Development and application of a dynamic approach to 3D radiative transfer in subkilometer-scale numerical weather prediction models

Richard Hermann Maier

München 2025

Development and application of a dynamic approach to 3D radiative transfer in subkilometer-scale numerical weather prediction models

Richard Hermann Maier

Dissertation
der Fakultät für Physik
der Ludwig-Maximilians-Universität
München

vorgelegt von
Richard Hermann Maier
aus Regensburg

München, den 7. Mai 2025

Erstgutachter: Prof. Dr. Bernhard Mayer
Zweitgutachter: Dr. Fabian Hoffmann
Tag der mündlichen Prüfung: 18. Juni 2025

Zusammenfassung

Die zunehmende horizontale Auflösung numerischer Wettervorhersagemodelle macht den dreidimensionalen (3D) Strahlungstransport zwischen den einzelnen vertikalen Säulen dieser Modelle immer relevanter. Allerdings sind 3D-Strahlungstransportlöser nach wie vor sehr rechenintensiv, was ihren Einsatz in der operationellen Wettervorhersage bis heute verhindert. Um dieses Problem anzugehen, entwickelten Jakub and Mayer (2015) den TenStream-Löser. Er erweitert die etablierte Zweistrommethode auf drei Dimensionen, indem er zehn statt zwei Ausbreitungsströme verwendet, um den Transport von Strahlung durch die Erdatmosphäre zu beschreiben. Aufbauend auf dieser Methode stellt diese Arbeit den dynamischen TenStream-Löser vor, der eine weitere Beschleunigung des ursprünglichen TenStream-Modells darstellt. Im Vergleich zu herkömmlichen Lösern wird die Rechenzeit dabei vor allem durch drei Methoden reduziert. Erstens wird die Strahlung nicht bei jedem Aufruf des Modells neu berechnet, sondern mithilfe eines Zeitschrittverfahrens auf Basis der Ergebnisse des vorherigen Zeitschritts aktualisiert. Zweitens wird die Konvergenz in Richtung der neuen Lösung durch eine Optimierung des Iterationsverfahrens durch das zugrundeliegende lineare Gleichungssystem beschleunigt. Da sich das aktualisierte Strahlungsfeld zudem nicht grundlegend von dem vorherigen unterscheiden sollte, werden drittens nur die ersten Schritte des Iterationsverfahrens durchgeführt; der Algorithmus wird also bewusst abgebrochen, bevor vollständige Konvergenz erreicht ist. Dieser Ansatz ermöglicht die Berechnung der ausgehenden Strahlungsflüsse einer jeden Gitterbox allein auf Basis der eingehenden Flüsse aus benachbarten Gitterboxen, wodurch der Strahlungstransport näher an die Behandlung von Advektion im dynamischen Kern eines Wettermodells heranrückt und sich zudem erheblich leichter parallelisieren lässt.

Anhand einer vorberechneten Zeitreihe flacher Cumulusbewölkung wird dieser neue Löser anschließend sowohl hinsichtlich seiner Rechenzeit als auch bezüglich seiner Genauigkeit evaluiert. Hinsichtlich seiner Rechenzeit zeigt sich, dass der dynamische TenStream-Löser etwa dreimal langsamer als eine klassische 1D-δ-Eddington-Näherung, jedoch deutlich schneller als andere 3D-Löser ist. Um seine Genauigkeit zu beurteilen, werden die Ergebnisse des neuen Lösers mit einer klassischen 1D-δ-Eddington-Näherung, dem ursprünglichen TenStream-Löser und dem 3D-Monte-Carlo-Modell MYSTIC verglichen, wobei letzteres als Benchmark fungiert. Auf Gitterboxebene zeigt sich dabei, dass die durch den dynamischen TenStream-Löser berechneten Heizraten sowie Nettobestrahlungsstärken am Boden und am Oberrand der Atmosphäre den Ergebnissen des ursprünglichen TenStream-Lösers sehr nahe kommen und

somit den MYSTIC-Benchmark sehr viel besser abbilden als die 1D-δ-Eddington-Näherung. Indem der dynamische TenStream-Löser im Vergleich zur 1D-δ-Eddington-Näherung weniger oft aufgerufen wird, ergibt sich zudem, dass der neue Löser deutlich genauere Ergebnisse liefert als eine 1D-δ-Eddington-Näherung bei vergleichbarem Rechenaufwand. Allerdings führt eine solche Reduktion der Aufrufe im Laufe der Zeit auch zu einem zunehmenden Bias gegenüber der vollen TenStream-Lösung, der umso größer wird, je seltener der neue Löser aufgerufen wird.

Um die Auswirkungen des neuen Lösers auf die Entwicklung von Wolken in Wettermodellen zu untersuchen, wird dieser zudem noch auf eine Grobstruktursimulation angewandt. Dabei zeigt sich, dass sich die Wolken dort ähnlich wie in der Simulation mit dem ursprünglichen TenStream-Löser zu Wolkenstraßen organisieren, die senkrecht zum Einfallswinkel der Sonne ausgerichtet sind. In der von 1D-Strahlung angetriebenen Simulation tritt hingegen keinerlei derartige Organisation auf. Außerdem wird gezeigt, dass von 3D-Strahlung angetriebene Wolken tagsüber horizontal größer und vertikal mächtiger als ihre von 1D-Strahlung angetriebenen Pendants werden und zudem über einen höheren Flüssigwassergehalt verfügen. Nachts hingegen sind die Wolken im direkten Vergleich dünner und enthalten weniger Flüssigwasser. Es werden zwei Mechanismen identifiziert, die diese Unterschiede mit Merkmalen des 3D-Strahlungsfelds in Verbindung bringen. Zum einen befinden sich die Wolken in den 3D-Simulationen über Regionen mit erhöhter solarer Nettobodenstrahlung, anstatt — wie in der 1D-Simulation — direkt über ihrem eigenen Schatten positioniert zu sein, was die mit den Wolken verbundenen Aufwinde verstärkt, anstatt sie abzuschwächen. Zum anderen zeigt sich auch, dass die horizontal gemittelte thermische Nettoabstrahlung des Bodens in den von 3D-Strahlung angetriebenen Simulationen geringer ausfällt, wobei das daraus entstehende Ungleichgewicht in der Strahlungsbilanz des Bodens hauptsächlich durch eine Erhöhung des horizontal gemittelten latenten Wärmeflusses ausgeglichen wird, was zu einer verstärkten Freisetzung von Wasserdampf in die Atmosphäre führt. Beide Mechanismen werden sowohl vom ursprünglichen, als auch dem dynamischen TenStream-Löser erfasst, was eindrucksvoll die Fähigkeit des letzteren demonstriert, 3D-Strahlungseffekte und deren Einfluss auf Wolken auch mit deutlich reduziertem Rechenaufwand effektiv abzubilden.

Abstract

The increasing horizontal resolution of numerical weather prediction (NWP) models makes inter-column three-dimensional (3D) radiative transfer more and more important. However, 3D radiative transfer solvers are still computationally expensive, largely preventing their use in operational weather forecasting. To address this limitation, Jakub and Mayer (2015) developed the TenStream solver. It extends the well-established two-stream method to three dimensions by using ten instead of two streams to describe the transport of radiative energy through Earth's atmosphere. Building upon this method, this thesis presents the dynamic TenStream solver, a further acceleration of the original TenStream model. Compared to traditional solvers, its speed-up is achieved through three main concepts. First, instead of recalculating radiation from scratch every time the model is called, a time-stepping scheme is used to update the radiative field based on the result from the previous radiation time step. Second, convergence toward the new solution is accelerated by optimizing the iteration procedure through the underlying system of linear equations. And third, since the updated radiative field should not be markedly different from the previous one, just the first few steps of an iterative scheme toward convergence are performed, essentially exiting the algorithm before full convergence is reached. With this concept, the outgoing radiative fluxes of each grid box can be updated by taking only incoming fluxes from neighboring grid boxes into account, aligning radiative transfer more closely with the treatment of advection in the dynamical core of an NWP model and facilitating model parallelization.

Using a precomputed shallow cumulus cloud time series, the performance of this new solver is evaluated in terms of both speed and accuracy. In terms of speed, the dynamic TenStream solver is shown to be about three times slower than a classical 1D δ-Eddington approximation, but noticeably faster than currently available 3D solvers. To evaluate the accuracy of the new solver, its results, as well as calculations carried out with a 1D δ-Eddington approximation and the original TenStream solver, are compared to benchmark calculations performed with the 3D Monte Carlo model MYSTIC. At the grid box level, the dynamic TenStream solver is shown to calculate heating rates as well as net irradiances at the surface and at the top of the atmosphere that closely match those obtained by the original TenStream solver, thus providing a much better representation of the MYSTIC benchmark than the 1D δ-Eddington results. By calling the dynamic TenStream solver less frequently than the 1D δ-Eddington approximation, the new solver is furthermore shown to produce significantly more accurate results than a 1D δ-Eddington approximation carried out with a similar computational demand. At these lower calling frequencies, however, the incomplete solves in the dynamic TenStream solver also lead to a buildup of bias over time, which becomes larger the lower the calling frequency is.

To assess its impact on cloud development, the new solver is furthermore coupled to a large-eddy simulation with an interactive land surface. Similar to simulations driven by the original TenStream solver, daytime clouds driven by the dynamic TenStream solver are shown to organize into cloud streets oriented perpendicular to the angle of solar incidence, unlike the random positioning observed with 1D radiation. Additionally, clouds driven by 3D radiation are demonstrated to grow larger, become thicker and contain more liquid water during the day, but get thinner and contain less liquid water at night compared to their 1D-driven counterparts. Two mechanisms are identified that link these differences to features of the 3D radiative field. First, the clouds in the 3D simulations are shown to be positioned above areas of enhanced solar net surface irradiance rather than above their own shadows, strengthening rather than weakening the associated updrafts. Second, the domain-averaged net thermal emission at the ground is shown to be smaller in simulations driven by 3D radiation, with the resulting surface energy imbalance primarily compensated by an increase in the domain-averaged latent heat flux, leading to a greater release of water vapor into the atmosphere. Both of these mechanisms are captured by both the original and the dynamic TenStream solver, demonstrating the latter's ability to reproduce 3D radiative effects and their influence on clouds at a significantly lower computational cost.

Publication

Parts of this thesis have been published in:

> Maier, R., Jakub, F., Emde, C., Manev, M., Voigt, A., and Mayer, B.: A dynamic approach to three-dimensional radiative transfer in subkilometer-scale numerical weather prediction models: the dynamic TenStream solver v1.0, Geoscientific Model Development, 17, 3357–3383, https://doi.org/10.5194/gmd-17-3357-2024, 2024.

Specifically, the content of Section 2 of the publication is presented in Chapter 3 of this thesis, whereas Sections 3 and 4 are covered in Chapter 4. The content of these sections has been included in the thesis with only minor modifications, primarily to reflect the different structure of this work. Additionally, selected and more substantially revised material from the abstract, as well as from Sections 1 and 5 of the publication appears in the abstract and in Chapters 1 and 6 of this thesis.

Contents

Chapter 1

Introduction

Radiation is the main driver of atmospheric dynamics, shaping it across a variety of length scales. Already on the global scale, Earth's spherical nature causes significant differences in the radiative energy received at different latitudes. While equatorial regions are intensely heated, the poles receive far less energy, establishing a temperature gradient that underpins the entire global atmospheric circulation. But also on smaller scales, radiation drives a range of atmospheric processes. By heating near-surface air masses, it initiates convection, causing the warmed air to rise, cool, and eventually condense into clouds. These clouds, in turn, reshape the distribution of radiative energy in the atmosphere by scattering incoming sunlight and casting shadows on the ground, thereby influencing the subsequent evolution of the convective cells. At night, the absence of solar radiation cools the surface, facilitating condensation near the ground and enabling fog formation. A shared characteristic of all these processes is their initiation by radiation-induced temperature changes, with sources of radiative energy heating air masses and sinks cooling them.

This impact on temperature directly links radiation to the primitive equations of atmospheric dynamics — the set of nonlinear partial differential equations at the heart of modern numerical weather prediction (NWP). Based on initial conditions, these equations enable the prediction of the atmosphere's future evolution. Because they have no known analytical solution, they are solved numerically, for instance by dividing the atmosphere into grid boxes and discretizing the equations in both space and time. One of the key variables integrated forward in time in this process is temperature, whose tendencies are partially governed by the aforementioned sources and sinks of radiative energy. For each grid box, these contributions — quantified by heating rates and net surface irradiances — are calculated using radiative transfer models that, ideally, account for full three-dimensional (3D) transport of energy, ensuring that the radiation-induced temperature changes are captured as accurately as possible.

Depending on scale, two distinct regimes of this 3D transport of energy can be identified. At the model grid scale, 3D radiative transfer accounts for the horizontal and vertical transport of radiative energy between neighboring grid boxes. At the sub-grid scale, it addresses the 3D transport of radiative energy within a single heterogeneous grid box, capturing the effects of internal variability, such as the presence of both cloudy and clear-sky regions within a single cell. The computation of both of these effects is computationally expensive, which has largely

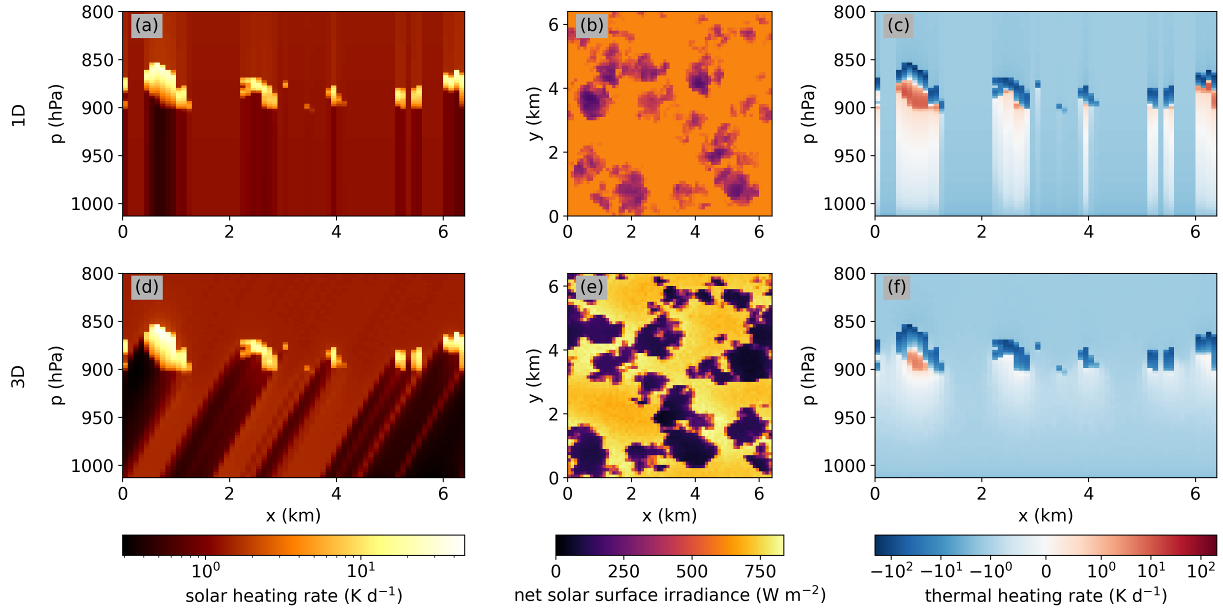

Figure 1.1: Comparison of radiative transfer calculations performed with a 1D δ-Eddington approximation (panels (**a**)–(**c**)) and the 3D solver MYSTIC (panels (**d**)–(**f**)). Panels (**a**) and (**d**) show xz cross sections of heating rates in the solar spectral range, whereas panels (**c**) and (**f**) depict these heating rates in the thermal spectral range. In the middle, panels (**b**) and (**e**) illustrate the net surface irradiance in the solar spectral range. The simulations assume the Sun shining from the east at a zenith angle of 50°. The colorbars for each pair of plots are provided directly underneath them: logarithmic for panels (**a**) and (**d**), linear for panels (**b**) and (**e**), and a combination of logarithmic (for values exceeding 1 or below -1 K d^{-1}) and linear (in between) for panels (**c**) and (**f**).

prevented their inclusion in operational weather forecasting. As a result, most NWP models continue to rely on one-dimensional (1D) independent-column approximations (ICAs), such as the Monte Carlo Independent Column Approximation (McICA; Pincus et al. (2003)), currently employed at the Deutscher Wetterdienst (DWD) and the European Centre for Medium-Range Weather Forecasts (ECMWF) (DWD, 2021; Hogan and Bozzo, 2018). These models assume that radiative transfer between grid boxes only takes place in the vertical and neglect any horizontal energy transport — both between and within individual grid boxes.

However, both grid-scale and sub-grid-scale 3D effects have been shown to be important for the correct calculation of radiative transfer in the atmosphere. While sub-grid-scale 3D effects primarily arise at coarser resolutions, where individual grid boxes encompass both cloudy and clear-sky regions and cannot be treated as homogeneous, the increasing horizontal resolution of NWP models makes especially inter-column radiative transfer increasingly important (O'Hirok and Gautier, 2005). To illustrate this, Fig. 1.1 compares radiative transfer calculations performed with a conventional 1D solver and a full 3D solver at a relatively high horizontal resolution of 100 m. Panels (a) and (d) show xz cross sections of solar heating rates. The yellowish areas above 900 hPa indicate clouds, which act as relatively strong absorbers and therefore as strong net sources of radiative energy in the solar spectral range, whereas the dark areas beneath the yellowish areas indicate the shadows of these clouds. In the 3D simulation in panel (d), these shadows are displaced to the west, consistent with the angle of solar incidence, whereas in the 1D simulation in panel (a), they are cast directly underneath the clouds. This shadow displacement is one of the most prominent 3D radiative effects and becomes increasingly important as the horizontal resolution of the models increases, since at finer resolutions, shadows extend across a growing number of vertical columns instead of being confined to one. The effect can also be observed in panels (b) and (e), which illustrate the impact of 3D radiation at the surface. For instance, the shadow in the lower right of the 1D simulation appears much further west — at approximately $x = 4\,\mathrm{km}$ — in the 3D simulation. Apart from this displacement, the surface plots also reveal that shadows are significantly darker in the 3D simulation, whereas the areas surrounding them are noticeably brighter. This occurs because, with 1D radiation, scattered light remains confined to the same column as the cloud shadow, whereas with 3D radiative transfer, it is also scattered into areas surrounding the cloud shadows (Gristey et al., 2020). The amount of radiative energy in these areas is further increased by radiation scattered from cloud sides (Hogan and Shonk, 2013), as well as radiation entering through gaps between clouds, which subsequently becomes trapped between the clouds and the surface (Hogan et al., 2019). These effects, which cannot be accounted for with 1D radiation, result in net surface irradiances outside the cloud shadows that clearly exceed the clear-sky values of the 1D simulation. But also in the thermal spectral range, 3D radiative transfer plays an increasingly important role. Comparing panels (c) and (f), we can for example see that with 1D radiation, thermal shadows cast by clouds are much more pronounced, as they cannot be weakened toward the ground through interactions with neighboring columns. Additionally, while clouds cool at the top and warm at the base in 1D radiative transfer calculations, 3D radiation also accounts for cloud-side cooling, significantly reducing the amount of cloud-bottom warming. As with the other 3D effects discussed, this cloud-side cooling becomes increasingly important at higher resolutions, where clouds occupy more than a single grid box, allowing these finer-scale processes to be resolved.

Consequently, the impact of 3D radiative transfer has mainly been investigated using large-eddy simulations (LESs), numerical models operating at hectometer-scale horizontal resolutions and below. Within these high-resolution models, 3D radiation has been shown to strongly influence both the organization and evolution of clouds. Klinger et al. (2017), for example, found that incorporating 3D radiative transfer in the thermal spectral range results in systematically larger cooling and stronger organizational effects compared to simulations performed with 1D radiative transfer approximations. Similarly, Jakub and Mayer (2017) demonstrated that inter-column 3D radiative transfer in the solar spectral range can foster the formation of cloud streets that are not observed to the same extent in 1D simulations. And focusing on cloud characteristics, Veerman et al. (2020, 2022) and Tijhuis et al. (2024) showed that clouds in daytime LES runs coupled to 3D radiative transfer become thicker, grow larger in horizontal extent, and exhibit higher domain-averaged liquid water paths than their 1D-driven counterparts. Nonetheless, despite this demonstrated impact on cloud development, the implications of 3D radiative transfer on weather forecasts remain largely unexplored, primarily due to its high computational cost.

To address this issue, in recent years, considerable effort has been put into making 3D radiative transfer models computationally more feasible. For sub-grid-scale 3D effects, the Speedy Algorithm for Radiative Transfer through Cloud Sides (SPARTACUS; Schäfer et al., 2016; Hogan et al., 2016), for example, provides a fast method to capture 3D radiative effects at the resolutions of currently employed global atmospheric models. It does so by adding terms to the well-established two-stream approach that account for the radiative transport between cloudy and clear regions within a single heterogeneous model column. Meanwhile, significant work has also gone into the acceleration of inter-column radiative transport at subkilometer-scale horizontal resolutions, where model grid boxes can be gradually treated as homogeneous. Many such models simplify the expensive angular component of 3D radiative transfer calculations by considering only a discrete number of angles (e.g., Lovejoy et al., 1990; Gabriel et al., 1990; Davis et al., 1990). Most recently, the TenStream solver (Jakub and Mayer, 2015) built upon this idea. It is capable of calculating 3D radiative fluxes and heating rates in both the solar and thermal spectral ranges by extending the 1D two-stream formulation to ten streams, thereby allowing for horizontal transport of energy. Similarly, the neighboring column approximation (NCA; Klinger and Mayer, 2016, 2020) offers a fast analytical method for computing inter-column 3D heating rates in the thermal spectral range. To do so, it estimates cloud side effects by taking only the immediate neighbors of a specific grid box into account. Apart from these two approaches, significant progress has also been made in accelerating highly accurate 3D Monte Carlo solvers for the use in LES models, with Veerman et al. (2022), for example, speeding up the method by utilizing graphics processing units (GPUs). This allowed them to perform LES runs driven by a full Monte Carlo solver for the first time ever.

Despite these advances, 3D solvers remain too slow to be used operationally. For instance, the GPU-accelerated Monte Carlo solver of Veerman et al. (2022) is still at least 6.4 times slower than the two-stream model it was compared to — and that only when using just 32 photons per spectral band and model column. At this low photon count, however, results remain notably noisy, both in irradiances and heating rates. Achieving substantially more accurate results, with root-mean-square errors of $6.88\,\mathrm{W\,m^{-2}}$ in irradiances and $0.17\,\mathrm{K\,d^{-1}}$ in heating rates, requires increasing the photon count eightfold to 256 photons per spectral band and model column, which raises the computational cost to 18.5 times that of the two-stream model. This high computational burden continues to prevent the use of 3D solvers in operational weather forecasting, especially given that radiation is already called far less frequently than the dynamical core in NWP models.

To overcome these limitations, this thesis takes first steps toward a new, "dynamic" 3D radiative transfer model: the dynamic TenStream solver. Currently designed for subkilometer-scale horizontal resolutions, where model grid boxes can be treated as homogeneous, it builds upon the original TenStream solver while introducing a novel approach that is speeding up inter-column radiative transfer by treating radiation more like model dynamics. Specifically, inspired by how NWP models integrate their primitive equations over time, the solver does not recalculate radiation from scratch each time it is called but instead updates the radiative field based on the result from the previous radiation time step. Starting from this updated field, it then performs only the first few steps of an iterative scheme toward the new solution, thereby limiting interactions to neighboring grid boxes. This approach not only enables a significant acceleration of radiative transfer calculations, but also facilitates the otherwise tedious parallelization of 3D radiative transfer methods, which typically require information from all parts of the domain. The primary objective of this thesis is to demonstrate the feasibility of this new solver in computing heating rates and net surface irradiances at a noticeably faster speed than other 3D solvers, while also providing a significant improvement in terms of accuracy over currently employed 1D schemes. Additionally, this work investigates how coupling the new solver to model dynamics affects cloud evolution, assessing whether it produces clouds that more closely resemble those driven by full 3D radiation than those coupled to 1D radiation, while also exploring the mechanisms responsible for these differences.

To this end, the remainder of this thesis is structured as follows: Chapter 2 introduces the most important theoretical aspects of radiative transfer and its interaction with numerical weather prediction models. Chapter 3 then presents a detailed description of the new dynamic TenStream solver, beginning with an overview of the original TenStream model. Chapter 4 evaluates the performance of this new solver in terms of both speed and accuracy by comparing it to a conventional 1D solver, the original TenStream model, and a benchmark simulation provided by the 3D Monte Carlo solver MYSTIC (Mayer, 2009). In Chapter 5, the dynamic TenStream solver is then coupled to a large-eddy simulation model to examine its impact on cloud evolution compared to simulations driven by 1D radiation and the original TenStream solver, thereby also investigating links between the different radiative fields and the clouds they produce. Chapter 6 concludes the thesis with a summary and outlook.

Chapter 2

Theoretical background

The aim of this thesis is to develop a three-dimensional (3D) radiative transfer solver that is both fast and accurate, to evaluate its performance based on these criteria, and to explore its impact on clouds when coupled to a fully interactive numerical simulation. This chapter outlines the theoretical foundations required for these tasks. It begins with a brief overview of the basic principles of numerical weather prediction, with a particular focus on its coupling to radiation, before introducing key aspects of radiative transfer theory relevant to this work.

2.1 Numerical weather prediction

2.1.1 The primitive equations

Already in the early twentieth century, Bjerknes (1904) proposed the fundamental idea that still forms the backbone of modern numerical weather prediction (NWP). In his work, he identified a set of seven nonlinear partial differential equations — known as the *primitive equations* — which describe the conservation of momentum, mass, energy, and water in the atmosphere and together constitute an initial value problem. This means that if the initial state of the atmosphere at a given time is known with sufficient accuracy, these equations enable the prediction of the atmosphere's future evolution (Bjerknes, 2009). In particular, they describe the future development of seven key meteorological variables: the zonal (u), meridional (v) and vertical (w) components of the wind vector $\vec{v}$; temperature (T); pressure (p); density (ρ); and the water vapor mixing ratio (q_v), or related quantities. The corresponding equations can be expressed as follows:

$$\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -\frac{1}{\rho}\vec{\nabla} p - 2\left(\vec{\Omega} \times \vec{v}\right) + \vec{g}^* + \vec{F} \qquad \text{(momentum equation)}, \tag{2.1}$$

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\rho\vec{\nabla} \cdot \vec{v} \qquad \text{(continuity equation)}, \tag{2.2}$$

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{R_d\, T}{c_p\, p}\frac{\mathrm{d}p}{\mathrm{d}t} + \frac{\dot{q}}{c_p} \qquad \text{(thermodynamic equation)}, \tag{2.3}$$

$$\frac{\mathrm{d}q_v}{\mathrm{d}t} = \dot{q}_v \qquad \text{(water vapor equation),} \tag{2.4}$$

$$p = \rho R_d T \qquad \text{(ideal gas law).} \tag{2.5}$$

In addition to the variables already introduced, these equations contain time (t); Earth's rotation vector ($\vec{\Omega}$); effective gravity ($\vec{g}^*$); friction ($\vec{F}$); the specific gas constant for dry air (R_d); the specific heat capacity of air at constant pressure (c_p); and the sources and sinks of specific heat ($\dot{q}$) and water vapor ($\dot{q}_v$) in the atmosphere. In the form given above, these equations were adapted from Martin (2006) (momentum equation), Coiffier (2011) (thermodynamic equation), and Inness and Dorling (2012) (continuity equation, water vapor equation, and ideal gas law). Six of the equations are prognostic, meaning they describe the temporal evolution of their respective quantities, while the seventh — the ideal gas law — is diagnostic, as it does not involve a time derivative. To clarify the role of each primitive equation and highlight their coupling to radiation, let us break them down. Unless cited otherwise, the following information is taken from Martin (2006), Coiffier (2011) and Inness and Dorling (2012).

The momentum equation

Equation (2.1), also referred to as the *equations of motion*, is directly derived from Newton's second law. This law states that a body remains at rest or in uniform motion unless acted upon by an external force. When such a force is applied, the rate of change of momentum is equal to the sum of all forces acting on the body. In the atmosphere, these "bodies" are air parcels — imaginary volumes of air large enough for macroscopic quantities such as pressure to be meaningfully defined, yet small enough for these properties to be roughly uniform within them. Newton's second law enables the calculation of the acceleration of these air parcels — that is, the change in $\vec{v}$ over time — by summing all forces per unit mass acting upon them, namely

(a) *The pressure gradient force*: This is the primary force responsible for initiating motion of air in the atmosphere. It accelerates air parcels toward regions of lower pressure, i.e., in the opposite direction of $\vec{\nabla} p$. The acceleration $\vec{a}_p$ resulting from this force $\vec{F}_p$ is given by

$$\vec{a}_\mathrm{p} = \frac{\vec{F}_p}{m} = -\frac{1}{\rho}\vec{\nabla} p\,,$$

where m denotes the mass of the air parcel and ρ its density.

(b) *Gravity*: A fundamental force that pulls air parcels toward Earth's center.

(c) *Friction*: As air parcels interact with Earth's surface, momentum is dissipated through turbulent friction. This frictional effect is transmitted upward through the atmosphere via shear stress, decelerating air parcels and generating turbulent eddies that transport momentum between them, with additional momentum transport by convective eddies. The combined momentum change of these processes per unit mass is represented by $\vec{F}$.

Since Newton's second law assumes an inertial frame of reference, additional apparent forces must be considered to account for Earth's rotation. These include:

(a) *The Coriolis force*: This force accounts for the apparent deflection of air parcels due to Earth's rotation. A common analogy is a ball pushed toward the edge of a rotating disk. When viewed from above, the ball's motion does not follow a straight line because the counterclockwise-rotating disk deflects the ball to the right. In the atmosphere, a prominent example is the trade winds, which move air from high-pressure regions in the subtropics toward low-pressure areas near the equator. Due to Earth's rotation, these winds are deflected westward in the Northern Hemisphere. This deflection results from the interaction between Earth's rotation vector $\vec{\Omega}$, which points outward from Earth's center in the Northern Hemisphere, and the initial southward-facing wind vector $\vec{v}$. Together, they lead to an apparent westward acceleration of air parcels, which is described by

$$\vec{a}_{\mathrm{c}} = -2\left(\vec{\Omega} \times \vec{v}\right).$$

(b) *The centrifugal force*: This force accounts for the apparent outward acceleration of air parcels away from Earth's axis of rotation. It is expressed as

$$\vec{a}_{\mathrm{cf}} = -\vec{\Omega} \times \left(\vec{\Omega} \times \vec{r}\right),$$

where $\vec{r}$ is the position vector perpendicular to the axis of rotation, with a magnitude equal to the distance from a given point on Earth's surface to the rotational axis. Since the centrifugal force primarily weakens the gravitational pull, it is typically combined with gravity to form the effective gravitational acceleration $\vec{g}^*$.

The sum of these real and apparent forces ensures the conservation of momentum in the atmosphere and constitutes Eq. (2.1), which describes the future motion of air parcels and, consequently, the temporal evolution of the three components of the wind vector as

$$\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \vec{a}_{\mathrm{p}} + \vec{a}_{\mathrm{c}} + \vec{g}^* + \vec{F} = -\frac{1}{\rho}\vec{\nabla}p - 2\left(\vec{\Omega} \times \vec{v}\right) + \vec{g}^* + \vec{F}.$$

The continuity equation

Equation (2.2) ensures the conservation of mass in the atmosphere. It is typically derived for a stationary control volume, such as a fixed cube of air. For such a fixed volume, a net inflow of air increases its density, while a net outflow decreases it. Using the mass flux density $\vec{j}_m = \rho\vec{v}$, this relationship can be formulated as

$$\frac{\partial \rho}{\partial t} = -\vec{\nabla} \cdot \left(\rho \vec{v}\right). \tag{2.6}$$

This equation quantifies the change in density at a specific location, representing an *Eulerian* perspective. To derive the expression for a moving air parcel (a *Lagrangian* perspective), we must consider that the parcel's density is a function not only of time t but also of its time-dependent position $(x(t), y(t), z(t))$: $\rho(t, x(t), y(t), z(t))$. The total derivative of density with respect to time is then given by

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = \frac{\partial \rho}{\partial t} + \frac{\partial \rho}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial \rho}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial \rho}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}t} = \frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + w\frac{\partial \rho}{\partial z} = \frac{\partial \rho}{\partial t} + \vec{v} \cdot \vec{\nabla}\rho. \tag{2.7}$$

This relationship, which connects the local and total derivatives of a quantity (here, the density) with respect to time, generally applies to all total derivatives in Eqs. (2.1) to (2.4). It is a fundamental relationship that must be applied whenever switching from an Eulerian to a Lagrangian perspective, and vice versa. Rearranging Eq. (2.7) for the local change in density, $\frac{\partial\rho}{\partial t}$, shows that this local tendency is influenced both by the intrinsic changes experienced by the moving air parcels themselves and by the advection of the quantity by the wind, since

$$\frac{\partial\rho}{\partial t} = \frac{\mathrm{d}\rho}{\mathrm{d}t} - \vec{v}\cdot\vec{\nabla}\rho\,. \tag{2.8}$$

To better understand this relationship, let us consider the following limiting cases:

(a) *No wind* ($\vec{v} = 0$): In the absence of wind, the local change in density is solely determined by the intrinsic changes in density that the stationary air parcels experience, following the relationship

$$\frac{\partial\rho}{\partial t} = \frac{\mathrm{d}\rho}{\mathrm{d}t}\,.$$

(b) *Constant density in moving air parcels* ($\frac{\mathrm{d}\rho}{\mathrm{d}t} = 0$): When the density of the moving air parcels remains constant, the local change in density is driven purely by the advection of air parcels with different densities. For example, consider a purely westerly flow ($u > 0$, $v = w = 0$) where air to the west has a lower density ($\frac{\partial\rho}{\partial x} > 0$). In this case, lower-density air is advected, leading to a decrease in local density over time, which is described by

$$\frac{\partial\rho}{\partial t} = -u\frac{\partial\rho}{\partial x} < 0\,.$$

In general, Eq. (2.8) states that the local change in density results from both intrinsic changes within moving air parcels, and the advection of density by the wind. Using the identity

$$\vec{\nabla}\left(\rho\vec{v}\right) = \vec{v}\cdot\vec{\nabla}\rho + \rho\vec{\nabla}\cdot\vec{v}, \tag{2.9}$$

this relationship can now be used to formulate the conservation of mass for a moving air parcel. This continuity equation in Lagrangian form is exactly the expression provided in Eq. (2.2) and can be derived as follows:

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} \overset{(2.7)}{=} \frac{\partial\rho}{\partial t} + \vec{v}\cdot\vec{\nabla}\rho \overset{(2.6)}{=} -\vec{\nabla}\cdot\left(\rho\vec{v}\right) + \vec{v}\cdot\vec{\nabla}\rho \overset{(2.9)}{=} -\rho\vec{\nabla}\cdot\vec{v}\,.$$

The thermodynamic equation

Equation (2.3) expresses the conservation of energy in the atmosphere. It is based on the first law of thermodynamics, which states that the change in internal energy U of an air parcel is equal to the sum of the heat Q added to it and the work W done on it. Mathematically, this is expressed as

$$\Delta U = Q + W. \tag{2.10}$$

This definition, along with the other thermodynamic relationships used in this sub-subsection, follows Schroeder (2000). To understand how the first law of thermodynamics is connected to Eq. (2.3), we apply it to the atmospheric context. In this setting, the work done on an air parcel is typically compression work. An infinitesimal amount of this work is given by $\delta W = -p\,\mathrm{d}V$, where V represents the parcel's volume. For an infinitesimal change in internal energy $\mathrm{d}U$, we can use this expression along with the relationship $\mathrm{d}U = C_v\,\mathrm{d}T$, where C_v is the heat capacity at constant volume, to rewrite Eq. (2.10) as

$$C_v\,\mathrm{d}T = \delta Q - p\,\mathrm{d}V. \tag{2.11}$$

To express $\mathrm{d}V$ in this equation in terms of T and p, we apply the ideal gas law,

$$pV = nRT, \tag{2.12}$$

where n is the number of moles of gas and $R = 8.31\,\mathrm{J\,mol^{-1}\,K^{-1}}$ is the universal gas constant. Taking the total differential of this expression for an air parcel with a fixed number of moles and solving for $p\,\mathrm{d}V$ gives

$$\mathrm{d}(pV) = nR\,\mathrm{d}T = V\,\mathrm{d}p + p\,\mathrm{d}V \qquad \Rightarrow \quad p\,\mathrm{d}V = nR\,\mathrm{d}T - V\,\mathrm{d}p. \tag{2.13}$$

Substituting Eq. (2.13) into Eq. (2.11), and using the relationship $C_p = C_v + nR$ for the heat capacity at constant pressure, we obtain

$$C_v\,\mathrm{d}T = \delta Q - nR\,\mathrm{d}T + V\,\mathrm{d}p \qquad \Rightarrow \quad (C_v + nR)\,\mathrm{d}T = C_p\,\mathrm{d}T = \delta Q + V\,\mathrm{d}p. \tag{2.14}$$

Now, using another variant of the ideal gas law, $pV = mR_d T$, and the mass-specific quantities $c_p = \frac{C_p}{m}$ and $\delta q = \frac{\delta Q}{m}$, Eq. (2.14) can be rewritten as

$$c_p\,\mathrm{d}T = \delta q + \frac{V}{m}\,\mathrm{d}p = \delta q + \frac{R_d T}{p}\,\mathrm{d}p. \tag{2.15}$$

Finally, converting the total derivative to a time derivative and using $\dot{q} = \frac{\delta q}{\mathrm{d}t}$, we obtain the form of the first law of thermodynamics provided in Eq. (2.3), which is given by

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{R_d\,T}{c_p\,p}\frac{\mathrm{d}p}{\mathrm{d}t} + \frac{\dot{q}}{c_p}.$$

The first term on the right-hand side of this equation describes the temperature changes that an air parcel experiences due to compression and expansion. Since no heat is exchanged in such processes, they are referred to as *adiabatic*. The second term, on the other hand, mostly accounts for temperature changes due to *diabatic* processes, where heat is transferred to or from the air parcel. This includes sensible and latent heat fluxes, which arise from temperature differences (sensible heat) or phase changes such as condensation and evaporation (latent heat). The latter can also occur entirely within the air parcel, representing an example where $\dot{q} \neq 0$ and no heat is exchanged with the environment. Beyond that, sources and sinks of radiative energy are the primary contributors to this term, making it the direct link between the primitive equations and radiation. The radiative transfer models used to quantify these sources and sinks of radiative energy typically compute heating rates, which represent temperature changes over time. Hence, they account for the entirety of the second term and not just $\dot{q}$.

Oftentimes, Eq. (2.3) is expressed not in terms of temperature, but in terms of *potential temperature*

$$\theta = T \left(\frac{p_0}{p}\right)^{\frac{R_d}{c_p}} . \tag{2.16}$$

It represents the temperature an air parcel with temperature T and pressure p would have if it were brought dry adiabatically to a reference pressure of $p_0 = 1000\,\mathrm{hPa}$. Taking the total differential of this expression gives

$$\mathrm{d}\theta = \left(\frac{p_0}{p}\right)^{\frac{R_d}{c_p}} \mathrm{d}T + T p_0^{\frac{R_d}{c_p}} \left(-\frac{R_d}{c_p}\right) p^{-\frac{R_d}{c_p}-1} \mathrm{d}p = \frac{\theta}{T} \mathrm{d}T - \frac{\theta R_d}{c_p\, p} \mathrm{d}p . \tag{2.17}$$

When converting this total derivative to a time derivative and solving for $\frac{\mathrm{d}T}{\mathrm{d}t}$, we obtain

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{\theta}{T}\frac{\mathrm{d}T}{\mathrm{d}t} - \frac{\theta R_d}{c_p\, p}\frac{\mathrm{d}p}{\mathrm{d}t} \qquad \Rightarrow \qquad \frac{\mathrm{d}T}{\mathrm{d}t} = \frac{T}{\theta}\frac{\mathrm{d}\theta}{\mathrm{d}t} + \frac{R_d T}{c_p\, p}\frac{\mathrm{d}p}{\mathrm{d}t} . \tag{2.18}$$

Substituting Eq. (2.18) into Eq. (2.3) finally results in

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{\theta}{T}\frac{\dot{q}}{c_p} . \tag{2.19}$$

This is the thermodynamic equation in terms of potential temperature. Since θ is by definition conserved in adiabatic processes, Eq. (2.19) lacks the first term of Eq. (2.3), which accounts for temperature changes due to compression work. As a result, and provided that no phase changes occur inside the parcel, an air parcel's potential temperature evolves solely through diabatic processes, with radiative heating and cooling being the dominant contributors. This conservative property is also the main reason why the use of θ is often preferred in meteorology.

The water vapor equation

Equation (2.4) ensures the conservation of water in the atmosphere. It states that for any air parcel, the change in water vapor mixing ratio q_v — defined as the mass of water vapor per unit mass of dry air — is equal to the net sources (evaporation) and sinks (condensation) of water vapor within the parcel. This relationship can be expressed as

$$\frac{\mathrm{d}q_v}{\mathrm{d}t} = \dot{q}_v .$$

At first glance, this equation appears independent of the other primitive equations, as it does not explicitly contain shared variables such as temperature or wind speed. However, the net source term $\dot{q}_v$ is inherently linked to other atmospheric variables. To demonstrate this connection, we follow Wallace and Hobbs (2006b) and introduce additional moisture parameters that govern the sources and sinks of water vapor in Eq. (2.4). One such quantity is the *water vapor pressure* e, given by

$$e = \rho_v R_v T, \tag{2.20}$$

where ρ_v is the density of water vapor and $R_v = 461.51\,\mathrm{J\,kg^{-1}\,K^{-1}}$ is its specific gas constant. Physically, e represents the partial pressure that water vapor exerts within an air parcel. Thus, Eq. (2.20) is simply the ideal gas law applied to water vapor. Alongside the partial pressures of all other atmospheric gases, e contributes to the total pressure of the air parcel. To illustrate its role in governing the sources and sinks of q_v in Eq. (2.4), let us consider an initially dry air parcel placed above a water surface. As water molecules evaporate from the surface, this parcel gains water vapor, increasing e. At the same time, some water molecules condense back into the liquid phase. As long as evaporation exceeds condensation, the air remains unsaturated. However, once both processes reach equilibrium, the air is said to be saturated with respect to water. The corresponding *saturation vapor pressure* e_w^* depends on temperature and can be approximated using empirical formulas such as

$$e_w^* = 6.112\,\mathrm{hPa}\exp\left(\frac{17.62\,\vartheta}{243.12\,°\mathrm{C} + \vartheta}\right), \tag{2.21}$$

where ϑ is the temperature in degrees Celsius (WMO, 2025). Since e_w^* increases exponentially with temperature, cooling an air parcel lowers its saturation vapor pressure, while warming raises it. This directly influences the *relative humidity* of the air parcel, defined as

$$\mathrm{RH} = \frac{e}{e_w^*}. \tag{2.22}$$

When cooled, e_w^* decreases much more rapidly than e does, causing the relative humidity to rise. Once it reaches 100%, condensation occurs, providing a net sink of water vapor for Eq. (2.4). Conversely, when a saturated air parcel warms, its saturation vapor pressure increases exponentially, leading to evaporation and creating a net source of water vapor. This dependence on temperature illustrates how Eq. (2.4) is indeed fundamentally coupled to the other primitive equations.

The ideal gas law

The final primitive equation is the *ideal gas law*, which states that the atmosphere behaves as an ideal gas. Starting from its usual form $p\,V = m\,R_d\,T$, where $R_d = 287\,\mathrm{J\,kg^{-1}\,K^{-1}}$ is the specific gas constant for dry air, the density $\rho = \frac{m}{V}$ is introduced to obtain the form provided in Eq. (2.5), which is given by

$$p = \rho\,R_d\,T\,.$$

Unlike the other primitive equations, Eq. (2.5) does not describe the temporal evolution of one of the seven key atmospheric variables. Instead, it is a diagnostic equation connecting the pressure, density and temperature in an air parcel.

2.1.2 The basic principle of numerical weather prediction

With the exception of the ideal gas law, all primitive equations are prognostic, meaning they describe the temporal evolution of one of the seven key atmospheric variables. Thus, given the initial state of the atmosphere at a given time, these equations enable the prediction of its future evolution. However, they do not form a fully closed system of equations, because the sources and sinks of momentum, heat, and water vapor in Eqs. (2.1), (2.3), and (2.4) must be specified in advance before starting with the forward integration (Inness and Dorling, 2012). Since the heat source term is primarily governed by radiative heating and cooling, this once again highlights the crucial role of radiation in weather prediction: Starting from its contribution to the heat source term, it directly influences the temperature field. Through the ideal gas law, these temperature tendencies then modify the pressure field, creating pressure gradients that subsequently drive atmospheric motion via Eq. (2.1). Some of this motion results in the upward transport of air masses to lower-pressure regions, where they expand and cool according to Eq. (2.3). As they cool, their saturation vapor pressure, described by Eq. (2.21), decreases, eventually leading to saturation and the condensation of water vapor into clouds. This is just one example of how radiation drives atmospheric dynamics and how the primitive equations collectively describe the resulting atmospheric evolution. Despite this theoretical framework, however, weather forecasts remain imperfect. This is due to two fundamental limitations:

1. As previously noted, the primitive equations constitute an initial value problem, meaning that accurately specifying the initial atmospheric state is crucial for producing reliable forecasts (Inness and Dorling, 2012). This becomes even more critical due to the chaotic nature of the primitive equations, where even small differences in the initial conditions can lead to vastly different future atmospheric states — a phenomenon often referred to as the butterfly effect (Lorenz, 1963; Motter and Campbell, 2013). However, the state of the atmosphere can never be captured in full detail, as observations do not provide complete coverage, and measurement instruments always have finite accuracy. To mitigate these limitations, modern weather forecasting relies heavily on data assimilation, where model output is combined with observational data to construct the best possible estimate of the initial conditions (Inness and Dorling, 2012). Yet, no matter how sophisticated this process is, initial conditions will always contain some degree of error, inevitably causing forecasts to diverge from reality over time. This imposes an inherent limit on the predictability of the weather, with the study of Selz et al. (2022) indicating a maximum forecast horizon of approximately two weeks.

2. Recalling that

$$\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \frac{\partial \vec{v}}{\partial t} + \left(\vec{v} \cdot \vec{\nabla}\right) \vec{v}, \tag{2.23}$$

   the equations of motion contain terms that are quadratic in terms of their prognostic variables. Due to this nonlinearity, they have no known analytical solution and must be solved numerically (Holton and Hakim, 2013), for example, by dividing the atmosphere into grid boxes and discretizing the equations in both space and time. In this case, the grid spacing determines which processes can be explicitly resolved and which cannot. For instance, global atmospheric models such as the high-resolution deterministic forecasts of the European Centre for Medium-Range Weather Forecasts (ECMWF), which operate at a horizontal resolution of 9 km (Hogan and Bozzo, 2018), lack the resolution needed to explicitly simulate convective clouds, which requires a grid spacing of 4 km or less (Prein et al., 2015). The effects of these convective clouds and other sub-grid-scale processes such as turbulence on the resolved scales must therefore be accounted for by so-called *parameterizations* (Bauer et al., 2015). Since many of the underlying processes are highly complex and not yet fully understood, they are generally represented using simplified models. They then enter the primitive equations via the source terms of turbulent friction ($\vec{F}$), heat ($\dot{q}$), and water vapor ($\dot{q}_v$) in the momentum, thermodynamic, and water vapor equations. One of the key processes that must be parameterized is radiative transfer, which is the main focus of this thesis and will be discussed in detail in Sect. 2.2. Other commonly parameterized processes include convection, cloud microphysics, and surface and boundary layer processes. To get an idea of these processes and the spatial scales they act at, let us quickly break them down. Unless cited otherwise, the following descriptions are adapted from Inness and Dorling (2012):

- *Convection* accounts for the vertical transport of heat, water vapor, and momentum within convective cells, as well as the formation of clouds and precipitation associated with them. The corresponding parameterizations aim to formulate the statistical effects of these processes rather than explicitly predicting individual clouds (Arakawa, 2004). Since convective processes typically act up to the kilometer-scale, models with grid spacings of tens of kilometers or more must parameterize the entire convection process. At kilometer-scale horizontal resolutions, however, parts of the convective cells become resolved, introducing additional complexity regarding which processes still require parameterization and which can be explicitly simulated.
- *Cloud microphysics* describes the formation and growth of cloud droplets and ice crystals on the scale of individual droplets and ice crystals. Since these scales are much smaller than the grid spacing of nearly all current numerical models, the corresponding parameterizations are largely the same for global atmospheric and regional high-resolution models.
- *Surface and boundary layer processes* account for two main effects: First, the dissipation of momentum due to turbulent friction, which must be parameterized because models cannot explicitly resolve every hill, building, or vegetation element at Earth's surface. Second, the transport of momentum, heat, and moisture by turbulent and convective eddies within the boundary layer. Both effects operate across a wide range of spatial scales, making their parameterization strongly resolution-dependent.

Although necessary to specify the sources and sinks of momentum, heat, and water vapor in Eqs. (2.1), (2.3), and (2.4), all of these parameterizations share the problem that

Table 2.1: Overview of atmospheric model types, including their typical grid spacings and the extent to which they explicitly resolve convection and turbulence.

Model type	Abbrev.	Grid spacing	Convection resolving	Turbulence resolving
General circulation model	GCM	$\mathcal{O}(10\,\mathrm{km})$	No	No
Regional or limited-area model	LAM	$\mathcal{O}(1\,\mathrm{km})$	Partially	No
Cloud-resolving model	CRM	$\mathcal{O}(100\,\mathrm{m})$	Yes	No
Large-eddy simulation	LES	$\mathcal{O}(10\,\mathrm{m})$	Yes	Partially
Direct numerical simulation	DNS	$\mathcal{O}(1\,\mathrm{mm})$	Yes	Yes

they approximate the influence of unresolved processes on the resolved scales using simplified models, thereby introducing additional uncertainties into numerical weather forecasts.

Taken together, these two challenges — the chaotic nature of the primitive equations and the fact that they can only be solved numerically — highlight the two fundamental steps required for numerical weather prediction: first, determining the initial atmospheric state as accurately as possible; and second, integrating the primitive equations forward in time to predict the atmosphere's future evolution, thereby accounting for the effects of unresolved processes on the resolved scales. The basic principle of numerical weather prediction can therefore be summarized as follows:

1. *Determination of the initial atmospheric state:* First, the initial state of the atmosphere must be determined as accurately as possible. To achieve this, data assimilation methods combine observational data with model output to construct the best possible estimate of the current atmospheric state.

2. *Numerical integration of the primitive equations:* Once the initial state of the atmosphere is determined, the primitive equations must be integrated forward in time to predict its future evolution. Since they have no known analytical solution, this integration is performed numerically. However, the spatial discretization required for this numerical solution prevents certain atmospheric processes from being explicitly resolved. The effects of these sub-grid-scale processes on the resolved scales must therefore be accounted for through parameterizations, which provide the necessary source terms to the primitive equations, allowing the system to be advanced despite not being fully closed.

2.1.3 Types of atmospheric models

As noted in the previous subsection, the spatial resolution of an atmospheric model determines which processes can be explicitly resolved and which must be parameterized. Consequently, based on their spatial resolution, these models can be categorized into different types. Table 2.1 provides an overview of this categorization. Based on this table, let us briefly examine each model type and clarify how the large-eddy simulations used in this thesis are positioned within them:

- *General circulation models* (GCMs) solve the primitive equations on a global scale. They typically operate at horizontal grid spacings of tens of kilometers. For instance, the operational models of the ECMWF and the Deutscher Wetterdienst (DWD) currently use grid spacings of 9 km (Hogan and Bozzo, 2018) and 13 km (Reinert et al., 2025, p. 19), respectively. At these resolutions, neither convection nor turbulence can be explicitly resolved, requiring both of these processes to be parameterized.

- *Regional and limited-area models* (LAMs) provide higher-resolution forecasts over specific regions. They are typically nested within larger-scale models, which provide their initial and boundary conditions. LAMs refine the coarse information from these models by generating smaller-scale atmospheric features at their higher resolutions (de Elía et al., 2002), enabling more detailed predictions for specific regions such as individual countries. They typically operate at kilometer-scale horizontal resolutions, with the DWD ICON-D2 model for example employing a 2.1 km grid spacing (DWD, 2025). At this resolution, these models are considered convection-permitting, meaning that some aspects of the convective process can be explicitly resolved rather than parameterized.

- *Cloud-resolving models* (CRMs) aim to explicitly resolve cumulus convection, eliminating the need for convection parameterizations. Instead, clouds are simulated using more sophisticated microphysics parameterizations that account for the formation and growth of cloud droplets and ice crystals on the microphysical scale. CRMs typically operate at subkilometer-scale resolutions, with horizontal grid spacings on the order of hundreds of meters. Recently, advances in computing power have enabled the development of global cloud-resolving models (GCRMs), which, for research applications, enable the use of cloud-resolving simulations on the global scale (Satoh et al., 2019).

- *Large-eddy simulations* (LESs) provide an explicit representation of large-scale turbulent motions, while a turbulence model parameterizes the effect of unresolved small-scale eddies (Mason, 1994). Hence, LES models resolve convection and parts of the turbulent flow, making them a valuable tool for studying transport processes in the atmospheric boundary layer (Stoll et al., 2020). LES models typically operate at horizontal grid spacings ranging from 10 m to 100 m (Satoh et al., 2019). In Chapter 5 of this thesis, the LES model PALM (Raasch and Schröter, 2001; Maronga et al., 2015, 2020) is employed to investigate how the radiative transfer solver developed in this work affects cloud development. A LES model operated at a horizontal resolution of 100 m was chosen for this study, as this spatial scale allowed sub-grid-scale cloud variability to be neglected, and grid cells to be treated as either fully cloudy or cloud-free.

- *Direct numerical simulations* (DNSs) eventually solve the full equations of motion by explicitly resolving the entire turbulent flow (Moin and Mahesh, 1998). To achieve this, they must capture even the smallest turbulence scales characterized by the Kolmogorov length scale (Remmler et al., 2014), which is typically on the order of millimeters for atmospheric flows (Vallis, 2017), making these models computationally extremely expensive. As a result, DNS models are primarily used for fundamental turbulence research, with recent advances in computing power even enabling their use in studying individual clouds. Simulations of larger atmospheric regions, however, remain computationally infeasible with this type of model (Satoh et al., 2019).

2.1.4 Convective cloud formation

Before concluding this section on numerical weather prediction, let us examine the formation of convective clouds in numerical models that are capable of explicitly resolving this process. This discussion provides essential background information for Chapter 5, where a simulation is set up in which shallow cumulus clouds develop over the course of the day. Unless stated otherwise, the content in this section is adapted from Wallace and Hobbs (2006b).

As discussed earlier, convection begins with the warming of air masses near Earth's surface. Since small-scale pressure differences adjust almost instantaneously at the speed of sound, the pressure of this warmed air can be assumed to be in equilibrium with the surrounding air. According to Eq. (2.5), this implies that the density ρ' of the warmed air parcels is lower than the density ρ of the surrounding air. As a result, the warmed air parcels experience an upward acceleration, which is the result of two counteracting forces:

(a) *The gravitational force F_g*, which pulls the warmed air parcels with mass m' and volume V' down toward the ground and is given by

$$F_g = -m' g = -\rho' V' g\,. \tag{2.24}$$

(b) *The buoyant force F_b*, which, according to Archimedes' principle, acts upward with a magnitude equal to the gravitational force acting on the ambient air displaced by the parcels. It can be expressed as

$$F_b = \rho\, V' g\,. \tag{2.25}$$

The primes in these equations indicate that the corresponding quantities refer to the air parcel rather than to the surrounding air. Summing both forces and applying Newton's second law yields the vertical acceleration of the air parcels, which is given by

$$m' \frac{\mathrm{d}w}{\mathrm{d}t} = \rho' V' \frac{\mathrm{d}w}{\mathrm{d}t} = F_b + F_g = (\rho - \rho')\, V' g \qquad \Rightarrow \qquad \frac{\mathrm{d}w}{\mathrm{d}t} = \frac{\rho - \rho'}{\rho'} g\,. \tag{2.26}$$

Using the ideal gas law ($\rho = \frac{p}{R_d T}$) and assuming pressure equilibrium ($p = p'$), this acceleration can be rewritten in terms of the temperature difference between the air parcels (T') and the surrounding air (T), so that Eq. (2.26) becomes

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \frac{\rho - \rho'}{\rho'} g = \frac{\frac{1}{T} - \frac{1}{T'}}{\frac{1}{T'}} g = \frac{T' - T}{T} g\,. \tag{2.27}$$

This demonstrates that the upward motion of the warmed air parcels depends on both $T'(z)$ and $T(z)$. As long as the parcels remain warmer than their environment ($T' > T$), they continue to ascend. However, once their temperature equals that of the surrounding air ($T' = T$), the parcels reach a state of neutral buoyancy and eventually stop rising. Thus, to understand the vertical motion of warmed air masses — and consequently the formation of convective clouds — it is necessary to examine both the temperature evolution within the rising air parcels and the ambient temperature profile.

Temperature changes in a rising air parcel

Let us start with the temperature evolution in rising air parcels. Since vertical air motions occur on relatively short timescales, these movements are generally treated as adiabatic processes, meaning that no heat is exchanged with the environment. In such cases, the potential temperature is a particularly useful quantity, as it is by definition conserved in adiabatic processes, provided that no phase changes occur in the air parcel. To demonstrate this, we recall Eq. (2.17) and express it as a derivative with respect to z, which yields

$$\frac{\mathrm{d}\theta'}{\mathrm{d}z} = \frac{\theta'}{T'}\frac{\mathrm{d}T'}{\mathrm{d}z} - \frac{\theta' R_d}{c_p\, p}\frac{\mathrm{d}p}{\mathrm{d}z}\,. \tag{2.28}$$

To obtain an expression for $\frac{\mathrm{d}T'}{\mathrm{d}z}$, we convert Eq. (2.15) to a derivative with respect to z and assume a purely adiabatic process without any phase changes in the air parcel, i.e., $\delta q = 0$, which results in

$$\frac{\mathrm{d}T'}{\mathrm{d}z} = \frac{R_d\, T'}{c_p\, p}\frac{\mathrm{d}p}{\mathrm{d}z}\,. \tag{2.29}$$

Substituting Eq. (2.29) into Eq. (2.28) then shows that

$$\frac{\mathrm{d}\theta'}{\mathrm{d}z} = \frac{\theta'}{T'}\frac{R_d\, T'}{c_p\, p}\frac{\mathrm{d}p}{\mathrm{d}z} - \frac{\theta' R_d}{c_p\, p}\frac{\mathrm{d}p}{\mathrm{d}z} = 0\,, \tag{2.30}$$

as expected from the conservation of potential temperature in adiabatic processes. The actual temperature of a rising air parcel, however, does change with height. To determine this change, we can use the fact that, to first order, the atmosphere is in hydrostatic balance — an equilibrium between the downward-facing gravitational force and the upward-facing pressure gradient force, which can be expressed as

$$-\frac{1}{\rho}\frac{\partial p}{\partial z} - g = 0 \qquad \Rightarrow \qquad \frac{\partial p}{\partial z} = -\rho\, g\,. \tag{2.31}$$

Note that this equation describes the pressure change with height in the surrounding air, whereas the rising air parcel itself is certainly not in a static state. However, since pressure differences between the parcel and its environment adjust almost instantaneously, we can apply this pressure change to the air parcel as well. This allows us to substitute Eq. (2.31) into Eq. (2.29), thereby also applying the relation $\frac{1}{\rho'} = \frac{R_d\, T'}{p}$, giving

$$\frac{\mathrm{d}T'}{\mathrm{d}z} = \frac{R_d\, T'}{c_p\, p}\frac{\mathrm{d}p}{\mathrm{d}z} = -\frac{\rho}{\rho'}\frac{g}{c_p}\,. \tag{2.32}$$

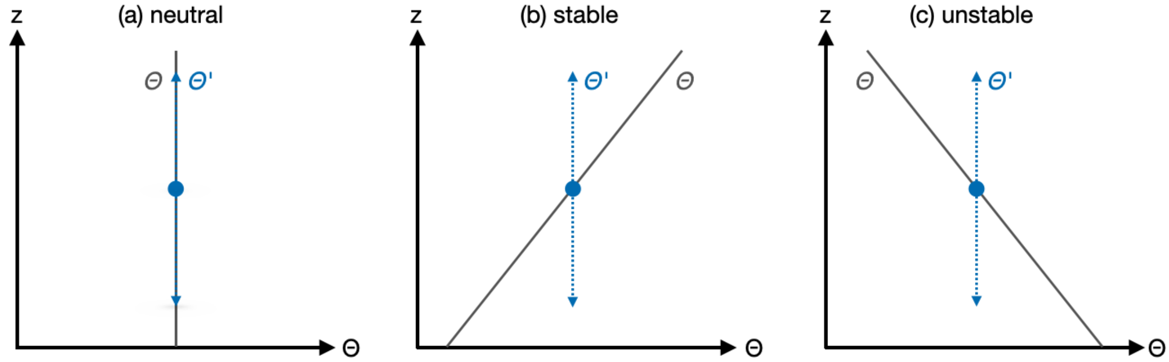

Figure 2.1: Schematic illustration of the three atmospheric stability regimes. An example of the potential temperature profile for each case is shown in gray. Using these profiles, the plots depict how the potential temperature of a vertically displaced air parcel, initially in perfect equilibrium with its surroundings, compares to that of the environment. To this end, the blue dot represents the air parcel in its initial state, while the dotted blue arrows indicate its potential temperature evolution when displaced upward or downward.

When assuming that the density of the rising air is not much different from that of the surrounding air, i.e., $\rho \approx \rho'$, this expression can be further simplified to

$$\frac{\mathrm{d}T'}{\mathrm{d}z} = -\frac{g}{c_p} = -9.8\,\mathrm{K\,km^{-1}}. \tag{2.33}$$

In the last step, we used $g = 9.81\,\mathrm{m\,s^{-1}}$ and $c_p = 1004\,\mathrm{J\,kg^{-1}\,K^{-1}}$. Equation (2.33) is known as the *dry adiabatic lapse rate*. It describes the temperature change in an air parcel lifted without condensation. However, as the air parcel rises and cools, it eventually reaches its saturation vapor pressure, leading to condensation. This process releases latent heat, which partially counteracts the cooling due to expansion, resulting in a reduced lapse rate known as the *saturated adiabatic lapse rate*. The magnitude of this lapse rate depends on the parcel's moisture content and is always less negative than the dry adiabatic lapse rate. In the limit of all the water vapor condensed out, it converges toward the dry adiabatic lapse rate again. Apart from these considerations, it is important to note that Eq. (2.33) remains an approximation that assumes small density differences between the parcel and its surroundings. The more fundamental result is given by Eq. (2.30), which states that the potential temperature is conserved during dry adiabatic lifting.

Influence of atmospheric stability on rising air parcels

Having examined the temperature evolution within a rising air parcel, we can now turn to the second key factor governing the vertical motion of air masses: the temperature profile of the surrounding air. To explore this, we consider an air parcel initially in perfect equilibrium with its environment, meaning it has the same temperature and pressure as the surrounding air. We then analyze its response to small vertical displacements. Figure 2.1 illustrates this process for three different ambient potential temperature profiles. Regardless of the profile, Eq. (2.30) states that the potential temperature is conserved within the air parcel. Consequently, in all three cases, the parcel's potential temperature evolution follows the same trajectory, as visualized by the blue dotted arrows in each panel. The parcel's response to these displacements, however, depends strongly on the potential temperature profile of the surrounding atmosphere. A crucial point to recall here is that, due to the pressure equilibrium between the air parcel and its environment, differences in potential temperature correspond directly to differences in actual temperature at the same height. Specifically, if the air parcel has a higher potential temperature than its surroundings ($\theta' > \theta$), then its actual temperature (T') will also be higher than the surrounding temperature (T). Using this relationship and Fig. 2.1, we can identify the three different atmospheric stability regimes:

(a) *Neutral stratification* ($\frac{\partial\theta}{\partial z} = 0$): The atmosphere is said to be neutrally stratified when the potential temperature remains constant with height. In this case, the air parcel's potential temperature — and consequently, its actual temperature — stays equal to that of the surrounding air at all heights. According to Eq. (2.27), this means the parcel experiences no net acceleration; it neither rises further nor returns to its original position, instead remaining in equilibrium at its new height.

(b) *Stable stratification* ($\frac{\partial\theta}{\partial z} > 0$): The atmosphere is said to be stably stratified when the potential temperature increases with height. A lifted air parcel in such an atmosphere remains cooler and denser than its surroundings ($\theta' < \theta$), resulting in a downward buoyant force that drives it back toward its original position. Similarly, if displaced downward, it becomes warmer and less dense, generating an upward buoyant force that pushes it back up. In both cases, any displacement is counteracted, ensuring that the air parcel returns to equilibrium.

(c) *Unstable stratification* ($\frac{\partial\theta}{\partial z} < 0$): The atmosphere is said to be unstably stratified when the potential temperature decreases with height. Here, a lifted air parcel remains warmer and less dense than its surroundings ($\theta' > \theta$), leading to continued upward acceleration. Conversely, if displaced downward, it stays cooler and denser, resulting in further downward acceleration. This means that any initial displacement is amplified, moving the parcel progressively away from its original height.

These stability regimes, combined with the previously derived temperature evolution inside a rising air parcel, allow us to describe the vertical motion of a warmed air parcel at the surface. From Eq. (2.27), we know that such a warmed air parcel experiences an upward buoyant force. In addition, Eq. (2.30) states that its potential temperature remains constant during ascent. As a result, the upward motion of the parcel is primarily governed by the stability of the surrounding atmosphere. In neutrally or unstably stratified atmospheric regions, the potential temperature of the surrounding air remains constantly lower or even decreases relative to that of the warmed air parcel, allowing it to rise continuously. In stably stratified atmospheric regions, on the other hand, the potential temperature increases with height. Consequently, the rising air parcel eventually reaches a level where its potential temperature equals that of the surrounding air, bringing its ascent to an end. On average, the atmosphere is in such a stably stratified state (Andrews, 2010), with an average temperature decrease of about $-6.5\,\mathrm{K\,km^{-1}}$ in the troposphere (Wallace and Hobbs, 2006a).

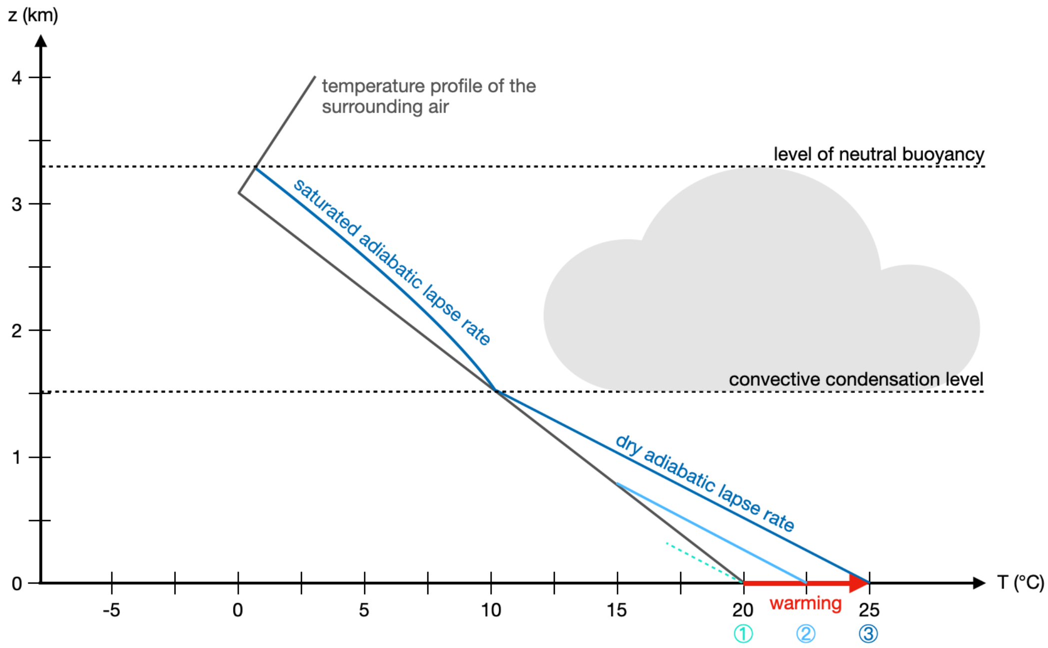

Figure 2.2: Schematic illustration of convective cloud formation in an initially stably stratified atmosphere. The gray line represents the initial temperature profile, where no convection occurs. As near-surface air warms (red arrow), the formation of a convective cloud is illustrated in three steps, each represented by a distinct shade of blue and labeled with an encircled number:

1. *Initial stable state:* Before warming occurs, the air at the surface is in equilibrium with its surroundings and experiences no buoyant force. If displaced upward, its temperature follows the dry adiabatic lapse rate, as indicated by the dotted turquoise line, making it colder and denser than the surrounding air. The resulting negative buoyancy forces the air back to the surface, preventing convection.

2. *Onset of convection:* As the surface air warms, it becomes buoyant and begins to rise. Since the air is initially assumed to be unsaturated, it follows the dry adiabatic lapse rate during this ascent. The warmer the surface air gets, the higher it rises before reaching a level where its temperature equals that of the surrounding air, gradually deepening the convective layer.

3. *Cloud formation:* Once surface warming reaches a critical convective temperature (25°C in this example), the rising air reaches the convective condensation level (CCL), where it becomes saturated. From this point onward, condensation occurs and a cloud starts to form. In this process, latent heat is released, altering the parcel's lapse rate to the saturated adiabatic lapse rate, which is less steep than the dry adiabatic lapse rate. The air continues to rise until reaching the level of neutral buoyancy (LNB), where its temperature matches that of the surrounding air, marking the cloud top.

Formation and quantitative description of convective clouds

Building on this understanding, we can now examine the formation of convective clouds in the atmosphere. Figure 2.2 presents a schematic illustration of this process. Just as in the global average, the temperature profile in this example (shown in gray) features a lapse rate of $-6.5\,\mathrm{K\,km^{-1}}$, with the temperature decreasing from 20°C at the surface to 0°C slightly above 3 km height. Above this level, a capping inversion is introduced, with the temperature increasing at a rate of $3\,\mathrm{K\,km^{-1}}$, to promote shallow rather than deep convection. Based on this temperature profile, the formation of a convective cloud from initially unsaturated air parcels at the ground is illustrated in three steps, each visualized in a different shade of blue:

1. **Initial stable state**
   In the first stage, the surface air has not yet been heated and is still in equilibrium with its surroundings. If lifted, it would cool at approximately $-9.8\,\mathrm{K\,km^{-1}}$, following the dry adiabatic lapse rate quantified in Eq. (2.33). This cooling is visualized by the dotted turquoise line in Fig. 2.2. Along this path, the lifted air becomes progressively cooler and denser than the surrounding air, which, according to Eq. (2.27), results in a downward buoyant force that returns the parcel back to its original position.

2. **Onset of convection**
   However, as solar heating warms the near-surface air masses, they become warmer than the surrounding environment. This initiates convection, as the warmed air parcels now experience an upward buoyant force. As they rise, the initially unsaturated air masses follow the dry adiabatic lapse rate, cooling more rapidly with height than the surrounding air, which cools at only $-6.5\,\mathrm{K\,km^{-1}}$ in this example. Consequently, the rising air eventually reaches a height where its temperature matches that of the environment, marking the upper boundary of the convective layer.

3. **Cloud formation**
   As surface warming continues, the rising air parcels reach progressively greater heights, gradually deepening the convective layer, until eventually, the *convective temperature* (see, e.g., Peppler and Lamb, 1989) of 25°C in this example is reached. At this point, the rising air cools enough to reach a height where its water vapor pressure equals the saturation vapor pressure, leading to condensation. This height is referred to as the *convective condensation level* (CCL; see, e.g., Peppler and Lamb, 1989) and marks the lower end of the cloud that now begins to form. Because condensation releases latent heat, the rising air now cools at the saturated adiabatic lapse rate instead of the dry adiabatic lapse rate, resulting in a much slower cooling rate. This allows the rising air to reach significantly greater heights than before. Eventually, however, the temperature of the rising air masses equals that of the surrounding air again. This height is also referred to as the *level of neutral buoyancy* (LNB; see, e.g., Wang et al., 2020). The difference in height between this LNB and the CCL determines the vertical extent of the convective cloud that has formed.

In summary, convective clouds thus form when the near-surface air warms sufficiently for rising air parcels to exceed the convective condensation level, allowing their water vapor to condense into clouds. To describe the structure and evolution of these clouds, let us now finally introduce some key quantities that will be used throughout the remainder of this thesis. Just like in the example presented in Fig. 2.2, we will focus on shallow convective clouds for that, although the radiative transfer solver introduced in this work can theoretically handle deep convection as well. These shallow cumulus clouds typically remain below the 0°C isotherm and consist solely of liquid water, which is why they are also referred to as warm clouds (Wallace and

Hobbs, 2006c). In numerical models, the microphysical structure of these clouds is represented, among other things, by the spatial distribution of the cloud water mixing ratio across the grid boxes occupied by the cloud. Similar to the water vapor mixing ratio, this *cloud water mixing ratio* q_c is defined as the mass m_w of liquid water per unit mass m_d of dry air:

$$q_c = \frac{m_w}{m_d}\,. \tag{2.34}$$

In addition to that, the *total water mixing ratio* q specifies the combined mass of water in all three phases (water vapor, liquid water, and ice) per unit mass of dry air. Furthermore, and as an alternative to q_c, the *liquid water content* (LWC) is often used. It expresses the mass of liquid water per unit volume V of air (Wallace and Hobbs, 2006c) and is therefore defined as

$$\mathrm{LWC} = \frac{m_w}{V}\,. \tag{2.35}$$

The liquid water content and q_c are related via the air density ρ, following the relationship

$$\mathrm{LWC} = \frac{m_w}{V} = \rho\,\frac{m_w}{m_d} = \rho\, q_c\,. \tag{2.36}$$

The vertically integrated liquid water content finally defines the *liquid water path*

$$\mathrm{LWP} = \int_0^\infty \mathrm{LWC}(z)\,\mathrm{d}z\,. \tag{2.37}$$

It quantifies the total column-integrated liquid water per unit area and is therefore strongly dependent on cloud thickness. Together with other, purely geometric quantities such as cloud depth, these measures will be used to characterize cloud fields driven by different radiative transfer models in Chapter 5, as well as to describe any cloud field considered in the remainder of this thesis.

2.2 Radiative transfer

The importance of radiation for atmospheric dynamics has already been emphasized in various parts of this work. In the introduction, for example, radiation was identified as the main driver of both large-scale atmospheric motions, such as the global atmospheric circulation, and smaller-scale processes like convection. While the former results from the differential heating of Earth at different latitudes, the latter is initiated by the warming of air masses near Earth's surface. It was consequently pointed out that a common characteristic of all these processes is their initiation through radiation-induced temperature changes, with sources of radiative energy heating air masses and sinks cooling them. In the preceding section on numerical weather prediction, it was then shown that these sources and sinks of radiative energy are also exactly the means by which radiation enters the primitive equations — the set of nonlinear partial differential equations describing the atmosphere's future evolution. When discussing their forward integration in NWP models, it was further noted that radiation is among the processes not explicitly resolved by these models, requiring parameterization instead. Since radiation is the main focus of this thesis, the following section now provides an overview of key aspects of radiative transfer theory and its parameterization in NWP models.

2.2.1 Electromagnetic radiation

To describe radiative transfer in the atmosphere, we first introduce fundamental properties of electromagnetic radiation along with key radiative quantities required to characterize it. Unless stated otherwise, the content of this subsection is adapted from Petty (2006a) and Wallace et al. (2006).

Radiative transfer generally describes the transport of radiative energy through a medium. In classical physics, this process is modeled using electromagnetic waves, which consist of oscillating electric and magnetic fields oriented perpendicular to each other and propagating in a direction perpendicular to both fields. In a vacuum, these waves propagate at the speed of light, $c = 299792458\,\mathrm{m\,s^{-1}}$, whereas in a medium with refractive index n, their speed is reduced to $\frac{c}{n}$. Furthermore, the frequency f of the oscillating waves determines their wavelength λ, following the relationship

$$\lambda = \frac{c}{f}. \tag{2.38}$$

However, not all properties of electromagnetic radiation can be explained through its wave nature alone. A famous example is the photoelectric effect, which describes the emission of electrons from a material exposed to electromagnetic radiation. According to classical wave theory, this emission should depend solely on the intensity of the incoming light. An electron should therefore be dislodged once enough energy has accumulated to free it, with sufficiently dim light causing a delayed emission. However, experimental results show that electron emission depends not on light intensity but solely on frequency, with electrons dislodged only if a certain threshold frequency is exceeded. If the frequency of the incoming radiation is below this threshold, no electrons are emitted, regardless of intensity. Einstein (1905) famously demonstrated that this behavior can be explained by interpreting light not as a wave but rather as discrete packets of energy — photons. The *radiative energy* Q of each of these photons is given by

$$Q = hf = h\frac{c}{\lambda}, \tag{2.39}$$

where $h = 6.62607015 \times 10^{-34}\,\mathrm{J\,s}$ is Planck's constant, a fundamental physical constant. Adapting this particle-like behavior, individual photons must exceed a certain frequency to provide enough energy to dislodge an electron, thus explaining the photoelectric effect. In this thesis, we primarily adopt this quantum perspective, treating electromagnetic radiation as an ensemble of photons.

Using this perspective, we can now introduce the quantities needed to quantitatively describe electromagnetic radiation in the atmosphere. To this end, we consider a surface, such as Earth's surface or any imaginary surface within Earth's atmosphere, exposed to electromagnetic radiation. The total amount of energy received by this surface over a given time can theoretically be determined by summing the energy Q of all photons passing through it during that period. Yet, to capture the radiative field in full detail, one must also account for variations in incoming radiation over time, such as changes in the Sun's incidence angle.

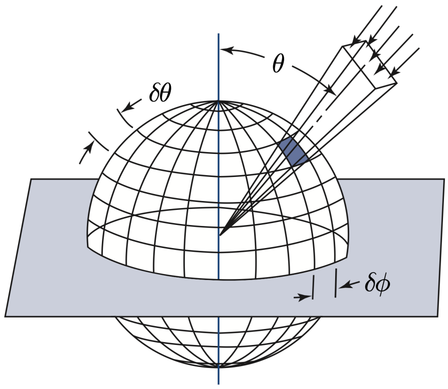

Figure 2.3: Schematic illustration of spectral radiance, which is defined as the amount of radiative energy per unit time and unit wavelength interval arriving from a given direction and passing through a unit area normal to that direction (shaded in dark purple). The direction of the incoming radiation is specified by the zenith angle θ and azimuth angle ϕ, with the corresponding infinitesimal solid angle spanned by the differential angle elements $\delta\theta$ and $\delta\phi$. Reprinted from Fig. 4.3 in Wallace et al. (2006), with permission from Elsevier.

Moreover, different parts of the surface will likely receive varying amounts of energy, as some areas may be shadowed while others are directly exposed to the Sun. For each infinitesimal surface element, the incoming radiative energy also depends on direction. A surface element directly exposed to the Sun, for example, receives most of its radiation from the Sun's direction, whereas other parts of the sky contribute far less to the total energy received. Additionally, different wavelengths contribute varying amounts to the total received radiative energy. Hence, in the most fundamental terms, the radiative field is characterized by the *spectral radiance* L_λ, also known as monochromatic or spectral intensity, which is defined as

$$L_\lambda = \frac{\mathrm{d}Q}{\mathrm{d}t\,\cos\theta\,\mathrm{d}A\,\mathrm{d}\Omega\,\mathrm{d}\lambda}\,. \tag{2.40}$$

It specifies the amount $\mathrm{d}Q$ of radiative energy per unit time $\mathrm{d}t$ and unit wavelength interval $\mathrm{d}\lambda$ arriving from a given direction $\mathrm{d}\Omega$ and passing through a unit area $\cos\theta\,\mathrm{d}A$ normal to that direction. Figure 2.3 provides a schematic illustration of this definition. Using this figure, we can see that the direction the spectral radiance refers to is described using a solid angle. Just as a conventional angle is defined as the arc length it spans on a circle divided by the circle's radius, a solid angle is defined as the surface area it spans on a sphere divided by the square of the radius, which can be expressed as

$$\Omega = \frac{A}{r^2}\,. \tag{2.41}$$

Hence, the solid angle covering an entire sphere, with $A = 4\pi r^2$, is equal to 4π steradian (sr), with the steradian being the dimensionless unit of the solid angle. Similarly, the infinitesimal solid angle spanned by the differential angle elements $\mathrm{d}\theta$ and $\mathrm{d}\phi$ is given by

$$\mathrm{d}\Omega = \frac{r\,\mathrm{d}\theta\,r\sin\theta\,\mathrm{d}\phi}{r^2} = \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi\,. \tag{2.42}$$

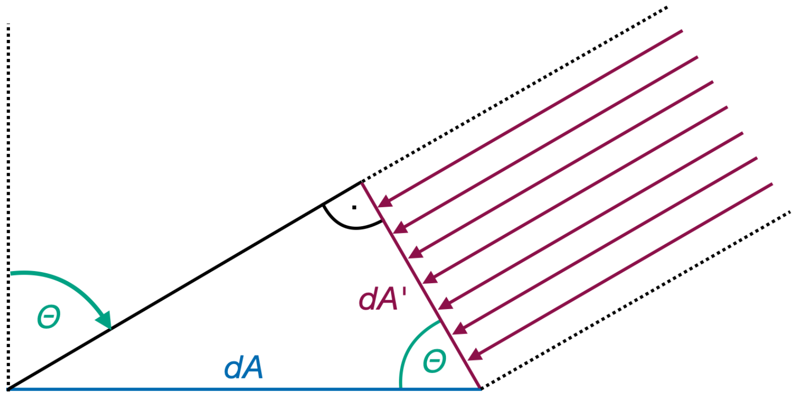

Figure 2.4: Schematic illustration of the relationship between an infinitesimal surface element $\mathrm{d}A$ and the projected area $\mathrm{d}A'$ normal to the direction of incidence used in the definition of spectral radiance.

The direction (θ, ϕ) that $\mathrm{d}\Omega$ refers to is usually specified in terms of the unit vector $\hat{\Omega}$. Besides this directional dependence, Fig. 2.3 also highlights the area $\mathrm{d}A'$ used in the definition of L_λ, which is shown in dark purple. Figure 2.4 illustrates the relationship between this area and the surface element $\mathrm{d}A$ the spectral radiance is actually calculated for. It takes the form

$$\mathrm{d}A' = \cos\theta\,\mathrm{d}A, \tag{2.43}$$

explaining the presence of $\cos\theta$ in Eq. (2.40). By combining all the information so far, we can also obtain the unit of spectral radiance, which is commonly expressed as $\mathrm{W\,m^{-2}\,sr^{-1}\,\mu m^{-1}}$ in atmospheric sciences. The "μm" in this unit emphasizes the dependence on wavelength, distinguishing it from the m^2 term associated with the surface area, explicitly accounting for the typical wavelength range of atmospheric radiation, which extends from approximately 0.1 μm in the ultraviolet to about 100 μm in the infrared.

Although spectral radiance is fundamental for modeling radiative transfer in the atmosphere, it is often more practical to consider the total radiation passing through a surface, irrespective of its directionality. This leads to the definition of the *spectral irradiance* E_λ, which quantifies the radiative energy $\mathrm{d}Q$ per unit time $\mathrm{d}t$ and unit wavelength interval $\mathrm{d}\lambda$ passing through an infinitesimal surface element $\mathrm{d}A$ and is given by

$$E_\lambda = \frac{\mathrm{d}Q}{\mathrm{d}t\,\mathrm{d}A\,\mathrm{d}\lambda} = \int_\Omega L_\lambda \cos\theta\,\mathrm{d}\Omega \overset{(2.42)}{=} \int_{\phi_0}^{\phi_1}\int_{\theta_0}^{\theta_1} L_\lambda \cos\theta \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi. \tag{2.44}$$

It is typically expressed in units of $\mathrm{W\,m^{-2}\,\mu m^{-1}}$ in atmospheric sciences. As shown in Eq. (2.44), it can be obtained by integrating the spectral radiance over a specified solid angle Ω. This solid angle is typically given by one of two hemispheres. Downward spectral irradiance $E_\lambda^\downarrow$ accounts for radiation arriving at the surface from above, integrating L_λ over the upper hemisphere where $\theta_0 = 0$ and $\theta_1 = \frac{\pi}{2}$. Conversely, upward spectral irradiance $E_\lambda^\uparrow$ accounts for radiation passing through the surface from below, integrating L_λ over the lower hemisphere where $\theta_0 = \frac{\pi}{2}$ and $\theta_1 = \pi$. In both cases, the azimuthal integration limits are given by $\phi_0 = 0$ and $\phi_1 = 2\pi$.

Since atmospheric radiation spans a broad range of wavelengths rather than a single monochromatic value, spectral irradiance is often not a particularly practical quantity. Instead,

Table 2.2: Overview of all the radiative quantities defined in this section. Apart from the name and symbol, both the differential definition and the unit of each quantity are provided.

Quantity	Symbol	Differential definition	Unit
Radiative energy	Q	—	J
Radiative flux	Φ	$\frac{\mathrm{d}Q}{\mathrm{d}t}$	W
Irradiance	E	$\frac{\mathrm{d}Q}{\mathrm{d}t\,\mathrm{d}A}$	$\mathrm{W\,m^{-2}}$
Radiance	L	$\frac{\mathrm{d}Q}{\mathrm{d}t\cos\theta\,\mathrm{d}A\,\mathrm{d}\Omega}$	$\mathrm{W\,m^{-2}\,sr^{-1}}$

broadband quantities such as the *irradiance* are used, which is obtained by integrating spectral irradiance over a finite wavelength range $[\lambda_1, \lambda_2]$ of the electromagnetic spectrum and therefore takes the form

$$E = \int_{\lambda_1}^{\lambda_2} E_\lambda \,\mathrm{d}\lambda = \frac{\mathrm{d}Q}{\mathrm{d}t\,\mathrm{d}A}. \tag{2.45}$$

It is expressed in units of $\mathrm{W\,m^{-2}}$. In the context of the radiative transfer solver developed in this thesis, we are furthermore interested in the total radiative energy per unit time crossing the different finite-sized faces of a model grid box. To obtain this quantity, irradiance must be integrated over all infinitesimal surface elements $\mathrm{d}A$ forming the corresponding surface. This defines the *radiative flux* Φ, sometimes also referred to as radiative power, which is given by

$$\Phi = \int_A E \,\mathrm{d}A = \frac{\mathrm{d}Q}{\mathrm{d}t}. \tag{2.46}$$

Φ is typically expressed in units of W. Table 2.2 finally summarizes all the radiative quantities introduced in this section. It lists only broadband quantities, while the corresponding spectral counterparts can be obtained by differentiating each quantity with respect to wavelength.

2.2.2 Radiative transfer in the atmosphere

Using these radiative quantities, this subsection now provides an overview of radiative transfer in the atmosphere. Unless cited otherwise, its content follows Petty (2006b, 2006c) and Wallace et al. (2006).

To describe radiative transfer in the atmosphere, we must first note that any object with a temperature $T > 0\,\mathrm{K}$ emits electromagnetic radiation. An idealized example is that of a blackbody, an object that completely absorbs all incident radiation. To maintain thermal equilibrium and avoid indefinite energy accumulation, such a perfect absorber must also be a perfect emitter. The corresponding spectral radiance emitted by such a blackbody is described by Planck's law, which is given by

$$B_\lambda(T) = \frac{2hc^2}{\lambda^5} \frac{1}{\exp\left(\frac{hc}{\lambda k_B T}\right) - 1}. \tag{2.47}$$

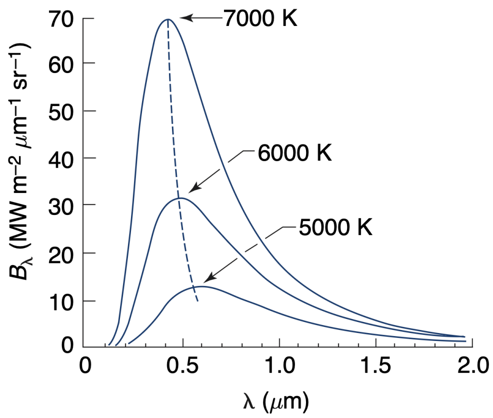

Figure 2.5: Spectral radiance emitted by a blackbody at different temperatures as a function of wavelength. Reprinted from Fig. 4.6 in Wallace et al. (2006), with permission from Elsevier.

Here, $k_B = 1.380649 \times 10^{-23}\,\mathrm{J\,K^{-1}}$ is the Boltzmann constant, another fundamental physical constant. Equation (2.47) uses B_λ instead of L_λ as a symbol for spectral radiance to indicate that this spectral radiance specifically refers to blackbody emission. Apart from that, the equation shows that $B_\lambda(T)$ is independent of θ and ϕ, meaning that blackbody radiation is isotropic. Figure 2.5 illustrates $B_\lambda(T)$ for three different temperatures as a function of wavelength. Examining the three curves, we can see that the emission spectra at different temperatures do not intersect. Apart from that, they are all characterized by a distinct peak, a sharp drop-off toward shorter wavelengths and a more gradual decline toward longer wavelengths. The peak wavelength λ_{max} is inversely proportional to the temperature of the blackbody and can be determined using Wien's displacement law, which is given by

$$\lambda_{\mathrm{max}} = \frac{2897\,\mu\mathrm{m\,K}}{T}\,. \tag{2.48}$$

In general, however, most objects are not perfect blackbodies but rather graybodies, meaning they absorb only part of the incident radiation, while the remainder is reflected, transmitted, or scattered. At any wavelength λ, such a graybody is characterized by its *absorptivity* α_λ, which is defined as the fraction of the incident spectral radiance that is absorbed by the object:

$$\alpha_\lambda = \frac{L_\lambda^{\mathrm{absorbed}}}{L_\lambda^{\mathrm{incident}}}\,. \tag{2.49}$$

It is important to note that a graybody does not necessarily have to be a solid object; for example, every part of the atmosphere also behaves as a graybody, with correspondingly low absorptivities in the visible spectrum — otherwise, solar radiation would not reach the surface. However, as long as $\alpha_\lambda > 0$ and $T > 0\,\mathrm{K}$, a graybody also emits electromagnetic radiation. This is described by Kirchhoff's law, which states that, at any wavelength λ, the absorptivity $\alpha_\lambda(\theta,\phi)$ of an object for radiation arriving from a given direction is equal to its emissivity $\epsilon_\lambda(\theta,\phi)$ in the same direction. Mathematically, this can be written as

$$\alpha_\lambda(\theta,\phi) = \epsilon_\lambda(\theta,\phi)\,. \tag{2.50}$$

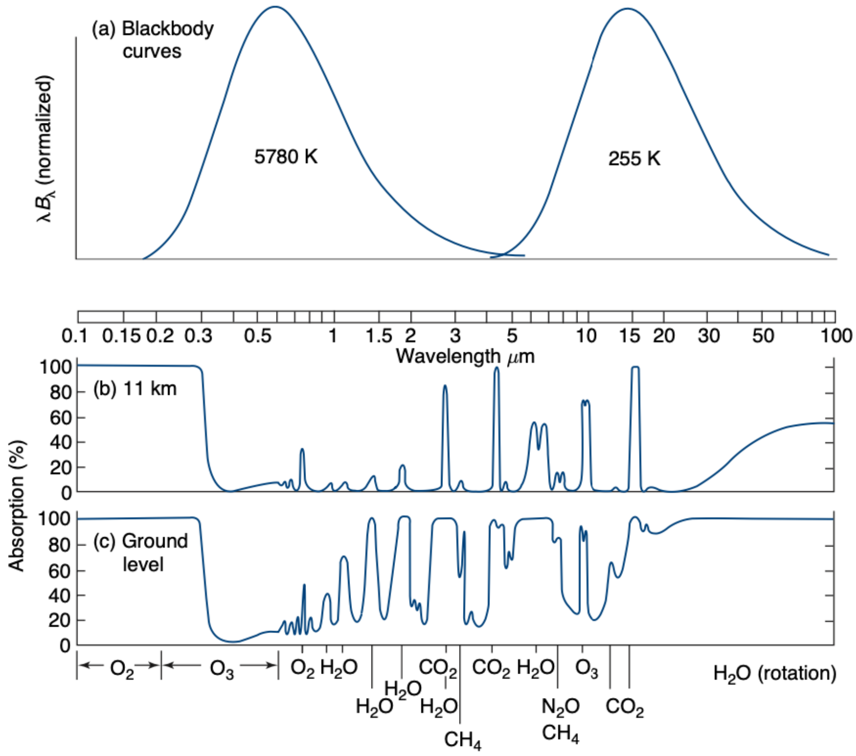

Figure 2.6: Panel (**a**) shows the blackbody emission spectra representative of the Sun (left) and Earth (right) as a function of wavelength λ. The y-axis of this plot represents spectral radiance B_λ multiplied by wavelength λ, with the curves normalized so that the area under both is equal. Panel (**b**) depicts the absorptivity of the atmosphere above 11 km height, whereas panel (**c**) shows the total absorptivity of the entire atmosphere. Reprinted from Fig. 4.7 in Wallace et al. (2006), with permission from Elsevier.

The *emissivity* ϵ_λ in this expression quantifies the amount of spectral radiance an object emits relative to a blackbody at the same temperature and is defined as

$$\epsilon_\lambda = \frac{L_\lambda^{\text{emitted}}}{B_\lambda(T)}. \tag{2.51}$$

Just like α_λ, the emissivity ϵ_λ of an object ranges between 0 and 1, with $\epsilon_\lambda = 1$ indicating that the object behaves like a blackbody at the corresponding wavelength. Combined with Eq. (2.50), Eq. (2.51) can be used to quantify the spectral radiance emitted by any object with an absorptivity α_λ and a temperature $T > 0\,\text{K}$. For graybodies, where $\alpha_\lambda = \epsilon_\lambda < 1$, this emitted spectral radiance is always greater than zero but remains lower than that of a blackbody at the same temperature.

Building on this foundation, the two primary sources of radiative energy in the atmosphere can be identified, each associated with a distinct wavelength regime, as illustrated in Fig. 2.6:

(a) *Solar radiation* (also referred to as *shortwave radiation*) originates from the Sun and enters the atmosphere at its upper boundary. It can be approximated by the radiation emitted from a blackbody at the Sun's effective surface temperature of 5780 K, as shown by the left spectrum in panel (a). Solar radiation covers wavelengths from approximately 0.1 μm in the ultraviolet to 4 μm in the near-infrared, with a significant portion falling within the visible band (0.38 μm–0.78 μm). The peak emission occurs at around 0.5 μm wavelength. Examining panels (b) and (c), we can also see that the atmosphere is largely transparent in this spectral range, allowing much of the incoming solar radiation to reach the surface. This also explains why, as discussed earlier in Sect. 2.1.4, solar radiation primarily warms near-surface air masses.

(b) *Thermal radiation* (also referred to as *longwave radiation*), on the other hand, is emitted by Earth and its atmosphere. It can be approximated by the radiation emitted from a blackbody at a temperature of 255 K, as shown by the right spectrum in panel (a). Thermal radiation spans wavelengths from approximately 4 μm in the near-infrared to 100 μm in the infrared, with a maximum at around 11 μm. Unlike solar radiation, it does not pass through the atmosphere as easily, with the atmosphere being far less transparent in this spectral range, as one can see in panel (c), indicating that radiative transfer in the solar and thermal spectral ranges is dominated by different processes.

Examining panel (a) of Fig. 2.6 more closely, one might wonder why the y-axis displays λB_λ instead of the spectral radiance B_λ alone. The reason lies in the logarithmic scaling of the x-axis. By plotting λB_λ instead of B_λ, any area under the curves in the plot, given by $\lambda B_\lambda \,\mathrm{d}\ln\lambda$, still correctly represents the corresponding broadband radiance $\mathrm{d}B$, since

$$\mathrm{d}B = B_\lambda \,\mathrm{d}\lambda = B_\lambda \lambda \,\mathrm{d}\ln\lambda \qquad \text{with} \quad \mathrm{d}\ln\lambda = \lambda^{-1}\,\mathrm{d}\lambda\,. \tag{2.52}$$

Another aspect that might seem surprising is that the two blackbody spectra intersect at around 4 μm wavelength, despite Fig. 2.5 clearly showing that such an intersection does not occur. This apparent contradiction arises because the spectra in Fig. 2.6 are normalized so that the total area under each curve is equal. Without this normalization, the terrestrial blackbody spectrum would be barely visible in comparison to the solar one, as its emitted spectral radiance is significantly lower across all wavelengths. Despite this difference, however, thermal radiation is just as important for Earth's global energy budget as its solar counterpart. From the Sun's perspective, Earth simply spans only a tiny solid angle, meaning that only a small fraction of the Sun's total emitted radiation reaches Earth. Earth, on the other hand, compensates for the corresponding net intake of solar radiation by emitting thermal radiation isotropically in all directions. On a global and temporal average, the net outgoing thermal radiation must equal the net incoming solar radiation — otherwise, Earth's temperature would continuously rise or fall. The key takeaway from Fig. 2.6, however, is that while solar radiation is primarily concentrated at wavelengths between 0.1 μm and 4 μm, thermal radiation dominates in the range from 4 μm to 100 μm. This clear spectral separation allows for an independent treatment of solar and thermal radiative transfer in most atmospheric applications.

With this foundation, we can now finally discuss the main aspects of radiative transfer in the atmosphere. To this end, Fig. 2.7 presents a schematic representation of Earth's global mean energy budget, illustrating all major radiative interactions in the atmosphere. Starting in the top left, we can see that, on a global and temporal average, 340 W m^{-2} of solar radiation enters the

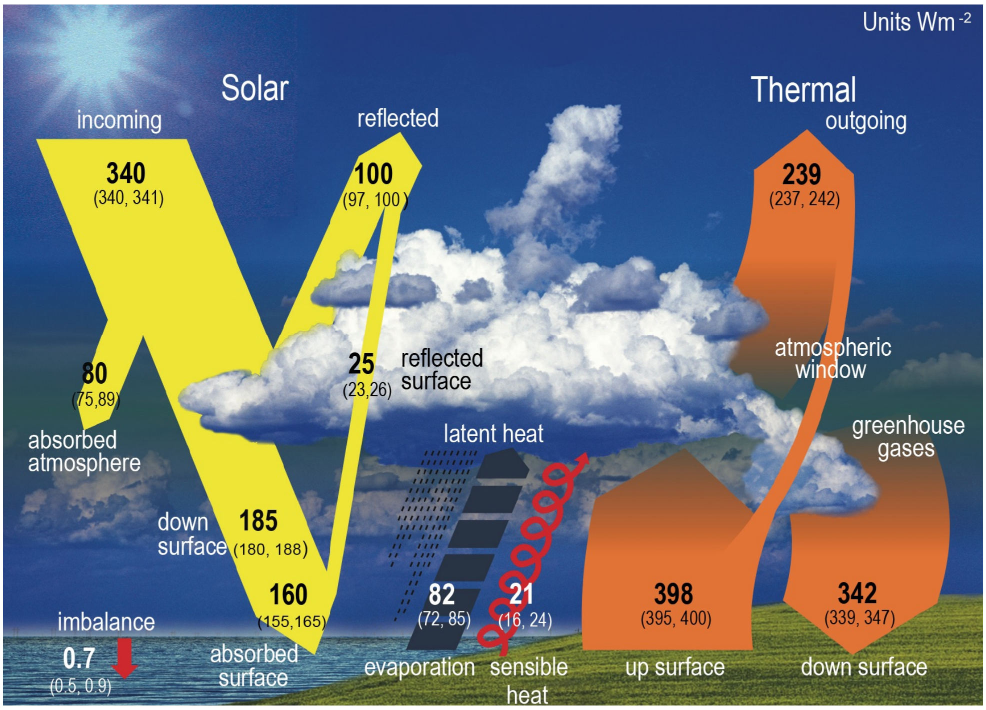

Figure 2.7: Schematic representation of the global mean energy budget of the Earth. Numbers indicate best estimates for the magnitudes of the globally averaged energy balance components in $\mathrm{W\,m^{-2}}$ together with their uncertainty ranges in parentheses (5–95% confidence range), representing climate conditions at the beginning of the 21st century. Figure 7.2 (upper panel) from Forster et al. (2021).

atmosphere at its upper boundary. As it propagates downward, an average of 23.5% ($80\,\mathrm{W\,m^{-2}}$) of this incoming solar radiation is absorbed by the atmosphere, while 22.1% ($75\,\mathrm{W\,m^{-2}}$) is reflected back to space, primarily by clouds. Upon reaching the surface, an additional 13.5% ($25\,\mathrm{W\,m^{-2}}$) of the remaining $185\,\mathrm{W\,m^{-2}}$ is reflected back to space, leaving $160\,\mathrm{W\,m^{-2}}$ to be absorbed by the surface. Overall, solar radiation is thus primarily affected by scattering and absorption as it travels through the atmosphere. However, as shown earlier in panel (c) of Fig. 2.6, the atmosphere is also relatively transparent in this spectral range, allowing 54.4% of the incoming solar radiation to reach the surface.

At the surface, this net intake of solar radiation is then balanced by three main mechanisms: First, surface heating establishes a sensible heat flux of $21\,\mathrm{W\,m^{-2}}$ into the atmosphere. Second, the evaporation of water results in a latent heat flux of $82\,\mathrm{W\,m^{-2}}$. The remaining $57\,\mathrm{W\,m^{-2}}$ — except for $0.7\,\mathrm{W\,m^{-2}}$, which account for the ongoing energy imbalance driving global warming — are eventually balanced through thermal radiative transfer. In this spectral range, both the Earth and the atmosphere emit radiation. However, while Earth's surface emits an average of $398\,\mathrm{W\,m^{-2}}$, the comparatively colder atmosphere emits only $342\,\mathrm{W\,m^{-2}}$ back toward the surface. This results in a net radiative loss of $56\,\mathrm{W\,m^{-2}}$, closing the surface energy budget. Beyond that, Fig. 2.7 also highlights that the atmosphere is far less transparent to thermal radiation than it is to solar radiation, with only a small fraction of the surface-emitted radiation escaping directly into space, while the majority is absorbed and re-emitted by the atmosphere.

Unlike solar radiation, which is primarily scattered, transmitted, and only to some extent absorbed by the atmosphere, the transport of thermal radiation is thus primarily governed by absorption and emission throughout all parts of the atmosphere. Ultimately, $239\,\mathrm{W\,m^{-2}}$ of thermal radiation are emitted to space, leaving a small imbalance of $0.7\,\mathrm{W\,m^{-2}}$ compared to the net solar energy input of $240\,\mathrm{W\,m^{-2}}$ — which, as mentioned before, drives the ongoing global warming.

Besides illustrating all these major radiative interactions, Fig. 2.7 also highlights the fundamental role of clouds in Earth's global mean energy budget and their contrasting effect across the two different atmospheric wavelength regimes. In the solar spectral range, the figure clearly shows that clouds primarily reflect a substantial fraction of the incoming solar radiation back to space, thereby cooling the surface. In the thermal spectral range, however, clouds act as strong absorbers, as shown in panel (c) of Fig. 2.6, which highlights the significant contribution of water to the overall atmospheric absorptivity in this spectral range. Following Kirchhoff's law, strong absorbers are also strong emitters, meaning that clouds contribute significantly to the $342\,\mathrm{W\,m^{-2}}$ of thermal radiation emitted toward the surface. This also explains why temperatures drop significantly more during clear-sky nights, where the absence of clouds leads to substantially lower thermal downward radiation and, consequently, a greater overall energy loss at the surface.

2.2.3 The radiative transfer equation

Having introduced all major radiative interactions in the atmosphere, this subsection now develops a quantitative description of these processes. To this end, we consider the change $\mathrm{d}L_\lambda(\hat{\Omega})$ in spectral radiance along an infinitesimal path segment $\mathrm{d}s$. Along this path, spectral radiance can decrease due to photon absorption or scattering out of the direction of interest and increase due to photon emission or scattering from other directions into the direction of interest. In the following, quantitative expressions for each of these processes are developed, ultimately leading to the radiative transfer equation (RTE). Unless cited otherwise, the content of this subsection is adapted from Petty (2006d, 2006e, 2006f, 2006g) and Wallace et al. (2006).

Extinction

Absorption and scattering out of a given direction are collectively termed extinction, since both processes effectively reduce the spectral radiance along a given path segment. To quantify this reduction, we must recall that the atmosphere is a mixture of various gaseous, liquid and solid components. This includes molecular gases such as nitrogen, oxygen, water vapor, and trace gases, as well as aerosols and the water droplets and ice crystals forming clouds. The extinction caused by any one of these constituents can be analyzed using Fig. 2.8, which illustrates a small volume element $\mathrm{d}V = A\,\mathrm{d}s$ along a path segment $\mathrm{d}s$, containing N_i particles of a given species. Each of these particles is associated with an *extinction cross-section* $\sigma_{\mathrm{ext},\lambda,i}$, representing the effective area it presents for removing photons from the incident beam via absorption or scattering. The probability that a photon is extinguished within this volume is then given by the total extinction cross-section of all particles, $N_i\,\sigma_{\mathrm{ext},\lambda,i}$, relative to the

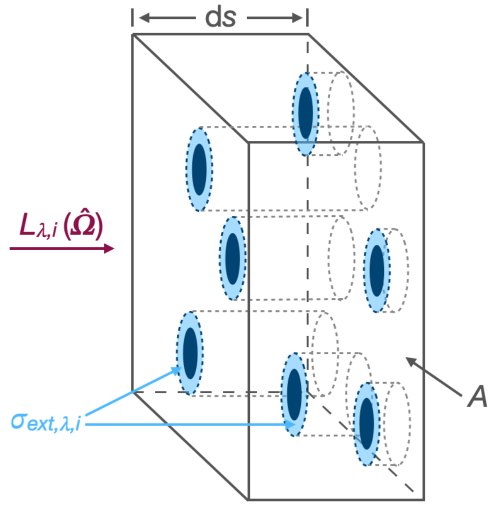

Figure 2.8: Schematic illustration of the change in spectral radiance $L_{\lambda,i}(\hat{\Omega})$ along an infinitesimal path segment ds due to extinction by a single atmospheric constituent. Within the corresponding volume element d$V = A$ds, the probability of a photon being extinguished is determined by the sum $N_i\,\sigma_{\text{ext},\lambda,i}$ of the extinction cross-sections $\sigma_{\text{ext},\lambda,i}$ of the N_i particles in the volume, divided by the total cross-sectional area A. Inspired by Fig. 7.16 in Demtröder (2015).

total cross-sectional area A. Introducing the *number density* $n_i = \frac{N_i}{V} = \frac{N_i}{A\,\mathrm{d}s}$ of species i, the corresponding reduction d$L_{\lambda,\text{ext},i}(\hat{\Omega})$ in spectral radiance along the infinitesimal path segment ds can be formulated as

$$\mathrm{d}L_{\lambda,\text{ext},i}(\hat{\Omega}) = -\frac{N_i\,\sigma_{\text{ext},\lambda,i}}{A}\,L_\lambda(\hat{\Omega}) = -n_i\,\sigma_{\text{ext},\lambda,i}\,L_\lambda(\hat{\Omega})\,\mathrm{d}s\,. \tag{2.53}$$

Defining the *extinction coefficient* $\beta_{\text{ext},\lambda,i}$ of species i as $\beta_{\text{ext},\lambda,i} = n_i\,\sigma_{\text{ext},\lambda,i}$, we can then obtain the total extinction along the path by summing the contributions expressed via Eq. (2.53) over all M atmospheric constituents, which yields

$$\mathrm{d}L_{\lambda,\text{ext}}(\hat{\Omega}) = \sum_{i=1}^{M}\mathrm{d}L_{\lambda,\text{ext},i}(\hat{\Omega}) = -\sum_{i=1}^{M}\beta_{\text{ext},\lambda,i}\,L_\lambda(\hat{\Omega})\,\mathrm{d}s = -\beta_{\text{ext},\lambda}L_\lambda(\hat{\Omega})\,\mathrm{d}s\,. \tag{2.54}$$

Here, $\beta_{\text{ext},\lambda}$ represents the total extinction coefficient due to all atmospheric components, which is defined as

$$\beta_{\text{ext},\lambda} = \sum_{i=1}^{M}\beta_{\text{ext},\lambda,i} = \sum_{i=1}^{M}n_i\,\sigma_{\text{ext},\lambda,i}\,. \tag{2.55}$$

It is often decomposed into the contributions from different atmospheric species, such as molecular gases, aerosols, and clouds, or, alternatively, into the contributions from the two fundamental processes governing extinction: absorption and scattering. These are characterized by the *absorption coefficient* $\beta_{\text{abs},\lambda}$ and the *scattering coefficient* $\beta_{\text{sca},\lambda}$, so that

$$\beta_{\text{ext},\lambda} = \beta_{\text{abs},\lambda} + \beta_{\text{sca},\lambda}\,. \tag{2.56}$$

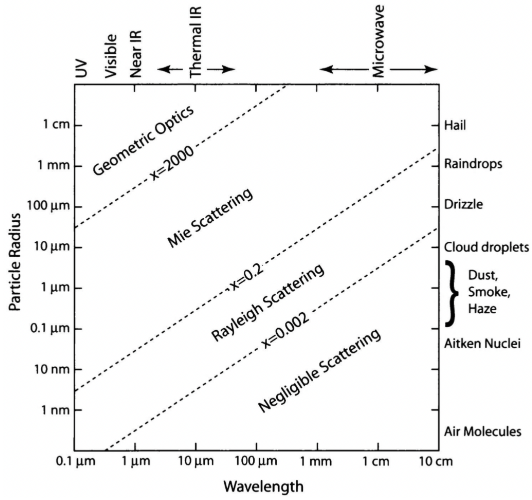

Figure 2.9: Scattering regimes of radiation interacting with spherical particles of radius r as a function of wavelength λ. The dashed lines mark the boundaries between the different scattering regimes, each labeled with the corresponding value of the size parameter x. Additionally, various atmospheric constituents associated with specific particle radii are listed on the right, whereas the spectral regions corresponding to the different wavelengths are indicated at the top. Figure 12.1 from Petty (2006g), used with permission from the author.

While $\beta_{\text{abs},\lambda}$ is governed by the absorption spectra of the various atmospheric components and therefore is strongly wavelength-dependent, $\beta_{\text{sca},\lambda}$ is determined by the scattering regime that radiation of wavelength λ is subject to when interacting with particles of radius r. These regimes are characterized by the dimensionless *size parameter*

$$x = \frac{2\pi r}{\lambda}. \tag{2.57}$$

Depending on the value of x, the scattering falls into one of four regimes illustrated in Fig. 2.9:

(a) If $x \lessapprox 0.002$, scattering by the corresponding species is negligible compared to that by other atmospheric components. While this does not apply to any atmospheric constituent in the solar spectral range, air molecules in the thermal spectral range fall into this regime. This explains why scattering plays an insignificant role in most parts of the atmosphere in this spectral range, as previously noted in Sect. 2.2.2.

(b) If $0.002 \lessapprox x \lessapprox 0.2$, the incoming radiation undergoes *Rayleigh scattering*, which is an approximation of the Mie theory discussed below for size parameters $x \ll 1$. In this regime, the scattering coefficient exhibits a strong wavelength dependence, given by $\beta_{\text{sca},\lambda} \propto \lambda^{-4}$, meaning that shorter wavelengths are scattered much more efficiently than longer ones. Rayleigh scattering primarily governs the interaction of solar radiation with air molecules, explaining why the cloud- and aerosol-free sky appears blue — shorter (blue) wavelengths are simply scattered much more than longer (red) ones are. As illustrated in Fig. 2.10, Rayleigh scattering at $x = 0.1$ exhibits a relatively uniform angular distribution, with maxima in both the forward and backward directions and minima perpendicular to the incident direction.

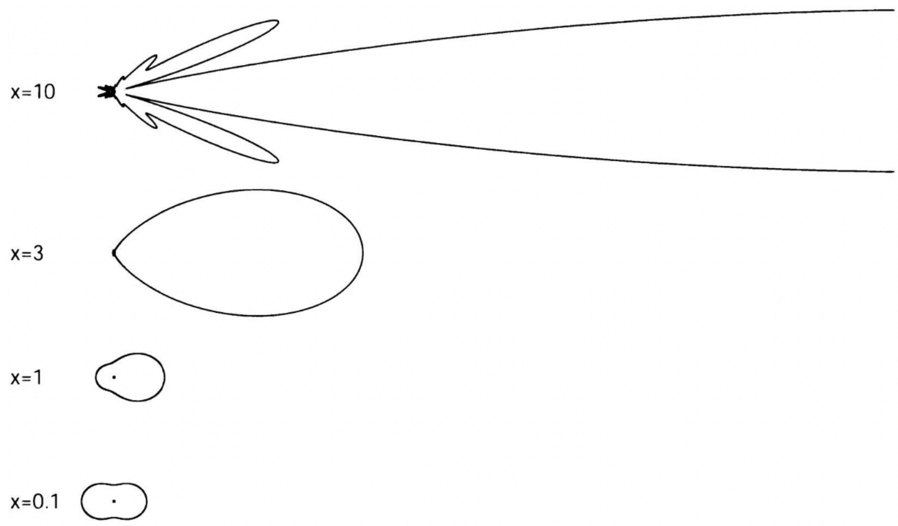

Figure 2.10: Angular distribution of the scattering probability for radiation coming from the left, shown for various values of the size parameter x. Figure 12.8 from Petty (2006g), used with permission from the author.

(c) If $0.2 \lessapprox x \lessapprox 2000$, the radiation is subject to *Mie scattering*, which governs the interaction of solar radiation with cloud droplets and aerosols, such as the dust, smoke and haze mentioned in Fig. 2.9. Compared to Rayleigh scattering, Mie scattering exhibits a much weaker wavelength dependence, with an approximate scaling of $\beta_{\mathrm{sca},\lambda} \propto \lambda^{-1.3}$ for aerosols (Reuter et al., 2001, p. 152), though the exponent actually varies between 0 and 2 depending on particle size (Steinacker and Umdasch, 2014). This weak wavelength dependence explains why clouds appear white in the solar spectral range — all wavelengths are simply scattered to a similar extent. Looking at Fig. 2.10, we can further see that scattering in the Mie regime at $x = 10$ is characterized by a strong forward-scattering peak that is so dominant that it even extends beyond the figure's scale. As x increases, this peak becomes even more pronounced, eventually approaching a delta function in the forward direction. Apart from this strong forward peak, Mie scattering also exhibits a much more complex angular dependence compared to Rayleigh scattering. At $x \gtrapprox 500$, for instance, two distinct peaks emerge at scattering angles of approximately 137° and 130° in the backward direction, corresponding to light scattered into the primary and secondary rainbows, respectively. The exact angles of these peaks exhibit a slight wavelength dependence, producing the characteristic color transitions from blue to red in the primary rainbow and from red to blue in the secondary rainbow. Apart from that, another notable feature of the Mie regime is a pronounced enhancement of scattering at approximately 180° in the backward direction, known as the glory.

(d) For $x \gtrapprox 2000$, Mie scattering gradually transitions into the *geometric optics* regime. This applies, for example, to raindrops in the solar spectral range. In this regime, classical ray optics provide an effective framework for explaining optical phenomena such as the rainbow. However, certain effects, such as the strong forward-scattering peak that persists in this range, cannot be fully accounted for with this theory.

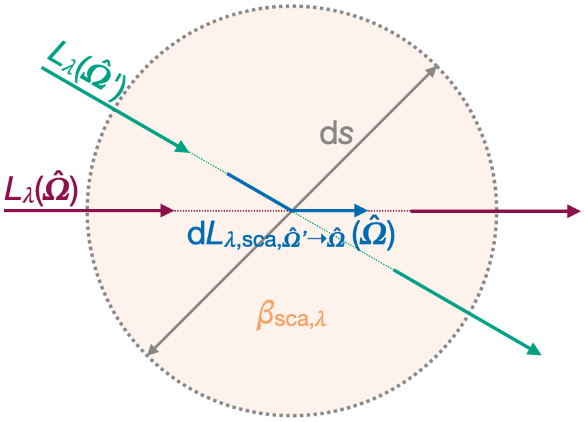

Figure 2.11: Two-dimensional schematic illustration of radiation scattered from an incident direction $\hat{\Omega}'$ into the direction of interest $\hat{\Omega}$. As radiation passes through the infinitesimal spherical volume element $\mathrm{d}V = \frac{4}{3}\pi\,(\frac{\mathrm{d}s}{2})^3$, a fraction of the spectral radiance $L_\lambda(\hat{\Omega}')$, determined by the scattering coefficient $\beta_{\mathrm{sca},\lambda}$, is scattered into all directions. The portion scattered into $\hat{\Omega}$, indicated by the blue arrow, contributes to an increase in the spectral radiance $L_\lambda(\hat{\Omega})$ in that direction.

Scattering

Having established that scattering reduces the spectral radiance $L_\lambda(\hat{\Omega})$ along a given path segment $\mathrm{d}s$, this section now examines how radiation scattered from other directions into the direction of interest contributes to an increase in $L_\lambda(\hat{\Omega})$ along the same path. To quantify this contribution, we consider an infinitesimal spherical volume element $\mathrm{d}V = \frac{4}{3}\pi\left(\frac{\mathrm{d}s}{2}\right)^3$ centered along $\mathrm{d}s$, as illustrated in Fig. 2.11. Within this infinitesimal volume, the scattering coefficient $\beta_{\mathrm{sca},\lambda}$ can be assumed constant. Additionally, due to the spherical symmetry of the volume, radiation arriving from any direction, specified by the unit vector $\hat{\Omega}'$, traverses the volume along equally long paths. Taken together, this implies that the spectral radiance $L_\lambda(\hat{\Omega}')$ from any given direction $\hat{\Omega}'$ is scattered by the same fraction $\beta_{\mathrm{sca},\lambda}\,\mathrm{d}s$ within the volume. The corresponding decrease in $L_\lambda(\hat{\Omega}')$ is given by

$$\mathrm{d}L_{\lambda,\mathrm{sca}}(\hat{\Omega}') = -\beta_{\mathrm{sca},\lambda} L_\lambda(\hat{\Omega}')\,\mathrm{d}s\,. \tag{2.58}$$

To quantify the fraction of this scattered radiation that is redirected into the direction of interest $\hat{\Omega}$, we introduce the *scattering phase function* $p(\hat{\Omega}',\hat{\Omega})$. This probability density function (PDF) describes the angular distribution of radiation scattered from an incident direction $\hat{\Omega}'$. Unlike conventional PDFs, however, the integral of $p(\hat{\Omega}',\hat{\Omega})$ over all possible directions $\hat{\Omega}'$ is not normalized to 1, but instead equals 4π, which can be expressed by

$$\int_{4\pi} p(\hat{\Omega}',\hat{\Omega})\,\mathrm{d}\Omega' = 4\pi \tag{2.59}$$

for each direction of interest $\hat{\Omega}$. As a result, the fraction of scattered radiation originally propagating in direction $\hat{\Omega}'$ that is redirected into $\hat{\Omega}$ is given by $\frac{p(\hat{\Omega}',\hat{\Omega})\,\mathrm{d}\Omega'}{4\pi}$. Using this, the

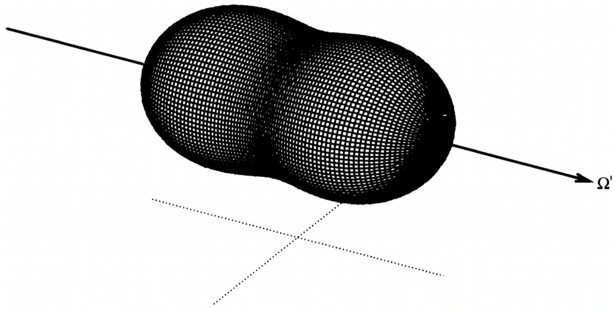

Figure 2.12: Three-dimensional rendering of the scattering phase function $p(\hat{\Omega}', \hat{\Omega})$ for Rayleigh scattering. The vector $\hat{\Omega}'$ represents the direction of the incident radiation. Slightly modified Figure 12.3 from Petty (2006g), used with permission from the author.

increase $\mathrm{d}L_{\lambda,\mathrm{sca},\hat{\Omega}'\to\hat{\Omega}}$ in $L_\lambda(\hat{\Omega})$ due to scattering from $\hat{\Omega}'$ into $\hat{\Omega}$ can be expressed as

$$\begin{aligned}\mathrm{d}L_{\lambda,\mathrm{sca},\hat{\Omega}'\to\hat{\Omega}} &= \underbrace{\beta_{\mathrm{sca},\lambda} L_\lambda(\hat{\Omega}')\,\mathrm{d}s}_{\substack{\text{amount of } L_\lambda(\hat{\Omega}')\\ \text{that is scattered}\\ \text{into any direction}}} \cdot \underbrace{\frac{p(\hat{\Omega}',\hat{\Omega})\,\mathrm{d}\Omega'}{4\pi}}_{\substack{\text{fraction of the}\\ \text{scattered radiation}\\ \text{that is scattered into } \hat{\Omega}}}\\ &= \frac{\beta_{\mathrm{sca},\lambda}}{4\pi} p(\hat{\Omega}',\hat{\Omega}) L_\lambda(\hat{\Omega}')\,\mathrm{d}\Omega'\mathrm{d}s. \end{aligned} \tag{2.60}$$

To determine the total increase in $L_\lambda(\hat{\Omega})$ due to scattering from all possible directions $\hat{\Omega}'$ into $\hat{\Omega}$, we integrate Eq. (2.60) over the entire sphere, which yields

$$\mathrm{d}L_{\lambda,\mathrm{sca}}(\hat{\Omega}) = \frac{\beta_{\mathrm{sca},\lambda}}{4\pi} \int_{4\pi} p(\hat{\Omega}',\hat{\Omega}) L_\lambda(\hat{\Omega}')\,\mathrm{d}\Omega'\,\mathrm{d}s. \tag{2.61}$$

The scattering coefficient in this equation is often expressed in terms of the extinction coefficient $\beta_{\mathrm{ext},\lambda}$. To this end, the *single-scattering albedo* $\omega_{0,\lambda}$ is introduced, which is defined as

$$\omega_{0,\lambda} = \frac{\beta_{\mathrm{sca},\lambda}}{\beta_{\mathrm{ext},\lambda}} = \frac{\beta_{\mathrm{sca},\lambda}}{\beta_{\mathrm{abs},\lambda} + \beta_{\mathrm{sca},\lambda}}. \tag{2.62}$$

By substituting Eq. (2.62) into Eq. (2.61), we obtain

$$\mathrm{d}L_{\lambda,\mathrm{sca}}(\hat{\Omega}) = \frac{\omega_{0,\lambda}\beta_{\mathrm{ext},\lambda}}{4\pi} \int_{4\pi} p(\hat{\Omega}',\hat{\Omega}) L_\lambda(\hat{\Omega}')\,\mathrm{d}\Omega'\,\mathrm{d}s. \tag{2.63}$$

This is the standard expression for the increase in spectral radiance $L_\lambda(\hat{\Omega})$ along a path segment $\mathrm{d}s$ due to scattering from all directions $\hat{\Omega}'$ into the direction of interest $\hat{\Omega}$. The scattering phase function $p(\hat{\Omega}',\hat{\Omega})$ in this equation describes exactly the angular distribution of scattered radiation discussed in the preceding section on extinction. When scattering occurs at spherical or randomly oriented particles, this function is rotationally symmetric, as illustrated in Fig. 2.12, which presents a three-dimensional rendering of $p(\hat{\Omega}',\hat{\Omega})$ for Rayleigh scattering. In such cases, the scattering probability depends only on the scattering angle Θ between the incident direction $\hat{\Omega}'$ and the scattered direction $\hat{\Omega}$, allowing $p(\hat{\Omega}',\hat{\Omega})$ to be expressed as $p(\Theta)$ and represented in two dimensions without any loss of information, as shown in Fig. 2.10. Consequently, both Fig. 2.12 and the plot for $x = 0.1$ in Fig. 2.10 illustrate the key characteristics of Rayleigh scattering with equal amounts of forward and backward scattering, maxima in both of these directions, and minima perpendicular to the incident direction. The symmetry required for this simplification of the phase function applies to many atmospheric constituents, such as cloud droplets, air molecules, and small aerosols, even though air molecules and aerosols are generally non-spherical. However, they lack a preferred orientation and can therefore be treated as randomly oriented. In contrast, larger particles such as falling raindrops, snowflakes, or ice crystals do exhibit preferred orientations, necessitating more complex scattering phase functions instead.

Emission

Finally, we examine the increase in spectral radiance $L_\lambda(\hat{\Omega})$ along an infinitesimal path segment $\mathrm{d}s$ due to photon emission. To quantify this contribution, we first recall that the reduction in $L_\lambda(\hat{\Omega})$ along $\mathrm{d}s$ due to absorption is given by

$$\mathrm{d}L_{\lambda,\mathrm{abs}}(\hat{\Omega}) = -\beta_{\mathrm{abs},\lambda} L_\lambda(\hat{\Omega})\,\mathrm{d}s\,. \tag{2.64}$$

Following Eq. (2.49), this implies that the absorptivity α_λ of the medium along $\mathrm{d}s$ can be expressed as

$$\alpha_\lambda \overset{(2.49)}{=} \frac{L_\lambda^{\mathrm{absorbed}}}{L_\lambda^{\mathrm{incident}}} = \frac{-\mathrm{d}L_{\lambda,\mathrm{abs}}(\hat{\Omega})}{L_\lambda(\hat{\Omega})} = \frac{\beta_{\mathrm{abs},\lambda} L_\lambda(\hat{\Omega})\,\mathrm{d}s}{L_\lambda(\hat{\Omega})} = \beta_{\mathrm{abs},\lambda}\,\mathrm{d}s\,. \tag{2.65}$$

Now, according to Kirchhoff's law specified in Eq. (2.50), the absorptivity of a medium is equal to its emissivity ϵ_λ. Following the definition of ϵ_λ from Eq. (2.51), we can then quantify the spectral radiance emitted along $\mathrm{d}s$, and thus the increase $\mathrm{d}L_{\lambda,\mathrm{em}}(\hat{\Omega})$ in $L_\lambda(\hat{\Omega})$ due to emission as follows:

$$\mathrm{d}L_{\lambda,\mathrm{em}}(\hat{\Omega}) \overset{(2.51)}{=} \epsilon_\lambda B_\lambda(T) \overset{(2.50)}{=} \alpha_\lambda B_\lambda(T) \overset{(2.65)}{=} \beta_{\mathrm{abs},\lambda} B_\lambda(T)\,\mathrm{d}s\,. \tag{2.66}$$

Similar to Eq. (2.61), the absorption coefficient $\beta_{\mathrm{abs},\lambda}$ in this equation is often expressed in terms of the extinction coefficient $\beta_{\mathrm{ext},\lambda}$. To achieve this, we rewrite the single-scattering albedo $\omega_{0,\lambda}$ as

$$\omega_{0,\lambda} \overset{(2.62)}{=} \frac{\beta_{\mathrm{sca},\lambda}}{\beta_{\mathrm{ext},\lambda}} = \frac{\beta_{\mathrm{abs},\lambda} + \beta_{\mathrm{sca},\lambda} - \beta_{\mathrm{abs},\lambda}}{\beta_{\mathrm{abs},\lambda} + \beta_{\mathrm{sca},\lambda}} = 1 - \frac{\beta_{\mathrm{abs},\lambda}}{\beta_{\mathrm{ext},\lambda}} \quad \Rightarrow \quad \beta_{\mathrm{abs},\lambda} = (1-\omega_{0,\lambda})\beta_{\mathrm{ext},\lambda}\,. \tag{2.67}$$

Substituting Eq. (2.67) into Eq. (2.66) then yields the standard expression for the emission term in the radiative transfer equation, quantifying the increase in spectral radiance $L_\lambda(\hat{\Omega})$ along $\mathrm{d}s$ due to photon emission as

$$\mathrm{d}L_{\lambda,\mathrm{em}}(\hat{\Omega}) = (1-\omega_{0,\lambda})\beta_{\mathrm{ext},\lambda} B_\lambda(T)\,\mathrm{d}s\,. \tag{2.68}$$

Since at typical atmospheric temperatures, $B_\lambda(T)$ is predominantly concentrated in the wavelength range from 4 μm to 100 μm, this emission term only provides significant contributions to radiative transfer in the thermal spectral range.

Having quantified all atmospheric interactions that either increase or decrease the spectral radiance $L_\lambda(\hat{\Omega})$ along a given path segment ds, we can now combine the three terms specified in Eqs. (2.54), (2.63), and (2.68) to obtain the total change d$L_\lambda(\hat{\Omega})$ in $L_\lambda(\hat{\Omega})$ along ds, which is given by

$$\begin{aligned} \mathrm{d}L_\lambda(\hat{\Omega}) &= \mathrm{d}L_{\lambda,\mathrm{ext}}(\hat{\Omega}) + \mathrm{d}L_{\lambda,\mathrm{sca}}(\hat{\Omega}) + \mathrm{d}L_{\lambda,\mathrm{em}}(\hat{\Omega}) \\ &= \left(-L_\lambda(\hat{\Omega}) + \frac{\omega_{0,\lambda}}{4\pi}\int_{4\pi} p(\hat{\Omega}',\hat{\Omega}) L_\lambda(\hat{\Omega}')\,\mathrm{d}\Omega' + (1-\omega_{0,\lambda})B_\lambda(T)\right)\beta_{\mathrm{ext},\lambda}\,\mathrm{d}s\,. \end{aligned} \tag{2.69}$$

Rearranging for $\frac{\mathrm{d}L_\lambda(\hat{\Omega})}{\beta_{\mathrm{ext},\lambda}\,\mathrm{d}s}$, we finally arrive at the standard form of the *radiative transfer equation* (RTE), as presented, for example, in Mayer (2009):

$$\frac{\mathrm{d}L_\lambda(\hat{\Omega})}{\beta_{\mathrm{ext},\lambda}\,\mathrm{d}s} = -L_\lambda(\hat{\Omega}) + \frac{\omega_{0,\lambda}}{4\pi}\int_{4\pi} p(\hat{\Omega}',\hat{\Omega}) L_\lambda(\hat{\Omega}')\,\mathrm{d}\Omega' + (1-\omega_{0,\lambda})B_\lambda(T)\,. \tag{2.70}$$

Given appropriate boundary conditions specifying $L_\lambda(\hat{\Omega})$ at the edges of the domain — such as the extraterrestrial solar radiance entering Earth's atmosphere at the top — this integro-differential equation theoretically allows for the determination of $L_\lambda(\hat{\Omega})$ for all directions $\hat{\Omega}$, wavelengths λ, and locations within a domain. In the context of weather forecasting, however, the primary interest is not in solving for $L_\lambda(\hat{\Omega})$, but in deriving the resulting sources and sinks of heat in the atmosphere, as these provide essential input for the thermodynamic equation (Eq. (2.3)) within the primitive equations, which govern the atmosphere's future evolution. The heating and cooling rates determining this heat source term are obtained from net irradiances at the boundaries of atmospheric volume elements, which, in turn, can be computed from the spectral radiances $L_\lambda(\hat{\Omega})$. To demonstrate this, we consider an infinitesimal atmospheric volume element $\mathrm{d}V = \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$. Each of its six faces is associated with two irradiances: one accounting for radiation entering the volume — the incoming irradiance — and one accounting for radiation leaving the volume — the outgoing irradiance. At the upper surface element of dV, given by $\mathrm{d}A = \mathrm{d}x\,\mathrm{d}y$, for example, the incoming irradiance is given by the downward irradiance $E^{\downarrow}$, which accounts for all the radiation passing through dA from above — that is, radiation arriving from zenith angles $\theta \in [0, \frac{\pi}{2}]$ and azimuth angles $\phi \in [0, 2\pi]$. It can therefore be calculated as

$$E^{\downarrow} = \int_0^\infty \int_0^{2\pi} \int_0^{\frac{\pi}{2}} L_\lambda(\theta,\phi)\,\cos\theta\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi\,\mathrm{d}\lambda\,. \tag{2.71}$$

Similarly, the outgoing irradiance is given by the upward irradiance $E^{\uparrow}$, which accounts for all the radiation passing through dA from below — that is, from zenith angles $\theta \in [\frac{\pi}{2}, \pi]$ and azimuth angles $\phi \in [0, 2\pi]$. It can be obtained through

$$E^{\uparrow} = \int_0^\infty \int_0^{2\pi} \int_{\frac{\pi}{2}}^{\pi} L_\lambda(\theta,\phi)\,\cos\theta\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi\,\mathrm{d}\lambda\,. \tag{2.72}$$

These irradiances can then be used to compute the *heating rate* $\frac{\partial T}{\partial t}$ within the volume element $\mathrm{d}V$, which quantifies the desired local temperature change over time due to radiative heating or cooling and is given by

$$\frac{\partial T}{\partial t} = \frac{1}{\rho\, c_p} \vec{\nabla} \cdot \vec{E}_{\text{net}} = \frac{1}{\rho\, c_p} \left(\frac{\partial E_{\text{net},x}}{\partial x} + \frac{\partial E_{\text{net},y}}{\partial y} + \frac{\partial E_{\text{net},z}}{\partial z} \right) \tag{2.73}$$

(Mayer, 2018). In here, the vertical net irradiance, for example, is given by $E_{\text{net},z} = E^{\downarrow} - E^{\uparrow}$. To illustrate the meaning of this equation, let us consider a simplified case where horizontal net irradiances are negligible ($E_{\text{net},x} = E_{\text{net},y} = 0$). In this case, the heating rate is solely determined by the change in the vertical net irradiance with height, following the relationship

$$\frac{\partial T}{\partial t} = \frac{1}{\rho\, c_p} \frac{\partial E_{\text{net},z}}{\partial z} . \tag{2.74}$$

Now, let us assume that more radiation enters the volume than exits it from both above and below. This is the case when both the downward flux at the top ($E^{\downarrow}_{\text{top}}$) and the upward flux at the bottom ($E^{\uparrow}_{\text{bottom}}$) of the volume are larger than their respective counterparts, i.e.,

$$\begin{aligned} E^{\downarrow}_{\text{top}} &> E^{\uparrow}_{\text{top}} &\Rightarrow\quad E_{\text{net},z,\text{top}} &= E^{\downarrow}_{\text{top}} - E^{\uparrow}_{\text{top}} &> 0 \quad \text{and} \\ E^{\downarrow}_{\text{bottom}} &< E^{\uparrow}_{\text{bottom}} &\Rightarrow\quad E_{\text{net},z,\text{bottom}} &= E^{\downarrow}_{\text{bottom}} - E^{\uparrow}_{\text{bottom}} &< 0 . \end{aligned}$$

In this case, the vertical net irradiance is positive, since

$$\frac{\partial E_{\text{net},z}}{\partial z} = \frac{E_{\text{net},z,\text{top}} - E_{\text{net},z,\text{bottom}}}{\mathrm{d}z} > 0,$$

which leads to $\frac{\partial T}{\partial t} > 0$. Thus, Eq. (2.73) simply states that if more radiative energy enters the volume than exits it, the temperature in the volume increases. Conversely, if more radiation exits than enters, cooling occurs. This heating rate thus provides exactly the source term $\frac{\dot{q}}{c_p}$ required in Eq. (2.3) of the primitive equations, demonstrating how the solution of the RTE is coupled to atmospheric dynamics.

2.2.4 Approximate solutions to the radiative transfer equation

Unfortunately, the full radiative transfer equation has no known analytical solution. This is primarily due to its second term, which quantifies the increase in spectral radiance due to scattering from other directions into the direction of interest. Through this term, the spectral radiance $L_\lambda(\hat{\Omega})$ in any given direction $\hat{\Omega}$ is coupled to the spectral radiances $L_\lambda(\hat{\Omega}')$ of all other directions $\hat{\Omega}'$. As a result, solving Eq. (2.70) requires considering all directions simultaneously, making an analytical solution infeasible (Mayer, 2009). However, solving the full RTE is not always necessary. In the solar spectral range, for instance, the thermal emission term can be neglected, whereas in the thermal spectral range, scattering is insignificant throughout large parts of the atmosphere, as noted in Sect. 2.2.3. It is particularly this omission of scattering that allows for a significant simplification of the RTE, known as *Schwarzschild's equation*. It follows directly from Eq. (2.70) by setting the scattering coefficient to zero ($\beta_{\mathrm{sca},\lambda} = 0$), which in turn reduces the extinction coefficient to the absorption coefficient ($\beta_{\mathrm{ext},\lambda} = \beta_{\mathrm{abs},\lambda}$) and results in a single-scattering albedo of zero ($\omega_{0,\lambda} = 0$). Additionally, since all quantities now refer to the same direction, we can omit the explicit angular dependence of L_λ and instead emphasize the dependence on the path coordinate s along direction $\hat{\Omega}$. This yields the following form of Schwarzschild's equation (Wallace et al., 2006):

$$\frac{\mathrm{d}L_\lambda(s)}{\mathrm{d}s} = -\beta_{\mathrm{abs},\lambda}(s)\,(L_\lambda(s) - B_\lambda(T,s))\,. \tag{2.75}$$

Unlike the full RTE, this differential equation has a well-known analytical solution. Namely, given an initial spectral radiance $L_\lambda(s_0)$ at $s = s_0$, $L_\lambda(s)$ at a distance s along the path is given by

$$L_\lambda(s) = L_\lambda(s_0)\,e^{-\int_{s_0}^{s}\beta_{\mathrm{abs},\lambda}(s')\,\mathrm{d}s'} + \int_{s_0}^{s}\beta_{\mathrm{abs},\lambda}(s')\,B_\lambda(T,s')\,e^{-\int_{s'}^{s}\beta_{\mathrm{abs},\lambda}(s'')\,\mathrm{d}s''}\,\mathrm{d}s' \tag{2.76}$$

(Wallace et al., 2006). Although this expression may seem complex at first, its meaning becomes clear when examining its two components separately:

- The first term, $L_\lambda(s_0)\,e^{-\int_{s_0}^{s}\beta_{\mathrm{abs},\lambda}(s')\,\mathrm{d}s'}$, is known as the Beer–Bouguer–Lambert extinction law and represents the gradual absorption of the initial spectral radiance $L_\lambda(s_0)$ as it propagates from s_0 to s. To this end, the exponent of the exponential function accounts for the cumulative absorption along the path by integrating the absorption coefficients $\beta_{\mathrm{abs},\lambda}(s')$ over all infinitesimal path segments $\mathrm{d}s'$. The longer the path and the larger the absorption coefficients, the greater the reduction of the initial spectral radiance.

- The second term, $\int_{s_0}^{s}\beta_{\mathrm{abs},\lambda}(s')\,B_\lambda(T,s')\,e^{-\int_{s'}^{s}\beta_{\mathrm{abs},\lambda}(s'')\,\mathrm{d}s''}\,\mathrm{d}s'$, on the other hand, accounts for radiation emitted along the path from s_0 to s, as well as its partial absorption before reaching s. To understand its different parts, we recall that according to Eq. (2.66), the spectral radiance emitted along a path segment $\mathrm{d}s'$ is given by

  $$\beta_{\mathrm{abs},\lambda}(s')\,B_\lambda(T,s')\,\mathrm{d}s'\,.$$

  However, not all of this emitted radiation reaches s, since a fraction is also absorbed again along the remaining path from s' to s. The fraction that survives this absorption is given by the exponential factor $e^{-\int_{s'}^{s}\beta_{\mathrm{abs},\lambda}(s'')\,\mathrm{d}s''}$. Thus, the contribution from radiation emitted at s' to the spectral radiance at s can be expressed as

  $$\beta_{\mathrm{abs},\lambda}(s')\,B_\lambda(T,s')\,e^{-\int_{s'}^{s}\beta_{\mathrm{abs},\lambda}(s'')\,\mathrm{d}s''}\,\mathrm{d}s'\,.$$

  By integrating this expression over all path segments $\mathrm{d}s'$ from s_0 to s, we obtain the total contribution of emission to the spectral radiance at s, as expressed in the second term of Eq. (2.76).

Even though Eq. (2.76) provides a general solution for radiative transfer in the absence of scattering, it remains impractical for calculating atmospheric heating rates, as these require broadband irradiances rather than spectral radiances. To obtain these irradiances, L_λ must be computed for a sufficient number of solid angles covering the entire corresponding hemisphere and for enough wavelength bands to adequately resolve the atmospheric absorption spectrum, resulting in a substantial number of calculations for even one such value. Each of these calculations further requires a different form of the integrals in Eq. (2.76), as each solid angle traverses a distinct atmospheric region with varying extinction coefficients and ambient temperatures. In a 1D atmosphere, where properties vary only with height, these pathways remain manageable, allowing for a relatively fast radiative transfer solver based on Schwarzschild's equation. In a fully 3D atmosphere, however, they become significantly more complex, as radiation traverses different grid boxes instead of the same atmospheric layers at each solid angle. Additionally, the number of required calculations increases substantially, as more solid angles are needed to properly capture the horizontal heterogeneity of the atmosphere, and spectral irradiances must be computed for all faces of a grid box rather than just its top and bottom. Beyond these computational aspects, the neglect of scattering makes Eq. (2.76) unsuitable in the solar spectral range, further emphasizing the need for alternative solutions to the RTE in a fully 3D atmosphere.

In the context of NWP models, a major challenge in developing such solvers is parallelization. Due to their high computational cost, NWP models typically distribute their calculations across multiple processing cores, with each core responsible for a specific atmospheric subdomain. Vertically, these subdomains usually span the entire atmosphere, whereas horizontally, they cover only a fraction of the full domain. While this decomposition generally works well for advancing the primitive equations forward in time — since air parcels move only limited distances between time steps — it poses a significant challenge for radiative transfer, which operates across much larger spatial scales due to its interaction at the speed of light. To illustrate this, consider a cloud casting a shadow at Earth's surface. If the Sun is positioned at an angle, this shadow may fall in a different subdomain than the cloud itself, demonstrating how radiative transfer can easily break model parallelization.

Because of all these complexities, radiative transfer is usually not fully solved but rather handled using approximate solutions. This subsection introduces three such methods, all of which will be used in the remainder of this thesis: the δ-Eddington approximation, the Monte Carlo method, and the TenStream solver. Unless cited otherwise, the following information is adapted from Zdunkowski et al. (2007), Mayer (2009) and Jakub and Mayer (2015).

The 1D δ-Eddington approximation

Among the three approximate solutions discussed in this subsection, the 1D δ-Eddington approximation is the simplest, yet also the most widely used. It is based on the *plane-parallel approximation*, which assumes that the optical properties of the atmosphere vary in the vertical direction but remain constant in the horizontal. This assumption makes it well-suited for cloudless skies or homogeneously overcast conditions, where the atmosphere exhibits little to no horizontal variation. The approximation breaks down, however, when numerous individual

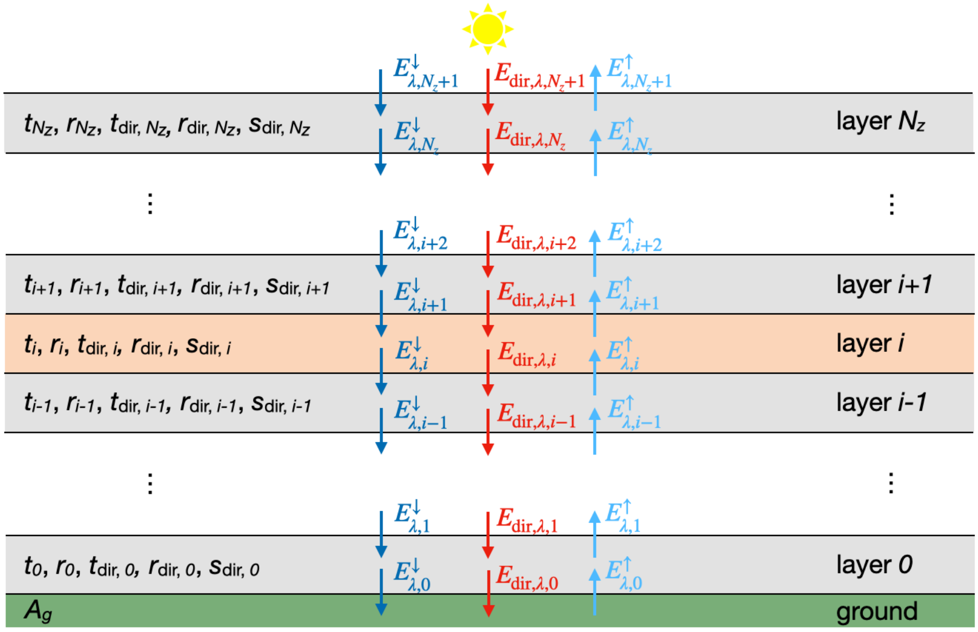

Figure 2.13: Schematic illustration of the two-stream method. In this 1D approximation, the radiative transfer through the different layers of an atmospheric column is described by the diffuse downward spectral irradiances $E^{\downarrow}_{\lambda,i}$ (dark blue arrows) and the diffuse upward spectral irradiances $E^{\uparrow}_{\lambda,i}$ (light blue arrows). Additionally, direct solar radiation that has not yet interacted with the medium is treated separately using the direct spectral irradiances $E_{\mathrm{dir},\lambda,i}$ (red arrows). The radiative transfer within each atmospheric layer i is governed by the layer's transmissivity (t_i), reflectivity (r_i), direct transmissivity ($t_{\mathrm{dir},i}$), as well as the upward ($r_{\mathrm{dir},i}$) and downward ($s_{\mathrm{dir},i}$) scattered fractions of the direct radiation. At the upper and lower domain limits, boundary conditions must be specified, with the ground albedo A_g determining surface reflectivity.

clouds shape the sky, introducing substantial horizontal variability into the atmosphere's optical properties. In early NWP models, this was not an issue. With horizontal grid spacings of about 100 km in the 1990s (e.g., Cullen, 1993), these models simply lacked the resolution needed to explicitly resolve cumulus clouds and other smaller-scale atmospheric features, making the plane-parallel assumption reasonably valid. Today, however, horizontal resolutions have improved significantly, with current regional models featuring grid spacings on the order of a few kilometers (e.g., DWD, 2025), enabling them to resolve individual convective clouds and the associated horizontal variability in the atmospheric optical properties. Yet, despite these advancements, the plane-parallel approximation remains widely used, primarily because of its computational efficiency.

To apply this approximation, the originally three-dimensional atmosphere is reduced to a collection of independent vertical columns, each composed of horizontally infinite layers. Consequently, each grid box can only interact with its upper and lower neighbors in this approximation, but not in the horizontal direction. *Two-stream methods*, such as the δ-Eddington approximation discussed here, further simplify this interaction by considering only two directional components, namely upward and downward spectral irradiances. Figure 2.13 presents a schematic illustration of this concept in an atmospheric column composed of $N_z + 1$ layers.

The characteristic two streams of the method are shown in blue, with the dark blue arrows denoting downward spectral irradiances $E^{\downarrow}_{\lambda,i}$ and the light blue arrows indicating upward spectral irradiances $E^{\uparrow}_{\lambda,i}$. In addition to these two streams, direct solar radiation that has not yet interacted with the atmosphere is treated separately using the direct spectral irradiances $E_{\mathrm{dir},\lambda,i}$, which are illustrated by the red arrows in Fig. 2.13. In contrast to this *direct* radiation, $E^{\downarrow}_{\lambda,i}$ and $E^{\uparrow}_{\lambda,i}$ are also referred to as *diffuse* downward and upward spectral irradiances, since they describe radiation that has already interacted with the medium. This separation into direct and diffuse radiation is necessary because the directionally integrated quantities used in two-stream methods — namely, $E^{\downarrow}_{\lambda,i}$, $E^{\uparrow}_{\lambda,i}$, and $E_{\mathrm{dir},\lambda,i}$ — cannot accurately represent both the highly collimated, beam-like character of direct radiation and the broader angular distribution of diffuse radiation within a single variable. Using these streams, the propagation of radiation through any given layer i — highlighted in orange in Fig. 2.13 — can then be described by the same fundamental processes that shape the full RTE:

(a) *Transmission:* Absorption and scattering out of the stream of interest (direct, diffuse downward, or diffuse upward) reduce the spectral irradiance as it propagates through the layer. The fraction of the incoming radiation that ultimately exits the layer in the same direction is determined by the layer's transmissivity. For direct radiation ($E_{\mathrm{dir},\lambda,i+1}$), this transmitted fraction is given by $t_{\mathrm{dir},i}$, whereas for diffuse radiation ($E^{\downarrow}_{\lambda,i+1}$ and $E^{\uparrow}_{\lambda,i}$), it is governed by t_i.

(b) *Scattering:* Scattering from other streams into the stream of interest increases the spectral irradiance as it propagates through the layer. Radiation is scattered both from the direct stream into the diffuse ones and between the two diffuse streams. The fractions of the incoming direct radiation ($E_{\mathrm{dir},\lambda,i+1}$) scattered into the upward ($E^{\uparrow}_{\lambda,i+1}$) and downward ($E^{\downarrow}_{\lambda,i}$) diffuse streams are given by $r_{\mathrm{dir},i}$ and $s_{\mathrm{dir},i}$, respectively. Similarly, the fraction of the incoming diffuse streams ($E^{\downarrow}_{\lambda,i+1}$ and $E^{\uparrow}_{\lambda,i}$) that is scattered into the opposite outgoing stream is determined by the layer's reflectivity r_i.

(c) *Emission:* Thermal radiation emitted by the layer increases the outgoing diffuse spectral irradiance in both directions. Since each atmospheric layer behaves as a graybody with a temperature $T > 0\,\mathrm{K}$, this contribution is given by the layer's emissivity,

$$e_i = a_i = 1 - t_i - r_i, \tag{2.77}$$

multiplied by the effective spectral blackbody irradiance $B_{\mathrm{eff},\lambda,i}$ emitted in the corresponding direction.

By combining these contributions, the radiative transfer through a single layer i of the vertical column can be expressed as

$$\underbrace{\begin{pmatrix} E^{\uparrow}_{\lambda,i+1} \\ E^{\downarrow}_{\lambda,i} \\ E_{\mathrm{dir},\lambda,i} \end{pmatrix}}_{=\vec{E}_{\lambda,\mathrm{out},i}} = \underbrace{\begin{pmatrix} t_i & r_i & r_{\mathrm{dir},i} \\ r_i & t_i & s_{\mathrm{dir},i} \\ 0 & 0 & t_{\mathrm{dir},i} \end{pmatrix}}_{=\mathbf{T}_{\lambda,i}} \cdot \underbrace{\begin{pmatrix} E^{\uparrow}_{\lambda,i} \\ E^{\downarrow}_{\lambda,i+1} \\ E_{\mathrm{dir},\lambda,i+1} \end{pmatrix}}_{=\vec{E}_{\lambda,\mathrm{in},i}} + \underbrace{\begin{pmatrix} e_i B^{\uparrow}_{\mathrm{eff},\lambda,i} \\ e_i B^{\downarrow}_{\mathrm{eff},\lambda,i} \\ 0 \end{pmatrix}}_{=\vec{B}_{\lambda,i}} . \tag{2.78}$$

The vector $\vec{E}_{\lambda,\text{out},i}$ on the left-hand side of this matrix equation contains all spectral irradiances exiting layer i. On the right-hand side, the matrix $\mathbf{T}_{\lambda,i}$ contains all the transport coefficients governing the absorption and scattering of the incoming radiation $\vec{E}_{\lambda,\text{in},i}$ as it propagates through the layer. Additionally, the vector $\vec{B}_{\lambda,i}$ accounts for the thermal radiation emitted into the two diffuse streams.

With boundary conditions specified at the top and bottom of the domain, the combined equations for all $N_z + 1$ layers of the atmospheric column form a system of coupled linear equations, which can be solved using various numerical methods. The required boundary conditions are determined by the incoming solar radiation at the top of the column and by surface reflection and emission at the bottom and are therefore given by

$$E_{\text{dir},\lambda,N_z+1} = E_{0,\lambda} \cos\theta_{\text{inc}} \tag{2.79}$$
$$\text{(incoming direct solar radiation at the top)},$$
$$E^{\downarrow}_{\lambda,N_z+1} = 0 \tag{2.80}$$
$$\text{(no incoming diffuse radiation at the top), and}$$
$$E^{\uparrow}_{\lambda,0} = A_g\,(E^{\downarrow}_{\lambda,0} + E_{\text{dir},\lambda,0}) + (1 - A_g)\,\pi B_\lambda(T_g) \tag{2.81}$$
$$\text{(reflection and emission at the ground)},$$

where $E_{0,\lambda}$ is the extraterrestrial solar spectral irradiance, θ_{inc} the solar zenith angle, A_g the ground albedo, and $\pi B_\lambda(T_g)$ the spectral irradiance emitted by a blackbody at ground temperature T_g.

The transport coefficients t_i, r_i, $t_{\text{dir},i}$, $r_{\text{dir},i}$ and $s_{\text{dir},i}$ required for solving the two-stream linear equation system depend on the optical properties of each layer. In the δ-Eddington approximation, they are obtained from the *Eddington approximation*, an analytical solution of the RTE assuming that the spectral radiance L_λ takes the form

$$L_\lambda = L_{\lambda,0} + \mu L_{\lambda,1}, \tag{2.82}$$

where $\mu = \cos\theta$. This implies that L_λ consists of an isotropic component ($L_{\lambda,0}$) and an additional linear term ($L_{\lambda,1}$) that introduces directionality, making L_λ most intense in the vertical direction ($\theta = 0°, 180°$), and weakest in the horizontal ($\theta = 90°$). Under this assumption, the Eddington approximation provides analytical expressions for each layer's reflectivity r_i, transmissivity t_i, direct transmissivity $t_{\text{dir},i}$, as well as the upward ($r_{\text{dir},i}$) and downward ($s_{\text{dir},i}$) scattered fractions of the direct radiation, that depend only on four parameters:

(a) the *solar incidence angle* $\mu_0 = \cos\theta_{\text{inc}}$,

(b) the *optical depth* $\Delta\tau = \beta_{\text{ext},\lambda}\,\Delta z$ of the layer,

(c) the *single-scattering albedo* $\omega_{0,\lambda}$, and

(d) the *asymmetry parameter* g, which characterizes the preferred scattering direction within the layer. It is defined as

$$g = \frac{1}{2}\int_{-1}^{1} \cos\Theta\, p(\cos\Theta)\, \mathrm{d}\cos\Theta \tag{2.83}$$

and represents the average value of $\cos\Theta$ for a given phase function $p(\cos\Theta)$, where Θ specifies the angle between the incident direction $\hat{\Omega}'$ and the scattered direction $\hat{\Omega}$ (Petty, 2006f). The value of g ranges from -1 to 1. A value of $g = 0$ corresponds to equal amounts of radiation scattered into the forward and backward directions. When $g > 0$, radiation is preferentially forward-scattered, with $g = 1$ indicating that all radiation continues exactly in the incident direction. Conversely, when $g < 0$, radiation is preferentially backward-scattered, with $g = -1$ meaning that all radiation is scattered directly backward.

Although these quantities can be inserted directly into the expressions for the transport coefficients provided by the Eddington approximation, the resulting coefficients become inaccurate when scattering in the layer is dominated by the Mie regime. This is because the assumption in Eq. (2.82) causes the Legendre expansion of the phase function $p(\cos\Theta)$ to be truncated after the linear term. As a result, the strong forward-scattering peak characteristic of Mie scattering, as discussed in Sect. 2.2.3, cannot be accurately represented in this approximation, leading to substantial inaccuracies in the computed transport coefficients. To address this issue, the large forward-scattered fraction of radiation is no longer treated as part of the scattered radiation but rather as transmitted radiation. To achieve this, the *δ-scaled scattering phase function* $p_\delta(\hat{\Omega}', \hat{\Omega})$ is introduced, which is given by

$$p_\delta(\hat{\Omega}', \hat{\Omega}) = p_\delta(\mu', \phi', \mu, \phi) = 4\pi f \delta(\mu - \mu')\delta(\phi - \phi') + (1 - f)\, p^*(\mu', \phi', \mu, \phi) \tag{2.84}$$

$$= p_\delta(\cos\Theta) = 2 f \delta(1 - \cos\Theta) + (1 - f)\, p^*(\cos\Theta)\,. \tag{2.85}$$

Here, the fraction f of the scattered radiation that remains in the forward direction is handled separately using two Dirac delta functions, which sharply peak at $\mu = \mu'$ and $\phi = \phi'$. The remaining fraction, $1 - f$, which accounts for scattering into all other directions, can then be described by a much smoother phase function $p^*(\mu', \phi', \mu, \phi)$ that no longer needs to capture the sharp forward peak. Since the phase function is often expressed in terms of the scattering angle Θ, Eq. (2.85) provides an equivalent formulation of Eq. (2.84), written as a function of $\cos\Theta$ instead of the directional unit vectors $(\hat{\Omega}', \hat{\Omega}) = (\mu', \phi', \mu, \phi)$. By substituting Eq. (2.84) into the RTE without the emission term, while also using Eq. (2.42) along with $\mu = \cos\theta$, we obtain

$$\frac{\mathrm{d}L_\lambda(\mu,\phi)}{\mathrm{d}s} = -\beta_{\mathrm{ext},\lambda} L_\lambda(\mu,\phi) + \frac{\omega_{0,\lambda}\beta_{\mathrm{ext},\lambda}}{4\pi} \int_0^{2\pi}\int_{-1}^{1} p_\delta(\mu', \phi', \mu, \phi) L_\lambda(\mu', \phi')\, \mathrm{d}\mu'\mathrm{d}\phi' \tag{2.86}$$

$$\begin{aligned}
&= -\beta_{\mathrm{ext},\lambda} L_\lambda(\mu,\phi) + \frac{\omega_{0,\lambda}\beta_{\mathrm{ext},\lambda}}{4\pi} \Bigg(4\pi f \int_0^{2\pi}\int_{-1}^{1} \delta(\mu - \mu')\delta(\phi - \phi') L_\lambda(\mu', \phi')\, \mathrm{d}\mu'\mathrm{d}\phi' \\
&\qquad + (1 - f)\int_0^{2\pi}\int_{-1}^{1} p^*(\mu', \phi', \mu, \phi) L_\lambda(\mu', \phi')\, \mathrm{d}\mu'\mathrm{d}\phi'\Bigg) \\
&= -\left(1 - f\omega_{0,\lambda}\right)\beta_{\mathrm{ext},\lambda} L_\lambda(\mu,\phi) + \frac{(1 - f)\omega_{0,\lambda}\beta_{\mathrm{ext},\lambda}}{4\pi} \int_0^{2\pi}\int_{-1}^{1} p^*(\mu', \phi', \mu, \phi) L_\lambda(\mu', \phi')\, \mathrm{d}\mu'\mathrm{d}\phi' \\
&= -\beta^*_{\mathrm{ext},\lambda} L_\lambda(\mu,\phi) + \frac{\omega^*_{0,\lambda}\beta^*_{\mathrm{ext},\lambda}}{4\pi} \int_0^{2\pi}\int_{-1}^{1} p^*(\mu', \phi', \mu, \phi) L_\lambda(\mu', \phi')\, \mathrm{d}\mu'\mathrm{d}\phi'\,.
\end{aligned} \tag{2.87}$$

In the last step of this calculation, the *δ-scaled extinction coefficient* $\beta^*_{\mathrm{ext},\lambda}$ and the *δ-scaled* *single-scattering albedo* $\omega^*_{0,\lambda}$ have been defined as

$$\beta^*_{\text{ext},\lambda} = \left(1 - f\omega_{0,\lambda}\right)\beta_{\text{ext},\lambda} \qquad \text{and} \qquad \omega^*_{0,\lambda} = \frac{(1-f)\omega_{0,\lambda}}{1-f\omega_{0,\lambda}}. \tag{2.88}$$

By examining Eq. (2.87), we see that when using the δ-scaled scattering phase function defined in Eq. (2.84) along with the definitions for $\beta^*_{\text{ext},\lambda}$ and $\omega^*_{0,\lambda}$, the RTE retains the same form as the original RTE without emission provided in Eq. (2.86). However, the forward-scattered fraction of the radiation is now incorporated into the extinction term, while the second term — describing radiation scattered from other directions (μ', ϕ') into the direction of interest (μ, ϕ) — is governed solely by the smoothed phase function $p^*(\mu', \phi', \mu, \phi)$, which no longer includes the strong forward peak. And since Eq. (2.87) maintains the same structure as Eq. (2.86), the delta-scaled quantities $\Delta\tau^* = \beta^*_{\text{ext},\lambda}\Delta z$, $\omega^*_{0,\lambda}$, and g^* can be used to calculate transport coefficients from the Eddington approximation whenever certain atmospheric constituents of a layer are subject to Mie scattering, with the *δ-scaled asymmetry parameter* g^* given by

$$g^* = \frac{g-f}{1-f}. \tag{2.89}$$

This expression is obtained by inserting Eq. (2.85) into Eq. (2.83). For completeness, note that the emission term was omitted in this derivation, as the Eddington approximation only describes the transport of radiation through an atmospheric layer, and not its source terms.

All in all, such a two-stream method that derives its transport coefficients from the Eddington approximation using δ-scaled optical properties is called a *δ-Eddington approximation* (Joseph et al., 1976). Unlike direct solutions of the RTE, it computes only upward and downward spectral irradiances, bypassing the expensive angular dependence of radiative transfer calculations. This omission not only reduces computational cost but also allows heating rates to be obtained directly using Eq. (2.74). Additionally, because the spectral irradiances in any given layer i depend only on those within the same vertical column, each atmospheric column can be treated independently. This structure facilitates model parallelization, further enhancing the δ-Eddington approximation's computational efficiency and explaining why it is still widely used in NWP models. However, as a 1D approximation, it does not account for horizontal energy transport and therefore neglects all 3D radiative effects.

Monte Carlo radiative transfer

A fundamentally different approach to solving the RTE is provided by Monte Carlo radiative transfer. Instead of seeking an analytical or deterministic solution, this method adopts a statistical perspective. To this end, individual photons are traced along their random paths through the atmosphere, from their emission or entry point until they are either absorbed or escape into space. Figure 2.14 shows the trajectory of one such photon on its way through a cloud field used in Chapter 4 of this work. Given a sufficiently large number of these trajectories, Monte Carlo radiative transfer enables the computation of both spectral radiances and irradiances in arbitrarily complex, inhomogeneous 3D atmospheres. In the following, the key elements

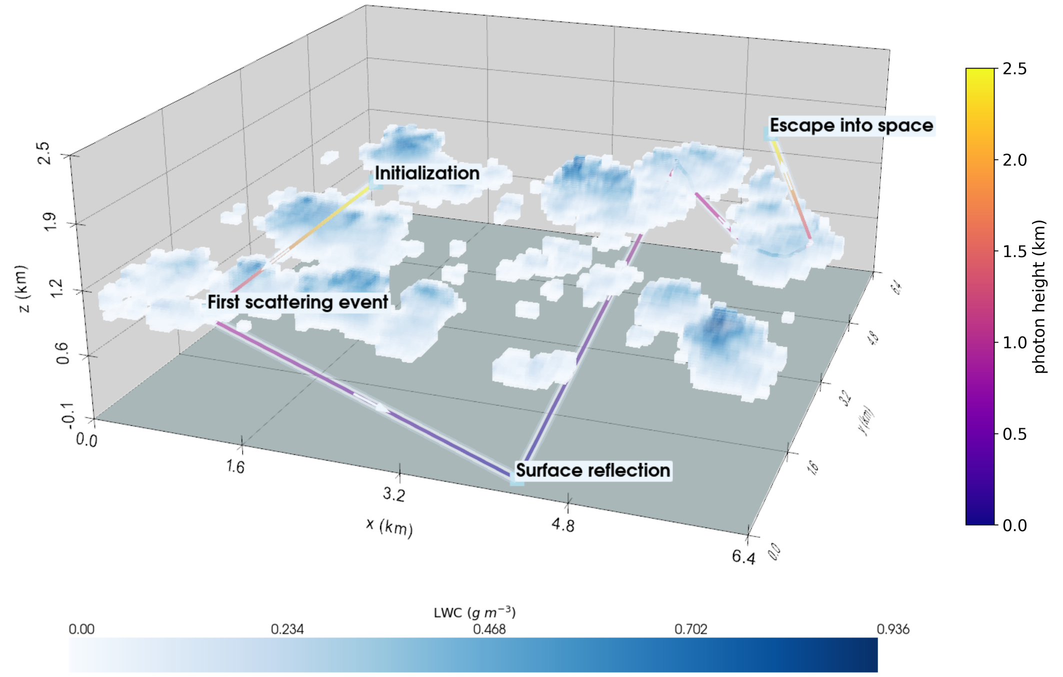

Figure 2.14: Visualization of a Monte Carlo-simulated photon originating from the Sun on its random path through the first cloud field of the shallow cumulus time series used in Chapter 4 of this work. The cloud field is illustrated using the liquid water content (LWC) of the individual grid boxes, with all grid boxes below a threshold of $10^{-10}\,\mathrm{g\,m^{-3}}$ hidden. The photon trajectory is shown color-coded by height, with arrows indicating the direction of propagation. Additionally, four key events along the path are annotated: the photon's initialization at $z = 2.5\,\mathrm{km}$ with a zenith angle of 50° and an azimuthal direction toward the west; the first scattering event; the point where the photon is reflected at the surface; and its escape back to space. Note that the atmosphere was intentionally limited to a shallow domain with an upper boundary at $z = 2.5\,\mathrm{km}$ for this visualization.

of this photon-tracing approach are summarized, outlining both how individual photons are traced on their way through the atmosphere and how an ensemble of the resulting trajectories can be used to compute various radiative quantities.

Let us begin by describing how individual photons are traced. Similar to other radiative transfer methods, Monte Carlo simulations are performed on a grid that divides the atmosphere into a set of finite volumes with constant optical properties. Within this grid, the tracing of photons begins with their generation, where each photon is assigned an initial location and direction, both depending on the corresponding atmospheric wavelength regime. In the solar spectral range, the photons originate from the Sun and enter the atmosphere at its upper boundary. Accordingly, they are initialized at random locations along the top of the grid, with a propagation direction determined by the Sun's zenith and azimuth angles. The photon in Fig. 2.14, for instance, is initialized at a height of $z = 2.5\,\mathrm{km}$, with a zenith angle of 50° and an azimuthal direction toward the west. In the thermal spectral range, on the other hand, Earth and its atmosphere serve as the photon sources. Photons are therefore initiated at random locations within the atmosphere, with probabilities determined by both local emissivity and temperature, so that more photons are emitted from regions with higher temperatures and larger emissivities. And since thermal emission is isotropic, the initial direction of these photons is chosen randomly.

Once generated, the photons then propagate in their initial direction until they are either scattered or absorbed — a process collectively referred to as extinction. The key point of the Monte Carlo method is that this distance to extinction is chosen randomly for each photon, based on the extinction probability along its path. This probability can be derived by integrating Eq. (2.54), which governs the change $\mathrm{d}L_{\lambda,\mathrm{ext}}$ in spectral radiance due to extinction, over a finite path from s_0 to s, assuming $L_\lambda(s_0) = L_{\lambda,0}$, which leads to

$$\int_{L_{\lambda,0}}^{L_\lambda} \frac{\mathrm{d}L'_\lambda}{L'_\lambda} = -\int_{s_0}^{s} \beta_{\mathrm{ext},\lambda}(s')\,\mathrm{d}s' \quad \Rightarrow \quad L_\lambda(s) = L_{\lambda,0}\exp\left(-\int_{s_0}^{s} \beta_{\mathrm{ext},\lambda}(s')\,\mathrm{d}s'\right) = L_{\lambda,0}\exp(-\tau)\,. \quad (2.90)$$

In the last step of this derivation, the *optical depth* has been defined as

$$\tau = \int_{s_0}^{s} \beta_{\mathrm{ext},\lambda}(s')\,\mathrm{d}s'\,. \quad (2.91)$$

Eq. (2.90) is known as the *Beer–Bouguer–Lambert extinction law* and describes the exponential reduction of an initial spectral radiance $L_{\lambda,0}$ along a path through a medium characterized by the extinction coefficient $\beta_{\mathrm{ext},\lambda}(s)$. The ratio $\frac{L_\lambda(s)}{L_{\lambda,0}}$ gives the fraction of transmitted to initial spectral radiance, which, for a single photon, can be interpreted as its survival probability $P_{\mathrm{sur}} = \exp(-\tau)$. Conversely, the function $P_{\mathrm{ext}}(\tau) = 1 - \exp(-\tau)$ denotes the extinction probability of the photon. As a cumulative distribution function, it converges toward a probability of 100% for large optical depths. Solving this function for τ and inserting uniformly distributed random numbers $P \in [0,1]$ yields random optical path lengths given by

$$\tau = -\ln(1-P)\,, \quad (2.92)$$

which collectively follow the Beer–Bouguer–Lambert extinction law. Hence, these optical path lengths provide exactly the random distances that photons travel before extinction occurs. To determine the extinction location of a photon based on such a random optical path length, the optical depth is accumulated along the photon's trajectory until it matches the sampled value. In the example shown in Fig. 2.14, this point is reached after the photon enters one of the clouds in the southwestern part of the domain. The fate of the photon is then determined using another random number $P' \in [0,1]$. If $P' > \omega_{0,\lambda}$, the photon is absorbed and no longer traced. If $P' \le \omega_{0,\lambda}$, it is scattered, which is the case encountered by the photon in our example. In that case, a new direction must be assigned to the photon. This direction is sampled from the scattering phase function $p(\theta', \phi', \theta, \phi)$, which specifies the probability that a photon arriving from direction (θ', ϕ') is scattered into direction (θ, ϕ). Once this new direction has been assigned, a new τ is drawn using Eq. (2.92), and the photon continues to propagate until it is again either absorbed or scattered. This process is repeated until the photon either escapes the domain or is absorbed, resulting in trajectories like the one shown in Fig. 2.14.

A special case arises when the photon reaches the surface before extinction occurs. In this case, its fate is determined by another random number $P'' \in [0,1]$ and the ground albedo A_g. If

$P'' > A_g$, the photon is absorbed by the surface; otherwise, it is reflected and assigned a new direction. Such a surface reflection also occurred for the photon depicted in Fig. 2.14, after it escaped the cloud where it experienced its first scattering event. Similar to the scattering direction discussed earlier, the new direction assigned to such a reflected photon is sampled from the *bi-directional reflectance distribution function* $\mathrm{BRDF}(\theta_i, \phi_i, \theta_o, \phi_o)$ of the surface, which quantifies the probability of reflection into direction (θ_o, ϕ_o) based on the incoming direction (θ_i, ϕ_i).

Ultimately, all photons are either absorbed or escape into space. The photon in Fig. 2.14 is an example of the latter, as it was eventually scattered back into space. To calculate radiative quantities from the resulting photon pathways, a large number of photons must be traced and counted whenever they pass through the desired location. For example, to calculate the downward spectral irradiance $E_\lambda^\downarrow$ at the top of a given grid box in the solar spectral range, all photons crossing the top face of that grid box from above must be counted. The number N_s of these photons per surface area A_s, divided by the total number N_t of photons launched across the top of the domain with area A_t, then defines the fraction of the extraterrestrial solar spectral irradiance $E_{0,\lambda} \cos\theta_{\mathrm{inc}}$ passing through that grid box from above — and thus the downward spectral irradiance at that point, which is therefore obtained through

$$E_\lambda^\downarrow = E_{0,\lambda} \cos\theta_{\mathrm{inc}} \frac{A_t}{N_t} \frac{N_s}{A_s} . \tag{2.93}$$

With the Monte Carlo method, these spectral irradiances can be calculated for every face of a grid box, enabling the computation of fully three-dimensional heating rates using Eq. (2.73) — something not possible in 1D solvers such as the δ-Eddington approximation discussed previously. Additionally, the algorithm can incorporate arbitrarily complex scattering and surface reflection processes into the calculation of these quantities, provided the corresponding probability distributions for sampling new directions are known. Moreover, since the optical path lengths determining photon propagation are evaluated using local extinction coefficients $\beta_{\mathrm{ext},\lambda}$, the method naturally extends to fully vertically and horizontally inhomogeneous domains, making Monte Carlo radiative transfer suitable for solving the full RTE in arbitrarily complex 3D atmospheres.

However, as a statistical method, Monte Carlo simulations are also inherently noisy. This noise can be reduced by increasing the number of photons used in the simulation, which comes at the cost of increased computational demand but also enables arbitrarily precise results in the limit of an infinite number of photons. More specifically, the standard deviation σ of a radiative quantity with mean value μ can be shown to be inversely proportional to the square root of the number N_s of photons contributing to the result, provided that this number is much smaller than the total number N_t of photons used in the simulation ($N_s \ll N_t$). In this case, σ can be approximated by

$$\frac{\sigma}{\mu} \approx \frac{1}{\sqrt{N_s}} . \tag{2.94}$$

This behavior follows directly from the statistical nature of the Monte Carlo simulations, which can be viewed as a series of yes/no experiments in which individual photons either contribute to the estimate of a given radiative quantity or not. Specifically, Eq. (2.94) resembles the result of a Poisson distribution with parameter $\lambda = p\,N_t = N_s$, where p denotes the probability of a photon contributing to the result, provided that N_t is sufficiently large so that the number of contributing photons is given by $N_s = p\,N_t$. For such a Poisson distribution, the standard deviation $\sigma = \sqrt{\lambda}$ scales with the square root of the mean $\mu = \lambda$ (Feller, 1950), just as expressed in Eq. (2.94), with $\lambda = N_s$. A consequence of this relationship is that to double the accuracy of a result, four times as many photons are needed. The number of photons required to achieve a given level of accuracy can therefore become substantial. For instance, assume that about half the photons are absorbed before reaching the surface, as indicated by the solar component of Fig. 2.7, and that one aims for a 1% relative uncertainty in the downward solar spectral irradiance at the surface. Then roughly $N_s = (1/0.01)^2 = 10\,000$ photons would need to reach the surface per vertical column, implying that approximately $N_t \approx 2N_sN_xN_y = 20\,000\,N_xN_y$ photons must be launched. For a model domain consisting of 100 000 vertical columns, this would already require tracing two billion photons, illustrating that highly accurate Monte Carlo simulations can be extremely expensive in terms of computational cost. Yet in the limit of a large number of photons, they also yield the most accurate solutions possible and are therefore ideally suited for benchmark calculations — which is also exactly what they are used for in this thesis.

The TenStream solver

The final approximate solution to the RTE discussed in this subsection is the TenStream solver. It extends the two-stream method presented earlier to three dimensions by using ten diffuse streams instead of two, and three direct streams instead of one, to represent the transport of radiative energy through the atmosphere. Like two-stream methods, it directly computes spectral irradiances, thereby avoiding the expensive angular component of radiative transfer calculations. Furthermore, as a 3D solver, the TenStream method computes irradiances for all faces of a grid box rather than just its top and bottom, enabling the calculation of fully three-dimensional heating rates using Eq. (2.73). As an approximate solution to the full 3D RTE, the TenStream solver further offers greater accuracy than the 1D δ-Eddington approximation, while remaining significantly less computationally demanding than a Monte Carlo simulation performed with a sufficiently large number of photons. The TenStream solver thus provides a trade-off between accuracy and computational cost. Since it also forms the foundation of the dynamic TenStream solver developed in this thesis, it is not discussed in detail here, but instead introduced more thoroughly in the following chapter.

Chapter 3

Toward dynamic treatment of radiation

From here on, the entirety of this thesis will be centered around the development of a fast, yet accurate three-dimensional radiative transfer solver, its evaluation, and the demonstration of its use in subkilometer-scale numerical simulations. The development of this new solver was driven by two main constraints: On the one hand, it should compute radiative fluxes and heating rates at a significantly faster speed than other inter-column 3D solvers. On the other hand, it should also provide a noticeable improvement in terms of accuracy over currently employed 1D schemes. However, any three-dimensional radiative transfer solver that aims toward the speed of currently employed 1D approximations, while also attempting to capture the full 3D solution as accurately as possible, will naturally have some limitations unless it is based on computational breakthroughs. This also holds up for the work presented in this thesis, which is based on the TenStream solver, an already relatively fast 3D method for the calculation of radiative fluxes and heating rates, which achieves its speed by simplifying the expensive angular part of 3D radiative transfer calculations. Based upon this work, this chapter presents the dynamic TenStream solver, the central development of this thesis, which aims to provide an even more attractive speed-accuracy trade-off by introducing a time-stepping scheme and incomplete solves.

The content of this chapter was published in Sect. 2 of Maier et al. (2024).

3.1 The original TenStream model

We build upon the TenStream model (Jakub and Mayer, 2015), which extends the established two-stream formulation to three dimensions. Figure 3.1 shows the definition of its streams, i.e., radiative fluxes (in units of W), for a single rectangular grid box, with the indices (i, j, k) indicating the position of the box in a Cartesian grid of size $N_x \times N_y \times N_z$.

Ten streams (Φ_0, Φ_1, …, Φ_9; depicted in blue) are used to describe the 3D transport of diffuse radiation. As in the two-stream formulation, two of them (Φ_0 (upward) and Φ_1 (downward)) characterize the transport in the vertical, whereas four additional streams are introduced to describe the transport in each of the two additional horizontal dimensions. The

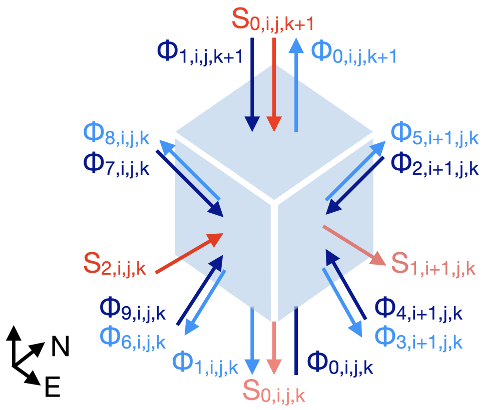

Figure 3.1: Schematic illustration of all fluxes entering and exiting a rectangular grid box (i, j, k) in the TenStream solver and their respective indices. Diffuse fluxes are shown in blue, while fluxes of direct radiation are displayed in red. Fluxes entering the grid box are shown in a darker tone than the ones exiting. The two pairs of diffuse fluxes on each of the sideward-oriented faces of the cuboid point into and out of the upper and lower hemispheres, respectively. Fluxes at the sides of the cuboid facing to the north and west are not visible.

transport of direct radiation, i.e., radiation originating from the Sun that has not yet interacted with the atmosphere, is treated separately using the three additional streams S_0, S_1 and S_2, one for each dimension (shown in red in Fig. 3.1). Using these streams, the radiative transport through a single grid box (i, j, k) in the case of the Sun shining from the southwest can be expressed by the following matrix equation:

$$\underbrace{\begin{pmatrix} \Phi_{0,i,j,k+1} \\ \Phi_{1,i,j,k} \\ \vdots \\ \Phi_{9,i,j+1,k} \\ S_{0,i,j,k} \\ S_{1,i+1,j,k} \\ S_{2,i,j+1,k} \end{pmatrix}}_{=\vec{\Phi}_{\mathrm{out},i,j,k}} = \underbrace{\begin{pmatrix} a_{00,i,j,k} & \dots & a_{09,i,j,k} & b_{00,i,j,k} & b_{01,i,j,k} & b_{02,i,j,k} \\ a_{10,i,j,k} & \dots & a_{19,i,j,k} & b_{10,i,j,k} & b_{11,i,j,k} & b_{12,i,j,k} \\ \vdots & & \vdots & \vdots & \vdots & \vdots \\ a_{90,i,j,k} & \dots & a_{99,i,j,k} & b_{90,i,j,k} & b_{91,i,j,k} & b_{92,i,j,k} \\ 0 & \dots & 0 & c_{00,i,j,k} & c_{01,i,j,k} & c_{02,i,j,k} \\ 0 & \dots & 0 & c_{10,i,j,k} & c_{11,i,j,k} & c_{12,i,j,k} \\ 0 & \dots & 0 & c_{20,i,j,k} & c_{12,i,j,k} & c_{22,i,j,k} \end{pmatrix}}_{=\boldsymbol{T}_{i,j,k}} \cdot \underbrace{\begin{pmatrix} \Phi_{0,i,j,k} \\ \Phi_{1,i,j,k+1} \\ \vdots \\ \Phi_{9,i,j,k} \\ S_{0,i,j,k+1} \\ S_{1,i,j,k} \\ S_{2,i,j,k} \end{pmatrix}}_{=\vec{\Phi}_{\mathrm{in},i,j,k}} + \underbrace{\begin{pmatrix} e_{0,i,j,k} \cdot B_{\mathrm{eff},0,i,j,k} \\ e_{1,i,j,k} \cdot B_{\mathrm{eff},1,i,j,k} \\ \vdots \\ e_{9,i,j,k} \cdot B_{\mathrm{eff},9,i,j,k} \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{=\vec{B}_{i,j,k}} . \tag{3.1}$$

Note the following about this equation:

- The vector $\vec{\Phi}_{\mathrm{in},i,j,k}$ consists of all the radiative fluxes entering grid box (i, j, k). For reasons of clarity, will use the expression $\Phi_{\mathrm{in},m,i,j,k}$ to address an individual entry m of this vector, implying that, for example, $\Phi_{\mathrm{in},10,i,j,k}$ equals $S_{0,i,j,k+1}$ in the case of the Sun shining from the southwest.

- The matrix $\mathbf{T}_{i,j,k}$ describes the scattering and absorption of the ingoing radiation $\vec{\Phi}_{\mathrm{in},i,j,k}$ on its way through the grid box, with $a_{00,i,j,k}$, for example, quantifying the fraction of the upward flux entering the grid box at the bottom $(\Phi_{0,i,j,k})$ that exits the box in the same direction through the top $(\Phi_{0,i,j,k+1})$. While the "a" coefficients describe the transport of diffuse radiation, the "b" coefficients quantify the fraction of direct radiation that gets scattered, thus providing a source term for the 10 diffuse streams. The "c" coefficients describe the amount of direct radiation that is transmitted through the grid box without interacting with the medium. All of these transport coefficients depend on the optical properties (optical thickness, single-scattering albedo, asymmetry parameter, grid-box aspect ratio and angle of solar incidence) of the particular grid box. They are precomputed using Monte Carlo methods and stored in lookup tables (Jakub and Mayer, 2015). We will use the expression $t_{mn,i,j,k}$ to refer to the entry in row m and column n of the full matrix $\mathbf{T}_{i,j,k}$.

- The vector $\vec{B}_{i,j,k}$ quantifies the amount of thermal radiation that is emitted in the direction of every one of the 10 diffuse streams. Its entries $B_{m,i,j,k}$ are calculated by multiplying the black-body radiation that is emitted in the corresponding direction ($B_{\text{eff},m,i,j,k}$) by the emissivity of the grid box in that direction. According to Kirchhoff's law, this emissivity of a grid box in a certain direction is the same as the absorptivity of radiation coming from that direction, which, in turn, is 1 minus the transmittance in that direction. For example, the emissivity $e_{0,i,j,k}$ of grid box (i, j, k) in the upward direction is equal to the fraction of the downward-facing radiative flux $\Phi_{1,i,j,k+1}$ that is absorbed on the way through that grid box, which, in turn, is 1 minus the sum of all fractions $a_{n1,i,j,k}$ of $\Phi_{1,i,j,k+1}$ exiting grid box (i, j, k), i.e.,
$$e_{0,i,j,k} = 1 - \sum_{n=0}^{9} a_{n1,i,j,k},$$
where $a_{n1,i,j,k}$ refers to the corresponding entries in the second column of matrix $\mathbf{T}_{i,j,k}$.
- The vector $\vec{\Phi}_{\text{out},i,j,k}$ consists of all radiative fluxes exiting the grid box (i, j, k). For every stream, it contains all the radiative energy that has not interacted with the grid box on its way through plus, in case of the diffuse streams, the radiative energy that has been scattered and emitted in that direction along that way. Similar to the ingoing flux vector, we use the expression $\Phi_{\text{out},m,i,j,k}$ to refer to an entry m of the full vector $\vec{\Phi}_{\text{out},i,j,k}$.

Combined, the equations for all the $N_x \times N_y \times N_z$ grid boxes make up a large system of coupled linear equations that must be provided with boundary conditions at the edges of the domain. At the top and surface, these are determined by the incoming solar radiation on one side and by ground reflection and emission on the other. Specifically,

$$\begin{aligned} S_{0,i,j,N_z+1} &= E_0 \cos\theta_{\text{inc}}\, \Delta x\, \Delta y \\ &\quad \text{(incoming solar radiation at the top) and} \\ \Phi_{0,i,j,0} &= A_g\,(\Phi_{1,i,j,0} + S_{0,i,j,0}) + (1 - A_g)\,\pi\, B_g\, \Delta x\, \Delta y \\ &\quad \text{(reflection and emission at the ground).} \end{aligned}$$

Here, E_0 denotes the extraterrestrial solar irradiance (in units of $\mathrm{W\,m^{-2}}$), θ_{inc} the solar zenith angle, A_g the ground albedo, B_g the emitted black-body radiance of the ground (in units of $\mathrm{W\,m^{-2}\,sr^{-1}}$), and Δx and Δy the horizontal grid box lengths (in units of m). The boundary conditions employed at the sides of the domain depend on the model configuration and can be either cyclic or provided by neighboring subdomains. The resulting system of linear equations can then be solved by various numerical methods. In the original TenStream solver, they are provided by the parallel linear algebra library PETSc (Balay et al., 2023).

3.2 Introducing time-stepping and incomplete solves: the dynamic TenStream solver

However, solving this large system of linear equations is a difficult task, especially when it needs to be parallelized for large NWP simulations. The main reason behind this difficulty is the fundamentally different approaches to how radiation and dynamics are treated in numerical models. On the one hand, solving the equations of motion that govern advection in the dynamical core of an NWP model represents an initial value problem that has no known analytical solution. Hence, these equations are discretized in space and time and solved by a time-stepping scheme, where model variables are gradually propagated forward in time by applying the discretized equations to values obtained at previous time steps (Holton and Hakim, 2013). An individual grid box thereby only needs information about itself and its nearby surroundings, facilitating model parallelization. Radiative transfer, on the other hand, is treated as a boundary value problem, where information is not gradually propagated through the domain but rather spreads almost instantaneously at the speed of light, involving the entire model grid. Three-dimensional radiative transfer can thus easily break model parallelization, as a radiative flux at any position in the domain can theoretically depend on all other radiative fluxes throughout the domain. This can be seen by looking at the coupled structure of the equations in the original TenStream solver in Eq. (3.1).

3.2.1 The Gauß–Seidel method

We tackle this problem by treating radiation similarly to initial value problems. To this end, we build upon the TenStream linear-equation system revisited in Sect. 3.1 and examine its solution with the Gauß–Seidel method, as described in, e.g., Wendland (2017). According to this iterative method, a system of linear equations must be transformed in such a way that one equation is solved for every unknown variable. This form is given by the equations in Eq. (3.1), with the unknown variables being all the radiative fluxes in the entire domain. Providing first guesses for all of these variables, one then iterates through all these equations and sequentially updates all the radiative fluxes on the left-hand sides of the equations by applying either the first guess or, if already available, the updated values to the corresponding variables on the right-hand sides of the equations. Applied to the TenStream equations, this means that one gradually iterates through all the grid boxes of the entire domain. For every grid box, one then calculates updated values for the outgoing fluxes $\Phi^{(l+1)}_{\text{out},m,i,j,k}$ on the left-hand side of Eq. (3.1) by applying either the already updated ingoing fluxes $\Phi^{(l+1)}_{\text{in},m,i,j,k}$ or, if those are not yet available, their values $\Phi^{(l)}_{\text{in},m,i,j,k}$ from the previous Gauß–Seidel iteration step to the variables on the right-hand sides of the equations. This is mathematically expressed as

$$\Phi^{(l+1)}_{\mathrm{out},m,i,j,k} = \sum_{n=0}^{9+3} t_{mn,i,j,k} \begin{cases} \Phi^{(l+1)}_{\mathrm{in},n,i,j,k} & \text{if } \Phi^{(l+1)}_{\mathrm{in},n,i,j,k} \text{ has already been calculated} \\ \Phi^{(l)}_{\mathrm{in},n,i,j,k} & \text{otherwise} \end{cases} + B_{m,i,j,k}\,. \tag{3.2}$$

Here, the indices m and n denote an individual entry of the outgoing flux vector $\vec{\Phi}_{\mathrm{out},i,j,k}$, the ingoing flux vector $\vec{\Phi}_{\mathrm{in},i,j,k}$ or the thermal source vector $\vec{B}_{i,j,k}$, whereas l quantifies the Gauß–Seidel iteration step and $t_{mn,i,j,k}$ refers to the corresponding entry in matrix $\mathbf{T}_{i,j,k}$ in Eq. (3.1). Completing this procedure for all the grid boxes and boundary conditions accomplishes one Gauß–Seidel iteration. One can then repeat this procedure with the updated radiative fluxes serving as the new first guess until the values eventually converge to the solution of the linear-equation system. The thermal source terms are not part of the first guess and have to be calculated from scratch, following the pattern outlined in Sect. 3.1, before starting with the Gauß–Seidel algorithm.

3.2.2 Dynamic treatment of radiation

To significantly speed up 3D radiative-transfer calculations, we apply the Gauß–Seidel method in combination with two main concepts: a time-stepping scheme and incomplete solves.

To introduce the time-stepping scheme, we make use of the fact that the Gauß–Seidel algorithm requires us to choose an initial guess for where to start. So, instead of solving the whole TenStream linear-equation system from scratch every time, we use the result obtained at the previous call of the radiation scheme as a starting point for the algorithm. Assuming that the field of optical properties determining the radiative fluxes has not changed fundamentally between two calls of the radiation scheme, this first guess should already be a good estimator of the final result. However, for the very first call of the radiation scheme, we cannot use a previously calculated result. In order to choose a reasonable starting point of the algorithm for this first call as well, we could use a full TenStream solve. However, such a solve would be computationally expensive and rely on numerical methods provided by the PETSc library, which we want to get rid of with our new solver to allow for easier integration into operational models. So, instead of performing a full TenStream calculation, we decided to solve the TenStream linear-equation system for a clear-sky situation as a starting point. This is the spin-up mentioned in Fig. 3.2. Since there is no horizontal variability in the cloud field in a clear-sky situation and our model does not feature any horizontal variability in the background atmosphere, we can perform this calculation for a single vertical column at a dramatically increased speed compared to a calculation involving the entire model grid. We cannot use a 1D solver for that, however, because we also need to pass initial values to the sideward-facing fluxes in the TenStream equation system. Assigned to the radiative fluxes of all vertical columns in the entire domain, these values then provide first guesses for all the TenStream variables that can be assumed to be much closer to the final result than starting with values of zero — even if the background atmosphere was not horizontally homogeneous and we would have to take the average of that background first.

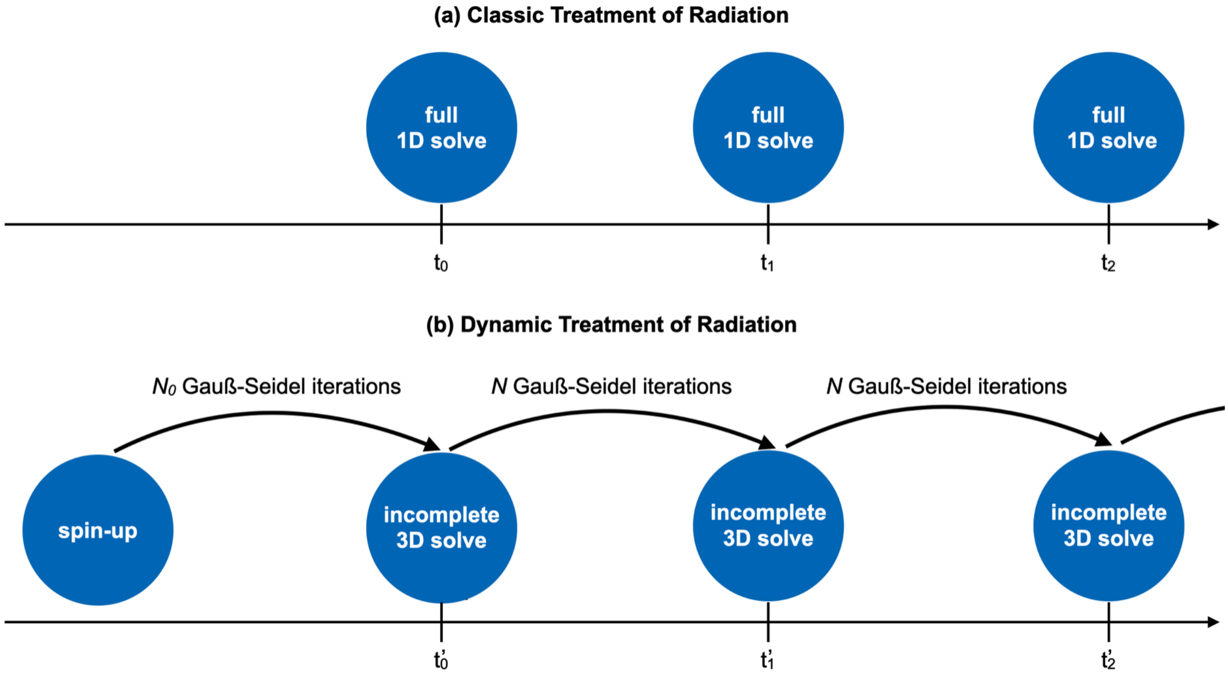

Figure 3.2: Schematic illustration of the dynamic treatment of radiation compared to the classic treatment. Instead of performing full 1D solves from scratch every time the radiation scheme is called, we use the result obtained at the last call as a starting point for an incomplete 3D solve, adjusting the previously calculated radiative fluxes toward the new full 3D solution.

Based on the idea that the radiative field does not fundamentally change between two calls of the radiation scheme, we furthermore just perform a limited number N of iterations of the Gauß–Seidel algorithm every time the radiation scheme is called, essentially not letting it fully converge. Unless the radiative fluxes have changed dramatically compared to the last calculation, adjusting the variables toward the new solution should already provide a good approximation of the full solution, especially since it incorporates inter-column 3D effects, unlike the 1D independent-column solutions used nowadays.

The combination of these two efforts is visualized in Fig. 3.2. Instead of calculating a full 1D solution from scratch every time radiation is called, our dynamic approach uses the previously obtained result as the starting point of a new incomplete 3D solve. This treatment of radiation puts it much closer to the way initial value problems like advection in the dynamical core of an NWP model are handled. Both use previously calculated results to update their variables. And, looking at an individual grid box, updating the outgoing fluxes by applying Eq. (3.2) only requires access to the fluxes entering that exact same grid box and thus only to neighboring values, just like in the discretized equations describing advection in the dynamical core of an NWP model.

But, even though the calculation of updated outgoing fluxes only requires access to fluxes entering the exact same grid box, this update process can indeed involve more distant grid boxes, since their calculation uses ingoing fluxes calculated in the very same Gauß–Seidel iteration wherever possible. And, since these ingoing fluxes are outgoing fluxes of a neighboring

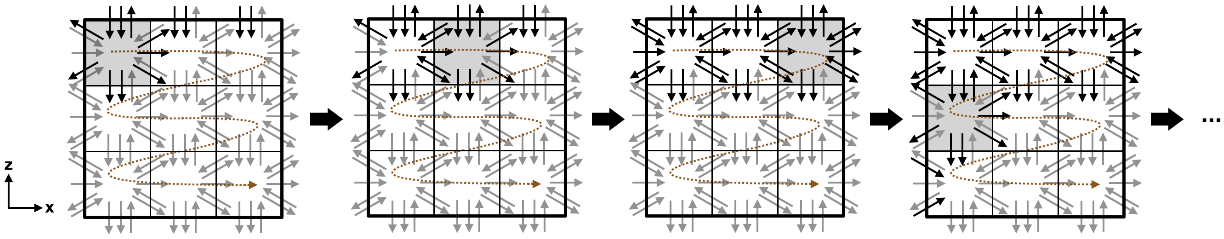

Figure 3.3: Two-dimensional schematic illustration of the first four steps of a Gauß–Seidel iteration, showing both diffuse and direct TenStream fluxes in the case of the Sun shining from the west or left-hand side. As one sequentially iterates through the grid boxes, ingoing fluxes are used to update the outgoing fluxes of the corresponding grid box (highlighted in gray). Gray arrows, in contrast to black arrows, indicate fluxes that have not yet been updated in this Gauß–Seidel iteration. Ingoing fluxes at the domain borders are dependent on the type of boundary conditions used. For this schematic, we applied periodic boundary conditions in the horizontal direction, while fluxes entering at the top of the domain are updated right from the beginning.

grid box that may have also been calculated using already updated radiative fluxes, information can spread across the domain wherever possible, involving, e.g., entire subdomains in NWP models. This is visualized in Fig. 3.3, which shows the first few steps of a Gauß–Seidel iteration in two dimensions only. Looking, for example, at the third step, outgoing fluxes of the upper-right grid box (highlighted in gray) are updated using the corresponding ingoing fluxes. Thereby, the ingoing flux of direct radiation entering the grid box on the left-hand side, for example, already contains radiative transfer through the two grid boxes on its left-hand side. This shows that the iteration direction through the grid boxes within a Gauß–Seidel iteration is crucial, as information can spread much faster in the direction one iterates through the grid boxes. Since the Gauß–Seidel algorithm allows us to freely choose the order in which to proceed through the system of linear equations, we can use this order to our advantage. First, we use the fact that, whereas diffuse radiation spreads into all directions simultaneously, direct radiation clearly propagates in the direction of the Sun. Hence, for the solar spectral range, we first iterate through the grid boxes in the direction given by solar incidence in the horizontal and then from top to bottom in the vertical, as indicated by the dashed brown arrow in Fig. 3.3. In contrast to this 2D example, both horizontal dimensions are affected by the position of the Sun in the fully 3D case, of course. If the Sun is shining from the southwest, for example, we would hence first iterate from south to north and from west to east in the horizontal before iterating from top to bottom. In the thermal spectral range, however, emitted radiation is larger in the lower part of the domain due to the vertical temperature gradient in the atmosphere. Hence, we iterate from bottom to top in the vertical there. Independent of the spectral range, however, we still need to consider that diffuse radiation spreads in all directions simultaneously, which we do not account for by using a fixed iteration direction. Thus, every time we finish iterating through all the grid boxes, which completes a Gauß–Seidel iteration step, we reverse the direction of iteration in all three dimensions to not favor the propagation of information in one direction.

Combined, these efforts should allow us to very efficiently calculate radiative transfer in three dimensions. First, the time-stepping scheme enables us to already start with a reliable first guess instead of calculating everything from scratch. Next, we speed up the rate of convergence by choosing a proper order in which to proceed through the linear-equation system, taking parameters such as the current angle of solar incidence into account. Since the updated solution should not be markedly different from the previous one, we furthermore only perform a limited number of Gauß–Seidel iterations, essentially exiting the algorithm before full convergence is reached, noting that an incomplete 3D solution should still be better than a 1D solution that neglects all 3D effects, as we will also see later on (in Chapter 4). And, finally, updating the outgoing radiative fluxes of any grid box within a Gauß–Seidel iteration just requires access to the fluxes entering the exact same grid box, which facilitates model parallelization. Implemented into the method, incomplete dynamic TenStream solves, each with N Gauß–Seidel iterations, would then be calculated in parallel for the different subdomains, with communication between these subdomains ideally taking place just once afterward, at the end of the radiation scheme call. In this case, the spread of information would be limited to the scopes of the individual subdomains for every call of the radiation scheme.

3.2.3 Calculation of heating rates

In the end, though, we are not just interested in calculating radiative fluxes. We are especially interested in computing 3D heating rates. They quantify local changes in temperature with time due to sources and sinks of radiative energy in the atmosphere and can be calculated using the net irradiance divergence, following the relationship

$$\frac{\partial T}{\partial t} = \frac{1}{\rho\, c_p} \vec{\nabla} \cdot \vec{E}_{\text{net}} = \frac{1}{\rho\, c_p} \left(\frac{\partial E_{\text{net},x}}{\partial x} + \frac{\partial E_{\text{net},y}}{\partial y} + \frac{\partial E_{\text{net},z}}{\partial z} \right) \tag{3.3}$$

(Mayer, 2018). Here, T denotes the temperature, t the time, ρ the air density, c_p the specific heat capacity of air at constant pressure and $\vec{E}_{\text{net}}$ the net irradiance (in units of $\text{W}\,\text{m}^{-2}$), with components $E_{\text{net},x}$, $E_{\text{net},y}$ and $E_{\text{net},z}$ when expressed in Cartesian coordinates. When applied to the TenStream fluxes (in units of W) outlined in Sect. 3.1, we have to find expressions for the net flux in all three dimensions and then divide these by the area of the grid box surface they refer to. For the calculation of net fluxes, we have to recall that TenStream features two streams to describe the transport of diffuse radiation on each of its sides. Since these two streams describe the flux entering and exiting a grid box in the upper and lower hemispheres, respectively, the total flux entering or exiting a grid box on one of its sides is given by the sum of these two streams. The net flux in any of the three dimensions is thus given by adding up all diffuse and direct fluxes entering the grid box in that dimension and subtracting those exiting it in the very same dimension. The heating rate of a grid box can thus be expressed as

$$\begin{aligned}\left(\frac{\Delta T}{\Delta t}\right)_{i,j,k} = \frac{1}{\rho\, c_p} \Bigg[& \frac{1}{\Delta x} \frac{1}{\Delta y\, \Delta z} \Bigg(\underbrace{\sum_{m=2}^{5} \left(\Phi_{\text{in},m,i,j,k} - \Phi_{\text{out},m,i,j,k}\right)}_{\substack{\text{net diffuse radiative flux}\\ \text{in x direction}}} + \underbrace{\left(\Phi_{\text{in},11,i,j,k} - \Phi_{\text{out},11,i,j,k}\right)}_{\substack{\text{net direct radiative flux}\\ \text{in x direction}}} \Bigg) \\ & + \frac{1}{\Delta y} \frac{1}{\Delta x\, \Delta z} \Bigg(\underbrace{\sum_{m=6}^{9} \left(\Phi_{\text{in},m,i,j,k} - \Phi_{\text{out},m,i,j,k}\right)}_{\substack{\text{net diffuse radiative flux}\\ \text{in y direction}}} + \underbrace{\left(\Phi_{\text{in},12,i,j,k} - \Phi_{\text{out},12,i,j,k}\right)}_{\substack{\text{net direct radiative flux}\\ \text{in y direction}}} \Bigg)\end{aligned}$$

$$+\frac{1}{\Delta z}\frac{1}{\Delta x\,\Delta y}\Bigg(\underbrace{\sum_{m=0}^{1}\left(\Phi_{\mathrm{in},m,i,j,k}-\Phi_{\mathrm{out},m,i,j,k}\right)}_{\substack{\text{net diffuse radiative flux}\\ \text{in z direction}}}+\underbrace{\left(\Phi_{\mathrm{in},10,i,j,k}-\Phi_{\mathrm{out},10,i,j,k}\right)}_{\substack{\text{net direct radiative flux}\\ \text{in z direction}}}\Bigg)\Bigg]$$

$$=\frac{1}{\rho\,c_p}\frac{1}{\Delta x\,\Delta y\,\Delta z}\sum_{m=0}^{12}\left(\Phi_{\mathrm{in},m,i,j,k}-\Phi_{\mathrm{out},m,i,j,k}\right), \tag{3.4}$$

with Δx, Δy and Δz quantifying the size of the grid box. However, this formula raises some problems when used in combination with the incomplete solves introduced in Sect. 3.2.2. To explain this, we once more look at Fig. 3.3. While, for example, fluxes exiting the upper-left grid box are updated in the very first step, the diffuse flux entering that exact same grid box from the bottom is updated much later in the fourth step. Hence, when the whole Gauß–Seidel iteration is completed, the fluxes exiting a certain grid box do not necessarily match the ones entering it anymore; i.e., the fluxes are not consistent anymore. This can lead to heating rates that are unphysically large or negative in the solar spectral range. To avoid this problem, we have to rephrase the outgoing fluxes in Eq. (3.4) in terms of ingoing fluxes, as given by the equations in Eq. (3.1), resulting in

$$\left(\frac{\Delta T}{\Delta t}\right)_{i,j,k}=\frac{1}{\rho\,c_p}\frac{1}{\Delta x\,\Delta y\,\Delta z}\sum_{m=0}^{12}\left(\Phi_{\mathrm{in},m,i,j,k}-\sum_{n=0}^{12}t_{mn,i,j,k}\,\Phi_{\mathrm{in},n,i,j,k}-B_{m,i,j,k}\right). \tag{3.5}$$

Since this expression incorporates the radiative transfer throughout the corresponding grid cell, it ensures that all fluxes involved in the calculation of the heating rate are consistent with each other and thus provides physically correct 3D heating rates.

Chapter 4

Evaluation of the dynamic TenStream solver decoupled from model dynamics

After the previous chapter described the functionality of the dynamic TenStream solver, this chapter will present the first part of its evaluation. For this first part, the performance of the new solver is evaluated decoupled from model dynamics. In contrast to that, radiation and dynamics are normally tightly coupled to one another in numerical weather prediction models, as the radiative transfer model determines the sources and sinks of radiative energy in the atmosphere, which then drive the model dynamics, leading to a new atmospheric state, which in turn leads to updated sources and sinks of radiative energy. This coupling, however, results in different atmospheric states depending on the radiative transfer model used. Therefore, to evaluate just the performance of the different radiative transfer models, it is helpful to decouple radiation from dynamics by applying the different models onto an atmospheric time series that has been simulated in advance. By doing so, one can investigate the differences between the different radiative transfer solvers when applied to the very same atmospheric states and thus to the very same clouds in particular. This chapter will thus first introduce the precomputed shallow cumulus cloud time series used in this evaluation, before presenting the different radiative transfer solvers applied to it, which the new dynamic TenStream solver was consequently compared with. After outlining the different methods used for the evaluation, the second part of this chapter will then discuss the results of this evaluation.

The content of this chapter was published in Sect. 3 and 4 of Maier et al. (2024).

4.1 Methodology

The dynamic TenStream solver outlined in Chapter 3 was implemented in the libRadtran library for radiative transfer (Emde et al., 2016; Mayer and Kylling, 2005), allowing the performance of the new solver to be tested with respect to other solvers using an otherwise identical framework. Using this environment, our goal is to demonstrate that the new dynamic TenStream solver produces more accurate results than 1D independent-column solvers employed nowadays while still being noticeably faster than typical 3D solvers. Therefore, this section will first introduce our test setup as well as the solvers we compare dynamic TenStream with. Then, we will explain how we determine its performance in terms of both speed and accuracy. Since 3D solvers are computationally much more demanding than 1D solvers, in this analysis, special emphasis will also be placed on the calling frequency of the radiative-transfer calculations in order to elaborate whether the dynamic TenStream solver still performs better when operated with a similar computational demand to current 1D solvers.

4.1.1 Cloud and model setup

Our test setup is centered around a shallow cumulus data set with 3D cloud output data prepared by Jakub and Gregor (2022), which was computed using the University of California, Los Angeles (UCLA) large-eddy simulation (LES) model (Stevens et al., 2005). The dynamics in this LES simulation were driven not by radiation but by a constant net surface flux, as described in the corresponding namelist input files. Originally, the data set features both a high temporal resolution of 10 s and 256 × 256 grid boxes with a high spatial resolution of 25 m in the horizontal. It is 6 h long and characterized by a continuously increasing cloud fraction, starting with a clear-sky situation and ending up with a completely overcast sky. In addition, a southerly wind with a speed between 3 and 4.7 $\mathrm{m\,s^{-1}}$ transports the clouds through the domain (Gregor et al., 2023).

We have chosen this data set for two reasons. First, the high temporal resolution allows us to investigate the effect of incomplete solves in the dynamic TenStream solver with regard to the calling frequency of the solver. As we outlined in Sect. 3.2.2, we expect these incomplete solves to perform best if the cloud field that mainly determines the radiative field does not change much in between two radiation time steps. Due to the high temporal resolution, we can investigate how well the incomplete solves perform if we call the solver less often by comparing runs with low calling frequencies to runs with the highest possible calling frequency of 10 s. On the other hand, we need the high spatial resolution of the data, since the dynamic TenStream solver does not yet take sub-grid-scale cloud variability into account. However, we may not need a horizontal resolution of 25 m for that. Thus, to test the dynamic TenStream solver on a resolution that is closer to that of operational weather models without having to account for sub-grid-scale cloud variability, we decided to reduce the horizontal resolution of the cloud fields to 100 m. To avoid problems with an artificially low liquid-water content (LWC) at cloud edges when averaging the cloud field to that resolution, we constructed these less-resolved cloud fields by simply using the data for just every fourth grid box in both horizontal dimensions. The resulting time series still features a temporal resolution of 10 s, but the cloud data grid is reduced to 64 × 64 grid boxes with a resolution of 100 m in the horizontal. In the vertical, the modified cloud data set consists of 220 layers with a constant height of 25 m, thus reaching up to a height of 5.5 km. Using this modified grid, the shape of the grid boxes is also closer to the one in NWP models, with their horizontal extent being larger than their vertical extent.

For our test setup, we focus on the 100 time steps between 8000 and 9000 s into the simulation, where the shallow-cumulus-cloud field has already formed but has not yet reached a

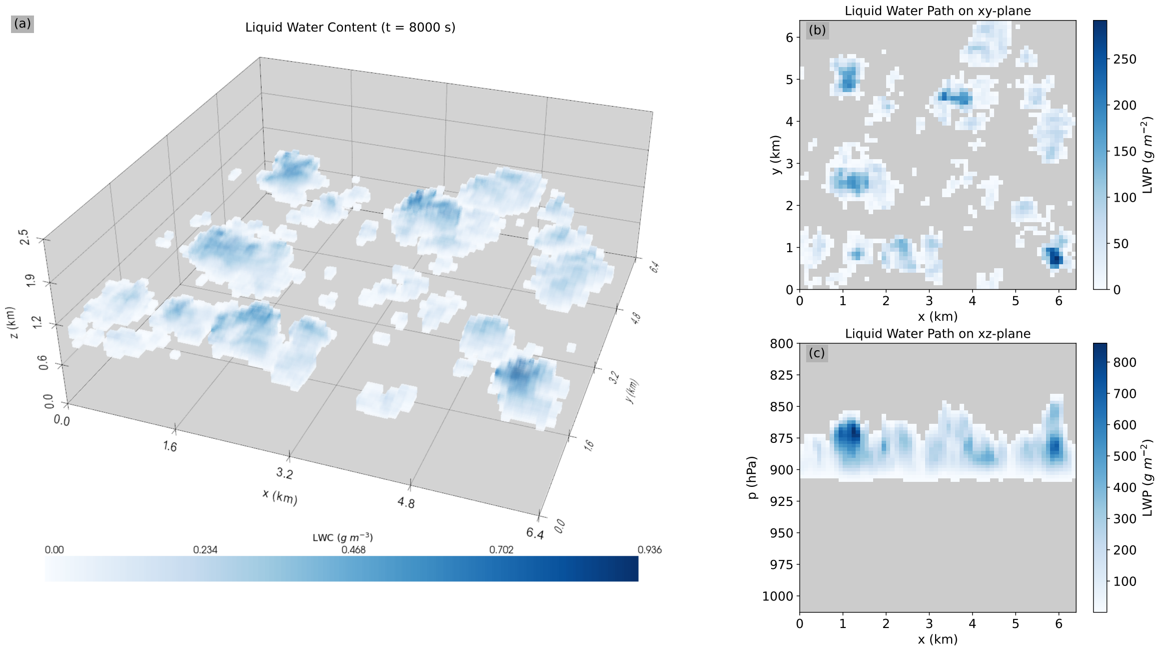

Figure 4.1: First time step of the shallow-cumulus-cloud field used in the evaluation. Panel **(a)** shows a 3D visualization of the liquid water content in the cloud field, whereas **(b)** and **(c)** display the vertically and horizontally integrated liquid-water content for the same cloud field, respectively.

very high cloud fraction, as neither a clear nor a completely overcast sky are beneficial for 3D cloud-radiative effects. Figure 4.1 shows the modified cloud field for the very first time step in this time frame. Looking at the vertically integrated liquid-water content in panel (b) in particular, one can see that our reduced horizontal resolution of 100 m allows us to still resolve the structure of the clouds.

Apart from the cloud field, the 1976 US standard atmosphere (Anderson et al., 1986) interpolated onto the vertical layers given by the cloud data grid serves as the background atmosphere. Above the cloud data grid, the native US standard atmosphere levels provided by libRadtran are used, so that the full grid features 264 layers in the vertical up to a height of 120 km. In both the solar and the thermal spectral ranges, the simulations are carried out using the molecular absorption parameterization from Fu and Liou (1992, 1993). In the solar spectral range, the Sun is placed at a constant zenith angle of 50° and in the east. The zenith angle was chosen to be quite low so that 3D effects such as cloud side illumination and shadow displacement are more pronounced, representing a typical morning scene. Furthermore, the surface albedo in the solar spectral range is set to 0.125, resembling the global mean value of Trenberth et al. (2009), whereas the ground emissivity is set to 0.95 in the thermal spectral range.

4.1.2 Overview of the radiative-transfer solvers

We apply four different radiative-transfer solvers to the aforementioned shallow-cumulus-cloud time series: the newly developed dynamic TenStream solver, the original TenStream solver, a classic 1D δ-Eddington approximation and a fully 3D Monte Carlo solver.

Let us discuss the setup of the dynamic TenStream solver first. As we outlined in Sect. 3.2.2, it has to be provided with a first guess the very first time it is called, due to the unavailability of a previously calculated result at this point in time. To evaluate the performance of the new solver, it is a good idea to use the best-possible solution for this first guess. This way, one can examine whether the results obtained from there on using dynamic TenStream featuring incomplete solves diverge from those retrieved by the original TenStream solver using full solves. Hence, we initially perform 2000 iterations for the clear-sky spin-up described in Sect. 3.2.2, followed by $N_0 = 500$ Gauß–Seidel iterations that also involve the cloud field to ensure that the radiative field is fully converged at the beginning of the time series. These two steps are visualized by the spin-up and the arrow with the first N_0 Gauß–Seidel iterations in Fig. 3.2. From there on, we just use a minimum of two Gauß–Seidel iterations every time the solver is called. Using two instead of just one iteration ensures that the iteration direction mentioned in Sect. 3.2.2 is altered at least once per call. In this way, we guarantee that information is not preferably transported in one specific direction. Furthermore, to investigate the effect of using more than just two Gauß–Seidel iterations per call, we also performed nine additional runs with integer multiples of two Gauß–Seidel iterations, i.e., with $4, 6, \ldots, 20$ Gauß–Seidel iterations per call.

Since the dynamic TenStream solver is based on the original TenStream solver, reproducing its results despite applying incomplete solves is the best outcome that we can expect. Thus, the original TenStream solver (Jakub and Mayer, 2015) serves as a best-case benchmark for our new solver. On the other hand, our goal is to significantly outperform currently employed 1D independent column approximations. Consequently, the δ-Eddington solver incorporated into the libRadtran radiative-transfer library serves as a worst-case benchmark for our new solver that we should definitely surpass.

Finally, we also apply the 3D Monte Carlo solver MYSTIC (Mayer, 2009) to the shallow-cumulus-cloud time series. When operated with a large-enough number of photons, it allows us to determine the most accurate 3D heating rates possible. Hence, these results can be used as a benchmark for all the other solvers. For our MYSTIC simulations, we used a total of 400 000 000 photons for every time step, which is about 100 000 per vertical column, resulting in domain-average mean absolute errors in both heating rates and irradiances that are not larger than 1 % of their respective domain averages. For the exact libRadtran input files used for each of these solvers, the reader is referred to Appendix A of this work.

4.1.3 Speed and accuracy evaluation

As we mentioned earlier, our goal is to evaluate the performance of the dynamic TenStream solver in terms of both speed and accuracy. However, in particular, determining the speed of a solver with respect to others is not a straightforward task, as it is highly dependent on the environment the code is executed in. Since the dynamic TenStream solver is still in an early stage of development and this work is primarily focused on demonstrating the feasibility of the main concepts of the solver, we wanted to keep the speed analysis as simple as possible. We decided to perform three radiative-transfer computations for each of the previously mentioned solvers on the same workstation for the first cloud field in our time series. The average of these three run times for every solver should at least provide a rough estimate of the speeds of the different solvers relative to each other. All calculations were performed on a single core. We compare the computational time for incomplete dynamic TenStream solves with two Gauß–Seidel iterations per call, which is the same setup as the one we will use for the investigation of the performance of the new solver later on, to run times for full solves by the 1D δ-Eddington approximation, the original TenStream solver and MYSTIC. This comparison is not entirely fair for the original TenStream solver, though, as this solver can be run in a time-stepping scheme as well, thus relying on previously calculated results, which noticeably increases its speed compared to the calculations from scratch that we are using. However, this time-stepping option for the original TenStream solver is not yet available within libRadtran.

To assess the accuracy of the dynamic TenStream solver, we study the entire time series. We focus our analysis on how well the solvers perform in determining 3D heating rates and net irradiances at the upper and lower domain boundaries. As mentioned in Sect. 4.1.2, values derived by MYSTIC serve as benchmark values. We evaluate the performance of the other three solvers compared to MYSTIC using two different error measures: a mean absolute error (MAE) and a mean bias error (MBE). The mean absolute error describes the amount by which the heating rate or net irradiance of an individual grid box deviates from the benchmark solution on average and is given by

$$\begin{aligned}
\mathrm{MAE} = \left\langle \left|\xi - \xi_{\mathrm{ref}}\right| \right\rangle, \quad \text{with} \quad & \xi \in \left\{ \left(\frac{\Delta T}{\Delta t}\right)_{i,j,k}; \Delta E_{\mathrm{net},i,j,\mathrm{sfc}}; \Delta E_{\mathrm{net},i,j,\mathrm{TOA}} \right\}, \\
& \Delta E_{\mathrm{net},i,j,\mathrm{sfc}} = \left(S_{0,i,j,0} + \Phi_{1,i,j,0} - \Phi_{0,i,j,0}\right) \Delta x \Delta y, \text{ and} \\
& \Delta E_{\mathrm{net},i,j,\mathrm{TOA}} = \left(S_{0,i,j,N_z+1} + \Phi_{1,i,j,N_z+1} - \Phi_{0,i,j,N_z+1}\right) \Delta x \Delta y.
\end{aligned} \tag{4.1}$$

Here, $\langle \ldots \rangle$ denotes a spatial average, whereas the subscript "ref" refers to a reference value, i.e., the MYSTIC values in our case. For the definition of the other quantities, you may refer back to Sec. 3.1 and 3.2.3. Since the mean absolute error is sensitive to how the values of an individual grid box deviate from the benchmark solution, it is a measure of whether a solver gets the overall heating rate or net irradiance pattern right. It is sensitive to double-penalty errors; i.e., it gets large when local minima and maxima in this pattern are displaced between the benchmark solution and the investigated solver. We have chosen an absolute error measure rather than a relative one here because individual heating rates or net irradiances can be close to zero and thus blow up a relative error measure. The mean bias error, on the other hand, is an error measure targeted toward the domain mean heating rate or net irradiance and is defined as

$$\mathrm{MBE} = \langle \xi \rangle - \langle \xi_{\mathrm{ref}} \rangle, \quad \text{with} \quad \xi \in \left\{ \left(\frac{\Delta T}{\Delta t}\right)_{i,j,k}; \Delta E_{\mathrm{net},i,j,\mathrm{sfc}}; \Delta E_{\mathrm{net},i,j,\mathrm{TOA}} \right\}. \tag{4.2}$$

In contrast to the MAE, the MBE compares domain-average values to each other and is thus a measure of whether we get the domain-average heating rate or net irradiance right. It is not sensitive to the spatial pattern of these quantities; rather, it tells us whether there is, on average, too much or too little absorption in the domain compared to the benchmark solution. Domain averages of heating rates and net irradiances are usually not close to zero, so we can also take a look at the relative mean bias error (RMBE) here, which is given by

$$\mathrm{RMBE} = \frac{\langle \xi \rangle - \langle \xi_{\mathrm{ref}} \rangle}{\langle \xi_{\mathrm{ref}} \rangle}, \quad \text{with} \quad \xi \in \left\{ \left(\frac{\Delta T}{\Delta t} \right)_{i,j,k}; \Delta E_{\mathrm{net},i,j,\mathrm{sfc}}; \Delta E_{\mathrm{net},i,j,\mathrm{TOA}} \right\}. \tag{4.3}$$

Applied to the shallow-cumulus-cloud time series at its full temporal resolution of 10 s, these two error measures allow us to determine the accuracy of the dynamic TenStream, the original TenStream and the δ-Eddington calculations compared to the MYSTIC benchmark run at any point in time. We can also ensure that the benchmark solution itself has a significantly smaller error than the other solvers when compared to this benchmark. Therefore, we use the standard deviation σ_{ref}, which can be determined for every single MYSTIC value. However, this standard deviation describes the mean squared deviation of a MYSTIC value from its mean, whereas the MAE that we are using looks at the mean absolute deviation. For normally distributed random variables, however, the mean absolute deviation is simply given by

$$\mathrm{MAD} = \sqrt{\frac{2}{\pi}}\,\sigma, \tag{4.4}$$

with σ being the standard deviation (Geary, 1935). We can assume that the benchmark run is of sufficient quality if this MAD of the MYSTIC values is much smaller than the corresponding mean absolute deviations between MYSTIC and the values of the other solvers. Hence, we can use the domain-average MAD of the benchmark solution to quantify the domain-average MAE of the benchmark solution at any point in time, following

$$\mathrm{MAE}_{\mathrm{ref}} = \left\langle \sqrt{\frac{2}{\pi}}\,\sigma_{\mathrm{ref}} \right\rangle. \tag{4.5}$$

We cannot provide a number for the MBE of the benchmark solution, though, as we only know how much the individual MYSTIC values are scattered around their mean, not whether this mean has an inherent bias. Hence, we simply have to assume that our benchmark simulation is unbiased.

So far, this evaluation would only tell us the accuracy of the different solvers compared to the benchmark run when operated at the same highest-possible calling frequency of 10 s. However, radiation is usually called far less often than the dynamical core of the model. Also, 3D radiative-transfer solvers are computationally much more demanding than 1D solvers, raising the question of how well dynamic TenStream performs when operated with a similar computational demand to the 1D δ-Eddington approximation. To address these questions, we also investigate the effect of the radiation calling frequency on the temporal evolution of the aforementioned error measures.

In order to explain our approach to this, we take a look at Fig. 4.2. The figure demonstrates how we determine the aforementioned error measures for a solver operated at a lower calling frequency of 30 s with respect to the MYSTIC benchmark run, which is computed at the highest-possible calling frequency of 10 s. At $t = 8020$ s (that is, 20 s into our time series), these error measures are given by comparing the not-yet-updated solution that was originally calculated at $t = 8000$ s to the values of the benchmark solution obtained at exactly $t = 8020$ s. In this

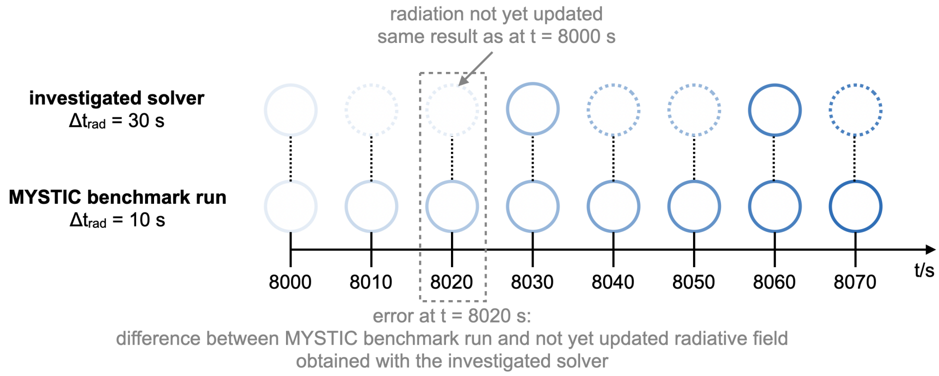

Figure 4.2: Schematic illustration of how we determine the error of a solver operated at a lower calling frequency of $\Delta t_{rad} = 30$ s compared to the benchmark solution computed at the highest-possible calling frequency of $\Delta t_{rad} = 10$ s at any point in time. To this end, the circles in the figure indicate the results of the corresponding solvers at any given time, with the colors symbolizing the time step at which these results have been calculated and the dotted circles, in contrast to the full circles, indicating results that have not been updated at that point in time. The color of a dotted circle is thus equal to that of the corresponding last full circle.

way, we can investigate how the error metrics of a not-updated radiative field grow until it is eventually updated again. This investigation is particularly important for the dynamic TenStream solver, as it just performs incomplete solves every time it is called. As we expect these to work best if the overall properties determining the radiative field have not changed much in between two calls of the solver, this method allows us to investigate whether our new solver still converges toward the results of the original TenStream solver when operated at lower calling frequencies. For this investigation, we decided to take calling frequencies of 10, 30 and 60 s into account. These calling frequencies are still very high for operational weather forecasts, where the radiation time step is typically around 1 h (Hogan and Bozzo, 2018), but we have to consider that our cloud field also features a significantly higher spatial resolution of 100 m in the horizontal compared to 2.1 km in the DWD ICON-D2 model (DWD, 2025) and 9 km in the ECMWF high-resolution deterministic forecasts (Hogan and Bozzo, 2018). At the LES resolution of 100 m that we use for our evaluation, the Weather Research and Forecasting (WRF) model, for example, recommends using a radiation time step as high as 1 min km^{-1} of horizontal resolution (UCAR, 2025), resulting in a suggested radiation time step of 6 s for our test case. This ensures that air moves less than a grid box distance between two calls of the radiation scheme for typical wind speeds of $|\vec{v}| \approx 10$ m s^{-1}. Our highest calling frequency of 10 s is at least close to that number, with the other two calling frequencies of 30 and 60 s definitely representing scenarios where radiation is called less often than recommended.

4.2 Discussion of the results

4.2.1 Solver speed

The relative speeds of the different radiative-transfer solvers introduced in Sect. 4.1.2 compared to the run time of the 1D δ-Eddington approximation are shown in Table 4.1. As we described

Table 4.1: Computing times of the different solvers (names in bold font) relative to those of the 1D δ-Eddington approximation, taken as an average over three runs performed on the same workstation for the very first time step of the LES cloud time series.

	Solar spectral range	Thermal spectral range
δ-Eddington *1D two-stream solver*	1.0	1.0
dynamic TenStream *incomplete 3D solver with two Gauß–Seidel iterations*	3.6	2.6
original TenStream *full 3D solver*	50.8	24.1
MYSTIC *full 3D benchmark solver using 400 000 000 photons*	1068.9	1611.3

in Sect. 4.1.3, all solvers used in this test were executed on a single core of the same workstation and were therefore in a very similar environment. This workstation featured an Intel Xeon W-2245 CPU and 64 GB RAM, with performance primarily limited by the network storage, where all the data were placed. We can see that in this experiment, the newly developed dynamic TenStream solver with two Gauß–Seidel iterations is 3.6 times slower than the 1D δ-Eddington approximation in the solar spectral range and just 2.6 times slower in the thermal spectral range. Comparing these numbers to the findings in Jakub and Mayer (2016), they are in line with what we could have expected in terms of the speed of the new solver. According to this paper, retrieving the coefficients of the TenStream linear-equation system from the lookup tables in both the solar and thermal spectral ranges takes about as long as performing one δ-Eddington calculation. On top of that, we have to calculate the fluxes for every grid box of the domain, just as in a δ-Eddington calculation. However, for the dynamic TenStream calculation, we have to determine fluxes for 10 instead of 2 streams per grid box and calculate all these fluxes twice, as we perform two Gauß–Seidel iterations whenever the solver is called. And, even though the number of streams (in particular) will most likely not scale linearly with run time, we can thus certainly expect that the new solver will be at least twice as slow as a δ-Eddington approximation. The factors of 3.6 and 2.6 are in line with these expectations.

Although the dynamic TenStream solver is thus noticeably slower than a 1D solver, it is still significantly faster when compared to other 3D solvers executed under similar circumstances — namely the original TenStream and MYSTIC solvers in Table 4.1. The original TenStream solver, for example, is at least 24 times slower in this experiment than the 1D δ-Eddington approximation, with MYSTIC being even slower. As we pointed out earlier, this comparison is not entirely fair for the original TenStream solver, though, as it can also be run in a time-stepping scheme. Jakub and Mayer (2016) showed that, in this case, the original TenStream solver can be up to only a factor of 5 slower than 1D δ-Eddington solves, which, however, is still noticeably slower than the dynamic TenStream solver presented here.

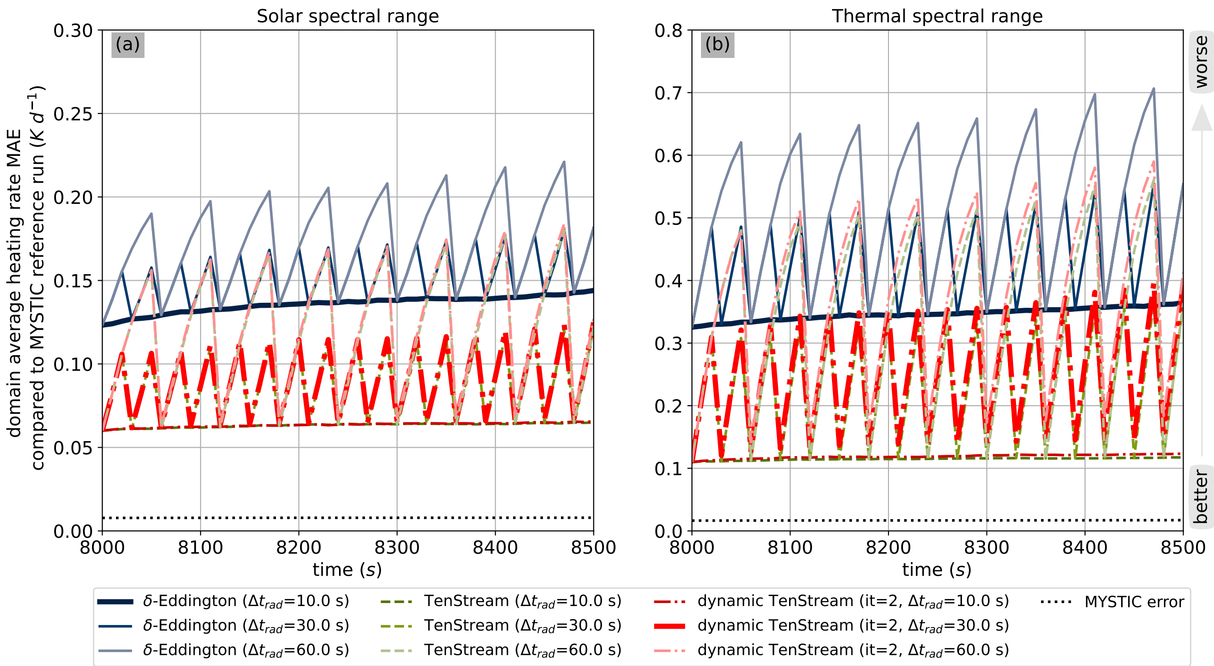

Figure 4.3: Temporal evolution of the mean absolute error in heating rates for the 1D δ-Eddington approximation (blue lines), the original TenStream solver (dashed green lines) and the newly developed dynamic TenStream solver (dash–dotted red lines) with respect to the MYSTIC benchmark run at calling frequencies of 10, 30 and 60 s (different shades of the corresponding color) for both the solar (**a**) and thermal (**b**) spectral ranges. Due to the statistical nature of Monte Carlo simulations, the MYSTIC benchmark run itself is subject to some uncertainty. The corresponding MAE calculated using Eq. (4.5) is visualized by the dotted black line. For reasons of visual clarity, we show only the first half of the time series here.

4.2.2 Performance in determining heating rates

Next, let us have a look at how the dynamic TenStream solver performs in calculating heating rates. Since we are primarily interested in its performance in the surroundings of the continuously evolving clouds, we only use the LES part of the domain for this evaluation, i.e., the part between the surface and 5.5 km height. As mentioned in Sect. 4.1.3, the analysis will be centered around two different error measures: a mean absolute error and a mean bias error. Figure 4.3 shows the temporal evolution of the MAE for the different solvers at calling frequencies of 10, 30 and 60 s. At this point, we should recall that the mean absolute error is a measure of how well a solver performs on average in determining the heating rate for a certain grid box.

When operated at the highest-possible calling frequency of 10 s, we can see that the MAE is relatively constant over time for all the solvers, as we compare radiative-transfer calculations carried out at a certain point in time to benchmark calculations obtained at the exact same time step. The MAE in this case is solely determined by the error generated by the solvers themselves when applied to relatively similarly structured shallow cumulus clouds, so this behavior is expected. Looking at the magnitude of the MAE for the different solvers, we can see that for both spectral regions, the δ-Eddington approximation (dark-blue line) performs worst, whereas the 3D TenStream solver is a noticeable improvement. Pleasingly, the MAE of our dynamic TenStream solver at a calling frequency of 10 s (dark-red dash–dotted line) is almost on par with the error obtained with the original TenStream solver. It is only in the thermal spectral range that the error gets slightly larger with time. This shows that, in this example, at a calling frequency of 10 s, just two Gauß–Seidel iterations per call are already sufficient to almost reproduce the results of the original TenStream solver.

At lower calling frequencies, the radiative field is not updated at every time step of the cloud time series anymore. Consequently, the MAE of each solver rises until the solver is called again. The resulting sawtooth structure can be observed in the MAE time series of all the solvers at calling frequencies of 30 and 60 s. In case of the traditional solvers, a full solve is performed every time they are called. Thus, the MAE at lower calling frequencies always reduces to the value obtained at a calling frequency of 10 s when the corresponding solver is called. This is not necessarily true for the dynamic TenStream solver, however, as it only performs an incomplete solve involving two Gauß–Seidel iterations every time it is called. If this incomplete solve is not sufficient, it could lead to a divergent behavior of the MAE time series for this solver. Looking closely, we can also see that for both of the lower calling frequencies, the MAE of the dynamic TenStream solver does not always match the errors obtained at a calling frequency of 10 s when updated. However, even at a calling frequency of 60 s, we cannot observe a divergent behavior, and the newly developed solver is almost able to reproduce the results of the original TenStream solver whenever called.

Moreover, we have seen that our new solver is about 3 times slower than a traditional 1D δ-Eddington approximation. Looking at Fig. 4.3, we can now see that, on average over time, dynamic TenStream even performs better than the δ-Eddington approximation at a calling frequency of 10 s (bold blue line) when it is operated at a calling frequency of 30 s (dash–dotted bold red line), and thus with a similar computational demand to the 1D solver, in both the solar and thermal spectral ranges.

Switching to the other error measure, Fig. 4.4 visualizes the temporal evolution of the mean bias error for the different solvers. In contrast to the MAE discussed before, this error metric describes whether we get the domain-average heating rate right. As we can clearly see, the MBE is, again, largest for the 1D δ-Eddington approximation and, once more, significantly smaller for the original TenStream model. When operated at the highest-possible calling frequency of 10 s, the mean bias error of the dynamic TenStream solver is also very similar to that of the original TenStream model. However, at lower calling frequencies, we can clearly see that the bias increases with time, although it never gets larger than the bias of the 1D results. It is also clearly visible that the bias is more negative than the original TenStream bias (dashed green lines) in the solar spectral range, whereas it is less positive in the thermal spectral range. Since the domain-average heating rate in the solar spectral range is positive, this implies that our new solver underestimates absorption in this spectral range, especially compared to the original TenStream solver it is based on. This underestimation gets larger the less the dynamic TenStream solver is called. As the liquid-water content in the domain gradually increases with time and more liquid water in the clouds leads to more absorption, this could imply that the dynamic TenStream solver does not fully take this increase into account. This does not explain the behavior of the new solver in the thermal spectral range, though, where domain-average

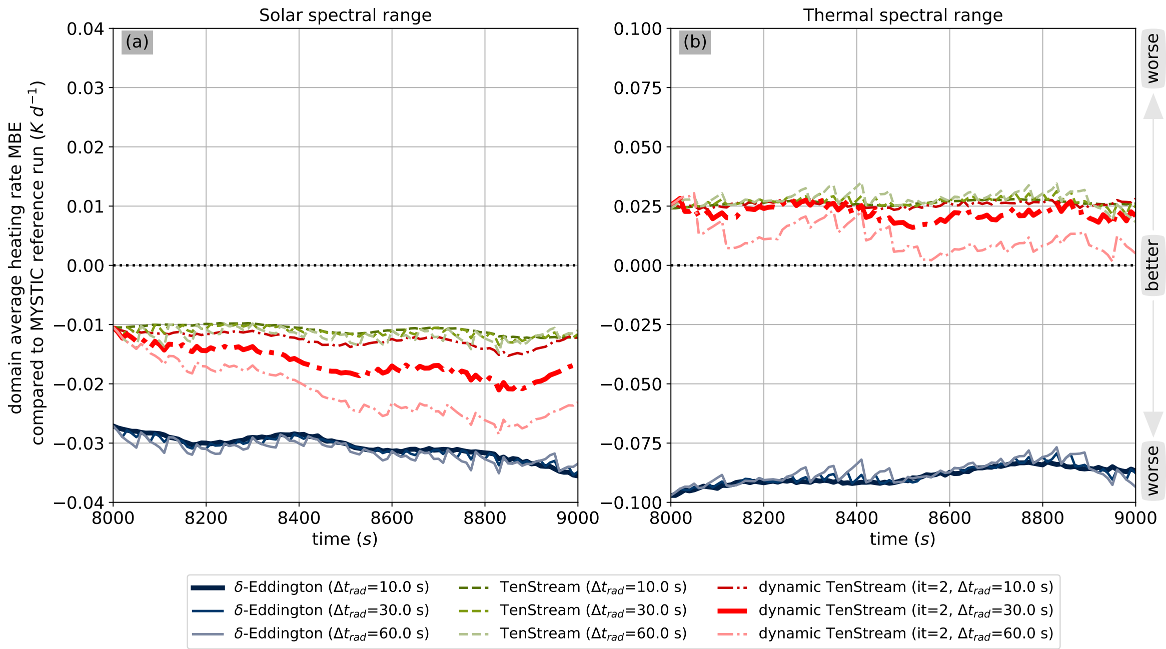

Figure 4.4: Temporal evolution of the mean bias error in heating rates for the different solvers with respect to the MYSTIC benchmark run at calling frequencies of 10, 30 and 60 s for both the solar (**a**) and thermal (**b**) spectral ranges. A run with no bias is visualized by the dotted black line.

heating rates are negative. Hence, the positive MBE values observed for both the original as well as the dynamic TenStream solver imply that the heating rates are not as negative there as they should be. But, in contrast to the solar spectral range, these heating rates get more negative the less the dynamic TenStream solver is called, so the dynamic TenStream solver overestimates the magnitude of these thermal heating rates when compared to the original TenStream solver it is based on. Using this and the results obtained from the MAE time series, we can draw our first few conclusions:

1. For an individual grid box, our new solver is able to determine heating rates much more accurately than current 1D solvers, even when operated with a similar computational demand.

2. When looking at domain averages, the dynamic TenStream solver begins to develop a bias compared to the original TenStream solver it is based on. This bias becomes larger the lower the calling frequency is, but it remains smaller than the bias of the 1D δ-Eddington calculations at any point in time.

4.2.3 Performance in determining net irradiances at the upper and lower domain boundaries

Besides heating rates, we are also interested in how well dynamic TenStream performs in determining net irradiances at the top and bottom of its domain. We will start by looking

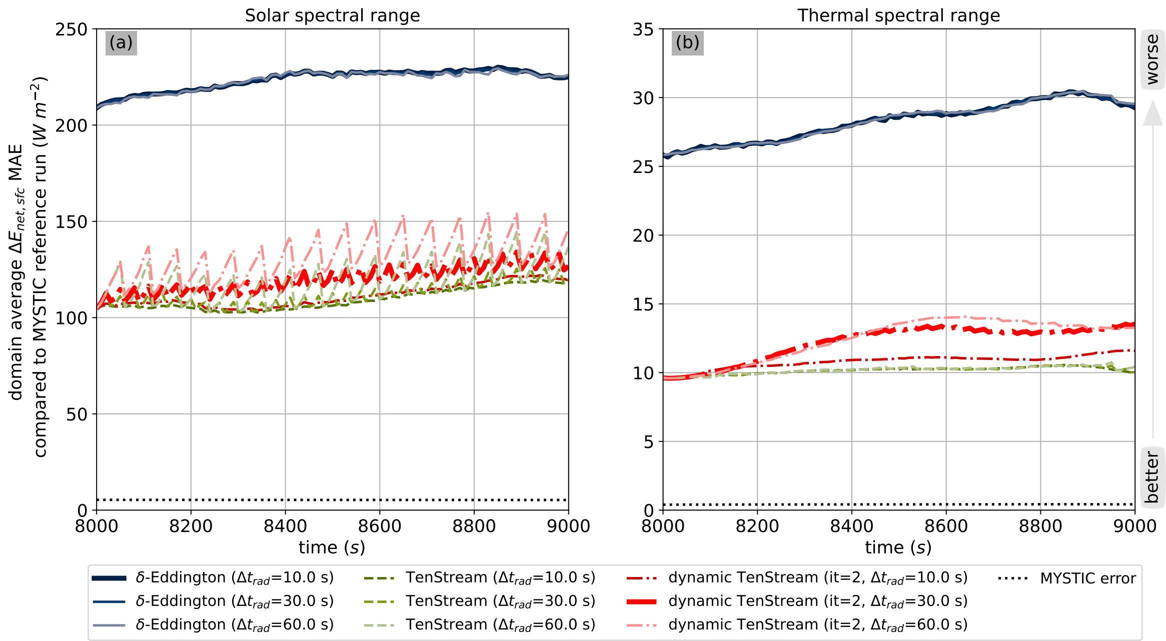

Figure 4.5: Temporal evolution of the mean absolute error in the net surface irradiance for the different solvers with respect to the MYSTIC benchmark run at calling frequencies of 10, 30 and 60 s for both the solar (**a**) and thermal (**b**) spectral ranges. The MAE of the MYSTIC benchmark run itself is visualized by the dotted black line.

at the results for the net surface irradiances and thus absorption at the ground. Figure 4.5 shows the temporal evolution of the MAE for this quantity in an otherwise similar fashion to Figs. 4.3 and 4.4. As for the heating rates, we can see that the 1D δ-Eddington approximation (blue lines) performs worst, with the original TenStream solver (dashed green lines) being a noticeable improvement once more, remaining significantly below the errors of all 1D runs throughout the entire time series, even at lower calling frequencies. Again, our newly developed dynamic TenStream solver (dash–dotted red lines) is able to almost maintain the MAE of the full TenStream calculations at the highest-possible calling frequency of 10 s, whereas its error slightly increases with time for the two lower calling frequencies. However, this slight divergence from the original TenStream MAE quickly stabilizes and also remains significantly below every single δ-Eddington run, even when the solver is only called every 60 s. What is interesting, though, is that the temporal evolution of the MAE in the thermal spectral range does not show a sawtooth structure at lower calling frequencies, in contrast to all other plots involving the MAE so far. As we discussed earlier, this sawtooth structure is mainly caused by the fact that at lower calling frequencies, we do not update the radiative field for some time steps while the clouds are still moving through the domain, resulting in gradually increasing double-penalty errors. The fact that this behavior is not observed in the thermal spectral range indicates that the net surface irradiance field does not feature such small-scale structures in the thermal.

To conclude the analysis for the surface, let us once more also have a look at the MBE. Again, in contrast to the MAE, this error measure does not tell us how well dynamic TenStream

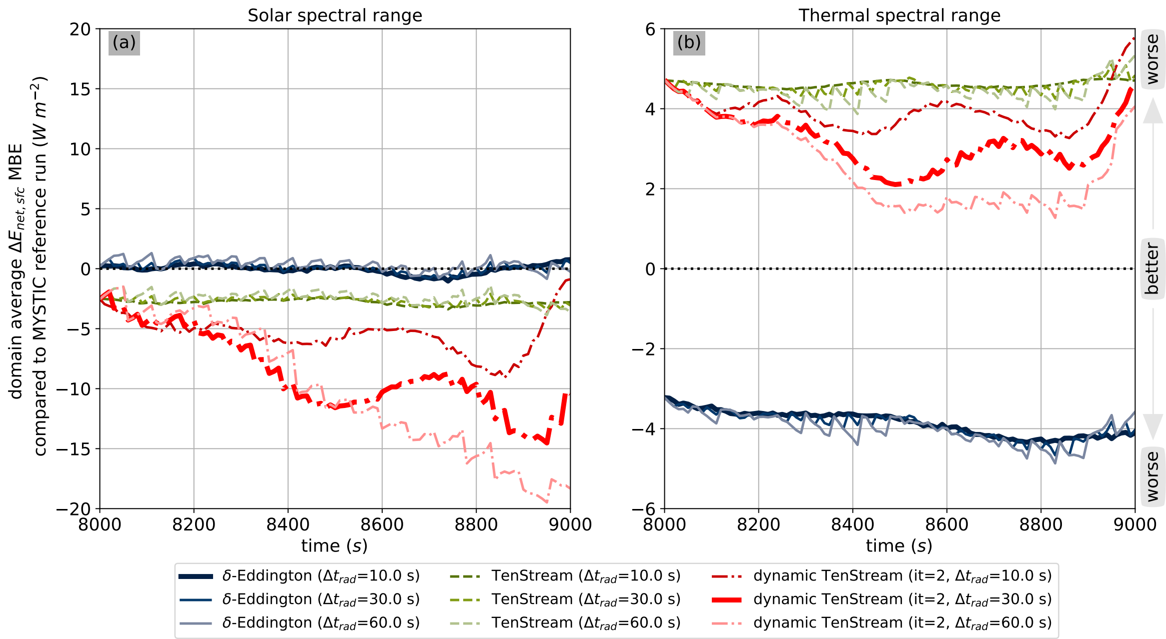

Figure 4.6: Temporal evolution of the mean bias error in the net surface irradiance for the different solvers with respect to the MYSTIC benchmark run at calling frequencies of 10, 30 and 60 s for both the solar (**a**) and thermal (**b**) spectral ranges. A run with no bias is indicated by the dotted black line.

performs in determining the net surface irradiance for a single grid box, but rather whether we get the domain-average surface absorption right. The corresponding plot is shown in Fig. 4.6 and reveals a current weakness of both the original TenStream solver as well as our new solver, as we can clearly see that the MBE is almost always larger for these two solvers than it is for the 1D δ-Eddington approximation. And, as we have already seen in the results for the heating rates, the lower the calling frequency, the more the MBE of the dynamic TenStream solver diverges from the MBE of the original TenStream solver. Here, however, this behavior is more severe than it was for the heating rates, since the benchmark for our new solver — the original TenStream solver — already performs a bit worse than the 1D solver. Its MBE of about $-2.5\,\mathrm{W\,m^{-2}}$ in the solar and $5\,\mathrm{W\,m^{-2}}$ in the thermal spectral range translates to an RMBE of about $-0.5\,\%$ and $-6\,\%$, respectively (not shown here), compared to numbers of around $0\,\mathrm{W\,m^{-2}}$ (0 %) in the solar and $-4\,\mathrm{W\,m^{-2}}$ (5 %) in the thermal spectral range for the δ-Eddington approximation. However, it should be noted that the almost non-existent MBE of the δ-Eddington approximation in the solar spectral range is primarily caused by two counteracting 3D radiative effects that happen to cancel each other out at almost exactly the solar zenith angle of 50° used here, whose choice was motivated in Sect. 4.1.1.

To show that, Fig. 4.7 visualizes the MBE for both the δ-Eddington approximation and the original TenStream solver as a function of the solar zenith angle for the first time step of our time series. By looking at the blue line, we can see that the δ-Eddington approximation underestimates the mean net surface irradiance for solar zenith angles below 50°, while it overestimates it for angles above 50°. Following the reasoning of Gristey et al. (2020), this is

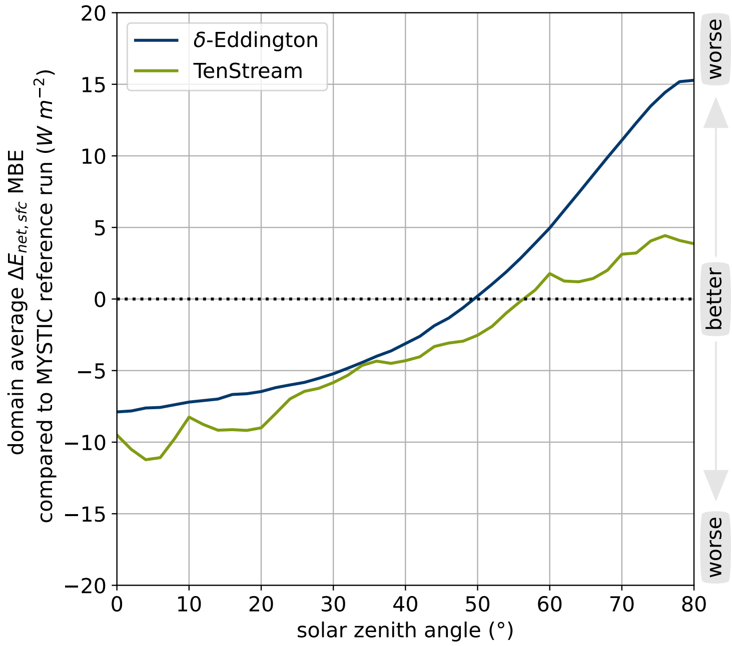

Figure 4.7: Mean bias error in the net surface irradiance as a function of the solar zenith angle for both the 1D δ-Eddington approximation (blue line) and the original TenStream solver (green line), evaluated at the first time step of the shallow-cumulus-cloud time series.

most likely because at low solar zenith angles, 1D solvers cannot account for the scattering of light at clouds into surrounding clear-sky columns, leading to an underestimation of the mean net surface irradiance. At high solar zenith angles, on the other hand — that is, when the Sun is close to the horizon — 1D solvers severely underestimate the size of cloud shadows, as they cast them directly underneath the clouds instead of at a slant angle, leading to an overestimation of the mean net surface irradiance. As we can see in Fig. 4.7, both of these effects cancel out at an angle of about 50°, which is the one used here, resulting in the near-zero MBE of the δ-Eddington approximation in the solar spectral range in Fig. 4.6. Despite this coincidence, however, Fig. 4.7 also shows that the original TenStream solver performs slightly worse than the δ-Eddington approximation for any zenith angle below about 50°. However, the difference in MBE between the two solvers is quite small, and the magnitudes of their respective RMBEs do not get much larger than −1 % for any angle below 50° (not shown here).

The dynamic TenStream solver, however, underestimates surface absorption in the solar spectral range even more than the original TenStream solver does, with the effect increasing the less frequently the new solver is called. Looking at the runs with calling frequencies of 10 and 30 s, however, one can clearly see that this divergence from the original TenStream MBE quickly stabilizes itself at values around $-5\,\mathrm{W\,m^{-2}}$ (−1 %) and $-12\,\mathrm{W\,m^{-2}}$ (−2 %), indicating that the bias will not grow continuously. The same behavior can be observed in the thermal spectral range, except that, similar to the behavior of the heating rates, the buildup of the bias compared to the original TenStream solver actually improves the MBE of the new solver at lower calling frequencies there. Since net surface irradiances in the thermal spectral range

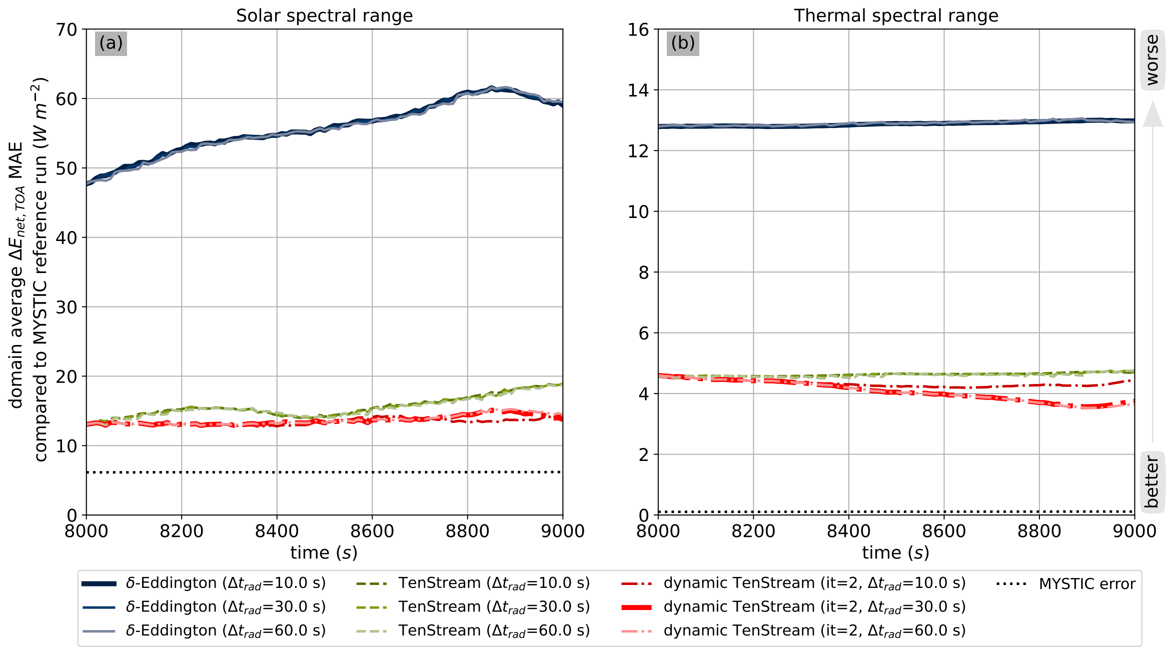

Figure 4.8: Temporal evolution of the mean absolute error in the net irradiance at TOA for the different solvers with respect to the MYSTIC benchmark run at calling frequencies of 10, 30 and 60 s for both the solar (**a**) and thermal (**b**) spectral ranges. The MAE of the MYSTIC benchmark run itself is visualized by the dotted black line.

are negative, the positive MBE values for the original TenStream solver in Fig. 4.6 indicate an underestimation in the net surface irradiance, i.e., values that are not negative enough, with the dynamic TenStream solver counteracting this bias the less often it is called — although this is, of course, more of a coincidence.

Finally, coming to the upper boundary of our domain, Fig. 4.8 shows the temporal evolution of the MAE in the net irradiance at the top of the atmosphere (TOA) in an otherwise similar fashion to Fig. 4.5. Again, the incomplete solves in the dynamic TenStream solver lead to a slight divergence of the MAE of this solver (red lines) compared to the original TenStream solver (green lines) in both spectral ranges. However, this divergence remains small compared to the difference between the 3D TenStream solver and the 1D δ-Eddington approximation, even at the lowest investigated calling frequency of 60 s. This indicates that the dynamic TenStream solver is also much better in capturing the spatial structure of the net irradiances at TOA than the traditional δ-Eddington approximation is.

Similar to the surface, however, this does not fully apply in terms of domain averages. The corresponding temporal evolution of the MBE is shown in Fig. 4.9. Starting with the thermal spectral range displayed in panel (b), our new solver again shows only a comparatively small divergence from the original TenStream solver with time and performs significantly better than the δ-Eddington approximation throughout the entire time series, regardless of the calling frequency used. In the solar spectral range, however, the original TenStream solver already performs a bit worse than the δ-Eddington approximation does, with time-average MBEs of about $-4\,\mathrm{W\,m^{-2}}$ for the TenStream solver compared to $-3\,\mathrm{W\,m^{-2}}$ for the 1D solver.

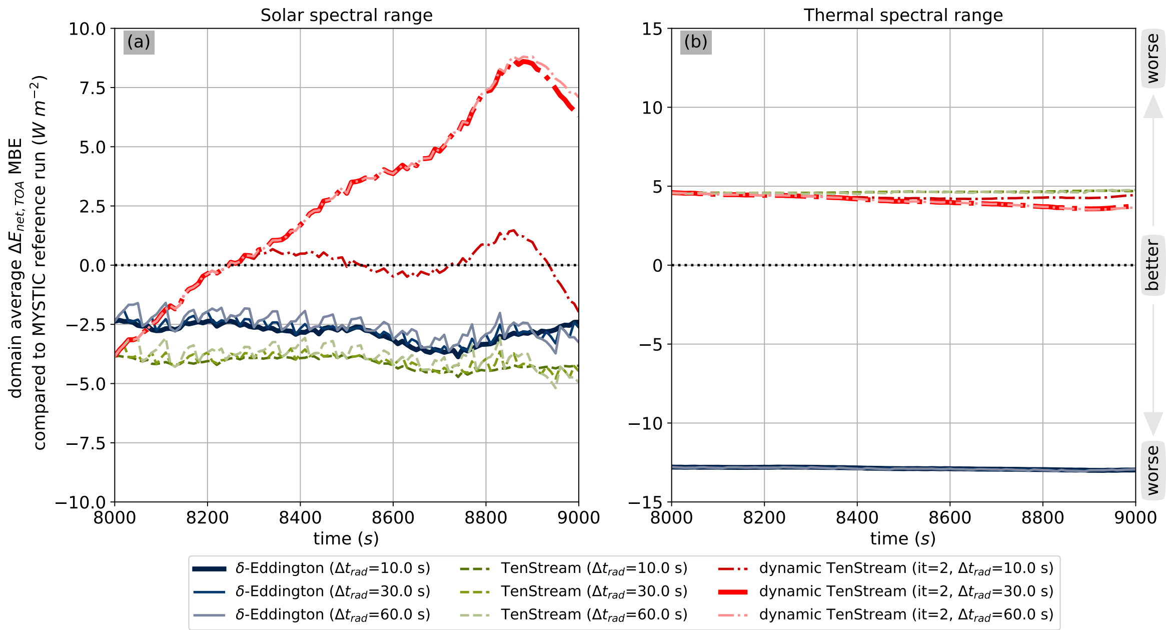

Figure 4.9: Temporal evolution of the mean bias error in the net irradiance at TOA for the different solvers with respect to the MYSTIC benchmark run at calling frequencies of 10, 30 and 60 s for both the solar (**a**) and thermal (**b**) spectral ranges. A run with no bias is indicated by the dotted black line.

More noticeably though, the incomplete solves in the dynamic TenStream solver lead to a fairly pronounced divergence in terms of the MBE from the original TenStream solver when compared to the difference between the 1D and original TenStream solvers. However, for every calling frequency investigated, this divergent behavior peaks at values that translate to RMBEs of no larger than 1.25 % (not shown here). Taking both domain boundaries into account, we can thus draw similar conclusions to those for the heating rates:

1. At the grid box level, our new solver determines far better net irradiances at both the surface and TOA than current 1D solvers do, even when operated at much lower calling frequencies.

2. Looking at domain averages, however, the incomplete solves within the dynamic TenStream solver lead to a buildup of bias with time. In terms of magnitude relative to the original TenStream solver, this bias becomes larger the lower the calling frequency and exceeds the bias of current 1D solvers, especially in the solar spectral range.

4.2.4 Dependence on the number of Gauß–Seidel iterations

So far, we have just looked into dynamic TenStream runs performed with only two Gauß–Seidel iterations whenever the solver is called. We focused on this computationally affordable setup as it already led to promising results. To investigate how the results presented so far change when applying more than two Gauß–Seidel iterations, we have performed nine additional

Table 4.2: Computing times of dynamic TenStream runs with N Gauß–Seidel iterations per call relative to those with two Gauß–Seidel iterations, taken as an average over three runs performed on the same workstation for the very first time step of the LES cloud time series.

Number N of Gauß–Seidel iterations	2	4	6	8	10	12	14	16	18	20
Solar spectral range	1.0	1.2	1.4	1.6	1.7	1.9	2.1	2.3	2.5	2.6
Thermal spectral range	1.0	1.1	1.3	1.4	1.5	1.6	1.8	1.9	2.0	2.2

runs using integer multiples of two Gauß–Seidel iterations, i.e., up to 20 iterations per call. Following the explanation given in Sect. 4.1.2, we use integer multiples of 2 instead of 1 in order to ensure that information is not preferably transported into one specific direction of the domain.

In order to evaluate the improved performance of these additional runs, it is important to have a rough estimate of their additional computational cost. Therefore, we have measured the computing times of these runs exactly as we did it in Sect. 4.2.1 for all the other solvers. Table 4.2 shows these computing times relative to a calculation with two Gauß–Seidel iterations per call. As we can see, using four instead of two iterations does not double the computational cost, as there is a considerable amount of overhead that always takes the same amount of time before even starting with the Gauß–Seidel iterations, such as retrieving the TenStream coefficients from the corresponding lookup tables. However, apart from this offset, computing time scales roughly linearly with the number of Gauß–Seidel iterations, as two more iterations always add about 10 % to 20 % of the baseline cost of a calculation with two Gauß–Seidel iterations to the computing time. This fraction is smaller for the thermal spectral range because of a larger overhead due to the additional calculation of thermal emission.

Having this additional computational burden in mind, we can now have a look at Fig. 4.10. Panels (a) and (b) in this figure show the time- and domain-average MAE in heating rates from the dynamic TenStream solver for the shallow-cumulus-cloud time series as a function of the number N of Gauß–Seidel iterations. Correspondingly, the values at $N = 2$ on the far left are the time averages of the sawtooth curves in Fig. 4.3 for the corresponding calling frequencies. The dashed lines represent the temporal mean MAEs for the original TenStream solver. The MAEs of the dynamic TenStream solver converge toward these dashed lines in the limit of a large number of iterations. For lower calling frequencies, this limit that the dynamic TenStream solver is converging to is larger than it is for higher calling frequencies because the solver is called less often, leading to the buildup of a large MAE with time until the solver is eventually called again, as we have seen in Fig. 4.3. Since the MAEs of the dynamic TenStream solver were already almost on a par with the original TenStream solver when just two Gauß–Seidel iterations were used, the MAE is already nearly converged at $N = 2$ and does not greatly improve when using more iterations. It is only in the thermal spectral range and at lower calling frequencies that we see a slight improvement in the mean MAE when applying more iterations, especially when doubling the number of iterations from two to four.

In contrast to the MAE, however, we observed a noticeable buildup of bias with time for the dynamic TenStream solver that increases the less the solver is called. Consequently, the MBE

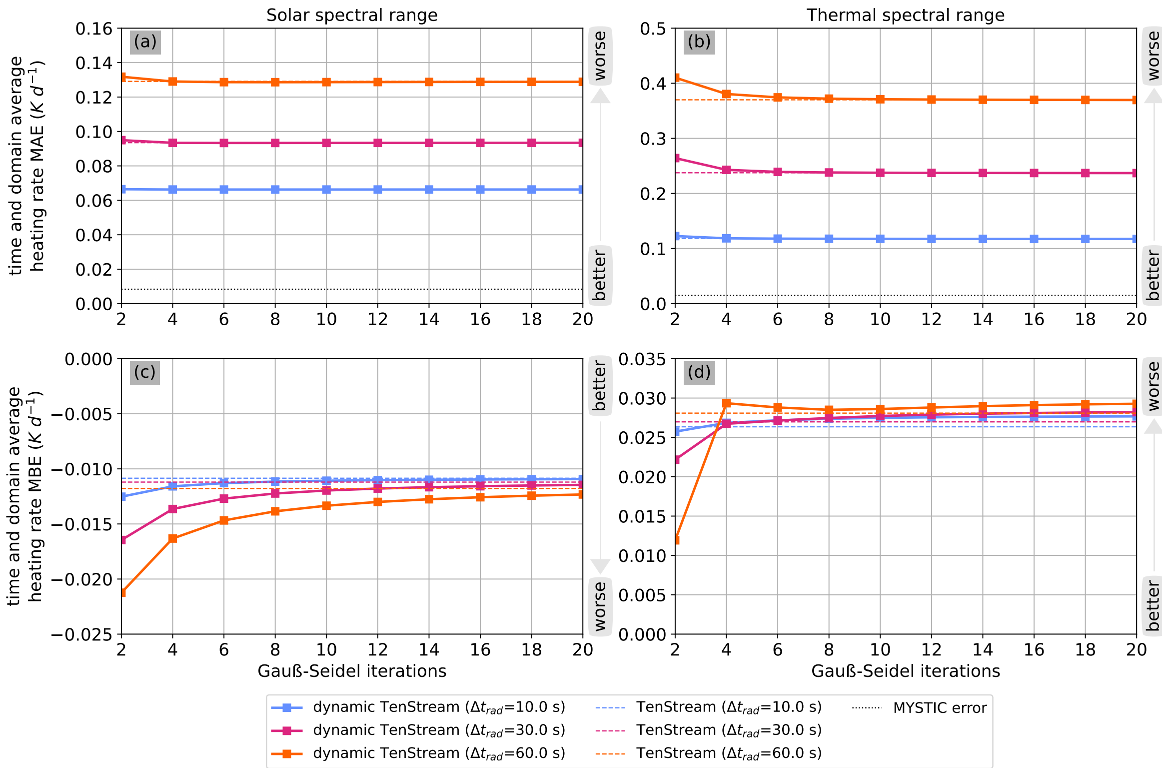

Figure 4.10: Time- and domain-average mean absolute error (**a**, **b**) and mean bias error (**c**, **d**) in heating rates with respect to the MYSTIC benchmark run as a function of the number of Gauß–Seidel iterations used in the dynamic TenStream solver for both the solar (left panels) and thermal (right panels) spectral ranges. The three different colors show the errors for calling frequencies of 10 s (blue), 30 s (purple) and 60 s (orange). Solid lines connect the values for the dynamic TenStream solver, while the dashed lines with constant MAE or MBE represent the errors of a full TenStream solve at the corresponding calling frequency, toward which the dynamic TenStream values are converging. In panels **(a)** and **(b)**, the MAE of the MYSTIC benchmark run itself is visualized by the dotted black line.

in panels (c) and (d) of Fig. 4.10 starts at values that are significantly far from convergence at $N = 2$, especially for the lowest two calling frequencies. The more Gauß–Seidel iterations we apply, the more this difference in bias compared to the original TenStream solver disappears. We can also see that the initially better bias of our new solver in the thermal spectral range at a calling frequency of 60 s quickly converges toward the bias of the original TenStream solver, as dynamic TenStream is based on this solver. To evaluate whether it is worth decreasing the magnitude of the MBE compared to the original TenStream solver by applying more iterations, let us have a look at the additional computational cost of these iterations in Table 4.2. Using four instead of two Gauß–Seidel iterations adds only 10 % to 20 % to the total computational time while leading to a noticeable decrease in both the MAE and (especially) the MBE. In this regard, one could even think about calling dynamic TenStream less frequently but with more Gauß–Seidel iterations. As we have seen in Sect. 4.2.1, using our new solver at a calling frequency of 30 s is about as expensive as calling a δ-Eddington approximation every 10 s. Taking Table 4.2 into account, we can see that using $N = 20$ instead of $N = 2$ iterations is a bit

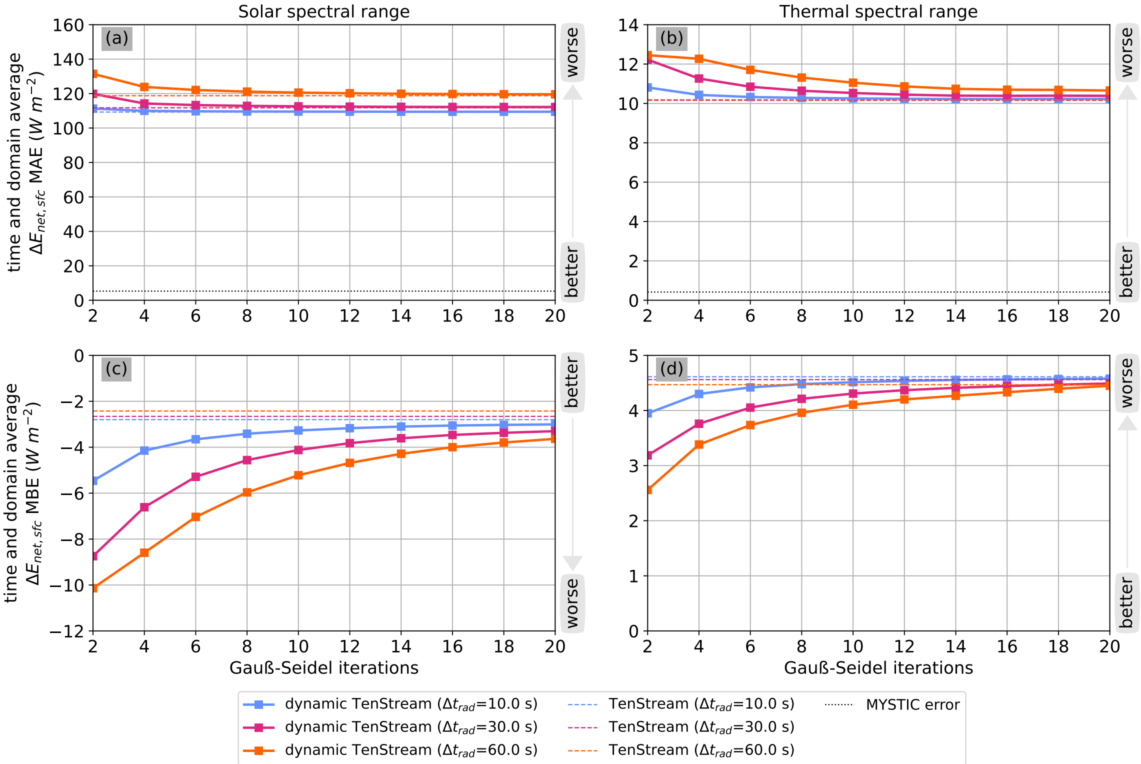

Figure 4.11: Time- and domain-average mean absolute error (**a**, **b**) and mean bias error (**c**, **d**) in the net surface irradiance with respect to the MYSTIC benchmark run as a function of the number of Gauß–Seidel iterations at calling frequencies of 10, 30 and 60 s. Solid lines connect the values for the dynamic TenStream solver, while the constant dashed lines represent the errors of a full TenStream solve at the corresponding calling frequency. In panels (**a**) and (**b**), the MAE of the MYSTIC benchmark run itself is visualized by the dotted black line.

more than twice as expensive. Hence, we could argue that a dynamic TenStream configuration with $N = 20$ at a lower calling frequency of 60 s also imposes about the same computational cost as a δ-Eddington approximation at a calling frequency of 10 s. However, while such a setup would lead to a better time-average MBE than our configuration with $N = 2$ and a calling frequency of 30 s, it would also lead to a very noticeable increase in the mean MAE. To put it figuratively, using more iterations at a lower calling frequency reduces the bias at the expense of the spatially correct representation of the heating rates. In terms of these heating rates, we can thus draw two main conclusions:

1. Using more Gauß–Seidel iterations per call primarily counteracts the buildup of a bias with time, as the incomplete solves with two Gauß–Seidel iterations per call already resemble the spatial structure of the full TenStream results very accurately.

2. When using more Gauß–Seidel iterations but a lower calling frequency in order to maintain the total computational cost, one improves the representation of domain averages at the expense of the spatial structure of the results.

Especially at the surface, though, one should definitely think about using more than just two

Gauß–Seidel iterations per call. To motivate that, Fig. 4.11 shows the same plots as Fig. 4.10 but for net surface irradiances instead of heating rates. As for the heating rates, we can see that the use of more than two Gauß–Seidel iterations per call primarily counteracts the buildup of the MBE with time. In contrast to the heating rates, however, lower calling frequencies do not impact the magnitude of the MAE as much. This indicates that, even at lower calling frequencies, the dynamic TenStream solver is able to adequately capture the spatial structure of the net surface irradiances. Consequently, using our new solver with $N = 20$ iterations at a calling frequency of 60 s leads to better results than achieved with $N = 2$ and a 30 s calling frequency here — both in the solar as well as in the thermal spectral range.

We can thus conclude that, even though the computationally most affordable runs using just two Gauß–Seidel iterations per call lead to promising results, it might be beneficial to use configurations involving slightly more iterations, as they add a comparatively small additional computational cost to the solver while significantly counteracting the buildup of a bias with time. The results for the net irradiance at TOA only underline the statements for the surface and are thus not shown in here.

4.2.5 Visualization of dynamic TenStream heating-rate fields

We want to conclude this first part of the evaluation by visually comparing the dynamic TenStream results to those calculated by the other solvers introduced in Sect. 4.1.2. In contrast to the previous subsection, we restrict ourselves to dynamic TenStream runs with just two Gauß–Seidel iterations per call here. A special focus of this comparison will be on how well dynamic TenStream performs visually in updating the radiative field depending on the calling frequency. To make this comparison as hard as possible for our new solver, we decided to look at the last time step at which the radiative field is simultaneously updated for all three calling frequencies that we consider; that is, at $t = 8960$ s. Instead of this point in time, we could, of course, also take a look at a point in time where the different dynamic TenStream solves have just not been updated. By doing so, one would focus more on how closely not-yet-updated radiative fields still resemble the benchmark result. Here, however, we want to focus more on how well our new solver performs in updating the radiative field depending on how much it has changed between two calls of the radiation scheme. From this point of view, $t = 8960$ s is the last point in time where all three dynamic TenStream runs have just been updated. Hence, they are subject to the most incomplete solves and furthest away from the initial spin-up there, increasing the chance of potential artifacts in the radiative field because the TenStream linear-equation system has not been fully solved for quite a while.

Figure 4.12 shows xz cross sections for this point in time for the solar spectral range, with the colors indicating the heating rates along the cross section using a logarithmic color scale — except for the lowermost row in all the panels, which visualizes the net surface irradiance. In general, the bright yellow areas with correspondingly large heating rates indicate the position of clouds, while the dark areas signify shadows below the clouds. Right from the start, we can see that the largest visual differences do not occur in between the different incomplete dynamic TenStream solves but between the 1D δ-Eddington approximation in panel (a) and the 3D solvers in panels (b)–(f). As 1D radiation does not allow for the horizontal transport of

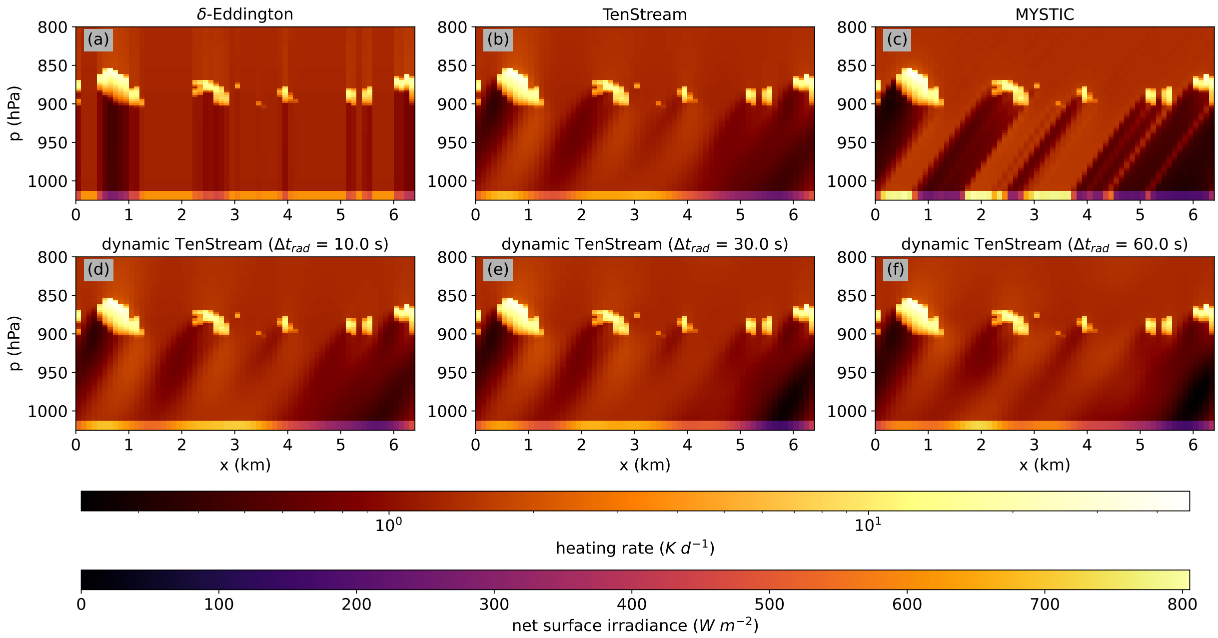

Figure 4.12: *xz* cross section of the heating-rate fields obtained by the different radiative-transfer solvers in the solar spectral range at $t = 8960$ s. Heating rates are visualized by a logarithmic color scale for the δ-Eddington (**a**), original TenStream (**b**) and MYSTIC (**c**) solvers as well as for the dynamic TenStream solver when operated at calling frequencies of 10 s (**d**), 30 s (**e**) or 60 s (**f**). Additionally, the horizontal line at the bottom of each plot visualizes the corresponding net surface irradiances obtained by the solver.

energy, in panel (a), shadows cannot be cast according to the angle of solar incidence; they are only cast right underneath the clouds. This also affects absorption at the ground, with regions of low surface absorption located right below the clouds rather than being displaced like in the MYSTIC benchmark run. We can see that the visual structure of this benchmark result is much better resembled by the TenStream solver shown in panel (b). Here, clouds are also illuminated at their sides, and horizontal transport of energy allows shadows to be cast in the direction of the solar incidence angle. However, we can see that both these shadows and the regions of low surface absorption are much more diffuse than in the MYSTIC benchmark run — although they are still a much better representation of the benchmark than the 1D solution.

Having these characteristics in mind, we can now discuss the results for the new dynamic TenStream solver, which are shown in the last row of Fig. 4.12. The three panels show the results for the new solver when it has been called every 10 s (panel d), 30 s (panel e) or 60 s (panel f) before. At first glance, we can see that the results for the new solver are very similar to those obtained by the original TenStream solver in panel (b), even when operated at the low calling frequency of 60 s. Remember that, in this run, just two Gauß–Seidel iterations toward convergence were carried out at only (8960–8000)/60 = 16 points in time since the spin-up. Since our solver is based on the TenStream solver, this is almost the best result we could have obtained. We can see that, just like the TenStream solver, dynamic TenStream allows for full 3D transport of energy, with shadows and regions of low surface absorption being cast not just directly underneath clouds. Looking closely, one can, however, see differences between the

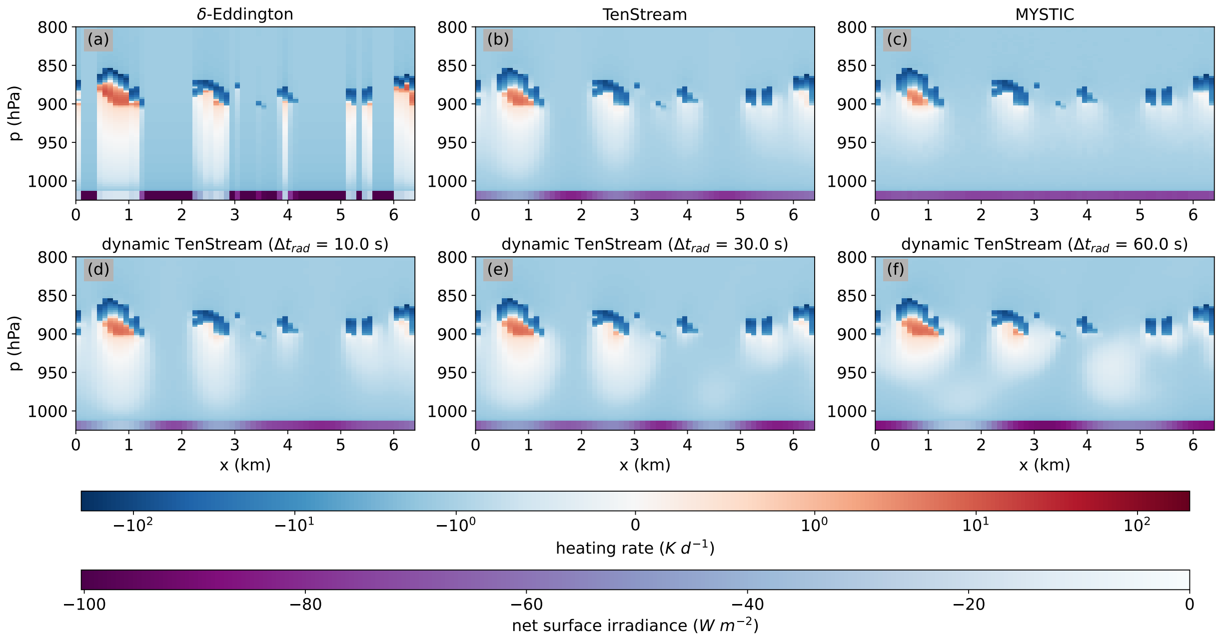

Figure 4.13: xz cross section of the heating rate fields obtained by the different radiative-transfer solvers in the thermal spectral range at $t = 8960$ s. The structure of the plot is identical to Fig. 4.12, except that the color scale is logarithmic for heating rates both above 1 and below $-1\,\mathrm{K\,d^{-1}}$ and linear in between.

results obtained at different calling frequencies. Panel (d), which shows the results for a calling frequency of 10 s, most accurately resembles the original TenStream result, which becomes most visible within the shadows cast by the clouds on the right-hand side of the domain. They are overestimated by both lower-calling-frequency runs between about 5 and 6 km in the x direction, with heating rates being too low there compared to the original TenStream result. Also, surface absorption differs quite a bit between the different dynamic TenStream runs. The structure obtained by the original TenStream solver is again most accurately resembled by the dynamic TenStream run with a calling frequency of 10 s, whereas the surface absorption is overestimated a bit around 5 km in the x direction in the 30 s run and features a much more pronounced region of high absorption at around 2 km in the 60 s run.

Before we make a closing statement, let us also have a look at the results in the thermal spectral range shown in Fig. 4.13. Again, we can see that the result for the 1D δ-Eddington approximation in panel (a) features the most differences when compared to all the other panels showing results obtained by 3D solvers. Compared to the MYSTIC benchmark run, we can see that the thermal shadows cast by the clouds are much more pronounced in 1D and not weakened in direction of the ground due to interactions with neighboring columns. This also leads to a very distinct pattern of strongly negative and not so negative net-surface-irradiance areas at the ground in the 1D results, whereas the net surface irradiance is almost uniform in the MYSTIC benchmark result. This also provides proof for our observation in Sect. 4.2.3, where we have noted that the benchmark results for the net surface irradiance in the thermal spectral range should be pretty uniform in order to avoid the sawtooth pattern in the MAE time series that we typically saw when evaluating solvers at lower calling frequencies. Furthermore, we can also see that the 1D δ-Eddington approximation is not able to consider cloud-side cooling due to its lack of horizontal transport of energy, leading to much more pronounced cloud bottom warming in the 1D results than in the MYSTIC benchmark. The original TenStream solver depicted in panel (b) is able to consider almost all of these 3D effects and is therefore, once more, visually close to the MYSTIC result. Looking closely, we can, however, see that the thermal shadows are a bit more pronounced there, which also leads to regions where the net surface irradiance is a bit weaker below the clouds, in contrast to the very uniform pattern produced by the MYSTIC benchmark solver.

Comparing these results to those of our newly developed dynamic TenStream solver, we can see that it is also almost able to reproduce the results of the original TenStream solver in the thermal spectral range, even when operated at lower calling frequencies. However, the result obtained with a calling frequency of 10 s shown in panel (d) clearly resembles the original TenStream result most closely. At lower calling frequencies, we can see small artifacts, most noticeably in the form of larger or completely floating thermal shadows (the white areas in the plots) that do not seem to belong to any cloud at all, whereas they are normally placed directly underneath them. These regions are residual shadows of already dissolved clouds which the incomplete solves have not been able to get rid of yet. Evidence for this hypothesis is provided by looking at the same plot at previous time steps (not shown here). These residual shadows also influence the net surface irradiance pattern, which is most prominently visible between 1 and 2 km in panel (f). In total, these residual shadows are minor artifacts, though, as we have to consider that we were only able to visualize them by using a logarithmic color scale. And, we also have to keep in mind that, in particular, panel (e), which shows the results achieved at a calling frequency of 30 s, was obtained using a similar computational demand to performing 1D δ-Eddington calculations every 10 s. In contrast to these results, however, the dynamic TenStream result features horizontal transport of radiative energy, resulting in much more realistically distributed heating rates and net surface irradiance patterns.

In summary, we can hence say that for both the solar and thermal spectral ranges, dynamic TenStream is able to visually almost reproduce the results obtained by the original TenStream solver, even when operated at lower calling frequencies. At those frequencies, however, minor artifacts like residual shadows are introduced. The reason for these artifacts is the incomplete solves, which can delay lower-order 3D effects, such as feedback effects from other clouds or the surface. The term "feedback effects" refers to the fact that the 3D radiative effects of a cloud can theoretically alter the conditions determining the 3D radiative effects of any other cloud in the domain. Because these feedback effects require multiple back-and-forth transports of information, they cannot be fully accounted for when solving radiation incompletely. For example, incomplete solves can perfectly consider 3D radiative effects of an emerging cloud at the location of the cloud itself, but the feedback on these heating rates due to lower upward-facing radiative fluxes from the shadow this cloud casts may be delayed to a later call of the scheme if the two Gauß–Seidel iterations that we perform per call are not sufficient to transport this feedback back to the cloud itself.

Chapter 5

Effects of using the dynamic treatment of radiation in large-eddy simulations

The previous chapter showed that the dynamic TenStream solver is capable of determining radiative fluxes and heating rates much more accurately than a classical 1D δ-Eddington approximation, especially in terms of the spatial structure of the results — even when the dynamic TenStream solver is operated at a lower calling frequency than the 1D solver, so that their computational costs are comparable. To demonstrate this, the new solver was applied to a pre-calculated shallow cumulus cloud time series, which enabled its performance to be tested relative to other solvers using the exact same cloud fields. While this approach allowed for easy point-to-point comparisons between the different solvers, it did not show how the use of 3D radiative transfer, in contrast to traditional 1D solvers, changes the evolution of the atmosphere and its clouds, nor whether the dynamic treatment of radiation introduced in Chapter 3 is able to reproduce these 3D-related changes. To investigate this aspect as well, this chapter presents the second part of the evaluation of the dynamic TenStream solver, in which it is coupled to large-eddy simulations performed with the Parallelized Large-Eddy Simulation Model (PALM; Raasch and Schröter (2001); Maronga et al. (2015, 2020)). Based on these simulations, this chapter examines the impact of the incomplete solves in the dynamic TenStream solver on the evolution of the atmosphere and its clouds compared to simulations performed with a classical 1D approximation on the one hand, and full 3D TenStream solves on the other. To this end, this chapter first introduces the model setup and methods used for the evaluation, before presenting and discussing its results.

5.1 Methodology

5.1.1 Simulation setup

All simulations discussed in this chapter were performed with PALM (Raasch and Schröter, 2001; Maronga et al., 2015, 2020). Similar to the pre-calculated cloud fields used in the previous chapter, the aim was to perform simulations in which shallow cumulus clouds develop over the course of the day. The purpose of this subsection is to provide an overview of how PALM was set up in order to generate these shallow cumulus cloud fields, as well as which simulations were ultimately performed. Therefore, the first part of this subsection briefly outlines the domain used, before the second part describes how the model was initialized. The third part of this section then introduces the different radiative transfer solvers that were applied. Finally, the last part gives a detailed overview of all the different PALM simulations that were carried out.

Domain size

In terms of horizontal domain size, all simulations were perfomed with the same 100 m grid spacing that was used in Chapter 4. Unlike there, however, 256 × 256 grid boxes were used, resulting in a 25.6 × 25.6 km^2 large domain — that is sixteen times the size of the domain in the previous chapter. This should lead to much better statistics when investigating cloud characteristics dependent on the radiative transfer model used. In the vertical, however, the grid was reduced to just 80 grid boxes with a spacing of 50 m, resulting in a domain that extends up to 4 km height. This lower vertical resolution was chosen in order to reduce the storage demand of the simulation results. One should note that this vertical resolution however still ensures that the z-x-aspect ratio of the grid boxes is smaller than 1, just like in a typical numerical weather prediction model.

Model initialization

The simulations were set up to take place at the location of Munich, that is at 48.1°N and 11.6°E, but at an altitude of 0 m, i.e., at sea level with a surface pressure of 1013.25 hPa. They were initially started on 14 June 2023 at 22:00 UTC, which corresponds to exactly midnight local time (00:00 CEST on 15 June 2023).

In addition to location and time, PALM is initialized by specifying vertical profiles for potential temperature, total water mixing ratio, and zonal and meridional wind speed. Figure 5.1 shows the relatively simple profiles that were provided as initial conditions for these variables. Let us discuss the setup for the potential temperature profile first. Normally, nocturnal cooling would probably lead to a very stably stratified layer close to the surface at midnight. However, for simplicity — and because the model is given enough time to adjust before sunrise — the atmosphere was assumed to be initially well mixed up to a height of 800 m, with constant values of $\theta = 288$ K for the potential temperature and q = 7 g kg^{-1} for the total water mixing ratio, as shown in panels (a) and (b), respectively. This results in relative humidity values starting at 62.2 % at the surface and reaching up to 93.6 % at 800 m height. In order to prevent clouds from forming right at the beginning of the simulations, a thin, stably stratified layer with a 1 K 100 m^{-1} increase in potential temperature was then added directly above this well-mixed layer, between 800 m and 850 m height. Clouds can then form easily once the surface heats up after sunrise, as the then warmer air parcels at the surface will have the ability to cross this stably stratified layer and thus reach the convective condensation level. By assigning another neutrally stratified layer between 850 m and 1300 m height, these clouds can also easily reach a certain depth. Above 1300 m, a 1 K 100 m^{-1} increase in potential temperature was specified

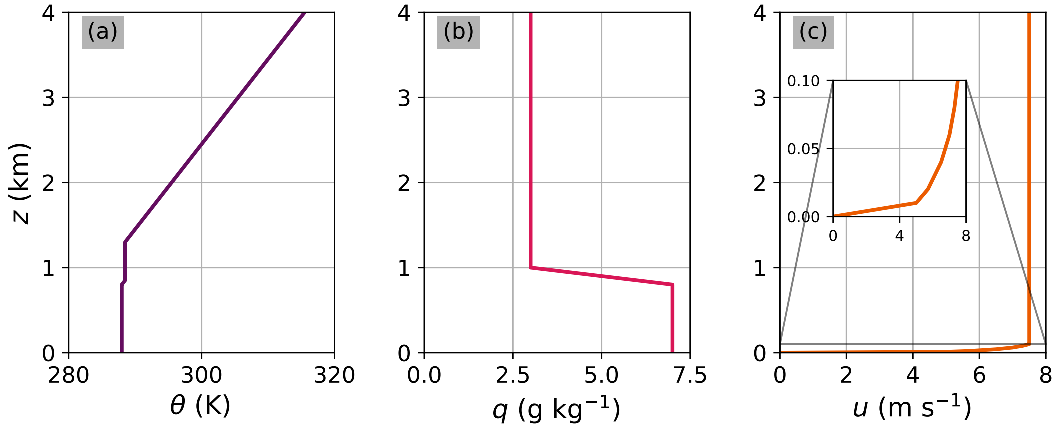

Figure 5.1: Initial profiles of the PALM simulations for potential temperature **(a)**, total water mixing ratio **(b)**, and zonal wind speed **(c)**. The smaller graph in panel **(c)** shows a zoomed-in detail of the zonal wind speed profile between the surface and 100 m height. The temporal evolution of these profiles across the different simulations evaluated in this chapter is provided in Appendix D of this work.

again, ensuring that the clouds cannot easily grow up to the top of the domain at 4 km height.

In contrast to this potential temperature profile, the total water mixing ratio profile illustrated in panel (b) was initialized a bit differently. Above the first well-mixed layer, it decreases at a rate of $2\,\mathrm{g\,kg^{-1}}\ 100\,\mathrm{m^{-1}}$ up to a height of 1000 m. This decrease ensures that the water vapor pressure does not reach supersaturation after relative humidity values have risen to 93.6 % at 800 m altitude. Above 1 km height, the total water mixing ratio was then set to a constant value of $3\,\mathrm{g\,kg^{-1}}$, which, together with the potential temperature profile, results in relative humidity values that slowly decrease from about 52 % at 1300 m height to 45 % at the top of the domain.

In terms of wind, the simulations were initialized with a purely westerly flow, i.e., the meridional wind component was set to $0\,\mathrm{m\,s^{-1}}$ at each height, whereas the zonal component was initialized with a logarithmic wind profile up to a height of 100 m. This logarithmic wind profile is shown in the inset figure of panel (c). It starts with a wind speed of $0\,\mathrm{m\,s^{-1}}$ at the surface and ends with a wind speed of $7.5\,\mathrm{m\,s^{-1}}$ at an altitude of 100 m. Note, however, that a grid spacing of only 50 m was used in the vertical, so that this logarithmic wind profile consists of just three values in the simulations: that is, $0\,\mathrm{m\,s^{-1}}$ in the lowermost layer, $7.2\,\mathrm{m\,s^{-1}}$ in the second and $7.5\,\mathrm{m\,s^{-1}}$ in the third. Above, the zonal component of the wind vector was set to a constant value of $7.5\,\mathrm{m\,s^{-1}}$ for all heights, as shown in the main plot of panel (c). To maintain this westerly flow throughout the entire model run, the Coriolis force was further disabled, since otherwise the mean wind kept rotating clockwise.

In addition to these vertical profiles, the land–surface model of PALM was also used and therefore had to be initialized. Regarding the soil type, the simulations were performed over a flat, short grassland surface, ensuring conservation of water across the combined soil–atmosphere system. The remaining setup of the soil model was largely based on an example in the PALM documentation (PALM, 2025a), with only the soil temperature profile modified to match the rest of the model initialization. To this end, the soil temperature at the surface was set to 288 K, consistent with the potential temperature, and gradually decreased with depth to a deep soil temperature of 280 K across the soil layers. Additionally, the soil moisture was set to $0.18\,m^3\,m^{-3}$ in all layers, which is somewhere between the wilting point ($0.13\,m^3\,m^{-3}$) and the saturation moisture ($0.43\,m^3\,m^{-3}$) of the medium-fine soil type used (PALM, 2025b).

For further details and to ensure that the simulations performed in this thesis can be easily reproduced, you may refer to the parameter file that was used for all the model runs, which is provided in Sect. B.1 of this thesis.

Radiative transfer solvers

The main goal in terms of the setup was to perform different PALM simulations that vary only in the radiative transfer model used. To achieve this, use was made of the fact that the TenStream framework had been coupled to PALM in the past. Similar to the libRadtran library employed in Chapter 4, this framework allows for the application of different radiative transfer solvers using an otherwise identical environment. Besides offering a classical 1D δ-Eddington approximation and the original TenStream solver, all the main features of the dynamic TenStream solver presented in Chapter 3 have also been included into the TenStream framework — among them, the ability to perform incomplete solves, the correct calculation of 3D heating rates in this case, and the speed-up in convergence by properly iterating through the underlying system of linear equations. In addition to that, the TenStream framework is fully parallelized, unlike the libRadtran implementation used so far. It should be noted, however, that the TenStream implementation of the dynamic TenStream solver also differs from the libRadtran version in some respects. Most notably, the TenStream implementation requires the use of the parallel linear algebra library PETSc (Balay et al., 2023), which the libRadtran version was specifically designed to avoid. But also the spin-up method and the execution of the Gauß–Seidel iterations differ slightly between the two implementations.

To highlight these differences, but to also show that the TenStream implementation behaves almost identically to the libRadtran version when used with certain options, an analysis provided in Appendix C compares the two versions of the dynamic TenStream solver in the libRadtran library, where both of them are implemented. The analysis shows that when half as many Gauß–Seidel iterations are used for the calculation of direct radiative fluxes, the TenStream implementation performs nearly identically to the libRadtran version used in Chapter 4. The only significant difference between the two implementations is shown to lie in the spin-up procedure used when performing radiative transfer calculations from scratch. This difference, however, is irrelevant for the PALM simulations conducted in this chapter, since similar to the setup in Chapter 4, a full TenStream solve will be performed at the very first call of the dynamic TenStream solver, followed by incomplete solves with just two Gauß–Seidel iterations every time the radiation module is called thereafter. And since these follow-up incomplete solves use the results from the corresponding previous radiation time step as a first guess, the initial spin-up method is not important at all. For the purposes of this setup, both implementations can therefore be safely regarded as equivalent, as long as in the TenStream implementation, half as many Gauß–Seidel iterations are used for direct radiation as for diffuse radiation.

With this knowledge, the TenStream framework can be used to perform PALM simulations that differ only in the radiative transfer solver employed, including simulations with the newly developed dynamic TenStream solver. Specifically, and similar to the evaluation in Chapter 4, the following radiative transfer solvers have been applied:

1. **A 1D δ-Eddington approximation**
   This 1D approximation represents the type of radiative transfer scheme used in most models today and serves as a worst-case benchmark for evaluating the new dynamic TenStream solver.

2. **The dynamic TenStream solver**
   The TenStream implementation of the dynamic TenStream solver introduced in Chapter 3 is the main focus of this evaluation. For the PALM simulations in this chapter, it was configured to perform a full TenStream solve on its first use. After that, it is operated with a minimum of just two Gauß–Seidel iterations (and one for direct solar radiation) each time it is called. Since the TenStream implementation of the dynamic TenStream solver is fully parallelized, these iterations are performed independently on each subdomain, with communication between the cores occurring only once at the end of each radiation scheme call. The motivation behind this setup mirrors that in Chapter 4: by applying a full 3D solve in the beginning, one can assess whether subsequent incomplete solves lead to different results than simulations using full TenStream solves throughout. Furthermore, as explained in Sect. 4.1.2, using two iterations instead of one ensures that the iteration direction mentioned in Sect. 3.2.2 is altered at least once per call.

3. **The original TenStream solver**
   Simulations performed with the original TenStream solver, also referred to as *TenStream reference solver* from here on, serve as benchmark simulations. Since the dynamic TenStream solver is based on this original TenSteam model, reproducing its results despite using incomplete solves represents the best possible outcome.

The exact `tenstream.options` files used for the simulations with each of these solvers can be found in Sect. B.2. Regardless of the radiative transfer solver applied, the atmospheric trace gas concentrations were provided by the 1976 US standard atmosphere (Anderson et al., 1986). Furthermore, all PALM simulations were carried out with a radiation time step of 30 s, as this proved to be a good speed-accuracy trade-off for the radiative transfer calculations in the previous chapter, which were performed with the same horizontal grid spacing of 100 m.

Overview of the PALM simulation setup

Using these radiative transfer solvers, we can now discuss the different PALM simulations that were performed with them. A schematic overview of the setup is provided in Fig. 5.2. It shows that initially, two different simulations were conducted: a main run and a statistically independent but otherwise identical control run. Both of these simulations were started on 14 June 2023 at 22:00 UTC using the model initialization described earlier. The only difference between the two simulations is that, at certain times (every 150 s in our runs), different random seeds of perturbations were applied to their horizontal wind fields until a predefined perturbation

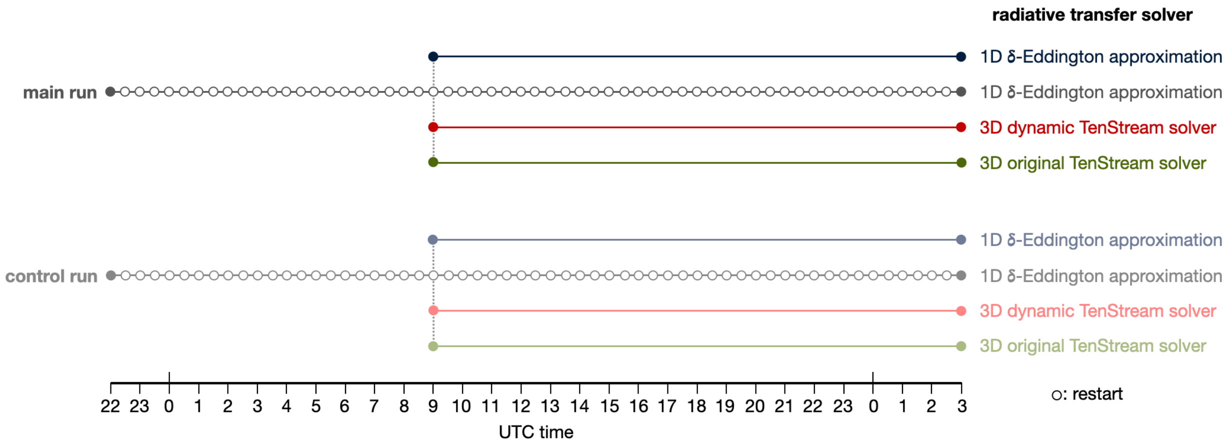

Figure 5.2: Schematic illustration of all PALM simulations discussed in this chapter. On 14 June 2023 at 22:00 UTC, two simulations, a main run and a statistically independent, but otherwise identical control run (both shown in gray here), were started. These two initial model runs were driven by 1D radiation and restarted every 30 minutes, as visualized by the unfilled circles in their respective timelines. Starting at 09:00 UTC on 15 June 2023, three additional restart runs were started from each of the two initial runs. All these in total six restart runs were calculated in one go. They only differed in the radiative transfer model applied: the blue runs were driven by a 1D δ-Eddington approximation, the red ones by the dynamic TenStream solver, and the green ones by the original TenStream solver. Filled circles, as opposed to unfilled circles, visualize the start and end points of every PALM simulation.

energy limit was reached. This limit was set to the default value of 0.01 $m^2\,s^{-2}$. The resulting slightly different wind fields in the main and control run represent the uncertainties in the initial conditions of the setup and effectively create a two-member ensemble, enabling an estimation of whether certain features in the main run are robust or fall within the simulation's inherent uncertainty.

From these slightly differently perturbed initial states, both simulations then ran for 29 hours, ending on 16 June 2023 at 03:00 UTC. During this time, the simulations were first given enough time to spin up and adjust their initial state until the Sun rose on 15 June 2023 at 03:13 UTC. After sunrise, both model runs then encompassed almost a full diurnal cycle, including sunset at 19:15 UTC and ending shortly before the next sunrise on 16 June 2023 at 03:12 UTC. Throughout this entire time, both initial runs were driven by 1D radiation and restarted every 30 minutes. At each of these restart points, visualized by the unfilled circles in Fig. 5.2, PALM stopped the simulation and saved its current state, before restarting it from exactly this saved atmospheric state.

Now, remember that our objective was to conduct different PALM simulations that vary only in the radiative transfer model used. The restart mechanism allows such simulations to be initiated from any intermediate time step shown in Fig. 5.2 by restarting the model run from the corresponding saved atmospheric state with modified runtime parameters, i.e., different radiative transfer solvers in this case. For this evaluation, the simulations were restarted from 09:00 UTC, i.e, 11:00 a.m. local time. At this point in time, shallow cumulus clouds have just started to form in the domain, making it a suitable moment for investigating how the subsequent development of these clouds and the surrounding atmosphere differs depending on the radiative transfer solver used. Consequently, starting from 09:00 UTC, three different restart runs were performed for both the main and the control run, each of them coupled to one of the three different radiative transfer solvers introduced in the previous sub-subsection: either the 1D δ-Eddington approximation (shown in blue in Fig. 5.2), the dynamic TenStream solver (red), or the original TenStream solver (green). These restart runs form the main foundation of this evaluation. Performing them for both the main and the control run was important, as it will allow us to estimate whether differences between runs driven by different radiative transfer solvers are statistically significant or not. Apart from that, all restart runs were conducted in one go, i.e., without further restarts. They ran exactly as long as the two initial runs, i.e., until 16 June 2023 at 03:00 UTC, that is 05:00 a.m. local time, covering approximately ten hours of daytime and eight hours of nighttime, thereby providing a robust dataset for analyzing both of these regimes.

In terms of the general runtime configuration, all simulations were performed with a model time step of 5 s, which is significantly smaller than the radiation time step of 30 s. However, this shorter time step was necessary to satisfy the Courant–Friedrichs–Lewy (CFL) condition for maximum wind speeds of $u = 20\,\mathrm{m\,s^{-1}}$ in the horizontal and $w = 10\,\mathrm{m\,s^{-1}}$ in the vertical, given the horizontal and vertical grid spacings of $\Delta x = 100\,\mathrm{m}$ and $\Delta z = 50\,\mathrm{m}$, respectively, since the condition requires

$$u \cdot \Delta t \overset{!}{<} \Delta x \qquad \text{and} \qquad w \cdot \Delta t \overset{!}{<} \Delta z \tag{5.1}$$

(Courant et al., 1928). Moreover, all simulations were executed on 64 CPU cores, using an 8 × 8 grid of subdomains, each containing 32 × 32 vertical columns. This domain decomposition is particularly important for the dynamic TenStream solver, since its incomplete solves are performed for each of the subdomains in parallel, with interaction between subdomains occurring only once at the end of each radiation scheme call. Information can thus only spread within individual subdomains during the radiative transfer calculations, and propagation to neighboring subdomains is delayed to subsequent calls of the scheme.

Regarding the cloud model, the built-in "morrison" scheme was applied, which uses two-moment cloud microphysics according to Seifert and Beheng (2005), Khairoutdinov and Kogan (2000), Khvorostyanov and Curry (2006) and Morrison and Grabowski (2007). Lastly, the data output for all the model runs was written at a temporal resolution of 60 s. This is considerably coarser than the 10 s resolution used in Chapter 4, but a necessary reduction in terms of the overall storage size of the simulations. Although this means that output is only available for every second radiation time step, Sect. 5.2.2 will show that the domain-averaged cloud characteristics primarily analyzed in this chapter vary on much longer temporal scales, making this reduced output frequency also a scientifically acceptable compromise.

5.1.2 Evaluation methods

The entire evaluation in this chapter is focused on the six restart runs that were just introduced, i.e., the simulations shown in color in Fig. 5.2, which, started from either the main or control run of the setup, are driven by the 1D δ-Eddington approximation, the dynamic TenStream solver, or the original TenStream solver. For simplicity, they will be referred to as "δ-Eddington (control)", "dynamic TenStream (control)" and "TenStream reference (control)" runs from here on. Using these simulations, the primary objective in this chapter is to investigate whether, starting from the same atmospheric state, the atmosphere and its clouds develop differently in these simulations, depending on whether 1D or 3D radiation is applied. Additionally, and even more important for this work, the aim is to assess whether the dynamic treatment of radiation in the dynamic TenStream solver is able to reproduce these potential 3D-related differences.

Accuracy evaluation

One way to investigate the aforementioned differences is by comparing the temporal evolution of certain model quantities across simulations driven by different radiative transfer solvers. Since the solvers fully interact with the model dynamics in these simulations, we cannot just look at radiative quantities for this analysis, but also have the opportunity to evaluate the impact on other model variables, such as the cloud water mixing ratio. And since all of these variables are subject to a diurnal cycle, it is already instructive to study the time series of the respective domain-averaged values. In addition, differences with respect to a benchmark solution are of particular interest. For this evaluation, the TenStream reference run serves as the benchmark simulation. At any point in time, for a given quantity ξ, the accuracy of another simulation with respect to this benchmark run can be quantified using the mean bias error (MBE), which is given by

$$\mathrm{MBE} = \langle \xi \rangle - \langle \xi_{\mathrm{ref}} \rangle = \langle \xi - \xi_{\mathrm{ref}} \rangle. \tag{5.2}$$

If, for given mean bias errors, the relative values $\langle \xi - \xi_{\mathrm{ref}} \rangle / \langle \xi_{\mathrm{ref}} \rangle$ remain well-defined and do not become overly sensitive to $\langle \xi_{\mathrm{ref}} \rangle$ approaching zero, the relative mean bias error (RMBE) can also be considered, which is defined as

$$\mathrm{RMBE} = \frac{\langle \xi \rangle - \langle \xi_{\mathrm{ref}} \rangle}{\langle \xi_{\mathrm{ref}} \rangle}\,. \tag{5.3}$$

Note that small values of $\langle \xi_{\mathrm{ref}} \rangle$ are not problematic per se for this error measure. Rather, it is time series in which $\langle \xi_{\mathrm{ref}} \rangle$ varies strongly throughout the day — as is the case, for example, for net solar surface irradiance — that can cause the RMBE to fluctuate wildly or even become undefined, despite the underlying mean bias errors remaining largely unchanged. In such cases, the MBE provides a more stable and interpretable measure.

Unlike in Chapter 4, point-based error measures, such as the mean absolute error or root-mean-square error, are not utilized in this evaluation. This is because simulations coupled to different radiative transfer solvers lead to diverging atmospheric states, particularly in terms of clouds. Hence, even if two simulations featured clouds with nearly identical characteristics, slight positional differences could result in substantial errors when using point-based error metrics. And since this evaluation prioritizes cloud characteristics over exact locations, such error measures have not been applied. By contrast, the two bias quantities introduced above are based on domain-averaged values and are therefore unaffected by these double-penalty errors. Hence, it is these two metrics that are used to examine whether the general temporal evolution of a variable ξ differs depending on the radiative transfer model used. At this point, it should also be noted that within this simulation setup, a model run driven by a particular radiative transfer solver can only be assumed to be worse than the benchmark run if its MBE is much larger than the MBE of the TenStream reference control run, which can be interpreted as the bias of the benchmark simulation itself. Likewise, the MBE of a simulation driven by a particular radiative transfer solver can only be assumed to be different from that of a simulation driven by another solver if the difference between their MBEs is much larger than the differences between their respective main and control run biases.

Quantification of cloud characteristics

Apart from the temporal evolution of the mean bias error in certain model quantities, a particular interest lies in how the clouds in the simulations develop depending on the radiative transfer solver used. Therefore, inspired by the work of Tijhuis et al. (2024), three quantities are used to characterize the cloud fields at any point in time: cloud cover, average liquid water path (LWP) in cloudy columns, and average cloud depth. Following the approach in Lim and Hoffmann (2023), a grid box is defined as cloudy if its cloud water mixing ratio q_c exceeds $0.01\,\mathrm{g\,kg^{-1}}$. The application of this threshold is necessary because many grid boxes in the simulations are subject to very small q_c values, often as low as $10^{-35}\,\mathrm{g\,kg^{-1}}$. While physically negligible, the large number of these very small values strongly influences quantities such as cloud cover, so they are excluded from the analysis by applying the aforementioned threshold. The three cloud characteristics measures can then be defined as follows:

1. **Cloud cover**
   At any point in time, the cloud cover is given by the fraction of vertical columns in the domain that contain at least one grid box with $q_c > 0.01\,\mathrm{g\,kg^{-1}}$.

2. **Average liquid water path in clouds**
   This quantity is defined as the mean LWP of all vertical columns that contain at least one grid box with $q_c > 0.01\,\mathrm{g\,kg^{-1}}$.

3. **Average cloud depth**
   For every cloudy column (i, j), the index of the highest ($k_{\mathrm{max},i,j}$) and lowest ($k_{\mathrm{min},i,j}$) grid box with $q_c > 0.01\,\mathrm{g\,kg^{-1}}$ is determined. The cloud depth in this column is then given by

   $$\Delta z_{\mathrm{cld},i,j} = \left(k_{\mathrm{max},i,j} + 1 - k_{\mathrm{min},i,j}\right) \Delta z\,, \tag{5.4}$$

   where Δz is the vertical grid spacing. The average cloud depth is subsequently calculated as the mean of $\Delta z_{\mathrm{cld},i,j}$ across all cloudy columns. Note that this method assumes only one cloudy layer per column, as cloud depths are calculated from the vertical extent between the highest and lowest cloudy model level in each column, ignoring any cloud-free layers in between. This assumption, however, is valid for the shallow cumulus clouds used in this evaluation.

At this point, it is important to note that all these quantities describe general characteristics of cloud fields rather than properties of individual clouds. However, similar to the work of Tijhuis et al. (2024), this is precisely the goal, as the objective of this work is to identify systematic differences between cloud fields driven by 1D and 3D radiation. And unlike the work of Tijhuis et al. (2024), the present evaluation also accounts for radiative transfer in the thermal spectral range, enabling the investigation of these differences during nighttime as well.

5.2 Discussion of the results

5.2.1 Simulation overview

Before delving into details, let us first take a look at the general evolution of the clouds in the different simulations. To this end, Figs. 5.3 and 5.4 show ten snapshots of the temporal evolution of the LWP in the simulations driven by the 1D δ-Eddington approximation, the dynamic TenStream solver, and the original TenStream model. For this overview, only simulations started from the main run of the setup are shown. Plots of the LWP are used because this quantity provides a good measure of both cloud position and thickness.

Starting with the first row of panels, we can see that initially, all simulations were indeed started from the same cloud field, allowing differences in their subsequent development to be observed depending on the radiative transfer solver used. Results are shown for 09:01 UTC instead of 09:00 UTC for this first time step, as model output is not immediately available after the simulations start at 09:00 UTC. Two hours later, at 11:00 UTC, the clouds coupled to the different radiative transfer solvers have already developed differently. While they all increased in size compared to the plots at 09:00 UTC, we can clearly see that their organization differs between the simulation driven by 1D radiation shown in panel (d) and the one driven by the original TenStream model shown in panel (f). The clouds in panel (d) are still pretty unorganized. In panel (f), however, we can observe the build-up of cloud streets, just as they were proposed in Jakub and Mayer (2017). These cloud streets are supposed to form perpendicular to the angle of solar incidence and parallel to the mean wind flow. The latter is a constant westerly flow, whereas the Sun is positioned at an azimuth angle of 172° at 11:00 UTC, i.e., roughly in the south. And indeed, the clouds are oriented perpendicular to that direction, forming streets with an east–west orientation, which is particularly visible in the upper part of panel (f). The cloud streets are also visible in the simulation coupled to the dynamic TenStream solver shown in panel (e), providing the first proof of a 3D-related effect that is captured by the dynamic treatment of radiation in that solver.

Moving on to panels (j)–(l), four hours later, the cloud streets in the simulations coupled to 3D radiative transfer have become markedly more pronounced. Because the Sun moved to an azimuth angle of 262° in the meantime, which is almost exactly in the west, the cloud streets are oriented from north to south now, still perpendicular to the angle of solar incidence. In contrast to that, clouds in the simulation driven by 1D radiation are still more or less randomly positioned. This absence of organization in the simulation driven by 1D radiation together with organization perpendicular to the mean westerly flow in the simulations coupled to 3D radiative transfer indicates that the cloud streets in these simulations are not dynamically induced, but really driven by radiation. Apart from these organizational aspects, we can also

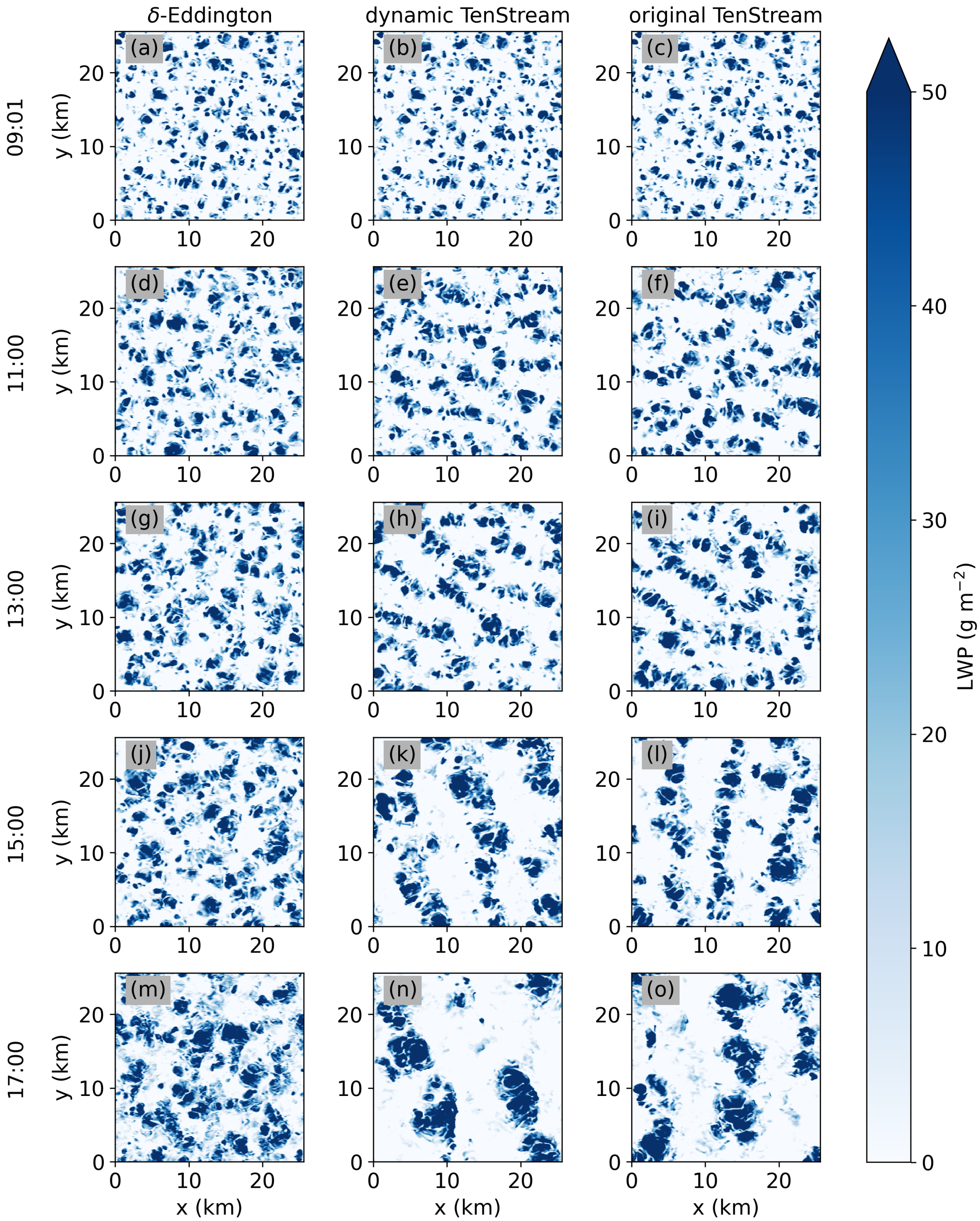

Figure 5.3: First part of the temporal evolution of the liquid water path (LWP) in the PALM simulations driven by the 1D δ-Eddington approximation (left), the dynamic TenStream solver (middle), and the original TenStream model (right), shown for five time steps between 09:01 and 17:00 UTC. To enhance the contrast of the plots, the maximum LWP value for the colorbar was set to 50 g m^{-2} instead of the global maximum value of 363 g m^{-2}.

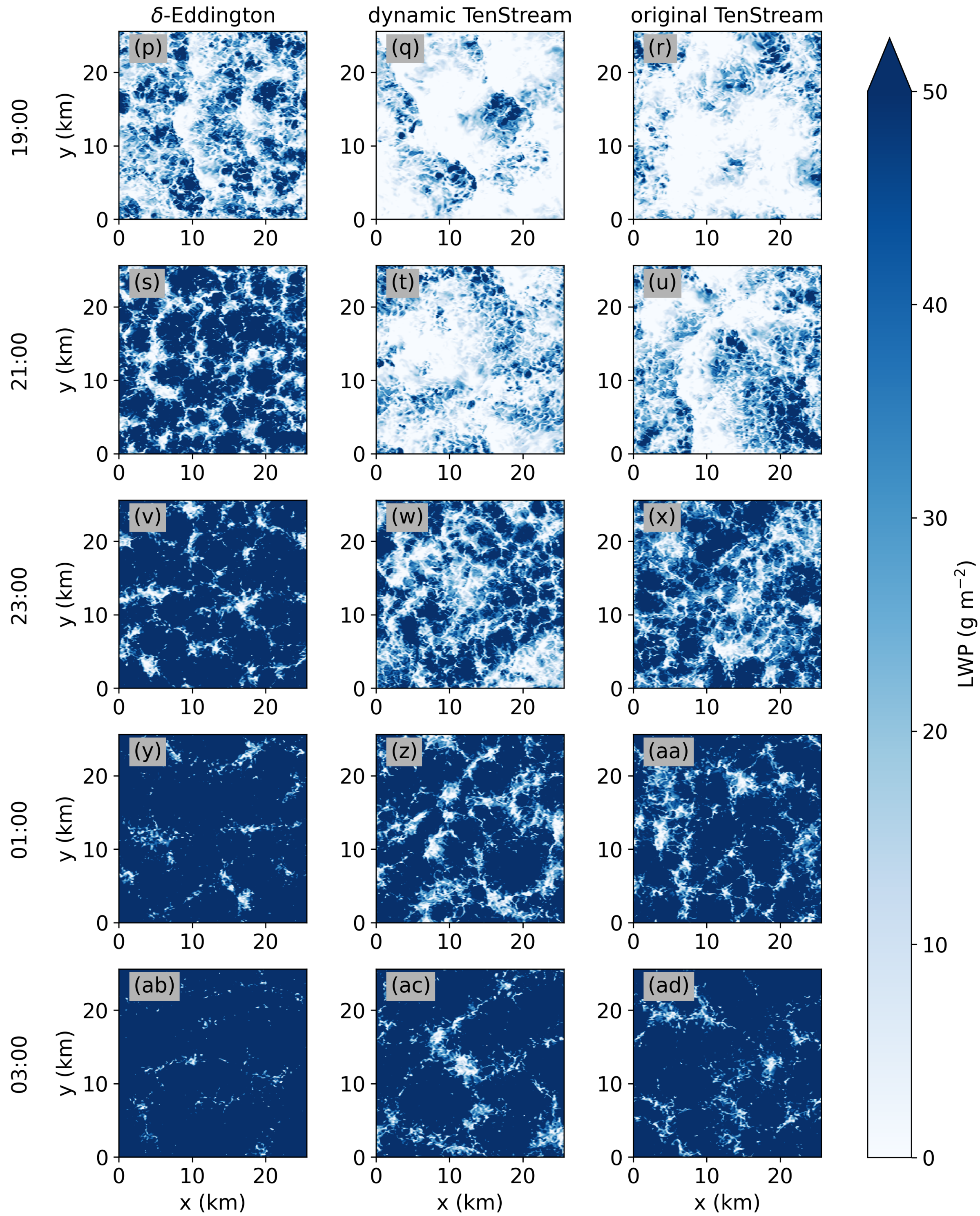

Figure 5.4: Second part of the temporal evolution of the LWP in the PALM simulations driven by the 1D δ-Eddington approximation (left), the dynamic TenStream solver (middle), and the original TenStream model (right), shown for five time steps between 17:00 and 03:00 UTC the following day. To enhance the contrast of the plots, the maximum LWP value for the colorbar was set to 50 g m^{-2} instead of the global maximum value of 363 g m^{-2}.

see that individual clouds in both panels (k) and (l) are noticeably larger and thicker than in panel (j), although the increased thickness is not as apparent in the plots because the maximum value for the colorbar was set to a value well below the global LWP maximum to enhance contrast. Apart from that, however, this development of larger and thicker clouds during daytime in simulations driven by 3D radiative transfer is consistent with findings in other studies (Veerman et al., 2020; Tijhuis et al., 2024).

As the day proceeds, the organization of clouds in the simulations driven by 3D radiative transfer starts to break down and is no longer visible in plots (q) and (r) at 19:00 UTC. However, this is also just minutes before sunset at 19:15 UTC, with the Sun being at an elevation of only 2°. It is also around this point in time that the cloud cover starts to noticeably increase across all simulations. Experiments with the model initialization showed that this behavior is strongly influenced by the initial soil moisture content assigned to the simulations. When reducing its value from $0.18\,\mathrm{m^3\,m^{-3}}$ to $0.16\,\mathrm{m^3\,m^{-3}}$, which is still well above the wilting point of $0.13\,\mathrm{m^3\,m^{-3}}$, clouds start to dissolve around sunset rather than fully covering the domain. While this dissolution in the evening may be more realistic for shallow cumulus clouds on a typical summer day, the presence of clouds at nighttime allows differences between simulations driven by 1D and 3D radiative transfer to be examined after sunset as well, when only radiative transfer in the thermal spectral range plays a role. And as we can see, there is a noticeable change in the characteristics of the clouds in that part of the day that could be observed across a variety of simulation setups experimented with. Namely, it are now the clouds in the simulation driven by 1D radiation that are becoming noticeably thicker than their 3D-driven counterparts, as can be seen starting with panels (p)–(r). This overall structure then remains largely unchanged throughout the rest of the night: all simulations continue to be overcast with clouds and it is only their thickness that further increases throughout the night, with the simulation driven by 1D radiation maintaining the thickest clouds.

Apart from all of this, Figs. 5.3 and 5.4 demonstrate that, at least visually, the clouds in the dynamic TenStream run show very similar characteristics to the ones in the TenStream reference run. Unlike the simulation driven by the 1D δ-Eddington approximation, the dynamic TenStream solver is able to reproduce 3D-related features such as cloud streets and the development of larger and thicker clouds during daytime. However, if the solves produced by the two 3D solvers were identical, they would also lead to the exact same cloud fields in this setup. That said, the fact that the simulations, starting from the same cloud field, develop differently when compared point by point, shows that the small deviations between the solvers highlighted in the previous chapter lead to differently positioned clouds when coupled to model dynamics. These slightly different positioned clouds, however, still represent the reference solution far better than the simulation coupled to the 1D δ-Eddington approximation does.

5.2.2 Effects of radiation on cloud characteristics

Next, let us quantify these observed differences in the cloud fields. To this end, Fig. 5.5 shows the temporal evolution of the cloud characteristics quantities introduced in Sect. 5.1.2 for the six PALM simulations that are considered in this evaluation. Let us discuss the results during daytime first. Panel (a) shows that in this time period, the cloud cover is at around 45% for the

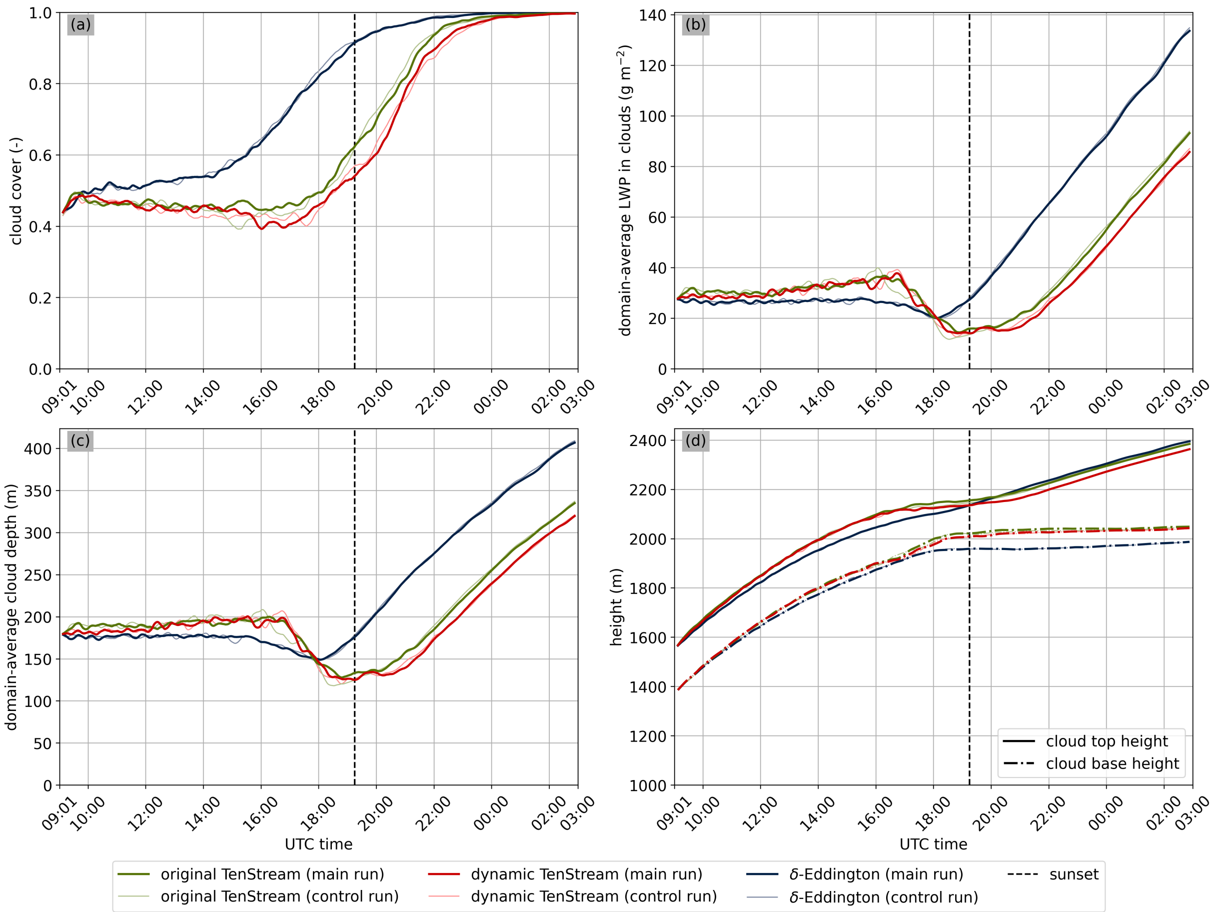

Figure 5.5: Temporal evolution of cloud cover (**a**), average liquid water path in clouds (**b**), average cloud depth (**c**), and cloud top and base heights (**d**) for the six restart runs introduced in Sect. 5.1.1. The vertical line in all the plots indicates the sunset time. To improve readability, a 15-minute running mean was applied to the data.

simulations driven by 3D radiative transfer (shown in red and green), whereas it is noticeably higher, mostly above 50%, for the simulations driven by 1D radiation (shown in blue). The cloud cover also remains relatively constant over time in the 3D simulations, whereas it steadily increases in the 1D cases. This differs from findings in earlier studies, which usually found similar cloud cover in simulations driven by 1D and 3D radiative transfer (e.g., Jakub and Mayer, 2017; Veerman et al., 2020; Tijhuis et al., 2024). However, the overall cloud cover in these studies was usually much lower. And for values above approximately 30%, Tijhuis et al. (2024) actually also found mostly lower cloud cover in simulations coupled to 3D radiation, in good agreement with the results shown here (see their Fig. 3).

Apart from that, panels (b) and (c) show that the clouds in simulations coupled to 3D radiative transfer also become thicker and feature a higher domain-averaged liquid water path during daytime than their 1D-driven counterparts, similar to how it was shown in other studies (Veerman et al., 2020, 2022; Tijhuis et al., 2024). Both effects are not very large, but still bigger than the differences between the respective main and control runs, and thus significant. To

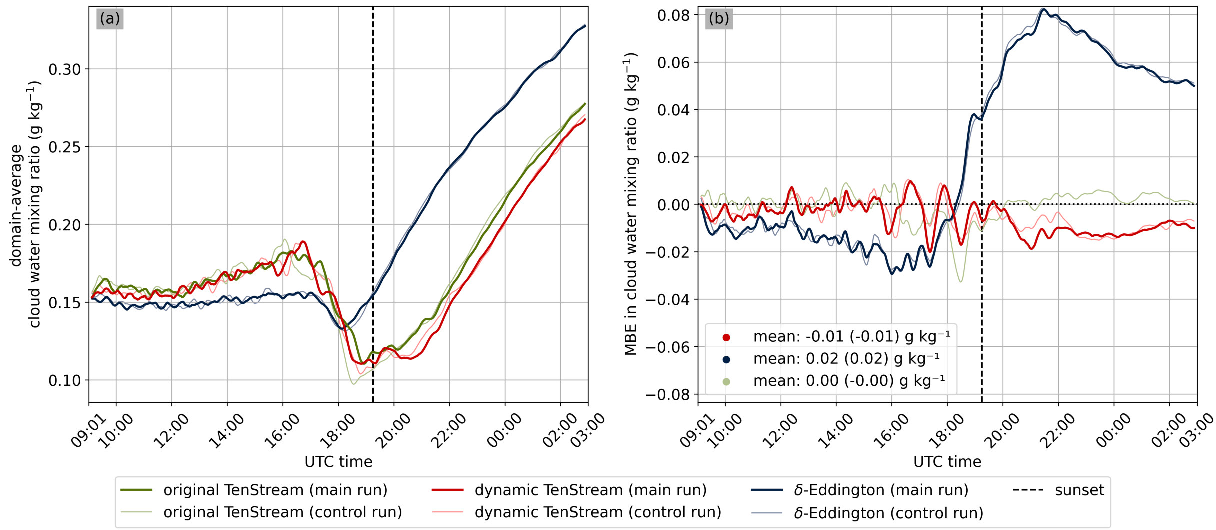

Figure 5.6: Panel (**a**) shows the temporal evolution of the domain-averaged cloud water mixing ratio q_c in cloudy pixels, i.e., for grid boxes with $q_c > 0.01\,\mathrm{g\,kg^{-1}}$. Panel (**b**) visualizes the differences in this time series relative to the TenStream reference run, with the legend providing temporal mean values (the numbers for the corresponding control runs are provided in brackets). As in Fig. 5.5, results are shown for all six (five in panel (**b**)) PALM simulations considered in this evaluation, with the vertical line in both plots marking the time of sunset. To enhance readability, a 15-minute running mean was applied to the data.

put it in numbers, the average LWP in clouds during daytime is $25.5\,\mathrm{g\,m^{-2}}$ in the δ-Eddington main run, whereas it is $28.5\,\mathrm{g\,m^{-2}}$ in the dynamic TenStream main run and $29.3\,\mathrm{g\,m^{-2}}$ in the TenStream reference main run. In terms of depth, these numbers are given by 169.6 m, 178.5 m and 180.8 m, respectively. Hence, the clouds in the simulations coupled to 3D radiation are about 5–7% thicker and feature an about 12–15% larger LWP during the day, with similar numbers for the control run. As we can see in Fig. 5.6, this increase in LWP is also not only caused by the increased depth of the clouds, but also by a higher overall cloud water mixing ratio, i.e., more liquid water per kilogram of air in the clouds coupled to 3D radiation.

Switching to nighttime, panel (a) of Fig. 5.5 shows that all simulations converge toward a completely overcast sky after sunset. However, the corresponding increase in cloud cover occurs significantly earlier in the 1D simulations. Alongside this, the cloud characteristics change substantially. Now, it is the clouds coupled to 1D radiation that become thicker and contain more liquid water. As shown in panel (d) of Fig. 5.5, this increased thickness is primarily due to the cloud base height stabilizing earlier in the 1D simulations, while the cloud top height develops similarly across all simulations. Together, this results in thicker clouds in the simulations driven by 1D radiation, although the development at night itself is not much different from the other simulations. Hence, the thicker clouds at night might actually be related to the higher cloud cover before sunset and the correspondingly reduced radiative energy at the surface, which could lead to an earlier suppression of convection. However, similar model runs with lower initial soil moisture, which did not feature a comparable increase in cloud cover near sunset, still lead to the development of the same nighttime cloud characteristics, i.e., the development of thicker clouds that contain more liquid water in simulations coupled to 1D

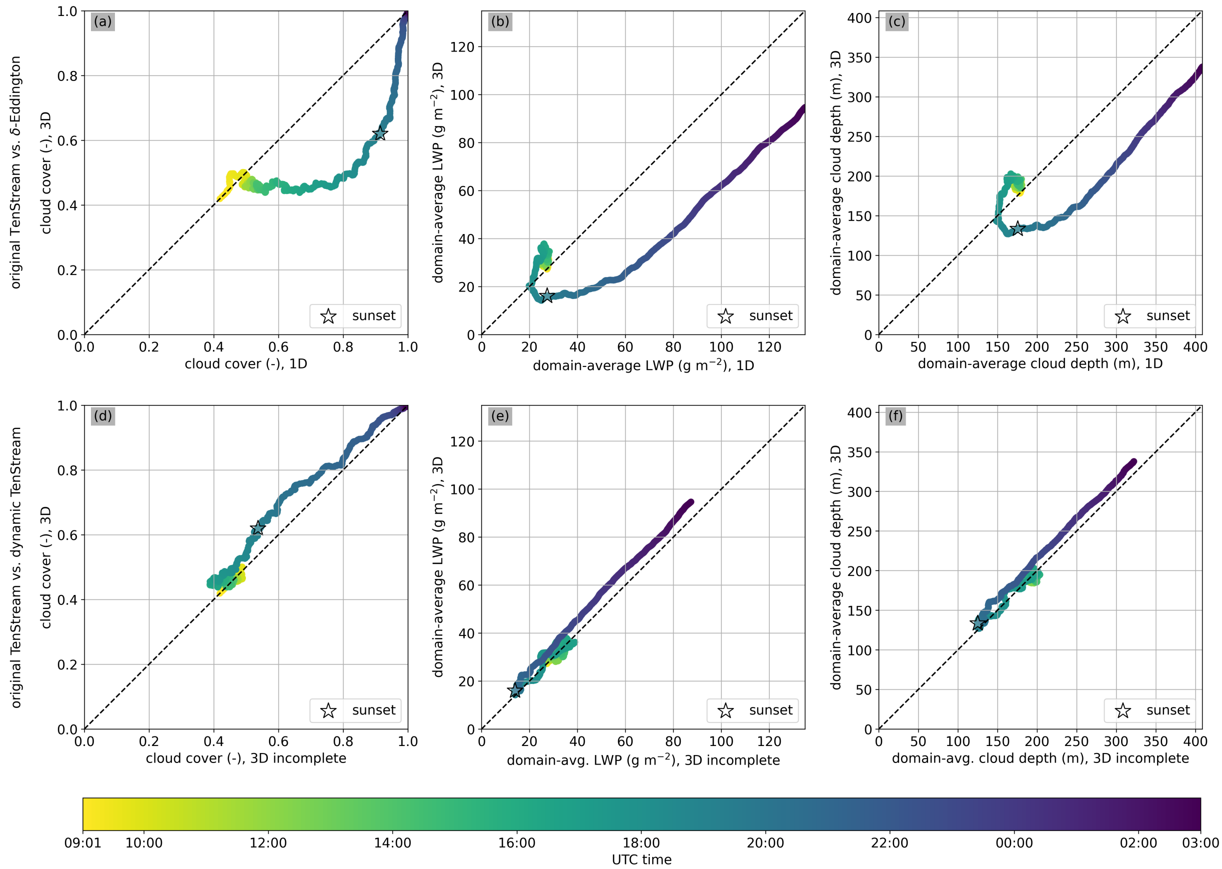

Figure 5.7: Comparison of domain-averaged cloud characteristics between the δ-Eddington and the TenStream reference run (top row), as well as between the dynamic TenStream and the TenStream reference run (bottom row). The individual scatter plots show these comparisons for the cloud cover (**a**, **d**), average LWP in clouds (**b**, **e**) and average cloud depth (**c**, **f**). Every data point is color-coded with respect to time. In addition to that, in each panel, the data point at sunset is highlighted with a star. Only data of the simulations performed from the main run are shown.

radiative transfer, which suggests that these features are fairly robust overall.

At this point, the top row of Fig. 5.7 summarizes the results discussed so far using an alternative visualization motivated by the work of Tijhuis et al. (2024). As it is shown in panel (a), we saw that simulations coupled to the original TenStream solver feature a considerably lower cloud cover than those coupled to 1D radiation, although this might only apply to simulations with cloud covers above about 30%. We have also seen that in terms of liquid water path and cloud depth, we need to distinguish between day and night. During the day, simulations coupled to the original TenStream solver produce thicker clouds with more liquid water. Correspondingly, in panels (b) and (c), all data points before approximately 18:00 UTC lie to the left of the identity line. After sunset, however, these characteristics change, as clouds in the TenStream reference run become thinner and contain less liquid water than their 1D-driven counterparts. Consequently, all post-sunset data points in panels (b) and (c) shift to the right of the identity line.

Up to this point, the discussion has primarily focused on differences between simulations driven by 1D and 3D radiative transfer, without addressing whether the incomplete solves in the dynamic TenStream solver can adequately reproduce these 3D-related differences. To evaluate this, the bottom row of Fig. 5.7 compares the domain-averaged cloud characteristics between the simulation driven by the dynamic TenStream solver and the original TenStream model. The plots show that the cloud characteristics agree pretty well between the two solvers, with small differences between daytime and nighttime. During the day, the clouds driven by the two solvers exhibit nearly identical characteristics, with differences falling within the variability of the simulations themselves, as one can see by going back to Fig. 5.5. At night, however, the incomplete solves in the dynamic TenStream solver slightly underestimate cloud cover, average liquid water path in clouds, and cloud depth. Nonetheless, these discrepancies remain small compared to those observed in the δ-Eddington runs and suggest that the incomplete solves effectively capture all relevant 3D-related effects in terms of cloud characteristics.

It is important to note, however, that the performance of the dynamic TenStream solver is highly dependent on the simulation setup, and particularly on factors such as resolution, domain decomposition, and cloud cover. The relatively high cloud cover in the simulations here, for example, is very beneficial for its performance, as the clouds are relatively close to each other. This proximity reduces the distance — measured in grid boxes — over which information crucial for 3D radiative transfer must be transported. When the overall cloud cover drops below 30% — which is common for shallow cumuli, with Neggers et al. (2003), for instance, reporting cloud covers between 10% and 20% on a prototypical shallow cumulus day — the performance of the solver weakens, as the increased spacing between clouds requires information to traverse more grid boxes, and potentially even multiple subdomains. At this point, performing just two Gauß–Seidel iterations per radiation scheme call may not be sufficient anymore. However, the dynamic TenStream solver was developed with numerical weather prediction models in mind. At their kilometer-scale resolution, clouds are inherently closer together in terms of grid box distance, which should significantly mitigate this issue.

5.2.3 Differences in cloud–radiation interactions

So far, the analysis has primarily focused on the different development of clouds in simulations driven by 1D and 3D radiative transfer, showing that the dynamic TenStream solver successfully captures most of the 3D effects on clouds, despite relying on incomplete solves. However, the causes of these differences have not yet been investigated. Identifying them is challenging, as the clouds in the fully interactive PALM simulations are not only shaped by the corresponding radiative fields but also actively influence them, making it difficult to separate cause from effect. Nevertheless, this section still aims to identify at least some of the connections between the different radiative fields and the resulting cloud characteristics.

Regarding the differences in the radiative fields, we have already seen in Sect. 4.2.5 that sources and sinks of radiative energy in the atmosphere — i.e., heating rates — are distributed very differently in simulations driven by 1D and 3D radiative transfer. This also affects the radiation at the ground, which we will focus on now. To this end, Fig. 5.8 shows five time steps of the temporal evolution of the solar net surface irradiance for the three simulations coupled

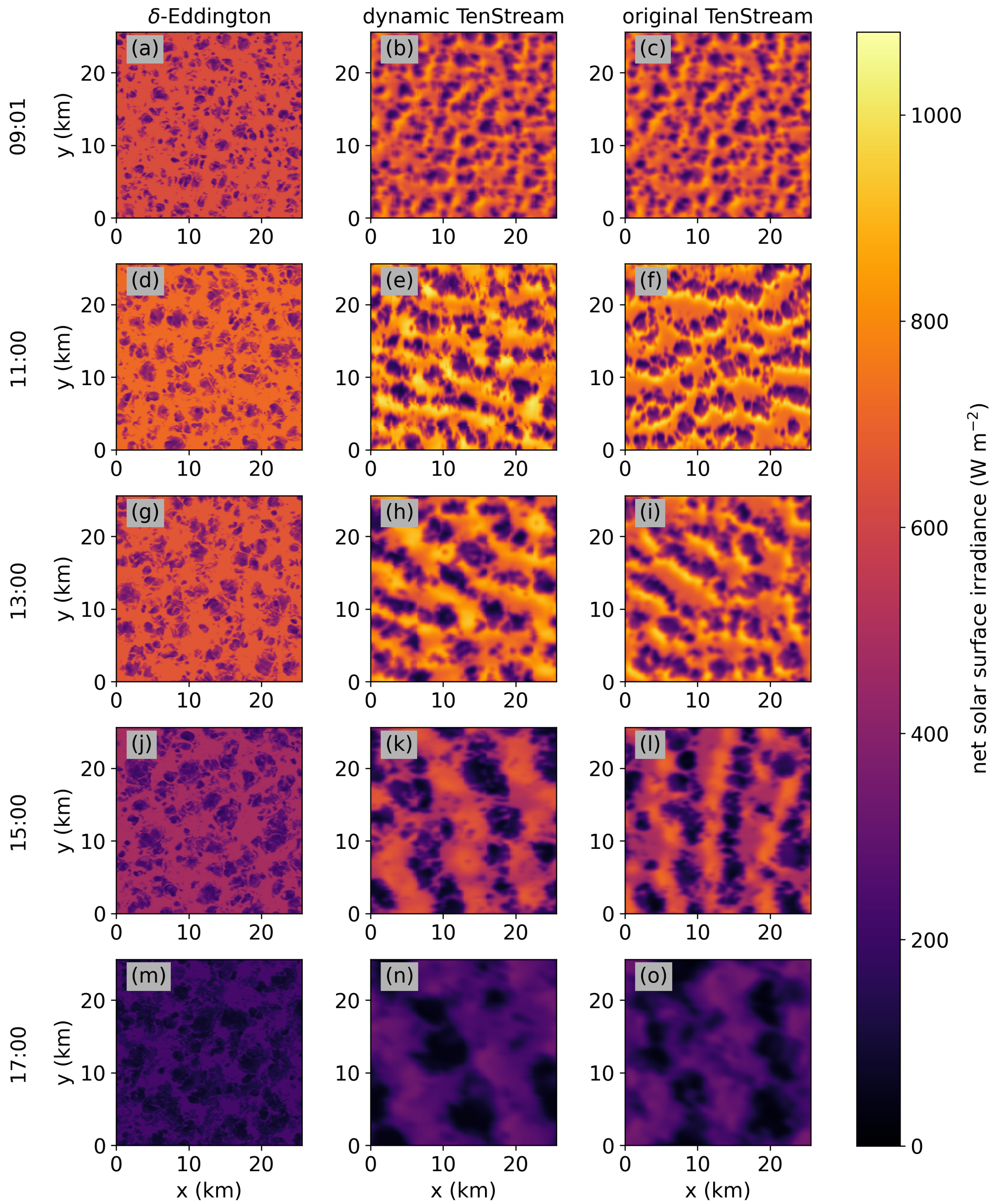

Figure 5.8: Temporal evolution of the net surface irradiance in the solar spectral range for the simulations driven by the δ-Eddington approximation (left), the dynamic TenStream solver (middle), and the original TenStream solver (right), shown for five time steps between 09:01 UTC and 17:00 UTC. Only simulations performed from the main run of the setup are shown.

to the three different radiative transfer solvers considered in this evaluation. Similar to the other results so far, we can directly see distinct differences between the fields produced by the 1D solver (left panels) and those created by the 3D solvers (middle and right panels).

To discuss these differences, let us first look at panels (d)–(f), which show the net surface irradiance fields at 11:00 UTC. Starting with panel (d), remember that in 1D radiative transfer calculations, the atmosphere is divided into independent vertical columns. If such a column contains a cloud, it casts a shadow directly beneath it. These are the purple pixels in panel (d), which perfectly match with the corresponding cloud field, as you can confirm by going back to panel (d) in Fig. 5.3. The remaining columns are all subject to very similar clear-sky conditions, resulting in the pretty uniform background color in that panel. In contrast to that, the surface irradiance field of the original TenStream solver, shown in panel (f), is far more complex. Cloud shadows in this simulation are displaced according to the angle of solar incidence. At 11:00 UTC, the Sun is positioned at an azimuth angle of 172°, or roughly in the south. The cloud shadows are thus shifted slightly to the north of the clouds. Additionally, the net surface irradiance values in the shadows are lower than in panel (d), which is due to the thicker clouds that contain more liquid water in this simulation, as well as generally darker cloud shadows in simulations coupled to 3D radiation (e.g., Gristey et al., 2020). Even more striking are the bright areas at the southern edges of the shadows, where the net surface irradiance exceeds the clear-sky values in panel (d). These so-called cloud enhancements (Tijhuis et al., 2022) occur because radiation coming from the cloud sides or entrapped between the surface and the cloud bases enhances the diffuse downward radiation (Hogan and Shonk, 2013; Hogan et al., 2019). Examining the plot at different times confirms that these cloud enhancements consistently appear at the sunward edges of the cloud shadows. For instance, in panel (c), the Sun is positioned at an azimuth angle of 121° (i.e., in the east–south–east), whereas in panel (l), it is at an angle of 262° (i.e., in the west), causing the enhancements to appear to the east of the shadows in panel (c) and to the west of them in panel (l).

The dynamic TenStream solver in the middle panels of Fig. 5.8 successfully reproduces all these features, although it slightly overestimates the size of the cloud enhancements. This overestimation can be at least partially attributed to the delayed interaction between radiative fluxes at the clouds and the surface when using incomplete solves, especially when this interaction spans multiple subdomains. In such cases, radiative enhancements caused by a cloud may persist at a specific location on the ground even after the cloud has moved on with the mean wind flow, enlarging the associated features on the surface. Following this argumentation, these artifacts are expected to be most pronounced when the solar zenith angle is low, resulting in interactions that involve multiple subdomains, and when the clouds move away from the direction of solar incidence and thus away from their radiatively enhanced areas. And indeed, the overestimation is relatively small in panel (b), where the clouds are moving toward the Sun, but become much more pronounced in panel (k), where the Sun is at a zenith angle of 51° and the clouds move away from the direction of solar incidence, allowing the cloud enhancement areas to increase in size.

By providing a close-up of panel (e), Fig. 5.9 highlights another artifact of the incomplete solves. In this magnification of the net surface irradiance field, we can clearly identify the underlying domain decomposition into 8 × 8 subdomains through discontinuities at the

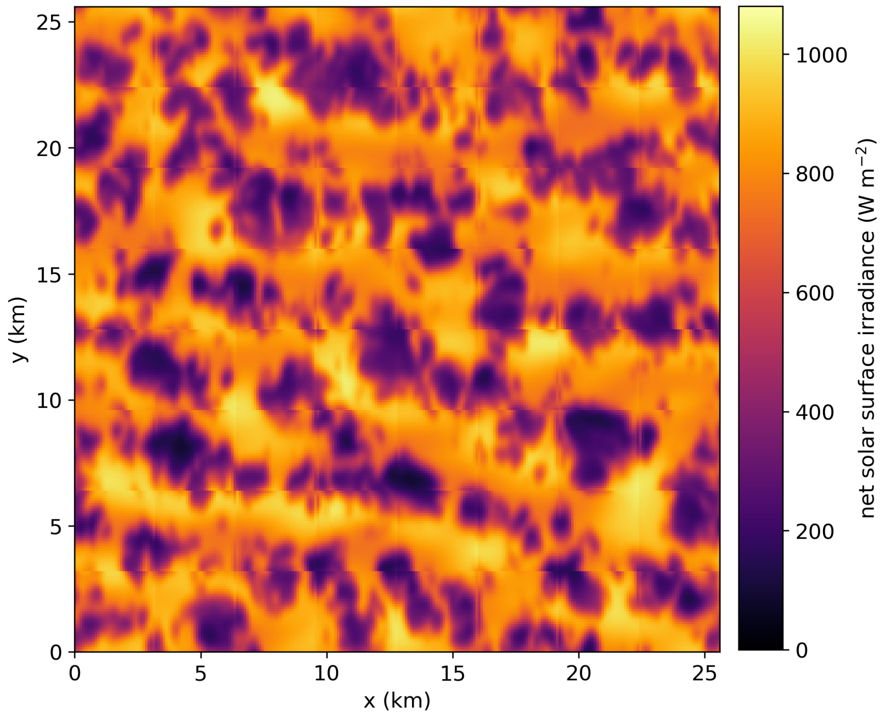

Figure 5.9: Magnified version of panel (e) in Fig. 5.8, highlighting the discontinuities between subdomains in the dynamic TenStream solver.

respective domain boundaries. These discontinuities occur because the incomplete solves in the dynamic TenStream solver are performed independently for all the subdomains, with interactions between them taking place just once at the end of each radiation scheme call. As a result, the individual subdomains have not yet updated their radiative fluxes based on the new boundary conditions at the end of each call, leading to the observed artifacts. However, these discontinuities are relatively small and do not significantly impact the representation of any one of the 3D effects discussed previously.

Now, let us examine how these effects might relate to the differences in cloud characteristics observed in the previous section. To this end, we compare panels (j)–(l) in Fig. 5.8 with panels (j)–(l) in Fig. 5.3. Fig. 5.10 facilitates this comparison by projecting the outlines of the corresponding clouds onto the underlying net surface irradiance fields. Starting with panel (a), we can see that in the simulation coupled to 1D radiation, cloud shadows are unsurprisingly placed directly below the corresponding clouds. This placement of shadows, however, suppresses the updrafts responsible for cloud formation, thereby reducing both the size and lifetime of the clouds (e.g., Schumann et al., 2002; Horn et al., 2015). In contrast, the original TenStream solver in panel (c) shows a displacement of the shadows to the east of the corresponding clouds. Beneath them, we instead observe regions of increased net surface irradiance that even exceed the clear-sky values in panel (a). Panel (b), which illustrates the results for the dynamic TenStream solver, shows a similar positioning of the clouds above areas with enhanced net surface irradiance. However, unlike in panel (c), the clouds in panel (b)

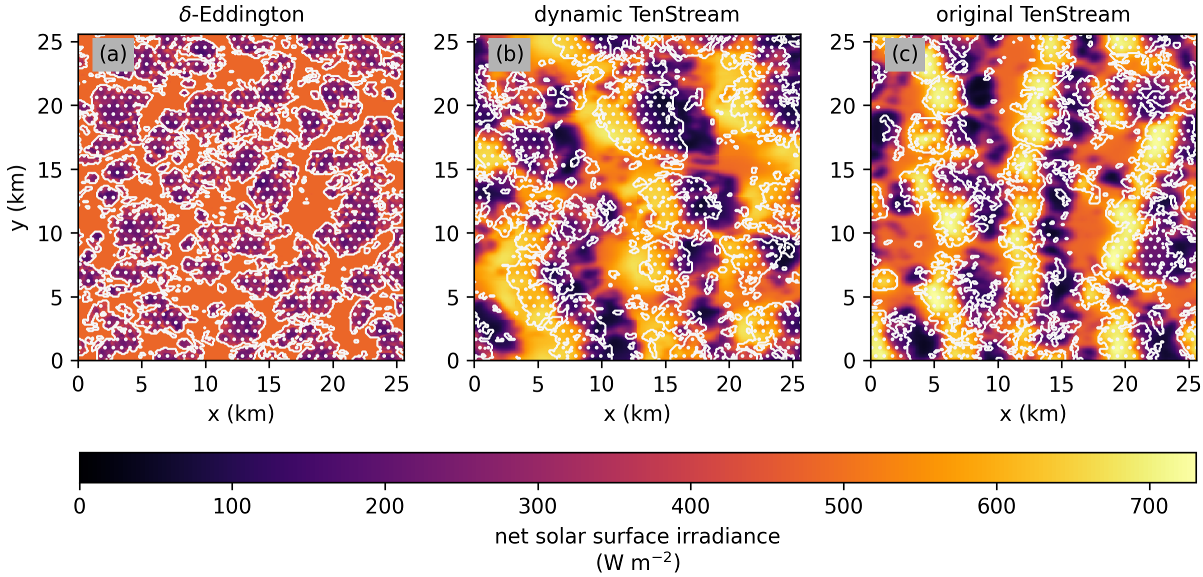

Figure 5.10: Positioning of the clouds relative to the net solar surface irradiance fields at 15:00 UTC in the simulations coupled to the δ-Eddington (**a**), dynamic TenStream (**b**) and original TenStream (**c**) solvers. To this end, the white contour lines in each plot enclose areas where the vertical columns above each grid point contain at least one grid box with $q_c > 0.01\,\mathrm{g\,kg^{-1}}$, classifying them as cloudy. To improve the overall contrast of the plots, the maximum value of the colorbar was set to the highest net surface irradiance across the three panels, rather than the global maximum from all simulations, as it was done in Fig. 5.8.

clearly extend into the shadowed areas as well. This behavior can again be attributed to the incomplete solves, which delay the interaction between radiative fluxes at the clouds and the surface. Specifically, as the clouds move eastward, both the response of the surface to the clouds and the subsequent radiative feedback from the surface back to the clouds can lag behind. Despite this lag, however, substantial portions of the clouds remain positioned over regions of increased net surface irradiance. The enhanced surface heating in these areas subsequently likely strengthens, rather than weakens, the corresponding updrafts, potentially explaining why the clouds in both 3D simulations are larger and thicker compared to those in the 1D simulation during daytime.

Ultimately, this hypothesis is difficult to prove, as the clouds are not only shaped by the corresponding radiative fields but also actively influence them through their evolving characteristics. However, at least some evidence that the observed changes in cloud characteristics are primarily driven by the radiative field can be found in panels (a)–(c) of Fig. 5.8, where the clouds have not yet evolved but are already subject to the discussed effects. Of course, this it is not enough to prove our hypothesis, though. Exactly these challenges associated with the coupling of radiation and model dynamics were also highlighted by Tijhuis et al. (2024), who investigated a similar hypothesis. They came to comparable conclusions, although their more statistical approach found the wind–sun angle to be an even more important factor for the development of larger and thicker clouds during daytime than just the shadow displacement. The mechanism behind this dependence on the wind–sun angle is that clouds driven by 3D

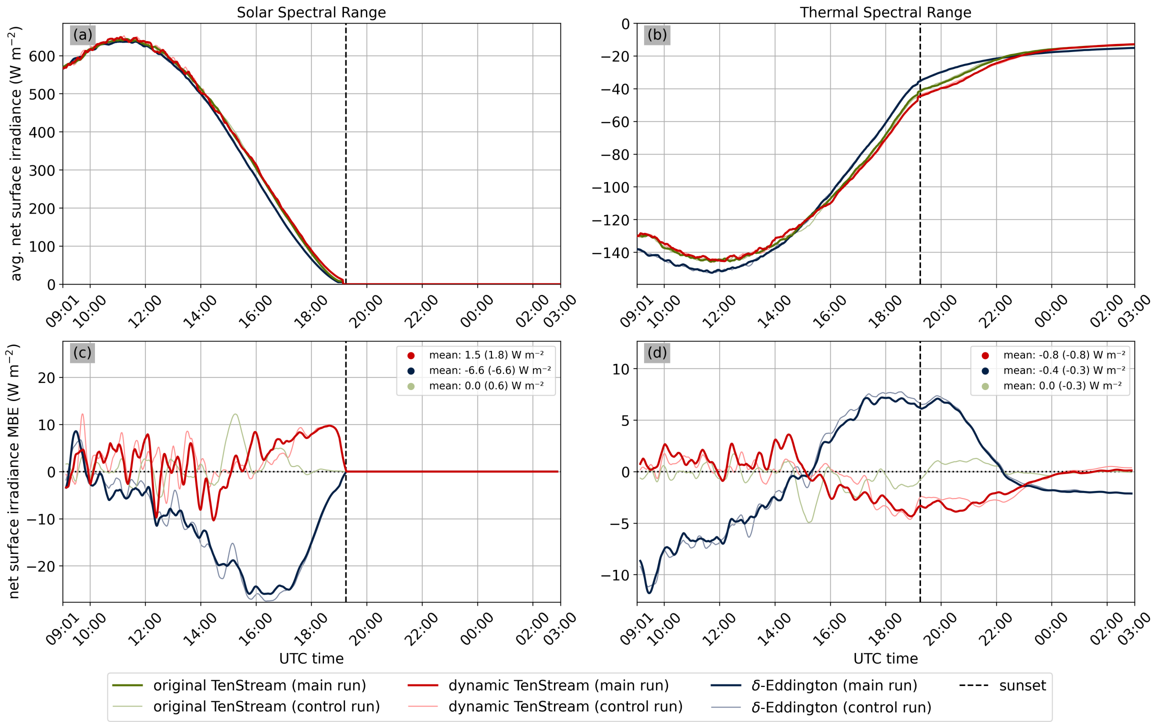

Figure 5.11: Temporal evolution of the domain-averaged net surface irradiance in the solar (**a**) and thermal (**b**) spectral ranges for the six simulations considered in this evaluation. Panels **(c)** and **(d)** show the corresponding differences to the TenStream reference run, with temporal mean differences for the simulations coupled to the different solvers listed in the legends (values for the control runs are given in brackets). To improve readability, a 15-minute running mean was applied to the data in **(c)** and **(d)**. The vertical line in each plot indicates the time of sunset.

radiation cannot significantly outgrow their 1D counterparts if they move into their own shadows, as this suppresses their updrafts, similar to what occurs in 1D simulations. Conversely, if the clouds do not move into their own shadows, the updrafts can persist or even intensify, allowing the clouds to live longer and grow larger (Tijhuis et al., 2024). In the qualitative analysis presented here, no such strong dependency on the wind–sun angle is observed, since the clouds continue to grow even when moving into their own shadows, starting with the panels at 13:00 UTC in Fig. 5.8. However, this does not necessarily negate the existence of this mechanism. Instead, it suggests that in the simulations here, the surface response to cloud movements may be rapid enough to maintain the updrafts in sunlit regions. Otherwise, the results align with those of Tijhuis et al. (2024), although the connection between the clouds and areas of increased net surface irradiance has been emphasized much more strongly here. The representation of these cloud–enhanced areas relies on unique 3D effects, such as cloud–side escape (Hogan and Shonk, 2013) and entrapment (Hogan et al., 2019), which can only be accounted for with full 3D radiative transfer, underscoring the importance of 3D radiative transfer for the correct representation of cloud characteristics in numerical models.

Up to this point, our analysis has primarily focused on differences in the spatial structure of the net surface irradiance fields and their potential relation to the differences in cloud characteristics observed in Sect. 5.2.2. What remains to be investigated, however, is whether the simulations coupled to the δ-Eddington approximation or the dynamic TenStream solver exhibit inherent biases when compared to the TenStream reference solution. To investigate this aspect as well, we turn to Fig. 5.11, where panel (a) illustrates the temporal evolution of the domain-averaged net surface irradiances presented in Fig. 5.8. Despite the substantial differences in the spatial structure of the corresponding fields, we can see that the domain averages evolve quite similarly across all simulations. Panel (c) quantifies the deviations to the TenStream reference run, showing that for most of the day, both the δ-Eddington and the dynamic TenStream runs differ by less than $10\,\mathrm{W\,m^{-2}}$ from the reference solution. It is only after about 14:00 UTC that the δ-Eddington run becomes subject to larger deviations, peaking at around $25\,\mathrm{W\,m^{-2}}$. However, going back to Fig. 5.5, we can see that this divergence coincides with a noticeable increase in cloud cover in the 1D simulations after 14:00 UTC, which naturally reduces the net surface irradiance compared to the 3D simulations, where the cloud cover remains lower. Apart from this specific difference, we can conclude that neither the δ-Eddington approximation nor the dynamic TenStream solver introduces a substantial bias into the net surface irradiance in the solar spectral range. In particular, no significant accumulation of bias over time is observed when using the dynamic TenStream solver, unlike in Sect. 4.2.3. It is worth noting, however, that this bias accumulation mostly remained below $10\,\mathrm{W\,m^{-2}}$ in Fig. 4.6, and hence within the variability of the bias seen in panel (c) of Fig. 5.11.

Before comparing these results to those of other studies, let us first recall which differences in terms of the domain-averaged solar net surface irradiance we generally expect between 1D and 3D radiative transfer solvers. As illustrated in Fig. 4.7, 1D radiative transfer tends to underestimate the net surface irradiance at small solar zenith angles, as it cannot account for cloud-side escape and entrapment, which both enhance the diffuse downward radiation. Conversely, at large solar zenith angles, 1D solvers typically overestimate the net surface irradiance, because they cannot cast shadows at slant angles, thereby underestimating their size. In the fully interactive PALM simulations, however, especially the underestimation at small solar zenith angles is counteracted by the development of larger and thicker clouds in the 3D simulations, which increase absorption and backscattering to space, as explained in Tijhuis et al. (2024). As a result, and consistent with their findings, no substantial differences in the domain-averaged net surface irradiance are observed between simulations coupled to 1D and 3D radiation. This underscores that, at least in the solar spectral range, 3D radiative transfer is primarily important to accurately model the spatial structure of the radiative fields. However, it does not substantially affect the domain-averaged properties of these fields.

This is slightly different in the thermal spectral range. Panel (b) of Fig. 5.11 shows the temporal evolution of the domain-averaged net surface irradiance for the different simulations in this spectral range. We can see that initially, the simulations coupled to 1D radiation experience systematically stronger cooling, similar to what we observed earlier in Fig. 4.6. This is most likely because the increased thermal emission of clouds affects not only the cloudy columns themselves when using 3D radiative transfer, but also neighboring ones due to the horizontal transport of energy in these solvers. Especially early on, when the surface temperature — which primarily determines the upward thermal irradiance at the surface — is still similar across all simulations, the correspondingly increased downward irradiance at the

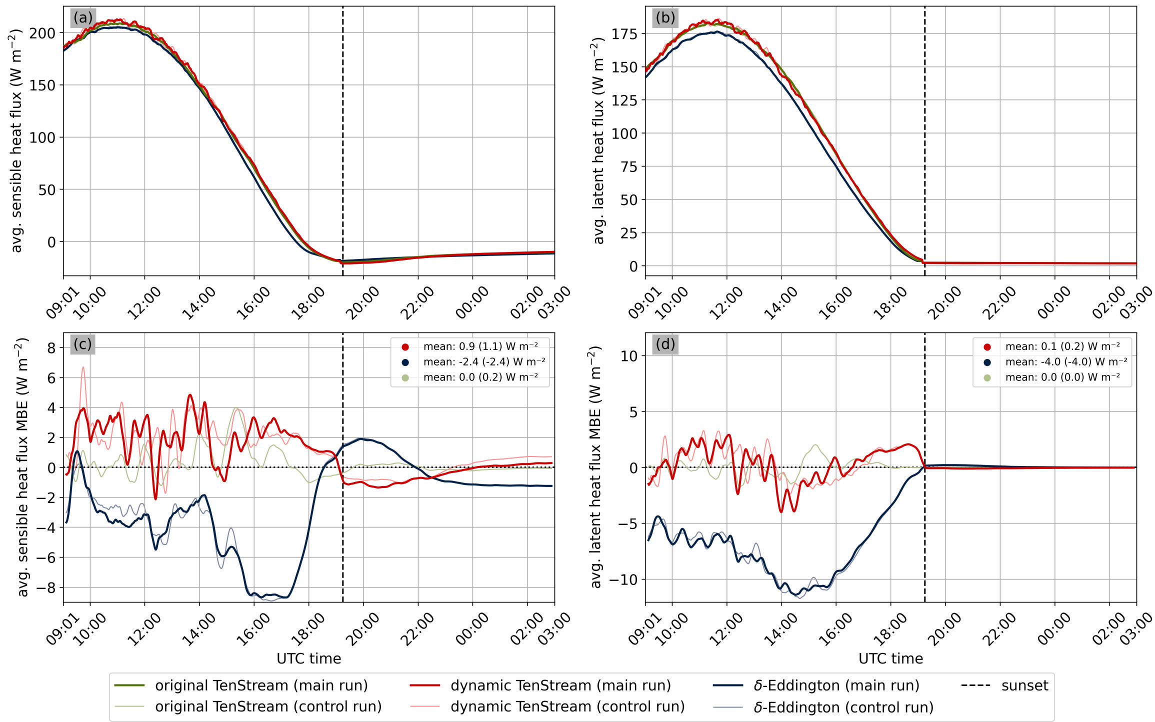

Figure 5.12: Temporal evolution of the domain-averaged sensible (**a**) and latent (**b**) heat fluxes for all six simulations considered in this evaluation. Panels **(c)** and **(d)** show the corresponding differences to the TenStream reference run, with temporal mean differences for the simulations coupled to the different solvers listed in the legends (values for the control runs are provided in brackets). To improve readability, a 15-minute running mean was applied to the data in panels **(c)** and **(d)**. The vertical line in each plot marks the time of sunset.

surface is likely the main factor explaining the systematically weaker net cooling observed with 3D radiation. This pattern changes, however, at around 15:00 UTC, when the cloud cover in the 1D simulations noticeably increases. Along with this shift in cloud cover, the net thermal emission in the 1D simulations decreases and becomes lower than in the 3D simulations. As shown in Sect. 4.2.5, net thermal emission is significantly reduced below clouds when using 1D radiative transfer, explaining this noticeable decrease in net thermal emission as the cloud cover starts to rise. In contrast, the simulations coupled to the dynamic TenStream solver show much smaller deviations from the reference run, with differences staying below $5\,\mathrm{W\,m^{-2}}$ throughout the entire time, as one can see in panel (d) of Fig. 5.11. This once more indicates that the dynamic TenStream solver, despite its use of incomplete solves, is still able to capture all the key features of the full 3D solutions.

Since the temporal evolution of the domain-averaged net solar surface irradiance showed only minor differences between the solvers, the observed differences in the domain-averaged net thermal surface irradiance must be balanced by other components of the surface energy budget. In general, the net solar irradiance at the ground is balanced by net thermal emission, the ground heat flux, and the release of sensible and latent heat into the atmosphere. Hence, the observed differences in the net thermal irradiance can result in changes in any of the latter three components. To investigate which of them compensates for the observed differences, Fig. 5.12 illustrates the temporal evolution of the sensible and latent heat fluxes across all simulations considered in this evaluation. Looking at panels (a) and (b), first note that the Bowen ratio — i.e., the ratio of the sensible to the latent heat flux (Bowen, 1926) — is larger than one in all simulations. This is rather unusual for vegetated surfaces such as the flat grassland used here, and more characteristic of urban environments or semi-arid regions (e.g., Stull, 2006; Kotthaus and Grimmond, 2014), suggesting that the soil in the model is relatively dry. Apart from this general remark, panel (a) of Fig. 5.12 shows that the sensible heat flux evolves quite similarly across all simulations until about 15:00 UTC. At that point, the simulations coupled to 1D radiation begin to slightly diverge from those using 3D radiation. As discussed earlier, this divergence coincides with a substantial increase in cloud cover in the 1D simulations at that time, subsequently leading to a reduction of the net solar surface irradiance. Panel (a) indicates that this reduction in solar energy input also affects the release of sensible heat into the atmosphere, which is subsequently reduced by up to $8\,\mathrm{W\,m^{-2}}$ in the 1D simulations, as we can see in panel (c).

A much more striking difference between the 1D and 3D simulations is revealed in panel (b), which shows that the domain-averaged latent heat flux is considerably higher in the simulations coupled to 3D radiation, accompanied by a slightly faster decrease in soil moisture in these simulations (not shown here). This finding suggests that the reduced thermal emission in the 3D simulations observed in Fig. 5.11 — which could be balanced by a decrease in net solar irradiance, or by increases in the ground, sensible, or latent heat fluxes, or a combination thereof — is primarily offset by an increased release of water vapor into the atmosphere. In PALM, this latent heat flux is parameterized as

$$LE = -\frac{\rho\, l_v}{r_a + r_s}\left(q_v\left(\frac{\Delta z}{2}\right) - q_{v,\mathrm{sat}}(T_0)\right), \tag{5.5}$$

where ρ is the density of dry air, $l_v = 2.5 \times 10^6\,\mathrm{J\,kg^{-1}}$ is the specific latent heat of vaporization, r_a and r_s are the aerodynamic and surface resistances (in units of $\mathrm{s\,m^{-1}}$), $q_v\left(\frac{\Delta z}{2}\right)$ is the water vapor mixing ratio at height $z = \frac{\Delta z}{2}$, with Δz denoting the vertical grid spacing, and $q_{v,\mathrm{sat}}(T_0)$ is the saturation vapor mixing ratio at the radiative temperature T_0 of the surface skin layer (Maronga et al., 2020). An investigation of the individual components of this expression showed that the higher latent heat flux in the 3D simulations is most likely caused by an increase in $q_{v,\mathrm{sat}}(T_0)$, and thus in the radiative temperature T_0 of the surface skin layer. This increase in T_0, in turn, is governed by imbalances in the ground energy budget, following the relationship

$$C_0 \frac{\mathrm{d}T_0}{\mathrm{d}t} = R_n - H - LE - G, \tag{5.6}$$

where C_0 denotes the heat capacity of the surface skin layer (in units of $\mathrm{J\,m^{-2}\,K^{-1}}$), R_n is the net radiative intake at the surface, H is the sensible heat flux, and G is the ground heat flux (Maronga et al., 2020). In the 3D simulations, R_n is increased across both spectral ranges, while H, LE, and G initially remain unchanged. This explains the increase in T_0 in these simulations, which subsequently leads to increases in H, LE, and G. Since $q_{v,\mathrm{sat}}$ is directly proportional to the saturation vapor pressure e_w^*, which itself depends exponentially on T_0 (see Eq. (2.21)), even small increases in T_0 can lead to substantial increases in $q_{v,\mathrm{sat}}(T_0)$ and, consequently, in LE, explaining why the elevated skin layer temperature affects the latent heat flux more strongly than the sensible one. Overall, the increased latent heat flux in the 3D simulations provides another potential explanation for why clouds grow thicker and larger during the day than their 1D-driven counterparts, as more water vapor is released into the atmosphere with 3D radiation. And looking at panel (d), we can see that the simulations coupled to the dynamic TenStream solver also capture this effect, with deviations from the TenStream reference solution remaining below $4\,\mathrm{W\,m^{-2}}$ compared to deviations of more than $10\,\mathrm{W\,m^{-2}}$ for the 1D simulations.

All in all, two potential links have now been identified between radiation and cloud characteristics that help explain why clouds in simulations coupled to 3D radiative transfer grow larger, become thicker, and contain more liquid water during the day than their 1D-driven counterparts:

1. First, clouds in simulations coupled to 3D radiative transfer turned out to be positioned above areas of increased net surface irradiance, rather than above their own shadows. This placement causes the associated updrafts to persist or even strengthen instead of weakening.

2. Additionally, 3D radiative transfer reduces the domain-averaged net thermal emission at the ground, most likely because the increased thermal emission of clouds enhances the downward longwave radiation not only in cloudy, but also in neighboring columns. This reduction in net thermal surface irradiance affects the ground energy budget and appears to be primarily balanced by an increase in the domain-averaged latent heat flux, resulting in a greater release of water vapor into the atmosphere.

Both of these effects are captured not only by the full original TenStream model but also by the newly developed dynamic TenStream solver. This demonstrates that the radiative transfer solver proposed in this thesis offers a computationally efficient approach that not only captures the essential features of 3D radiative fields, but also reproduces most of the resulting cloud characteristics in fully interactive PALM simulations. These include the development of larger and thicker clouds containing more liquid water during the day, as well as thinner clouds with less liquid water at night. The causes of these nighttime cloud characteristics, however, are more difficult to explain and not addressed in this thesis.

Before concluding this chapter, it should also be noted that the original TenStream solver used as a reference in this chapter is an approximation itself. The differences between this approximation and highly accurate 3D radiative transfer solvers such as the Monte Carlo model MYSTIC have been discussed in Chapter 4 and are mostly much smaller than the 3D radiative effects and their subsequent influences on cloud development observed in this chapter. Especially the results during daytime have furthermore been shown to be in good agreement with those of Tijhuis et al. (2024), which were obtained with a GPU-accelerated Monte Carlo ray tracer. Thus, while the exact magnitude of some of the effects discussed in this chapter might slightly vary when using a highly accurate 3D radiative transfer model such as MYSTIC, the key features and underlying mechanisms are expected to remain unchanged.

Chapter 6

Summary and outlook

As numerical weather prediction (NWP) models move toward higher horizontal resolutions, inter-column three-dimensional (3D) radiative effects become increasingly important. Already at cloud-resolving scales, cloud shadows, for instance, are no longer confined to the vertical model columns of their respective clouds, but can extend into several neighboring ones. Additionally, radiation scattered from cloud sides can enhance the diffuse downward radiation in adjacent cloud-free columns (Hogan and Shonk, 2013), whereas radiation entering through gaps between clouds can become trapped between them and the surface, thereby increasing the diffuse downward radiation below clouds as well (Hogan et al., 2019). In the thermal spectral range, higher-resolution models feature a more pronounced cloud-side cooling, as clouds are resolved by a larger number of grid boxes, consequentially allowing for a more detailed representation of this effect. Together, all of these 3D radiative effects influence the spatial distribution of sources and sinks of radiative energy in the atmosphere, which in turn drive the weather. Accurately capturing these effects is therefore crucial for predicting future atmospheric states. However, due to the high computational cost of 3D radiative transfer solvers, most NWP models still treat radiation as a one-dimensional (1D) process, limiting interactions to the vertical and neglecting any horizontal transport of energy.

To address this issue, Jakub and Mayer (2015) developed the TenStream solver, a 3D radiative transfer approximation that, compared to other 3D solvers, is relatively fast. It extends the well-established two-stream method to three dimensions by introducing additional streams that account for the horizontal transport of energy through Earth's atmosphere. Building upon this method, this thesis introduced the dynamic TenStream solver, a further acceleration of the original TenStream model. Currently designed for the use at subkilometer-scale horizontal resolutions, where model grid boxes can be assumed homogeneous, it relies on the idea that the radiative field does not totally change between two consecutive calls of the radiation scheme. Based on this idea, it introduces a dynamic treatment of radiation, which is based on three main concepts: First, radiation is not calculated from scratch every time the solver is called. Instead, a time-stepping scheme is used to update the radiative field based on the result from the previous radiation time step. Second, starting from this previously calculated result, convergence towards the new solution is accelerated by optimizing the iteration procedure through the underlying system of linear equations, taking parameters such as the angle of solar incidence into account. And third, since the updated solution is not expected to be radically different from the previous one, only a limited number of iterations towards convergence are performed, essentially exiting the algorithm before full convergence is reached. This last concept in particular allows the radiative transfer calculations to be performed independently for each grid box within a single iteration, as each grid box only needs to take the ingoing radiative fluxes from its neighbors to determine updated outgoing ones. Through multiple iterations or sequential execution across a model subdomain, information can still be propagated through multiple grid boxes, enabling the transport of radiation across entire subdomains within a single call of the model. At its core, however, each update step remains local, relying only on input from neighboring cells. This brings radiative transfer much closer to the treatment of advection in the dynamical core of an NWP model, as both use previously calculated results to update their variables and thereby only require access to neighboring grid boxes. And compared to traditional 3D radiative transfer models, where the calculation of any radiative flux generally depends on all other radiative fluxes in the domain, this also significantly simplifies model parallelization.

To demonstrate the feasibility of this new concept, the dynamic TenStream solver was implemented into the libRadtran library for radiative transfer (Emde et al., 2016; Mayer and Kylling, 2005) and applied to 100 time steps of a shallow cumulus cloud time series prepared by Jakub and Gregor (2022). The performance of the new solver, operated with just two iterations toward convergence per call, was then evaluated in terms of both speed and accuracy by comparing it to three other, well-established radiative transfer models: a 1D δ-Eddington approximation, the original TenStream solver and the Monte Carlo model MYSTIC (Mayer, 2009), which served as a benchmark. Using these models, the dynamic TenStream solver was shown to be about three times slower than the 1D δ-Eddington approximation, but noticeably faster than the other two 3D solvers in the comparison, which were at least a factor of five slower. To evaluate the accuracy of the different solvers, their calculations of heating rates and net irradiances at the upper and lower domain boundaries were compared against the MYSTIC benchmark solution. Since all solvers were uncoupled from model dynamics for this evaluation, they did not influence the evolution of the clouds in the time series, enabling direct point-to-point comparisons between the different solvers. The high temporal resolution of the time series further allowed for an assessment of the performance of the new solver across different calling frequencies, from the highest possible frequency of 10 s to lower ones, where the radiative field changes more rapidly in between different radiation time steps. In terms of the spatial structure of the results, the dynamic TenStream solver was shown to closely resemble the MYSTIC benchmark results, capturing both heating rates and net irradiances at domain boundaries much more accurately than the 1D δ-Eddington approximation does. When called less often and averaged over time, the new solver was also shown to outperform δ-Eddington calculations carried out with a similar computational demand. At these lower calling frequencies, however, the incomplete solves in the dynamic TenStream solver also caused an increase in bias over time, whose magnitude got larger the lower the calling frequency was. However, even at the lowest calling frequency investigated, this build-up of bias eventually stabilized and, with respect to heating rates, stayed below the bias of any 1D run at any point in time. In addition to that, increasing the number of iterations toward convergence was shown to notably reduce the magnitude of this bias, with only a minor increase in computational cost.

Altogether, the performance evaluation clearly demonstrated the ability of the dynamic TenStream solver to especially capture the spatial structure of 3D radiative fields. Given the markedly different structure of these fields compared to their 1D counterparts, the next step was to investigate whether simulations coupled to the dynamic TenStream solver produced clouds that were substantially different from those in 1D simulations and more similar to those in full 3D simulations. To explore this, PALM (Raasch and Schröter, 2001; Maronga et al., 2015, 2020) was used to set up a large-eddy simulation with an interactive land surface, in which shallow cumulus clouds developed over the course of the day. Once the clouds had formed, three restart runs of this simulation were performed, each coupled to a different radiative transfer model: either a classical 1D δ-Eddington approximation, the original TenStream model, or the new dynamic TenStream solver. Starting from the same cloud field, this setup then enabled a comparison of how the clouds in the simulation develop depending on the radiative transfer model used. To analyze the differences between the three simulations, a distinction was made between daytime and nighttime conditions. During the day, clouds in the 3D simulations were shown to organize into cloud streets aligned perpendicular to the angle of solar incidence, consistent with the findings of Jakub and Mayer (2017), whereas in the 1D simulation, they remained more or less randomly positioned. Additionally, the clouds in the 3D simulations grew larger, became thicker, and contained more liquid water than those in the 1D simulation, in agreement with Veerman et al. (2020, 2022) and Tijhuis et al. (2024). After sunset, however, these characteristics changed. Now, the clouds in the 1D simulations became thicker and contained more liquid water than their 3D-driven counterparts. Throughout both day and night, the dynamic TenStream solver successfully captured all these 3D effects, with discrepancies from the original TenStream model remaining small compared to the differences observed for the 1D simulation. In contrast to these domain-averaged cloud characteristics, the positioning of the individual clouds in the two 3D simulations however indeed differed over time, illustrating that even the small differences in their corresponding radiative fields influenced individual cloud development over time, despite sharing very similar overall characteristics.

Building on these observations, this thesis then sought to identify the mechanisms driving the described differences in cloud characteristics, with a particular focus on the daytime regime. A comparison of the corresponding net solar surface irradiance fields revealed first key differences. As expected, cloud shadows in the 3D simulations were displaced according to the angle of solar incidence, in contrast to the 1D simulation, where they were positioned directly beneath the corresponding clouds. At the sunward edges of the shadows, furthermore, areas of enhanced net surface irradiance were identified in the 3D simulations, with values even exceeding those in the clear-sky columns of the 1D simulation. Unlike their 1D-driven counterparts, the clouds in the 3D simulations turned out to be located above these areas of enhanced net surface irradiance. This positioning likely caused the associated updrafts to persist rather than weaken, promoting the observed differences in cloud characteristics. In addition, 3D radiative transfer was found to reduce the domain-averaged net thermal emission at the ground, most likely because the increased thermal emission of clouds enhances the downward longwave radiation not only in cloudy, but also in neighboring columns with this type of solver. This reduction in net thermal surface irradiance affected the ground energy budget in these simulations and appeared to be primarily balanced by an increase in the domain-averaged latent heat flux, resulting in a greater release of water vapor into the atmosphere. Together, both of these effects helped explain why clouds in the 3D simulations grew larger, became thicker, and contained more liquid water during the day than those in the 1D simulation. And even more importantly, both the original TenStream model and the dynamic TenStream solver captured them, further underscoring the latter's ability to capture 3D radiative effects at a significantly lower computational cost.

Overall, these results clearly demonstrated the capabilities of the dynamic TenStream solver introduced in this thesis. Not only was it shown that the new method can compute heating rates and radiative fluxes that closely align with 3D benchmark results — while offering a notable improvement over 1D solvers operated with a similar computational demand — but also that, when coupled to model dynamics, it successfully reproduces all the cloud characteristics observed in simulations using full 3D solvers. These findings become even more interesting when considering recent advancements in the field of another major computational bottleneck in radiative transfer calculations, namely the number of spectral bands required for accurate integrated longwave and shortwave heating rates. For the speed evaluation in Sect. 4.2.1, for instance, the wavelength parameterization by Fu and Liou (1992, 1993) was used, which features a total of 54 and 67 spectral bands in the solar and thermal spectral ranges, respectively (Oreopoulos et al., 2012). That is already a relatively low number considering that most models currently use the RRTMg parameterization (Mlawer et al., 1997; AER, 2025), which incorporates a total of 112 and 140 spectral bands in the solar and thermal spectral ranges, respectively. Recent developments have shown, however, that these numbers can be dramatically reduced without a significant loss in precision in the calculation of both radiative fluxes and heating rates. de Mourgues et al. (2023), for example, demonstrated that in the thermal spectral range, even 30 spectral bands are sufficient to calculate heating rates very similar to those obtained by a line-by-line calculation. Compared to RRTMg, this is more than four times less spectral bands. Similarly, Hogan and Matricardi (2022) showed that just 32 spectral bands in both the solar and thermal spectral ranges yield very accurate irradiances and heating rates, with additional spectral bands providing little to no increase in precision. These more efficient spectral parameterizations, combined with the speed improvements achieved with the dynamic TenStream solver, could accelerate 3D radiative transfer toward the speed of currently employed 1D solvers, potentially enabling the use of 3D radiative transfer in NWP models for the first time ever.

That said, the potential speed-up from reducing the number of spectral bands depends strongly on how radiative transfer is implemented in these models. Since calculations for different spectral bands are independent of each other, they can be performed in parallel. In that case, reducing the number of bands does not necessarily decrease the total computational time of the radiation scheme. Instead, reductions in computational cost would need to come from other components of the solver — for example, by optimizing the retrieval of the coefficients for the TenStream linear equation system from the lookup tables, which currently accounts for about one third of the total runtime of dynamic TenStream calculations. But also beyond such optimizations, further development of the solver is required before the vision of performing 3D radiative transfer at the cost of current 1D solvers can become reality. First of all, performance tests should be extended to include multiple-layer cloud fields — e.g., shallow cumulus clouds with cirrus clouds above — as well as deep convective clouds in order to investigate whether two iterations toward convergence per call, as used in most parts of this thesis, are still sufficient under these circumstances, as more complex cloud fields also involve more radiative interactions in the vertical. Earlier simulations with the dynamic TenStream solver have shown that incomplete solves can lead to "ping-pong" effects in these cases, where distant grid boxes update radiative influences on each other back and forth in between different dynamic TenStream calls. While these ping-pong effects were vastly reduced by the use of the Gauß–Seidel method, it will be interesting to see whether vertically more complex cloud fields pose a greater challenge for the solver. In addition, the development of a rule for how many Gauß–Seidel iterations to use to ensure reliable results depending on the model setup is another main future target. In this context, it would also be worthwhile to explore whether occasional full solves could be a computationally feasible means of ensuring that the results of the dynamic TenStream solver remain close to those of the original TenStream solver. Additionally, the investigation of more sophisticated first guesses for the incomplete solves could further improve performance. One possibility would be to advect the radiative field from the previous radiation time step along with the other atmospheric fields. Since the radiative field is not expected to drastically change between two calls of the radiation model, such a first guess should already better account for the updated position of the clouds, enabling the incomplete solves to focus primarily on correcting for the changed optical properties of the clouds, which would likely speed up convergence. Finally, when shifting to the NWP scale, one will certainly need to consider sub-grid-scale cloud variability, for example by extending the TenStream lookup tables to also account for cloud fraction. This would finally enable the application of the dynamic TenStream solver at the scale it was originally intended for, paving the way toward answering the open question of how 3D radiative transfer affects weather forecasts.

Appendix A

Bash scripts for the libRadtran simulations

This appendix provides the bash scripts that were used to perform the libRadtran simulations discussed in Chapter 4 of this work. These scripts were necessary because libRadtran does not natively support a time-stepping scheme, which is, however, essential for the dynamic TenStream solver, as it relies on the radiative field from a previous radiation time step as the starting point for its incomplete solves. To nevertheless simulate such a time-stepping scheme, the bash script was used to sequentially call the dynamic TenStream solver for the different cloud fields in the time series employed in Chapter 4. Additionally, a history mode was implemented into the solver, which saves the irradiance field for every wavelength band at the end of each run. These saved irradiances can then be read by the dynamic TenStream solver during subsequent calls, allowing it to initialize its incomplete solves from the previously computed solution. This appendix now presents the script implementing this time-stepping scheme for the dynamic TenStream calculations, as well as the scripts used for the δ-Eddington, original TenStream, and MYSTIC simulations also presented in Chapter 4.

A.1 Bash script for the dynamic TenStream simulations

The following script was used to perform the dynamic TenStream simulations in the solar spectral range. It is intended to be placed in the folder `SPECIFIC_DATA_PATH/Bash_Scripts`, where `SPECIFIC_DATA_PATH` must point to the working directory containing the folders `Bash_Scripts`, `Cloudfiles`, and `Results`. Additionally, `LIBRADTRAN_PATH` must point to the location of the libRadtran installation used for the simulations.

```
TSTART=8000
TEND=9000

for DT_RAD in 10 30 60
do

for ts in $(seq $TSTART $DT_RAD $TEND)
do
```

```
if [ $ts -gt $TSTART ]; then
 it=2
else
 it=500
fi

cat > $SPECIFIC_DATA_PATH/Bash_Scripts/input_solar.inp << eof
# background atmosphere
atmosphere_file $SPECIFIC_DATA_PATH/afglus_modified_interpolated.dat
data_files_path $LIBRADTRAN_PATH/data
mc_basename     ../Results/Dynamic_TenStream/dts.dtrad_$DT_RAD.ts_$ts.it_$it
source solar

# position of the sun
sza 50.0
phi0 270.0

# surface albedo
albedo 0.125

# wavelengths and absorption parametrization
wavelength_index 1 6
mol_abs_param FU

# radiative transfer solver settings
rte_solver dynamic_tenstream
dynamic_tenstream_iterations $it
dynamic_tenstream_heat_unit K_per_day
dynamic_tenstream_history

# cloud
wc_file 3D $SPECIFIC_DATA_PATH/Cloudfiles/dat/cloudfile.t${ts}.dat

output_process sum
zout all
eof

$LIBRADTRAN_PATH/bin/uvspec < $SPECIFIC_DATA_PATH/Bash_Scripts/input_solar.inp
done

for band in $(seq 0 1 5)
do
rm $SPECIFIC_DATA_PATH/Bash_Scripts/E_diffuse_iv_${band}_*
rm $SPECIFIC_DATA_PATH/Bash_Scripts/E_direct_iv_${band}_*
```

```
done
done
```

As specified by the values of TSTART and TEND at the beginning, this script applies the dynamic TenStream solver to the cloud fields of the shallow cumulus cloud time series prepared by Jakub and Gregor (2022) during the time period from 8000 s to 9000 s into the simulation. Three different calling frequencies are considered: 10 s, 30 s, and 60 s, defined by the variable DT_RAD. For each of these calling frequencies, the solver is sequentially called at every timestep between TSTART and TEND, using increments of DT_RAD. During the first call (i.e., at `ts = TSTART`), where no previously calculated irradiance field is yet available, 500 Gauß–Seidel iterations (`dynamic_tenstream_iterations`) are performed to ensure that the radiative field is fully converged. For all subsequent timesteps, only a minimum of two iterations are executed, using the previously computed irradiance field as the starting point of the calculation. The `dynamic_tenstream_history` flag ensures that these irradiance fields are saved for each wavelength band at the end of every run and, if available, read at the beginning of the next run. Once the entire time series has been processed for a given calling frequency, the saved irradiance fields are deleted in the final loop of the script, so that the next set of simulations starts from scratch. Apart from these time-stepping mechanics, the generated input file includes all simulation settings described in Sect. 4.1.1. The background atmosphere is defined by `afglus_modified_interpolated.dat`, the solar zenith angle is fixed at 50°, and the Sun is assumed to be in the east (`phi0 = 270.0`). The surface albedo is set to 0.125, and the molecular absorption is parameterized using the scheme of Fu and Liou (1992, 1993) (`mol_abs_param FU`). Furthermore, the relevant cloud field for each timestep is loaded using the `wc_file 3D` option, referring to the cloud file `cloudfile.t${ts}.dat` required at the current simulation time.

The bash script for the dynamic TenStream simulations in the thermal spectral range differs from the solar version only in a few respects. First, `source solar` is replaced by `source thermal`. Moreover, the position of the Sun does not need to be specified in this spectral range, and the albedo must be set to the thermal value of 0.05, as specified in Sect. 4.1.1. The final difference is that Fu bands 7 to 18 are used instead of 1 to 6. Accordingly, the loop that deletes the irradiance fields at the end of the script must be modified to span `$(seq 6 1 17)` instead of `$(seq 0 1 5)`. Everything else remains unchanged.

A.2 Input files for the other radiative transfer solvers

The other three radiative transfer solvers considered in the evaluation in Chapter 4 — namely, the 1D δ-Eddington approximation, the original TenStream solver, and the Monte Carlo model MYSTIC — all calculate the radiative field from scratch each time they are called. Hence, their results depend only on the current time step and the corresponding cloud field, but not on the calling frequency of the solver. As a result, the bash script does not require the first loop over DT_RAD, nor does it explicitly need the loop over the timesteps (`ts`), as the solutions for different timesteps are independent of one another and can be calculated in parallel. The main difference between these solvers therefore lies in the input file, particularly in the section specifying the radiative transfer solver settings. For the 1D δ-Eddington approximation, this section must be modified as follows:

```
# radiative transfer solver settings
ipa_3d
rte_solver twostrebe
heating_rate layer_fd
mc_forward_output heating K_per_day
```

Here, `ipa_3d` specifies that the originally three-dimensional atmosphere should be treated as a set of independent vertical columns, whereas `rte_solver twostrebe` selects the δ-Eddington approximation used in the evaluation. The option `mc_forward_output heating K_per_day` indicates that, in addition to irradiances, also heating rates should be returned, with units of $\mathrm{K\,d^{-1}}$. `heating_rate layer_fd` further specifies that these heating rates should be calculated using forward differences of the irradiance across each layer, which is the default setting. In contrast, the original TenStream solver requires the following configuration:

```
# radiative transfer solver settings
rte_solver mystic
mc_tenstream solver_3_10
mc_sample_grid 64 64
mc_forward_output heating K_per_day
```

The option `rte_solver mystic` appears here because the TenStream solver is implemented within the MYSTIC framework in libRadtran. In addition to that, the number of grid boxes in both horizontal dimensions must be specified for this solver. In the setup described in Sect. 4.1.1, a grid of 64×64 boxes is used, so that `mc_sample_grid` is set to `64 64`. Another particularity of this solver is that it must be executed using `uvspec_mpi` instead of `uvspec`. The corresponding execution command therefore reads:

```
$LIBRADTRAN_PATH/bin/uvspec_mpi $SPECIFIC_DATA_PATH/Bash_Scripts/input_solar.inp
```

Finally, for the MYSTIC solver, the relevant section of the input file must be modified as follows:

```
# radiative transfer solver settings
rte_solver mystic
mc_photons 4000000
mc_sample_grid 64 64
mc_forward_output heating K_per_day
```

Here, `mc_photons 4000000` specifies the number of photons used in each individual simulation. While actually a total of 400 000 000 photons was used per time step, this number was split across 100 independent simulations with 4 000 000 photons each for computational efficiency. This is why the value 4 000 000 appears in the excerpt shown above.

Appendix B

Input files for the PALM simulations

This appendix provides the parameter namelist file that was used for all the PALM simulations discussed in Chapter 5, as well as the TenStream options files that distinguish the various restart runs introduced in Sect. 5.1.1.

B.1 Parameter file

The following parameter file (p3d file) was used for all the PALM simulations discussed in Chapter 5, i.e., for the initial run and the restart runs of both the main and control run, except for the following modifications:

- For the restart runs performed with the different radiative transfer solvers, the values of both "dt_restart" and "restart_time" were set to 144000.0, i.e., the total simulation time of the model. By doing so, these restart runs were carried out in one go.
- For the control runs, the option "ensemble_member_nr = 1" was enabled in order to create a model run that is statistically independent from the main run.

Since all the simulations that that were run relied on the restart mechanism in PALM, an additional parameter file for the restart runs (p3dr file) had to be provided. For our simulations, this restart parameter file was identical to the one below, except that "initializing_actions" had to be set to 'read_restart_data'.

```
!--------------------------------------------------------------------------------
!-- INITIALIZATION PARAMETER NAMELIST
!   Documentation: https://palm.muk.uni-hannover.de/trac/wiki/doc/app/inipar
!--------------------------------------------------------------------------------

&initialization_parameters
!
!-- grid parameters
!--------------------------------------------------------------------------------
```

```
nx                          = 255,   ! number of gridboxes in x-direction (nx+1)
ny                          = 255,   ! number of gridboxes in y-direction (ny+1)
nz                          =  80,   ! number of gridboxes in z-direction (nz+1)

dx                          = 100.0, ! horizontal grid spacing in x-direction in m
dy                          = 100.0, ! horizontal grid spacing in y-direction in m
dz                          =  50.0, ! vertical grid spacing in m

!
!-- initialization
!---------------------------------------------------------------------------------
initializing_actions        = 'set_constant_profiles',

ug_surface                  = 0.0,   ! u-comp of geostrophic wind at surface
vg_surface                  = 0.0,   ! v-comp of geostrophic wind at surface

uv_heights                  = 0.0, 10.0, 20.0, 40.0, 60.0, 80.0, 100.0,
u_profile                   = 0.0,  5.0,  5.7,  6.5,  7.0,  7.3,   7.5,
v_profile                   = 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,   0.0,

pt_surface                  = 288.0, ! initial surface potential temp
pt_vertical_gradient        =   0.0,    1.0,    0.0,     1.0,
pt_vertical_gradient_level =   0.0,  800.0,  850.0,  1300.0,

humidity                    = .T.,
q_surface                   = 0.007,
q_vertical_gradient         = 0.0,  -0.002,     0.0,
q_vertical_gradient_level  = 0.0,   800.0,  1000.0,

longitude                   = 11.6,
latitude                    = 48.1,

origin_date_time            = '2023-06-15 00:00:00 +02',

!
!-- boundary conditions
!---------------------------------------------------------------------------------
bc_pt_b                     = 'dirichlet',
bc_q_b                      = 'dirichlet',

!
!-- numerics
!---------------------------------------------------------------------------------
fft_method                  = 'temperton-algorithm',  ! build-in fft method
!ensemble_member_nr         = 1,                      ! produce statistically
```

```
                                                 ! independent simulation
!
!-- physics
!-------------------------------------------------------------------------------
omega                      = 0.0,                ! no Coriolis force
/ ! end of initialization parameter namelist


!-------------------------------------------------------------------------------
!-- RUNTIME PARAMETER NAMELIST
!   Documentation: https://palm.muk.uni-hannover.de/trac/wiki/doc/app/d3par
!-------------------------------------------------------------------------------

&runtime_parameters
!
!-- run steering
!-------------------------------------------------------------------------------
end_time                   = 104400.0, ! simulation time of the 3D model
dt_restart                 =   1800.0,
restart_time               =   1800.0,

dt                         =      5.0,

create_disturbances        =   .TRUE., ! randomly perturbate horiz. velocity
dt_disturb                 =    150.0, ! interval for random perturbations
disturbance_energy_limit   =     0.01, ! upper limit for perturbation energy

data_output_2d_on_each_pe  =  .FALSE., ! don't do 2D output on each MPI rank

npex                       =        8, ! number of processors in x direction
npey                       =        8, ! number of processors in y direction

!
!-- Run-control/timeseries output settings
!------------------------------------------------------------------------------
dt_run_control             =      0.0, ! run control output after each timestep

!
!-- data output
!------------------------------------------------------------------------------
netcdf_data_format         =    5,     ! use netCDF4 (HDF5) format
                                       ! with parallel I/O support
dt_data_output             = 60.0,     ! output interval for general data
dt_data_output_av          = 60.0,     ! output interval for averaged data
dt_dopr                    = 60.0,     ! output interval for profile data
```

```
data_output                = 'u', 'v', 'w',
                             'p', 'q', 'ql', 'qc', 'qr', 'qv', 'nc', 'lwp*_xy',
                             'prr', 'prr_rain', 'pra*_xy',
                             'ta', 'theta', 'ta_2m*_xy', 'tsurf*_xy',
                             'm_soil', 't_soil',
                             'shf*_xy', 'qsws*_xy',
                             'rad_lw_in', 'rad_lw_out',
                             'rad_sw_in', 'rad_sw_out',
                             'rad_lw_hr', 'rad_sw_hr',
                             'rad_net*_xy',
                             'rad_lw_in*_xy', 'rad_lw_out*_xy',
                             'rad_sw_in*_xy', 'rad_sw_out*_xy',

data_output_pr             = 'u', 'v', 'w', 'p', 'rho', 'hyp',
                             'q', 'ql', 'qc', 'qr', 'qv', 'rh',
                             'prr', 'prr_cloud',
                             'rad_sw_in', 'rad_sw_out',
                             'rad_lw_in', 'rad_lw_out',

section_xy                 = 0, 20,     ! grid index for 2D XY cross sections
/ ! end of runtime parameter namelist


!--------------------------------------------------------------------------------
!   cloud model
!--------------------------------------------------------------------------------
&bulk_cloud_parameters
cloud_scheme               = 'morrison',
cloud_water_sedimentation = .T.,
/ ! end of bulk cloud parameter namelist

!--------------------------------------------------------------------------------
!   radiation model
!--------------------------------------------------------------------------------
&radiation_parameters
radiation_scheme          = 'tenstream',
albedo_type               =  5,         ! short grassland/meadow/shrubland
constant_albedo           = .T.,
use_broadband_albedo      = .T.,
dt_radiation              = 30.0,
radiation_interactions_on = .F.,
/

!--------------------------------------------------------------------------------
```

```
!   land-surface model
!-------------------------------------------------------------------------------
&land_surface_parameters
surface_type             = 'vegetation',
vegetation_type          =  3,          ! short grass
soil_type                =  3,          ! medium-fine
conserve_water_content   = .T.,
dz_soil                  =  0.01,  0.02,  0.04,  0.07,  0.15,  0.21,  0.72,  1.89,
root_fraction            =   0.1,   0.2,   0.3,   0.2,   0.1,  0.05,  0.05,   0.0,
soil_temperature         = 288.0, 286.0, 285.0, 284.0, 283.0, 283.0, 282.0, 282.0,
deep_soil_temperature    = 280.0,
soil_moisture            =  0.18,  0.18,  0.18,  0.18,  0.18,  0.18,  0.18,  0.18,
/
```

B.2 TenStream options files

As explained in Sect. 5.1.1, the three restart runs performed from each of the two initial PALM simulations differ only in the radiative transfer solver used. Therefore, these simulations were performed with different `tenstream.options` files, which are provided below.

File for the simulations with the 1D δ-Eddington approximation

```
-solver 2str
```

File for the simulations with the dynamic TenStream solver

```
-solver 3_10

-solar_dir_explicit
-solar_dir_ksp_max_it 1

-solar_diff_explicit
-solar_diff_ksp_max_it 1
-solar_diff_pc_sub_it 2

-thermal_diff_explicit
-thermal_diff_ksp_max_it 1
-thermal_diff_pc_sub_it 2

-accept_incomplete_solve
```

File for the simulations with the original TenStream solver

```
-solver 3_10
```

```
-solar_dir_explicit
-solar_diff_explicit
-thermal_diff_explicit
```

Appendix C

Comparison of the two implementations of the dynamic TenStream solver

While the analysis in Chapter 4 is based on the libRadtran implementation of the dynamic TenStream solver, the large-eddy simulations in Chapter 5 rely on a different implementation that is part of the TenStream framework. These two implementations differ in several aspects, with the most notable being that the TenStream version is fully parallelized. Moreover, the TenStream implementation requires the use of the parallel linear algebra library PETSc (Balay et al., 2023), which the libRadtran version was specifically designed to avoid. Apart from that, there are also some differences in the spin-up method and in the execution of the Gauß–Seidel iterations between the two implementations.

To highlight these differences, but to also show that the TenStream implementation behaves almost identically to the libRadtran version when used with certain options, this appendix compares the two realizations in the libRadtran library, where both of them are implemented. Since both implementations are based on the TenStream solver, they certainly produce the same result in the limit of a large number of Gauß–Seidel iterations. They may however differ in their convergence behavior leading to this result, depending on their implementation. Hence, the goal of this appendix is to compare this convergence behavior for the libRadtran and TenStream implementations of the dynamic TenStream solver. To this end, calculations are performed that use the first time step of the shallow cumulus cloud field introduced in Chapter 4 alongside the settings described in Sect. 4.1.1. Based on this cloud field, the idea is to calculate the root-mean square error (RMSE) in heating rates of incomplete solves performed with N Gauß–Seidel iterations with respect to the fully converged TenStream solution, with

$$\mathrm{RMSE} = \sqrt{\left\langle (\xi - \xi_{\mathrm{ref}})^2 \right\rangle} \qquad \text{and} \qquad \xi = \left(\frac{\Delta T}{\Delta t}\right)_{i,j,k}, \tag{C.1}$$

where $\langle \ldots \rangle$ denotes a spatial average, consistent with the definitions in Chapter 4. If this RMSE as a function of the number N of Gauß–Seidel iterations is the same for both implementations, they can be considered equivalent. To this end, the TenStream version of the dynamic TenStream solver was operated using the following `tenstream.options` file:

```
-solar_dir_explicit
-solar_dir_ksp_max_it N/2
-solar_dir_ksp_complete_initial_run 0

-solar_diff_explicit
-solar_diff_ksp_max_it N
-solar_diff_ksp_complete_initial_run 0

-thermal_diff_explicit
-thermal_diff_ksp_max_it N
-thermal_diff_ksp_complete_initial_run 0

-accept_incomplete_solve
-initial_guess_from_last_uid 0
```

In there, N has to be replaced with the number of Gauß–Seidel iterations that are to be performed. In addition to that, note the following about this file and the differences between the two dynamic TenStream implementations:

1. The `-solar_dir_explicit`, `-solar_diff_explicit` and `-thermal_diff_explicit` options ensure that the Gauß–Seidel method is used to solve the TenStream linear equation system. However, unlike the libRadtran implementation, the TenStream implementation separates the calculation of direct and diffuse radiation in the solar spectral range. Direct radiation — that is the three lowermost equations in Eq. (3.1) — is always calculated first before continuing with the calculation of diffuse radiation. In addition to that, for direct radiation, the iteration direction described in Sect. 3.2.2 is not altered in every Gauß–Seidel iteration, as direct radiation always propagates from top to bottom in the vertical and in the direction of solar incidence in the horizontal. This separate calculation of direct and diffuse radiation makes perfect sense for the original TenStream solver, as the propagation of direct radiation is independent from the diffuse part and can thus easily be treated separately with benefits for the calculation of both direct and diffuse radiative fluxes: While the calculation of the direct fluxes can always follow the direction of solar incidence that way, the calculation of the diffuse fluxes is provided with up-to-date direct fluxes right from the beginning. The reason why this separation was not implemented in the libRadtran version of the dynamic TenStream solver is that we wanted to minimize the number of iterations through grid boxes by just iterating through them once in every single Gauß–Seidel iteration.

   To make the two implementations nevertheless comparable, only $N/2$ Gauß–Seidel iterations are performed for the calculation of direct solar radiation in the TenStream implementation (`-solar_dir_ksp_max_it N/2`), compared to N for diffuse radiation (`-solar_diff_ ksp_max_it N` and `-thermal_diff_ksp_max_it N`), with N being an even integer number, as it was motivated in Sect. 4.1.2. This way, the TenStream implementation always performs the same number of Gauß–Seidel iterations for diffuse radiative fluxes as the libRadtran implementation, without outperforming it in terms of direct radiation, since every second iteration in the libRadtran implementation is performed alongside the diffuse fluxes from the bottom to the top of the domain and opposite to the angle of solar incidence, thus providing hardly any updates for the direct radiative fluxes — except for grid boxes at the domain boundaries, where boundary conditions were previously applied. Note, however, that even this way, the two implementations will always slightly differ for even numbers $N \geq 4$ of Gauß–Seidel iterations. This is because while the two implementations may perform the same $N/2$ effective iterations for direct radiation and the same N iterations for diffuse radiation, in the TenStream implementation, all Gauß–Seidel iterations for direct radiation will still be performed prior to any Gauß–Seidel iteration involving diffuse radiation, thus feeding the calculation of diffuse radiation with already more precise source terms. We will however soon see that this effect is not very large, as direct radiation converges relatively quickly.

2. Unlike the libRadtran implementation, the TenStream implementation iterates through the domain from top to bottom first in both the solar and thermal spectral ranges.

3. By default, the TenStream implementation effectively performs a full solve of the TenStream linear equation system the first time it is called. Since for this comparison, we want to get an idea of the convergence behavior of the solver as a function of the number of Gauß–Seidel iterations used, we do not want it to perform such a full solve that ignores the prescribed number of iterations. Hence, the options `..._ksp_complete_initial_run 0` disable this feature for the various parts of the solver. However, even with these full solves disabled, the TenStream implementation still performs a spin-up, just like the libRadtran implementation performs the single-column clear-sky spin-up introduced in Sect. 3.2.2. Unlike this spin-up, however, the TenStream implementation uses the radiative fluxes that it obtains for a spectral band as starting values for the calculation of the next spectral band, even when the option `-initial_guess_from_last_uid 0` is used. While this spin-up certainly does not work as well as the single-column clear-sky spin-up in the libRadtran implementation, it does not require any additional computations and is certainly more efficient than starting the calculations of each spectral band from zero.

Besides the TenStream version, which was run with the settings listed above, the libRadtran implementation of the dynamic TenStream solver was operated with two different setups for this comparison. On the one hand, in its native setting, i.e., with the clear-sky spin-up introduced in Sect. 3.2.2, followed by N Gauß–Seidel iterations using the iteration mechanism outlined in the very same section. On the other hand, a modified version of the libRadtran implementation was developed that also iterates through the domain from top to bottom first in both spectral ranges and uses the spectral band spin-up of the TenStream version.

Figure C.1 finally shows the convergence behavior in terms of the RMSE in heating rates for all of these implementations as a function of the number N of Gauß–Seidel iterations. Similar to the evaluation in Sect. 4.2.2, only heating rates in the LES part of the domain are used for the plot. As expected, we can see that all three implementations converge toward the full

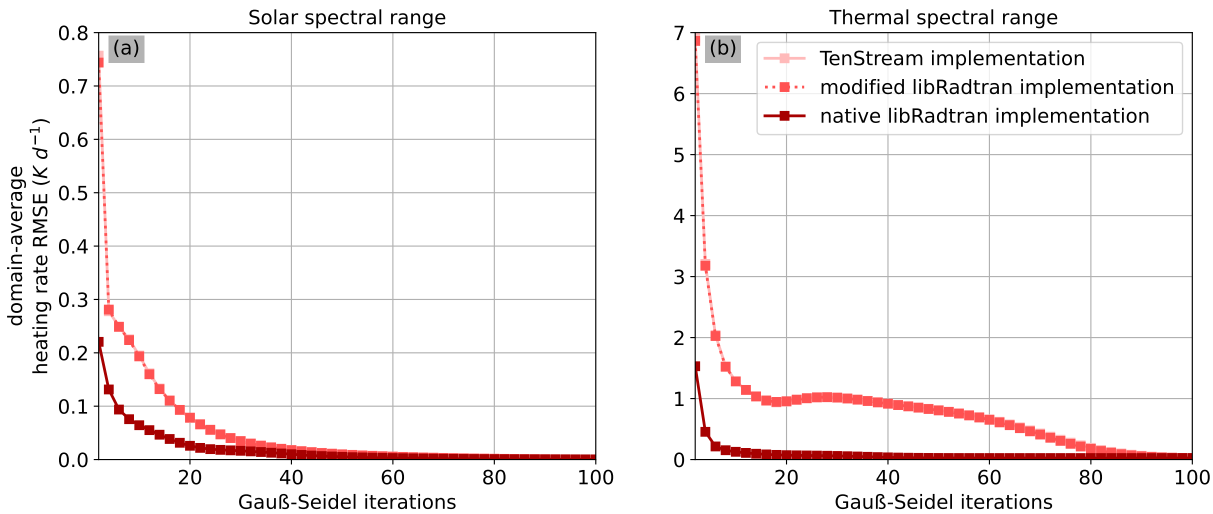

Figure C.1: RMSE in heating rates for both the native libRadtran (dark red lines) and TenStream (light red lines) implementations of the dynamic TenStream solver for a calculation from scratch compared to the fully converged TenStream solution as a function of the number N of Gauß–Seidel iterations. In addition to that, the dotted medium red lines show the convergence behavior of a version of the libRadtran implementation using the same spin-up method and iteration scheme as the TenStream implementation. Results are shown for both the solar (**a**) and thermal (**b**) spectral ranges. Following the motivation in Sect. 4.1.2, only integer multiples of two Gauß–Seidel iterations are used.

TenStream solution in the limit of a large number of iterations. Furthermore, we can clearly see that the native libRadtran implementation that uses the clear-sky spin-up (dark red lines) converges much faster than the TenStream implementation with the spectral band spin-up (light red lines). And most importantly, the libRadtran implementation almost matches the TenStream version when the same iteration mechanism and spin-up method are used (the dotted medium red lines and the light red lines are basically on top of each other). We can also see that despite the separate calculation of direct and diffuse radiation in the TenStream implementation, which should improve the source term in the calculations of diffuse radiative fluxes for $N \geq 4$, the resulting RMSE values in the solar spectral range are not much different from those of the modified libRadtran implementation.

When using half as many Gauß-Seidel iterations for the calculations of direct radiative fluxes, it can therefore be safely assumed that the TenStream implementation works exactly as the libRadtran implementation used in Chapter 4, especially since the spin-up is not relevant for the PALM simulations. Similar to the setup in Chapter 4, the idea for these simulations is to start with a fully converged solution at the very first call of the 3D solver, which is then followed by incomplete solves using just two Gauß–Seidel iterations every time the radiation module is called thereafter. These follow-up incomplete solves use the results obtained at the corresponding previous radiation time step as a first guess anyway, so that the initial spin-up method is not important at all. And apart from this spin-up (and the iteration direction in the thermal spectral range), it has just been demonstrated that both implementations essentially lead to the same results when used with the aforementioned options.

Appendix D

Temporal evolution of the initial vertical profiles in the PALM simulations

This appendix presents a figure showing the temporal evolution of the vertical profiles from which the PALM simulations discussed in Chapter 5 were initialized. Specifically, it depicts the temporal evolution of the profiles illustrated in Fig. 5.1 for the three simulations driven by the 1D δ-Eddington approximation, the dynamic TenStream solver, and the original TenStream model, all initialized from the main run of the setup described in Sect. 5.1.1. The initial profiles for potential temperature (θ), total water mixing ratio (q), zonal (u) and meridional (v) wind speed, and liquid water mixing ratio (q_l) in each of these simulations are shown in gray and correspond to the atmospheric state from which the main run of the setup was initially started on 14 June 2023 at 22:00 UTC. After the three restart simulations were launched from this main run on 15 June 2023 at 09:00 UTC, the subsequent evolution of the profiles is shown at two-hour intervals, beginning at 09:01 UTC and continuing until the end of the simulations on 16 June 2023 at 03:00 UTC.

As seen in the first and second columns of Fig. D.1, the profiles are initially well-mixed with respect to both θ and q up to a height of 800 m. As surface temperatures rise, all simulations overcome the thin, stably stratified layer initially specified between 800 m and 850 m height in the θ profile and gradually deepen their convective boundary layers throughout the day, ultimately reaching heights slightly above 2 km — a development consistent with that of the cloud top heights marked by the upper boundaries of the q_l profiles in panels (e), (j), and (o). The deepening of the convective boundary layer is also evident in the total water mixing ratio profiles shown in panels (b), (g), and (l), which initially feature a sharp drop-off between 800 m and 1000 m height. As the day progresses, this sharp gradient ascends markedly, consistently aligning with the upper boundary of the convective layer in the θ profile. By the end of the day, θ reaches values of approximately 292.5 K in the mixed layer, whereas q slowly decreases from about 5.7 $\mathrm{g\,kg^{-1}}$ near the surface to around 5.0 $\mathrm{g\,kg^{-1}}$ near the top of that layer. After sunset on 15 June 2023 at 19:15 UTC, a strongly stably stratified layer then forms near the surface in all simulations, with near-surface θ decreasing to 288 K, similar to the value in the initial state. Above this stably stratified layer, the atmosphere remains well-mixed in a layer that continues to deepen slightly overnight — again in good agreement with the evolution of cloud top heights (see Fig. 5.5). In addition to θ and q, the third and fourth columns of Fig. D.1 show that all simulations maintain a purely westerly flow throughout the simulation period, with wind speeds in the boundary layer gradually decreasing over time due to surface friction.

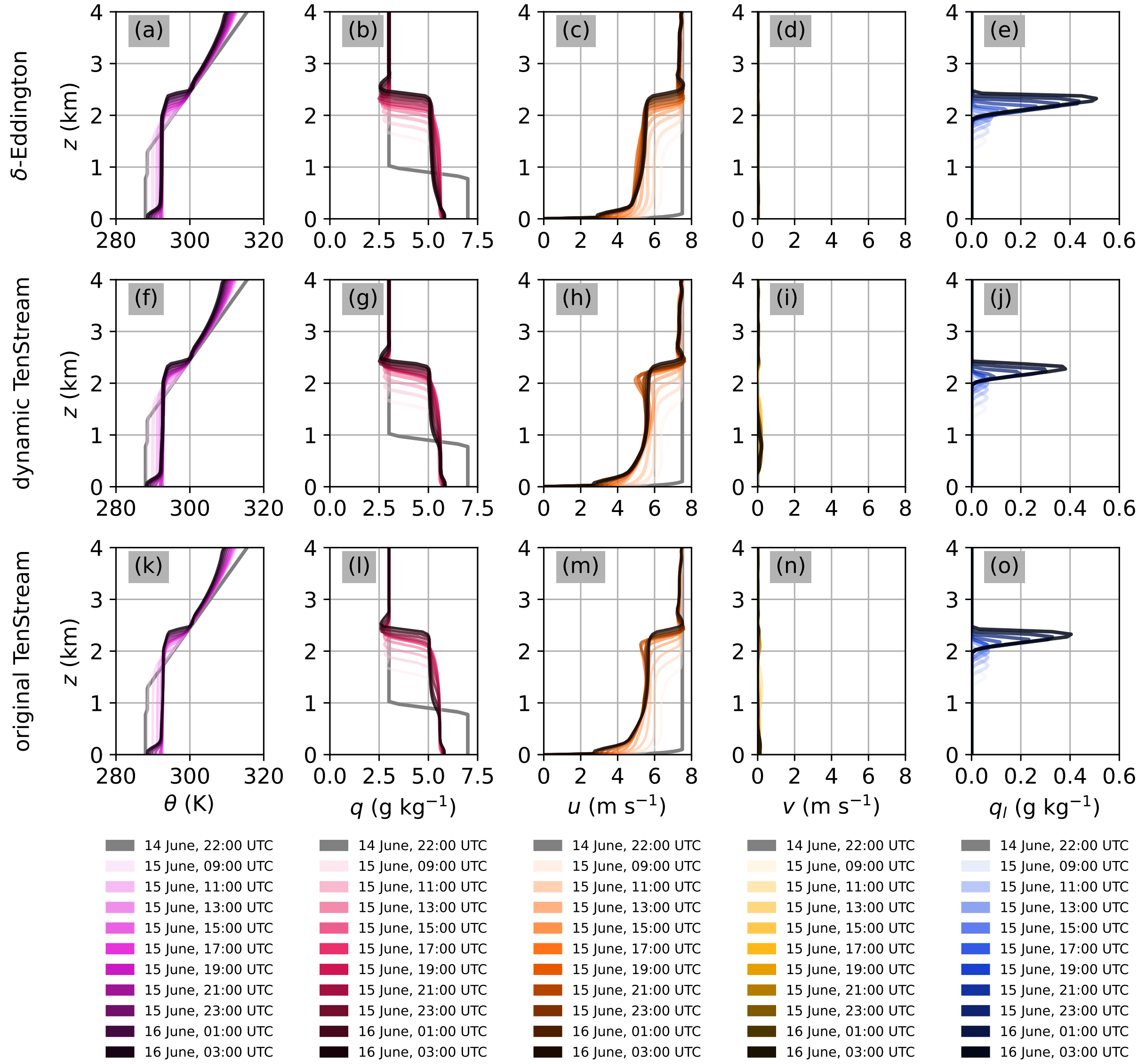

Figure D.1: Temporal evolution of the vertical profiles for potential temperature (first column), total water mixing ratio (second column), zonal (third column) and meridional (fourth column) wind speed, and liquid water mixing ratio (fifth column) in the PALM simulations driven by the δ-Eddington (top row), dynamic TenStream (middle row), and original TenStream (bottom row) solvers. The gray profile in each panel shows the initial state as illustrated in Fig. 5.1, whereas the colored profiles depict the temporal evolution of the quantities within the respective simulation in steps of two hours. The timestamps corresponding to each color are indicated in the legends below each column. Only simulations initialized from the main run of the setup are shown.

Bibliography

AER: RRTM and RRTMG, URL: http://rtweb.aer.com/rrtm_frame.html, last access: 3 May 2025, 2025.

Anderson, G., Clough, S., Kneizys, F., Chetwynd, J., and Shettle, E.: AFGL atmospheric constituent profiles (0-120 km), Tech. Rep. AFGL-TR-86-0110, Air Force Geophys. Lab., Hanscom Air Force Base, Bedford, Mass., URL: https://apps.dtic.mil/sti/pdfs/ADA175173.pdf, last access: 3 May 2025, 1986.

Andrews, D. G.: Introduction, in: An Introduction to Atmospheric Physics, chap. 1, pp. 1–18, Cambridge University Press, second edn., https://doi.org/10.1017/CBO9780511800788, 2010.

Arakawa, A.: The Cumulus Parameterization Problem: Past, Present, and Future, Journal of Climate, 17, 2493–2525, https://doi.org/10.1175/1520-0442(2004)017<2493:ratcpp>2.0.co;2, 2004.

Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J.: PETSc/TAO Users Manual, Tech. Rep. ANL-21/39 - Revision 3.19, Argonne National Laboratory, https://doi.org/10.2172/1968587, 2023.

Bauer, P., Thorpe, A., and Brunet, G.: The quiet revolution of numerical weather prediction, Nature, 525, 47–55, https://doi.org/10.1038/nature14956, 2015.

Bjerknes, V.: Das Problem der Wettervorhersage, betrachtet vom Standpunkte der Mechanik und der Physik, Meteorologische Zeitschrift, 21, 1–7, 1904.

Bjerknes, V.: The problem of weather prediction, considered from the viewpoints of mechanics and physics, Meteorologische Zeitschrift, 18, 663–667, https://doi.org/10.1127/0941-2948/2009/416, 2009.

Bowen, I. S.: The Ratio of Heat Losses by Conduction and by Evaporation from any Water Surface, Physical Review, 27, 779–787, https://doi.org/10.1103/physrev.27.779, 1926.

Coiffier, J.: Weather prediction equations, in: Fundamentals of Numerical Weather Prediction, chap. 2, pp. 15–38, Cambridge University Press, https://doi.org/10.1017/cbo9780511734458.006, 2011.

Courant, R., Friedrichs, K., and Lewy, H.: Über die partiellen Differenzengleichungen der mathematischen Physik, Mathematische Annalen, 100, 32–74, https://doi.org/10.1007/bf01448839, 1928.

Cullen, M. J. P.: The unified forecast/climate model, Meteorological Magazine, 122, 81–94, URL: https://digital.nmla.metoffice.gov.uk/IO_faa7dfb4-d31a-4a1b-92ba-16a145be1509, last access: 3 May 2025, 1993.

Davis, A., Gabriel, P., Lovejoy, S., Schertzer, D., and Austin, G. L.: Discrete angle radiative transfer: 3. Numerical results and meteorological applications, Journal of Geophysical Research: Atmospheres, 95, 11 729–11 742, https://doi.org/10.1029/jd095id08p11729, 1990.

de Elía, R., Laprise, R., and Denis, B.: Forecasting Skill Limits of Nested, Limited-Area Models: A Perfect-Model Approach, Monthly Weather Review, 130, 2006–2023, https://doi.org/10.1175/1520-0493(2002)130<2006:FSLONL>2.0.CO;2, 2002.

de Mourgues, M., Emde, C., and Mayer, B.: Optimized Wavelength Sampling for Thermal Radiative Transfer in Numerical Weather Prediction Models, Atmosphere, 14, 332, https://doi.org/10.3390/atmos14020332, 2023.

Demtröder, W.: Gase, in: Experimentalphysik 1, chap. 7, pp. 185–210, Springer Berlin Heidelberg, https://doi.org/10.1007/978-3-662-46415-1_7, 2015.

DWD: Operational NWP System: Model configuration upgrade of ICON, URL: https://www.dwd.de/DE/fachnutzer/forschung_lehre/numerische_wettervorhersage/nwv_aenderungen/_functions/DownloadBox_modellaenderungen/icon/pdf_2021/pdf_icon_14_04_2021.pdf?__blob=publicationFile&v=3, last access: 3 May 2025, 2021.

DWD: Regionalmodell ICON-D2, URL: https://www.dwd.de/DE/forschung/wettervorhersage/num_modellierung/01_num_vorhersagemodelle/regionalmodell_icon_d2.html;jsessionid=1BCA6E1BC5087CF224B9C063693ECF9E.live31094?nn=512942, last access: 3 May 2025, 2025.

Einstein, A.: Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt, Annalen der Physik, 322, 132–148, https://doi.org/10.1002/andp.19053220607, 1905.

Emde, C., Buras-Schnell, R., Kylling, A., Mayer, B., Gasteiger, J., Hamann, U., Kylling, J., Richter, B., Pause, C., Dowling, T., and Bugliaro, L.: The libRadtran software package for radiative transfer calculations (version 2.0.1), Geoscientific Model Development, 9, 1647–1672, https://doi.org/10.5194/gmd-9-1647-2016, 2016.

Feller, W.: Random Variables; Expectation, in: An Introduction to Probability Theory and Its Applications, vol. 1, chap. 9, pp. 164–190, Wiley, 1950.

Forster, P., Storelvmo, T., Armour, K., Collins, W., Dufresne, J.-L., Frame, D., Lunt, D. J., Mauritsen, T., Palmer, M. D., Watanabe, M., Wild, M., and Zhang, H.: The Earth's Energy Budget, Climate Feedbacks and Climate Sensitivity, in: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change, edited by Masson-Delmotte, V., Zhai, P., Pirani, A., Connors, S., Péan, C., Berger, S., Caud, N., Chen, Y., Goldfarb, L., Gomis, M., Huang, M., Leitzell, K., Lonnoy, E., Matthews, J., Maycock, T., Waterfield, T., Yelekçi, O., Yu, R., and Zhou, B., chap. 7, p. 923–1054, Cambridge University Press, https://doi.org/10.1017/9781009157896.009, 2021.

Fu, Q. and Liou, K. N.: On the Correlated k-Distribution Method for Radiative Transfer in Nonhomogeneous Atmospheres, Journal of the Atmospheric Sciences, 49, 2139–2156, https://doi.org/10.1175/1520-0469(1992)049<2139:OTCDMF>2.0.CO;2, 1992.

Fu, Q. and Liou, K. N.: Parameterization of the Radiative Properties of Cirrus Clouds, Journal of the Atmospheric Sciences, 50, 2008–2025, https://doi.org/10.1175/1520-0469(1993)050%3C2008:POTRPO%3E2.0.CO;2, 1993.

Gabriel, P., Lovejoy, S., Davis, A., Schertzer, D., and Austin, G. L.: Discrete angle radiative transfer: 2. Renormalization approach for homogeneous and fractal clouds, Journal of Geophysical Research: Atmospheres, 95, 11 717–11 728, https://doi.org/10.1029/jd095id08p11717, 1990.

Geary, R. C.: The Ratio of the Mean Deviation to the Standard Deviation as a Test of Normality, Biometrika, 27, 310–332, https://doi.org/10.2307/2332693, 1935.

Gregor, P., Zinner, T., Jakub, F., and Mayer, B.: Validation of a camera-based intra-hour irradiance nowcasting model using synthetic cloud data, Atmospheric Measurement Techniques, 16, 3257–3271, https://doi.org/10.5194/amt-16-3257-2023, 2023.

Gristey, J. J., Feingold, G., Glenn, I. B., Schmidt, K. S., and Chen, H.: Surface Solar Irradiance in Continental Shallow Cumulus Fields: Observations and Large-Eddy Simulation, Journal of the Atmospheric Sciences, 77, 1065–1080, https://doi.org/10.1175/jas-d-19-0261.1, 2020.

Hogan, R. J. and Bozzo, A.: A Flexible and Efficient Radiation Scheme for the ECMWF Model, Journal of Advances in Modeling Earth Systems, 10, 1990–2008, https://doi.org/10.1029/2018ms001364, 2018.

Hogan, R. J. and Matricardi, M.: A Tool for Generating Fast k-Distribution Gas-Optics Models for Weather and Climate Applications, Journal of Advances in Modeling Earth Systems, 14, e2022MS003 033, https://doi.org/10.1029/2022ms003033, 2022.

Hogan, R. J. and Shonk, J. K. P.: Incorporating the Effects of 3D Radiative Transfer in the Presence of Clouds into Two-Stream Multilayer Radiation Schemes, Journal of the Atmospheric Sciences, 70, 708–724, https://doi.org/10.1175/jas-d-12-041.1, 2013.

Hogan, R. J., Schäfer, S. A. K., Klinger, C., Chiu, J. C., and Mayer, B.: Representing 3-D cloud radiation effects in two-stream schemes: 2. Matrix formulation and broadband evaluation, Journal of Geophysical Research: Atmospheres, 121, 8583–8599, https://doi.org/10.1002/2016jd024875, 2016.

Hogan, R. J., Fielding, M. D., Barker, H. W., Villefranque, N., and Schäfer, S. A. K.: Entrapment: An Important Mechanism to Explain the Shortwave 3D Radiative Effect of Clouds, Journal of the Atmospheric Sciences, 76, 2123–2141, https://doi.org/10.1175/JAS-D-18-0366.1, 2019.

Holton, J. R. and Hakim, G. J.: Numerical Approximation of the Equations of Motion, in: An Introduction to Dynamic Meteorology, chap. 13.2, pp. 455–464, Elsevier, fifth edn., https://doi.org/10.1016/B978-0-12-384866-6.00013-1, 2013.

Horn, G. L., Ouwersloot, H. G., Vilà-Guerau de Arellano, J., and Sikma, M.: Cloud Shading Effects on Characteristic Boundary-Layer Length Scales, Boundary-Layer Meteorology, 157, 237–263, https://doi.org/10.1007/s10546-015-0054-4, 2015.

Inness, P. and Dorling, S.: NWP Models – the Basic Principles, in: Operational Weather Forecasting, chap. 4, pp. 53–107, Wiley, https://doi.org/10.1002/9781118447659.ch4, 2012.

Jakub, F. and Gregor, P.: UCLA-LES shallow cumulus dataset with 3D cloud output data, https://doi.org/10.57970/5D0K9-Q2N86, LMU Munich, Faculty of Physics [data set], 2022.

Jakub, F. and Mayer, B.: A three-dimensional parallel radiative transfer model for atmospheric heating rates for use in cloud resolving models—The TenStream solver, Journal of Quantitative Spectroscopy and Radiative Transfer, 163, 63–71, https://doi.org/10.1016/j.jqsrt.2015.05.003, 2015.

Jakub, F. and Mayer, B.: 3-D radiative transfer in large-eddy simulations – experiences coupling the TenStream solver to the UCLA-LES, Geoscientific Model Development, 9, 1413–1422, https://doi.org/10.5194/gmd-9-1413-2016, 2016.

Jakub, F. and Mayer, B.: The role of 1-D and 3-D radiative heating in the organization of shallow cumulus convection and the formation of cloud streets, Atmospheric Chemistry and Physics, 17, 13 317–13 327, https://doi.org/10.5194/acp-17-13317-2017, 2017.

Joseph, J. H., Wiscombe, W. J., and Weinman, J. A.: The Delta-Eddington Approximation for Radiative Flux Transfer, Journal of the Atmospheric Sciences, 33, 2452–2459, https://doi.org/10.1175/1520-0469(1976)033<2452:tdeafr>2.0.co;2, 1976.

Khairoutdinov, M. and Kogan, Y.: A New Cloud Physics Parameterization in a Large-Eddy Simulation Model of Marine Stratocumulus, Monthly Weather Review, 128, 229–243, https://doi.org/10.1175/1520-0493(2000)128<0229:ancppi>2.0.co;2, 2000.

Khvorostyanov, V. I. and Curry, J. A.: Aerosol size spectra and CCN activity spectra: Reconciling the lognormal, algebraic, and power laws, Journal of Geophysical Research: Atmospheres, 111, D12 202, https://doi.org/10.1029/2005jd006532, 2006.

Klinger, C. and Mayer, B.: The Neighboring Column Approximation (NCA) – A fast approach for the calculation of 3D thermal heating rates in cloud resolving models, Journal of Quantitative Spectroscopy and Radiative Transfer, 168, 17–28, https://doi.org/10.1016/j.jqsrt.2015.08.020, 2016.

Klinger, C. and Mayer, B.: Neighboring Column Approximation—An Improved 3D Thermal Radiative Transfer Approximation for Non-Rectangular Grids, Journal of Advances in Modeling Earth Systems, 12, e2019MS001 843, https://doi.org/10.1029/2019ms001843, 2020.

Klinger, C., Mayer, B., Jakub, F., Zinner, T., Park, S.-B., and Gentine, P.: Effects of 3-D thermal radiation on the development of a shallow cumulus cloud field, Atmospheric Chemistry and Physics, 17, 5477–5500, https://doi.org/10.5194/acp-17-5477-2017, 2017.

Kotthaus, S. and Grimmond, C.: Energy exchange in a dense urban environment – Part I: Temporal variability of long-term observations in central London, Urban Climate, 10, 261–280, https://doi.org/10.1016/j.uclim.2013.10.002, 2014.

Lim, J. and Hoffmann, F.: Between Broadening and Narrowing: How Mixing Affects the Width of the Droplet Size Distribution, Journal of Geophysical Research: Atmospheres, 128, e2022JD037 900, https://doi.org/10.1029/2022jd037900, 2023.

Lorenz, E. N.: Deterministic Nonperiodic Flow, Journal of the Atmospheric Sciences, 20, 130–141, https://doi.org/10.1175/1520-0469(1963)020<0130:dnf>2.0.co;2, 1963.

Lovejoy, S., Davis, A., Gabriel, P., Schertzer, D., and Austin, G. L.: Discrete angle radiative transfer: 1. Scaling and similarity, universality and diffusion, Journal of Geophysical Research: Atmospheres, 95, 11 699–11 715, https://doi.org/10.1029/jd095id08p11699, 1990.

Maier, R., Jakub, F., Emde, C., Manev, M., Voigt, A., and Mayer, B.: A dynamic approach to three-dimensional radiative transfer in subkilometer-scale numerical weather prediction models: the dynamic TenStream solver v1.0, Geoscientific Model Development, 17, 3357–3383, https://doi.org/10.5194/gmd-17-3357-2024, 2024.

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, https://doi.org/10.5194/gmd-8-2515-2015, 2015.

Maronga, B., Banzhaf, S., Burmeister, C., Esch, T., Forkel, R., Fröhlich, D., Fuka, V., Gehrke, K. F., Geletič, J., Giersch, S., Gronemeier, T., Groß, G., Heldens, W., Hellsten, A., Hoffmann, F., Inagaki, A., Kadasch, E., Kanani-Sühring, F., Ketelsen, K., Khan, B. A., Knigge, C., Knoop, H., Krč, P., Kurppa, M., Maamari, H., Matzarakis, A., Mauder, M., Pallasch, M., Pavlik, D., Pfafferott, J., Resler, J., Rissmann, S., Russo, E., Salim, M., Schrempf, M., Schwenkel, J., Seckmeyer, G., Schubert, S., Sühring, M., von Tils, R., Vollmer, L., Ward, S., Witha, B., Wurps, H., Zeidler, J., and Raasch, S.: Overview of the PALM model system 6.0, Geoscientific Model Development, 13, 1335–1372, https://doi.org/10.5194/gmd-13-1335-2020, 2020.

Martin, J. E.: Mass, Momentum, and Energy: The Fundamental Quantities of the Physical World, in: Mid-Latitude Atmospheric Dynamics: A First Course, chap. 3, pp. 43–76, Wiley, ISBN: 978-0-470-86465-4, 2006.

Mason, P. J.: Large-eddy simulation: A critical review of the technique, Quarterly Journal of the Royal Meteorological Society, 120, 1–26, https://doi.org/10.1002/qj.49712051503, 1994.

Mayer, B.: Radiative transfer in the cloudy atmosphere, The European Physical Journal Conferences, 1, 75–99, https://doi.org/10.1140/epjconf/e2009-00912-1, 2009.

Mayer, B.: Erwärmungs- und Abkühlungsraten – Wie wichtig ist 3D-Strahlungstransport?, in: Strahlungsbilanzen, vol. 100 of *Promet – Meteorologische Fortbildung*, chap. 11, pp. 98–110, Deutscher Wetterdienst, URL: http://nbn-resolving.de/urn:nbn:de:101:1-2019090907541032149387, last access: 3 May 2025, ISBN: 978-3-88148-511-1, 2018.

Mayer, B. and Kylling, A.: Technical note: The libRadtran software package for radiative transfer calculations - description and examples of use, Atmospheric Chemistry and Physics, 5, 1855–1877, https://doi.org/10.5194/acp-5-1855-2005, 2005.

Mlawer, E. J., Taubman, S. J., Brown, P. D., Iacono, M. J., and Clough, S. A.: Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave, Journal of Geophysical Research: Atmospheres, 102, 16 663–16 682, https://doi.org/10.1029/97jd00237, 1997.

Moin, P. and Mahesh, K.: DIRECT NUMERICAL SIMULATION: A Tool in Turbulence Research, Annual Review of Fluid Mechanics, 30, 539–578, https://doi.org/10.1146/annurev.fluid.30.1.539, 1998.

Morrison, H. and Grabowski, W. W.: Comparison of Bulk and Bin Warm-Rain Microphysics Models Using a Kinematic Framework, Journal of the Atmospheric Sciences, 64, 2839–2861, https://doi.org/10.1175/jas3980, 2007.

Motter, A. E. and Campbell, D. K.: Chaos at fifty, Physics Today, 66, 27–33, https://doi.org/10.1063/pt.3.1977, 2013.

Neggers, R. A. J., Duynkerke, P. G., and Rodts, S. M. A.: Shallow cumulus convection: A validation of large-eddy simulation against aircraft and Landsat observations, Quarterly Journal of the Royal Meteorological Society, 129, 2671–2696, https://doi.org/10.1256/qj.02.93, 2003.

O'Hirok, W. and Gautier, C.: The Impact of Model Resolution on Differences between Independent Column Approximation and Monte Carlo Estimates of Shortwave Surface Irradiance and Atmospheric Heating Rate, Journal of the Atmospheric Sciences, 62, 2939–2951, https://doi.org/10.1175/jas3519.1, 2005.

Oreopoulos, L., Mlawer, E., Delamere, J., Shippert, T., Cole, J., Fomin, B., Iacono, M., Jin, Z., Li, J., Manners, J., Räisänen, P., Rose, F., Zhang, Y., Wilson, M. J., and Rossow, W. B.: The Continual Intercomparison of Radiation Codes: Results from Phase I, Journal of Geophysical Research: Atmospheres, 117, D06 118, https://doi.org/10.1029/2011JD016821, 2012.

PALM: Land surface model, URL: https://palm.muk.uni-hannover.de/trac/wiki/doc/tec/lsm, last access: 3 May 2025, 2025a.

PALM: Land surface parameters, URL: https://palm.muk.uni-hannover.de/trac/wiki/doc/app/land_surface_parameters, last access: 3 May 2025, 2025b.

Peppler, R. A. and Lamb, P. J.: Tropospheric Static Stability and Central North American Growing Season Rainfall, Monthly Weather Review, 117, 1156–1180, https://doi.org/10.1175/1520-0493(1989)117<1156:tssacn>2.0.co;2, 1989.

Petty, G. W.: Properties of Radiation, in: A First Course in Atmospheric Radiation, chap. 2, pp. 11–54, Sundog Publishing, second edn., ISBN: 978-0-9729033-1-8, 2006a.

Petty, G. W.: The Electromagnetic Spectrum, in: A First Course in Atmospheric Radiation, chap. 3, pp. 55–73, Sundog Publishing, second edn., ISBN: 978-0-9729033-1-8, 2006b.

Petty, G. W.: Thermal Emission, in: A First Course in Atmospheric Radiation, chap. 6, pp. 113–154, Sundog Publishing, second edn., ISBN: 978-0-9729033-1-8, 2006c.

Petty, G. W.: Atmospheric Transmission, in: A First Course in Atmospheric Radiation, chap. 7, pp. 155–203, Sundog Publishing, second edn., ISBN: 978-0-9729033-1-8, 2006d.

Petty, G. W.: Atmospheric Emission, in: A First Course in Atmospheric Radiation, chap. 8, pp. 204–235, Sundog Publishing, second edn., ISBN: 978-0-9729033-1-8, 2006e.

Petty, G. W.: The Radiative Transfer Equation With Scattering, in: A First Course in Atmospheric Radiation, chap. 11, pp. 320–342, Sundog Publishing, second edn., ISBN: 978-0-9729033-1-8, 2006f.

Petty, G. W.: Scattering and Absorption By Particles, in: A First Course in Atmospheric Radiation, chap. 12, pp. 343–386, Sundog Publishing, second edn., ISBN: 978-0-9729033-1-8, 2006g.

Pincus, R., Barker, H. W., and Morcrette, J.-J.: A fast, flexible, approximate technique for computing radiative transfer in inhomogeneous cloud fields, Journal of Geophysical Research: Atmospheres, 108, 4376, https://doi.org/10.1029/2002JD003322, 2003.

Prein, A. F., Langhans, W., Fosser, G., Ferrone, A., Ban, N., Goergen, K., Keller, M., Tölle, M., Gutjahr, O., Feser, F., Brisson, E., Kollet, S., Schmidli, J., van Lipzig, N. P. M., and Leung, R.: A review on regional convection-permitting climate modeling: Demonstrations, prospects, and challenges, Reviews of Geophysics, 53, 323–361, https://doi.org/10.1002/2014rg000475, 2015.

Raasch, S. and Schröter, M.: PALM - A large-eddy simulation model performing on massively parallel computers, Meteorologische Zeitschrift, 10, 363–372, https://doi.org/10.1127/0941-2948/2001/0010-0363, 2001.

Reinert, D., Prill, F., Frank, H., Denhard, M., Baldauf, M., Schraff, C., Gebhardt, C., Marsigli, C., Förstner, J., Zängl, G., Schlemmer, L., Blahak, U., and Welzbacher, C.: DWD Database Reference for the Global and Regional ICON and ICON-EPS Forecasting System, URL: https://www.dwd.de/DWD/forschung/nwv/fepub/icon_database_main.pdf, last access: 3 May 2025, 2025.

Remmler, S., Hickel, S., Fruman, M. D., and Achatz, U.: Direct Numerical Simulation of Breaking Atmospheric Gravity Waves, in: High Performance Computing in Science and Engineering '14, pp. 593–607, Springer International Publishing, https://doi.org/10.1007/978-3-319-10810-0_39, 2014.

Reuter, H., Hantel, M., and Steinacker, R.: Meteorologie, in: Erde und Planeten, chap. 3, pp. 131–310, Walter de Gruyter, https://doi.org/10.1515/9783110198027.131, 2001.

Satoh, M., Stevens, B., Judt, F., Khairoutdinov, M., Lin, S.-J., Putman, W. M., and Düben, P.: Global Cloud-Resolving Models, Current Climate Change Reports, 5, 172–184, https://doi.org/10.1007/s40641-019-00131-0, 2019.

Schroeder, D. V.: Energy in Thermal Physics, in: An Introduction to Thermal Physics, chap. 1, pp. 1–48, Addison Wesley Longman, ISBN: 0-201-38027-7, 2000.

Schumann, U., Dörnbrack, A., and Mayer, B.: Cloud-shadow effects on the structure of the convective boundary layer, Meteorologische Zeitschrift, 11, 285–294, https://doi.org/10.1127/0941-2948/2002/0011-0285, 2002.

Schäfer, S. A. K., Hogan, R. J., Klinger, C., Chiu, J. C., and Mayer, B.: Representing 3-D cloud radiation effects in two-stream schemes: 1. Longwave considerations and effective cloud edge length, Journal of Geophysical Research: Atmospheres, 121, 8567–8582, https://doi.org/10.1002/2016jd024876, 2016.

Seifert, A. and Beheng, K. D.: A two-moment cloud microphysics parameterization for mixed-phase clouds. Part 1: Model description, Meteorology and Atmospheric Physics, 92, 45–66, https://doi.org/10.1007/s00703-005-0112-4, 2005.

Selz, T., Riemer, M., and Craig, G. C.: The Transition from Practical to Intrinsic Predictability of Midlatitude Weather, Journal of the Atmospheric Sciences, 79, 2013–2030, https://doi.org/10.1175/jas-d-21-0271.1, 2022.

Steinacker, R. and Umdasch, S.: Die meteorologische Sichtweite, in: Aktuelle Aspekte der Flugmeteorologie II: Mit dem Wetter leben, vol. 39 of *Promet – Meteorologische Fortbildung*, chap. 20, pp. 80–90, Deutscher Wetterdienst, URL: http://nbn-resolving.de/urn:nbn:de:101:1-2017012510492, last access: 3 May 2025, 2014.

Stevens, B., Moeng, C.-H., Ackerman, A. S., Bretherton, C. S., Chlond, A., de Roode, S., Edwards, J., Golaz, J.-C., Jiang, H., Khairoutdinov, M., Kirkpatrick, M. P., Lewellen, D. C., Lock, A., Müller, F., Stevens, D. E., Whelan, E., and Zhu, P.: Evaluation of Large-Eddy Simulations via Observations of Nocturnal Marine Stratocumulus, Monthly Weather Review, 133, 1443–1462, https://doi.org/10.1175/mwr2930.1, 2005.

Stoll, R., Gibbs, J. A., Salesky, S. T., Anderson, W., and Calaf, M.: Large-Eddy Simulation of the Atmospheric Boundary Layer, Boundary-Layer Meteorology, 177, 541–581, https://doi.org/10.1007/s10546-020-00556-3, 2020.

Stull, R.: The Atmospheric Boundary Layer, in: Atmospheric Science: An Introductory Survey, chap. 9, pp. 375–417, Elsevier, second edn., https://doi.org/10.1016/b978-0-12-732951-2.50014-4, 2006.

Tijhuis, M., van Stratum, B. J. H., Veerman, M. A., and van Heerwaarden, C. C.: An Efficient Parameterization for Surface Shortwave 3D Radiative Effects in Large-Eddy Simulations of Shallow Cumulus Clouds, Journal of Advances in Modeling Earth Systems, 15, e2022MS003 262, https://doi.org/10.1029/2022ms003262, 2022.

Tijhuis, M., van Stratum, B. J. H., and van Heerwaarden, C. C.: The impact of coupled 3D shortwave radiative transfer on surface radiation and cumulus clouds over land, Atmospheric Chemistry and Physics, 24, 10 567–10 582, https://doi.org/10.5194/acp-24-10567-2024, 2024.

Trenberth, K. E., Fasullo, J. T., and Kiehl, J.: Earth's Global Energy Budget, Bulletin of the American Meteorological Society, 90, 311–324, https://doi.org/10.1175/2008bams2634.1, 2009.

UCAR: Namelist.input: Best Practices, URL: https://www2.mmm.ucar.edu/wrf/users/namelist_best_prac_wrf.html, last access: 3 May 2025, 2025.

Vallis, G. K.: Basics of Incompressible Turbulence, in: Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation, chap. 11, p. 413–444, Cambridge University Press, second edn., https://doi.org/10.1017/9781107588417.012, 2017.

Veerman, M. A., Pedruzo-Bagazgoitia, X., Jakub, F., Vilà-Guerau de Arellano, J., and van Heerwaarden, C. C.: Three-Dimensional Radiative Effects By Shallow Cumulus Clouds on Dynamic Heterogeneities Over a Vegetated Surface, Journal of Advances in Modeling Earth Systems, 12, e2019MS001 990, https://doi.org/10.1029/2019ms001990, 2020.

Veerman, M. A., van Stratum, B. J. H., and van Heerwaarden, C. C.: A Case Study of Cumulus Convection Over Land in Cloud-Resolving Simulations With a Coupled Ray Tracer, Geophysical Research Letters, 49, e2022GL100 808, https://doi.org/10.1029/2022gl100808, 2022.

Wallace, J. M. and Hobbs, P. V.: Introduction and Overview, in: Atmospheric Science: An Introductory Survey, chap. 1, pp. 1–24, Elsevier, second edn., https://doi.org/10.1016/B978-0-12-732951-2.50006-5, 2006a.

Wallace, J. M. and Hobbs, P. V.: Atmospheric Thermodynamics, in: Atmospheric Science: An Introductory Survey, chap. 3, pp. 63–112, Elsevier, second edn., https://doi.org/10.1016/B978-0-12-732951-2.50008-9, 2006b.

Wallace, J. M. and Hobbs, P. V.: Cloud Microphysics, in: Atmospheric Science: An Introductory Survey, chap. 6, pp. 209–270, Elsevier, second edn., https://doi.org/10.1016/B978-0-12-732951-2.50011-9, 2006c.

Wallace, J. M., Hobbs, P. V., and Fu, Q.: Radiative Transfer, in: Atmospheric Science: An Introductory Survey, chap. 4, pp. 113–152, Elsevier, second edn., https://doi.org/10.1016/B978-0-12-732951-2.50009-0, 2006.

Wang, D., Jensen, M. P., D'Iorio, J. A., Jozef, G., Giangrande, S. E., Johnson, K. L., Luo, Z. J., Starzec, M., and Mullendore, G. L.: An Observational Comparison of Level of Neutral Buoyancy and Level of Maximum Detrainment in Tropical Deep Convective Clouds, Journal of Geophysical Research: Atmospheres, 125, e2020JD032 637, https://doi.org/10.1029/2020jd032637, 2020.

Wendland, H.: Iterative Methods for Solving Linear Systems, in: Numerical Linear Algebra: An Introduction, Cambridge Texts in Applied Mathematics, chap. 4, pp. 101–131, Cambridge University Press, https://doi.org/10.1017/9781316544938.005, 2017.

WMO: Formulae for the computation of measures of humidity, in: Guide to Instruments and Methods of Observation: Volume I – Measurement of Meteorological Variables, chap. Annex 4.B, pp. 197–198, World Meteorological Organization (WMO), URL: https://library.wmo.int/idurl/4/68695, last access: 3 May 2025, 2025.

Zdunkowski, W., Trautmann, T., and Bott, A.: Two-stream methods for the solution of the radiative transfer equation, in: Radiation in the Atmosphere: A Course in Theoretical Meteorology, chap. 6, pp. 158–203, Cambridge University Press, https://doi.org/10.1017/CBO9780511535796, 2007.

Acknowledgements

The completion of this thesis marks the end of a journey that began more than five years ago, already under unusual circumstances. After returning from a skiing holiday in Austria, where I caught a virus, I began the work on this PhD thesis from home. What I initially assumed would be a matter of days turned into more than two years of remote work, as the Covid-19 pandemic unfolded shortly afterwards. More than five years later, this circle has somewhat closed, with the final months of this thesis — mainly dedicated to writing — again carried out in relative isolation at home. The journey toward this PhD would not have been possible without the support of many people. I would therefore like to take this opportunity to thank and acknowledge all those who supported me throughout this time.

First of all, I would like to thank my supervisor, Prof. Dr. Bernhard Mayer. Together with Prof. Dr. Christoph Knote, he was the one who convinced me to pursue this thesis in the first place, as I was initially unsure whether I wanted to do a PhD. He also provided the foundational idea for this work and, throughout the entire time, consistently contributed new ideas, was always open to discussions, and generally maintained an open door. There are certainly worse supervisors one could have. I am also grateful for the patience he showed when things took longer than expected — especially during the writing phases. At this point, I would also like to acknowledge Dr. Claudia Emde, Dr. Fabian Hoffmann and Prof. Dr. Aiko Voigt for co-supervising parts of this work. Your input during our meetings over the past few years clearly helped improve this thesis, and I truly appreciated your thoughtful remarks throughout.

Next, I would like to thank Dr. Fabian Jakub. As the developer of the original TenStream solver and someone with broad technical expertise, he was an enormous help with all kinds of technical questions — related to, but not limited to, the TenStream solver. Throughout the entire time, he patiently answered my (sometimes possibly stupid and repetitive) questions about the technical issues I encountered. Beyond that, we had numerous scientific discussions that contributed significantly to many of the methods applied in this thesis and offered valuable guidance in interpreting many of the results. Without his help, this work would certainly not have been possible — at least not within this timeframe. And although I could surely not fulfill all your wishes, I hope that at least some of your questions are covered in this work.

Apart from that, I would like to thank Dr. Claudia Emde, Dr. Fabian Hoffmann, Dr. Fabian Jakub, Dr. Jung-Sub Lim, Prof. Dr. Bernhard Mayer, and Lea Volkmer for proofreading parts of this thesis. As someone whose Bachelor's and Master's theses were both finished just hours before their respective deadlines, this was a somewhat new experience for me. The work certainly benefited from your comments, and I hope that at least some of the improvements are visible. At this point, I would also like to thank Dr. Claudia Emde, Dr. Fabian Jakub, Mihail Manev, Prof. Dr. Bernhard Mayer, and Prof. Dr. Aiko Voigt for proofreading the paper incorporated into this thesis. Like the rest of this work, it clearly profited from your feedback.

I would also like to take this opportunity to acknowledge the "Waves to Weather" project and its community, under whose umbrella large parts of this thesis were developed, and in particular Prof. Dr. George Craig, who led the project. The passion he brought to it was always inspiring — especially during the annual meetings, where his talks were consistently among the highlights and never failed to leave me with a renewed sense of optimism. I would also like to thank Dr. Audine Laurian for her flawless organization of these meetings, for saving us from a lot of bureaucratic tasks, and for her many outreach activities that hopefully helped increase the visibility of the project in the public. It was always a pleasure to be part of these events. Thanks also to my fellow PhD candidates Mihail Manev and Dr. Behrooz Keshtgar, with whom I had the pleasure of sharing a subproject and always enjoyed working with. Together with Sören Schmidt and Dr. Hyunju Jung, I further had the pleasure of serving on the ECS committee in the final year of the project. What started with mixed feelings turned out to be a great experience — especially thanks to the two of you. Organizing the final ECS Annual Meeting in Hamburg in May 2023 was a particular highlight in that regard and provided a great conclusion to that time.

Furthermore, I would definitely like to acknowledge the working environment at the Meteorological Institute in Munich. First and foremost, again, my thanks go to Prof. Dr. Bernhard Mayer for creating and leading such a great group. I am also grateful to Dr. Veronika Pörtge, Dr. Gregor Köcher, and Lea Volkmer for being such wonderful office mates. Thank you for never once complaining about my rather unusual working hours — or about the long stories I often felt the need to share. Of the three, I would especially like to acknowledge Veronika — I have known you since the beginning of my studies in October 2013, and it has truly been a pleasure to share this journey with you for such a long time. I also want to thank Mihail Manev and Dr. Jung-Sub Lim for the many interesting discussions we had during late nights at the university — they surely made those nighttime hours much more enjoyable. Many thanks as well to all the other great people at the institute — Alex, Anna, Barbara, Ben, Jonas, Leo, Leonie, Levin, Maria, Markus, Matjaž, Michi, Nikos, Paul, Philipp, Stefan, Sukanya, Takumi, and Tobias, just to name a few — for the familiar and welcoming atmosphere, and for the always enjoyable after-work activities: be it PhD hat constructions, gaming sessions, trips to the Isarphilharmonie, or the occasional lunch breaks I managed to join.

Last but not least, I would like to thank my family and friends. I have never once regretted moving to Munich in October 2013 to study physics and meteorology. And that is not only because the subject struck the perfect balance between classical physics and the more practical meteorology, which I came to love in no small part thanks to my geography teacher in 11^{th} and 12^{th} grade at school. More than anything, it was because of all the wonderful people I had the pleasure to meet here. Many of them — Veronika, Nina, and Kevin, just to name a few — I met during the very first week of my studies, and we have remained in contact ever since. Together with the unwavering support of my family, they were undoubtedly the cornerstone that made this thesis possible. Thank you for that — and sorry for the reduced contact during these intense final months of writing!

Financial support: The research leading to these results has been done within the subproject “B4 — Radiative interactions at the NWP scale and their impact on midlatitude cyclone predictability” of the Transregional Collaborative Research Center SFB/TRR 165 "Waves to Weather" (www.wavestoweather.de) funded by the German Research Foundation (DFG).

Disclaimer: The wording and phrasing in parts of this thesis were improved with the assistance of OpenAI’s ChatGPT models o1, o4, and o4-mini.