SPECIAL PROJECT PROGRESS REPORT

Reporting year	2019
Project Title:	Towards Cloud-Resolving Climate Simulations
Computer Project Account:	spnlcrom
Principal Investigator(s):	Daan Crommelin, Pier Siebesma, Gijs van den Oord, Fredrik Jansson, Maria Chertova
Affiliation:	Centrum Wiskunde & Informatica (Crommelin, Jansson), KNMI and TU Delft (Siebesma), Netherlands eScience Center (van den Oord, Chertova)
Name of ECMWF scientist(s) collaborating to the project (if applicable)	Glenn Carver has helped us with running OpenIFS and by providing initial states for the simulations
Start date of the project:	1/1/2017
Expected end date:	31/12/2019

Computer resources allocated/used for the current year and the previous one
(if applicable)
Please answer for all project resources

		Previous year		**Current year**	
		Allocated	Used	Allocated	Used
High Performance Computing Facility	(units)	15M	11.2M	15M	8.3M
Data storage capacity	(Gbytes)	100000		100000	18T/$SCRATCH, 9.3T/tape

Summary of project objectives (10 lines max)

The overarching goal of this project is to come to a better understanding of cloud-climate feedbacks, leading to reduced uncertainty in climate sensitivity estimates. To achieve this, we pursue a computational strategy of developing 3-dimensional superparameterization (3dSP) by embedding 3-d convection-resolving Large Eddy Simulation (LES) models in each grid column of a global model (OpenIFS). The LES models are embedded as a two-way nesting (or two-way coupling): the global model column state drives the LES model, and the LES feeds back to the global model. The nested LES models replace traditional convection parameterization schemes in the global model columns. We work with DALES, the Dutch Large Eddy Simulation model, as the convection-resolving LES. The computer resources of this special project are intended for performing simulations with the coupled (OpenIFS-DALES) 3dSP model for test cases including a cold air outbreak case (previously subject of the WGNE Grey Zone project).

Summary of problems encountered (10 lines max)

In 2017 we found out that AMUSE does not work well with the Cray MPI which is installed on the ECMWF Cray, the reason being that when AMUSE spawns worker processes it launches them using MPI_Comm_spawn(), which the Cray MPI does not support. We solved this problem with a work-around where all workers are launched at the start of the simulation in a regular MPI job, after which the appropriate MPI communicators are created. This works for us, since we know ahead of time how many workers are needed for a particular simulation. Supposedly new versions of the Cray MPI will include MPI_Comm_spawn(). We are still using the work-around with pre-launched worker codes.

Summary of plans for the continuation of the project (10 lines max)

- Analyze and improve the performance of the coupled model.
- The Python interface to the DALES (LES-based, local, cloud-resolving model) will be presented as a software paper, and is going to be used in teaching, for interactive control of the DALES model.
- Beyond the current project, the superparameterized OpenIFS-model will be used as a one example application in a project on multiscale modelling (VECMA).
- Investigate how the model coupling scheme can be improved, in particular to account for variability on scales smaller than the global model's grid size, which affects advection of clouds from one superparameterized grid point to another.

List of publications/reports from the project with complete references

F. Jansson, G. van den Oord, I. Pelupessy, J. H. Grönqvist, A. P. Siebesma, D. Crommelin (2019) *Regional superparameterization in a Global Circulation Model using Large Eddy Simulations*, submitted.

Conference papers:

Pelupessy I. et al. (2019) *Creating a Reusable Cross-Disciplinary Multi-scale and Multi-physics Framework: From AMUSE to OMUSE and Beyond*. In: Rodrigues J. et al. (eds) Computational Science – ICCS 2019. Lecture Notes in Computer Science, vol 11539. Springer. **DOI:** 10.1007/978-3-030-22747-0_29

Conference abstracts:

Fredrik Jansson, Gijs van den Oord, Inti Pelupessy, Maria Chertova, Pier Siebesma, and Daan Crommelin (2019) [On the regional superparametrization of OpenIFS by 3D LES models](), Geophysical Research Abstracts, Vol. 21, EGU2019-11303

Jansson, F.R, van den Oord, G, Siebesma, A.P, & Crommelin, D.T. (2018). *Resolving clouds in a global atmosphere model - a multiscale approach with nested models*. In Proceedings - IEEE 14th International Conference on eScience, e-Science 2018. **DOI:**10.1109/eScience.2018.00043

Daan Crommelin , Pier Siebesma, Fredrik Jansson, Gijs van den Oord, Inti Pelupessy, Johanna Gronqvist, Maria Chertova (2019). *Regional Superparameterization with LES*. In Mathematisches Forschungsinstitut Oberwolfach Report No. 7/2019. DOI: 10.4171/OWR/2019/7

Fredrik Jansson, Gijs van den Oord, Inti Pelupessy, Maria Chertova, Johanna Grönqvist, Daan Crommelin, Pier Siebesma (2019). *High-resolution regional superparameterization of OpenIFS with DALES*. In UCP2019 - Understanding Clouds and Precipitation/ Book of Abstracts.

Summary of results

Progress July 2018 – June 2019

A comprehensive article describing the superparameterization setup and initial results has been submitted to JAMES and is currently under revision. The submitted preprint is attached to this report. We quote its abstract below:

"As an explorative step towards global Large Eddy Simulations, we investigate using comprehensive three dimensional Large Eddy Simulations as a superparameterization that can replace all traditional parameterizations of atmospheric processes that are currently used in global models. We present the technical design for a replacement of the parameterization for clouds, convection, and turbulence of the global atmospheric model of the European Centre for Medium-Range Weather Forecasts (ECMWF) by the Dutch Atmospheric Large Eddy Simulation (DALES) model. The model coupling consists of bidirectional data exchange between the global model and the high-resolution Large Eddy Simulation (LES) models embedded within the columns of the global model. Our setup allows for selective superparameterization, i.e. for applying superparameterization in local regions selected by the user, whilst keeping the standard parameterization of the global model intact outside this region. Our design allows the LES instances to run concurrently. First simulation results, employing this design, demonstrate the potential of our approach". [from Jansson et al., 2019]

The code for the coupling of OpenIFS and DALES is made available on GitHub, details can be found in the article. Also, the separate Python interfaces to OpenIFS and to DALES have been included in the official OMUSE repository.

Furthermore, many bug fixes and improvements for DALES developed in this project have been included in the newly released DALES 4.2.

Regional superparameterization in a Global Circulation Model using Large Eddy Simulations

Fredrik Jansson[1], Gijs van den Oord[2], Inti Pelupessy[2], Johanna H. Grönqvist[3], A. Pier Siebesma[4,5], Daan Crommelin[1,6]

[1]Centrum Wiskunde & Informatica, Amsterdam
[2]Netherlands eScience Center, Amsterdam
[3]Åbo Akademi University, Turku, Finland
[4]Royal Netherlands Meteorological Institute, de Bilt
[5]Delft University of Technology, Delft
[6]Korteweg-de Vries Institute, University of Amsterdam, Amsterdam

Key Points:

- Efficient Implementation of a Large Eddy Simulation (LES) based superparameterization into a global circulation model.
- Flexibility on the position and the size of the superparameterized area.
- Promising evaluation of the first results of an LES-based superparameterized simulation.

Corresponding author: Fredrik Jansson, `jansson@cwi.nl`

Abstract

As an explorative step towards global Large Eddy Simulations, we investigate using comprehensive three dimensional Large Eddy Simulations as a superparameterization that can replace all traditional parameterizations of atmospheric processes that are currently used in global models. We present the technical design for a replacement of the parameterization for clouds, convection, and turbulence of the global atmospheric model of the European Centre for Medium-Range Weather Forecasts (ECMWF) by the Dutch Atmospheric Large Eddy Simulation (DALES) model. The model coupling consists of bidirectional data exchange between the global model and the high-resolution Large Eddy Simulation (LES) models embedded within the columns of the global model. Our setup allows for selective superparameterization, i.e. for applying superparameterization in local regions selected by the user, whilst keeping the standard parameterization of the global model intact outside this region. Our design allows the LES instances to run concurrently. First simulation results, employing this design, demonstrate the potential of our approach.

1 Introduction

An accurate representation of clouds and convection in global weather and climate models and their interaction with the large-scale circulation remains one of the main challenges in atmospheric modeling. Uncertainties in the representation of clouds and convection are the prime sources of uncertainty in climate model sensitivity, and major contributors to longstanding biases in the representation of the precipitation patterns in current climate and their projections in future climate (Bony et al., 2015; Schneider, Teixeira, et al., 2017).

Cloud related processes occur over a wide range of scales ranging from cloud droplet formation at the micrometer scale to cloud convective updrafts and downdrafts that can be as large as 10 km, from which organized mesoscale systems can emerge extending over hundreds of kilometers. Current operational global models operate at numerical resolutions in the range of 10-100 km. As a consequence, cloud and convective processes are numerically not resolved and their impact on the resolved state is instead approximated by parameterizations, causing uncertainties of these unresolved processes.

The problem of parameterized clouds and convection is largely avoided when using Large Eddy Simulations (LES). The paradigm of LES is based on the idea that small unresolved turbulent eddies can be faithfully parameterized in terms of the resolved large eddies by making use of the self-similar structure of turbulence in the inertial subrange. The atmospheric inertial subrange is bounded by the depth of the atmospheric boundary layer which has a typical depth of 1 kilometer, indicating that a minimum resolution of O(100 m) is required to numerically resolve the relevant turbulence, convection, and cloud dynamics.

Two obvious but challenging pathways for improving the representation of clouds and convection in global models are either increasing the resolution of existing global models to turbulence-resolving scales, or extending the spatial domains of Large Eddy Simulations until a global scale is reached. Regarding the latter approach, realistic multiday large eddy simulations have been reported on domains approaching 1000 km^2 (Schalkwijk et al., 2015; Heinze et al., 2017). Global multiday simulations are approaching the 1 kilometer resolution scale and are capable of partly resolving the cloud dynamics and therefore usually referred to as global cloud resolving models (CRMs) (Miyamoto et al., 2013; Bretherton & Khairoutdinov, 2015). Global CRMs form a interesting playground to explore the interaction between the global circulation and the resolved moist convective systems, but one should also bear in mind that for resolutions used in global CRMs (1-5km), atmospheric turbulence and boundary layer clouds remain essentially unresolved.

A different pathway is offered by superparameterization (SP) (Grabowski & Smolarkiewicz, 1999; Grabowski, 2001, 2004; Khairoutdinov & Randall, 2001; Khairoutdinov et al., 2005), where existing global and local cloud resolving models are coupled. The common set-up for SP is to replace deep convection and cloud parameterization schemes in every model column of the global model by a CRM. Because of computational constraints, the CRMs used in SP are mostly two-dimensional (2D). A 3D CRM was used by Khairoutdinov et al. (2005), but the grid of the CRM was still very coarse and limited to 8×8 columns. Jung and Arakawa (2010) present a quasi-3D SP where the global models grid points are connected by narrow corridors consisting of 3D local models. An overview of other simulation studies with SP is provided by Tao and Chern (2017). Like global CRMs however, these SP approaches traditionally use horizontal resolution of 1-4 km and coarse vertical resolutions, and still require additional parameterizations for boundary layer clouds and turbulence.

Recently, a variation on superparameterization has been proposed (Grabowski, 2016), where the resolution of local CRMs becomes fine enough to be turbulence resolving. Parishani, Pritchard, Bretherton, Wyant, and Khairoutdinov (2017) included LES models with a fine spatial horizontal resolution (250m) and 125 vertical levels in their global model. However, to be able to run the global model with SP, the LESs used small domains (8×8 columns, i.e. 2 km $\times$ 2 km) covering only a small fraction of the domain of a single column of the global model.

Ideally, SP is carried out with a 3D high-resolution CRM that covers the full domain of each global model column. To reduce the enormous computational cost of this (hypothetical) SP set-up, in the studies mentioned above either the 3D CRM is simplified to 2D or quasi-3D, or the grid of each CRM is kept small (e.g. 8×8 horizontal). In the latter case, one can choose between high resolution on a small CRM domain (Parishani et al., 2017), or coarser resolution on a larger CRM domain (Khairoutdinov et al., 2005).

The SP approach that we will present in this study is different. Our aim is to use turbulence resolving resolutions on sufficiently large 3D-domains as a SP, in accordance with the resolution of the large-scale model. Computationally, this approach obviously does not allow SP to be applied globally. Therefore, rather than reducing the cost of the 3D CRM as sketched above, our set-up creates the possibility to use SP only in a selected region, while leaving the regular (non-SP) parameterization in use outside this region. The motivation for this approach is simply that using 3D LES as a SP provides the best benchmark for conventional parameterizations.

The benefits of such a 3D LES-based SP in a conventional hydrostatic global model over a global LES or CRM have been discussed in Grabowski (2016). Computationally, it is attractive since all the local models can run independent from each other and only have to exchange mean profiles with the large-scale model. This allows an efficient implementation on massively parallel computer systems since all the SP models can run independently on separate cores. Further acceleration can be achieved by running the SP models sparser in space and time (Xing et al., 2009) or by varying the vertical resolution of the SP models depending on cloud types that they need to resolve (Marchand & Ackerman, 2011). Conceptually there is also the advantage that the large-scale model can be formulated efficiently in a hydrostatic manner while the smaller scale LES based SP models can be conveniently expressed in an anelastic formulation. This way there is no need to find an appropriate soundproof compressible formulation of the dynamics on a global scale.

The drawback of any SP formulation is that it introduces a scale break at the resolution scale of the large-scale model. This hinders the spectral transfer of variability across this scale and will potentially influence mesoscale organization. However, it also offers an excellent opportunity to explore the effect of such a scale break which is present in every operational large-scale model.

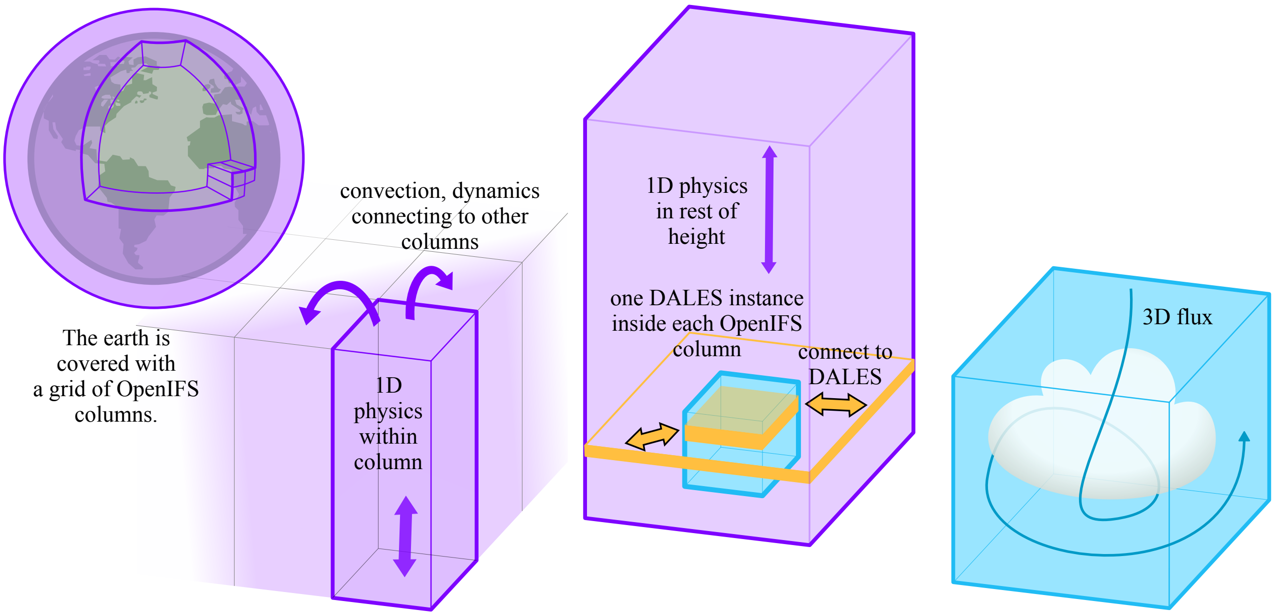

Figure 1. Overview of the superparameterized model. Some grid columns of the global model OpenIFS (purple), are selected for superparameterization. Each of them is coupled to a local DALES model (blue), which resolves clouds and convection in three dimension. The tendencies generated by these processes are fed back to the global model.

In this paper we will discuss the implementation and performance of the Dutch Atmospheric Large Eddy Simulation (DALES) model as a regional 3D LES based superparameterization into the Open Integrated Forecast System (OpenIFS) developed at the European Centre for Medium-Range Weather Forecasts (ECMWF) (Carver & Vana, 2017). Section 2 describes the complete methodology of the coupling while Section 3 concentrates more on the technical implementation. Section 4 presents results of a superparameterized atmospheric simulations over the Netherlands, comparisons with observations and an evaluation of the numerical performance. Section 5 contains conclusions, discussion, and outlook.

2 Methods

For coupling the global and cloud-resolving models, we follow the approach presented by Grabowski (2004) and also (Khairoutdinov & Randall, 2001; Khairoutdinov et al., 2005). In the grid columns of the global model which are selected for superparameterization, a local, Large Eddy Simulation (LES) model is embedded as shown in Fig. 1. The general idea is that for each coupled quantity, a forcing is introduced, which keeps the states of the two models consistent with each other. The coupling is bi-directional, so that the effects of clouds, turbulence and convection which are resolved in the local model are fed back to the global model.

Below, we summarize the coupling procedure, first in a simplified case where the two models are assumed to have similar vertical grid levels and to be formulated in terms of the same quantities. We then discuss the adaptations needed to couple models with different vertical levels and different physical quantities, as is the case for our set-up with coupling between DALES and OpenIFS.

2.1 Physical coupling of the models

We consider a 3D small-scale model embedded in a single column of the large-scale model. In earlier versions of the superparameterization scheme (Grabowski & Smolarkiewicz, 1999; Grabowski, 2001), the states of the large-scale and the small-scale models are relaxed towards each other, with a freely chosen time constant, which was taken as one hour. In the later scheme (Grabowski, 2004), the relaxation time constant is chosen equal to the time step of the large-scale model, making the models more tightly coupled. The aim is that for any coupled quantity Q, the horizontal slab average in the small-scale model at height z matches the value at the same height in the large-scale model:

$$Q(Z = z, t) = \langle q(x, y, z, t)\rangle. \tag{1}$$

Capital letters denote quantities in the large-scale model, lower-case letters denote the small-scale model. Q and q here may represent any of the prognostic variables, and the brackets $\langle . \rangle$ denote a horizontal slab average over the domain of the local model. In the appendix we analyze to what extent the desired equality (1) can be achieved with the scheme from (Grabowski, 2004).

The coupled variables generally include the horizontal wind velocities, the temperature, and the specific humidity. As in earlier superparameterization works, the vertical velocities are left uncoupled. Since each local model is a closed system due to periodic boundary conditions, the horizontal average of the vertical velocity is zero.

The models are coupled by introducing additional forcings in both models, i.e. additional contributions to the time derivatives of the coupled quantities. F_Q represents a forcing that stems from q and acts on Q in the large-scale model, while f_q represents a forcing stemming from Q and acting on q in the small-scale model.

The time-stepping procedure is as follows. The large-scale model performs a single time step from time T to $T+\Delta T$, then the small-scale model is evolved over the same time interval, in multiple steps of length Δt. Before the time evolution of each model, forcings are calculated based on the difference between the most recently obtained states of the two models.

(i) Given the state of both models at time T, represented by $Q(T)$ and $q(T)$, calculate forcings on the large-scale model

$$F_Q(T) = \frac{\langle q(T)\rangle - Q(T)}{\Delta T}. \tag{2}$$

(ii) Time-step the large-scale model

$$Q(T + \Delta T) = Q(T) + \Delta T\big[A_Q(T) + S_Q(T) + F_Q(T)\big], \tag{3}$$

where $A_Q(T)$ represents advection terms and $S_Q(T)$ represents source terms during the step from T to $T + \Delta T$.

(iii) Calculate the forcing on the small-scale model

$$f_q(T) = \frac{Q(T + \Delta T) - \langle q(T)\rangle}{\Delta T}. \tag{4}$$

(iv) Time-step the small-scale model

$$q(T + \Delta T) = q(T) + \sum_{t=T}^{T+\Delta T} \Delta t\big[a_q(t) + s_q(t) + f_q(T)\big]. \tag{5}$$

The sums over t here schematically represent evolving the small-scale model over several time steps, with $a_q(t)$ denoting advection terms and $s_q(t)$ denoting source terms in the small-scale model.

This choice of forcings is such that they couple the advection and source terms between the models, see also Grabowski (2004). In particular, one can show that

$$F_Q(T+\Delta T) = \frac{1}{\Delta T} \sum_{t=T}^{T+\Delta T} \Delta t \langle a_q(t) + s_q(t) \rangle, \tag{6}$$

and

$$f_q(T) = A_Q(T) + S_Q(T), \tag{7}$$

so that the forcings on the small-scale model equals the advective and source tendencies in the large-scale model and vice versa. Thus, each physical process should be taken into account in one of the models, but not in both. Otherwise the contribution will be counted twice. As is shown in the appendix, the equality (1) is satisfied exactly if all physical processes are accounted for in one model and none in the other: if $\langle a_q(t) + s_q(t) \rangle = 0$ for all t then $Q(T) = \langle q(T) \rangle$, whereas if $A_Q(T) + S_Q(T) = 0$ then $Q(T+\Delta T) = \langle q(t) \rangle$.

The Grabowski scheme does not explicitly describe the superparameterization procedure for the sequential-splitting method in the global model, which are used in the column physics routines in OpenIFS. In this algorithm the physics processes are ordered by decreasing time scales and every tendency is calculated with updated fields as its input, so that the tendencies of slower processes contribute to the evaluation of tendencies due to faster processes. We preserve this procedure by inserting the coupling to the local models at the stage in the OpenIFS time step where the parameterizations we substitute are evaluated, namely turbulence, convection and cloud schemes as shown in Fig. 2.

2.2 Interpolation and change of variables

The coupling scheme outlined so far is the standard superparameterization scheme as described in the references - where the vertical grids are assumed to be the same in the two models, and the models are formulated in the same variables. In our case of coupling OpenIFS and DALES, neither of these assumptions can be made, requiring a few extra steps in the model coupling. The two models may use different vertical grids, typically the local model has a denser grid than the global model, and will not extend beyond the tropopause. OpenIFS is formulated in so-called hybrid sigma pressure coordinates; in our test cases we used 90 vertical levels extending up to roughly 80 km. For DALES we typically used a vertical spacing of 25 m, extending up to 4 km. To exchange vertical profiles of quantities between global and the local models we use linear interpolation along the z-axis. We convert the OpenIFS hybrid model level profiles to altitude by fetching the full- and half-level pressure profiles P_f, P_h and using the hydrostatic approximation to determine the height of each layer:

$$dZ = \frac{-R_d T \big[1 + (R_v/R_d - 1)Q_V - (Q_L + Q_I)\big]}{g_0 P_f} dP. \tag{8}$$

Here, $R_d \approx 287.04$ J/kg K is the gas constant for dry air, $R_v \approx 461.5$ J/kg K is the gas constant for water vapor, and g_0 is the acceleration due to gravity. Q_V is the water vapor specific humidity, while Q_L and Q_I are the specific humidities of liquid water and ice. Above the vertical extent of the DALES models we set the forcings F_Q on the global model to zero.

Furthermore, the two models are formulated using different prognostic variables. The coupling thus requires a variable conversion step. Table 1 lists the quantities that are coupled between the two models, the conversions required are described in detail below.

OpenIFS quantities

symbol	internal array name	description	unit	coupling direction
U, V	`GMV`	horizontal velocity	m/s	bidirectional
T	`GMV`	temperature	K	bidirectional
Q_V	`GFL`	water vapor specific humidity	kg/kg	bidirectional*
Q_L	`GFL`	liquid water specific humidity	kg/kg	bidirectional*
Q_I	`GFL`	ice water specific humidity	kg/kg	bidirectional*
Z_{0M}, Z_{0H}	`ZAZOM`, `ZAZOH`	surface roughness	m	output
$\mathcal{F}_{SH}$	`PDIFTQ`	specific humidity flux	kg/m^{2}s	output
$\mathcal{F}_{QL}$	`PDIFTL`	liquid water specific humidity flux	kg/m^{2}s	output
$\mathcal{F}_{QI}$	`PDIFTI`	ice specific humidity flux	kg/m^{2}s	output
$\mathcal{F}_{TS}$	`PDIFTS`	sensible heat flux	W/m^2	output
P_S	`GMVS`	surface pressure	Pa	output
A	`GFL`	cloud fraction	-	input

DALES quantities

symbol	variable name	description	unit	coupling direction
u, v	`u0, v0`	horizontal velocity	m/s	bidirectional
θ_l	`thl0`	liquid water potential temperature	K	bidirectional
q_t	`qt0`	specific humidity	kg/kg	input
q_v		water vapor specific humidity	kg/kg	output
q_L		liquid water specific humidity	kg/kg	output
q_I		ice water specific humidity	kg/kg	output
$\mathcal{F}_q$	`wqsurf`	specific humidity flux	kg/kg m/s	input
$\mathcal{F}_{\theta_l}$	`wtsurf`	surface heat flux	K m/s	input
p_S		surface pressure	Pa	input
A		cloud fraction	-	output

Table 1. Quantities in OpenIFS and DALES used in the superparameterization scheme. *See text for the details of the humidity coupling.

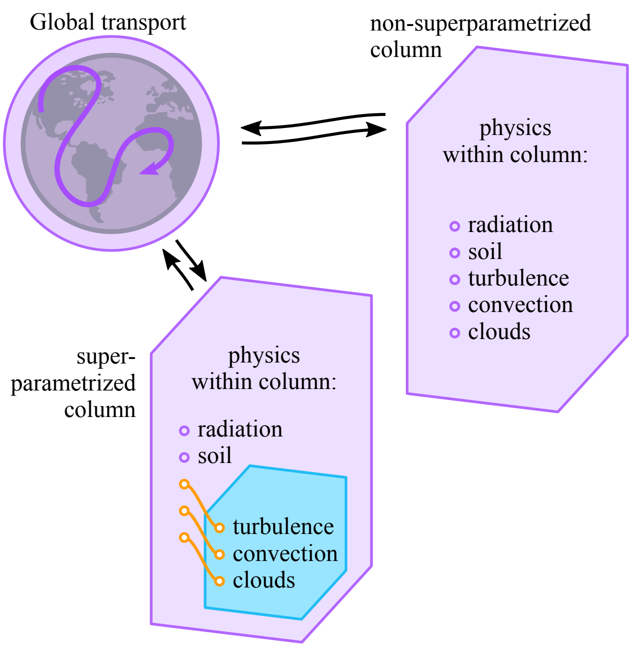

Figure 2. In OpenIFS the column physics schemes for turbulence, convection and clouds are turned off in the superparameterized columns. These processes are instead handled by DALES.

For temperature, OpenIFS uses a regular temperature T, while DALES uses the liquid water potential temperature θ_l (Heus et al., 2010). For the conversion, we use

$$\theta_l \approx \frac{T}{\Pi} - \frac{L}{c_{pd}\Pi} q_c, \tag{9}$$

where q_c is the cloud condensate defined as the sum of cloud liquid water q_L and cloud ice q_I. The Exner function Π is defined as

$$\Pi(p) = \left(\frac{p}{p_0}\right)^{R_d/c_{pd}}. \tag{10}$$

Here $c_{pd} \approx 1004$ J/kg K is the specific heat of dry air at constant pressure.

For the humidity, DALES uses only a total humidity q_t as a prognostic variable. At every time step, q_t is partitioned into vapor, liquid and ice according to the local temperature. OpenIFS on the other hand, has separate prognostic variables for these quantities, Q_V for water vapor, Q_L for liquid water and Q_I for ice. When forcing DALES, the total humidity q_t is nudged towards the total humidity of the global model, calculated as $Q_T = Q_V+Q_L+Q_I$. When coupling the humidities back to the global model, the diagnosed values of the DALES are used, so that each one of Q_V, Q_L, and Q_I is forced towards the horizontal average of the corresponding diagnosed quantity in DALES.

2.3 Surface scheme

Both DALES and OpenIFS contain a surface model which accounts for surface drag and for fluxes of heat and moisture between the atmosphere and the land or sea surface. We have chosen to use fluxes and surface roughness lengths calculated in OpenIFS, while letting Dales handle the effects these have on the atmosphere at the superparameterized grid points. In this way, we can rely on the land/sea mask, soil and vegetation data, and ocean wave model of OpenIFS, making it easy to set up a superparameterization anywhere without having to supply detailed surface information.

We achieve this by having DALES run with prescribed roughness lengths and surface fluxes of moisture and heat. These quantities are retrieved from the OpenIFS at every time step. To avoid double counting of the surface fluxes, we disable the contribution of the surface layer scheme in OpenIFS at the superparameterized grid points.

Since OpenIFS and DALES are built using quantities with different units (see table 1) unit conversions are necessary to consistently couple the surface fluxes. For the humidity flux, a scaling with the air density ρ is required:

$$\mathcal{F}_q = -\frac{\mathcal{F}_{QL} + \mathcal{F}_{QI} + \mathcal{F}_{SH}}{\rho}. \tag{11}$$

In the case of the heat flux, a conversion from heating power $\mathcal{F}_{TS}$ in OpenIFS to liquid water potential temperature flux $\mathcal{F}_{\theta_l}$ in DALES is needed,

$$\mathcal{F}_{\theta_l} = -\frac{\mathcal{F}_{TS}}{\Pi(P_S) c_{pd} \rho}, \tag{12}$$

where Π is the Exner function given in Eq. (10). The models also differ in sign conventions: in DALES positive fluxes are upwards, into the atmosphere, while in OpenIFS positive fluxes are downwards.

The coupling of the surface roughness lengths for momentum and heat provides a simple way to account for orographic variations and vegetation for the local models located over land, and for wave height for models above the sea.

2.4 Radiation, cloud condensate and cloud fraction

In our set-up, radiative heating and cooling in the atmosphere is currently handled only in the global model. This choice was mainly motivated from computational considerations as the handling of the radiation by DALES would result in a significant increase in computing time. DALES has the option to locally run the same radiation scheme as OpenIFS, so the option of diverting the radiation scheme to DALES is in principle available.

The cloud fraction A in the global model is derived by calculating the fraction of all the columns in DALES that contain a non-zero cloud condensate in the range from k_1 to k_2. In formula, the cloud fraction A in the global model is given by

$$A = \frac{1}{i_{\max} j_{\max}} \sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} I^{k_2,k_1}(i,j) \tag{13}$$

where i an j are the horizontal grid-indices of DALES, and I is a indicator function which takes the value 1 in the case of any cloud condensate in the subcolumn between levels k_1 and k_2 at the horizontal coordinates i and j, and zero otherwise (Neggers et al., 2011). This choice for deriving A as a cloud fraction "defined-by-area", or projected cloud fraction is deliberate, because in OpenIFS it is implicitly assumed that clouds do not exhibit any subgrid variability within the vertical extent of the layer. Physically this implies that in the case where not all vertical levels in a DALES subcolumn are occupied with cloud condensate, it is averaged out over this subcolumn.

2.5 Other modeling choices

In DALES, the highest 25% of the model levels constitutes a so-called sponge layer It removes fluctuations of wind, temperature and humidity, in order to damp gravity waves before they can reflect at the model top. The damping increases smoothly from the start of the sponge layer to the top of the system. There are several options for the damping mechanism. To be compatible with the superparameterization, we use a scheme where each quantity is relaxed towards the horizontal average of that quantity. Since this scheme preserves the horizontal average, it does not have a strong effect on the global model.

As in earlier superparameterization schemes, we do not couple the vertical velocity w between the two models. Due to the periodic boundary conditions of the local model, the horizontal average of $\langle w \rangle$ vanishes, which excludes the possibility of an explicit coupling. The effect of the large-scale vertical velocity on the prognostic fields in the global model is of course taken into account by the vertical advection and its effect is felt by the local model through the forcing as expressed by Eq. (4). Vice versa, the local model influences to thermodynamic state of the large-scale model and thereby indirectly also the large-scale vertical velocity within the hydrostatic formulation in a similar fashion that conventional convection parameterizations would do.

2.6 DALES horizontal extent and resolution

Since superparameterization does not involve lateral boundary forcings and all exchanged profiles involve bulk properties, the horizontal extent of the local models can in principle be chosen independently from the OpenIFS grid spacing. The DALES model size should be chosen to capture mesoscale structures as much as possible. However, the occurrence of such organization very much depends on the interaction with the large-scale dynamics and is often difficult to predict. For the horizontal DALES grid spacing we use an upper bound around 200 m; beyond this scale the DALES subgrid model can no longer accurately account for the unresolved turbulent motions.

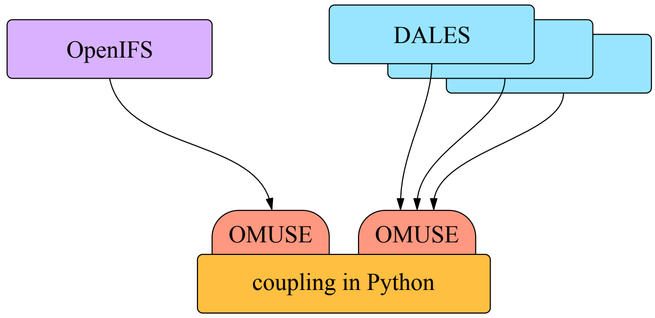

Figure 3. Organization of the superparameterized simulation code, with a top-level coupling program communicating with the global model and a number of local model instances through OMUSE interfaces.

3 Implementation

For superparameterization the OpenIFS model needs a bidirectional coupling to multiple instances of DALES, each one mapping to a different grid point of the global model. The central hypothesis in the design of the superparameterized simulation is that the bulk of the computing time in the coupled system is going to be spent during the time stepping of the local models. This is primarily because OpenIFS has a coarse grid and its numerical scheme allows large time steps, whereas the three-dimensional small-scale models are frequently restricted in their time step and thus have to perform many iterations to catch up with the global model. Hence it is important to allow the independent DALES instances to run concurrently on separate resources whenever available.

Previous superparameterization setups have generally embedded the local model in the column physics routines of the global model. This approach is feasible when superparameterization is applied uniformly in all model columns. In this project, we want to superparameterize only selected grid columns. Due to the organization of the OpenIFS program, it turns out to be difficult to embed DALES in the physics routine of selected columns, while letting the DALES instances run in parallel. For this reason, and also to keep the setup more modular, we settled on a different organization where the two model codes are kept as separate libraries, and an independent coupling program communicates with them. The coupling program is written in Python, and communicates with the models, which are written mainly in Fortran, through a framework named OMUSE. (Pelupessy et al. (2017), see also S. Portegies Zwart et al. (2009); Pelupessy, F. I. et al. (2013); S. F. Portegies Zwart, McMillan, van Elteren, Pelupessy, and de Vries (2013)). This setup is illustrated in Fig. 3. In the following sections we explain the modifications made in OpenIFS and DALES, the role of the OMUSE framework and the coupling program.

3.1 Interface to OpenIFS

For the superparameterization coupler to be able to communicate with OpenIFS, a function interface is defined, with functions for initializing the model, setting tendencies, performing a time step and retrieving the model state.

In the original OpenIFS physics routines, the different physical processes are evaluated in sequence, in order of decreasing characteristic time scale. The model uses a so-called sequential-splitting time stepping scheme, where the model state is updated after each process, so that later processes operate on a state modified by the earlier ones. To preserve this process ordering in the superparameterized model, the coupling to the

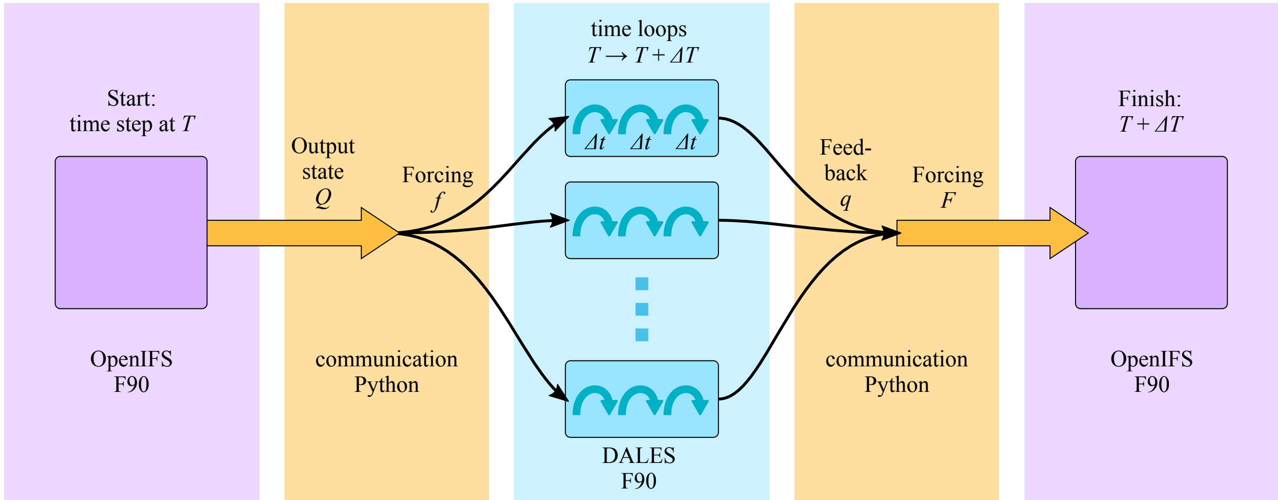

Figure 4. Single time step of the superparameterized OpenIFS.

small-scale model should take place at the stage of replaced processes, namely where the (boundary layer) turbulence, the convection and the cloud scheme in the OpenIFS physics routine are called, as shown in figure 2.

The vertical physics processes in the original OpenIFS are evaluated for one column at a time in a single loop (actually the physical processes act on blocks of columns for optimal cache usage and vectorization, so the external loop runs over these sub-blocks). This means that the states of the superparameterized columns just before the cloud scheme is called, are not all available at the same time. Since these states are required for setting tendencies on the local models before their time evolution start, the local models cannot be time stepped in parallel with this organization of the OpenIFS physics routines.

To overcome this problem, and insert the model coupling at the right stage while keeping the local models parallel, we have split the OpenIFS time step into three global pieces: (i) a routine taking all prognostic fields to a state that has evolved dynamically and that incorporates all vertical physics effects up to the turbulence scheme, (ii) a routine that executes the original turbulence, convection and the cloud scheme on the grid columns not selected for superparameterization, and (iii) a routine that executes all remaining physical processes after the cloud scheme that is being executed subsequently, e.g. mass-fixers and diagnostics. We have also moved all stack-allocated arrays in the original loop over columns to heap-allocated data structures so that the global model keeps its state during the local model time-stepping. The original time step is therefore equivalent to the consecutive execution of these three routines.

To disable the OpenIFS cloud and convection schemes as well as the boundary layer turbulence scheme for the superparameterized grid points, we have introduced a global *superparameterization mask*. All parameterization routines are being executed, but for grid points where the mask is set, the tendencies from these processes are set to zero so that their effects are discarded. In this approach additional diagnostics arising from these parameterizations can still be used, e.g. the surface heat and momentum fluxes are still computed by the OpenIFS surface scheme and transferred to the local models to provide boundary conditions at the beginning of each superparameterization time loop.

3.2 Interface to DALES

A similar library interface was created for DALES, with functions for initializing the model, setting tendencies, evolving the model until a given time, and fetching vertical profiles of the model variables. Creating a library interface for DALES required less involved changes than OpenIFS. The necessary changes were mainly to add interface functions for tendencies and for retrieving horizontal averages of the variables.

3.3 OMUSE coupling framework

OMUSE (Pelupessy et al., 2017) is a framework for creating Python interfaces for scientific codes written in various languages such as C or Fortran. With OMUSE, we created Python versions of the function interfaces to OpenIFS and DALES described above. OMUSE is MPI-aware, making it possible to transparently communicate with MPI-parallelized models.

Through the OMUSE interface, both models can be controlled from a Python program by calling Python functions. Internally, OMUSE translates these function calls to Fortran function calls in the model codes, using MPI. Having MPI as the communication channel between the coupler and the models enable the models to be distributed over multiple nodes in a cluster. Furthermore, OMUSE hides the parallel nature of the models — every function in the OMUSE interface is collective over the MPI tasks of a given model, freeing the coupler from dealing with lower-level details of how the models are parallelized. Using OMUSE is a way to keep the model modular, making it relatively easy to for example substitute DALES with another large eddy simulation code, or even a single-column cloud model.

3.4 The coupling code

The superparameterization couplings described in section 2 have been implemented in a Python program using the OMUSE interfaces to OpenIFS and DALES. Figure 4 shows the interaction of the different components during a time step of the combined model.

Our setup does not require communication between the DALES instances and OpenIFS directly. All interactions are transferred through the coupling program, which fetches and compares the model states and provides feedback to the models in the form of tendencies. For localized superparameterization the communication overhead remains limited. Given the way the superparameterization coupling is formulated, no 3D fields need to be exchanged — vertical profiles are sufficient. The cost of exchanging vertical profiles is generally small compared to the DALES runtime. Having a separate coupling code in Python allows rapid prototyping and easy output of tailored diagnostics of the exchanged tendencies. It also keeps the code modular and easier to maintain.

The OMUSE framework supports (at least on many HPC clusters) dynamic instantiation of models; the coupling script is built to launch DALES instances within a user-defined area at initialization, making it easy to select an area for superparameterization. Furthermore, unit conversions and vertical interpolation of profiles are implemented in the Python driver code. The system supports collection of basic performance statistics and adding a spin-up period at the start of the simulation, where the DALES instances are run for a specified time while being relaxed towards the vertical profiles of the OpenIFS model.

We note that the precipitation is not yet coupled back from the local models to the global model. The global model schemes which rely on the precipitation model for input, the radiation and soil schemes, use the values computed with the global model's parameterization. This may be of relevance for long runs, over time periods so long that the soil properties are influenced by precipitation.

Coupling parameters	
Number of DALES instances	42
Spin-up time	4 h
Duration	21 h

DALES parameters	
Vertical resolution	25 m
Horizontal resolution	200 m
Vertical extent	4 km
Horizontal extent	40 km
Grid size	$200 \times 200 \times 160$
Time step	adaptive, $\approx$ 1...20 s

OpenIFS parameters	
Grid	T511L91
Grid point distance	$\approx$ 40 km
Initial state and Sea Surface Temperature forcing	ERA-Interim
Start time	2012-04-13 at 00:00 UTC
Time step	900 s

Table 2. Parameters used in the simulations

4 Results

Simulations using OpenIFS with a superparameterized setup over the Netherlands are performed for April 13 2012. This day was characterized by a west-northwesterly flow steered by a low over the Northern part of the North Sea. Clear skies are observed over the relatively cold water of the North Sea but convection developed when the cold air was advected from the North Sea over the Netherlands resulting in developing shallow cumulus clouds between 1 and 3 km over land.

The OpenIFS initial state was constructed by interpolation from the reanalysis dataset ERA-Interim (Dee et al., 2011). The local models were initialized with the vertical profiles of their corresponding grid points in the global model, with noise added in the horizontal direction to break the symmetry. After initialization, a spin-up of the local models was performed, where they were run for 4 hours while being relaxed towards the state of the global model. After spin-up the actual simulation was started, and the global and local models were time-stepped together as described in section 2. OpenIFS was run with a T511L91 grid in all cases, giving a distance between neighboring grid points of about 40 km. The simulation parameters are summarized in table 2. For the superparameterized run, we choose the extent of the DALES domains to match the grid point distance in OpenIFS.

4.1 Cloud cover

We compare the appearance of the cloud fields from the simulation with satellite images and with results from the unmodified OpenIFS. We also show vertical profiles of temperature, humidity, and cloud liquid water from the superparameterized run, and compare them with profiles from the unmodified OpenIFS, from the ERA-5 reanalysis and from a radiosonde observation. A full validation of the superparameterized model remains beyond the scope of this work.

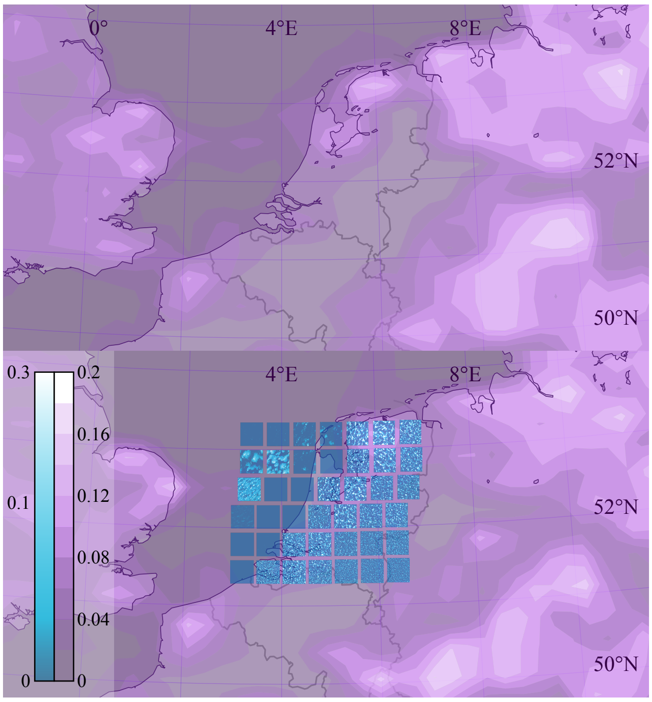

Figure 5. A superparameterized simulation over the Netherlands (bottom), compared to the same simulation run in standard OpenIFS (top). DALES instances are shown in blue, over a background showing the OpenIFS state in purple. Cloudiness (liquid water path) is shown in shades of white for both models. The state shown is for the time 2012-04-13 11:35 UTC; the simulation was started at midnight the same date, with a state from ERA-Interim.

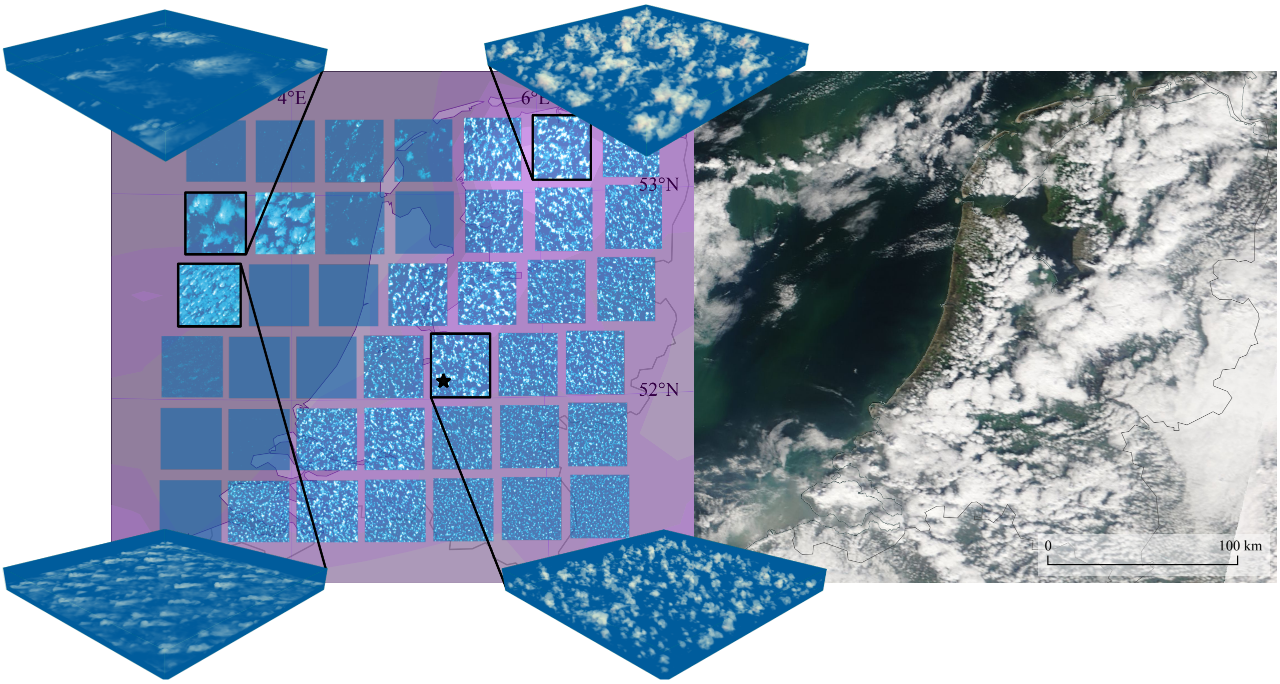

Figure 6. Zoom in on the superparameterized simulation and 3D views on selected local models, compared to a satellite image from Terra / MODIS at 2012-04-13 11:35 UTC. The star marks de Bilt, where radiosonde observations were performed.

Figure 5 shows a snapshot of the superparameterized simulation, compared to a similar run of regular OpenIFS. Animations of the simulations shown are available as supplementary material. The state of the simulations are shown at 11:35 UTC, with the simulations initialized at midnight UTC the same day. This time was chosen to coincide with the overpass of the Terra satellite for comparison.

The figures illustrate the clouds in OpenIFS and in DALES through the liquid water path. Since the cloud optical thickness has a non-linear dependence on the liquid water path, we use a non-linear color map for the DALES instances, (a gamma correction with $\gamma = 1/2$, so the pixel coloring is determined by $(q_L/q_{L\max})^\gamma$). The plot still offers only a crude approximation of how the clouds would actually appear on a satellite image, since the cloud optical thickness depends strongly on the droplet size distribution, which is not taken into account in the plot. A more detailed quantitative comparison follows in section 4.2.

A magnification of the superparameterized run are shown in figure 6 together with a satellite image of the same area. As can be seen in the figure, the local models over land show more clouds than the ones over the sea, in agreement with the satellite image and reanalysis data. However, large, stratiform cloud fields are underrepresented in the superparametrized OpenIFS.

When comparing DALES cloud fields with observations, it is good to keep in mind that the local models in a superparameterized setup give a representation of the convection and clouds at a grid point of the global model, but cannot be expected to accurately reproduce individual features or clouds seen in observations. One reason for this is that the initial state does not provide any small-scale information – the DALES simulations are initialized with vertical profiles from the global model. Secondly, the DALES sim-

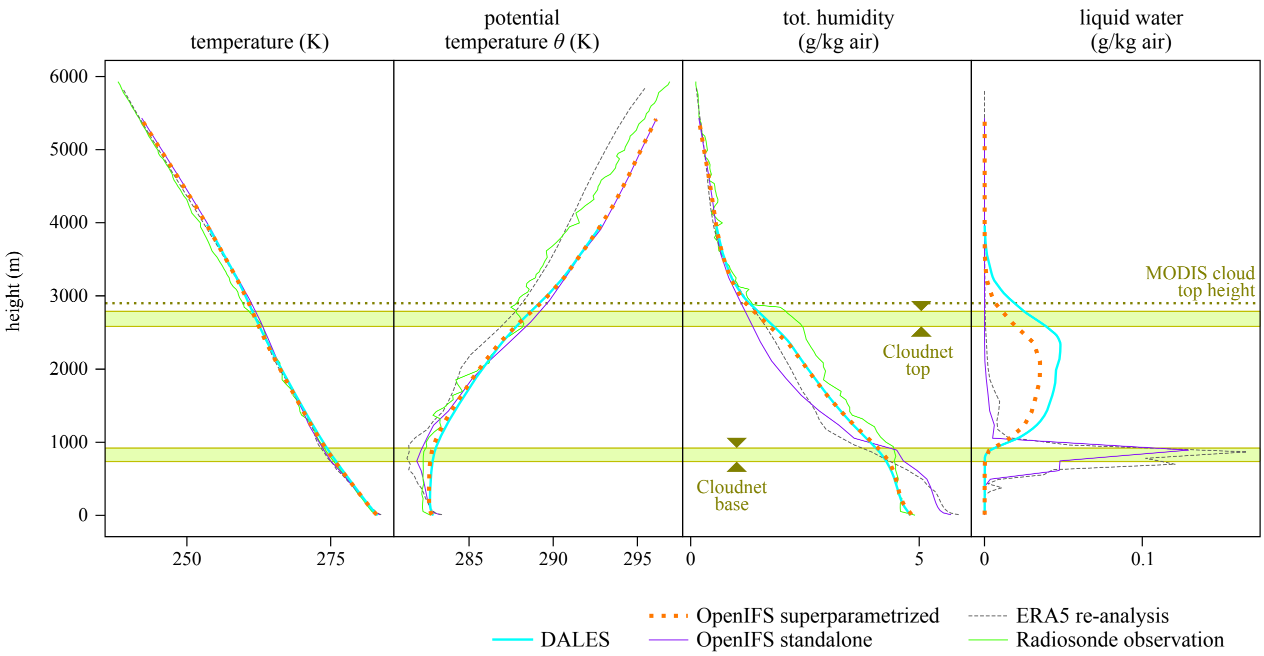

Figure 7. Vertical profiles at 2012-04-13 12:00 UTC, of the grid point closest to de Bilt where radiosonde observations were performed. As a reference we use the ERA5 reanalysis and the KNMI radiosonde observations. The horizontal line shows the cloud top height retrieved from MODIS. The wide horizontal lines show the range of cloud top and cloud base heights a half-hour interval from Cloudnet, recorded by the cloud radar at the Cabauw site, 22 km Southwest of de Bilt.

ulations are performed with periodic boundary conditions, so that spatial coordinates in them do not directly correspond to any particular geographic coordinates.

4.2 Vertical profiles

Vertical profiles of temperature, humidity and liquid water humidity at a single OpenIFS grid point are shown in figure 7. The superparameterized run is show with profiles from both DALES and the corresponding grid point in OpenIFS. It can be seen that in the superparameterized run, DALES and OpenIFS are consistent with each other as can be expected from the coupling scheme. The superparameterized run is compared to a similar run of the non-superparameterized OpenIFS, to the reanalysis from ERA5, and to radiosonde observations. The largest difference in results is seen in the liquid water profiles (right panel), where the superparameterization produces significantly higher clouds than standard OpenIFS. The superparameterization result agrees well with the cloud top height measurement of MODIS, 2900 m. Also the total humidity measured by the radiosonde shows a sharp step at this height, consistent with this being the cloud top height. We additionally compare the liquid water result with cloud radar recordings from Cloudnet (Illingworth et al., 2007), taken at the Cabauw site, 22 km Southwest of de Bilt. Over 30 minutes, the cloud top height was measured at 2690 $\pm$ 100 m, and the cloud base at 830$\pm$90 m. These ranges are indicated with green stripes in figure 7, and show a good agreement with the liquid water results from the superparameterized simulation.

When comparing simulation results with re-analysis, one should remember that the reanalysis was done with IFS, of which OpenIFS is a version. There may thus be a bias for the reanalysis to behave similar to OpenIFS. While the comparison presented here is certainly too limited to draw broad conclusions about the accuracy of the superparameterized simulation, the match between the superparameterized clouds and MODIS, the radiosonde observations, and the Cloudnet cloud top and base heights is encouraging. At the same time the comparisons show that in particular the liquid water profiles from the reanalysis are inconsistent with the MODIS and Cloudnet observations. We note that the parameters of DALES have not been tuned for this comparison.

4.3 Performance

The superparameterized run with the 42 larger DALES models presented above, consisting of 21 hours of simulation and 4 hours of DALES spin-up, took 39 hours on ten nodes of the Cray-XC40 system at ECMWF. Each node in this system contains two 18-core Intel Xeon EP E5-2695 V4 Broadwell processors. Each DALES instance used 8 MPI tasks, resulting in 336 DALES processes in total. The first node ran the coupling script, OpenIFS with 17 processes, and 12 DALES processes. The remaining nine nodes ran 36 DALES processes each. The total cost of this simulation was 14041 core-hours.

Figure 8 shows the distribution of computing time over OpenIFS, the DALES models and the communication, around the time of the snapshot shown in figure 5. Most of the time, 82 % is spent on the DALES models, followed by communication and coupling with 16 %. OpenIFS itself consumes only 2 % of the total time. There is considerable variation in the computational times for the different DALES instances, which appear since DALES uses adaptive time stepping. The more convection there is in a DALES volume, the shorter the time step needs to be, and the longer the simulation takes. This results in some work imbalance, since for each time step of the global model, the whole calculation must wait until the last DALES completes. If DALES would be parallelized with OpenMP in addition to the current MPI parallelization, it might be beneficial to dynamically adjust the number of tasks for each DALES instance to reduce the imbalance. Some further performance can most likely still be gained by carefully optimizing the job layout, and also by overlapping some of the communication steps with computation. In the following section we address ways of accelerating the local models themselves.

4.4 Acceleration

To reduce the computational cost of the superparameterized simulation, we consider ways of accelerating the local models, which consume most of the computational time. First, the horizontal extent of the local models may be chosen smaller than the global model's grid size (Xing et al. (2009) calls this the "reduced space strategy"). Global superparameterization studies have also frequently used this strategy, sometimes combined with making the local models 2-dimensional.

Second, the local models can be accelerated in time, using the mean state acceleration method of Jones, Bretherton, and Pritchard (2015), which was also used in the superparameterization context by Parishani et al. (2017) with promising results. Briefly, this method assumes a separation of time scales, between the time of eddy motion (fast) and the time on which the local model's horizontal averages change (slow). The technique is a good match with superparameterization, since only the horizontal averages, with the slow time scale, are coupled to the global model. In the mean-state acceleration technique, after every time step in the local model, the horizontal averages of the tendencies are calculated. These average tendencies are then applied to the model variables in a horizontally uniform way, in order to accelerate the rate of change for the horizontal means.

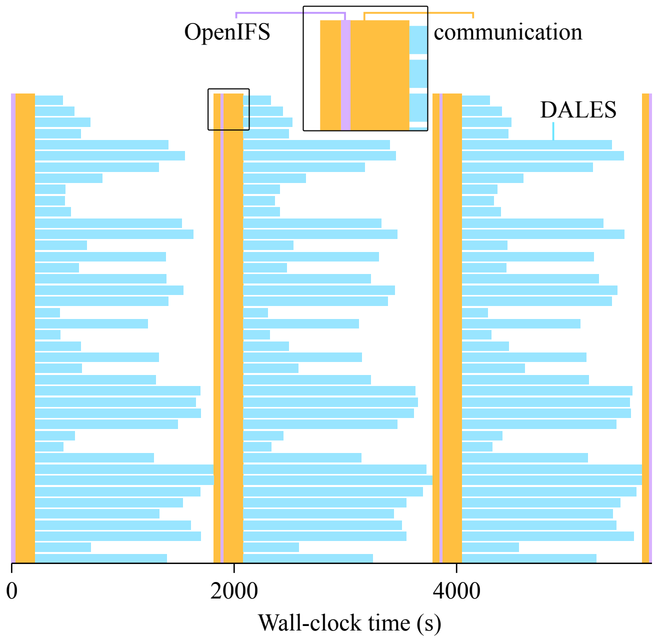

Figure 8. Timing diagram for the superparameterized run, showing three OpenIFS time steps around the time of the snapshot shown in figure 5. The blue horizontal bars show the wall-clock time required for each individual DALES instance to complete the time step. The vertical bars show the OpenIFS computation (purple) and communication between the coupler and the models (orange).

These techniques are demonstrated with two accelerated superparameterization runs with the same initial conditions as in the previous section. In the first, the horizontal extent of DALES is reduced to 64×64 columns, reducing the area to 10 % of the original. In the second run, the mean state acceleration is also applied, with an acceleration factor of 2. To evaluate the accuracy of the accelerated runs, we plot the RMS difference between the global model variables and the ERA-5 reanalysis over time (figure 9). The plots show that the differences introduced by the acceleration are rather small.

The computational time requirements with and without acceleration are shown in table 3. Reducing the area covered by DALES to 10% or the original, reduces the DALES run time by a factor of 8.8. Adding mean-state acceleration with an acceleration factor of 2, further reduces the DALES runtime by a factor of 1.9. The total run time decreases less dramatically, by a factor 4 or 4.9, since the amounts of time spent on coupling and on OpenIFS remain constant. A simple way to decrease the time fraction spent on the coupling and communication is to allocate fewer processes per DALES - for this comparison we kept the job layout the same for all runs. Another interesting optimization possibility is to increase the concurrency of computation in the model codes with communication between the coupler and the models.

These acceleration results seem promising for reducing the computational cost of superparameterization. We also note that the degrees of acceleration that can be achieved may depend on the case.

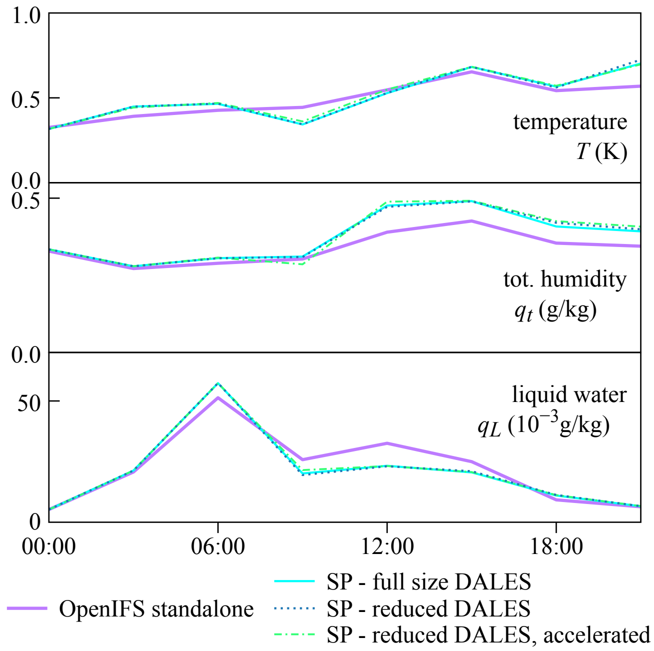

Figure 9. RMS difference between the superparameterized runs and the ERA-5 reanalysis, over all superparameterized grid points for the vertical levels up to the height of the DALES models.

		run time (1000 s)				speedup factor	
LES grid	time acc.	DALES	OpenIFS	coupler	total	DALES	total
$200 \times 200 \times 160$	-	115.4	2.6	22.0	140.0	1.0	1.0
$64 \times 64 \times 160$	-	13.1	2.6	19.2	34.9	8.8	4.0
$64 \times 64 \times 160$	2	6.9	2.6	19.2	28.7	16.7	4.9

Table 3. Comparison of the computational time for different parts of the simulation, with and without acceleration methods. All runs were performed as described in section 4.3, on the same computer system.

5 Conclusions and Outlook

We have demonstrated a superparameterization of a global atmospheric model with a three-dimensional, high-resolution atmospheric LES over a configurable region. We show an example with a 21 hour run where a region with an area of 240 km × 280 km is superparameterized with high-resolution LES models. Reducing the extent of the local models and applying mean-state acceleration drastically reduces the computational demands, with only minor deterioration of the results.

The coupling between the global and local models was implemented with a coupling program in Python, communicating with the model codes using the OMUSE framework. Implementing the coupler in Python as an independent program facilitated flexible development, while providing sufficient performance. Without acceleration methods, the major part of the computation is spent on the local models. With the introduction of acceleration methods for the local models, the performance increased to the point where the time spent in the coupler becomes significant, giving us a motivation to address the coupler performance in the future.

The simulation results demonstrate the potential of this approach. The cloudy regions, observed by MODIS are reproduced by the superparameterized grid boxes. Comparison with the local observations at the Cabauw site shows that the superparameterized version of OpenIFS simulates a deeper convective mixing leading to improved profiles of especially the specific humidity and the cloud amount, compared with the standard version of the OpenIFS without superparameterization.

The simulations also illustrate limitations of this approach. As can be observed from Fig. 6, the superparameterized simulation poorly represents the observed coherent cloud structures of sizes comparable with or beyond the resolution of the global model (40 km). This is in part the consequence of the periodic boundary conditions of the local model and the lack of a direct coupling between the neighboring local model instances. This prevents the growth of mesoscale cloud structures, emerging from the smaller turbulent scales resolved in the local model, beyond the domain size of the local model. The coupling between the local and the global model, as expressed by Eq. (2) and Eq. (4), is exclusively formulated in terms of the mean values of the prognostic (thermo)dynamic variables. Therefore, the coupling does not include any scale interaction of the variances of these variables, though many observational studies have shown a continuous growth of variances of temperature and humidity with the spatial scale as $\ell^{2/3}$, up to scales of several hundreds of kilometers without any scale break (Kahn et al., 2011). Work is therefore in progress to introduce an additional coupling of the variances of the prognostic variables, guided by the behavior of the resolved variance and cloud amount of the global model (Cusack et al., 1999).

This present work and results demonstrate that employing LES models as superparameterization is an attractive and efficient stepping stone toward global LES modeling which is relatively easy to implement in any existing global atmospheric model. More directly, it can also be used as an interactive zoom-in tool to obtain more accurate predictions for high-impact areas such as large national airports or large wind energy and solar power farms. In addition there are numerous other useful applications of this framework more in the context of model development and analysis.

As already mentioned in the introduction, this framework can serve as a useful benchmark for the development of new parameterized approaches. Many of the parameterization developments of cloud related processes of the last twenty years have been guided by Large Eddy Simulations of relevant cases which are forced by realistic large-scale forcings (Brown et al., 2002; Siebesma et al., 2003; vanZanten et al., 2011). The present framework provides new opportunities in this respect. It provides realistic benchmarks over longer periods, over larger areas, with realistic forcings that are easy to set up. The framework also allows the use of different local models, e.g. an alternative parameterization package, data-driven algorithms trained by the LES (Dorrestijn et al., 2013; Schneider, Lan, et al., 2017), or conceptual mixed layer models (Caldwell et al., 2013). This way it is possible to test and compare different approaches which all are in balance with the large scales due to the interactive coupling.

By increasing the resolution of the global model and accordingly, reducing the domain size of the local model, the present framework can also be used to quantify how the response of the local model will change. This will provide guidance for at which resolutions and for which processes a scale-aware parameterizations are required. Such experiments will also be useful in exploring how mesoscale organization is emerging. By varying the resolution, the effect of imposing a scale break at different spatial scales on the mesoscale organization can be systematically explored.

Name	URL	Archived snapshot, DOI
sp-coupler	`https://github.com/CloudResolvingClimateModeling/sp-coupler`	10.5281/zenodo.1968305
OMUSE v1.1	`https://bitbucket.org/omuse/omuse/`	10.5281/zenodo.1407941
AMUSE	`http://www.amusecode.org/`	archived with OMUSE
Dales with OMUSE interface	`https://github.com/CloudResolvingClimateModeling/dales`	10.5281/zenodo.1345110
OpenIFS with OMUSE interface	`https://git.ecmwf.int/scm/~g.vandenoord_esciencecenter.nl/oifs40r1-lib.git` (requires license and a user account at ECMWF)	
OpenIFS, licensing information	https://confluence.ecmwf.int/display/OIFS	

Table 4. References for the different codes used in the superparameterization setup

Acknowledgments

We thank Glenn Carver at the ECMWF for help with OpenIFS and for providing us with initial states for the simulation. We also thank Maria Chertova for comments on the manuscript and Bram van Es for work on the setup and testing of the coupled model.

This work was supported by the Netherlands eScience Center (NLeSC) under grant no. 027.015.G03. Furthermore, we acknowledge the use of ECMWF's computing and archive facilities in the research reported here. Also SURFsara provided computing resources.

The ERA5 and ERA-Interim data was provided by Copernicus Atmosphere Monitoring Service information 2018. Neither the European Commission nor ECMWF is responsible for any use that may be made of the Copernicus Information or Data it contains.

We acknowledge the use of imagery from the NASA Worldview application (https://worldview.earthdata.nasa.gov/) operated by the NASA/Goddard Space Flight Center Earth Science Data and Information System (ESDIS) project.

Radiosonde data was provided by KNMI. It was recorded with a Vaisala RS92 radiosonde, released from de Bilt, the Netherlands.

The maps were drawn with Matplotlib (Hunter, 2007) and Basemap. The 3D cloud images were raytraced in Blender.

Code Availability

All the codes required for the simulation are available under open-source licenses, except OpenIFS (for which a license can be requested from ECMWF). Table 4 lists the codes and URLs for repositories and DOI numbers for archived snapshots.

The top-level coupler code in Python is called sp-coupler. To run, it requires OMUSE and versions of OpenIFS and DALES which include the OMUSE interfaces. For installation instructions, see the documentation in the coupler repository. The coupler repository includes a Singularity recipe. Singularity (Kurtzer et al., 2017) is a software container system for scientific applications. The Singularity recipe is used to build a singularity image — a self-contained unit containing all the programs and all their dependencies needed to run the simulation. Building the Singularity image also requires access to OpenIFS source code from ECMWF.

Author contributions

DC and PS conceived of the project. FJ, GvdO, DC, PS defined the coupling procedure. FJ, GvdO, IP wrote the coupling code and the OMUSE interfaces to DALES. GvdO created the OMUSE interface to OpenIFS. FJ ran the simulations. JHG and FJ developed the visualizations. JHG drew the figures. FJ wrote the article text, with contributions and editing by all other authors.

Appendix: Analysis of the coupling scheme

Here we analyze the model coupling scheme in more detail. We show that the desired equality (1) between the global and the (averaged) local model state is achieved in cases where the advection and source terms are entirely situated either in the local model or in the global model.

Consider steps (i) to (iv) of the superparameterization scheme given in section 2. Combining (i) and (ii) gives

$$Q(T+\Delta T) = \langle q(T)\rangle + \Delta T(A_Q(T) + S_Q(T)). \tag{14}$$

In (iv), $f_q(T)$ does not change with t, and thus $\sum_{t=T}^{T+\Delta T} \Delta t f_q(T) = \Delta T\, f_q$. With (iii) and (iv), this leads to $q(T+\Delta T) = q(T) + Q(T+\Delta T) - \langle q(T)\rangle + \sum_{t=T}^{T+\Delta T} \Delta t(a_q(t) + s_q(t))$. Taking the horizontal average we find

$$\langle q(T+\Delta T)\rangle = Q(T+\Delta T) + \sum_{t=T}^{T+\Delta T} \Delta t\langle a_q(t) + s_q(t)\rangle. \tag{15}$$

In general, $Q(T) \neq \langle q(T)\rangle$ and $Q(T+\Delta T) \neq \langle q(T)\rangle$, as can be seen from the identities just derived. Thus, the equality (1) is generally not satisfied. However, if all of the advection and sources with nonzero average are accounted for in one model (global or local) and none in the other, the equality is satisfied (albeit possibly in a time-lagged sense). More precisely, assume $A_Q(T) + S_Q(T) = 0$, i.e. all advection and sources are in the local model. Then $Q(T+\Delta T) = \langle q(T)\rangle$ by construction. Conversely, if the local model has no advection or source terms with nonzero horizontal average, so that $\langle a_q(t)+s_q(t)\rangle = 0$ for all t, we have $\langle q(T+\Delta T)\rangle = Q(T+\Delta T)$.

If both $A_Q(T) + S_Q(T) \neq 0$ and $\langle a_q(t) + s_q(t)\rangle \neq 0$, we can consider the difference between $Q(T+\Delta T)$ on the one hand and a weighted average of $\langle q(T)\rangle$ and $\langle q(T+\Delta T)\rangle$ on the other hand. Defining the weighting parameter α with $0 \leq \alpha \leq 1$, we have

$$\begin{aligned} Q(T) - (\alpha\langle q(T)\rangle + (1-\alpha)\langle q(T+\Delta T)\rangle) = \\ \alpha\,\Delta T\,(A_Q(T) + S_Q(T)) - (1-\alpha) \sum_{t=T}^{T+\Delta T} \Delta t\langle a_q(t) + s_q(t)\rangle\,. \end{aligned} \tag{16}$$

The RHS equals zero if $\Delta T\,(A_Q(T) + S_Q(T))$ and $\sum_{t=T}^{T+\Delta T} \Delta t\langle a_q(t) + s_q(t)\rangle$ have the same sign and their ratio equals $(1-\alpha)/\alpha$. To satisfy the latter requirement, α must depend on time T, vertical level z and prognostic variable q, Q.

References

Bony, S., Stevens, B., Frierson, D. M. W., Jakob, C., Kageyama, M., Pincus, R., ... Webb, M. J. (2015, mar). Clouds, circulation and climate sensitivity. *Nature Geoscience*, *8*, 261. doi: 10.1038/ngeo2398

Bretherton, C. S., & Khairoutdinov, M. F. (2015). Convective self-aggregation feedbacks in near-global cloud-resolving simulations of an aquaplanet. *Journal of Advances in Modeling Earth Systems*, *7*(4), 1765-1787. doi: 10.1002/2015MS000499

Brown, A. R., Cederwall, R. T., Chlond, A., Duynkerke, P. G., Golaz, J.-C., Khairoutdinov, M., … Stevens, B. (2002). Large-eddy simulation of the diurnal cycle of shallow cumulus convection over land. *Quarterly Journal of the Royal Meteorological Society*, *128*(582), 1075-1093. doi: 10.1256/003590002320373210

Caldwell, P. M., Zhang, Y., & Klein, S. A. (2013). Cmip3 subtropical stratocumulus cloud feedback interpreted through a mixed-layer model. *Journal of Climate*, *26*(5), 1607-1625. doi: 10.1175/JCLI-D-12-00188.1

Carver, G., & Vana, F. (2017). Openifs home. *""*. Retrieved from `https://software.ecmwf.int/wiki/display/OIFS/OpenIFS1Home`

Cusack, S., Edwards, J. M., & Kershaw, R. (1999). Estimating the subgrid variance of saturation, and its parametrization for use in a gcm cloud scheme. *Quarterly Journal of the Royal Meteorological Society*, *125*(560), 3057-3076. doi: 10.1002/qj.49712556013

Dee, D. P., Uppala, S. M., Simmons, A. J., Berrisford, P., Poli, P., Kobayashi, S., … Vitart, F. (2011). The ERAInterim reanalysis: configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, *137*(656), 553-597. doi: 10.1002/qj.828

Dorrestijn, J., Crommelin, D. T., Siebesma, A. P., & Jonker, H. J. (2013). Stochastic parameterization of shallow cumulus convection estimated from high-resolution model data. *Theoretical and Computational Fluid Dynamics*, *27*(1-2), 133–148. doi: 10.1007/s00162-012-0281-y

Grabowski, W. W. (2001). Coupling cloud processes with the large-scale dynamics using the cloud-resolving convection parameterization (CRCP). *Journal of the Atmospheric Sciences*, *58*(9), 978-997. doi: 10.1175/1520-0469(2001)058⟨0978:CCPWTL⟩2.0.CO;2

Grabowski, W. W. (2004). An improved framework for superparameterization. *Journal of the Atmospheric Sciences*, *61*(15), 1940-1952. doi: 10.1175/1520-0469(2004)061⟨1940:AIFFS⟩2.0.CO;2

Grabowski, W. W. (2016). Towards global large eddy simulation: Super-parameterization revisited. *Journal of the Meteorological Society of Japan. Ser. II*, *94*(4), 327-344. doi: 10.2151/jmsj.2016-017

Grabowski, W. W., & Smolarkiewicz, P. K. (1999). CRCP: a cloud resolving convection parameterization for modeling the tropical convecting atmosphere. *Physica D: Nonlinear Phenomena*, *133*(1- 4), 171 - 178. doi: 10.1016/S0167-2789(99)00104-9

Heinze, R., Dipankar, A., Henken, C. C., Moseley, C., Sourdeval, O., Trmel, S., … Quaas, J. (2017). Large-eddy simulations over germany using icon: a comprehensive evaluation. *Quarterly Journal of the Royal Meteorological Society*, *143*(702), 69-100. doi: 10.1002/qj.2947

Heus, T., van Heerwaarden, C. C., Jonker, H. J. J., Pier Siebesma, A., Axelsen, S., van den Dries, K., … Vilà-Guerau de Arellano, J. (2010). Formulation of the Dutch Atmospheric Large-Eddy Simulation (DALES) and overview of its applications. *Geoscientific Model Development*, *3*(2), 415–444. doi: 10.5194/gmd-3-415-2010

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, *9*(3), 90–95. doi: 10.1109/MCSE.2007.55

Illingworth, A. J., Hogan, R. J., O'Connor, E., Bouniol, D., Brooks, M. E., Delano, J., … Wrench, C. L. (2007). Cloudnet: Continuous evaluation of cloud profiles in seven operational models using ground-based observations. *Bulletin of the American Meteorological Society*, *88*(6), 883-898. doi:

10.1175/BAMS-88-6-883

Jones, C. R., Bretherton, C. S., & Pritchard, M. S. (2015). Mean-state acceleration of cloud-resolving models and large eddy simulations. *Journal of Advances in Modeling Earth Systems*, *7*(4), 1643-1660. doi: 10.1002/2015MS000488

Jung, J., & Arakawa, A. (2010). Development of a quasi3d multiscale modeling framework: Motivation, basic algorithm and preliminary results. *Journal of Advances in Modeling Earth Systems*, *2*(4). doi: 10.3894/JAMES.2010.2.11

Kahn, B. H., Teixeira, J., Fetzer, E. J., Gettelman, A., Hristova-Veleva, S. M., Huang, X., … Zhao, M. (2011). Temperature and water vapor variance scaling in global models: Comparisons to satellite and aircraft data. *Journal of the Atmospheric Sciences*, *68*(9), 2156-2168. doi: 10.1175/2011JAS3737.1

Khairoutdinov, M., Randall, D., & DeMott, C. (2005). Simulations of the atmospheric general circulation using a cloud-resolving model as a superparameterization of physical processes. *Journal of the Atmospheric Sciences*, *62*(7), 2136-2154. doi: 10.1175/JAS3453.1

Khairoutdinov, M., & Randall, D. A. (2001). A cloud resolving model as a cloud parameterization in the NCAR community climate system model: Preliminary results. *Geophysical Research Letters*, *28*(18), 3617-3620. doi: 10.1029/2001GL013552

Kurtzer, G. M., Sochat, V., & Bauer, M. W. (2017, 05). Singularity: Scientific containers for mobility of compute. *PLOS ONE*, *12*(5), 1-20. doi: 10.1371/journal.pone.0177459

Marchand, R., & Ackerman, T. (2011). A cloud-resolving model with an adaptive vertical grid for boundary layer clouds. *Journal of the Atmospheric Sciences*, *68*(5), 1058-1074. doi: 10.1175/2010JAS3638.1

Miyamoto, Y., Kajikawa, Y., Yoshida, R., Yamaura, T., Yashiro, H., & Tomita, H. (2013). Deep moist atmospheric convection in a subkilometer global simulation. *Geophysical Research Letters*, *40*(18), 4922-4926. doi: 10.1002/grl.50944

Neggers, R. A. J., Heus, T., & Siebesma, A. P. (2011). Overlap statistics of cumuliform boundary-layer cloud fields in large-eddy simulations. *Journal of Geophysical Research: Atmospheres*, *116*(D21). doi: 10.1029/2011JD015650

Parishani, H., Pritchard, M. S., Bretherton, C. S., Wyant, M. C., & Khairoutdinov, M. (2017). Toward low-cloud-permitting cloud superparameterization with explicit boundary layer turbulence. *Journal of Advances in Modeling Earth Systems*, *9*(3), 1542–1571. doi: 10.1002/2017MS000968

Pelupessy, I., van Werkhoven, B., van Elteren, A., Viebahn, J., Candy, A., Portegies Zwart, S., & Dijkstra, H. (2017). The oceanographic multipurpose software environment (omuse v1.0). *Geoscientific Model Development*, *10*(8), 3167–3187. doi: 10.5194/gmd-10-3167-2017

Pelupessy, F. I., van Elteren, A., de Vries, N., McMillan, S. L. W., Drost, N., & Portegies Zwart, S. F. (2013). The astrophysical multipurpose software environment. *A&A*, *557*, A84. doi: 10.1051/0004-6361/201321252

Portegies Zwart, S., McMillan, S., Harfst, S., Groen, D., Fujii, M., Ó Nualláin, B., … Zemp, M. (2009). A multiphysics and multiscale software environment for modeling astrophysical systems. *New Astronomy*, *14*(4), 369 - 378. doi: https://doi.org/10.1016/j.newast.2008.10.006

Portegies Zwart, S. F., McMillan, S. L., van Elteren, A., Pelupessy, F. I., & de Vries, N. (2013). Multi-physics simulations using a hierarchical interchangeable software interface. *Computer Physics Communications*, *184*(3), 456 - 468. doi: https://doi.org/10.1016/j.cpc.2012.09.024

Schalkwijk, J., Jonker, H. J. J., Siebesma, A. P., & Van Meijgaard, E. (2015). Weather forecasting using gpu-based large-eddy simulations. *Bulletin of the American Meteorological Society*, *96*(5), 715-723. doi: 10.1175/BAMS-D-14-00114.1

Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017). Earth system modeling

2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, *44*(24), 12,396-12,417. doi: 10.1002/2017GL076101

Schneider, T., Teixeira, J., Bretherton, C. S., Brient, F., Pressel, K. G., Schär, C., & Siebesma, A. P. (2017, January). Climate goals and computing the future of clouds. *Nature Climate Change*, *7*, 3–. doi: 10.1038/nclimate3190

Siebesma, A. P., Bretherton, C. S., Brown, A., Chlond, A., Cuxart, J., Duynkerke, P. G., ... Stevens, D. E. (2003). A large eddy simulation intercomparison study of shallow cumulus convection. *Journal of the Atmospheric Sciences*, *60*(10), 1201-1219. doi: 10.1175/1520-0469(2003)60⟨1201:ALESIS⟩2.0.CO;2

Tao, W., & Chern, J. (2017). The impact of simulated mesoscale convective systems on global precipitation: A multiscale modeling study. *Journal of Advances in Modeling Earth Systems*, *9*(2), 790-809. doi: 10.1002/2016MS000836

vanZanten, M. C., Stevens, B., Nuijens, L., Siebesma, A. P., Ackerman, A. S., Burnet, F., ... Wyszogrodzki, A. (2011). Controls on precipitation and cloudiness in simulations of trade-wind cumulus as observed during rico. *Journal of Advances in Modeling Earth Systems*, *3*(2). doi: 10.1029/2011MS000056

Xing, Y., Majda, A. J., & Grabowski, W. W. (2009). New efficient sparse space-time algorithms for superparameterization on mesoscales. *Monthly Weather Review*, *137*(12), 4307-4324. doi: 10.1175/2009MWR2858.1