over the fine grid cells that belong to that particular coarse

cell. For each (I,J,K) tuple there exists a restricted set of (i,j,k) indices in the FG. In order that the nested PALM knows at all times which fine grid cells and coarse grid cells correspond, we compute a mapping for the FG and CG indices before starting the simulation, and we store this mapping in the memory.”

For the interpolation equation, I is global again but (to make the equation better readable) m is local, running from 1 to nx.

- In the 2nd-4th equations you introduce H_k . What is the value of the index k. Or does the repeated index imply summation? If so, what is the range of values that k can take: 1, 2 and 3 because of the dimensionality, or 1, 2, ... nx because of the number of FG cells in a CG cell?

We admit that the k is confusing here and we have replaced it by m as well (in the original document i and k should have been equal). There is no summation convention here (added in the text). We added some lines explaining the philosophy behind the interpolation equation.

15. 6 - equation (6): what is the range of values for i, j and k ? Is there a mapping that gives the global i,j,k values for a given I,J,K or, are these local i,j,k values, running as 1, 2, ... nx ?

Yes, there is a mapping. This mapping is essential for the nesting algorithm to match the corresponding cells in the fine and coarse grid, and it is computed in advance and stored in the memory. These i,j,k values are not global.

Added in the text (see also above): “In order that the nested PALM knows at all times which fine grid cells and coarse grid cells correspond, we compute a mapping for the FG and CG indices before starting the simulation, and we store this mapping in the memory.”

16. 7 - equation (7): idem

Equation for velocity interpolation has also been updated similar to the equation for scalars.

17. 8-7 and 11: please keep the discussion on the solution of the Poisson equation in one place.

What is the value of the pressure gradient that is imposed as a Neumann boundary condition? If it is zero, reflections could occur, but if you use something non-zero: how do you determine the value of this gradient? Is it derived from the CG pressure field?

We have re-arranged the sentences to keep the discussion of Poisson equation in one place.

We use a zero gradient Neumann condition for pressure. We would like to quote the opinion of R1 here: “In this model the nested upper boundary condition is relatively simple, which is OK for the Boussinesq LES problem presented here in which there are no sound waves and any vertically-propagating gravity waves would be very well-resolved.”. In what we refer to as the ‘simplified sponge-layer’, we have also split the level of the FG upper BC from the highest level of interpolation of FG to CG in order to reduce oscillations originating from the FG boundary.

18. 8-12 and 13: please clarify how the value of the imposed pressure gradient is determined/chosen. We impose zero gradient at the top and bottom of the nested domain.

19. 8-31: ‘... the higher number of PE available in the FG.’: this is stated as if the reader already knows that there are more PEs in the FG (although for any grid ratio above 2 it is indeed logical that the number of FG PEs is larger than the number of CG PEs). But in addition, it is unclear to me why the higher number of FG PEs would be relevant for the FG-to-CG communication.

To perform the antinterpolation operation, either the FG data can be sent to the CG and then be antinterpolated, or alternatively, and more efficiently, the antinterpolation operation is performed in the FG and then the antinterpolated values are sent to the CG. The latter approach benefits from the higher number of FG PE and smaller array dimensions of the antinterpolated values.

We have modified the sentence as:

“The exchange of arrays via MPI_SENDRECV routines is computationally expensive. Therefore, the size of the arrays communicated are minimized by performing the antinterpolation operation in the FG PE’s and storing the values in a temporary 3D array that is later sent via the global communicator to the appropriate CG PE. This approach is more efficient than performing the antinterpolation operation on the CG which has less PE's and needs communication of larger arrays from the FG.”

20. 9-2 ‘should be kept lower’: please explain the logic of this statement. I assume that the idea is that you want to reduce the total amount of idle CPU time on the FG PEs (N cores \times wait time), which can be achieved by under-utilization of the (only) M cores running CG (better waste time on a few CG cores than on many FG cores). In order to know how this plays out in practice, you should show in your results the amount of time spent in the various steps in a RK substep: which fraction (and absolute time) of a time step is devoted to which substep in figure 2, and how much of this time is wasted time.

The work load of CG PEs are kept lower than the FG PE to reduce the total amount of idle CPU time on the FG PEs. Unfortunately, in the nested simulation the time spent in various steps is not profiled. We will not be extending our current parallel implementation as the main developers of PALM will be developing the 'PALM Model Coupler', a unified tool to handle the communication between the grids for nested simulations, ocean-atmosphere coupling etc.

“Within the RK3 sub-steps, when one grid executes the pressure solver the other grid has to wait leading to more computational time at every sub-step. However, the waiting time can be minimized by effective load balancing, i.e. the number of grid points per PE in the CG should be kept lower than in the FG. The reduction in workload per CG PE is achieved with a few additional cores. The reduction in computational time per step in the CG means the idle wait time on the FG PE is also reduced.”

21. 9-12 ‘Dirichlet condition’: to which values are the velocities set: zero for vertical wind and geostrophic for horizontal?

We have updated the text as

“The Dirichlet boundary condition is applied for velocity at the top and bottom boundaries, the vertical velocity component is set to zero and the horizontal components are set to geostrophic wind.”

22. 9-13: what is the imposed temperature gradient at the surface?

Since we prescribe a constant surface heat flux, we use zero gradient Neumann condition for the potential temperature. Updated the text as:

“The potential temperature is set to Neumann condition at the bottom and the gradient is determined by MOST based on the prescribed surface heat flux and roughness length. The gradient of the initial profile is maintained at the top boundary.”

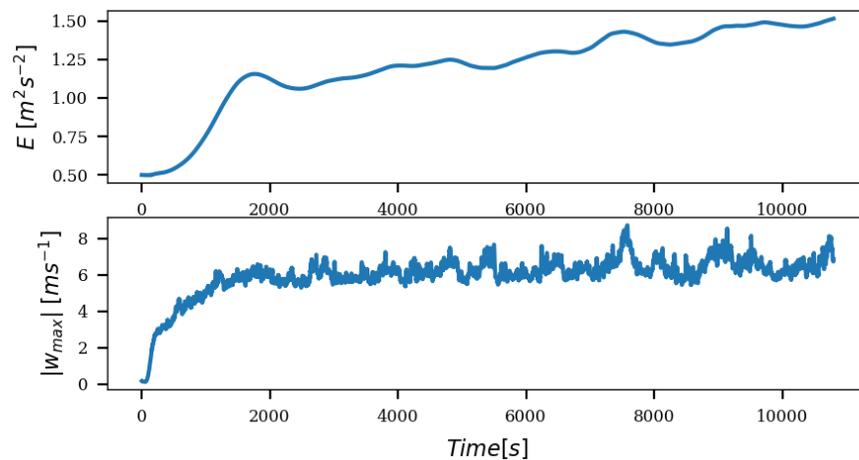
23. 9-15 and 16: is the wind profile interpolated linearly from zero at the surface to geostrophic at the top? Does this out-of-balance initialization lead to an inertial oscillation?

Earlier we had written “The u and v profiles are constructed starting from a zero value at the surface and reaches the geostrophic wind value at the top.”

We correct the statement as:

“The u and v initial profiles are set to be constant value of the geostrophic wind component in the domain and the vertical velocity is initialized to zero in the domain. “

The initial profiles are set to be constant ($u_g = 1$, $v_g = 0$). Whilst it is true that the initialization is out-of-balance, the amplitude is small with respect to the surface heating in our convective boundary layer. Observing the time series of total kinetic energy of the flow (3D domain average) shown in the figure below, we can see that the oscillations subside after 1 hour of spin up. Similarly, plotting the time series of the absolute maximum vertical velocity, we observe that the maxima is almost constant after 1 hour spin up phase.



24. Table 1: what is the boundary condition for wind? MOST with an imposed roughness length (what is the value) or an imposed stress?

It is MOST with an imposed roughness length. Updated Table 2 to list roughness length value as 0.1 m.

25. Table 2:

- please add the number of time steps needed to complete this simulation (in that way the reader can easily determine the time spent per gridpoint per time step.

We have added a column for time steps.

- For the reader it would also helpful to include the number of grid points per PE and the CPU time per grid point (but both numbers can be derived from the available data, so the reader could do it for themselves).

We have added a column for the number of grid per PE, this column indeed readily informs the reader that FG and SA-C have same workload per PE. However, we have not included the CPU time per grid point.

- the number of cores devoted to CG and FG respectively is not motivated. Whereas later on

you advise to assign to a CG PE 40-80% of the number of grid points that is assigned to a FG PE, here you use a fraction of 16%.

We used a machine with 20 cores per node and allotted all the cores in one node to the CG. Even though it is possible for the CG and FG to share a node, the domain decomposition restrictions often prevent an ideal grid configuration. The limitations in domain decomposition are now included in the practical considerations section 3.4:

“For better computational performance it is recommended that the number of grid points per PE in the CG is kept at only 40 to 80 percent of the FG value. The reduced work load of the CG is expected to minimize the waiting time of the FG during the concurrent time advancement by quicker CG pressure solver step. However, the actual improvement in performance will depend on the memory available, processor speed and the inter-node communication architecture of the computing cluster and the optimal load balancing can only be found through trials. Furthermore, the choice of the domain size is often restricted by the topology of the processor decomposition. In a 2D decomposition, the number of grid points along the x-direction should be an integer multiple of the number of PE along x and similarly for y-direction. This condition has to be individually satisfied for the CG and the FG.”

- it is unclear to what extent the PE's are saturated in terms of memory usage: could this problem be run on even a smaller number of processors to improve performance?

We have two simulation set-ups: one simulation is performed on a small cluster to demonstrate the quality of the results and the other simulation is performed on a supercomputer to demonstrate the computational performance. However, the first simulation is still relevant in understanding the effect of grid resolution on time step and the associated increase in the core-hours.

- Please include information on the time (absolute and/or as a fraction) that is used waiting for input from CG to FG or the other way around. This would be helpful to determine the optimal division of labor between CG PEs and FG PEs (in terms of grid points per node).

In our benchmark runs, we did not profile the time taken by each nesting routine, and therefore a detailed analysis of the overhead is not possible.

26. 10-2: what initial perturbation is applied to get turbulence started? How did you verify that after 9000 s the flow was in equilibrium?

Random perturbations are imposed to the horizontal velocity field. If the perturbation energy has exceeded this energy limit of $0.01 \text{ m}^2/\text{s}^2$, no more random perturbations are assigned.

Observing the time series of total kinetic energy of the flow (3D domain average) shown above in response to question - 23, we can see that the flow has reached a quasi-stationary state after 1 hour of spin up. Similarly, plotting the time series of the absolute maximum vertical velocity, we observe that the maxima is almost constant after 1 hour spin up phase.

27. 11-5: part of the ingredients for the scaling variables are in fact imposed boundary conditions (the surface heat flux), whereas indeed another part (the surface shear stress) results from the flow (and hence need to be derived from one of the simulation results (assuming that a roughness length is prescribed).

We have replaced u^* with w^* for the normalization of velocity variance as suggested in detailed comments - 31. While we acknowledge that the u^* and w^* result from the flow, it is more common to

normalize the velocity variance with these scaling variables instead of imposed flow variables like the geostrophic wind, because the latter is not a scaling variable in the ABL.

28. 11-6 and 8: surface heat flux in the expression for w^* : overbar is missing and this is not a turbulent flux (so do not use a covariance flux).

We now represent the surface heat flux with the symbol H_s .

29. 11-8: although it will not change the lines in the graph, normalizing the temperature with the surface value is very illogical. Please plot the temperature with some reference value (e.g. the surface value) subtracted and normalized with θ^* .

We have updated the figure to plot the temperature with surface value subtracted and normalized with θ^* .

30. 12-1: what would/could be the mechanism that makes that the higher resolution in the surface layer would affect the variance profile well above the FG domain?

We realized that in the initial runs we had failed to include the compiler flag (-fpmode strict) necessary to make every realization deterministic. As a result the plots in our initial submission have compiler optimization related random effects that were noticeable in the variance profile. The results of both the nested and the standalone simulation were affected. We have now ensured that the simulations are deterministic, which improves their comparability. The plots and their description have been updated accordingly.

In the updated plots, the CG profile is not noticeably affected by the higher resolution in the surface layer. Therefore, we have removed the statement.

31. 12-3: please use the same scaling variable for all velocity components! If not, the different variances (which together constitute the turbulent kinetic energy) cannot be compared.

Furthermore, the given flow is close to free convection, so using the friction velocity as a scaling variable does not make sense.

The velocity variances are now normalized only by the convective velocity scale. The figures have been updated

32. 12-5: you refer to an overshoot in the v variance. The u variance shows an overshoot as well.

I assume that the profiles shown are based on the resolved variances only. In that case, we should keep in mind that in the CG domain a larger proportion of the TKE is contained in the subgrid scales. Could this explain the jump? **Please include an analysis of the difference in SGS-TKE between the two domains at the top of the FG (of course there is the difficulty of separating the SGS TKE into the three components, but at least quasi-quantitatively such an analysis could shed light on these jumps/overshoots.**

The plots in our initial submission had random errors introduced due to wrong choice of compiler optimization flags that was visible in the v-variance. However, there are still minor artifacts

We agree that the jump could be explained by the large SGS TKE component in the CG. We updated the text as:

“At the nesting height the variances deviate more from the SA-F and approach the CG values. Due to conservation of total kinetic energy across the nest boundary more CG TKE is contained in the sub-grid scale. Consequently, the resolved CG variances could have an undershoot as compared to SA-F,

resulting in an undershoot of the FG variances too at the nesting height. Above the nesting height, the variance of u , v and w in CG are similar to SA-C.”

33. 12-7: how would the anterpolation influence the vertical velocity variance in the FG domain. Please explain the/a mechanism. Or is it a result of the fact that the upper boundary conditions for pressure at the top of the FG is not well-defined?

Zhou et al. (2018) in the vertically nested LES note that the kink in the higher-order profiles can be minimized by increasing the depth of the sponge layer. In our two-way nesting we have used a simplified sponge layer by limiting the anterpolation to one CG cell less than the nested height. This split in the level of the FG upper BC from the highest level of anterpolation of FG to CG reduces oscillations originating from the FG boundary.

In the description of the skewness plots we write:

“However, at the nesting height a small kink in the skewness is noticeable. Zhou et al. (2018) observe that the magnitude of the kink in the higher-order profiles can be minimized by increasing the depth of the sponge layer. Our simplified sponge layer approach appears to be unable to effectively minimize the kinks at the nesting height.”

34. 13 figure 6: the variance profiles give some information on the quality/realism of the simulated turbulence. One analysis that is missing (related to the point made above regarding the overshoot) is whether the increased resolved TKE is the amount that would be expected based on the increased resolution (and hence reduced reliance on the subgrid model). To properly analyse that one would need turbulent spectra to see how much kinetic energy is contained in the additionally resolved scales. Additionally, spectral analysis (preferably with 2D spectra) would help to show to what extent the extra resolved turbulence has the expected turbulent characteristics (increased variance is nice, but does not need to be additional turbulence, it could also be increased noise).

We have included a spectral analysis. The plots are described as:

“The horizontal spectra of SGS turbulent kinetic energy and vertical velocity are plotted in Fig. 9 at two levels, one within the nested grid and one above the nested grid height. The FG TKE spectra in Fig. 9 (c) perfectly overlaps the SA-F spectra. The CG spectra has higher energy than the SA-C, this corresponds to the higher CG TKE values observed in Fig. 8 (c). As the limit of the grid resolution is reached at high wavenumber, the drop in the CG spectra is marginally shifted compared to SA-C. This improvement at high-wavenumber is due to feedback from the FG. Similarly, the vertical velocity spectra in Fig. 8 (d) shows marginal improvement at high wavenumber for the CG with respect to SA-C. While the FG agrees with SA-C at high-wavenumber and at the spectra peak, at low wavenumber FG follows the CG spectra. At the level above the nested grid, the CG spectra agrees with SA-C for both TKE and the vertical velocity.”

35. 13-1: The heat flux profile is not the prime quantity at all! For a quasi-stationary convective boundary layer with imposed surface flux the heat flux profile is the most boring part of the simulation. Provided that the entrainment flux is represented well, the flux profile is by definition linear, varying between the imposed surface flux (so no surprises there) and the entrainment flux (which, admittedly, needs to be represented correctly by the simulation: still some freedom there). This linear flux profile is completely independent of the quality and resolution of the simulation. The only freedom there is is which part of that flux is carried by the

resolved scales and which part is carried by the subgrid model. Hence the perfect correspondence between all simulations (full FG, full CG, nested CG and nested FG). Hence, please do not use the heat flux profile as a measure of the quality of the simulation.

We agree that heat flux profile is not the best measure for the quality of a simulation. However, we would like to retain the plots because we are interested in the heat flux for other applications in our working group to study the energy balance closure in the surface layer.

36. 14-4: ‘...we increase the resolution further’: do you mean to increase the grid ratio, the size of the FG region, or the overall resolution of the CG domain?

We have replaced the statement with quantitative analysis using the updated time step information in Table 3. The manuscript is updated as:

“The computational resources used in the simulations discussed above are listed in Table 3. The resources needed by SA-C is only 8 core hours. While the nested simulations needed about 1879 core hours, the SA-F needed about 4 times more core hours. As the resolution is increased from 20 m in SA-C to 4 m in SA-F the number of time steps increased more than 5 times as higher resolution demands smaller time step size.”

37. 14-9: ‘in terms of communication time’: do you only look at communication time because that is the most restricting, or because you are only interested in that (in this context)? And why should the number of domains be equal in x and y direction: please explain the logic of this (and does it also hold if the length of the domain is different in x and y direction?)

We are interested in the communication time in the context of the domain decomposition because, the choice of processor decomposition has an effect on the communication performance. We update the text as:

“The best performance in terms of communication time in a standalone run is achieved when the number of sub-domains in the x and y directions are equal. In that case the number of ghost points at the lateral boundaries are optimally minimized.”

38. 15-1 to 5: why is the setup of these simulations (in terms of the total number of points and ratio of number of grid points between CG cells and FG cells) so different from the original runs? Are the performance results still relevant to understand those first runs? If so, why?

Please give the setup of these runs in a table similar to table 2 (not ‘number of grid points is around...’). We have two simulation set-ups: one simulation is performed on a small cluster to demonstrate the quality of the results, and another simulation is performed on a supercomputer to demonstrate the computational performance. The first simulation is relevant in understanding the effect of the grid resolution on the time step and the associated increase in core-hours. The second set of simulations demonstrates the scalability on large number of cores. Since both machines have different processor architecture, memory and processor per node, the results are not directly comparable.

In the performance benchmark set-up CG and FG PE’s are chosen to be a multiple of 16 to confirm with the 16 cores per node. To avoid load balancing bias in the scalability analysis, the ratio between the number of PEs for CG and FG is kept constant in all the five runs listed in Table 5. Keeping the processor ratio constant implies that the ratio between the number of grid points per PE in CG and FG is also held constant. Consequently, in this performance test, the FG has 1.25 times more grid points per PE than the CG in all the processor configurations tested.

We have added two tables: “Table 4. Number of grid points in nested and non-nested FG domain.” And “Table 5. Grid configuration of the nested and non-nested FG domain.”

39. 15- Figure 8:

- on a log-log scale everything looks nice. Please give a more informative representation.

E.g. use the strong scaling efficiency, which will vary between 1 and somewhere below 1 (for your data, using the left-most simulation as a reference, the efficiency goes down to about 90% for the right-most. But the question is, what would have been the CPU time for the smallest possible number of processors on which this case could have been run (memory-wise).

We computed the strong scaling efficiency as suggested and included the information in Table 5. We have only tested on total PE above 1664. However, for the smallest processor configuration, the FG group has 3.9 million grid points per PE when the usable memory on this machine is only 1.625 GB per PE.

- In addition, find a more informative way to quantify the waiting time overhead.

In our benchmark runs, we did not profile the time taken by each nesting routine, and therefore a detailed analysis of the overhead is not possible. To overcome this limitation in our analysis, we executed standalone simulations with the same number of grid points as in the FG domain. The difference between the nested and the standalone in the scalability plot provides a rough estimate of the overhead.

40. 16-9: ‘large scale forcing compatible’: do you refer to the large scale forcing in terms of pressure gradient/geostrophic wind? Or another large scale forcing? Why would it, or would it not, be compatible. Please clarify.

We refer to the large scale forcing feature in PALM, which is PALM driven by a.o. geostrophic wind from a gridpoint of a synoptic model (see Maronga 2015 for more details). We did not expect it to be incompatible but in model development it is not guaranteed that two different modules are directly compatible, so we made some separate tests of the large scale forcing feature with the nesting. We deem this is useful information for PALM users who would like to combine the nesting module with other modules in PALM.

41. 16-12: ‘accuracy’: for accuracy in what sense (interpolation errors, truncation errors, turbulence statistics, stability,) should the grid ratio be not too large?

We expect that too large nesting ratio would affect the accuracy of the turbulence statistics. Since we do not have extensive analysis of larger grid ratios we have removed the sentence.

42. 16-14: ‘first five grid points are unreliable’: for which variables does this hold, in which aspect are the grid points unreliable (I assume you mean ‘the results at the first five grid points vertically displaced from the surface’): turbulence characteristics, mean profiles, noise,? Do you have a reference for this bold statement?

We meant the first five grid layers in the vertical direction from the surface are unreliable because the resolved turbulence is not well developed yet. This is only based on our experience; due to lack of literature to support this point we have removed this sentence.

43. 16-16: I would like to see a quantitative motivation for this 40-80%.

Unfortunately we do not have a quantitative analysis to support this statement. However, this is based on our trials on two different architectures, one a small computing cluster and the other a supercomputer. Since multiple factors like the memory available, processor speed and the communication architecture of the high performance machine a generally applicable quantitative analysis is difficult. However, we have expanded the practical considerations in section 3.4 to mention these factors:

“For better computational performance it is recommended that the number of grid points per PE in the CG is kept at only 40 to 80 percent of the FG value. The reduced work load of the CG is expected to minimize the waiting time of the FG during the concurrent time advancement by quicker CG pressure solver step. However, the actual improvement in performance will depend on the memory available, processor speed and the inter-node communication architecture of the computing cluster and the optimal load balancing can only be found through trials.”

Very detailed comments

1. 2-14: ‘possible, by the time’ → ‘possible. By the time’

Corrected as: “As computing power progressed, higher resolution and larger domains became possible. By the time of Schmidt and Schumann (1989)”

2. 2-18: ‘supercomputers’: also the people before Kröniger et al. used supercomputers. So remove ‘with the help of supercomputers’.

Corrected as: “More recently, Kröniger et al. (2018) used”

3. 2-19: remove ‘speeds’

Corrected as: “to study the influence of the wind on”

4. 2-21: ‘higher detail’: ‘higher’ than what/when/who?

Corrected as: “The atmospheric boundary-layer community has greatly benefited from the higher spatial resolution available in these LES to study turbulent processes that cannot be obtained in field measurements”

5. 2-28: ‘Nesting has been applied...’: because the previous sentence talks about vertical nesting, the reader may think that this sentence gives examples of that. But then at the end it turns out to talk about horizontal nesting. Please rephrase.

The sentences have been re-arranged following detailed comments 2.

6. 3-3/4: ‘techniques are but often uses ...’: ‘uses’ should be ‘use’ (plural)

Corrected as: “but often use different terminology.”

7. 3-18: ‘... CG, there ...’ → ‘... CG, and there ...’

Corrected as: “FG receives information from the CG, and there is no feedback to the CG”

8. 3-24: make explicit that the ‘two different approaches’ only refer to the ‘anterpolation’ mentioned in the sentence before. Furthermore, nothing is said –explicitly- about the ‘pressure deficit correction’ (“there are two types of cars: blue cars”)

Only the post-insertion approach involves anterpolation. We have incorrectly spelled ‘pressure defect correction’ as deficit. Updated the text to clarify on ‘pressure defect correction’:

"In the post-insertion technique, once the Poisson equation for pressure is solved in the FG, the resolved fields are then antepolated to the CG. In the pressure defect correction approach, the pressure in the CG and FG are matched by adding a correction term to the CG momentum equations and an antepolation operation is not required."

9. 4-9: 'additional equation': additional to what? The SGS-TKE equation is in the Deardorff method, so it is not additional to his work.

Corrected as: "The sub-grid scale (SGS) turbulence is modelled based on the method proposed by Deardorff (1980)."

10. 4-11: 'The prognostic ...': move this sentence to below the equations (only after having presented the equations you need to talk about their discretisation).

Corrected as: "The prognostic equations for the resolved quantities are:"

11. 5-17: 'the grids' → 'grids' (this occurs in multiple places, please check).

Correct 'the grids' to 'grids' in all the occurrences.

12. 5-18: please explain here already that uppercase symbols refer to CG and lowercase symbols to FG.

Moved the sentence in front of the first use of the symbol: "Below we use upper case symbols for fields and variables in the CG, and lower case for the FG. "

13. 6-1: 'similar interpolation': in which way is it similar, and which way is it different?

Corrected as: "The same interpolation formulation is also used to initialize all vertical levels of the fine grid domain at the beginning of the nested simulation."

14. 6-5: 'scalars' → 'CG scalars'

The statement is general to the Arakawa C grid and not specific to the CG. We modified the sentence as: "In the Arakawa C-grid discretization that PALM uses, the scalars are defined as the spatial average over the whole grid cell"

15. 6-5: 'The scalars corresponding FG scalars (eq. 6)'. How much more are you saying than 'An average is an average'. If you want to state more, please make that clear and explicit.

Modified the sentence as clarified in the comment above:

"In the Arakawa C-grid discretization that PALM 5 uses, the scalars are defined as the spatial average over the whole grid cell, and therefore it is required that the CG scalar is the average of the corresponding FG scalars in (Eq. 15)."

16. 7-6: 'We implement...'. Well, that does not really come as a surprise: you gave that away already (see my suggestions for an alternative structure for section 2).

Restructured as suggested in detailed comment 10.

17. 8-7: '... is also updated...'. What else is updated? You mean the pressure? And is the vertical velocity updated throughout the FG, or are you only referring to the vertical velocity at the CG-FG interface?

The pressure solver along with the pressure also updates the vertical velocity. The pressure solver is updates entire FG domain. Updated the manuscript as:

“The Poisson equation is then solved for pressure in the FG and the vertical velocity in the FG is also updated by the pressure solver at this stage.”

18. 8-20: ‘...process. Whereas....’ → ‘...process, whereas....’

Corrected as: “The data between the processors of the same group are exchanged via the local communicator created during the splitting process, whereas the data between the two groups are exchanged via the global communicator exchanged via the global communicator”

19. 8-20: ‘exchange’ → ‘exchanged’

Corrected as: “The data between the processors of the same group are exchanged via the local communicator”

20. 8-23: ‘local PE’s 2D processor co-ordinate’: in what way is the PE different from the processor? → ‘local 2D processor (or PE) coordinate’

The terms processor and processing element are often used interchangeable. However, a subtle difference exists. While a processor is considered as a hardware unit, a processing element is a MPI task (a Unix process) that executes the program on a unique sub-set of data.

21. 9-11: ‘is set to’ → ‘has’

Corrected as: “The simulation domain has periodic boundary conditions”

22. 10, Tables 1 and 2: please format the tables properly as tables should be formatted (including column headings and a consistent demarcation of rows and columns)

Table 1: please note that the surface heat flux is not a turbulent flux (there is no vertical velocity (variation) at the surface. Furthermore, even if you would like to denote it as a turbulent flux, please add an overbar.

Format of all the tables have been updated. We adopts the symbol H_s for surface heat flux.

23. 11, figure 4: the lower panel is –vertically- not exactly to scale with the area indicated with the dashed line in the upper panel.

We have resized the subplots to scale.

24. 11-2: ‘flux profiles’ → ‘fluxes’

Corrected as: “The turbulent fluxes are obtained using the spatial covariance”

25. 11-3: the given expressions are not fluxes, but products of resolved deviations: please include an averaging operator to make it a flux.

We have included an averaging operator.

26. 11-13: ‘at the boundary layer height’ → ‘at the top of the boundary layer’

The sentence has been removed as the plot has been updated.

27. 12-1: ‘An one-way’ → ‘A one-way’

The sentence has been removed as the plot has been updated.

28. 12-5: ‘variance seen’ → ‘variance can be seen’.

The sentence has been removed as the plot has been updated.

29. 14-5: ‘Simulations with $O(1)$...’: are you referring to that resolution for a full domain, or

only for the FG part of a nested simulation? In fact, it is unclear where you are heading with lines 14-2 to 14-6.

We have replaced the statement with quantitative analysis using the updated time step information in Table 3. As answered in detailed comments 36.

30. 14-13: ‘new nested simulation’: new relative to? I assume that you mean new relative to the runs described in tables 1 and 2. These new simulations were made for the performance test only?

The new simulations were made only for testing performance. Updated the text as:

“Keeping these factors 5 in mind, we designed the nested simulation domains listed in Table 4 for the purpose of assessing the computational performance as the total number of processors is varied.”

31. 16-4: ‘Poisson equation’ → ‘the Poisson equation’

Corrected as: “Similarly, for solving the Poisson equation for the pressure”

32. 16-7: ‘FFT’ → ‘an FFT’

Corrected as: “therefore an FFT based pressure solver is an appropriate choice”

33. 16-28: ‘energy conserving methods’: I have not seen that term earlier in the paper. Where was this discussed before? Or are you referring to the anterpolation of SGS TKE? In that case, please be a bit more explicit.

We are referring to the anterpolation of SGS TKE. The text is updated as:

“The grids are coupled, i.e the interpolation of the boundary conditions and the feedback to the parent grid are performed, at every sub-step of the Runge-Kutta time integration. The anterpolation of the TKE involves the Germano identity to ensure the conservation of total kinetic energy.”

34. ‘... optimized for performance’: how were they optimized, where can I read about that optimziation?

We have updated the sentence as:

“The exchange of data between the two grids is achieved by MPI communication routines and the communication is optimized by derived datatypes.”

Vertically nested LES for high-resolution simulation of the surface layer in PALM (version 5.0)

Sadiq Huq¹, Frederik De Roo^{1,4}, Siegfried Raasch³, and Matthias Mauder^{1,2}

¹Institute of Meteorology and Climate Research, Atmospheric Environmental Research (IMK-IFU), Karlsruhe Institute of Technology (KIT), Kreuzteckbahnstrasse 19, 82467 Garmisch-Partenkirchen, Germany

²Institute of Geography and Geoecology (IfGG), Karlsruhe Institute of Technology, Kaiserstrasse 12, 76131 Karlsruhe, Germany

³Institute of Meteorology and Climatology, Leibniz Universität Hannover, Hannover, Germany

⁴Norwegian Meteorological Institute, Oslo, Norway

Correspondence to: matthias.mauder@kit.edu

Abstract. Large-eddy simulation (LES) has become a well-established tool in the atmospheric boundary-layer research community to study turbulence. It allows three-dimensional realizations of the turbulent fields, which large-scale models and most experimental studies cannot yield. To resolve the largest eddies in the mixed layer, a moderate grid resolution in the range of 10 to 100 m is often sufficient, and these simulations can be run on a computing cluster with few hundred processors, or even on a workstation for simple configurations. The desired resolution is usually limited by the computational resources. However, to compare with tower measurements of turbulence and exchange fluxes in the surface layer a much higher resolution is required. In spite of the growth in computational power, a high-resolution simulation LES of the surface layer is often not feasible: to fully resolve the energy containing eddies near the surface a grid spacing of $O(1\text{ m})$ is required. One way to tackle this problem is to employ a vertical grid nesting technique, where the surface is simulated at the necessary fine grid resolution, and it is coupled with a standard, coarse, LES that resolves the turbulence in the whole boundary-layer. We modified the LES model PALM (Parallelized Large-eddy simulation Model) and implemented a two-way nesting technique, with coupling in both directions between the coarse and the fine grid. The coupling algorithm has to ensure correct boundary conditions for the fine grid. Our nesting algorithm is realized by modifying the standard third order Runge-Kutta time stepping to allow communication of data between the two grids. The two grids are concurrently advanced in time while ensuring that the sum of resolved and subgrid-scale kinetic energy is conserved. We design a validation test and show that the ~~temporal~~ temporally averaged profiles from the fine grid agree well compared to the reference simulation with high-resolution in the entire domain. The overall performance and scalability of the nesting algorithm is found to be satisfactory. Our nesting results in more than 80 percent savings in computational power for 5 times higher resolution in each direction in the surface layer.

1 Introduction

Turbulence in the Atmospheric Boundary Layer (ABL) encompasses a wide range of scales from the boundary-layer scale down to the viscous dissipation scale. In ABL flows, Reynolds numbers (Re) of 10^8 are commonly encountered. Explicit simulation of the Navier-Stokes equations down to the dissipative scales (DNS: direct numerical simulation) for atmospheric processes is prohibitively expensive, as the required number of grid points in one direction scales with $Re^{3/4}$ (Reynolds, 1990). This corresponds to a three-dimensional ABL simulation domain with total number of grid points of order 10^{17} . The supercomputers of today cannot fit more than 10^{12} grid points in the memory. To be able to compute turbulence processes in the atmosphere nevertheless, the concept of large-eddy simulation (LES) has been introduced already a few decades ago, e.g. ~~Deardorff (1974); Moeng and Wyngaard (1988); Schmidt and Schumann (1989)~~ Deardorff (1974), Moeng and Wyngaard (1988) and Schmidt and Schumann (1989), where the presence of a subgrid-scheme allows that only the most energetic eddies are resolved.

One of the first large-eddy simulations (LES) by Deardorff (1974) used 64000 grid points to simulate a domain of $5 \text{ km} \times 5 \text{ km} \times 2 \text{ km}$ with a grid resolution of (125, 125, 50) m. The size of one such grid cell is just sufficient to resolve the dominant ~~large-eddies~~ large eddies and there are just enough grid points to represent the ABL. As computing power progressed, higher resolution and larger domains became possible, ~~by~~ By the time of Schmidt and Schumann (1989) the number of grid cells had raised to $160 \times 160 \times 48$, simulating an ABL of $8 \text{ km} \times 8 \text{ km} \times 2.4 \text{ km}$ with a resolution of (50, 50, 50) m. Khanna and Brasseur (1998) used 128^3 grid points to simulate a domain of $3 \text{ km} \times 3 \text{ km} \times 1 \text{ km}$ to study buoyancy and shear induced local structures of the ABL. Patton et al. (2016) used (2048, 2048, 1024) grid points with a grid resolution of (2.5, 2.5, 2) m to study the influence of atmospheric stability on canopy turbulence. More recently, ~~with the help of supercomputers~~, Kröniger et al. (2018) used $13 \cdot 10^9$ grid points to simulate a domain of $30.72 \text{ km} \times 15.36 \text{ km} \times 2.56 \text{ km}$ to study the influence of ~~wind speeds~~ stability on the surface-atmosphere exchange and the role of secondary circulations in the energy exchange. The atmospheric boundary-layer community has greatly benefited from the higher ~~detail~~ spatial resolution available in these LES to study turbulent processes that cannot be ~~measured in the field in three-dimensional detail~~ obtained in field measurements.

Still, especially in heterogeneous terrain, near topographic elements, buildings or close to the surface the required higher resolution is not always attainable due to computational constraints. In spite of the radical increase in the available computing power over the last decade, large-eddy simulation of high Reynolds number atmospheric flows with very high-resolution in the surface-layer remain a challenge. Considering the size of the domain required to reproduce boundary-layer scale structures, it is ~~challenging~~ computationally demanding to generate a single fixed grid that could resolve all ~~the~~ relevant scales satisfactorily. ~~In spite of the radical increase in the available computing power, Large-Eddy Simulation of atmospheric flows with very high-resolution in the surface-layer continue to be computationally expensive. However,~~ Alternatively, local grid refinement is possible in the Finite-Volume codes that are not restricted to structured grids. Flores et al. (2013) developed a solver for the OpenFOAM modelling framework to simulate atmospheric flows over complex geometries using an unstructured mesh approach. The potential of adaptive mesh refinement technique where the tree-based Cartesian grid is refined or coarsened dynamically, based on the flow structures, is demonstrated by van Hooft et al. (2018). In the Finite-Difference models, a grid

nesting technique can be employed to ~~reduce the number of grid points required. To resolve the surface layer a vertical nesting technique is needed. Nesting has been applied to several mesoscale models (Skamarock et al., 2008; Debreu et al., 2012) , but, in contrast to horizontal nesting, vertical nesting is less common~~achieve the required resolution. In the nested grid approach, a parent domain with a coarser resolution simulates the entire domain while a nested grid with a higher resolution extends only up to the region of interest. Horizontal nesting has been applied to several mesoscale models (Skamarock et al., 2008; Debreu et al., 2012). Horizontally nested LES-within-LES or LES embedded within a mesoscale simulation is available in the Weather Research and Forecast model (Moeng et al., 2007). ~~However, we would like to point out that the vertical nesting in WRF (Daniels et al., 2016) is not a conventional vertical nesting because the parent and the child grid still have the same vertical extent, the child grid is only more refined in the vertical. A non-parallelized vertical nesting was explored by (Sullivan et al., 1996) but the code is not in public domain and we could not find any record of further development or application of this code in publications. An LES-within-LES vertical nesting is implemented by Zhou et al. (2018) in the Advanced Regional Prediction System (ARPS) model.~~ Comparable grid nesting techniques are also widely employed by the engineering turbulence research community but often ~~uses~~use different terminology. Nesting in codes with cartesian grids are referred to as local or zonal grid algorithm (Kravchenko et al., 1996; Boersma et al., 1997; Manhart, 2004) and as overset mesh (Nakahashi et al., 2000; Kato et al., 2003; Wang et al., 2014) in unstructured or moving grid codes.

For our purposes, we will focus on vertical nesting, i.e. we consider a ~~finer grid~~Fine Grid nested domain (FG) near the lower boundary of the domain, and a ~~coarser grid in the remainder~~Coarse Grid parent domain (CG) in the entire of the boundary layer. While the latter's resolution ($< 50\text{m}$) is sufficient to study processes in the outer region where the dominant eddies are large and inertial effects dominate, such coarse resolution is not sufficient ~~when~~where fine-scale turbulence in the surface layer region is concerned. The higher resolution achieved by the vertical nesting ~~there~~ will then allow a more accurate representation of the turbulence in the surface layer region, by resolving its dominant eddies. For studies that require very high resolution near the surface (e.g. virtual tower measurements, wakes behind obstacles, dispersion within street canyons for large cities) a nesting approach is an attractive solution due to the reduced memory requirement. ~~An increased resolution only in the surface layer can be achieved by a LES-within-LES set-up by maintaining the same horizontal extent for the Fine Grid nested domain (FG) and the Coarse Grid parent domain (CG). The challenge of a vertically nested grid~~Challenge of the vertically nested simulation is that the FG upper boundary conditions need to be correctly prescribed by the CG. Though vertical nesting is less common than the horizontal nesting it has been implemented in some LES models. A non-parallelized vertical nesting was explored by Sullivan et al. (1996) but the code is not in public domain and we could not find any record of further development or application of this code in publications. A LES-within-LES vertical nesting is implemented by Zhou et al. (2018) in the Advanced Regional Prediction System (ARPS) model. We would like to point out that the vertical nesting available in Weather Research and Forecast model (Daniels et al., 2016) is not a conventional vertical nesting because the parent and the child grid still have the same vertical extent, the child grid is only more refined in the vertical.

An analysis of different nesting procedures for mesoscale simulation was performed by Clark and Hall (1991), they coined the terms one-way and two-way interactions. In one-way interaction, only the FG receives information from the CG, and there is no feedback to the CG. In two-way interaction, the FG top boundary conditions are interpolated from the CG and

the CG values in the overlapping region are updated with the FG resolved fields. The 'update' process, referred to as 'interpolation' by Sullivan et al. (1996), is similar to the restriction operation in Multi-Grid methods. Harris and Durran (2010) used a linear 1D shallow-water equation to study the influence of the nesting method on the solution and found the two-way interaction to be superior ~~when-if~~ the waves are well resolved. They introduce a filtered sponge boundary condition to reduce the amplitude of the reflected wave at the nested grid boundary. We will make use of the interpolation and antinterpolation formulas of Clark and Farley (1984). Clark and Hall (1991) studied two different approaches for updating the CG values ~~namely,~~ namely "Post-insertionpost-insertion" and "Pressure-Deficit-Correctionpressure defect correction". The two approaches were also investigated by Sullivan et al. (1996) in their vertical nesting implementation. In the post-insertion technique, once the Poisson equation for pressure is solved in the FG, the resolved fields are then antinterpolated to the CG. ~~Though, they note the latter~~ In the pressure defect correction approach, the pressure in the CG and FG are matched by adding a correction term to the CG momentum equations and an antinterpolation operation is not required. Though Sullivan et al. (1996) note the pressure defect correction approach to be more elegant, no significant difference in the results was reported. ~~Moreover, only the post-insertion technique permits different temporal resolution in the FG and CG. Sullivan et al. (1996) report that in the case of their Pseudo-Spectral LES, both the resolved and SGS fluxes need to be antinterpolated to the CG and such a procedure increases coding complexity.~~

In the following sections we describe the technical realization and numerical aspects of the two-way nesting algorithm. In the LES model PALM, a validation simulation is set-up and the results of the nested and standalone simulations are compared. A second simulation is set-up to evaluate the computational performance of the algorithm. The practical considerations and the limitations of the two-way nesting are then discussed.

20 2 Methods

2.1 Description of the standard PALM ~~Model~~model

The Parallelized Large-eddy simulation Model (PALM) is developed and maintained at the Leibniz University of Hannover (Raasch and Schröter, 2001; Maronga et al., 2015). We give a quick summary of the model here and highlight the aspects which will reappear when discussing our nesting modifications. ~~For a thorough description of the model, see Maronga et al. (2015).~~ PALM is a ~~Finite-Difference~~ finite difference solver for the non-hydrostatic incompressible Navier-Stokes equations in the Boussinesq approximation (~~Eqs. 1, 2, and 3~~). PALM solves for six prognostic equations: the three components of the velocity field (u, v, w), potential temperature (θ), humidity (q) and the sub-grid scale kinetic energy (e). The sub-grid scale (SGS) turbulence is modelled based on the method proposed by Deardorff (1980) ~~with an additional equation for the turbulent kinetic energy~~. The equations for the conservation of mass, energy and moisture (Eqs. 1, 2, 3 and 4) are filtered over a grid volume on a Cartesian grid. Adopting the convention of Maronga et al. (2015), the overbar denoting the filtered variables are omitted. However, the overbar is shown for SGS fluxes. The SGS variables are denoted by a double prime. The prognostic equations

for the resolved variables are:

$$\frac{\partial u_i}{\partial t} = -\frac{\partial u_i u_j}{\partial x_j} - \varepsilon_{ijk} f_j u_k + \varepsilon_{i3k} f_3 u_{k_{\text{gs},j}} - \frac{1}{\rho_0} \frac{\partial \pi^*}{\partial x_i} + g \frac{\theta_v - \langle \theta_v \rangle}{\theta_v} \delta_{i3} - \frac{\partial}{\partial x_j} \left(\overline{u_i'' u_j''} - \frac{2}{3} e \delta_{ij} \right), \quad (1)$$

$$\frac{\partial u_j}{\partial x_j} = 0, \quad (2)$$

5

$$\frac{\partial \theta}{\partial t} = -\frac{\partial u_j \theta}{\partial x_j} - \frac{\partial}{\partial x_k} \left(\overline{u_j'' \theta''} \right) - \frac{L_V}{c_p \Pi} \Psi_{q_v}, \quad (3)$$

$$\frac{\partial q_v}{\partial t} = -\frac{\partial u_j q_v}{\partial x_j} - \frac{\partial}{\partial x_k} \left(\overline{u_j'' q_v''} \right) + \Psi_{q_v}. \quad (4)$$

The symbols used in the above equations are listed in Table 1. The 1.5 order closure parameterization modified by Moeng and

10 Wyngaard (1988) and Saiki et al. (2000), assumes a gradient diffusion parameterization (~~Eq. 5~~Eqs. 6, 7, 8). The prognostic equation for the SGS-TKE reads as

$$\frac{\partial e}{\partial t} = -u_j \frac{\partial e}{\partial x_j} - \left(\overline{u_i'' u_j''} \right) \frac{\partial u_i}{\partial x_k} + \frac{g}{\theta_{v,0}} \overline{u_3'' \theta''} - 2K_m \frac{\partial e}{\partial x_j} - \epsilon, \quad (5)$$

with the SGS fluxes modelled as:

$$\overline{u_i'' u_j''} - \frac{2}{3} e \delta_{ij} = -K_m \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (6)$$

15

$$\overline{u_i'' \theta''} = -K_h \frac{\partial \theta}{\partial x_i}, \quad (7)$$

and

$$\overline{u_i'' q_v''} = -K_h \frac{\partial \theta}{\partial x_i}. \quad (8)$$

The eddy diffusivities are proportional to $e^{3/2}$ under convective conditions (Maronga et al., 2015). For a thorough description
20 of the governing equations and parameterizations, see Maronga et al. (2015).

The prognostic equations are discretized on a staggered Arakawa C-grid, where the scalars are evaluated in the center of the grid volume and velocities are evaluated at the center of the faces of the grid volume in their respective direction.

The advection terms are evaluated either with fifth-order upwind discretization according to Wicker and Skamarock (2002)
25 or with a 2nd order scheme according to Piacsek and Williams (1970). The prognostic equations are integrated in time using

Table 1. List of symbols in the governing equations and parameterizations.

<u>Symbol</u>	<u>Description</u>
f_i	<u>Coriolis parameter</u>
ρ_0	<u>Density of dry air at the surface</u>
π^*	<u>Modified perturbation pressure</u>
g	<u>Gravitational acceleration</u>
θ_v	<u>Virtual potential temperature</u>
L_v	<u>Latent heat of vaporization</u>
C_p	<u>Heat capacity of dry air at constant pressure</u>
q_v	<u>Specific humidity</u>
Ψ_{q_v}	<u>Source/sink term of q_v</u>
Π	<u>Exner function for converting between temperature and potential temperature</u>
K_h	<u>SGS eddy diffusivity of heat</u>
K_m	<u>SGS eddy diffusivity of momentum</u>

a third-order Runge-Kutta (RK3) scheme. The low storage RK3 scheme with three sub-steps proposed by Williamson (1980) guarantees a stable numerical solution. The Poisson Equation for pressure is solved with Fast-Fourier Transform (FFT) when periodic boundary conditions are applied in the lateral boundaries. There are three FFT algorithms available in PALM with FFTW being the optimal method for large scale simulations. Monin-Obukhov Similarity Theory (MOST) is assumed between the surface and the first grid point. A vertical zero pressure gradient at the surface guarantees the vertical velocity to be zero. Simulations can be driven by either prescribing the surface temperature or the surface sensible heat flux, similarly for the humidity. At the top of the simulation domain the horizontal velocities equal geostrophic wind and the vertical velocity is set to zero. The pressure can assume either a Dirichlet condition of zero value or a Neumann condition of zero vertical gradient. The scalar values ~~have~~ can have either a fixed value Dirichlet condition or a fixed gradient Neumann condition. The vertical gradient of SGS Turbulent Kinetic Energy (TKE) is set to zero at both top and bottom boundaries.

PALM is a parallelized model and the standard way of parallelization is by dividing the three-dimensional domain into vertical columns, each of which is assigned to one processing element (PE). Each vertical column possesses a number of ghost points needed for computation of derivatives at the boundary of the sub-domains. Each PE can only access data for a single sub-domain. All PEs execute the same program on a different set of data. For optimum load balancing between the PE the decomposed sub-domains should have the same size. In PALM, this condition is always satisfied as only sub-domains of the same size are allowed. ~~After every time-integration step, the~~ The data exchange between PEs ~~, needed~~ needed by the Poisson solver and to update the ghost points ~~, is~~ are performed via the Message Passing Interface (MPI) communication routines.

2.2 ~~Fine grid and coarse grid configuration~~ Nested model structure

2.2.1 Fine grid and coarse grid configuration

We are interested in achieving an increased resolution only in the surface-layer, the lowest 10% of the boundary layer, where surface exchange processes occur and where eddies generated by surface heterogeneity and friction are smaller than the dominant eddies in the mixed layer. We setup the LES-within-LES case by maintaining the same horizontal extent for the FG and the CG to have the whole surface better resolved. We allow the vertical extent of the FG to be varied as needed, typically up to the SL depth. This implementation of vertical grid nesting has two main challenges. The first challenge, that is purely technical in nature, is to implement routines that handle the communication of data between the CG and the FG. The second and the most important challenge is to ensure that the nesting algorithm yields an accurate solution in both ~~the~~ grids.

Below we use upper case symbols for fields and variables in the CG, and lower case for the FG. E.g. E and e denote the subgrid-scale turbulent kinetic energy (a prognostic variable in our LES) of CG and FG respectively. The nesting ratio is defined as the ratio of the CG spacing to the FG spacing, and $n_x = \Delta X / \Delta x$, corresponding symbols apply for y and z directions. The nesting ratios n_x , n_y and n_z have to be integer. It is possible to have either odd or even nesting ratio and it can be different in each direction. As the domain that is simulated in the FG is completely inside of the CG domain, each FG cell belongs to a CG cell. The two grids are positioned in such a way that a FG cell belongs to only one CG cell and one CG cell is made up by a number of FG cells given by the product of the nesting ratios $n_x \times n_y \times n_z$. This means that if the grid nesting ratio is odd, there will be one FG cell whose center is exactly at the same position as the center of the coarse cell as shown in Fig. 1 (b). The collection of FG cells that correspond to one CG cell is denoted by $\mathcal{C}(I, J, K)$, the collection of FG faces that corresponds to e.g. an yz -face of the CG is denoted by $\mathcal{C}_x(I_s, J, K)$, where it is understood that the I_s index is an index on the staggered grid in the x -direction to denote the position of the face, and similar for the other types of faces. ~~Below we use upper case symbols for fields and variables in the CG, and lower case for the FG. E.g. E and e denote the subgrid-scale turbulent kinetic energy (a prognostic variable in our LES) of CG and FG respectively.~~ We have used $f_x = 1/n_x$ to denote the inverse of the nesting ratio in the x dimension (corresponding symbols for y and z). A schematic diagram of the overlapping grids ~~are~~ is shown in Fig. 1 ~~(a)~~ (a).

~~For the boundary conditions at the top of the FG, the fields from the CG are interpolated to the FG, according to Clark and Farley (1984). In Eq. ??, Φ and ϕ represent CG and FG quantities, respectively. For the scalar fields, the interpolation is quadratic in all three directions. For the velocity components, the interpolation is linear in the logical direction of that component, and quadratic in the other two directions. A similar interpolation is carried out for the initialization of the fine grid. The quadratic interpolation equation reads as-~~

~~The antinterpolation of the prognostic quantities are performed by an averaging procedure according to Clark and Hall (1991). The scalars are defined as the spatial average over the whole grid cell, therefore it is required that the CG scalar is the average of the corresponding FG scalars (Eq. 15). However, the CG velocity components are the average of only the FG value at the corresponding faces (Eq. 14). The antinterpolation equations read as-~~

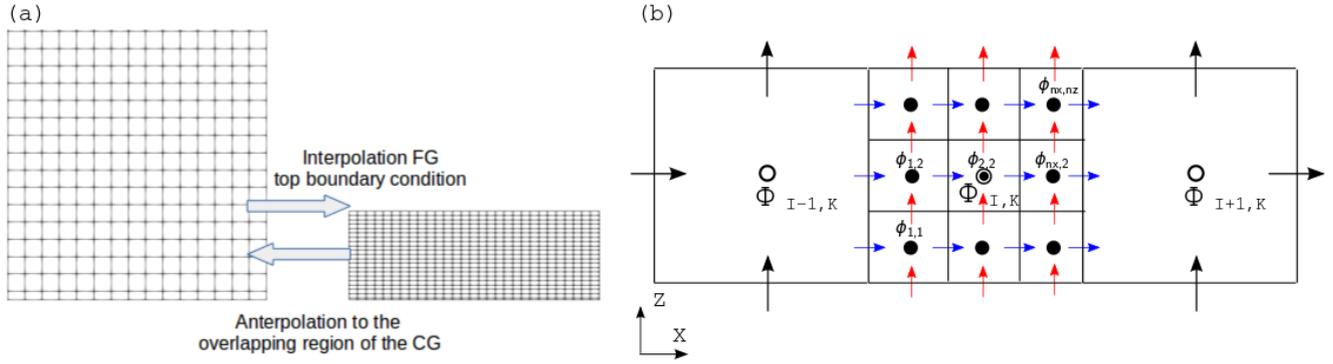


Figure 1. (a) Schematic of the interpolation and antierpolation between the grids. The FG top boundary condition is interpolated from the CG. The CG prognostic quantities in the overlapping region are antierpolated from the FG. (b) Schematic of Arakawa C grid for two grids with nesting ratio of three. The black arrows and circles are CG velocity and pressure, respectively. The blue and red arrows are horizontal and vertical velocity, respectively, in the FG. The filled black circle is the FG pressure. Where I and K are CG indices and nx and nz are the nesting ratio in x and z , respectively.

However, the TKE in the CG differs from the FG value. Due to the different resolution of the grids, in the FG the SGS motions are weaker. Therefore, TKE is antierpolated such that the Germano identity is maintained (Germano et al., 1991):

2.3 Summary of the Nesting Algorithm

2.2.1 Vertical nesting algorithm

- 5 We implement a two-way interaction algorithm, shown in Fig. 2, because in our first trials we found that one-way nesting did not improve the FG representation satisfactorily and hence was not pursued further. The FG prognostic quantities are initialized by interpolating the CG values in the overlapping region. Optionally, the initialization of the FG can be delayed until the CG has reached a fully turbulent state. Both the grids are restricted to have identical time steps. PALM finds the largest time step for each grid such that the CFL condition is individually satisfied and the minimum of the two values is then chosen
- 10 as the time integration step for both the grids. The right hand side of the prognostic equation except for the pressure is first computed concurrently in both the grids. The values of u, v, w, θ and q are then antierpolated to the CG in the overlapping region. The CG solves a Poisson equation for pressure. The new u, v, w, θ and q fields in the CG are interpolated to set the FG Dirichlet top boundary conditions. The Poisson equation is then solved for pressure in the FG and the vertical velocity in the FG is also updated by the pressure solver at this stage. Since all the velocity components follow Dirichlet condition at FG top boundary only Neumann condition is suitable for pressure (Manhart, 2004). PALM permits the use of a Neumann zero-gradient condition for pressure at both top and bottom boundary. It is advisable to use a Neumann boundary condition at the top and the bottom for the CG too. The TKE is then antierpolated maintaining the Germano identity and it is followed
- 15 by the computation of SGS eddy diffusivity for heat (k_h) and momentum (k_m) in the CG. This procedure is repeated at every

For the boundary conditions at the top of the FG, the fields from the CG are interpolated to the FG, according to Clark and Farley (1984). We define the top of the FG as the boundary level just above the prognostic level of each quantity. In Eq. 10, Φ and ϕ represent CG and FG quantities, respectively. For the scalar fields, the interpolation is quadratic in all three directions. For the velocity components, the interpolation is linear in its own dimension, and quadratic in the other two directions. The same interpolation formulation is also used to initialize all vertical levels of the fine grid domain at the beginning of the nested simulation. The interpolation is reversible as it satisfies the conservation condition of Kurihara et al. (1979):

$$\langle \phi \rangle = \langle \Phi \rangle . \quad (9)$$

For clarity, we illustrate the interpolation by focusing on one particular dimension, in this case x , but the same operation holds for y and ~~bottom for the CG too~~ z . The interpolation in the x -dimension reads as

$$\phi_m = \eta_-^m \Phi_{I-1} + \eta_0^m \Phi_I + \eta_+^m \Phi_{I+1}, \quad (10)$$

with m running from 1 to n_x , thus producing n_x equations for each CG cell I . For the interpolation in y and z there will be two additional indices, producing $n_x \times n_y \times n_z$ equations for all the FG cells corresponding to the CG parent cell. For the quadratic interpolation a stencil with 3 legs is used, relating the prognostic value of a FG cell to the value of its parent CG and the values of the immediate CG neighbour on the left and on the right of the parent cell, e.g. Φ_{I-1} and Φ_{I+1} for the x direction

as shown in Fig. 1 (b). The stencil coefficients are:

$$\begin{aligned} \eta_-^m &= \frac{1}{2} H_m (H_m - 1) + \alpha, \\ \eta_0^m &= (1 - H_m^2) - 2\alpha, \\ \eta_+^m &= \frac{1}{2} H_m (H_m + 1) + \alpha, \end{aligned} \quad (11)$$

with the weights H_m expressed in function of the inverse nesting ratio,

$$H_m = \frac{1}{2} ((2m - 1)f_x - 1), \quad (12)$$

and the coefficient α is chosen such that the conservation condition of Kurihara et al. (1979) is satisfied,

$$\alpha = \frac{1}{24} (f_x^2 - 1). \quad (13)$$

It can be observed that the sum of the η 's equals 1.

2.3.2 Anterpolation

The interpolation of the prognostic quantities are performed by an averaging procedure according to Clark and Hall (1991).

The interpolation equations for the velocities read as:

$$\begin{aligned}
 U_{I,J,K} &= \langle u \rangle_{j,k} = \sum_{j,k \in \mathcal{C}_{IJK}} u_{i^*,j,k} f_y f_z, \\
 V_{I,J,K} &= \langle v \rangle_{i,k} = \sum_{i,k \in \mathcal{C}_{IJK}} v_{i,j^*,k} f_x f_z, \\
 W_{I,J,K} &= \langle w \rangle_{i,j} = \sum_{i,j \in \mathcal{C}_{IJK}} w_{i,j,k^*} f_x f_y.
 \end{aligned} \tag{14}$$

For the scalars it is:

$$5 \quad \Phi_{I,J,K} = [\phi]_{i,j,k} = \sum_{i,j,k \in \mathcal{C}_{IJK}} \phi_{i,j,k} f_x f_y f_z. \tag{15}$$

Here the lower case indices only count over the fine grid cells that belong to that particular coarse grid cell. For each (I, J, K) tuple of a parent CG cell there exists a set \mathcal{C}_{IJK} containing the (i, j, k) tuples of its corresponding children FG cells. To ensure that the nested PALM knows at all times which fine grid cells and coarse grid cells correspond, we compute this mapping for the FG and CG indices before starting the simulation, and we store it in the memory of the parallel processing element. In the Arakawa C-grid discretization that PALM uses, the scalars are defined as the spatial average over the whole grid cell, and therefore it is required that the CG scalar is the average of the corresponding FG scalars in (Eq. 15). However, the velocities are defined at the faces of the cells in the corresponding dimension. Therefore in (Eq. 14) the CG velocity components are computed as the average over the FG values at the FG cells that correspond to the face of the CG cell, expressed by i^*, j^*, k^* respectively.

15 However, the TKE in the CG differs from the FG value, due to the different resolution of grids. In the FG the SGS motions are weaker because the turbulence is better resolved. Therefore, TKE is interpolated such that the sum of resolved kinetic energy and TKE (SGS kinetic energy) is preserved, by maintaining the Germano identity (Germano et al., 1991):

$$E = [e] + \frac{1}{2} \sum_{n=1}^3 ([u_n u_n] - [u_n][u_n]). \tag{16}$$

20 Here the straight brackets are the spatial average over the coarse grid cell ($f_x f_y f_z \times \sum_{i,j,k \in \mathcal{C}_{IJK}}$) and the n index runs over the three spatial dimensions. In other words, to obtain the CG TKE from the average FG TKE, we add the variance of the FG velocity components over the FG cells comprising the CG cell. Therefore CG TKE is always larger than FG TKE.

2.4 Parallel Inter-Grid Communication

MPI is the most widely used large scale parallelization library. The atmosphere-ocean coupling in PALM has been implemented following MPI-1 standards (Esau, 2014; Maronga et al., 2015). We follow a similar approach for the MPI communications, and have adopted MPI-1 standards for our nesting implementation. Concurrent execution of the two grids is achieved with the MPI_COMM_SPLIT procedure. The total available processors are split into two groups, denoted by color 0 or 1 for CG

and FG respectively, see Fig. 3. The data between the processors of the same group are exchanged via the local communicator created during the splitting process. ~~Whereas,~~ whereas the data between the two groups are ~~exchange~~ exchanged via the global communicator MPI_COMM_WORLD.

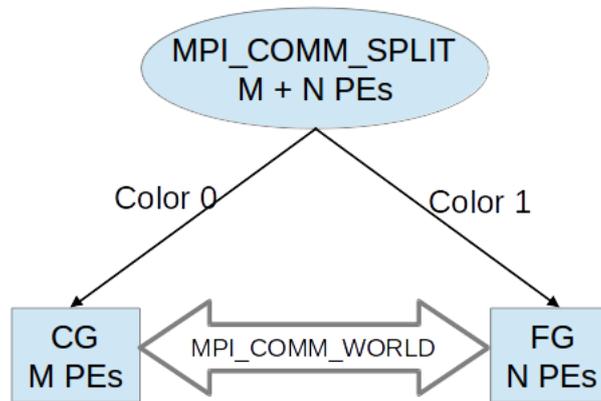


Figure 3. Schematic of the MPI processor grouping. The data exchange between the two groups are performed via the global communicator. M and N are the number of processors for CG and FG respectively.

Based on the nesting ratio and the processor topology of the FG and the CG group a mapping list is created and stored. Given the local PE's 2D processor co-ordinate – the list will identify the PEs in the remote group to/from which data needs to be sent/received; the actual communication then takes place via the global communicator. There are three types of communication in the nesting scheme:

- i. Initializing the FG (Send data from coarse grid to fine grid.) This is performed only once.
- ii. Boundary condition for the FG top face (Send data from coarse grid to fine grid.).
- 10 iii. Anterpolation (Send data from fine grid to coarse grid.).

The exchange of arrays via MPI_SENDRECV routines is computationally expensive. Therefore, the size of the arrays communicated are minimized by performing the anterpolation operation in the FG PE's and storing the values in a temporary 3D array that is later sent via the global communicator to the appropriate CG PE. This approach ~~also takes advantage of the higher number of PE available in~~ is more efficient than performing the anterpolation operation on the CG which has less PE's and needs communication of larger arrays from the FG. Furthermore, the array data that need to be communicated during the anterpolation operation and for setting the FG boundary condition are not contiguous in memory. The communication performance is enhanced by creating an MPI derived data type that ensures that the data is sent contiguously. Within the RK3 sub-steps, when one grid executes the pressure solver, the other grid has to wait, leading to more computational time at every sub-step. However, the ~~delay~~ waiting time can be minimized by effective load balancing, i.e. the number of grid points per PE

in the CG should be kept lower than in the FG. ~~This~~ The reduction in workload per ~~PE can be achieved with just~~ CG PE is achieved with a few additional cores. The reduction in computational time per step in the CG means the waiting time on the FG PE is also reduced.

3 Results and Discussion

5 3.1 Simulation ~~set-up~~ setup for the nesting validation test

To evaluate the accuracy of the two-way nesting algorithm we ~~set-up~~ setup a convective boundary layer simulation. Two overlapping grids with a nesting ratio of five in the lateral and vertical direction are employed. The simulation parameters are listed in Table 2. A standalone reference simulation with the same resolution as the coarse grid (SA-C) and another reference with the same resolution as the fine grid (SA-F) are performed for comparison. The grid configuration and the computational
10 resources used are listed in Table 3. The simulations were performed in a local computing cluster, each compute node has 64 GB of main memory and a 2.8 GHz Ivy Bridge processor with 20 cores. The simulation domain ~~is set to~~ has periodic boundary conditions in the lateral direction. ~~At~~ The Dirichlet boundary condition is applied for velocity at the top and bottom boundaries, the ~~velocity~~ vertical velocity component is set to zero and the horizontal components are set to ~~Dirichlet condition~~ geostrophic wind. At the top and bottom boundaries, the pressure and humidity are set to zero gradient Neumann condition. The potential
15 temperature is set to a Neumann condition at the bottom, and the gradient is determined by MOST based on the prescribed surface heat flux and roughness length. The gradient of the initial profile is maintained at the top boundary. In PALM, u_g and v_g represents the u - and v -component of the geostrophic wind at the surface. The u and v ~~profiles are constructed starting from a zero value at the surface and reaches~~ initial profiles are set to be constant, equal to the value of the geostrophic wind ~~value at the top. The component in the domain and the~~ vertical velocity is initialized to zero in the domain. The potential temperature is
20 initialized to a constant value of 300 K up to 800 m and above 800 m a lapse rate of ~~1 K 100 m⁻¹~~ 1 K (100 m)⁻¹ is prescribed. The humidity profile is initialized to a constant value of 0.005 ~~kg kg⁻¹~~ kg kg⁻¹. The simulation is driven by prescribing a surface heat flux of 0.1 ~~K ms⁻¹~~ K ms⁻¹ and a surface humidity flux of 4×10^{-4} ~~kg kg⁻¹ ms⁻¹~~ kg kg⁻¹ ms⁻¹. The domain is more than four times larger in the horizontal than the initial boundary layer height.

25

3.2 Analysis of the simulations

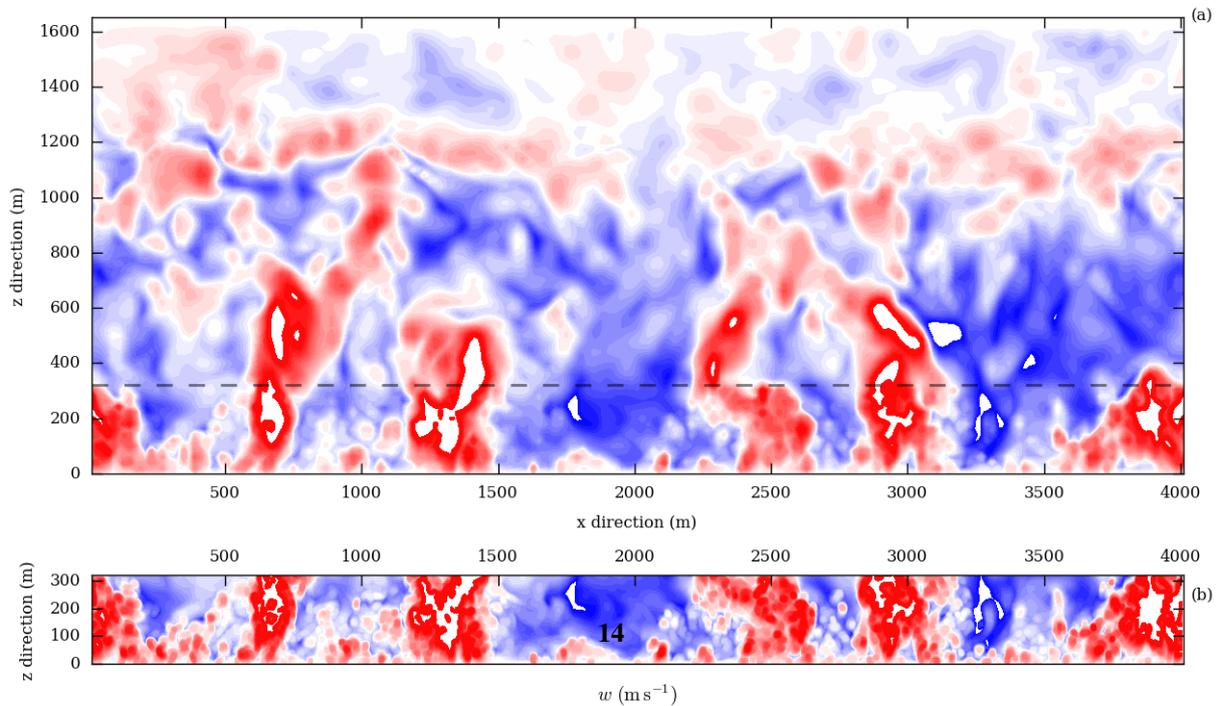
In a two-way nesting it is important that the flow structures are propagated from the FG to CG and vice versa, without any distortion. In Fig. 4, the contours in the CG region overlapping the FG have similar structures as the FG. The higher resolution in the FG enables more detailed contours whereas the antepolated CG contours are smoother. Furthermore, in the CG region
30 beyond the overlapping region no distortion to the contours are observed indicating that the antepolation does not introduce sharp gradients in the CG.

Table 2. Simulation Parameters for the nesting validation test.

<u>Simulation Parameters</u>	<u>Value</u>
Domain Size:	4.0 x 4.0 x 1.65 km ³
Fine grid vertical extent:	320 m
Kinematic surface heat flux:	$w'\theta'_0 H_s = 0.1 \text{ Kms}^{-1} \text{ Kms}^{-1}$
Kinematic surface humidity flux:	$w'q'_0 \lambda E_s = 4 \times 10^{-4} \text{ kgkg}^{-1}\text{ms}^{-1} \text{ kgkg}^{-1}\text{ms}^{-1}$
Geostrophic wind:	$u_g = 1 \text{ ms}^{-1} \text{ ms}^{-1}, v_g = 0 \text{ ms}^{-1} \text{ ms}^{-1}$
<u>Roughness length</u>	<u>0.1 m</u>
Simulated time:	10800 s
Spin-up time:	9000 s
Averaging interval:	1800 s

Table 3. Grid configuration of the nested and standalone reference domains.

<u>Case</u>	<u>No. of Grid-Points-grid points</u>	<u>(dx,dy,dz) m</u>	<u>cpu-CPU cores</u>	<u>core-hours-Core-hours</u>	<u>Grid points per core</u>	<u>T</u>
Coarse Grid (CG)	200 x 200 x 80 = 3.2 x 10 ⁶	20, 20, 20	20	290-376	1.6 x 10 ⁵	
Fine Grid (FG)	1000 x 1000 x 80 = 80 x 10 ⁶	4, 4, 4	80	1160-1503	1.0 x 10 ⁶	
Total				1450-1879		
Standalone Coarse (SA-C)	200 x 200 x 80 = 3.2 x 10 ⁶	20, 20, 20	20	5-8	1.6 x 10 ⁵	
Standalone Fine (SA-F)	1000 x 1000 x 400 = 400 x 10 ⁶	4, 4, 4	400	10345-8234	1.0 x 10 ⁶	



~~The vertical~~ Vertical profiles are used for quantitative comparison of the nested and the reference simulations. The turbulent fluctuations (e.g. θ'' , w'') are defined as the spatial deviations from the instantaneous horizontal average. The turbulent ~~flux profiles fluxes~~ (e.g. $w''\theta''$, $u''u''$, $\langle w''\theta'' \rangle$, $\langle u''u'' \rangle$) are obtained using the spatial covariance and are then horizontally averaged. All the horizontally averaged profiles (e.g. $\langle \theta \rangle$, $\langle w''\theta'' \rangle$) are also averaged over time but we omit the conventional overline notation for ~~simplicity~~. ~~The friction velocity (u_*)~~, ~~readability~~. The convective velocity scale (w_*) and temperature scale (θ_*) obtained from SA-F are used to normalize the profiles. The convective velocity is calculated as $w_* = (g\theta_0^{-1}w'\theta'_0z_i)^{1/3}$ $w_* = (g\theta_0^{-1}H_s z_i)^{1/3}$, where g is the gravitational acceleration, θ_0 is the surface temperature and z_i is the boundary layer height in the simulation. The convective temperature scale is calculated as $\theta_* = w'\theta'_0 w_*^{-1}$. ~~The~~ $\theta_* = H_s w_*^{-1}$. In Fig. 5 (a and c), the vertical profiles of difference between the potential temperature ($\langle \theta \rangle$) and its surface value normalized by the ~~surface value are shown in Fig. 5 (a and c)~~ convective temperature scale are plotted. Since the FG profiles are superior to the CG in the ~~over-lapping overlapping~~ region, the antepolated CG values are not plotted. In Fig. 5 (a), there is no visible difference between the standalone and the nested simulations. However, in the region closer to the surface, plotted in Fig. 5 (c), a better agreement between the SA-F and FG is observed. The potential temperature variance ($\langle \theta''\theta'' \rangle$) normalized by the square of the temperature scale (θ_*^2) ~~are is~~ shown in Fig. 5 (b and d). Here too FG provides better accuracy close to the surface. ~~It is important to note~~

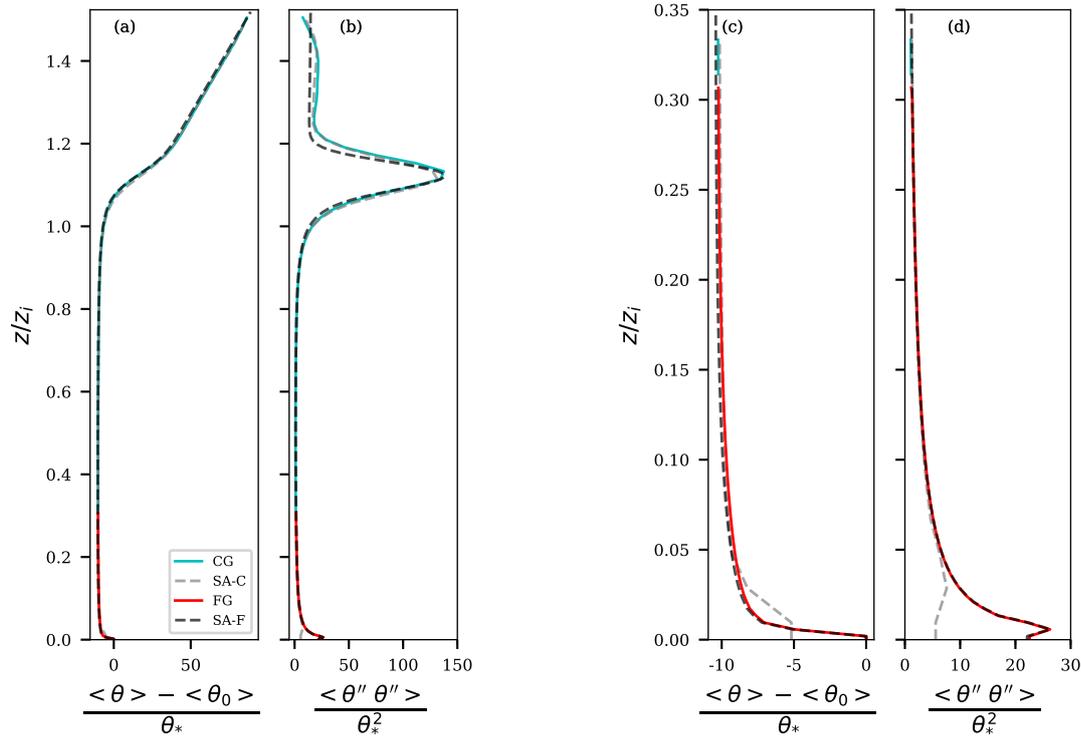


Figure 5. Vertical profile of horizontally averaged potential temperature normalized by surface value (a and c) and variance of potential temperature normalized by θ_*^2 (b and d). The nested grid profiles agree well with the SA-F in the surface layer. The improvement of the two-way nesting, at the boundary layer height, is seen in the good agreement in the profiles of CG and SA-F in (b).

In the vertical heat flux ($\langle w''\theta'' \rangle$) profiles in Fig. 5 (b), at the boundary layer height well above the overlapping region, the two-way nesting improves the CG as compared to the FG has good agreement with the SA-F in the surface layer for the resolved, SGS and the total flux profiles. In the CG regions above the nested grid height, a good agreement with the SA-C profile. An one-way-nested simulation will not benefit from the higher surface is found as well. The improvement due to the two-way nesting is seen in Fig. 6 (d and e), where the effects of low grid resolution of the FG as there is no feedback to the CG.

SA-C in resolved and SGS fluxes are evident. However, no grid dependent difference in the profile is observed in the total flux.

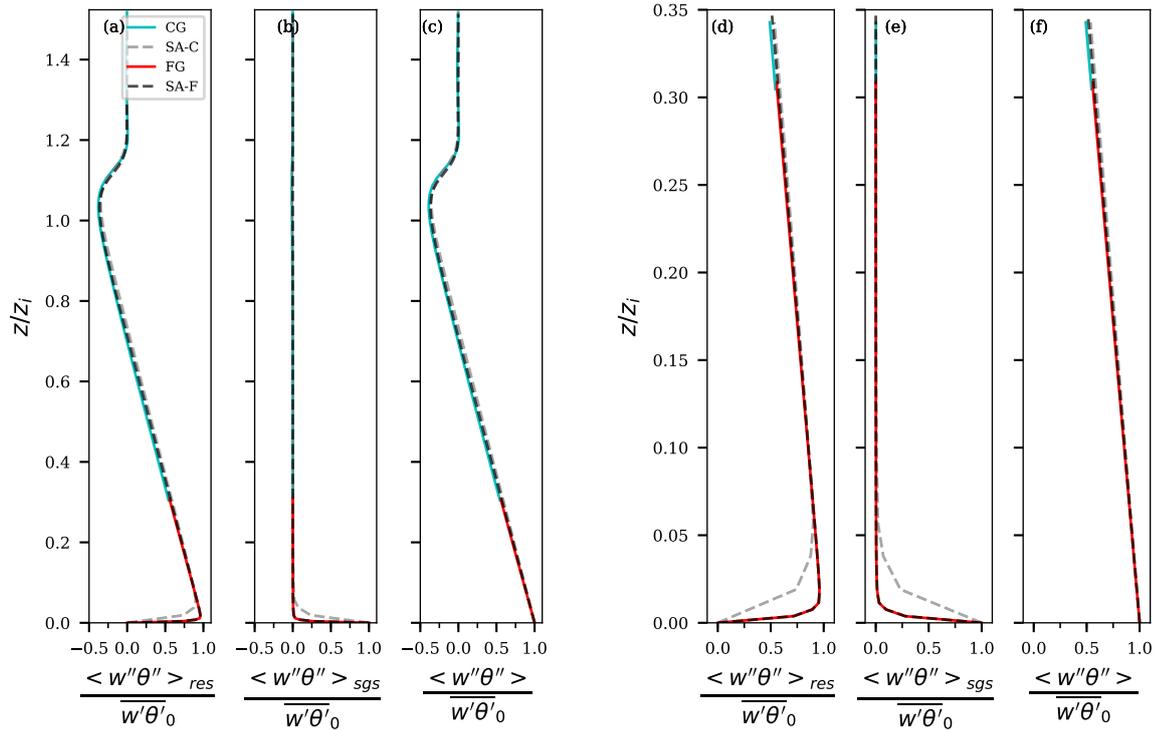


Figure 6. Vertical profile of horizontally averaged potential-temperature heat flux normalized by the surface value heat flux – resolved (a and ed) and variance of potential temperature normalized by θ_*^2 , sub-grid (b and de), and total flux (c and f). The nested grid-profiles agree well with the SA-F in the surface layer. The improvement of the two-way nesting, at the boundary layer height, is seen in the good agreement in significantly improves the profiles of CG resolved and SA-F SGS fluxes in (b) the surface layer.

The variance resolved variances of u , v and w velocity components are plotted in Fig. 7. The u and v variance are normalized by the square of the friction velocity (u_*^2) and the w variance is normalized by the square of the convective velocity (w_*^2). The u and v are plotted in Fig. 7. The FG v and w FG profiles have a better agreement with the SA-F than the v variance. A marginal overshoot of the u variance. The u and v variance seen variances in Fig. 7 (e). We can notice the remainder of a small kink in the vertical velocity variance d and e) lie between SA-C and SA-F indicating that the resolved variances are improved compared to the SA-C but not sufficiently resolved to match SA-F. At the nesting height the variances deviate more from the SA-F and approach the CG values. Due to conservation of total kinetic energy across the nest boundary, more CG TKE is contained in the sub-grid scale. Consequently, the resolved CG variances could have an undershoot as compared to SA-F, resulting in an undershoot of the FG variances too at the nesting height. This is a side effect of the antepolation and Above the nesting height, the variance of u , v and w in CG are similar to SA-C.

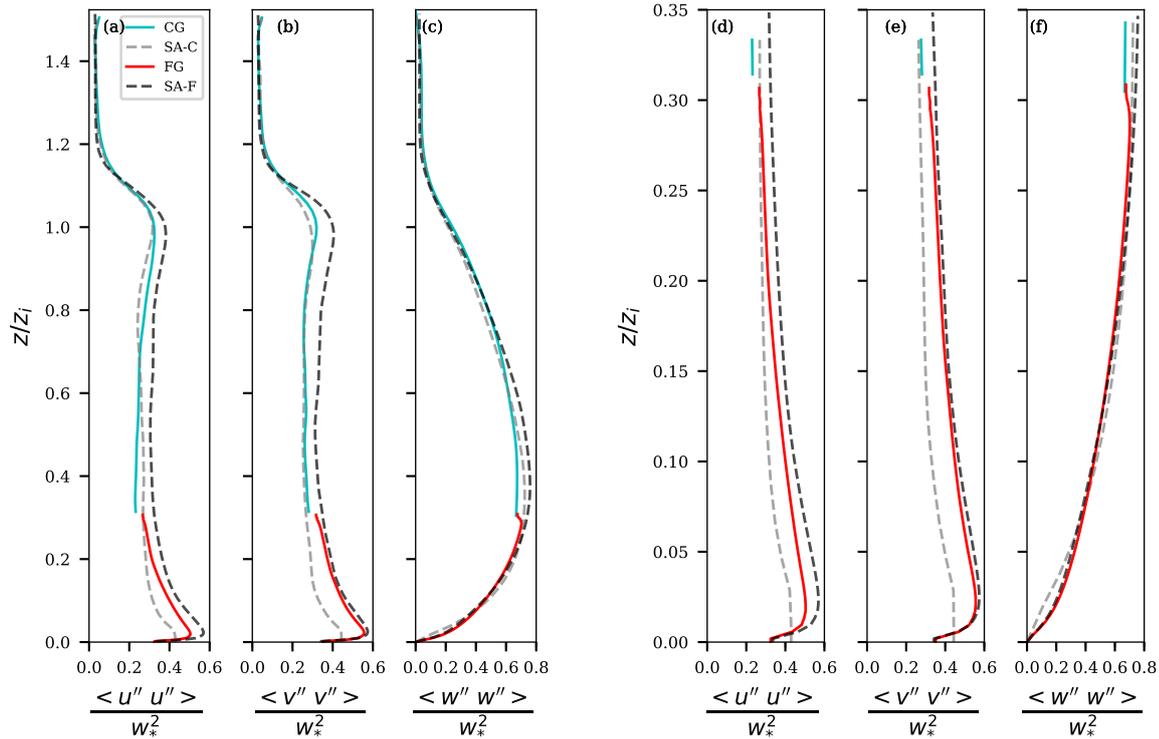


Figure 7. Vertical profile of horizontally averaged resolved variance of u (a and d), v (b and e), and w (c and f) normalized by w_*^2 . The variance of v and w show better agreement with the standalone reference in the surface layer.

The resolved vertical velocity skewness in Fig. 8 shows good agreement between the FG and SA-F close to the surface. However, at the nesting height a small kink in the skewness is noticeable. Zhou et al. (2018) observe that the magnitude of the kink in the higher-order profiles can be minimized by using a sponge layer, see Sullivan et al. (1996). In our two-way nesting we have used a increasing the depth of the sponge layer. Our simplified sponge layer by limiting the anteroplation to one CG cell less than the nested height, this segregation of the anteroplation region in the CG and top boundary condition level of the FG provides reasonable reduction of kinks in the profile.

approach appears to be unable to effectively minimize the kinks at the nesting height. The resolved skewness in CG is lower than SA-C possibly due to larger SGS TKE in the CG, as seen in Fig. 8 (d). The SGS TKE in Fig. 8 (d) shows exact match between FG and SA-F close to the surface and only marginal difference at the nesting height. However, CG values are considerably different from the SA-C values close to the surface due to the anteroplation maintaining Germano identity for conservation of kinetic energy across the grids. In the coarse resolution SA-C, near the surface, the SGS turbulence model appears to insufficiently model the SGS effects. Above the nesting height the CG is similar to SA-C.

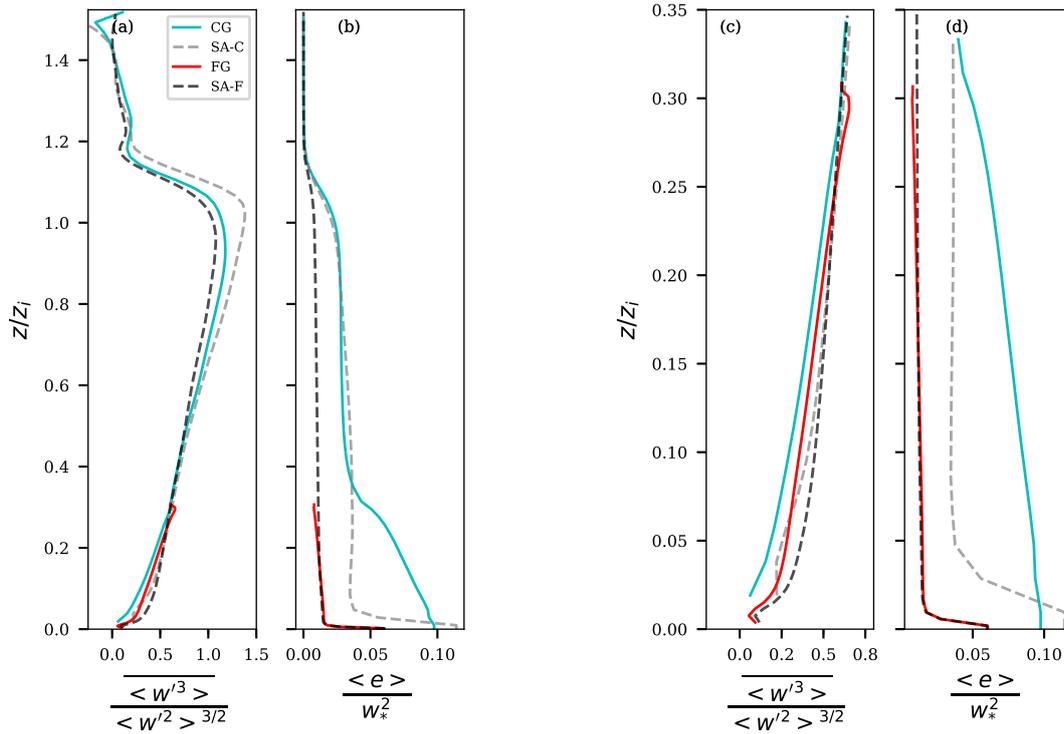


Figure 8. Vertical profile of horizontally averaged variance of u resolved vertical velocity skewness (a and d), v (b and ec), and w -SGS turbulent kinetic energy e (e-b and fd). The horizontal velocity components are normalized by u_*^2 and the vertical velocity variance is normalized by w_*^2 . The variance of u and w show better agreement with the standalone reference SGS TKE in the surface layer CG is higher than SA-C as a result of anterpolation maintaining the Germano identity.

The vertical heat flux profile is the prime quantity of interest in analysing surface layer simulations. In the $\langle w''\theta'' \rangle$ profiles horizontal spectra of SGS turbulent kinetic energy and vertical velocity are plotted in Fig. 6, the FG have perfect agreement with the SA-F in the surface layer for the resolved, SGS and the total flux profiles. In the CG regions above the nested grid height, too a good agreement with the SA-C is found. The improvement due to 9 at two levels, one within the nested grid and one above the nested grid height. The FG TKE spectra in Fig. 9 (c) perfectly overlaps the SA-F spectra. The CG spectra has higher energy than the SA-C, this corresponds to the higher CG TKE values observed in Fig. 8 (c). As the limit of the grid resolution is reached at high wavenumber, the two-way nesting is seen drop in the CG spectra is marginally shifted compared to SA-C. This improvement at high wavenumber is due to feedback from the FG. Similarly, the vertical velocity spectra in Fig. 6 (d and e), where the effects of low grid resolution of the 8 (d) shows marginal improvement at high wavenumber for the CG with respect to SA-C in resolved and SGS fluxes are evident. However, no griddependent difference in the profile is observed in the total flux. While the FG agrees with SA-C at high wavenumber and at the spectra peak, at low wavenumber

FG follows the CG spectra. At the level above the nested grid, the CG spectra agrees with SA-C for both TKE and the vertical velocity.

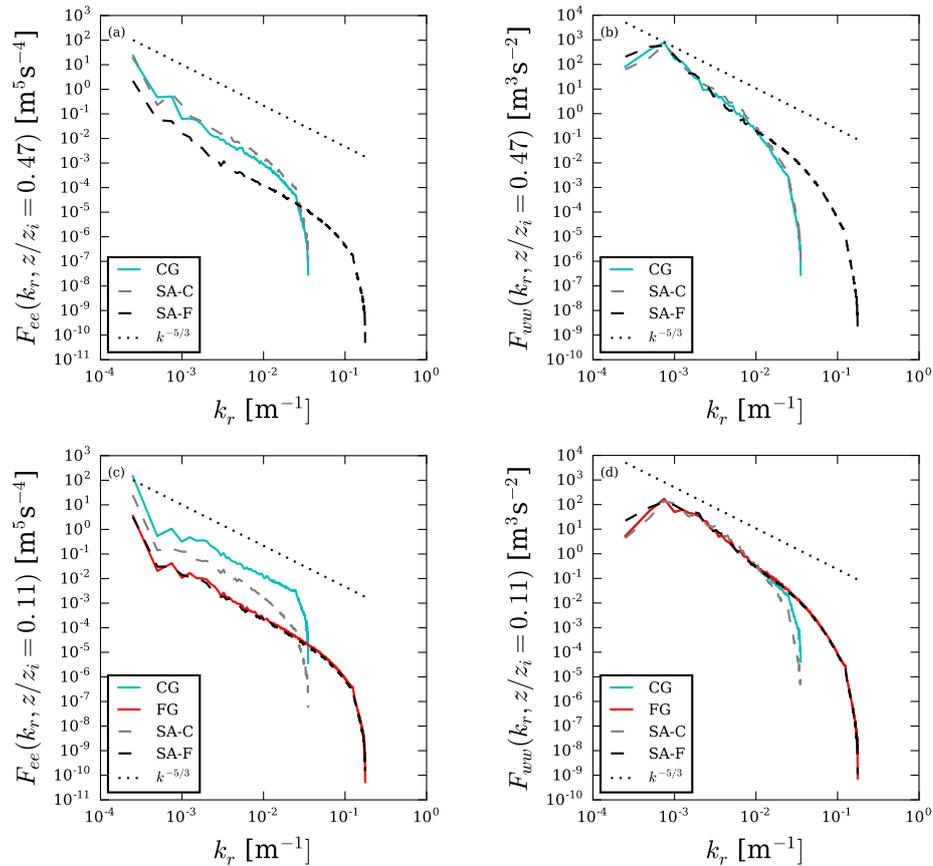


Figure 9. Vertical-profile Spectra of horizontally-averaged heat flux normalized by the surface heat flux—resolved SGS turbulent kinetic energy (e) (a and dc), sub-grid and vertical velocity (w) (b and ed). At $z/z_i = 0.47$ (a and total flux-b) and at $z/z_i = 0.11$ (c and fd). The two-way nesting significantly improves the resolved and SGS fluxes in: k_r is the surface layer horizontal wavenumber.

3.3 Computational Performance

The computational resources used in the simulations discussed above are listed in Table 3. The resources needed by SA-C is only 5–8 core hours. While the nested simulations needed about 1450–1879 core hours, the SA-F needed 7–about 4 times more core hours. If we increase the resolution further, the time step Δt will get smaller and consequently increase the total number of steps than the nested simulation. As the resolution is increased from 20 m in SA-C to be integrated. Simulations with $O(1)$ m resolution become prohibitively expensive to achieve. The two-way nesting algorithm reduces the memory requirement and 4 m in SA-F, the number of time steps increased more than 5 times as higher resolution demands smaller time step size. Though the number of time steps in FG is similar to SA-F, limiting the nested grid in the vertical direction has reduced the

number of ~~core hours needed by providing higher resolution only~~ CPU cores needed, and higher resolution in the surface layer ~~is achieved at a reduced computational cost.~~

Table 4. Number of grid points in nested and non-nested FG domain.

<u>Case</u>	<u>No. of grid points</u>
<u>Coarse Grid</u>	<u>$840 \times 840 \times 288 = 0.20 \times 10^9$</u>
<u>Fine Grid</u>	<u>$4200 \times 4200 \times 360 = 6.35 \times 10^9$</u>
<u>Total</u>	<u>6.55×10^9</u>
<u>Non-nested FG</u>	<u>$4200 \times 4200 \times 360 = 6.35 \times 10^9$</u>

Table 5. Grid configuration of the nested and non-nested FG domain.

<u>Run</u>	<u>Nested</u>			<u>Non-Nested FG</u>				
	<u>Total PE</u>	<u>CG PE</u>	<u>FG PE</u>	<u>Avg. time per step [s]</u>	<u>Efficiency [%]</u>	<u>Total PE</u>	<u>Avg. time per step [s]</u>	<u>Efficiency [%]</u>
<u>A</u>	<u>1664</u>	<u>64</u>	<u>1600</u>	<u>44.0</u>	<u>100</u>	<u>1600</u>	<u>14.9</u>	<u>100</u>
<u>B</u>	<u>3744</u>	<u>144</u>	<u>3600</u>	<u>19.9</u>	<u>98</u>	<u>3600</u>	<u>6.7</u>	<u>99</u>
<u>C</u>	<u>7488</u>	<u>288</u>	<u>7200</u>	<u>10.3</u>	<u>95</u>	<u>7200</u>	<u>3.6</u>	<u>92</u>
<u>D</u>	<u>8736</u>	<u>336</u>	<u>8400</u>	<u>9.3</u>	<u>90</u>	<u>8400</u>	<u>3.4</u>	<u>84</u>
<u>E</u>	<u>14976</u>	<u>576</u>	<u>14400</u>	<u>5.6</u>	<u>87</u>	<u>14400</u>	<u>2.3</u>	<u>74</u>

Several factors influence the computational performance of an LES code. Some factors depend on the hardware, for e.g.
5 the number of grid points per PE depends on the memory available per node. On the other hand, the communication time for data exchange between the PEs depend on the topology of the domain decomposition. The best performance in terms of communication time in a standalone run is achieved when the number of sub-domains in the x and y directions are equal. In that case the number of ghost points at the lateral boundaries are optimally minimized. In a nested simulation, the load per PE, i.e. the number of grid points per PE, in the two grids ~~vary~~varies. As the speed of the model integration depends on the PE
10 load, the load balancing between fine and coarse grid has an effect on the computational performance of the nested simulation. Keeping these factors in mind, we designed ~~a new nested simulation domain to measure~~ the nested simulation domains listed in Table 4 for the purpose of assessing the computational performance, as the total number of processors is varied. ~~The number of grid points in the CG is around 2×10^8 and in the FG it is around 6.3×10^9 .~~ To avoid load balancing bias in the scalability analysis, the ratio between the number of PEs for CG and FG is kept constant ~~to avoid load balancing bias in the scaling~~in all
15 the five runs listed in Table 5. Keeping the processor ratio constant implies that the ratio between the number of grid points per PE in CG and FG ~~are~~is also held constant. Consequently, in this performance test, the FG has 1.25 times more grid points per

PE than the CG in all the processor configurations tested. To compare the performance of nested model against the non-nested version of PALM under equivalent work load, a grid with the same dimensions of the FG is set-up. This non-nested grid also has the same load per PE and same number of cores as the FG. Such a non-nested set-up is acceptable for comparison since the number of PE in CG is negligible compared to the PE in FG in our set-up (e.g. 14400 PE in FG and only 576 PE in CG).

- 5 A pure standalone simulation with FG resolution throughout the boundary layer was not performed as it would need about 2.5×10^{10} grid points and such a large domain was computationally not feasible.

The performance is measured in terms of the time taken to simulate one time step. To increase the accuracy of this performance measurement, the simulation is integrated for ten time steps and the average of the time per step is plotted. The results presented in Fig. 10 shows close to linear scaling for up to 14976 PE in both nested and standalone runs. The difference in time per step between the nested and standalone runs can be interpreted as the additional computational time needed by the nesting algorithm. A jump in the time taken to compute one step is observed when more than 8192 PEs are used. This is a hardware dependent increase in communication time as the nodes are grouped as 'islands' on SuperMUC system at the Leibniz Supercomputing Centre. The communication within the nodes of the same island is faster than the communication across multiple islands. The strong scaling efficiency in Table 4 is calculated keeping the run with lowest number of PEs as the reference. As the number of grid points per PE is reduced from run A to E as shown in Table 5, the nested runs shows slightly better efficiency than the non-nested runs. The average time per step of the nested grid is 3 times higher than the non-nested set-up for run A but the factor decreases to about 2.5 for run E. This improvement is possibly due to reduction in waiting time between the FG and CG as the number of grid points per PE decreases.

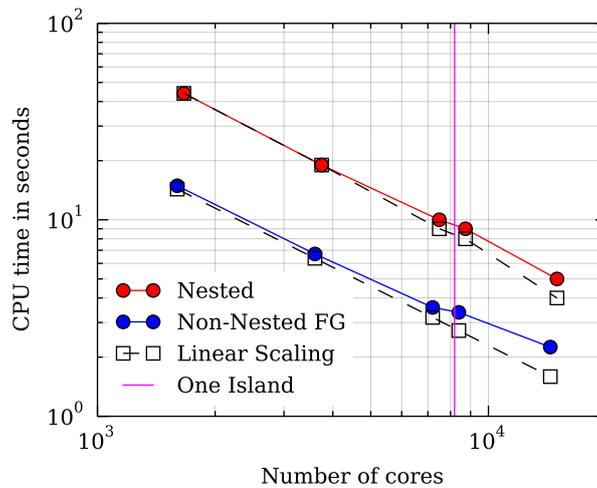


Figure 10. The nested simulations show close to linear scalability. A Non-Nested domain with same number of grid points as the FG is plotted to benchmark the scalability of the standard version of PALM on the same machine. The difference between the blue and the red line is approximately equal to the additional computational time needed by the nesting routines. The simulations were performed on SuperMUC at the Leibniz Supercomputing Center. Each node has 32 GB of main memory and two Sandy-Bridge processors with 2.7 GHz, each processor has 8 cores (Anastopoulos et al., 2013).

3.4 Practical Considerations

In this paragraph we summarize some guidelines for using this nesting approach. In PALM, the user has the choice to select between Wicker-Skamarock (Wicker and Skamarock, 2002) and Piacsek-Williams (Piacsek and Williams, 1970) for the advection scheme. Similarly, for solving the Poisson equation for the pressure, the user can choose between the FFT or Multi-Grid based solver. During the development and the validation of the two-way nesting, only the Wicker-Skamarock advection scheme and FFT based pressure solvers were tested. The two-way nesting supports only periodic boundary conditions in the horizontal for both CG and FG, and therefore an FFT based pressure solver is an appropriate choice. However, to be able to use Multi-Grid solvers, for e.g. in non-periodic horizontal boundary conditions, modifications to the two-way nesting algorithm will be needed. The large scale forcing feature in PALM is found to be compatible with the nesting algorithm without further modifications. Other features like canopy parameterization, radiation model, land surface models etc. have not been tested.

Our implementation of the vertical nesting allows only integer nesting ratio-ratios in all directions. ~~We recommend odd nesting ratio, in the range of 3 and 9, as the accuracy of the simulations decrease with the increase in nesting ratio.~~ The height of the nested domain has a direct influence on the accuracy of the two-way nesting algorithm. Based on our trials (not shown) we recommend that the overlapping-FG covers at least 12 grid levels of the CG. ~~In an LES, the first five grid points are unreliable and this condition extends also to the FG.~~ For better computational performance ~~it is recommended~~ we recommend that the number of grid points per PE in the CG is kept at only 40 to 80 percent of the FG value. The reduced work load of the CG ~~minimizes~~ is expected to minimize the waiting time of the FG during the concurrent time advancement by the quicker CG pressure solver step. However, the actual improvement in performance will depend on the memory available, processor speed and the inter-node communication architecture of the computing cluster and the optimal load balancing can only be found through trials. Furthermore, the choice of the domain size is often restricted by the topology of the processor decomposition. In a 2D decomposition, the number of grid points along the x-direction should be an integer multiple of the number of PE along x and similarly for y-direction. This condition has to be individually satisfied for the CG and the FG.

Though our nesting technique ~~computationally makes feasible~~ makes resolving the surface layer resolution down to 0.5 m for a moderately large domain computationally feasible, care should be taken to ensure the validity of such LES. In PALM, the height of the first grid point should be at the least twice greater than the local surface-roughness parameter. This technical restriction is common to all models that employ MOST and ensures the proper evaluation of the logarithm needed in the calculation of u^* . Furthermore, Basu and Lacser (2017) recently recommended that MOST boundary-conditions should be adapted for very high-resolution LES where the first grid point is smaller than 2-5 times the height of the roughness elements.

4 Summary

We presented a two-way grid nesting technique that enables high resolution LES of the surface layer. In our concurrently parallel algorithm, the two grids with different resolution overlap in the region close to the surface. The grids are coupled ~~at every sub-step of the Runge-Kutta time integration. The~~ , i.e the interpolation of the boundary conditions and the feedback to the parent grid are performed ~~by energyconserving methods,~~ at every sub-step of the Runge-Kutta time integration. The anterpolation

of the TKE involves the Germano identity to ensure the conservation of total kinetic energy. The exchange of data between the two grids is achieved by MPI communication routines and the ~~function calls are optimized for performance~~ communication is optimized by derived datatypes. Results of the convective boundary layer simulation show that grid nesting improves the vertical profiles of variance and the fluxes in the surface layer. In particular, the profiles of the vertical temperature flux are improved. ~~Due to the two-way nesting, there is also an improvement in the coarse grid in the profile of potential temperature variance.~~ The current vertical nesting only works with periodic boundary conditions and with the same horizontal extent in both the domains. The nested simulation needs ~~7-4~~ times less computational time than a full high resolution simulation for comparable accuracy in the surface layer. The scalability of the algorithm on up to 14976 CPUs is demonstrated.

5 Code availability

10 The PALM code is distributed under the GNU General Public License. The code (revision 2712) is available at <https://palm.muk.uni-hannover.de/trac/browser/palm?rev=2712>.

Author contributions. SH was the main developer of the model code, with FDR as side developer, SR supporting the code development and MM, SR and FDR supervising the development. The experiment was designed by SH, FDR, SR and MM and carried out by SH, who also performed the validation. Visualization was done by SH, and the original draft written by SH and FDR, with review and editing by SR and
15 MM. Funding acquisition and administration by MM.

Acknowledgements. This work was conducted within the Helmholtz Young Investigators Group “Capturing all relevant scales of biosphere-atmosphere exchange – the enigmatic energy balance closure problem”, which is funded by the Helmholtz-Association through the President’s Initiative and Networking Fund, and by KIT. Computer resources for this project have been provided by the Leibniz Supercomputing Centre under grant: pr481a. We thank Gerald Steinfeld for sharing his original notes and code of a preliminary nesting method in PALM.
20 We also thank Matthias Sühling and Farah Kanani-Sühling of the PALM group for their help in standardizing and porting the code, and we thank Michael Manhart for fruitful discussions.

References

- Anastopoulos, N., Nikunen, P., and Weinberg, V.: Best Practice Guide - SuperMUC v1.0. PRACE - Partnership for Advanced Computing in Europe 2013, <http://www.prace-ri.eu/best-practice-guide-supermuc-html>, 2013.
- Basu, S. and Lacser, A.: A Cautionary Note on the Use of Monin–Obukhov Similarity Theory in Very High-Resolution Large-Eddy Simulations, *Boundary-Layer Meteorology*, 163, 351–355, doi:10.1007/s10546-016-0225-y, 2017.
- Boersma, B. J., Kooper, M. N., Nieuwstadt, F. T. M., and Wesseling, P.: Local grid refinement in large-eddy simulations, *Journal of Engineering Mathematics*, 32, 161–175, doi:10.1023/A:1004283921077, 1997.
- Clark, T. L. and Farley, R. D.: Severe downslope windstorm calculations in two and three spatial dimensions using anelastic interactive grid nesting: A possible mechanism for gustiness, *J Atmos Sci*, 41, 329–350, doi:10.1175/1520-0469(1984)041<0329:SDWCIT>2.0.CO;2, 1984.
- Clark, T. L. and Hall, W. D.: Multi-domain simulations of the time dependent Navier Stokes equation: Benchmark error analyses of nesting procedures, *J Comput Phys*, 92, 456–481, doi:10.1016/0021-9991(91)90218-A, 1991.
- Daniels, M. H., Lundquist, K. A., Mirocha, J. D., Wiersema, D. J., and Chow, F. K.: A New Vertical Grid Nesting Capability in the Weather Research and Forecasting (WRF) Model, *Monthly Weather Review*, 144, 3725–3747, doi:10.1175/mwr-d-16-0049.1, 2016.
- Deardorff, J. W.: Three-dimensional numerical study of the height and the mean structure of a heated planetary boundary layer, *Bound-Layer Meteorol*, 7, 81–106, doi:10.1007/BF00224974, 1974.
- Deardorff, J. W.: Stratocumulus-capped mixed layers derived from a three-dimensional model, *Bound.-Lay. Meteorol.*, 18, 495–527, doi:10.1007/BF00119502, 1980.
- Debreu, L., Marchesiello, P., Penven, P., and Cambon, L.: Two-way nesting in split-explicit ocean models: Algorithms, implementation and validation, *Ocean Modelling*, 49-50, 1–21, doi:10.1016/j.ocemod.2012.03.003, 2012.
- Esau, I.: Indirect air–sea interactions simulated with a coupled turbulence-resolving model, *Ocean Dynamics*, 64, 689–705, doi:10.1007/s10236-014-0712-y, 2014.
- Flores, F., Garraud, R., and Muñoz, R. C.: CFD simulations of turbulent buoyant atmospheric flows over complex geometry: Solver development in OpenFOAM, *Computers & Fluids*, 82, 1–13, doi:10.1016/j.compfluid.2013.04.029, 2013.
- Germano, M., Piomelli, U., Moin, P., and Cabot, W. H.: A dynamic subgrid scale eddy viscosity model, *Phys. Fluid A*, 3, 7, doi:10.1063/1.857955, 1991.
- Harris, L. M. and Durran, D. R.: An Idealized Comparison of One-Way and Two-Way Grid Nesting, *Monthly Weather Review*, 138, 2174–2187, doi:10.1175/2010mwr3080.1, 2010.
- Kato, C., Kaiho, M., and Manabe, A.: An Overset Finite-Element Large-Eddy Simulation Method With Applications to Turbomachinery and Aeroacoustics, *Journal of Applied Mechanics*, 70, 32, doi:10.1115/1.1530637, 2003.
- Khanna, S. and Basseur, J. G.: Three-Dimensional Buoyancy- and Shear-Induced Local Structure of the Atmospheric Boundary Layer, *Journal of the Atmospheric Sciences*, 55, 710–743, doi:10.1175/1520-0469(1998)055<0710:tdbasi>2.0.co;2, 1998.
- Kravchenko, A., Moin, P., and Moser, R.: Zonal Embedded Grids for Numerical Simulations of Wall-Bounded Turbulent Flows, *Journal of Computational Physics*, 127, 412–423, doi:10.1006/jcph.1996.0184, 1996.
- Kröniger, K., De Roo, F., Brugger, P., Huq, S., Banerjee, T., Zinsser, J., Rotenberg, E., Yakir, D., Rohatyn, S., and Mauder, M.: Effect of Secondary Circulations on the Surface-Atmosphere Exchange of Energy at an Isolated Semi-arid Forest, doi:10.1007/s10546-018-0370-6, 2018.

- Kurihara, Y., Tripoli, G. J., and Bender, M. A.: Design of a Movable Nested-Mesh Primitive Equation Model, *Monthly Weather Review*, 107, 239–249, doi:10.1175/1520-0493(1979)107<0239:doammn>2.0.co;2, 1979.
- Manhart, M.: A zonal grid algorithm for DNS of turbulent boundary layers, *Computers and Fluids*, 33, 435–461, doi:10.1016/S0045-7930(03)00061-6, 2004.
- 5 Maronga, B., Gryschka, M., Heinze, R., Hoffmann, F., Kanani-Sühring, F., Keck, M., Ketelsen, K., Letzel, M. O., Sühring, M., and Raasch, S.: The Parallelized Large-Eddy Simulation Model (PALM) version 4.0 for atmospheric and oceanic flows: model formulation, recent developments, and future perspectives, *Geoscientific Model Development*, 8, 2515–2551, doi:10.5194/gmd-8-2515-2015, 2015.
- Moeng, C.-H. and Wyngaard, J. C.: Spectral analysis of large-eddy simulations of the convective boundary layer, *J. Atmos. Sci.*, 45, 3573–3587, doi:10.1175/1520-0469(1988)045<3573:SAOLES>2.0.CO;2, 1988.
- 10 Moeng, C.-H., Dudhia, J., Klemp, J., and Sullivan, P.: Examining Two-Way Grid Nesting for Large Eddy Simulation of the PBL Using the WRF Model, *Mon Wea Rev*, 135, 2295–2311, doi:10.1175/MWR3406.1, 2007.
- Nakahashi, K., Togashi, F., and Sharov, D.: Intergrid-Boundary Definition Method for Overset Unstructured Grid Approach, *AIAA Journal*, 38, 2077–2084, doi:10.2514/2.869, 2000.
- Patton, E. G., Sullivan, P. P., Shaw, R. H., Finnigan, J. J., and Weil, J. C.: Atmospheric Stability Influences on Coupled Boundary Layer and
- 15 Canopy Turbulence, *Journal of the Atmospheric Sciences*, 73, 1621–1647, doi:10.1175/jas-d-15-0068.1, 2016.
- Piasek, S. A. and Williams, G. P.: Conservation properties of convection difference schemes, *J. Comput. Phys.*, 198, 500–616, doi:10.1016/0021-9991(70)90038-0, 1970.
- Raasch, S. and Schröter, M.: PALM - A large-eddy simulation model performing on massively parallel computers, *Meteorologische Zeitschrift*, 10, 363–372, doi:10.1127/0941-2948/2001/0010-0363, 2001.
- 20 Reynolds, W. C.: The potential and limitations of direct and large eddy simulations, in: *Whither Turbulence? Turbulence at the Crossroads*, edited by Lumley, J. L., pp. 313–343, Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/3-540-52535-1_52, 1990.
- Saiki, E. M., Moeng, C.-H., and Sullivan, P. P.: Large-eddy simulation of the stably stratified planetary boundary layer, *Bound.-Lay. Meteorol.*, 95, 1–30, doi:10.1023/A:1002428223156, 2000.
- Schmidt, H. and Schumann, U.: Coherent structure of the convective boundary layer derived from large-eddy simulations, *J Fluid Mech*, 200,
- 25 511–562, doi:10.1017/S0022112089000753, 1989.
- Skamarock, W., Klemp, J., Dudhia, J., Gill, D., Barker, D., Wang, W., Huang, X.-Y., and Duda, M.: A Description of the Advanced Research WRF Version 3, doi:10.5065/d68s4mvh, 2008.
- Sullivan, P. P., McWilliams, J. C., and Moeng, C.-H.: A grid nesting method for large-eddy simulation of planetary boundary layer flows, *Bound-Layer Meteorol*, 80, 167–202, doi:10.1007/BF00119016, 1996.
- 30 van Hooft, J. A., Popinet, S., van Heerwaarden, C. C., van der Linden, S. J. A., de Roode, S. R., and van de Wiel, B. J. H.: Towards Adaptive Grids for Atmospheric Boundary-Layer Simulations, *Boundary-Layer Meteorology*, 167, 421–443, doi:10.1007/s10546-018-0335-9, 2018.
- Wang, G., Duchaine, F., Papadogiannis, D., Duran, I., Moreau, S., and Gicquel, L. Y.: An overset grid method for large eddy simulation of turbomachinery stages, *Journal of Computational Physics*, 274, 333–355, doi:10.1016/j.jcp.2014.06.006, 2014.
- 35 Wicker, L. J. and Skamarock, W. C.: Time-splitting methods for elastic models using forward time schemes, *Mon. Wea. Rev.*, 130, 2008–2097, doi:10.1175/1520-0493(2002)130<2088:TSMFEM>2.0.CO;2, 2002.
- Williamson, J. H.: Low-storage Runge-Kutta schemes, *J. Comput. Phys.*, 35, 48–56, doi:10.1016/0021-9991(80)90033-9, 1980.

Zhou, B., Xue, M., and Zhu, K.: A Grid-Refinement-Based Approach for Modeling the Convective Boundary Layer in the Gray Zone: Algorithm Implementation and Testing, *Journal of the Atmospheric Sciences*, 75, 1143–1161, doi:10.1175/jas-d-17-0346.1, 2018.