Testing the usage of neural networks in the shortwave radiation parameterization of the WRF model

Master Thesis

Written by *Wiebke Margitta Kolbe*

August 17, 2020

Supervised by
Eigil Kaas

University of Copenhagen

UNIVERSITY OF COPENHAGEN

Name of Institute:	University of Copenhagen
Name of Department:	Niels Bohr Institute
Author(s):	Wiebke Margitta Kolbe
Email:	jcl674@alumni.ku.dk
Title and subtitle:	Testing the usage of neural networks in the shortwave radiation parameterization of the WRF model
Supervisor(s):	Eigil Kaas
Handed in:	17.08.2020

Abstract

Radiative transfers in the atmosphere are difficult to compute accurately in numerical weather prediction (NWP) models, without the procedure becoming too computationally expensive.

In this thesis it has therefore been tested to substitute a part of the shortwave radiation parameterization in the Weather Research and Forecasting (WRF) model with neural networks, to investigate a possible increase in computational efficiency of such a modified parameterization and its accuracy.

The data set used to train the neural networks was created with the RRTMG-fast shortwave radiation parameterization scheme in the WRF model.

After several optimization processes, three configurations of neural networks were implemented and tested in the WRF model, replacing an computationally expensive part of the RRTMG-fast scheme.

To evaluate the three neural network modified shortwave schemes, four 96-hour simulations were carried out as case studies, to compare how the model performs in different weather situations.

Additionally to the original RRTMG-fast scheme and the three variants modified with neural networks, the four case studies were simulated with three other shortwave parameterization schemes as well: the RRTMG, New Goddard and Dudhia schemes.

Comparison of the results showed that the modified neural network schemes were able to make predictions similar to the original RRTMG-fast scheme, but were computationally slower.

A quick test addressed one of the causes, the activation function, and suggested that the computational time of the neural networks can be reduced significantly by using a different activation, though the new performance has yet to be evaluated, while possible further optimizations are addressed.

Acknowledgments

First and foremost I would like to express my gratitude to my supervisor Eigil Kaas, who even in challenging times, both academically, but also worldly, such as times of social distancing, has never stopped providing guidance and spreading his enthusiasm.

Additionally, warm thank you's go out to my fellow students and members of the atmospheric science group. Discussing and presenting each others projects and approaches with neural networks was a huge inspiration and motivation for me.

A special thanks also goes to Kristian Pagh Nielsen, who was always kind to share his knowledge and expertise about radiative transfers and their approximations in NWP models.

Last, but definitely not least, I would also like to thank my family and friends for their continued support. Even though I'm sure you by now have heard enough of my rambling to be able to write a small compendium about neural networks, radiation, and my personal favourite: Rayleigh scattering. I really appreciate it.

Contents

1 Introduction

In recent years the applications of machine learning and neural networks have increased and continue to do so, as computing power and huge data sets become more easily available.
For computation heavy weather and climate models it is therefore interesting to investigate if it is possible to take advantage of neural networks to save computation time while maintaining good quality forecasts. Supercomputers necessary for weather forecasting and advanced climate models consume a large amount of energy, e.g. the UK Met Office's supercomputer consumes ca. 2.7 Megawatt (MW) [MetOffice-website,]. Meanwhile, the new data center of ECMWF is build upon a 10 MW supply, planned to be upgraded to support 20 MW in the future [ECMWF, 2017]. The high electrical consumption rates, partly due to the cooling of the machines, has also lead to a cooperation between the Danish Meteorological Institute, DMI, and the Icelandic Meteorological Office, where a common supercomputer has been set up on Iceland, taking advantage of the general colder climate, lowering the necessary energy to operate [Ingeniøren, 2016]. Since computation time and energy consumption is directly correlated, this is another reason to focus on optimizing computation routines.

There is a strong interest in many projects working with the numerous ways of using neural networks in numerical weather prediction (NWP) models, e.g. bias correction of input data, learning about model error during data assimilation process, or replacing computation-heavy components of the model, in hopes of improving forecasts such as from ECMWF [Dueben, 2020]. Past studies have investigated the application of neural networks to improve predictions of a single atmospheric variable with different approaches, e.g. precipitation, using output data from a single model [Coblenz, 2015]; and multiple models [Krasnopolsky and Lin, 2012].
In other surveys the 500hPa geopotential height has been used to analyze how well neural networks are able to learn non-linear atmospheric dynamics, investigating challenges and different configurations for neural network based predictions ([Dueben and Bauer, 2018]; [Weyn et al., 2019]). The implementation of neural networks in model's parameterization schemes, such as in the longwave parameterization at ECMWF, has been tested as well ([Chevallier et al., 1998]; [Krasnopolsky et al., 2005]).

The computation of radiation in NWP models takes up a large portion of the computing time for the whole model, compared to other physical parameterizations and calculations. Estimating radiative transfers involve multiple challenges both at the surface and through the atmosphere, including clouds and clear sky absorption conditions [Hogan et al., 2018]. Therefore, the efficiency of the radiation parameterization is important, to reduce the time consumption as much as possible, while keeping the high quality.

Radiation, i.e. electromagnetic waves, come in a variety of different wavelengths. Properties like reflection, absorption and emission of those waves vary depending on the specific wavelength, e.g. emission of longer wavelengths by the earth is a key mechanism for the greenhouse effect, but the earth does not emit shortwave radiation, because the earth is too cold. The Radiation spectrum is usually divided into two wavelength sections, the shortwave and the longwave radiation, which will be explained further in the following sections. In atmospheric models both types are handled separately in the code, so there is a parameterization scheme for each of them.

In this study it has been focused on the shortwave radiation parameterization of the weather research and forecast (WRF) model. Case studies with an original radiation scheme and a modified scheme using neural networks have been executed, to test the application of neural networks in NWP models and possible improvement of the parameterization scheme. Both the computational efficiency and performance of the weather prediction will be analyzed.

The reader will be presented some theory behind radiative transfers in the atmosphere and the common methods of implementation into atmospheric models (parameterizations), as well as a short introduction to artifical neural networks and machine learning in section 2. Thereafter the focus will move towards the WRF model used in this study and its radiation parameterization schemes in section 3. The development and implementation of the used neural networks will be presented in section 4. Lastly results of the model runs with the original scheme and modified schemes with neural networks will be presented in section 5 and discussed in section 6, ending with an overall conclusion on the results in section 7.

2 Theory

Although there are methods to calculate radiative transfers with high precision, these are not actually used in NWP models, for multiple reasons.
One would be that the computation is simply too expensive for operational weather forecasting models to be of use. Another one is that the needed radiative variables, e.g. optical depth, albedo, etc., which will be presented in the following sections, are not part of or directly describable by the governing equations and thermodynamic fields the model is build upon, such as e.g. temperature or pressure. Those quantities must therefore be approximated with both the thermodynamic variables as well as additional physical quantities from other parameterization schemes, e.g. cloud/gas micro-physics.

While approximate solutions for radiative transports in parameterizations are used, the resulting uncertainties of those methods need to be kept at a minimum. Good parameterizations for radiative transfers are needed, as radiation plays an important role for not only heating and cooling in the atmosphere, but also for e.g. the surface energy balance calculated in the separate Land Surface parameterization, which depends on radiative surface fluxes. Since the different schemes of the model use and provide inputs among one another, one parameterization failing will hinder the model from advancing further.

Before examining the model code and the approximations made in the parameterizations, it is important to understand the physics that the WRF model tries to simulate. The following section will therefore first focus on the real world physics of radiation in the Earth's atmosphere and then on possible methods of implementation in NWP models.
The following theoretical section about radiative transfer takes inspiration of parts of the books [Wallace and Hobbs., 2006] and [Randall, 2015], which give a good introduction to atmospheric dynamics as well as [Liou, 2002] and [Thomas and Stamnes, 1999], which offer a more extensive description of radiative transfers in the atmosphere. Lastly a brief introduction to neural networks relevant to this study will be given.

2.1 Radiation in the real world

The main mechanism by which the Earth can exchange energy to and from outer space is radiation, i.e. electromagnetic waves. Alongside sensible and latent heat exchanges, radiative transfers also redistribute energy within the Earth's own system. The insolation, i.e. the incident solar radiation hitting the top of the atmosphere (TOA), is the most important upper boundary condition of the Earth's global circulation of the atmosphere. The amount of sunlight reaching the TOA varies with geography and time, as well as the Earth's geometry and orbit. The resulting imbalance of heat distribution is related to major atmospheric dynamics, redistributing energy and mass in the atmosphere. Both the Earth's outgoing radiative fluxes and large scale atmospheric motions can be observed with satellites.

As described in quantum physics, transitions between different (distinct) energy states lead to emission of waves with different wavelengths. These electromagnetic waves come in a wide spectrum of wavelengths and are able to travel through both vacuum and media. Radiation is often categorized into a shortwave spectrum, i.e. solar radiation, which is mostly visible light coming from the sun, and a longwave spectrum, thermal radiation, which is emitted by the Earth. When such a wave passes through the atmosphere, which contains different gases and aerosols, the wave might be absorbed, emitted or scattered, depending on both the quantities of the wavelength and the hit molecule. It is a complex process, since it depends on different properties of the individual wavelength and the traversed medium.

For instance, shortwave radiation is mostly scattered and reflected in the atmosphere, while only a very small amount is emitted. The reflection and scattering of these electromagnetic waves can become visible as colours. The effect of scattering depends on the specific wavelength and size of the particle, which can also be expressed as a size parameter x:

$$x = \frac{2\pi r}{\lambda} \tag{2.1}$$

The size parameter x is dimensionless and describes the ratio between the radius r of the particle and the wavelength λ. For very small particles $x << 1$, e.g. air molecules, which the atmosphere mostly consists of, the scattered intensity I is inversely proportional to the wavelength:

$$I \propto \lambda^{-4} \tag{2.2}$$

It follows that the intensity and scattering efficiency are large for small wavelengths. Thus, the small wavelengths at the shorter end of the solar radiation spectrum, visible as blue, get scattered a lot more than the larger, red wavelengths in the atmosphere. It is because of this process that the sky appears blue during the day, this phenomena specifically is called Rayleigh-scattering.

Not all of the solar radiation from the Sun that hits the Earth's TOA passes through the atmosphere and reaches the surface. Parts of it get reflected at the TOA by clouds, or gets absorbed on the way downwards. The surface does not absorb all of the radiation that it's hit with either, some gets reflected back up into the atmosphere, where it again can get absorbed or scattered. The incident angle with which the wave hits the medium is also important, as the absorption and reflection changes with elevations. White, light surfaces such as snow and clouds reflect a lot of the shortwave spectrum, though not of the longwave radiation, while dark surfaces such as the ocean absorb a lot of the radiation, i.e. the ocean has a low value of albedo. Albedo is a measure of how much solar radiation is reflected, an albedo of one describes a perfect reflector, while an albedo of zero indicates the absorption of all wavelengths.

Figure 2.1 depicts how much of the spectrum of the solar radiation of the Sun reaching the TOA, yellow shading, actually reaches the Earth's surface, red shading.

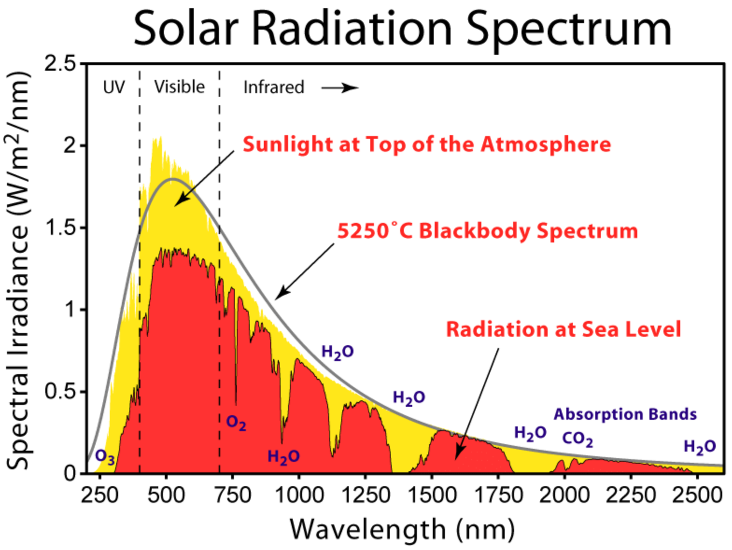


Figure 2.1: Solar radiation spectrum at the top of the atmosphere (yellow), and at surface (red). The ideal blackbody spectrum is shown as black curve. The part marked as the beginning of the Infrared spectrum is often referred to as near-Infrared radiation. Figure 4.1 from [Inness and Dorling., 2013]

It is noticeable, how the amount of absorption differs for the individual wavelengths, some wavelengths have even been absorbed completely. Different gases in the atmosphere, absorb different wavelengths, giving rise to those absorption bands. The most prominent ones for the shortwave radiation are those of ozone (O_3) in the ultraviolet (UV) part and water vapour (H_2O) in the near-Infrared (IR) part of the spectrum. The border between the small spectrum of visible light to ultra violet and near-Infrared wavelengths is shown in the figure as well.

The black curve shows the idealized blackbody spectrum. A blackbody is a theoretical idealized body which absorbs radiation of all wavelengths, as well as being able to emit radiation in the complete, continuous spectrum.
The intensity of emitted radiation of such a blackbody, is given by the Planck function:

$$B_\lambda(T) = \frac{c_1 \lambda^{-5}}{\pi(e^{c_2/\lambda T} - 1)} \tag{2.3}$$

where B_λ is the blackbody monochromatic intensity, i.e. blackbody intensity of a specific wavelength λ, depending completely on the temperature of the body.
T is temperature, $c_1 = 3.74 \times 10^{-10}\text{Wm}^2$ and $c_2 = 1.45 \times 10^{-2}\text{Wm}^2$.
By integrating πB_λ over all wavelenghts, one arrives at the Stefan-Boltzmann law:

$$F = \sigma T^4 \tag{2.4}$$

where F is the (blackbody) flux density and $\sigma = 5.67 \times 10^{-8}\text{Wm}^2\text{K}^{-4}$ the Stefan-Boltzmann constant.

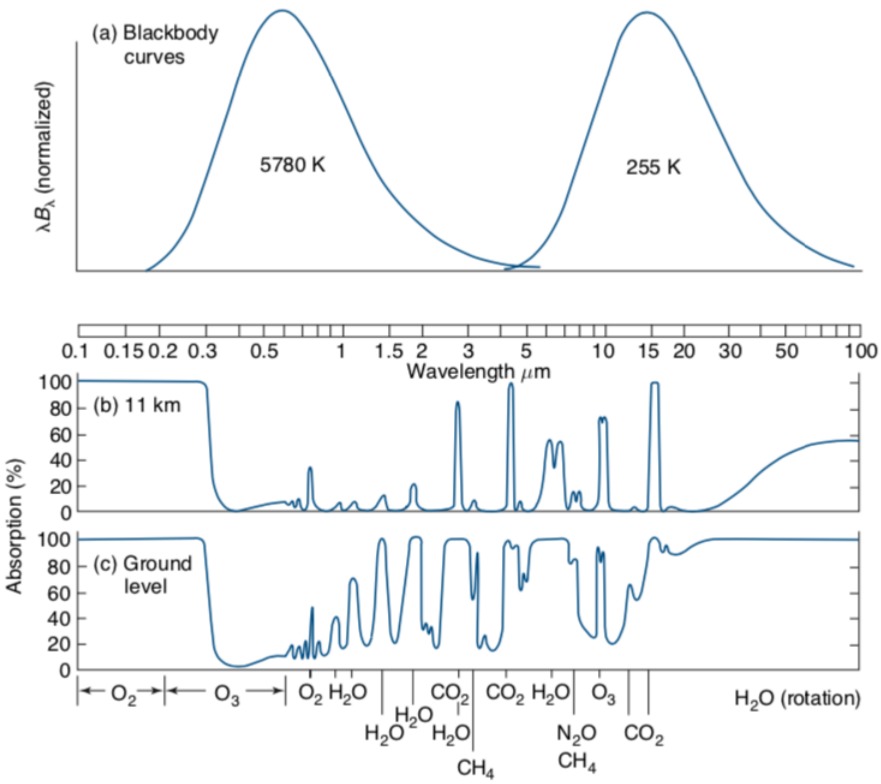


Figure 2.2: (a) Blackbody spectra for Sun (left) and Earth (right), normalized for an easier comparison between the two, as the magnitude of the Sun's curve is much larger than that of the Earth. Absorption spectrum shown for (b) upper part of the atmosphere above 11km height and (c) entire atmosphere. Figure 4.7 from [Wallace and Hobbs., 2006]

If one measures the flux density of a nonblack body and uses the Stefan-Boltzmann law from equation (2.4) to calculate the the temperature T, then T will not be the blackbody temperature, but rather the equivalent blackbody temperature T_E.
The flux density of the solar insolation, also called solar constant, is $F_S = 1368\text{Wm}^{-2}$. Even if the Earth is assumed to be in a radiative equilibrium state, where there is no energy change because of radiative transfers, the flux density of the emitted longwave radiation of the Earth F_E, is not equal to the flux F_S of the incoming solar radiation. There are two reasons for this:

First, there is a certain amount of solar radiation reflected by the Earth without any absorption, ca.30%, so the planetary albedo for the whole Earth can be approximated to $A = 0.3$. Therefore the amount of absorbed radiation becomes $(1 - A)F_S$.

Second, the incoming solar radiation from the sun hits only a cross-section of the Earth, a disk area $A_{disk} = \pi R^2$, where R is the Earth's radius, while the Earth emits longwave radiation around its whole surface, approximately the area of a sphere, $A_{shpere} = 4\pi R^2$, four times larger than the area of the cross-section. Thus the Earth's emitted flux density becomes:

$$F_E = (1-A)F_S\frac{A_{disk}}{A_{sphere}} = \frac{(1-A)F_S}{4} = 239.4Wm^{-2} \tag{2.5}$$

From this the equivalent blackbody temperature T$_E$ of the Earth becomes, using the Stefan-Boltzmann law from equation (2.4):

$$T_E = \sqrt[4]{\frac{F_E}{\sigma}} = 255K \tag{2.6}$$

So T$_E$ is calculated to be 255 K for the Earth, which is also noted in figure 2.2 (a). The curve of a blackbody spectrum as those shown in figures 2.1 and 2.2 was calculated with the Planck function, equation (2.3), and is only temperature dependent. Figure 2.2 (a) shows the ideal emission blackbody spectrum for both the Sun on the left, as seen before, as well as for the Earth on the right. It becomes clear, that because the Sun is much warmer than the Earth, the two spectra are almost not overlapping. This is taken advantage of in NWP models, where two sets of parameterizations are made separately, one for the solar, shortwave radiation, and one for the terrestrial, longwave radiation type [Inness and Dorling., 2013]. From figure 2.2 (b) and (c) it can be seen that the absorption bands and corresponding gases also differ for the two categories of wavelengths. For instance absorbs carbon dioxide (CO$_2$) radiation mostly in the longwave spectrum, e.g. with a very prominent peak at 16 μm. Additionally, since (b) and (c) depict absorption at different altitudes, it indicates that the efficiency as absorber and amount of different gases in the atmosphere varies.

The absorption bands of the longwave spectrum and their corresponding gases are also referred to as green house effect. While the shortwave spectrum shows radiation moving downwards through the atmosphere, the longwave spectrum is the radiation emitted by the Earth. However, gases such as water vapor and carbon dioxide can absorb much of this emitted radiation and re-emit it into all directions, causing a fair amount to transfer back to the surface, heating it. Thus these processes are important for the Earth's energy budget.

Not shown here, but also important in terms of scattering and reflecting radiation are aerosols in the atmosphere as well as clouds, which are also important due to their high albedo.

2.1.1 The Radiative Transfer Equation (RTE)

Figure 2.3 depicts a single radiation beam, i.e. radiation travelling in a specific direction such as a ray of light, moving through a medium, changing it's intensity I_λ due to scattering, absorption and emission.

In this case a beam of a single wavelength is considered, called monochromatic radiation, however, to calculate the intensity of a spectrum of wavelengths, one does only

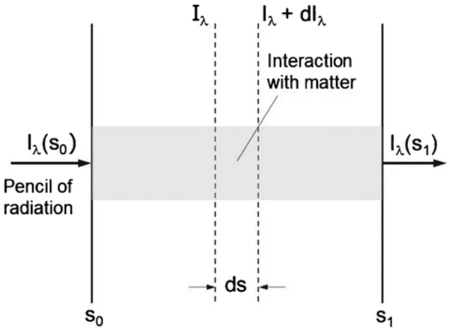


Figure 2.3: Illustration of a radiation beam depleting while passing through an extinction medium. Figure 1.12 from [Liou, 2002]

need to integrate over the desired wavelength bands:

$$I = \int_{\lambda_1}^{\lambda_2} I_\lambda d\lambda = \int_{\nu_1}^{\nu_2} I_\nu d\nu \tag{2.7}$$

I is then the total intensity, i.e. the energy emitted by the electromagnetic waves moving through a unit area per unit time. λ is the wavelength and $\nu = 1/\lambda$ is the wave number, i.e. the inverse of the wavelength. All radiation equations can be expressed both with λ and ν, note that the energy is inversely proportional to λ, i.e. a longer wavelength transports less energy.

Considering the monochromatic intensity I_λ in figure 2.3 again, and defining this as the initial intensity, while $I_\lambda + dI_\lambda$ describes the intensity of the beam after travelling through the medium with thickness ds, makes it possible to define the change of intensity dI_λ with:

$$dI_\lambda = -I_\lambda k_\lambda \rho r ds \tag{2.8}$$

where ρ is the density of air, r is the mass of absorbing (and/or scattering) gas per unit mass of air and k_λ is called the mass absorption coefficient, which depending on r describes extinction due to both absorption and scattering. The product $k_\lambda \rho r$ is the volume extinction coefficient, which includes both effects of absorption and scattering, depending on the medium/gas.

Both scattering and absorption lead to the extinction of a passing solar radiation beam. In the same way there can be a strengthening of the initial I_λ, if there is emission in the medium, as well as multiple scattering in all directions. This effect can then be combined into the source coefficient j_λ, analogue to equation (2.8):

$$dI_\lambda = j_\lambda \rho r ds \tag{2.9}$$

Note that there is no I_λ on the right hand side of this equation, as the emission and multiple scattering is not dependent on the initial intensity of the beam.

Combining equation (2.8) and (2.9) gives an expression for the complete change of I_λ:

$$dI_\lambda = -I_\lambda k_\lambda \rho r ds + j_\lambda \rho r ds \tag{2.10}$$

Defining the source function J_λ as ratio between the source coefficient j_λ and the mass absorption coefficient k_λ makes it possible to simplify the equation into:

$$J_\lambda \equiv j_\lambda / k_\lambda \tag{2.11}$$

$$\frac{dI_\lambda}{k_\lambda \rho r ds} = -I_\lambda + J_\lambda \tag{2.12}$$

Equation (2.12) is the general form of the radiative transfer equation (RTE), describing the interaction between radiation and a medium, taking into account scattering, absorption and emission. The goal of the radiation parameterization in a NWP model is to get close to the real solution of this equation.

2.1.2 The Beer-Bouguer-Lambert Law

Solving the radiative transfer equation depicted in equation (2.12) is not trival. Evaluating the source function J_λ for real world applications proves difficult and will be described in the following sections. For a simple case, where emission and effects of multiple scattering can be neglected, equation (2.12) reduces to the simple form:

$$\frac{dI_\lambda}{k_\lambda \rho r ds} = -I_\lambda \tag{2.13}$$

which is a differential equation that can be solved analytical if boundary conditions are provided. Considering the example from before, depicted in figure 2.3, the initial intensity is $I_\lambda(0)$ at $s = 0$ and the intensity is $I_\lambda(s_1)$ at a distance $s = s_1$. Then integrating equation (2.13) with these boundary values yields:

$$I_\lambda(s_1) = I_\lambda(0) \exp\Big(- \int_0^{s_1} k_\lambda \rho r ds \Big) \tag{2.14}$$

For a homogeneous medium k_λ is constant, i.e. it is independent of the distance s that the radiative beam travels through the medium. Therefore one can simplify equation (2.14) further for such cases, by defining the path length u:

$$u = \int_0^{s_1} \rho r ds \tag{2.15}$$

with which equation (2.14) becomes:

$$I_\lambda(s_1) = I_\lambda(0) e^{-k_\lambda u} \tag{2.16}$$

This equation is known as the Beer-Bouguer-Lambert Law, sometimes also referred to under shorter names as Beer's law or Lambert's law. It follows from this formula, that the intensity in a homogeneous medium decreases from its initial value as an exponential function depending only on the path length and the absorption coefficient. Note that since there is no dependency on the direction of the beam in equation (2.16), it is also applicable to flux and flux density calculations. Moreover, one can define the (monochromatic) transmissivity T_λ from equation (2.16) :

$$T_\lambda = \frac{I_\lambda(s_1)}{I_\lambda(0)} = e^{-k_\lambda u} \tag{2.17}$$

The layer's (monochromatic) transmissivity T_λ describes the amount of undepleted intensity, that managed to pass through the layer. From this definition it should become evident, that the transmissivity is a quantity that ranges between 0 and 1. For a value of 0 no intensity will pass through a medium, i.e. all intensity is either absorbed and/or scattered away, while a value of 1 describes a medium where the beam of a wavelength can pass through unhindered, i.e. the medium is transparent for the wavelength.

The same principle can be applied to the layer's (monochromatic) absorptivity A_λ, which quantifies the amount of absorbed intensity, and (monochromatic) reflectivity R_λ, i.e. the amount of radiation that is reflected by the medium through scattering processes.

Additionally, in absence of scattering, one can relate the (monochromatic) absorptivity A_λ to the transmissivity T_λ:

$$A_\lambda = 1 - T_\lambda = 1 - e^{-k_\lambda u} \tag{2.18}$$

This is an expression of energy conservation, as all radiation in a non-scattering medium either passes through or will be absorbed. Likewise, the energy conservation can be formulated for cases with scattering as:

$$1 = A_\lambda + T_\lambda + R_\lambda \tag{2.19}$$

2.1.3 RTE for Plane-Parallel Atmospheres

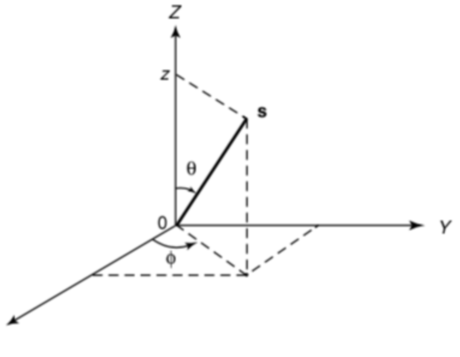


Figure 2.4: Illustration of the spherical coordinates used for a plane-parallel atmosphere. θ is the zenith angle, ϕ is the azimuthal angle and **s** is the position vector. Figure 1.15 from [Liou, 2002]

For most radiative transfers applications in the atmosphere it is useful to divide the atmosphere into plane-parallel portions. In a plane-parallel framework the physical variables, such as temperature, vary only in the vertical direction, i.e. they are functions of height or pressure only. Such a framework is natural for NWP models, where the atmosphere is divided into vertical columns, which will be further described in section 3.

The advantage of using a plane-parallel structure is that one can easily measure the distance between the normal of the plane of stratification and a radiative beam and its travel path, for all possible incident angles. Figure 2.4 depicts a coordinate system for the plane-parallel atmosphere, where **s** is the position vector, while θ and ϕ are the zenith and azimuthal angles respectively.
It follows from this geometry that $ds = \frac{dz}{\cos\theta}$, with which the general RTE in equation (2.12) takes the following form for plane-parallel atmospheres:

$$\cos\theta \frac{dI_\lambda(z;\theta,\phi)}{k_\lambda \rho r dz} = -I_\lambda(z;\theta,\phi) + J_\lambda(z;\theta,\phi) \tag{2.20}$$

One can now introduce a parameter called the optical depth (thickness) τ_λ, which describes the amount of depletion a radiative beam would experience during a direct passage through a layer, when $\theta = 0$. τ_λ will be important for the radiation parameterization in NWP models in the following sections. The dimensionless optical depth, measured downward from the upper boundary, the TOA, can be defined as:

$$\tau_\lambda \equiv \int_z^\infty k_\lambda \rho r dz' \tag{2.21}$$

With this optical depth τ_λ and $\mu = \cos\theta$, equation (2.20) can be written as:

$$\mu \frac{dI_\lambda(\tau_\lambda;\mu,\phi)}{d\tau_\lambda} = I_\lambda(\tau_\lambda;\mu,\phi) - J_\lambda(\tau_\lambda;\mu,\phi) \tag{2.22}$$

This is the general radiative transfer equation for plane-parallel atmospheres, which is the fundamental equation for all following discussions of radiative processes in the atmosphere.

2.1.4 General solution of the RTE for the solar spectrum

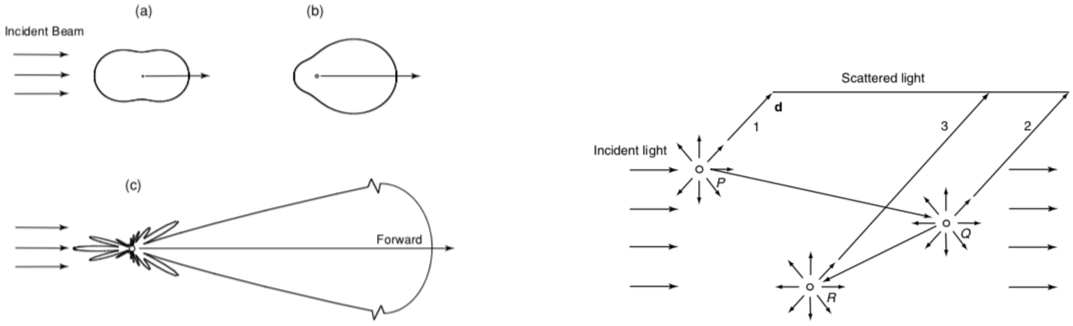
In section 2.1.2 a solution to the RTE for the idealized case of a non-scattering medium was derived. This section will focus on a solution to the general RTE in its newly derived plane-parallel form in equation (2.22) for shortwave radiation, i.e. the solar spectrum.

Considering a scattering medium and the general RTE, one needs to take into account both the effects that cause the extinction of the initial intensity, i.e. absorption and scattering, as well as the strengthening of the intensity due to emission and multiple scattering, which defines the source function J_λ, for all wavelengths.
The extinction of a beam's intensity in absence of scattering and emission has been described earlier in section 2.1.2. The focus will now be shifted to the source function J_λ.

As it was seen in section 2.1 and in figure 2.2, the Earth's emission lies mainly in the IR spectrum and can be described with the Planck's function of a black body, so that, in the absence of multiple scattering, the source function for the Earth can be expressed as:

$$J_\lambda = B_\lambda(T) \tag{2.23}$$

This would be true for an atmosphere where the emission of the earth would travel through the atmosphere unhindered, but that is not true in reality. Due to gasses absorbing and re-emitting this radiation, i.e. the greenhouse effect, one would need to correct for these processes with an additional absorption coefficient.
However, there is no significant overlap between the shortwave and longwave spectra as aforementioned, and thus the emissions can be neglected for applications in the solar spectrum.
Therefore only scattering processes need to be considered for the RTE.



(a) scattering

(b) multiple scattering

Figure 2.5: Illustrations for (a) scattering and (b) multiple scattering processes. Figure (a) shows different angular patterns of scattering, while Figure (b) shows an example of multi order scattering up to the third order. Figure 1.4 and 1.5 from [Liou, 2002]

There are two types of scattering that need to be examined. First the direct scattering of a solar beam due to a medium, and second the multiple scattering of diffuse radiation. Diffuse radiation describes radiative beams that have at least been scattered once, while solar beams that have traveled directly from the Sun to the Earth's surface are called direct radiation.

The multiple scattering is a sequence of scattering processes and all of them are affected by the scattering angle and properties of the scattering medium, i.e. the air and cloud particles, as well as aerosols.

Different kinds of scattering are depicted in figure 2.5 (a), where the same beam is scattered by different angular patterns, some scattering more evenly to all directions, others scattering more towards a certain direction, e.g. forward as seen in (c).
Figure 2.5 (b) shows an example of multiple scattering, up to the third order, i.e. the initial beam is scattered three times until it takes the direction of interest.

While the scattering at the three points P, Q and R is depicted to be the same at each of them, this is not necessary the case, depending on whether the scattering medium is homogeneous or not.

To calculate the scattered intensities the angular distribution must be known for the whole path the scattered light undertakes. For this a phase function $P_\lambda(\cos\Theta)$ is introduced.

The phase function holds information about the angular distributions through the scattering angle Θ, which can be expressed in the same spherical coordinate system as used for the plane-parallel RTE shown in figure 2.4, with $\mu = \cos\theta$.

Considering an initial incoming beam with incident angles μ' and ϕ', which is scattered (multiple times), until it has angles μ and ϕ as it leaves the scattering medium, lets the phase function become a function of μ, ϕ, μ' and ϕ'. $P_\lambda(\mu;\phi;\mu';\phi')$ describes then the angular distribution of the scattered beam.

$$\cos\Theta = \mu\mu' + (1-\mu^2)^{1/2}(1-\mu'^2)^{1/2}\cos(\phi'-\phi) \tag{2.24}$$

If the Sun's position is described by the angles μ_0 and ϕ_0, the scattering of a Sun's direct solar beam can be constructed with the solar zenith angle, which is μ_0, as well as a scattering coefficient $\beta_{s,\lambda}$, which describes how efficient the medium or particle is at scattering the beam. This direct scattering can then be formulated as follows for the source function:

$$J_\lambda = \beta_{s,\lambda} F_{\odot,\lambda} e^{-\tau_\lambda/\mu_0} P_\lambda(\mu,\phi;-\mu_0,\phi_0)\frac{1}{4\pi} \tag{2.25}$$

Here $F_{\odot,\lambda}$ is a part of the solar flux from the TOA at wavelength λ. The factor $\frac{1}{4\pi}$ is the ratio of the 4π solid angle. Note that as downward angles are per definition negative, a minus-sign was added to μ_0 in equation (2.25).

While equation (2.25) describes the contribution of the direct scattering to the source function, also multiple scattering processes, i.e. diffuse radiation, need to be accounted for. This can be done with the following double integral:

$$J_\lambda = \beta_{s,\lambda}\int_0^{2\pi}\int_{-1}^{1} I_\lambda(\tau;\mu',\phi')P_\lambda(\mu,\phi;\mu',\phi')\frac{d\mu' d\phi'}{4\pi} \tag{2.26}$$

Like the scattering coefficient $\beta_{s,\lambda}$, a extinction coefficient $\beta_{e,\lambda}$ can be defined. This is useful, as one can then define the ratio between scattering and extinction as single-scattering albedo $\tilde{\omega}_\lambda$:

$$\tilde{\omega}_\lambda = \frac{\beta_{s,\lambda}}{\beta_{e,\lambda}} \tag{2.27}$$

The term albedo was introduced earlier as a ratio of how much radiation is reflected or absorbed, i.e. how opaque a medium appears for a radiative beam, which now can be quantitatively described with the optical depth τ_λ. Likewise the single-scattering albedo $\tilde{\omega}_\lambda$ is now an expression quantifying the amount of scattering by a medium.

$\tilde{\omega}_\lambda = 0$ describes a non-scattering medium as investigated before with the Beer-Bouguer-Lambert law in section 2.1.2, while a medium with $\tilde{\omega}_\lambda = 1$ scatters all incoming radiation with wavelength λ.
Taking the general RTE equation (2.22) and adding the direct scattering in equation (2.25), diffuse scattering from equation (2.26) as well as the single-scattering albedo in equation (2.27), the RTE takes the form[1]:

$$\mu \frac{dI_\lambda(\tau_\lambda;\mu,\phi)}{d\tau_\lambda} = I_\lambda(\tau_\lambda;\mu,\phi) - \frac{\tilde{\omega}_\lambda}{4\pi}\int_0^{2\pi}\int_{-1}^{1} I_\lambda(\tau;\mu',\phi')P_\lambda(\mu,\phi;\mu',\phi')d\mu' d\phi' \\ -\frac{\tilde{\omega}_\lambda}{4\pi}F_{\odot,\lambda}e^{-\tau_\lambda/\mu_0}P_\lambda(\mu,\phi;-\mu_0,\phi_0) \tag{2.28}$$

To solve radiative transfer problems the three parameters: the optical depth τ_λ, the single-scattering albedo $\tilde{\omega}_\lambda$ and the phase function $P_\lambda(\cos\Theta)$ need to be determined. The phase function $P_\lambda(\cos\Theta)$ can be expressed as a series of Legendre polynomials P_l of lth order, which are commonly used in physics due to their mathematical properties. This allows to choose the accuracy of representation needed with the number of polynomials N:

$$P(\cos\Theta) = \sum_{l=0}^{N} \omega_l P_l(\cos\Theta) \tag{2.29}$$

Note that from now on the subscript λ for specific wavelengths will be neglected, to avoid confusion in the following equations. ω_l is the expansion coefficient for $l = 0, 1, ..., N$:

$$\omega_l = \frac{2l+1}{2}\int_{-1}^{1} P(\cos\Theta)P_l(\cos\Theta)d\cos\Theta \tag{2.30}$$

For $l = 0$, $\omega_0 = 1$, while the first order phase function ($l = 1$), for which $P_1(\cos\Theta) = \cos\Theta$, is used to define a commonly used parameter for radiative transfers in the atmospheres, the asymmetry factor g:

$$g \equiv \frac{\omega_1}{3} = \frac{1}{2}\int_{-1}^{1} P(\cos\Theta)\cos\Theta d\cos\Theta \tag{2.31}$$

The asymmetry factor is the first moment of the phase function and describes the propagation of scattered radiation, giving a relative indication of the ratio that is scattered forward. For an isotropic medium, such as Rayleigh scattering, $g = 0$, but g can also increase if the scattering has a more forward directed scattering, e.g. figure 2.5(a):(c), as well as become negative for cases where backward scattering dominates. Combining equation (2.24) and equation (2.29), the phase function becomes:

[1] To be consistent with the formalism in equation (2.25) and equation (2.26), $I_\lambda(\tau_\lambda;\mu;\phi)$ included a factor $\beta_{e,\lambda}$, while the optical depth actually had become $\tau_\lambda \equiv \int_z^\infty \beta_{e,\lambda}dz'$, instead of the expression in equation (2.21).

$$P(\mu,\phi;\mu',\phi') = \sum_{l=0}^{N} \omega_l P_l[\mu\mu' + (1-\mu^2)^{1/2}(1-\mu'^2)^{1/2}\cos(\phi'-\phi)] \tag{2.32}$$

Legendre polynomials have many mathematical and geometrical properties. In this case it becomes possible to decompose equation (2.32) into spherical harmonics with the addition theorem[2]:

$$P(\mu,\phi;\mu',\phi') = \sum_{m=0}^{N}\sum_{l=0}^{N} \omega_l^m P_l^m(\mu) P_l^m(\mu') \cos m(\phi'-\phi) \tag{2.33}$$

where

$$\omega_l^m = (2-\delta_{0,m})\omega_l \frac{(l-m)!}{(l+m)!} \tag{2.34}$$

for $l = m, ..., N$ with $0 \leq m \leq N$, as well as P_l^m as the associated Legendre polynomials and $\delta_{0,m}$ as the Dirac Delta function which is either 1 for $m = 0$ or zero otherwise.

Likewise, the intensity $I(\tau;\mu,\phi)$ can be expressed with spherical harmonics as:

$$I(\tau;\mu,\phi) = \sum_{m=0}^{N} I^m(\tau,\mu) \cos m(\phi'-\phi) \tag{2.35}$$

Inserting the spherical harmonic expressions for $P(\mu,\phi;\mu',\phi')$ and $I(\tau;\mu,\phi)$ from equation (2.33) and equation (2.35) into the RTE in equation (2.28) and taking advantage of the orthogonality of the associated Legendre polynomials, the RTE splits into $(N+1)$ independent equations of the form:

$$\begin{aligned}\mu\frac{dI^m(\tau,\mu)}{d\tau} = I^m(\tau,\mu) - (1-\delta_{0,m})\frac{\tilde{\omega}}{4}\sum_{l=m}^{N}\omega_l^m P_l^m(\mu)\int_{-1}^{1} P_l^m(\mu') I^m(\tau,\mu') d\mu' \\ -\frac{\tilde{\omega}}{4\pi}\sum_{l=m}^{N}\omega_l^m P_l^m(\mu) P_l^m(-\mu_0) F_\odot e^{-\tau/\mu_0}\end{aligned} \tag{2.36}$$

Each of the independent equations can be solved to determine I^m, which then can be used to calculate the (monochromatic) intensity I with equation (2.35).

For the case $m = 0$, the intensity I in equation (2.35) becomes independent of the azimuthal angle ϕ. This represents a medium that is homogeneous in the horizontal plane, which is a good approximation for many atmospheric models. The phase function for this case becomes:

$$P(\mu,\mu') = \sum_{l=0}^{N} \omega_l P_l(\mu) P_l(\mu') \tag{2.37}$$

[2] For a detailed description of the addition theorem see e.g. Apendix E in [Liou, 2002].

While the RTE takes the form:

$$\mu\frac{dI(\tau,\mu)}{d\tau} = I(\tau,\mu) - \frac{\tilde{\omega}}{2}\int_{-1}^{1} I(\tau,\mu')P(\mu,\mu')d\mu' \\ -\frac{\tilde{\omega}}{4\pi}F_{\odot}P(\mu,-\mu_0)e^{-\tau/\mu_0} \tag{2.38}$$

The upward $F_{dif}^{\uparrow}$ and downward $F_{dif}^{\downarrow}$ diffuse monochromatic flux densities can then be defined with $I(\tau,\mu)$:

$$F_{dif}^{\uparrow\downarrow}(\tau) = 2\pi\int_{0}^{\pm 1} I(\tau,\mu)\mu d\mu \tag{2.39}$$

where the positive integral limit corresponds to the upward flux, while the negative is used for the downward flux. Equation (2.39) shows the diffuse part of the solar flux, but does not take the direct, non-scattered solar radiation into account. The direct flux density can be defined with the Beer-Bouguer-Lambert law as derived in equation (2.16), so that:

$$F_{dir}^{\downarrow}(\tau) = \mu_0 F_{\odot}e^{-\tau/\mu_0} \tag{2.40}$$

Naturally there can not be a direct upward density flux from the solar radiation on Earth, as all upward directed solar radiation has been at least been scattered once by the atmosphere or surface.

Combining the diffusive and direct part of the solar flux densities the upward and downward fluxes become:

$$F^{\uparrow}(\tau) = F_{dif}^{\uparrow}(\tau) = 2\pi\int_{0}^{1} I(\tau,\mu)\mu d\mu \tag{2.41}$$

$$F^{\downarrow}(\tau) = F_{dif}^{\downarrow}(\tau) + F_{dir}^{\downarrow}(\tau) = 2\pi\int_{0}^{-1} I(\tau,\mu)\mu d\mu + \mu_0 F_{\odot}e^{-\tau/\mu_0} \tag{2.42}$$

The (monochromatic) net flux density, i.e. the difference between upward and downward fluxes is then:

$$F(\tau) = F^{\downarrow}(\tau) - F^{\uparrow}(\tau) \tag{2.43}$$

To compute the total solar net flux density F for the whole shortwave spectrum, one would need to integrate $F(\tau)$ over all wavelengths in the spectrum. The net flux can only be either zero for an atmosphere in equilibrium, or take a positive value, which indicates warming of the system. A warming due to the divergence of the solar flux can be quantified as the solar heating rate:

$$\frac{\partial T}{\partial t} = -\frac{1}{\rho c_p}\frac{\partial F}{\partial z} \tag{2.44}$$

where T is the temperature, t is the time, ρ is the density of air in the layer and c_p is the specific heat at constant pressure.

2.2 Radiation in atmospheric models

From the expression of the heating rate in equation (2.44) it becomes apparent that the solar radiation fluxes are not only important for the surface heating through the day and night cycle but also for the vertical thermal structure of the atmosphere. Even though the RTE can be simplified through general assumptions such as the plane-parallel atmosphere and horizontal homogeneous approximation, as shown in the previous section, the task of solving the general RTE (2.38) is still complex and computationally heavy for weather and climate models.

The three previously introduced radiative variables: the optical thickness τ, the single scattering albedo $\tilde{\omega}$ and the asymmetry factor g, are important for the algorithms used to compute radiative processes in atmospheric models, which will be focused on in this section.

Atmospheric dynamics can be described and calculated through the Navier-Stokes equation, the thermodynamic equation, the continuity equation and equation of state. However, the above mentioned radiative variables are not part of the Eulers equations solved by the NWP model, which will be further discussed in section 2.2.3, when spectral integration and bands are introduced.

There are a several processes in the atmosphere that are too complex, e.g. small scale mechanisms, that need to be approximated in parameterization schemes in NWP models, as shown in figure 2.6.

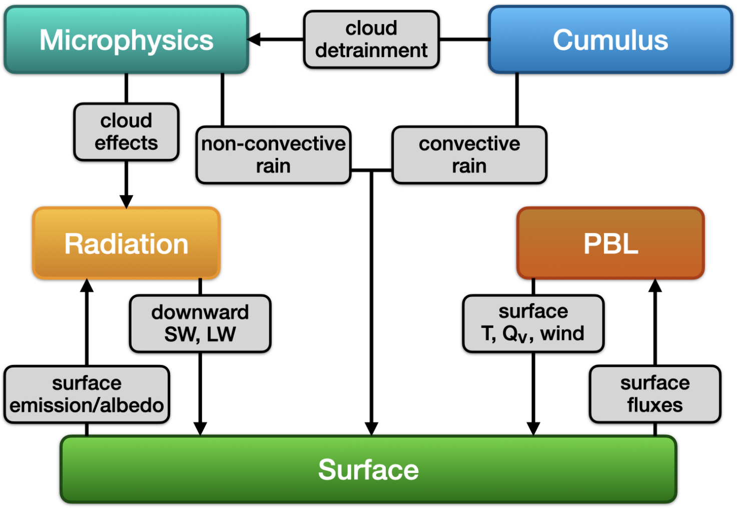


Figure 2.6: Illustration showing an example of the interaction between parameterization schemes in the WRF model

From figure 2.6 it is evident that the different parameterization schemes interact with one another. While the calculated radiation fluxes are important for the heating handled by the Land-Surface scheme, the radiation parameterization scheme depends on information about the surface, e.g. the albedo for the solar spectrum, and clouds from the other schemes as well.

A benefit of treating the radiative processes in a separate parameterization scheme is that the radiation fluxes are not computed at every model time step, which normally is only a few seconds to minutes long in a NWP model. The radiation scheme is typically called once every hour in a forecasting model, as e.g. the ecRad code is called hourly by the ECMWF model [Hogan and Bozzo, 2018].

Section 2.2.1 will focus on some of the commonly used approximate solutions for the RTE used in NWP models, i.e. the two-stream method and its variations.
The issue of non-homogeneity in the vertical direction will be presented in section 2.2.2, when the vertical integration will be described.
While the treatment of gases, aerosols and clouds will be presented together with the spectral bands and spectral integration in section 2.2.3, section 2.2.4 will describe the effect and calculation of clouds in more detail.
Since the WRF model supports different radiation parameterization schemes, section 3.3 will focus on the selected ones used in this study and present those with the methods that will be introduced in the following sections.

2.2.1 Two-stream-method

The computation of the radiative fluxes in NWP models requires that the RTE in equation (2.38) can be solved analytically. This means that the integral in the second term on the right hand side must be replaced by a finite sum. For this the Discrete-ordinates method was developed [Chandrasekhar, 1950], which is the starting point for the the two-stream method as well as four-stream method.
The concept of the Discrete-ordinates method is to use the Gauss' formula to substitute the integral with a sum over a finite number of quadrature points:

$$\int_{-1}^{1} f(\mu)d\mu \approx \sum_{j=-n}^{n} a_j f(\mu_j) \tag{2.45}$$

a_j are weights defined as:

$$a_j = \frac{1}{P'_{2n}(\mu_j)} \int_{-1}^{1} \frac{P_{2n}(\mu)}{\mu - \mu_j} d\mu \tag{2.46}$$

where μ_j are the zeros of the polynomials $P_{2n}(\mu)$ and the prime of $P'_{2n}(\mu_j)$ denotes the derivative with respect to μ_j.
Using (2.45) we can write (2.38), similarly to equation (2.36), as:

$$\begin{aligned} \mu_i \frac{dI(\tau, \mu_i)}{d\tau} &= I(\tau, \mu_i) - \frac{\tilde{\omega}}{2} \sum_{l=0}^{N} \omega_l P_l(\mu_i) \sum_{j=-n}^{n} a_j P_l(\mu_j) I(\tau, \mu_j) \\ &- \frac{\tilde{\omega}}{4\pi} F_{\odot} \left[\sum_{l=0}^{N} (-1)^l \omega_l P_l(\mu_i) P_l(\mu_0) \right] e^{-\mu_0/\tau}, \quad \text{for } i = -n, ..., n \end{aligned} \tag{2.47}$$

This is a general representation of multiple radiation streams, i.e. radiation beams propagating into the $\mu_i(-n, n)$ directions. In principle any (even) number of streams can be considered, for radiative transfers usually two or four streams are chosen [Liou, 1974]. For this general multi-stream method the following relations apply:

$$a_{-j} = a_j, \qquad \mu_{-j} = -\mu_j, \qquad \sum_{j=-n}^{n} a_j = 2 \tag{2.48}$$

In the case of two streams, i.e. $n = 1$, $N = 1$, $j = -1$ and 1, these imply:

$$\mu_1 = \frac{1}{\sqrt{3}}, \qquad a_1 = a_{-1} = 1 \tag{2.49}$$

Denoting the intensities $I^{\uparrow} = I(\tau, \mu_1)$ and $I^{\downarrow} = I(\tau, -\mu_1)$, one gets the following two equations for the two-stream approximation from (2.47) :

$$\mu_1 \frac{dI^{\uparrow}}{d\tau} = I^{\uparrow} - \tilde{\omega}(1-b)I^{\uparrow} - \tilde{\omega}bI^{\downarrow} - S^{-}e^{-\tau/\mu_0} \tag{2.50}$$

$$-\mu_1 \frac{dI^{\downarrow}}{d\tau} = I^{\downarrow} - \tilde{\omega}(1-b)I^{\downarrow} - \tilde{\omega}bI^{\uparrow} - S^{+}e^{-\tau/\mu_0} \tag{2.51}$$

where g is the previously introduced asymmetry factor from equation (2.31), which is zero for isotropic (Rayleigh) scattering:

$$g \equiv \frac{\omega_1}{3} = \frac{1}{2}\int_{-1}^{1} P(\cos\Theta)\cos\Theta d\cos\Theta = \langle\cos\Theta\rangle \tag{2.52}$$

and

$$b = \frac{1-g}{2}, \qquad S^{\pm} = \frac{F_{\odot}\tilde{\omega}}{4\pi}(1 \pm 3g\mu_1\mu_0) \tag{2.53}$$

From equation (2.50) and (2.51) it can be seen that the two intensities are interdependent from the third term on the right hand side, which is a representation of multiple scattering.

b and $(1-b)$ can be thought of as fractions of back- and forward-scattering, while $S^{\pm}$ is the direct solar source term.

To finde the solution to those two first-order inhomogeneous differential equations, two boundary conditions are required. For this the diffuse radiation at the surface and the TOA are usually assumed to be zero, which yields the solutions[3]:

$$I^{\uparrow} = I(\tau, \mu_1) = Kve^{k\tau} + Hue^{-k\tau} + \epsilon e^{-\tau/\mu_0} \tag{2.54}$$

$$I^{\downarrow} = I(\tau, -\mu_1) = Kue^{k\tau} + Hve^{-k\tau} + \gamma e^{-\tau/\mu_0} \tag{2.55}$$

where

$$v = \frac{1+a}{2}, \qquad u = \frac{1-a}{2}, \qquad a^2 = \frac{1-\tilde{\omega}}{1-\tilde{\omega}g} \tag{2.56}$$

$$\epsilon = \frac{\alpha+\beta}{2}, \qquad ,\gamma = \frac{\alpha-\beta}{2} \tag{2.57}$$

[3] A more detailed derivation of the following equations can be found in [Liou, 2002]

$$\alpha = \frac{Z_1 \mu_0^2}{1 - \mu_0^2 k^2}, \qquad \beta = \frac{Z_2 \mu_0^2}{1 - \mu_0^2 k^2}, \qquad k^2 = \frac{(1-\tilde{\omega})(1-\tilde{\omega}g)}{\mu_1^2} \tag{2.58}$$

$$Z_1 = -\frac{(1-\tilde{\omega}g)(S^- + S^+)}{\mu_1^2} + \frac{S^- - S^+}{\mu_1 \mu_0}, \quad Z_2 = -\frac{(1-\tilde{\omega}g)(S^- - S^+)}{\mu_1^2} + \frac{S^- + S^+}{\mu_1 \mu_0} \tag{2.59}$$

K and H need to be determined from the diffuse intensity boundary conditions. In the case of no diffuse radiation at the surface and the TOA these two constants become:

$$K = -\frac{\epsilon v e^{\tau_1/\mu_0} - \gamma u e^{-k\tau_1}}{v^2 e^{k\tau_1} - u^2 e^{-k\tau_1}}, \qquad H = -\frac{\epsilon u e^{\tau_1/\mu_0} - \gamma v e^{-k\tau_1}}{v^2 e^{k\tau_1} - u^2 e^{-k\tau_1}} \tag{2.60}$$

From the intensities the diffuse fluxes can be found with equation (2.39):

$$F^{\uparrow} = 2\pi \mu_1 I^{\uparrow}, \qquad F^{\downarrow} = 2\pi \mu_1 I^{\downarrow} \tag{2.61}$$

These solutions are only valid for non-conservative scattering atmospheres, i.e. $\tilde{\omega} < 1$. While solutions for conservative scattering, $\tilde{\omega} = 1$, can be derived from the equations (2.50) and (2.51), values for conservative scattering are in practice satisfied by setting $\tilde{\omega} = 0.99999$ and using the equations for the non-conservative case.

Since the development of the two-stream method there have appeared many similar methods for different applications, which all can be expressed in the same framework as the two-stream approximation [Meador and Weaver, 1980], [Yang et al., 2018], [Zhang et al., 2018].

By integrating the RTE (2.38), the diffuse fluxes can be expressed as:

$$\begin{aligned} \frac{1}{2\pi}\frac{dF^{\uparrow}(\tau)}{d\tau} &= \int_0^1 I(\tau,\mu)d\mu - \frac{\tilde{\omega}}{2}\int_0^1\int_{-1}^1 I(\tau,\mu)P(\mu,\mu')d\mu' d\mu \\ &- \frac{\tilde{\omega}}{4\pi}F_{\odot}e^{-\tau/\mu_0}\int_0^1 P(\mu,-\mu_0)d\mu \end{aligned} \tag{2.62}$$

$$\begin{aligned} \frac{1}{2\pi}\frac{dF^{\downarrow}(\tau)}{d\tau} &= \int_0^1 I(\tau,-\mu)d\mu + \frac{\tilde{\omega}}{2}\int_0^1\int_{-1}^1 I(\tau,\mu')P(-\mu,\mu')d\mu' d\mu \\ &+ \frac{\tilde{\omega}}{4\pi}F_{\odot}e^{-\tau/\mu_0}\int_0^1 P(-\mu,-\mu_0)d\mu \end{aligned} \tag{2.63}$$

The general two-stream approximation can then be written as:

$$\frac{dF^{\uparrow}(\tau)}{d\tau} = \gamma_1 F^{\uparrow}(\tau) - \gamma_2 F^{\downarrow}(\tau) - \gamma_3 \tilde{\omega} F_{\odot} e^{-\tau/\mu_0} \tag{2.64}$$

$$\frac{dF^{\downarrow}(\tau)}{d\tau} = \gamma_1 F^{\downarrow}(\tau) - \gamma_2 F^{\uparrow}(\tau) + (1-\gamma_3)\tilde{\omega} F_{\odot} e^{-\tau/\mu_0} \tag{2.65}$$

From the equations above it can be seen that the differential changes of the diffuse fluxes depend on both the diffuse upward and downward fluxes as well as the direct downward flux. The coefficients γ_1, γ_2 and γ_3 depend on the specific approximation and its assumptions about the intensity and phase function.

There are many different approximation methods, so only the ones relevant for this study, i.e. the ones used by the WRF radiation parameterization schemes, will be presented in the following paragraphs.

For the previously described two-stream approximation only two intensities are considered, traveling in the μ_1 and μ_-1 direction, while the phase function has been expanded in two terms of Legendre polynomials P_{2n}.

Another approach is the Eddington approximation, in which both the intensity and the phase functions get expanded in two polynomial terms.

The corresponding values for the γ_1, γ_2 and γ_3 coefficients for these two variants of the general two-stream approximation can be seen in table 2.1.

Method	γ_1	γ_2	γ_3
Two-stream	$\frac{1-\tilde{\omega}(1+g)/2}{\mu_1}$	$\frac{\tilde{\omega}(1-g)}{2\mu_1}$	$\frac{1-3g\mu_1\mu_0}{2}$
Eddington	$\frac{7-(4+3g)\tilde{\omega}}{4}$	$-\frac{1-(4-3g)\tilde{\omega}}{4}$	$\frac{2-3g\mu_0}{4}$

Table 2.1: Coefficients for the two-stream approximation in equation (2.64) and (2.65)

The solution of the general two-stream method in equation (2.64) and (2.65) is:

$$F^{\uparrow} = vKe^{k\tau} + uHe^{-k\tau} + \epsilon e^{-\tau/\mu_0} \tag{2.66}$$

$$F^{\downarrow} = uKe^{k\tau} + vHe^{-k\tau} + \gamma e^{-\tau/\mu_0} \tag{2.67}$$

where H and K need to be determined by the boundary conditions and:

$$v = \frac{1}{2}\left(1 + \frac{\gamma_1 - \gamma_2}{k}\right), \qquad u = \frac{1}{2}\left(1 - \frac{\gamma_1 - \gamma_2}{k}\right) \tag{2.68}$$

$$k^2 = \gamma_1^2 - \gamma_2^2, \qquad \epsilon = [\gamma_3(1/\mu_0 - \gamma_1) - \gamma_2(1 - \gamma_3)]\mu_0^2\tilde{\omega}F_{\odot} \tag{2.69}$$

$$\gamma = -[(1 - \gamma_3)(1/\mu_0 + \gamma_1) + \gamma_2\gamma_3]\mu_0^2\tilde{\omega}F_{\odot} \tag{2.70}$$

2.2.1.1 δ-Function adjustment

While the two-stream and Eddington methods yield good approximations for radiative transfers in optical thick layers, they are rather inaccurate when the scattering by particles has a strong forward peaked direction, as e.g. it is the case for cloud particles.

To take into account the effect which such large forward peaks have on multiple scattering processes, an adjustment is made to the absorption and scattering.

In practice this is done through the removal of the fraction, f, of the scattered energy inside the forward peak from the radiative variables τ, $\tilde{\omega}$ and g.

Let the apostrophe ' denote the adjusted variables, and the optical thickness τ be defined as the sum of its scattering τ_s and absorption τ_a component. Then the components of the optical thickness can be adjusted as:

$$\tau_s' = (1-f)\tau_s \tag{2.71}$$

$$\tau_a' = \tau_a \tag{2.72}$$

Note that the absorption is not affected by the forward peak. The total adjusted optical thickness becomes therefore:

$$\tau' = \tau_s' + \tau_a' = (1-f)\tau_s + \tau_a = (1-\tilde{\omega}f)\tau \tag{2.73}$$

Similarly, the adjusted single-scattering albedo $\tilde{\omega}'$ and the adjusted asymmetry factor g' can be expressed as:

$$\tilde{\omega}' = \frac{\tau_s'}{\tau'} = \frac{(1-f)\tilde{\omega}}{1-\tilde{\omega}f} \tag{2.74}$$

$$\tau_s'g' = \tau_s g - \tau_s f \qquad \Longleftrightarrow \qquad g' = \frac{g-f}{1-f} \tag{2.75}$$

Finally, f is the same as the second moment phase function as can be derived from equation (2.30) :

$$f = \frac{\omega_2}{5} \tag{2.76}$$

For cloud and aerosol particles the phase function can be expressed through the asymmetry factor g, called the Henyey-Greenstein phase function, which leads to the Henyey-Greenstein approximation $f = g^2$, linking the asymmetry factor to the fraction of the forward scattering.

The combination of this δ-adjustment with the Eddington approximation is called the δ-Eddington approach [Joseph et al., 1976], which is one of the most used methods in atmospheric models.

Another approach is the Practical Improved Flux method (PIFM) [Zdunkowski et al., 1980], [Räisänen, 2002], of which the coefficients are shown alongside the ones of the δ-Eddington method in table 2.2.

Method	γ_1	γ_2	γ_3
δ-Eddington	$\frac{7-(4+3g')\tilde{\omega}'}{4}$	$-\frac{1-(4-3g')\tilde{\omega}'}{4}$	$\frac{2-3g'\mu_0}{4}$
PIFM	$\frac{8-\tilde{\omega}'(5+3g')}{4}$	$\frac{3}{4}(\tilde{\omega}'(1-g'))$	$\frac{2-3g'\mu_0}{4}$

Table 2.2: Coefficients for the two-stream approximation with δ-function adjustment for two different methods, both for which $f = g^2$.

2.2.2 Vertical integration

For all approximations presented in the previous sections only a homogeneous layer was considered. While this is a relatively good assumption for atmospheric layers in the horizontal plane, it is not valid in the vertical direction.
Dividing the Earth's atmosphere into N vertical layers, as done in NWP models, shows that there are large variations in the vertical profile.

Theoretically it is possible to divide the atmosphere into so many thin vertical layers that each of them can be treated as homogeneous, one such layer is depicted in figure 2.7. The RTE can then be solved for each individual layer yielding the corresponding up- and downward fluxes per layer. However, this becomes tedious when the effects of multiple scattering need to be taken into account.

Imagine the radiative feedback mechanism between two layers, each with their own optical properties, represented through their optical thicknesses τ_1 and τ_2, as shown in figure 2.8. An incoming solar beam $(\mu_0 F.)$ will be partly reflected when reaching the first layer (R_1), while the other part is transmitted through the first layer $(\tilde{T}_1)$ towards the second, where it then also will be partly reflected and partly transmitted further. The reflected portion of the firstly transmitted radiation $(R_2\tilde{T}_1)$ travels then upwards back towards the first layer, where it again can either transmit upwards $(\tilde{T}_1^* R_2 \tilde{T}_1)$ or be again reflected downwards $(\tilde{R}_1^* R_2 \tilde{T}_1)$. This feedback mechanism can go on infinitely.
Thus, while these calculations are conceptually correct, NWP models require a method which includes multiple scattering, but is easier to resolve.
A commonly used vertical integration method of radiative fluxes in radiation parameterizations is the adding method, often coupled with the (δ-)2-stream method.

We will get back to the process of multiple scattering shown in figure 2.8, first let us consider a single homogeneous atmospheric layer as shown in figure 2.7.

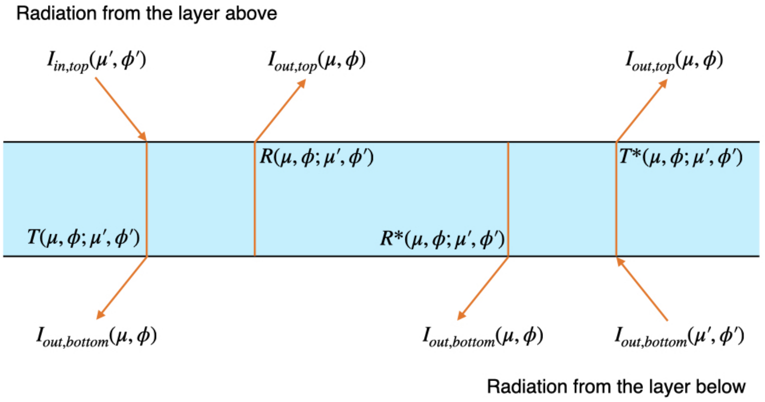

Radiation from the layer above
$I_{in,top}(\mu',\phi')$
$I_{out,top}(\mu,\phi)$
$I_{out,top}(\mu,\phi)$
$R(\mu,\phi;\mu',\phi')$
$T^*(\mu,\phi;\mu',\phi')$
$T(\mu,\phi;\mu',\phi')$
$R^*(\mu,\phi;\mu',\phi')$
$I_{out,bottom}(\mu,\phi)$
$I_{out,bottom}(\mu,\phi)$
$I_{out,bottom}(\mu',\phi')$
Radiation from the layer below

Figure 2.7: Schematic of incident radiation from above (left) and below (right) at a single atmospheric layer. Figure adapted from [Liou, 2002].

For this layer the incoming radiation from above is denoted $I_{in,top}$, while all radiation incident on the layer from below is called $I_{in,bottom}$.

From the intensity arriving from above $I_{in,top}$, the part that is transmitted throughout the layer $I_{out,bottom}$ is described by the transmissivity $\tilde{T}$, while the portion that is reflected $I_{out,top}$, can be described with the reflectivity R. Similarly, the intensity coming from below $I_{in,bottom}$ gets transmitted and reflected as $I_{out,top}$ and $I_{out,bottom}$, as described by the transmissivity $\tilde{T}^*$ and reflectivity R^*. The superscript * is used to denote values for radiation traveling upwards from below.
The reflectivities and transmissivities R, R^*, $\tilde{T}$ and $\tilde{T}^*$ can be calculated from the RTE in equation (2.38) by using the incident beam intensity $I_{in,top}$ and $I_{in,bottom}$, respectively.

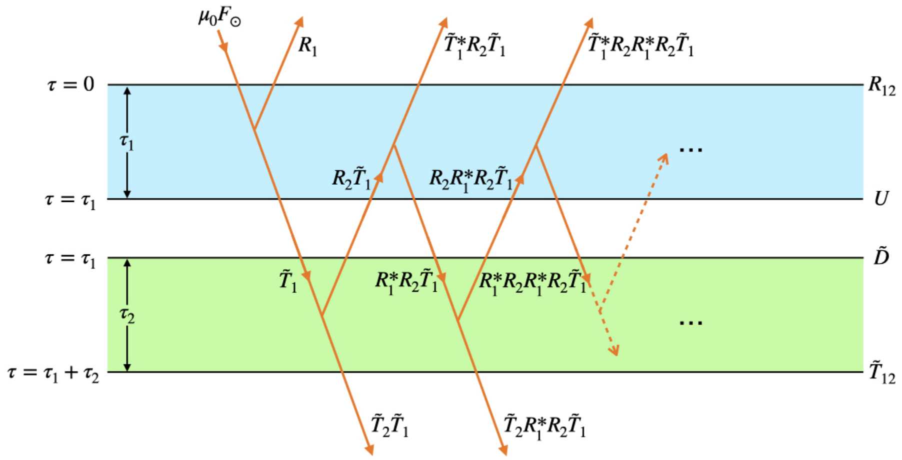


Figure 2.8: Illustration of two layers and terms used in the adding method. The layers are depicted individually with their optical thickness τ_1 and τ_2, reflection and transmission function R_1, R_2, $\tilde{T}_1$ and $\tilde{T}_2$, respectively. The superscript * denotes radiation traveling upwards from below. Figure adapted from [Liou, 2002].

Let us now again consider the case of two layers as depicted in figure 2.8. Here $\tilde{T}$ represents the total transmission, i.e. both the direct and diffuse portion, and R the reflection at one layer. The single-digit subscripts 1 and 2 denote which layer the reflection and transmission belong to. Likewise, layer 1 and 2 have an optical thickness defined as τ_1 and τ_2.
Additionally, all upward reflected radiation from the interface between the two layers will be defined as U, while all transmission traveling downwards through the interface will be denoted $\tilde{D}$.
Finally, R_{12} is defined as the total reflection at the top of layer 1, due to all multiple scattering between the two layers, while $\tilde{T}_{12}$ is the combined transmission at the bottom of layer 2, due to the same scattering processes.

Following the multiple scattering shown in the figure and the discussed feedback mechanisms from before, one can begin to write:

$$R_{12} = R_1 + \tilde{T}_1^* R_2 \tilde{T}_1 + \tilde{T}_1^* R_2 R_1^* R_2 \tilde{T}_1 + \tilde{T}_1^* R_2 R_1^* R_2 R_1^* R_2 \tilde{T}_1 + ... \tag{2.77}$$

$$\tilde{T}_{12} = \tilde{T}_2 \tilde{T}_1 + \tilde{T}_2 R_1^* R_2 \tilde{T}_1 + \tilde{T}_2 R_1^* R_2 R_1^* R_2 \tilde{T}_1 + ... \tag{2.78}$$

$$U = R_2\tilde{T}_1 + R_2R_1^*R_2\tilde{T}_1 + R_2R_1^*R_2R_1^*R_2\tilde{T}_1 + ... \tag{2.79}$$

$$\tilde{D} = \tilde{T}_1 + R_1^*R_2\tilde{T}_1 + R_1^*R_2R_1^*R_2\tilde{T}_1 + ... \tag{2.80}$$

Note how the series' converge, as e.g. for R_{12}:

$$\begin{aligned} R_{12} &= R_1 + \tilde{T}_1^*R_2\tilde{T}_1 + \tilde{T}_1^*R_2R_1^*R_2\tilde{T}_1 + \tilde{T}_1^*R_2R_1^*R_2R_1^*R_2\tilde{T}_1 + ... \\ &= R_1 + \tilde{T}_1^*R_2[1 + R_1^*R_2 + (R_1^*R_2)^2 + ...]\tilde{T}_1 \\ &= R_1 + \tilde{T}_1^*R_2(1 - R_1^*R_2)^{-1}\tilde{T}_1 \end{aligned} \tag{2.81}$$

Therefore the previous expressions can be written as:

$$R_{12} = R_1 + \tilde{T}_1^*R_2(1 - R_1^*R_2)^{-1}\tilde{T}_1 \tag{2.82}$$

$$\tilde{T}_{12} = \tilde{T}_2(1 - R_1^*R_2)^{-1}\tilde{T}_1 \tag{2.83}$$

$$U = R_2(1 - R_1^*R_2)^{-1}\tilde{T}_1 \tag{2.84}$$

$$\tilde{D} = (1 - R_1^*R_2)^{-1}\tilde{T}_1 \tag{2.85}$$

Afterwards the following relationships can be deduced from the equations above:

$$R_{12} = R_1 + \tilde{T}_1^*U \tag{2.86}$$

$$\tilde{T}_{12} = \tilde{T}_2\tilde{D} \tag{2.87}$$

$$U = R_2\tilde{D} \tag{2.88}$$

From the expression for R_{12} in equation (2.86) it becomes apparent that the total combined reflection due to multiple scattering between both layers is the sum of the reflected radiation of the first layer (R_1) and the upward transmitted radiation from the multiple scattering throughout the interface at U. Meanwhile, the total transmission of both layers $\tilde{T}_{12}$ is a result of the downward transmitted radiation through layer 2 at $\tilde{D}$.

$\tilde{T}$ denotes the total transmission, both direct and diffuse, as stated earlier. Using the Beer-Bouguer-Lambert Law from equation (2.16) in the same manner as to define the direct flux in equation (2.40), the total transmission can be divided into its direct component $e^{-\tau/\mu'}$ and diffuse portion T:

$$\tilde{T} = T + e^{-\tau/\mu'} \tag{2.89}$$

where for direct solar radiation $\mu' = \mu_0$ and for a beam traveling in the μ direction $\mu' = \mu$. Additionally, it proves useful to define an operator S of the form:

$$S = R_1^*R_2(1 - R_1^*R_2)^{-1} \text{ so that } (1 - R_1^*R_2)^{-1} = 1 + S \tag{2.90}$$

$\tilde{D}$ and $\tilde{T}_{12}$ can then be decomposed into their direct and diffuse parts, where T_1, T_2 and D are diffuse components only:

$$\begin{aligned} \tilde{D} &= D + e^{-\tau_1/\mu_0} \\ &= (1 + S)T_1 + Se^{-\tau_1/\mu_0} + e^{-\tau_1/\mu_0} \end{aligned} \tag{2.91}$$

$$\begin{aligned}\tilde{T}_{12} &= (T_2 + e^{-\tau_2/\mu_0})(D + e^{-\tau_1/\mu_0}) \\ &= e^{-\tau_2/\mu_0}D + T_2 e^{-\tau_1/\mu_0} + T_2 D + \exp\left[-\left(\frac{\tau_1}{\mu_0} + \frac{\tau_2}{\mu_0}\right)\right]\delta(\mu - \mu_0)\end{aligned} \tag{2.92}$$

The total diffuse transmission and reflection of both layers may be found with a set of iterative equations, which for the radiation coming from above take the following form for T_{12} and R_{12}:

$$Q = R_1^* R_2 \tag{2.93}$$

$$S = Q(1 - Q)^{-1} \tag{2.94}$$

$$D = T_1 + ST_1 + Se^{-\tau_1/\mu_0} \tag{2.95}$$

$$U = R_2 D + R_2 e^{-\tau_1/\mu_0} \tag{2.96}$$

$$T_{12} = e^{-\tau_2/\mu}D + T_2 e^{-\tau_1/\mu_0} + T_2 D \tag{2.97}$$

$$R_{12} = R_1 + e^{-\tau_1/\mu}U + T_1^* U \tag{2.98}$$

For the radiation travelling upwards from below, T_{12}^* and R_{12}^* can be computed with:

$$Q = R_2 R_1^* \tag{2.99}$$

$$S = Q(1 - Q)^{-1} \tag{2.100}$$

$$U = T_2^* + ST_2^* + Se^{-\tau_2/\mu'} \tag{2.101}$$

$$D = R_1^* U + R_1^* e^{-\tau_2/\mu'} \tag{2.102}$$

$$T_{12}^* = e^{-\tau_1/\mu}U + T_1^* e^{-\tau_2/\mu'} + T_1^* U \tag{2.103}$$

$$R_{12}^* = R_2^* + e^{-\tau_2/\mu}D + T_2 D \tag{2.104}$$

From this example it can be seen that the adding method is an efficient approach to determine the radiative fluxes between two layers, e.g. at the surface or the TOA.

As aforementioned, the atmosphere is divided into several vertical layers in NWP models. A number N layers is chosen, for which each layer is assumed to be homogeneous and is characterized by its own set of radiative variables (τ, $\tilde{\omega}$, g).

For homogeneous layers the transmission and reflection from above or below are identical. Thus we have for each l'th layer $T_l = T_l^*$ and $R_l = R_l^*$ for $l = 1, 2, ..., N$. Moreover, the surface is defined as an additional layer $N + 1$ with no transmission, $T_{N+1} = 0$, and the surface albedo as R_{N+1}.

An Illustration of such a vertical structure of the atmosphere is shown in figure 2.9. Note that $l = 1$ is the layer at the top of the atmosphere, while $l = N + 1$ is the surface layer.

As depicted in the figure the layers are added downward one by one from the TOA to the layer l to compute $T_{1,l}$ and $R_{1,l}$ for $l = 2, ..., (N + 1)$, as well as $T_{1,l}^*$ and $R_{1,l}^*$ for $l = 2, ..., N$. Similarly, the layers added upwards from the surface are used to obtain $T_{l+1,N+1}$ and $R_{l+1,N+1}$ for $l = (N - 1), ..., 1$.

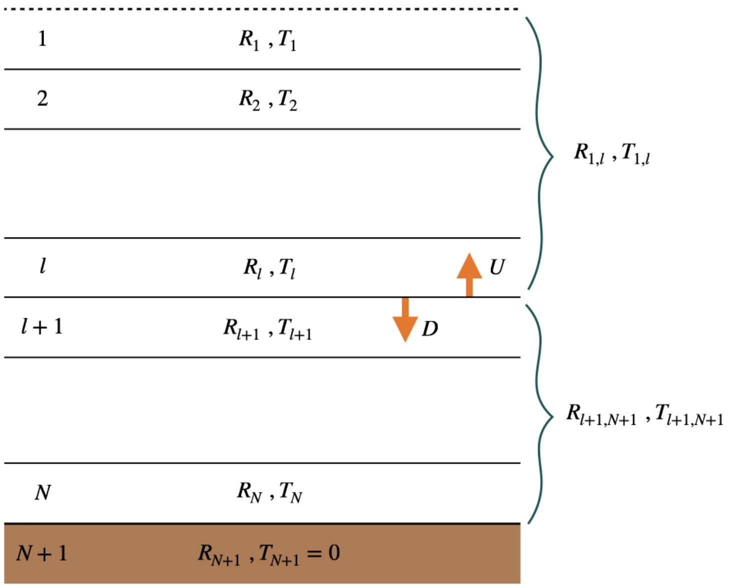


Figure 2.9: Depiction of the vertical layer structure and notation for the internal intensities in an atmosphere of the adding method. Figure adapted from [Liou, 2002].

Considering the layers $(1,\ l)$ and $(l+1,\ N+1)$, the adding method can be used to determine D and U:

$$D = T_{1,l} + ST_{1,l} + S\exp(-\tau_{1,l}/\mu_0) \tag{2.105}$$

$$U = R_{l+1,N+1}D + R_{l+1,N+1}\exp(-\tau_{1,l}/\mu_0) \tag{2.106}$$

where $\tau_{1,l}$ is the optical thickness from the TOA to the bottom of the l'th layer and S and Q are defined as:

$$S = Q(1-Q)^{-1} \tag{2.107}$$

$$Q = R^*_{l,1}R_{l+1,N+1} \tag{2.108}$$

The fluxes at the interface between layer l and $l+1$, taking into account all the scattering in the layers above and below, then become:

$$F^{\uparrow} = \mu_0 F_{\odot}\left(2\int_0^1 U(\mu,\mu_0)\mu\ d\mu\right) \tag{2.109}$$

$$F^{\downarrow}_{dif} = \mu_0 F_{\odot}\left(2\int_0^1 D(\mu,\mu_0)\mu\ d\mu\right) \tag{2.110}$$

$$F^{\downarrow}_{dir} = \mu_0 F_{\odot}\exp(-\tau_{1,l}/\mu_0) \tag{2.111}$$

$$F = (F^{\downarrow}_{dif} + F^{\downarrow}_{dir}) - F^{\uparrow} \tag{2.112}$$

where $F^{\uparrow}$ is the upward flux, $F^{\downarrow}_{dif}$ is the diffuse downward flux, $F^{\downarrow}_{dir}$ is the direct solar downward flux and F is the net flux.

2.2.3 Spectral bands

As previously mentioned the radiative variables (τ, $\tilde{\omega}$, etc.) are not part of the governing equations solved by NWP models. Thus they need to be specified either with prognostic and diagnosed variables, e.g. temperature, pressure, mixing ratios, etc.), in the NWP model, or through look-up tables, which are static data sets with e.g. information about gases such as carbon dioxide or ozone.

In section 2.1 it was shown that different gases absorb and interact with different wavelengths. Recall that the total solar flux F can be defined as an integral of all the monochromatic fluxes F_λ for each wavelength in the solar spectrum as:

$$F = \int_{\lambda_{solarmin}}^{\lambda_{solarmax}} F_\lambda d\lambda \tag{2.113}$$

Here the exact definition of the lower and upper wavelength boundaries of the solar wave spectrum depends on the individual radiation scheme. In general, radiation parameterization schemes define the start of the solar spectrum in the ultraviolet (UV) region (∼200 nm), while the ending boundary is choosen from a wider range, either closer to the near-infrared (NIR) range (∼4.000 nm) or even stretching into the thermal-IR range (∼10.000 nm).

Many radiation schemes divide the whole wave spectrum into spectral subdivisions, i.e. spectral bands. For each of these bands the physical contributions due to e.g. different gases are handled separately. The resulting averaged fluxes for each band $F_{\overline{\lambda}}$ can afterwards be summed up together for all bands to form the total flux:

$$F = \sum_{i=1}^{b} F_{\overline{\lambda},i} \Delta w_i \tag{2.114}$$

where b is the number of bands and Δw_i is the fractional solar flux for the i'th band. For shortwave radiation schemes common numbers of spectral bands are $10 \sim 15$. However there are also schemes with more, or less bands, as well as broadband integration schemes, i.e. schemes with only a single band.

Regardless of the number of spectral bands, the contribution of gases, as well as clouds and aerosols, need to be considered for each atmospheric layer. As seen before in figure 2.1, are there some gases that are more important for the absorption and scattering in the atmosphere than others. The most important gases in context of absorption in the atmosphere, such as water vapor and ozone, are therefore parameterized with greater detail than minor gases.

The contributing gases for an atmospheric layer can be treated independently, with each their own optical thickness τ_{gas}, which can be used to define the total absorption optical thickness τ_{ab} as a sum of all contributors:

$$\tau_{ab} \equiv \tau_{H_2O} + \tau_{O_3} + \tau_{CO_2} + \tau_{O_2} + O(\tau) \tag{2.115}$$

where τ_{H_2O}, τ_{O_3}, τ_{CO_2} and τ_{O_2} are the optical thicknesses of each gas, while $O(\tau)$ describes the contribution of minor gases.

Defining an atmospheric layer that stretches between heights z_1 and z_2, with $z_1 < z_2$, the optical thickness for a contributor is defined as per equation (2.21):

$$\tau_{gas} = \int_{z_1}^{z_2} k\rho_{gas} r dz = \int_{z_1}^{z_2} kq_{gas}\rho_d r dz \tag{2.116}$$

where for the last transformation the density ρ_{gas} is expressed in terms of the mixing ratio q_{gas} and the density of dry air ρ_d.
In the absence of clouds and aerosols, the total optical depth τ due to absorption (τ_{ab}) and scattering processes (τ_{sc}) is:

$$\tau = \tau_{ab} + \tau_{sc} \tag{2.117}$$

Taking the contribution of clouds (τ_{cld}) and aerosols (τ_{aer}) into account, the total optical thickness of an atmospheric layer can therefore be expressed as:

$$\tau = \tau_{ab} + \tau_{sc} + \tau_{cld} + \tau_{aer} \tag{2.118}$$

For each optical thickness, the single scattering albedo and asymmetry factor can be calculated for the gases as well as aerosols and clouds.

To do this the monochromatic absorption coefficient k_λ must be evaluated at each layer, for all wavelengths. It is reasonable to assume that the absorption coefficient within each layer is constant, when the layer has a constant pressure and temperature. However, a pure monochromatic absorption is not observed in the real atmosphere, as there are e.g. collisions between molecules, which lead to broadening of spectral lines. To take the pressure broadening, which turns out to follow the Lorentz profile[4], into account, the monochromatic absorption coefficient k_λ can be defined as:

$$k_\lambda = Sf(\nu - \nu_0) = \frac{S}{\pi}\frac{\alpha}{(\nu - \nu_0)^2 + \alpha^2} \tag{2.119}$$

where $\nu = \frac{1}{\lambda}$ is the wavenumber, $f(\nu - \nu_0)$ is the line shape factor following the Lorentz profile and S is the line strength defined as:

$$S = \int_{-\infty}^{\infty} kd\nu \tag{2.120}$$

α is the line half-width at the half-maximum and works as a scaling function depending on pressure and temperature:

$$\alpha(p,T) = \alpha_0 \left(\frac{p}{p_0}\right)\left(\frac{T_0}{T}\right)^n \tag{2.121}$$

where the reference pressure p_0 and temperature T_0 are usually set to 1013 hPa and 273 K for which the width at standard pressure α_0 is defined. n is an index in the range 0.5 to 1, depending on the molecule.

[4] For a detailed description on line and pressure broadening see e.g. chapter 1.3.2 in [Liou, 2002]

To evaluate the total optical thickness of an atmospheric layer with N gases, for one wavenumber ν (one wavelength λ) along the path length $u = \int \rho(z)dz$ one needs to calculate:

$$\tau_\nu = \sum_{j=1}^{N} \tau_{\nu,j} = \int_u \sum_{j=1}^{N} k_{\nu,j}(u)du \tag{2.122}$$

where $j = 1, 2, ..., N$ denotes the absorption line. The absorption coefficient can be written as a sum of the line strength and shape factor of all absorption lines as well:

$$k_\nu(p, T) = \sum_{j=1}^{N} S_j(T) f_{\nu,j}(p, T) \tag{2.123}$$

To calculate each individual absorption line j, it is necessary to compute the absorption coefficient k_ν at intervals which are smaller than the line half-width. Computing each line like this is called the line-by-line integration, and while this is the most precise method, it is also the computational heaviest. Since this method is not applicable for NWP models used to make weather forecasts, some simplifications need to be made in the radiation parameterization.

One idea is the division of the spectrum into a few spectral bands, for which the absorption coefficient is held constant for an interval of wavelengths selected based on statistics. However, for gases with many different absorption lines, such as seen earlier for e.g. carbon dioxide and water vapor in figure 2.1, this band approach is a poor representation of the real atmospheric absorption.

2.2.3.1 (Correlated) k-distribution

A common approach is the k-distribution method, which is a good compromise between accuracy and faster computation than the line-by-line method. The k-distribution arranges the spectral transmittances T together based on the absorption coefficient k_ν, since the transmittances do not depend on the order of k values in a given spectral interval. This means that the integration over the wavenumbers can be replaced by an integration in the k-space so that:

$$T_{\overline{\nu}}(u) = \int_{\Delta\nu} e^{-k_\nu u} \frac{d\nu}{\Delta\nu} = \int_0^\infty e^{-ku} f(k)dk \tag{2.124}$$

where $f(k)$ is the normalized probability distribution for k_ν in the interval $\Delta\nu$, where its minimum and maximum values have been set to $k_{min} \to 0$ and $k_{max} \to 1$, respectively, as well as $\int_0^\infty f(k)dk = 1$.

Note that equation (2.124) shows that the function $f(k)$ is just the inverse of the Laplace transformation, L^{-1}, of the spectral transmittance:

$$f(k) = L^{-1}(T_{\overline{\nu}}(u)) \tag{2.125}$$

Defining a cumulative probability function $g(k)$ with $g(0) = 0$,$g(k \to \infty)$ and $dg(k) = f(k)dk$ as:

$$g(k) = \int_0^k f(k)dk \tag{2.126}$$

Makes it possible to express the spectral transmittance as:

$$T_{\overline{\nu}}(u) = \int_0^1 e^{-k(g)u} dg \cong \sum_{j=1}^{M} e^{-k(g_j)u} \Delta g_j \tag{2.127}$$

Note that while $g(k)$ is a smooth function in the space of k, $k(g)$ is a smooth function in the space of g. Therefore the integral in the g-space can be rewritten as a finite sum, replacing the integral over the wavenumbers from equation (2.124).

The theory behind the k-distribution method assumed that the absorption coefficient k_ν is constant. For inhomogenous atmospheres, where the absorption coefficient varies with pressure and temperature as described in (2.123), this is not true. A variant of the k-distribution method applicable to inhomogeneous atmospheres is the correlated k-distribution.

The concept of this method is, that the vertical variations are accounted for through an assumption of correlation between absorption coefficients at different temperatures and pressures. For the correlated k-distribution the spectral transmittance can be expressed as:

$$T_{\overline{\nu}}(u) \cong \int_0^1 \exp\left[-\sum_i k_i(g)\Delta u_i\right] dg \tag{2.128}$$

2.2.4 Clouds

Clouds cover a large portion of the Earth's atmosphere and are the contributor with the largest influence on radiative transfers. There exist several different types of clouds (cumulonimbus, stratus, cirrus, etc.) that vary in form, size and composition. The effect of clouds on radiative transfers depends on the individual cloud's optical, geometrical and physical structure, resulting in a wide range of optical thicknesses for different clouds.

Clouds are composed of many different kind of particles, but in most radiation parameterizations the two cloud particle categories of water droplets and ice crystals are considered.

Water droplets and ice crystals are treated separately due to their difference in structure and refraction indices. Ice crystals are usually bigger than water droplets and their structure is more complicated than water droplets, which are treated as spherical droplets.

Clouds can consist of different particles, which also differ in size, which effects how opaque the cloud appears.

Consider a cloud only consisting of water droplets. One can define the mean effective radius a_e, which is a measure of the droplet size distribution inside the cloud:

$$a_e = \int a \cdot \pi a^2 n(a) da \bigg/ \int \pi a^2 n(a) da \tag{2.129}$$

where a is the radius and $n(a)$ is the actual droplet size distribution. The mean effective radius is the mean radius weighted by the droplet cross section, which means that a_e includes the scattering properties of spherical droplets.
It turns out that solar radiative transfers are mainly depended on this mean effective radius, rather than the actual droplet size distribution [Liou, 2002].
The amount of liquid water inside a cloud is called the liquid water content (LWC), which for spherical droplets is defined as:

$$\mathrm{LWC} = \frac{4\pi}{3}\rho_l \int a^3 n(a) da \tag{2.130}$$

where ρ_l is the density of water. For a cloud of thickness Δz, the amount of vertically integrated liquid water is called the liquid water path (LWP), which then is: LWP = LWC $\cdot$ Δz. The optical thickness is defined as:

$$\tau = \Delta z \cdot \int Q_e \pi a^2 n(a) da \tag{2.131}$$

where Q_e is called the efficiency factor for extinction, which is a function of the wavelength, droplet radius and refractive index. For cloud droplets and visible wavelengths $Q_e \cong 2$. Combining equations (2.129), (2.130) and (2.131) yields the relation:

$$a_e \cong \frac{3}{2\rho_l}\mathrm{LWP}/\tau \tag{2.132}$$

which is an important relationship between the droplet size, optical thickness and LWP in the cloud. Consider two clouds with the same LWP, equation (2.132) shows that the cloud with the smaller droplet (small a_e) would then have a larger optical thickness τ. The cloud with the larger optical thickness will appear more opaque and reflect more solar radiation.
A similar derivation can be made for ice clouds and ice crystals, though the scattering properties are more difficult to determine due to the difference in geometry and refraction, as aforementioned.
So to predict radiative effects due to clouds, information is needed about the cloud's optical and geometrical composition, as well as water/ice content, which is difficult due to the uncertainties of these quantities [Wolf et al., 2020].

Clouds form vertically and horizontally into different shapes and vary in thickness and opaqueness. In the previous sections horizontal homogeneous layers have been considered with the plane-parallel approach. The different shapes, sizes and compositions at which clouds form at different altitudes in the atmosphere pose a problem to this assumed horizontal homogeneity.
It follows a short presentation of two methods on how cloud effects can be treated in NWP models.

2.2.4.1 Independent Column Approximation (ICA)

Consider a domain R that stretches out tens or hundreds of kilometers into the horizontal and assume that the three-dimensional distribution of the cloud properties is known exactly. The averaged, spectral integrated flux $\langle F \rangle$ for this domain then is [Pincus et al., 2003]:

$$\langle F \rangle = \int S(\lambda) \Big(\int \int_R F_{3D}(x, y, \lambda) dxdy \Big) d\lambda \tag{2.133}$$

where $S(\lambda)$ is a weight depending on the incoming flux for each spectral integral $d\lambda$ and F_{3D} is the three-dimensional flux.

The horizontal variations of the flux are for large scales, such as the synoptic and mesoscale, negligible, and the atmospheric columns can therefore be treated independently. This is called the independent column approximation (ICA), for which $\langle F \rangle$ then can be approximated to $\langle F^{ICA} \rangle$:

$$\langle F \rangle \approx \langle F^{ICA} \rangle = \int S(\lambda) \Big(\int \int_R F_{1D}(x, y, \lambda) dxdy \Big) d\lambda \tag{2.134}$$

where F_{1D} denotes the one-dimensional radiative fluxes.

Radiative fluxes are very different for clear sky and cloudy conditions, i.e. they are more horizontally homogeneous in clear skies than in cloudy areas. Therefore it is common to separate the flux into a clear sky part $\langle F^{ICA}_{clr} \rangle$ and cloudy portion $\langle F^{ICA}_{cld} \rangle$ with the cloud cover A_c:

$$\langle F^{ICA} \rangle = (1 - A_c) \langle F^{ICA}_{clr} \rangle + \langle F^{ICA}_{cld} \rangle \tag{2.135}$$

A typical approach to resolve partial cloud coverage is to divide each layer into individual sections that either are cloud free or homogeneously cloud covered. The total radiative flux becomes then a sum of the partial fluxes at each section weighted by the cloud fraction. For this the distribution $p(s)$ for all possible states s of the cloudy atmosphere is introduced and taken the integral over:

$$\langle F^{ICA} \rangle = (1 - A_c) \int S(\lambda) F^{clr}_{1D}(\lambda) d\lambda + A_c \int S(\lambda) \Big(\int p(s) F_{1D}(s, \lambda) ds \Big) d\lambda \tag{2.136}$$

$$\begin{aligned} \langle F^{ICA} \rangle =& (1 - A_c) \sum_k^K w(\lambda_k) S(\lambda_k) F^{clr}_{1D} \\ &+ A_c \sum_k^K w(\lambda_k) S(\lambda_k) \sum_j^J p(s_j) F_{1D}(s_j, \lambda_k) \end{aligned} \tag{2.137}$$

where the spectral integration in equation (2.136) has been approximated as discrete sums with weights w in equation (2.137).

In NWP models $\langle F^{ICA} \rangle$ is typically evaluated in every grid cell. This can, however, become computationally expensive, depending on the the number of layers filled with clouds and how those overlap, as the calculations are done for the spectral integral, i.e. for all spectral bands.

2.2.4.2 Monte Carlo Independent Column Approximation (McICA)

One method to reduce the needed computations for overlapping cloud covers is the Monte Carlo Independent Column Approximation (McICA).
The computation of the cloudy flux $\langle F_{cld}^{ICA} \rangle$ from equation (2.136) involves a two-dimensional integral, one over the wavelength λ, and a second over the cloud states s. The concept of the McICA method is, to choose random cloud states s_{random} for each spectral interval:

$$\langle F_{cld}^{ICA} \rangle \approx \sum_{k}^{K} w(\lambda_k) S(\lambda_k) F_{1D}(s_{random}, \lambda_k) \tag{2.138}$$

This means that the flux $\langle F_{cld}^{ICA} \rangle$ is calculated for a randomly choosen cloud state s_{random} from the probability distribution $p(s)$, which is the Monte Carlo method from statistics, thus the name McICA.
While this method will reduce the computational cost, it will also introduce a sampling error for each calculated $\langle F_{cld}^{ICA} \rangle$. This error is random and for many calculations the bias goes towards zero [Pincus et al., 2003].

2.2.4.3 Maximum-random cloud overlap

Another common approach to estimate cloud overlapping in solar radiation parameterizations is the maximum-random cloud overlap, which is a combination of the maximum and random overlapping technique [Morcrette and Fouquart, 1986].
The choice of a minimum, maximum or random overlap method depends on the spatial resolution of the model. For a very coarse horizontal resolution the minimum method might be the best approach.

All techniques involve two steps. First the radiative fluxes are calculated for the cloud configurations allowed by the chosen overlap method. Then all those fluxes are linearly combined with their cloud fractions as weights, to yield the total flux in the grid cell.

Imagine an atmosphere divided into three layers, potentially covered by clouds. Denote the layers as low, mid and high atmospheric layers, each with their own cloud cover C_l, C_m and C_h.
For the random overlap method each of the three layers is considered independent, which means that there can be eight combined cloud covers defined.
The first one is the clear sky fraction C_{clr}:

$$C_{clr} = \prod_{i=1}^{3} (1 - C_i) = (1 - C_l)(1 - C_m)(1 - C_h) \tag{2.139}$$

The next three are combined fractions, where always only one layer is covered by clouds at the same time C_j^1:

$$C_j^1 = C_j \prod_{i \neq j}^{i=1,2,3} (1 - C_i) \tag{2.140}$$

Additionally, there can be defined three combined fractions, in which two layers covered by clouds overlap C^2_{ij}:

$$C^2_{ij} = (1 - C_k)C_iC_j \tag{2.141}$$

Finally, there is one combined fraction, where all layers include clouds and overlap C^3:

$$C^3 = \prod_{i=1}^{3} C_i = C_iC_jC_k \tag{2.142}$$

The indices i,j and k represent each one of the three layers respectively.
Similarly, for the maximum overlap approach one can define four combined cloud fractions:

$$C_{clr} = 1 - \max(C_l, C_m, C_h) \tag{2.143}$$
$$C^1_j = \max(0, C_j - (C_i, C_k)) \tag{2.144}$$
$$C^2_{ij} = \max(0, \min(C_i, C_j) - C^3) \tag{2.145}$$
$$C^3 = \min(C_l, C_m, C_h) \tag{2.146}$$

This means that there are only half as many computations needed for the maximum overlap method than for the random method.
However, the assumption of the maximum method, that all cloud layers overlap, often leads to exaggerations for the cloud cover. Therefore a common approach is to combine the random and maximum overlap method as depicted in figure 2.10, here for ten layers. In this combined approach all layers are divided into three categories: a low, mid and high atmosphere. Inside each group the layers are combined through the maximum overlap method, while the three categories are treated independently as randomly overlapped.

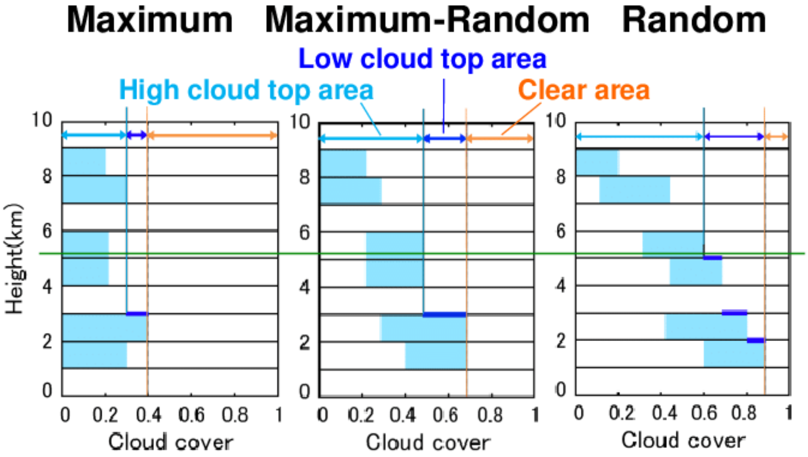


Figure 2.10: Illustration of the (left) maximum, (middel) maximum-random and (right) random cloud overlap methods. The blue blocks represent clouds in atmospheric layers. The high cloud top, low cloud top and clear areas are indicated by the three different colored arrows. Figure modified by [Kawai et al., 2014], originally adapted from [Hogan and Illingworth, 2000].

2.3 Artificial Neural Networks

Machine learning takes up an increasingly larger part in new developments, as there exist various neural network types for different purposes.
Not only can neural networks learn to recognize patterns, e.g. in pictures (classification challenges), but they can also be used for regression problems, such as the prediction of variables as done by NWP models, which will be examined in this study.
As the name indicates, artificial neural networks seek to work with and learn from data in a similar way as the human brain processes information.
While the feedforwad neural network used in this study is the simplest type of neural networks, it is still suitable and optimizable for various applications, such as regression problems.
This section's general introduction to the architecture of feedforward neural networks is mainly based on the books [Goodfellow et al., 2016] and [Aggarwal, 2018].
The fine-tuning and optimisation process of various hyperparameters will be gone through in detail in section 4.

In practice, all neural networks in this study have been coded and trained in Python [Van Rossum and Drake, 2009] using the application programming interface (API) Keras [Chollet et al., 2015] with Tensorflow [Abadi et al., 2015] as backend. For the implementation into the WRF model, a fortran based model, the Fortran-Keras Bridge (FKB) [Ott et al., 2020] was used. The FKB is a neural Fortran library that is specifically designed to simplify the process of incorporating neural networks trained in Keras into fortran codes.

2.3.1 Feedforward neural network

A feedforward neural network consists of multiple layers, each containing a number of nodes, also called neurons. In a simple feedforward neural network all layers are fully connected, i.e. every node in one layer is connected to all nodes of its neighboring layers. The number of layers determines the depth of the model, which is where the term "deep learning" arose from.
Figure 2.11 depicts a simple feedforward model, with three input variables, two output variables and two intermediate layers, each with four nodes. The intermediate layers are called hidden layers, as their nodes' values do not represent input or output types of values, but rather just mathematical intermediate values.
The name "feedforward" refers to neural networks, where information only propagates in one direction, that means that the node in a layer only depends on the values of the previous layer.
A neural network tries to find the best approximation of some real, but usually unknown function f^*, which connects the input variables x_i and output variables y_i as $\mathbf{y} = f^*(\mathbf{x})$, where $\mathbf{x}$ and $\mathbf{y}$ are vectors containing all input and output variables respectively.
To get the best appromixate solution with a function f, a neural network tunes several parameters $\overline{\theta}$) while training with known input and output variables, which mathematically can be described as: $\mathbf{y} = f(\mathbf{x}, \overline{\theta})$.

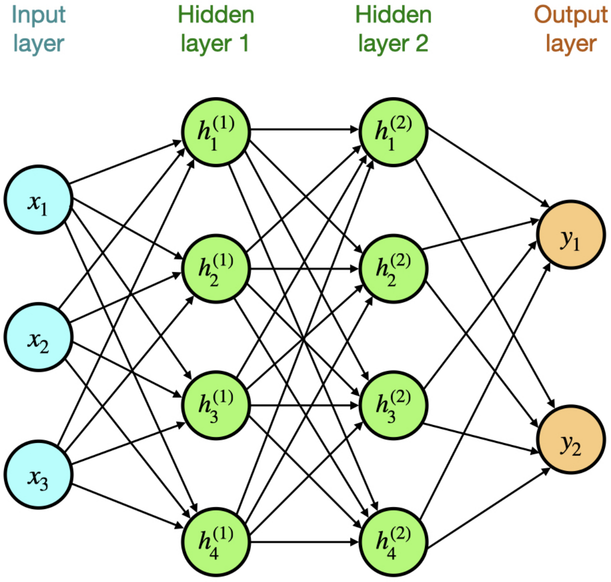


Figure 2.11: Illustration of a simple feedforward neural network. The network consists of three input variables, two output variables and two hidden layers each with four nodes/neurons. The (forward) information flow in the network is indicated by the arrows all going in the same direction.

The final function f can be represented as a chain of a function per layer in the model as e.g. $f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$, which is a representation for a model with two hidden layers as shown in figure 2.11. Here $f^{(1)}$ and $f^{(2)}$ are the functions of the first and second hidden layer respectively. The outermost function $f^{(3)}$ is the function for the output layer.
The input layer consisting of the vector $\mathbf{x}$, can be thought of as the 0'th layer, so that the vector's values also can be expressed as: $x_i = h_i^{(0)}$.
As mentioned before, does a layer only get input from the previous layer, so that e.g. the first layer's parameters are calculated with: $\mathbf{h}^{(1)} = f^{(1)}(\mathbf{x}, \mathbf{w}, \mathbf{b})$, where $\mathbf{w}$ is a matrix containing the weights between the input layer's and first hidden layer's nodes and $\mathbf{b}$ is a vector containing the bias of the first hidden layer. Both weights and biases are constants and exist for every layer in the neural network.
Considering the values in the vector $\mathbf{h}^{(1)}$ individually, one can write:

$$h_i^{(l)} = f(a_i^{(l)}) \tag{2.147}$$

$$a_i^{(l)} = \left(\sum_j w_{ij}^{(l)} h_j^{(l-1)}\right) + b_i^{(l)} \tag{2.148}$$

, where the subscript i describes the i'th node in a layer, while the superscript $^{(l)}$ denotes the l'th layer for $l = 1, 2, ..., L$. Note that $\mathbf{a}$ are linear functions of the previous layer's nodes. f is the activation function, which is differentiable and adds non-linearity to the calculations, which otherwise would reduce the neural network to a linear regression model.

For a neural network with $L - 1$ hidden layers, the L'th layer is the output layer. The full feedforward model can then be described with:

$$h_i^{(0)} = x_i \tag{2.149}$$

$$h_i^{(l)} = f(a_i^{(l)}) \text{ for } l = 1, 2, ..., L-1 \tag{2.150}$$

$$a_i^{(l)} = \left(\sum_j w_{ij}^{(l)} h_j^{(l-1)} \right) + b_i^{(l)} \tag{2.151}$$

$$\hat{y}_i = h_i^{(L)} = \left(\sum_j w_{ij}^{(L)} h_j^{(L-1)} \right) + b_i^{(L)} \tag{2.152}$$

From equations (2.149) - (2.152) it becomes apparent, that while the number of biases increases linearly with the number of layers of the neural network, the number of weights grows even stronger as it is the product of the count of nodes between layers. The simple neural network with two hidden layers shown in figure 2.11 has e.g. 46 model parameters. Tripling the number of nodes only in the two hidden layers from four to 12 changes the count of total model parameters to 230, which is five times as many as before. It is common for hidden layers to be much larger than the input and output layer in deep neural networks. The number of weights for deep networks becomes therefore approximately proportional to the square of the count of nodes per layer in these large hidden layers.

As aforementioned, the activation function f is the reason why feedforward neural networks can learn non-linearities and a neural network without activation functions would be the same as a linear regression model. There are many activations available that are suitable for different applications. It is also possible to choose different activations for each individual model layer, however, there are some functions which will not work as activation functions.

If e.g. the activation function would be linear function, the neural network would not be able to learn about non-linear data either. No matter how many linear functions one chains together, the resulting composition function will still only be linear and thus unable to learn non-linearities.

Another simple, yet well known function is the step function:

$$f(a) = \begin{cases} 0 \text{ if } a < 0 \\ 1 \text{ if } a \geq 0 \end{cases} \tag{2.153}$$

The step function, which is either 0 or 1, can be interpreted as a node being activated or not, as all nodes only can give one of those two values to the next layer. The importance of the individual nodes is then expressed through the weights of the layer, which represent the relative strength of the signal of the nodes from the previous layer. The bias is a measure of the general probability that a node will be activated and send a signal to the next layer. This is limiting the neural network to classifications of only two categories. Additionally, the derivative of the step function is the Dirac-delta function, which is zero everywhere apart from at $a = 0$, which can lead to

inefficient optimizations for larger networks, thus the step function is unsuited for deep neural networks.
The sigmoid function is similar to the step function, with a few properties that makes it a favoured activation function. It is shown together with some of the most common activation functions used in neural networks in figure 2.12.

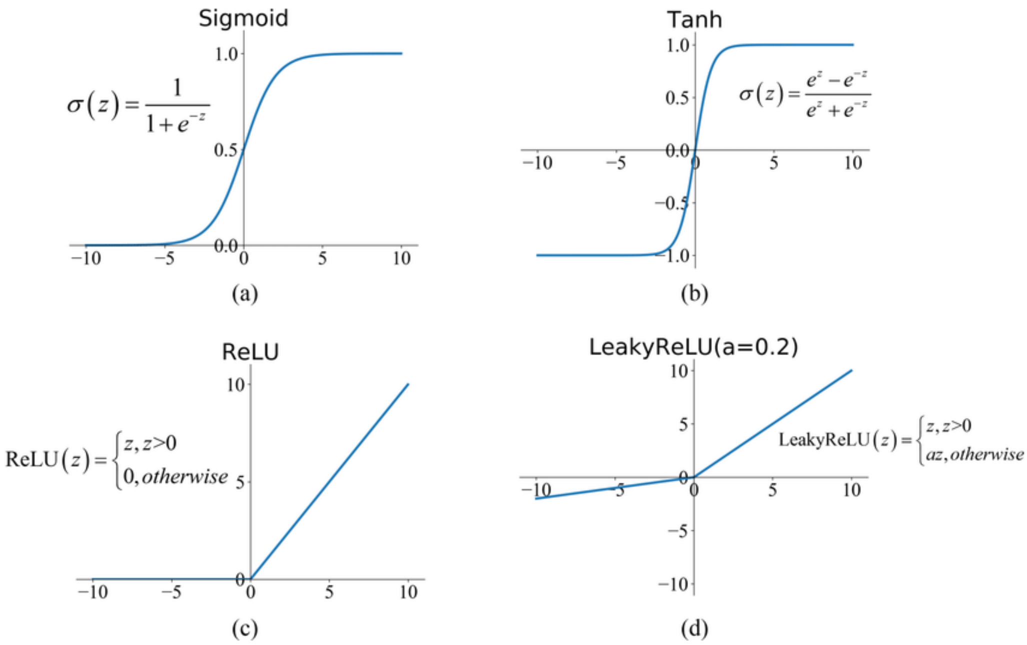


Figure 2.12: Commonly used activation functions in neural networks, [Feng et al., 2019]

Mathematically the sigmoid function is defined as:

$$f(a) = \frac{1}{1 + e^{-a}} \tag{2.154}$$

Like the step function, the sigmoid functions minimum and maximum values are 0 and 1, but it can also take on all values in the range between them. The sigmoid function is a non-linear function, which is fully differentable with multiple derivatives different from zero, which makes it possible to optimize the model parameters, i.e. the weights and biases, with the gradient descent, which will be described in section 2.3.2. If the sigmoid function is used as activation for the output layer, the values of the nodes, which range between 0 and 1, can represent the probabilities of different classes. It is also a useful function if the output variables only are allowed to become values in this range. One disadvantage of the sigmoid function is, that the function converges to a constant value for large weights, as its derivative goes to zero. Such a saturation of the function can lead to a slowing down of the training of the neural network.
A function which does not saturate as easily is the rectified linear unit (ReLU) function:

$$f(a) = \begin{cases} 0 \text{ if } a < 0 \\ a \text{ if } a \geq 0 \end{cases} \tag{2.155}$$

The ReLU activation is commonly used for regression problems, as this function does not constrain the output values as much as e.g. the sigmoid function.
The leakyReLU function is a modified version of the ReLU function, that has a small positive slope for negative values, to avoid the problem of the constant zero gradient of ReLU for small values.
The fourth common activation function shown in figure 2.12 is the hyperbolic tangent function, tanh, which in its form is similar to the sigmoid function, but lies halfway in the negative range.

Figure 2.12 shows the unit activation functions. However, the actual activation in a neural network is influenced by other parameters, e.g. the bias which shifts the activation function to better fit the data, so that the predicted output values become closer to the real output values.
There is no generally best activation function for all problems, as the activation function in itself is dependent on the neural networks architecture, i.e. the number of layers and nodes. All those hyperparameters need to be tested and tuned for a specific problem.

2.3.2 Network training and the gradient descent

Consider training a neural network with information of a number N data points. The dataset can then be separated into the input vectors $\{\mathbf{x}_n\}$ and output (target) vectors $\{\mathbf{y}_n\}$, for $n = 1, 2, 3, ..., N$. The n'th input and output vector consist of the input and output variables used in the first and last layer of the neural network as described earlier and are therefore unrelated to the number of data points N.
NWP models divide the atmosphere into horizontal and vertical columns, which can yield a large amount of available data points, easily in the order of millions (10^6). In the context of radiative transfers the input vector $\mathbf{x}_n$ could contain e.g. the optical thicknesses τ, the single scattering albedos $\tilde{\omega}$ and the asymmetry factors g in the data point, while the output vector $\mathbf{y}_n$ includes the reflectivity R and transmissivity T of the data point.

Similarly as done in equation (2.152), one can describe the predicted output vector $\hat{\mathbf{y}}_n$ of a neural network as:

$$\hat{\mathbf{y}}_n = f(\mathbf{x}_n, \mathbf{W}) \tag{2.156}$$

where the vector $\mathbf{W}$ contains multiple model parameters, similar to $\overline{\theta}$, namely both the weights $w_{ij}^{(l)}$ and biases $b_i^{(l)}$.
Note that $\hat{\mathbf{y}}_n$ is the predicted output value of the neural network, whereas $\mathbf{y}_n$ is the vector with the true target values that the neural network uses to train and learn from.
The neural network seeks to minimize the difference between these two vectors and one defines therefore the measurement of this difference as the loss function $J(\mathbf{W})$:

$$J(\mathbf{W}) = \frac{1}{N}\sum_n e(\mathbf{y}_n, \hat{\mathbf{y}}_n) \tag{2.157}$$

which is a function of the model parameters $\mathbf{W}$, while e is a function that measures the error of each prediction. This function e is chosen for the individual problem and application. Common loss functions for regression problems are the absolute mean error:

$$J(\mathbf{W}) = \frac{1}{N}\sum_n ||\mathbf{y}_n - \hat{\mathbf{y}}_n|| \tag{2.158}$$

or the mean squared error:

$$J(\mathbf{W}) = \frac{1}{N}\sum_n ||\mathbf{y}_n - \hat{\mathbf{y}}_n||^2 \tag{2.159}$$

Since both of the above mentioned loss functions and the neural network consist of differentiable functions, the derivative of the loss function with respect to the model parameters $\frac{\partial J}{\partial W_i}$ can be computed. This derivative, which is a measurement of how sensitive the loss function is to each model parameter, can then be used to adjust each of the model parameters using gradient descent, optimizing the model:

$$W_i^{new} = W_i - \alpha\frac{\partial J}{\partial W_i} \tag{2.160}$$

where α is the rate at which the parameters will be updated, called the learning rate. The learning rate is another hyperparameter that is always positive and depends on the model configuration and individual problem.

In contrast to the forward information propagation in the feedforward model, the errors algorithm that updates the model parameters with the gradient is called back-propagation, as this is done after a set of predicted output values has been calculated by the model and requires computing several partial derivatives.

The optimization using gradient descent is computationally demanding and slow for large datasets, as the calculation of the gradient $\frac{\partial J}{\partial W_i}$ depends on all data points.

An alternative approach which arises from the same principle, but with the addition of stochastic elements is the stochastic gradient descent (SGD), which is less computational-heavy and therefore more applicable for deep learning.

To reduce the error between the predicted and target outputs the model searches for the minimum of the loss function J. However, since the loss function is generally non-linear and non-convex, this is a complex task, as there is a risk of converging towards a local minimum or saddle point. To prevent this stochastic elements can be used to more easily escape these points.

In practice this means that instead of computing the gradients for the whole dataset, only gradients for a smaller (random) subset, called a mini-batch, are calculated and used to update the model parameters. This eases the computational expense and adds a stochastic characteristic through the random choice of data points inside the mini-batch. For the SGD algorithm the loss function is only calculated for a mini-batch of size B:

$$J_B(\mathbf{W}) = \frac{1}{B}\sum_{n=1}^{B} e(\mathbf{y}_n, \hat{\mathbf{y}}_n) \tag{2.161}$$

which leads to the following expression for the updated model parameters:

$$W_i^{new} = W_i - \alpha \frac{\partial J_B}{\partial W_i} \tag{2.162}$$

The SGD algorithm does those calculations over as many mini-batches needed until it has used all data points. The number of mini-batches depends on the batchsize B, and the number of mini-batches needed for the model to encounter all data points once is called an epoch.

The batchsize and number of epochs for which the model trains on the dataset are another two hyperparameters, that need to be tuned for the specific problem.

The SGD algorithm is used as an optimizer by the neural network, which is yet another hyperparameter that can be tuned for the specific application. Several different optimization algorithms have been developed on the basis of the SGD algorithm, such as adaptive learning rates and momentum algorithms.

The momentum algorithm includes two additional parameters: The velocity parameter v and the momentum $\beta \in [0, 1[$, which regulate how quickly the effect of previous gradients decreases, i.e. how easy it is for the new gradient to change direction.

$$W_i^{new} = W_i + v \tag{2.163}$$

$$v = \beta v^{old} - \alpha \frac{\partial J_B}{\partial W_i} \tag{2.164}$$

For $\beta = 0$ the momentum algorithm's equation (2.163) reduces to equation (2.162) of the SGD algorithm. The purpose of incorporating previous gradients through the momentum term is to smooth out fluctuations in the gradient descent.

Another approach to optimizations are adaptive learning rates. An optimizer that both includes the benefits of the momentum algorithm and adaptive learning rates is the adaptive moment estimation, also called the Adam optimizer.

Besides the learning rate, every optimizer has also its own set of hyperparameters that need to be tuned manually for the individual problem to get the best results.

Adam is a popular choice as optimizer since it does require little tuning of its hyperparameters [Kingma and Ba, 2014] and includes many advantages of other algorithms [Aggarwal, 2018].

Another method to enhance the training of the model is to use a learning rate scheduler, i.e. to change the learning rate manually during training. One concept for such a scheduler is to decrease the learning rate after a certain number of iterations, as smaller learning rates are beneficial when the model is sufficient close to a good minimum of the loss function.

There are also cyclic learning rates schedules, where the learning rate varies between a specified range of values periodically, e.g. in a triangular pattern as proposed by

[Smith, 2015]. The goal of a temporary increase of the learning rate is to reduce the risk of the model to converge to a local minimum or saddle point.
The cyclic learning rate method in particular will be further described in section 4.2.1, where the tuning of the learning rate hyperparameter will be presented.

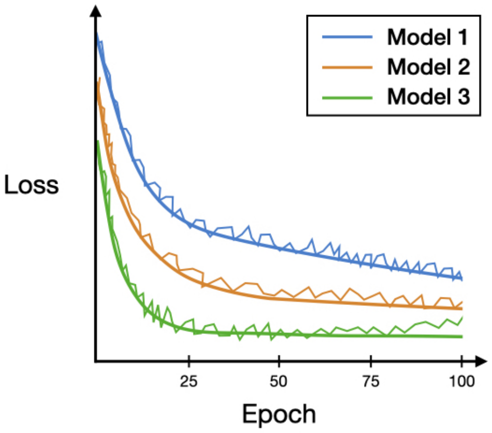


Figure 2.13: Example of training and validation learning curves for three different models. The thick and thin curves are the loss of the training and validation data respectively.

To determine when a model has found a good minimum of the loss function and has learned long enough, the learning curve, i.e. the loss at each epoch, is examined. A sketch of some learning curves is shown in figure 2.13.
Here, the loss of the training data is shown as thick lines, while the loss of the validation data, i.e. an additional independent dataset, is shown as thin lines. For all models the losses generally decrease with the number of epochs. Assuming that all three models are trained with the same training and validation datasets, the different rate at which the models learn is a result of different hyperparameter and neural network configurations.
From the curves it looks like model 1 is still learning, while the trainings loss has converged towards a mostly flat curve for model 2, which indicates that the model does not learn much more from the trainings data.
A second, independent dataset is used as validation dataset to see how well the model performs on data it has not encountered during training. Additionally the comparison between the training loss and validation loss helps to identify cases of overfitting to the trainings data. In the case of overfitting the model adjusts its model parameters too much to the specific characteristics of the trainings data, which reduces the model's ability to predict outputs for unknown data.
When overfitting occurs the trainings loss will continue to decrease, while the validation loss will stabilize or even start to increase. An example of this are the learning curves for model 3, where the validation loss has started to increase after epoch ~ 60, while the training loss seems to continue decreasing slightly.

It is therefore good practice to save the best model, instead of the model after the last epoch, where the best model usually is chosen as the model with the lowest validation loss. Since the model has already seen the values in the training and validation datasets these can not be used to evaluate the performance of the best model on unknown data. To get an unbiased estimate of the performance of the best model, one thus needs a third independent dataset, i.e. a test dataset.

3 The WRF-model

All simulations in this study were conducted with the Weather Research and Forecasting model (WRF), which is a regional, non-hydrostatic NWP model able to run idealized and real weather cases.
This section will serve as brief introduction to the structure and main elements of the model build upon the official technical model's description [Skamarock et al., 2019] and the official WRF user's guide [WRF-userguide,].

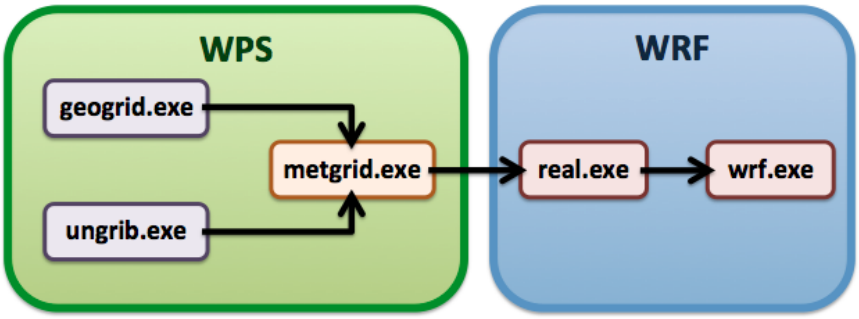


Figure 3.1: Illustration of the processes for a simulation with the WRF model

Figure 3.1 shows a simplified workflow of the WRF model and its processes. As indicated by the figure, the model's programs can be separated into two segments. First the input data and horizontal grid are prepared by the WRF Preprocessing System (WPS). Afterwards the data is vertically interpolated by `real.exe` and then the simulation is carried out by the dynamical solver, the Advanced Research WRF (ARW) in `wrf.exe`.

Section 3.1 will focus on the WPS and the domain configuration used for the simulations, while section 3.2 will describe the ARW's key features. Lastly, in section 3.3 an overview of the radiation parameterization schemes used in this study will be given.

3.1 The WRF Preprocessing System (WPS)

To make a simulation with the ARW, an initial model state and boundary conditions must be prepared. For this the WPS prepares a horizontal grid area, the model domain, with terrestrial and meteorological data.
Both the terrestrial data used to create the model domain with the `geogrid.exe` program and the meteorological data prepared by the `ungrib.exe` program must be provided externally.
The configurations for the domain size and location, as well as the start and end time for the simulation must be specified in the `namelist.wps` file.
Based on the provided simulation times, the `ungrib.exe` program prepares the input data, i.e. it unpacks the GRIB files containing gridded information about the atmospheric variables (temperature, pressure, etc.) into an intermediate file format.

For all simulations in this study, final operational analysis data from the Global Forecast System (GFS) by the National Centers for Environmental Prediction (NCEP) with a spatial resolution of 0.25° x 0.25°, which corresponds to ca. 28 x 28 km, has been used as meteorological input data [NCEP, 2015]. These data files contain meteorological data in 6-hour time-intervals.
The unpacked meteorological data is then interpolated horizontally onto the model domain by the `metgrid.exe` program and saved as netCDF files.

3.1.1 The domain and model setup

The domain created by the `geogrid.exe` program for this study is depicted in figure 3.2. An example of the corresponding configurations in the `namelist.wps` file is shown in Appendix A.

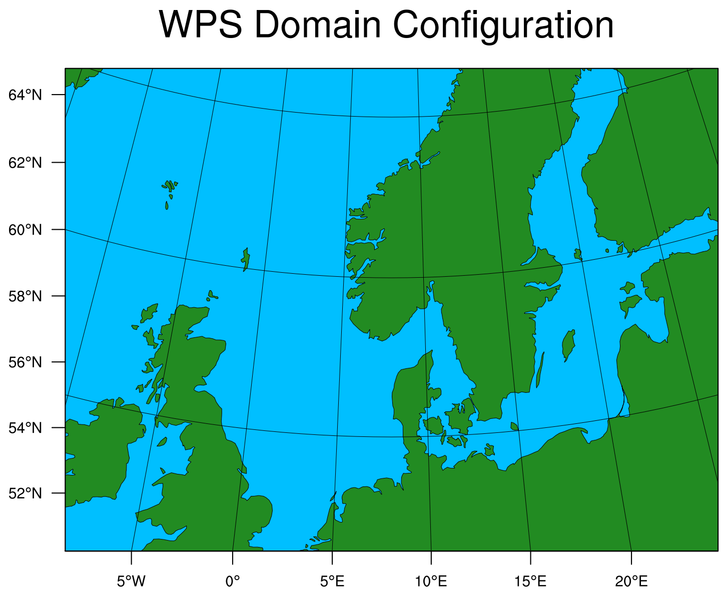


Figure 3.2: Domain used in the WRF model for this study

WPS supports different types of map projections. For the domain here, covering most of Scandinavia, the Lambert conformal conic projection has been chosen, since it is well suited for mid-latitudes.
The models horizontal grid resolution has been set to 10km x 10km. This is a higher horizontal resolution than the 28km x 28km spacing of the provided meteorological input data used for the lateral boundary conditions. The boundary conditions will therefore be nested down onto the finer grid by the model to avoid noise at the boundaries. The ratio of boundary and model domain resolutions should not much larger, as this could lead to distortions.
To achieve even higher resolutions from coarse input data WRF offers additional nesting options inside the main domain. For investigating radiative transfers a grid

size of 10km seemed sufficient and additional nesting was therefore not necessary.
The spatial dimensions of the domain are 230 x 170 (staggered) grid points, with 70 vertical layers. The spatial distribution of the model and its staggered dimensions are explained in the next section.

3.2 The Advanced Research WRF (ARW)

For simulating real weather cases with WRF, the output netCDF files from the `metgrid.exe` program need to be interpolated into the vertical model layers to create boundary conditions for the ARW, which is done by the `real.exe` program.
The vertical layers in the WRF model are described by a terrain following pressure coordinate η:

$$\eta = \frac{p_h - p_{hs}}{p_{hs} - p_{ht}} \tag{3.1}$$

where p_h is the hydrostatic component of pressure at a given level h, p_{hs} at the surface and p_{ht} at the upper boundary. The η-coordinate varies between $\eta = 1$ at the surface and $\eta = 0$ at the top pressure level, as depicted in figure 3.3.

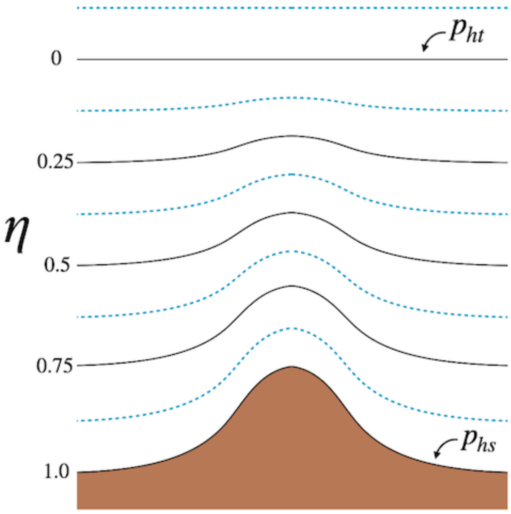


Figure 3.3: WRF's vertical levels (full and half levels, represented in black and blue lines respectively) for a terrain following vertical coordinate system.

The Euleran solver used by the ARW operates on an Arakawa C-staggered grid as illustrated in figure 3.4.
In such a grid the thermodynamic variables θ (pressure, temperature, humidity, etc.) are located in the center of the cell, called the mass points.

Meanwhile, the velocities are defined at the boundaries between two cells, staggered one half width away in their own direction, i.e. the zonal velocity u is staggered $\Delta x/2$ in the zonal direction x.
These points are called u, v and w points, respectively.
While u and v are staggered with the constants $\Delta x/2$ and $\Delta y/2$ in the horizontal, the vertical velocity w is staggered at half levels of η.
These half levels are defined at the middle between two full η levels and thus are not constant in the vertical as depicted on the right in figure 3.4. Full and half levels of η are also shown in figure 3.3 as black and blue lines, respectively.
Half levels correspond to the mass points, while the w points are located on full levels. Physical parameterizations, such as the radiation schemes, calculate their variables at mass points, i.e. at half levels.
Note that the uppermost half level in figure 3.3 is a fictitious level with no physical properties. However, there are some solar radiation schemes where this fictitious level is used to extrapolate properties between the top layer of the model and the TOA.

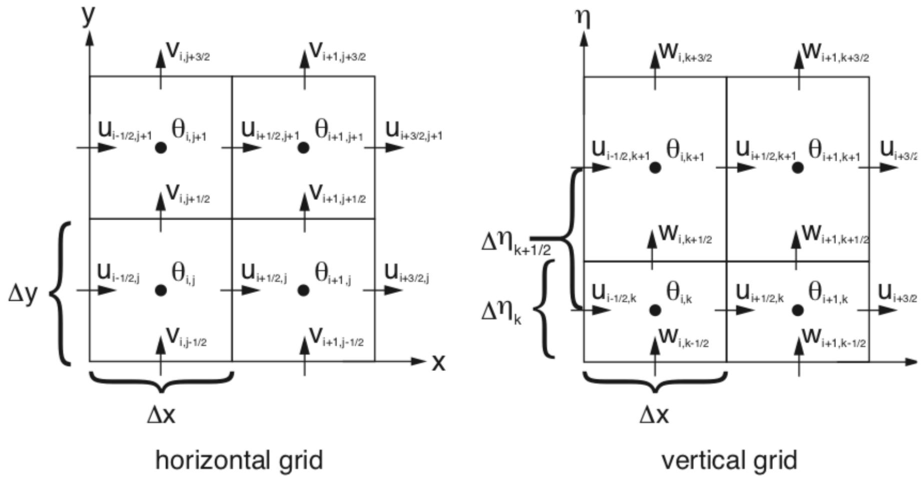


Figure 3.4: Illustration of Arakawa C grid cells and their spatial distribution horizontally (left) and vertically (right). Figure from [Skamarock et al., 2019]

The number and distribution of the full vertical layers η needs to be specified in the `namelist.input` file, as well as the horizontal grid corresponding to the one set up in `namelist.wps`.
Additionally, other configurations, such as the choice of physical parameterization schemes, the time step and options such as the digital filter initialization (DFI) need to be defined in the `namelist.input` file as well. An example of such a file can be found in Appendix A.

As shown earlier in figure 2.6, there are a number of parameterization schemes for unresolved physics in the WRF model, such as e.g. the microphysics, the radiation and the land-surface parameterizations. There exist a variety of different parameterization schemes to choose from for each one of them. While one can specify individual parameterizations, WRF also offers the option of predefined physics suites, which are tested sets of parameterization schemes for specific regions and weather phenomena. In this study the "CONUS" suite was chosen as initial setup, after which different radiation schemes were tested. An overview of the chosen radiation parameterizations is presented in section 3.3.

The model time step can be freely set in the `namelist.input` file, however, there exists a general limit on how long the time step can be in an Eulerian model, to keep the simulation stable. In essence a simulation will be stable, as long as no wave can travel further than the distance between two grid points, e.g. Δx in the x direction, in one time step Δt.

There are two types of time steps in the ARW, the model (advective) time step Δt for low frequency modes that needs to be specified in the namelist.input file, and the smaller acoustic time step for higher frequency modes, which is automatically set to a fraction of the model time step.

The maximum model time step Δt_{max} can be found with the Courant number:

$$\Delta t_{max} < \frac{C_{max}}{\sqrt{3}} \cdot \frac{\Delta x}{u_{max}} \tag{3.2}$$

where u_{max} is the maximum advection speed and Δx the distance between two grid points, i.e. the horizontal grid size in the x direction, as shown in figure 3.4.

The maximum Courant number C_{max} for the third-order Runge-Kutta time integration (RK3) used in the ARW, depends on the chosen spatial discretizations order of the advection scheme. It is, however, not recommended to use the maximum time step length, but rather to subtract a buffer of 25%.

As a rule of thumb, it is recommended to set the time step in seconds to $\Delta t = 6 \cdot \Delta x$, where Δx is the grid point distance in kilometers, which for our domain with $\Delta x =$ 10km gives us a time step of $\Delta t = 60$s [Skamarock et al., 2019].

3.2.1 Digital filter initialisation (DFI)

Another source of instabilities and noise in a simulation is the initial interpolation of the meteorological data on the discrete domain grid, which can lead to imbalances in the first hours of a simulation.

While one could discard the first few hours of a simulation as a spin-up time for the model to even out those imbalances, ARW has an optional digital filter initialisation (DFI).

The ARW provides different DFI options, which are depicted in figure 3.5.

For all simulations in this study, the recommended Twice DFI (TDFI) has been applied, which basically works as a low-pass filter, removing the initial imbalances as high frequencies, while keeping the lower modes, corresponding to the meteorological dynamics, thus dampening the highest fluctuations around the initial model state.

In practice the filter will first integrate adiabatically backwards from the initial state of the model, before integrating forward to create a filtered initial state of the model, as illustrated in the figure (last row).

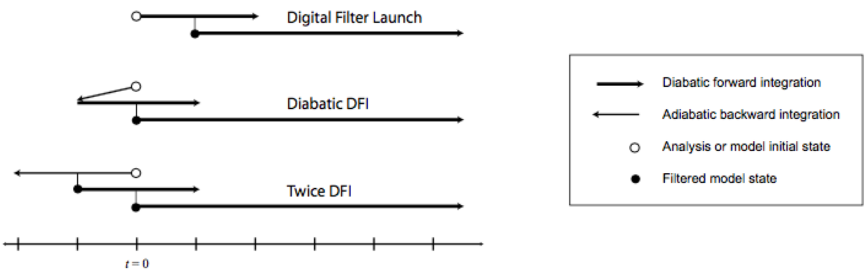

Digital Filter Launch
Diabatic DFI
Twice DFI
t = 0
Diabatic forward integration
Adiabatic backward integration
Analysis or model initial state
Filtered model state

Figure 3.5: Illustration of the three different digital filter initializations available in the WRF model. Figure from [Skamarock et al., 2019]

Therefore no constant spin-up time was considered for the simulations in this study. However, it should be noted, that since it was focused on the solar radiation, which only is present during daytime, and all simulations have been started at 00 UTC model time, there is a natural time where there is no sunlight reaching the domain covering Scandinavia at the beginning of each model run. This natural spin-up time is not constant, as it varies with seasons, i.e. it is shorter during the northern hemisphere's summer and longer during winter.

3.3 Radiation Parameterizations in WRF

The selected longwave and shortwave radiation parameterization from the `namelist.input` file is called in the model by the `radiation driver` in a frequency defined by the radiative time step *radt*.

The radiative time step is another parameter that needs to be specified in the `namelist.input` file and has been set to 1 hour, to match other operational models as mentioned before and making it possible to run the model for a longer duration at a lower computational cost to test how the model diverges for longer simulations.

The `radiation driver` handles the preprocessing of the input variables from the dynamical solver to the radiation parameterization, as well as the actual execution of the parameterization code and the postprocessing, i.e. returning the output to the main model code. The preprocessing includes, e.g. the evaluation of the information about cloud fractions from the microphysics and cumulus parameterizations.

The typical workflow of a radiation parameterization starts with the evaluation of the optical properties of the different contributors, i.e. gasses, aerosols and clouds, before the solver calculates the solar fluxes.

None of the various radiation parameterization schemes in the WRF model perform best in all situations or lead to the best predictions of all variables, as seen by other studies, e.g. [Stergiou et al., 2017].

The Rapid Radiative Transfer Model for general circulation models (RRTMG) seems to be one of the most similar parameterization schemes to those used in modern operational forecasting models run by the larger weather and climate centers, as e.g. the one used by the Integrated Forecasting System (IFS) by ECMWF [Hogan and Bozzo, 2018].

The RRMTG is also the most complex radiation parameterization available in the WRF model and was for those two reasons chosen as the main radiation scheme to be investigated.

Some early simulations were carried out to test the computational times of the different processes in both the longwave and shortwave RRTMG parameterization.

The early tests showed that the shortwave radiation scheme takes up ∼66% of the computation time of the `radiation driver`, each time it is executed, i.e. when there is sunlight. In practice the cosine of the solar zenith angle is evaluated for this condition. The longwave radiation scheme on the other hand is executed for all radiative time steps as its execution is independent of the solar zenith angle.

The percentage spend on the shortwave radiation parameterization of a whole model run depends therefore on the the chosen time period of the day as well as season.

Computation times will be compared and discussed in more detail in section 5.3.

However, in general the shortwave radiation took up a larger portion of the overall computation time than the longwave radiation for the RRTMG scheme. Therefore it has been focused on the shortwave radiation parameterization in this study.

The in WRF available RRTMG-fast radiation parameterization scheme is a newer, optimized version of the RRMTG parameterization and thus it has been chosen to use this RRTMG-fast as the main longwave and shortwave radiation parameterization scheme.

Section 4 will therefore present the development of neural networks that have learned from and can be implemented into the shortwave RRTMG-fast parameterization.

For all simulations in this study the RRTMG-fast longwave radiation parameterization has been used.

The case studies that will be presented later in section 5, have also been simulated with three other additional shortwave parameterizations: the RRTMG, the New Goddard and the Dudhia parameterization schemes.

For all radiation parameterizations the default configurations were used. This also means that there are no interactions with aerosols included in any of the used parameterizations, as this is not available for all parameterization schemes and the aerosol properties would need to be provided externally by e.g. a climatological record or by using the chemistry coupled WRF-Chem model.

In the following sections an overview of the different physical processes of the four used shortwave parameterizations is given referring to the methods described in section 2.2, as well as a short introduction to the `reftra_sw` subroutine of the RRMTG-fast parameterization.

3.3.1 RRTMG and RRTMG-fast

Both the RRTMG and the RRTMG-fast parameterization schemes are based on [Iacono et al., 2008]. The physics of the two schemes are the same, however, the

RRTMG-fast is optimized for the usage with GPUs, which allow for much faster computations than CPUs.

In the RRTMG the solar spectrum (ranging between 200nm and 12.195nm) is divided into 14 spectral bands, which each are subdivided into smaller spectral intervals, called g-points. There are 112 g-points in total, which are unevenly distributed between the 14 bands.

The spectral integration is done through the correlated k-distribution method.

Rayleigh scattering as well as effects of water vapor, ozone, trace gases such as oxygen, carbon dioxide and methan are all included in this parameterization.

Multiple scattering is incorporated through the vertical integration with the adding method.

Clouds are integrated through cloud fractions and overlaps estimated with the Monte Carlo Independent Column Approximation (McICA) linked with the maximum-random overlap method.

The RTE is solved with the two-stream approximation with the Practical Improved Flux method (PIFM), of which the coefficients can be seen in table 2.2.

Additionally there is the option to use the δ-Eddington or the discrete ordinates method instead.

3.3.1.1 Subroutine reftra_sw

The `reftra_sw` subroutine is of all processes in the RRTMG-fast shortwave parameterization one of those that is executed the most times, as well as one of the most computational heaviest, which in combination makes it the subroutine of the RRTMG-fast that takes up most of the computation time of the `radiation driver`.

The subroutine computes the direct and diffuse reflectivities R_{dir} & R_{dif}, as well as transmissivities T_{dir} & T_{dif} for either clear or cloudy layers.

Both input and output variables are structured in chunks of a number *ncol* (independent) columns each, since the code is adjusted for parallel computation.

The input variables of this subroutine are: The number of columns per chunk *ncol*, the number of vertical model layers $nlayers$, a logical flag for the computation of reflectivities and transmissivities related to the cloud area fraction $pcldfmc$, the asymmetry factor g, the optical thickness τ, the single scattering albedo $\tilde{\omega}$, the cosine of the solar zenith angle $\cos\mu_0$ and a logical flag ac indicating whether the columns are considered clear or cloudy.

The outputs, i.e. the reflectivities and transmissivities, are used to compute the radiative fluxes in the following subroutines.

Most of the variables in this routine are 3-dimensional, since they are specified by their number of column, the vertical layers as well as the g-points.

3.3.2 New Goddard

The New Goddard parameterization is an advanced radiation scheme based on [Chou and Suarez, 1999].

In this parameterization the solar spectrum (175nm - 10.000nm) is split into 11 spectral bands.
New Goddard includes Rayleigh scattering, ozone, minor gases such as oxygen and carbon dioxide, as well as cloud effects through the maximum-random approach.
The two-stream approximation with the δ-Eddington method is used to solve the RTE and compute the solar fluxes.

3.3.3 Dudhia

Dudhia is the simplest of all shortwave radiation parameterizations in WRF, based on [Dudhia, 1989]. It is a broadband integrated scheme that only estimates the downward flux (no upward stream).
Only Rayleigh scattering, water vapor and effects by clouds is included in the parameterization. No effects due to multiple scattering, trace gases or ozone are calculated, however, there is a bulk scattering parameter that should compensate for such effects. Instead of using one of the introduced methods to estimate cloud cover fraction distributions, the Dudhia scheme only distinguished between cloudy and clear layers, where the cloud cover is assumed to be evenly distributed through the whole layer. A look up table by [Stephens, 1978] is used to determine the cloud state.
The RTE is not solved explicitly in this parameterization, instead it is estimated how much the downward solar flux will be damped by the four effects mentioned above to estimate the downward surface flux.

4 Development of the Neural Network

Rather than substituting the whole shortwave parameterization with a neural network, it has been choosen to only focus on the most computational heavy part of the RRMTG-fast solar radiation scheme, the previously introduced `reftra_sw` subroutine, while the other parts of the radiation scheme keep their structure.
Other candidates to focus on among the other parts of the shortwave parameterization include the gas optics, i.e. the computation of the optical thicknesses in each spectral band, or the radiative solver, where the fluxes are calculated.

In this section the development of neural networks that can be used as substitutions for the `reftra_sw` subroutine will be described.
Since there is no definite method of optimizing neural networks for any given problem, this procedure included a trial and error phase early on, where different approaches and ideas were tested.
While the trying out of a variety of ideas and methods is helpful in the beginning to get a sense of what works with the specific problem, and what does not, it is not possible to give a well documented description of all of these trials.
Therefore this section will focus on some procedures more than others, while the main ideas and choices will be presented in greater detail.

The creation of the initial data set for training the neural networks will be presented in section 4.1. In section 4.2 the optimization of the different hyperparameters and structure of neural networks, as well as the benefit of data categorization will be described. The chosen neural network configurations for implementation into the WRF model based on these optimizations are shown and compared in section 4.3.

4.1 Dataset and preprocessing

The input and output of the `reftra_sw` subroutine are, as presented in section 3.3.1.1, radiative variables defined on a subgrid of g-points and atmospheric columns and layers. The variables are therefore not part of the normal WRF model output file and needed to be written out to separate files directly from the subroutine to create the training, validation and test data sets.
For this 12 24-hour simulations have been carried out with the WRF model and the RRTMG-fast scheme, one for the 15th day of each month in 2018 (simulations initialized at 00 UTC on the 15th), to cover a wide spectrum of different atmospheric conditions due to weather, time of day and seasons.
The data that is fed to the neural network should be diverse so that the machine learning model can learn the physics good enough to be able to predict unknown weather situations, in practice the neural network learns to reproduce the physical statistics from the training data. A large variability is therefore preferred for the data set and for effective learning the data points should be independent of each other.

The domain used in this study, introduced in section 3.1.1, has in total 229×169 horizontal grid cells. That results in the same number of columns with each 70 vertical layers, which means that per radiative time step, the `reftra_sw` subroutine calculates

229×169×70 = 2.709.070 points for each of the 112 g-points, i.e. 229×169×70×112 = 303.415.840 points in total for one radiative time step, i.e. per hour.

However, while the atmospheric columns are treated independently by the model, one can assume that neighbouring columns are correlated, as they will most likely have similar physical properties, e.g. a large cloud can stretch out over several horizontal and/or vertical neighbouring grid cells.
Therefore only every 100th input and output chunk of the `reftra_sw` subroutine is saved, that is 1 % of all the generated data. Further, while every chunk contains 8 columns, only the first is selected, so that leaves 0.125 % of all generated data points to be considered.

It should also be noted that the RRTMG-fast parameterization not only uses the fictitious half-level above the top model layer, as described in section 3.2, but also saves the surface properties in a 71th layer, which is however defined outside of the `reftra_sw` subroutine and therefore not taken as part of the training data for the neural network.

Since the shortwave radiation scheme only is executed for time steps where there is sun light in the domain, there are more columns sampled from the summer season than the winter season.
This resulted in 15029 columns, with 112 × 70 data points, still much more 100 million data points and far too many to be used for training neural networks on a standard computer as done in this study.

The columns are split randomly into three sets. Since the `reftra_sw` routine calculates its outputs point-wise, it was choosen to train the model on individual data points rather than vertical profiles as well, this means the architecture of the routine can be kept the same and using data points instead of profiles means also that there are is more data to train on, i.e. the number of individual data points is higher than the number of columns.
Each of the three randomly sampled data sets contains 39.270.560 data points.

Now one can take advantage of the very first `if-statement` in the `reftra_sw` routine. Before the actual computations, the routine checks the logical flags ac and $pcldfmc$ for the individual point. In essence, the routine sets the transmissivities to 1 and reflectivities to 0, if the point is declared as cloudy by ac, but $pcldfmc$ indicates that the layer is clear.

In early tests neural networks have been trained on data sets where those points where included. While it seemed like the neural network models could learn about this condition, there were some uncertainties and tests in the WRF model showed that the `if-statement` is computational cheaper than the routine's calculation or the prediction done by a neural network.
Therefore it was chosen to keep this condition in the new subroutine with the neural networks and since the neural network does not need to learn about this condition, this data is filtered out from the data set.
This resulted in a reduction of ∼30% of all data points in all three data sets.

Afterwards the distribution of clear and cloudy data points in the remaining data set was investigated.

Since we have filtered out so many data points for layers classified as "cloudy", it is not surprising that there are now many more clear sky layers left.
The remaining cloudy layers only make out ∼6% of each of the three data sets.
Different data distributions have been tried out, but choosing to create data sets where cloudy and clear sky cases were evenly represented (50-50%), worked better than splitting the data closer to the found data distribution (∼ 6%), as the cloud cases are more complex than clear sky conditions.
In the presence of clouds, the input and output variables take on values of a wider range, which are extreme data cases the neural networks need to learn about to correctly emulate the original `reftra_sw` subroutine.

The ∼6% make up ∼1.6 million data points in each set. It is chosen to use $(1024 \times 1500 =)$ 1536000 randomly selected data points for each category, clear and cloudy, for each data set. The data sets then contain each 3.072.000 data points in total, for training, validation and testing respectively.
Training grows slow the more data is used, and while more data can lead to better results, it needs to be introduced new, independent data, so that the neural network can learn something new.

The mixed, initial data set used for training is shown in figure 4.2 and 4.3, divided into the input and output variables, respectively.
Note that the optical thickness τ in figure 4.2 has been scaled to fit into the range of -1 to 1. This was one additional step needed to be done before feeding the data to the neural network: preprocessing the data.

Normalization is a common method used for better learning performance, since the loss function has a tendency of being more sensitive to some variables than others if the orders of magnitude of the variables in the data set vary strongly relative to each other [Aggarwal, 2018].
It is therefore preferred to scale all variables to have the same order of magnitude.
With the exception of the optical thickness, all input and output variables of the `reftra_sw` subroutine can only take on values between 0 and 1.
The optical thickness, however, can take on a wide range of magnitudes, from $6 \cdot 10^{-7}$ to ∼ 125.000.

Different approaches for the scaling of the optical thicknesses have been tested and it turned out that the scaling has a huge impact on the prediction skill of the neural network, in the worst case leading to computations of nonphysically small or large radiative fluxes in the following subroutines.

Typical approaches of normalizing include mean centering the data by subtracting the mean value of the variable, standardization, where the mean centered values also divided by the their standard deviation or min-max normalization, where the data is scaled to the range of 0 to 1 [Aggarwal, 2018].

While these methods can theoretically be applied on the sampled training, validation and test data sets, there are additional constraints to the normalization procedure that need to be considered in this case, since the normalization must not only be applied on the optical thickness in the three data sets, but inside the WRF model as well, when the neural network will be used to make predictions in the modified subroutine.

The normalization needs to be consistent between the data sets and in the WRF model, i.e. the optical thicknesses need to be scaled uniformly. Since the goal is to modify the `reftra_sw` subroutine, while trying to keep the rest of the parameterization structure unchanged, the scaling with the mean value or standard deviation would pose a challenge. Since the shortwave radiation scheme works in parallel on the small chunks of columns, the mean or standard deviation value in those chunks would vary a lot. The same optical thicknesses would therefore be scaled to different values, which would lead to bad predictions by the neural network.

So instead another typical approach is chosen, which is dividing the variable by a constant maximum value. Additionally, the method that turned out to work best is to first take the logarithm of the optical thicknesses, a common approach for variables that range between many orders of magnitudes, and then to divide by a constant maximum value, which has been chosen based on the magnitudes of the logarithmic scaled values of the training, validation and test data sets.
The unscaled and scaled optical thicknesses of the training data set for both the cloudy and clear sky cases can be seen in figure 4.1.

The scaling by first taking the logarithm before dividing by a maximum value might yield better results opposed to simply dividing the optical thicknesses by their maximum value, due to the wide range of magnitudes they can take, as well as the unevenly distribution of those orders of magnitudes. Not even 5% of all the optical thicknesses in the training data set are larger than 100, most of them are actually smaller than 1, as can be seen from the histogram on the left in figure 4.1. Dividing by a large maximum value leads then to even smaller values for a large portion of the optical thicknesses, which seems to be less effective for training than scaling by the taking the logarithm first.

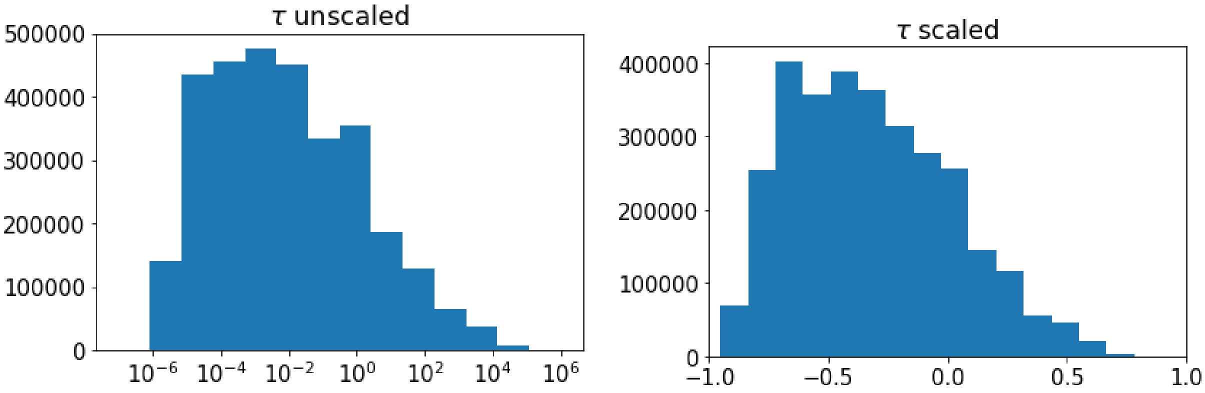


Figure 4.1: Histogram showing the distribution of τ (left) unscaled and (right) scaled of the training data set. For the scaling of τ, first the logarithm was taken and then the values were divided by a constant, chosen based on the maximum values of the logarithmic values of τ from the training, validation and test data set.

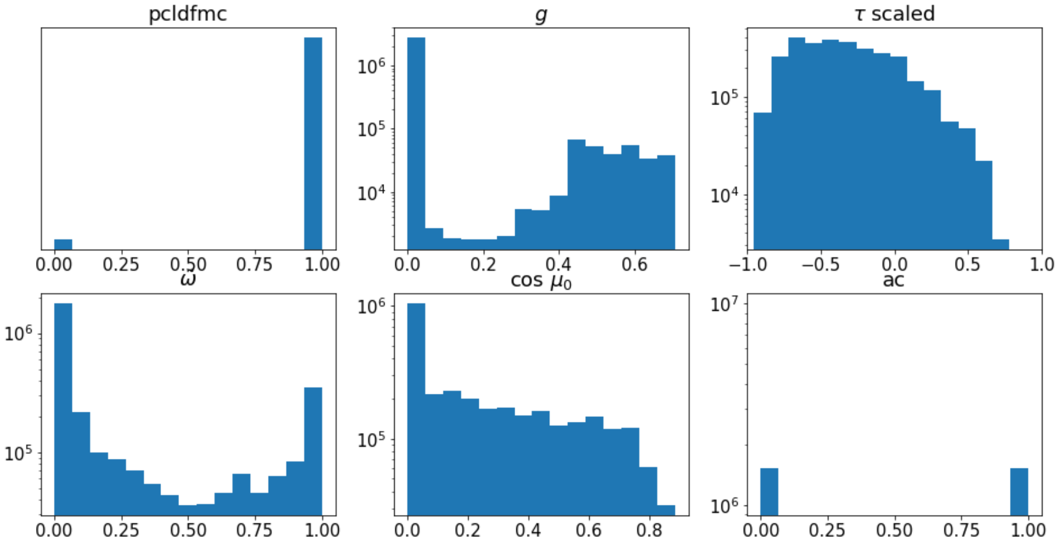


Figure 4.2: Distribution of the input variables of the initial, mixed training data set with both clear and cloudy data cases. Note that τ has been scaled as described in section 4.1.

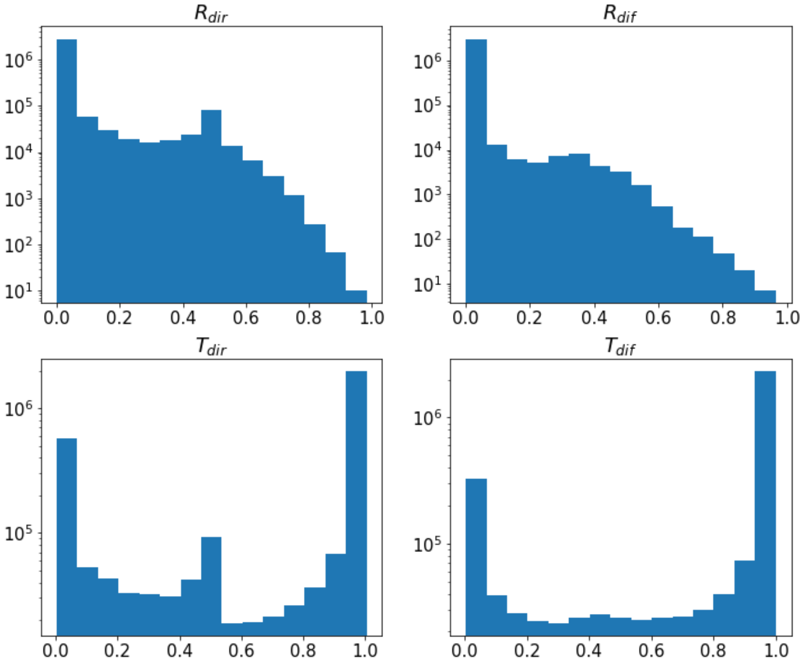


Figure 4.3: Distribution of the output data of the initial, mixed training data set with both clear and cloudy data cases.

4.2 Training and optimizing neural networks

Optimizing a neural network, i.e. finding the optimal combination of hyperparameters, is not a straightforward procedure, since the hyperparameters typically depend on one another. Different hyperparameters have been introduced in section 2.3 and will be further investigated in this section.

To compare different configurations of parameters, the neural networks are needed to be trained until their loss functions converge towards a minimum, which makes the optimization of neural network models a time-consuming procedure.

The hyperparameters of which different configurations will be presented and compared in the following sections are:

Batch size

The batch size, as introduced as size of the mini-batches in section 2.3.2, affects the computation of the gradient of the loss function. Depending on the batch size, a smaller or larger amount of samples is used to estimate the gradient, adding more or less noise its computation. The number of iterations per epoch also depends on the batch size, which both affects the computation time as well as the time it takes for the model to converge.

Activation function

Neural networks can learn nonlinearities through the usage of activation functions, which makes them an important hyperparameter.

While the four most common activations, as introduced in section 2.3.1, will be tested in section 4.2.4, it should be noted that the output layer always uses the sigmoid function. This choice has proven useful since all the values of the output variables range between 0 to 1 and other activation functions appeared to struggle with this constrain, leading to predictions outside of this range.

Network sizes

The network size, i.e. the number of layers and nodes per layer of a model, determines the number of model parameters and activation functions of the neural network. While large networks generally have the ability to learn more, they are also more likely to overfit, while a too small network might not have enough model parameters to correctly emulate the trainings data, i.e. it underfits.

The architecture of the model is also linked to its computational cost and it is therefore interesting to investigate different network sizes.

For the loss function the mean squared error has been found to work very well, while the widely used Adam optimizer has been chosen as optimization algorithm, since it does not require too much tuning, apart from the learning rate, as discussed in section 2.3.2.

The initial default training network used in the following sections, when nothing else is specified, is depicted in table 4.1. Note that since hyperparameters are interdependent, it can not be excluded that this configuration might prefer some choices of hyperparameters over others in the following tests, while a different initial configuration might lead to other results and conclusions.

Optimizer	**Loss function**	**Batch size**	**Activation function(s)**	**Network size**
Adam	Mean square error	1024	hidden layers: ReLu output layer: sigmoid	2 Layers, 50 nodes in each layer

Table 4.1: Baseline neural network model used for optimizations in the following sections

During the optimization process described in the following sections all hyperparameters are kept constant, except the one for which different options are tested. The sole exception to this rule is the learning rate, as this parameter depends strongly on the combination of the other hyperparameters.

The method for choosing the optimal learning rate for different model configurations is described in the following section 4.2.1.
In section 4.2.2 it is investigated how many and which input variables of the `reftra_sw` subroutine should be used as input for the neural network, as well as data categorization is introduced. Section 4.2.3 focuses on different batch sizes, while section 4.2.4 investigates the usage of different activation functions. Lastly, in section 4.2.5 different neural network sizes are tested and compared.

4.2.1 Learning rate

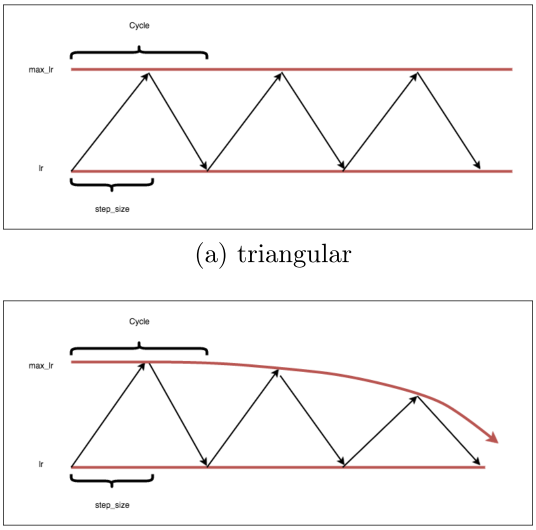
One of the most important hyperparameters is the learning rate, which determines how quickly a neural network learns, i.e. how fast the model parameters are updated. A model with a larger learning rate will learn faster, however, if the learning rate is too large, the loss function will never converge towards its minimum, limiting what can be learned, while a too small learning rate will make the model learn very slowly and might get stuck in a local minima. Optimizers with an adaptive learning rate, such as the Adam optimization algorithm, depend on a good choice of initial learning rate too.

In section 2.3.2 the idea of a varying learning rate through a learning rate schedule was discussed. One type of such learning rate schedules is the cyclic learning rate (CLR) schedule as presented by [Smith, 2015], where in order to prevent the model from getting stuck at local minima, occasionally larger learning rates are used.
This also implies that there is not just one specific optimal learning rate, but rather a range of values suitable as learning rate.
In this study the triangular and exponential cyclic learning rate have been tested. Both schedules are illustrated in figure 4.4.

In the case of triangular CLR the learning rate will change back and forth between a base learning rate and a maximum learning rate. The time it takes, i.e. the number of iterations during the training, for changing the learning rate back and forth between these two boundaries is referred to as one cycle, which is defined by the step size, which is the number of iterations it takes from changing the learning rate from one boundary to the other, i.e. half a cycle, as depicted in the figure. The base learning rate lr_{base}, the maximum learning rate max_lr and the step size δ are constants that need to be specified for the training.

Typically the step size δ is recommended to be 2 to 10 epochs long [Smith, 2015]. In this study a step size of 5 epochs, resulting in a cycle length of 10 epochs, worked well.

The exponential CLR is a variation of this schedule, where the maximum learning rate decays with an exponential factor γ^i, with $\gamma \leq 1$ and where i is the number of iteration i.e. the i'th mini-batch. For the exponential CLR γ is a constant that needs to be defined alongside lr_{base}, max_lr and δ.



(a) triangular

(b) exponential

Figure 4.4: Illustration of two cyclic learning rate schedules, one (a) triangular with constant minimum and maximum learning rates and one (b) where the maximum learnig rate decays exponentially. Figures from [CLR-github,]

Regardless whether it is chosen to use a constant learning rate, or such a learning rate schedule, the challenge of finding a good learning rate (range) remains.
For this [Smith, 2015] introduced a method for finding a good range for optimal learning rates for any neural network configuration, i.e. a learning rate range test.
The concept is to increase the learning rate after each iteration and save the loss for each learning rate over the span of a few epochs.
Thus one gets the relation between loss and learning rate, as shown in figure 4.5, where the optimal learning rate can be located where the function is steepest.

Since the model only needs to be trained for a few epochs, e.g. here it has been trained for 3, this is a quick method to find an optimal learning rate, or a range of learning rates for the CLR schedule.

The model trained in the figure is the baseline model described by table 4.1, trained on only cloudy data cases, which will be described in more detail in the next section 4.2.2.

While [Smith, 2015] suggested increasing the learning rate linearly, here the learning rate has been increased exponentially, so that the learning rate range test examines more of the smaller learning rates than of the larger ones. The learning rate has been increased from $10 \cdot 10^{-10}$ to 10.

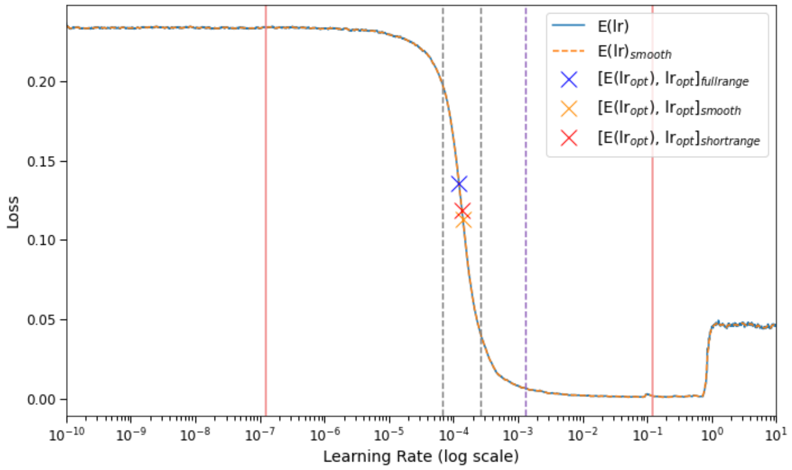


Figure 4.5: Example of the results from using a learning rate scanner, loss as function of learning rate.

In figure 4.5 the optimal learning rates found with different methods have been marked with crosses on the function. To find the steepest slope of the function (orange cross), the curve must first be smoothed, since small fluctuations of the loss can occur, especially for large learning rates, where the loss starts to diverge, as seen on the right end of the plot (blue curve). Note that this method depends on the way the function is smoothed.

A simpler approach to estimate the optimal learning rate is to calculate the mean value of the loss, and use the corresponding learning rate of this value (blue cross). However, this approach is influenced by the full range of losses, also the ones for large learning rates where the loss starts to diverge, as well as the small ones where the model does not learn anything.

Therefore another mean value is calculated, this time only inside an interval that ranges three magnitudes in each direction of the first mean value, as indicated by the vertical red lines in the figure. The new mean value of this shorter range (red cross) has been chosen as the optimal learning rate lr_{opt} for this model, which is $\sim 1.2 \cdot 10^{-4}$.

There are also different methods to estimate a good learning rate range for the CLR schedules from the figure.

One way is a visual inspection of the plot, setting the base and maximum learning rate based on where the slope of the curve starts to increase and decrease again. For the curve in figure 4.5 the boundaries can e.g. be estimated to $\sim 3 \cdot 10^{-5}$ and $\sim 9 \cdot 10^{-4}$.

Another approach is to set the boundaries as multiples of the optimal learning rate lr_{opt}. Typically something like $lr_{base} = \frac{1}{2}lr_{opt}$ and max_lr= $2 \cdot lr_{opt}$ is chosen, which are the boundaries illustrated by the dashed grey lines in figure 4.5.
Lastly, the dashed purple line shows the value of $10 \cdot lr_{opt}$, which corresponds to the purple loss shown in figure 4.6, where the initial maximum learning rate has been set to the large value of max_lr= $\gamma^i 10 \cdot lr_{opt}$, while $lr_{base} = \frac{1}{2}lr_{opt}$ as mentioned before.

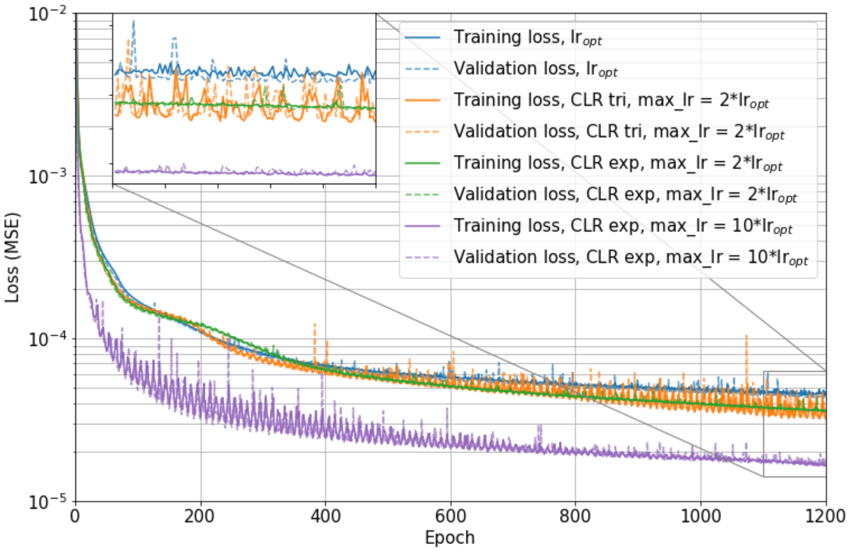


Figure 4.6: Comparison between learning curves of models trained with an optimal learning rate lr_{opt} (blue) and different learning rate schedules. Both the CLR triangular (orange) and exponential (green) are shown with an initial maximum learning rate max_lr= $2 * lr_{opt}$, as well as one exponential CLR with a start max_lr= $10 * lr_{opt}$ (purple). The training loss is depicted as solid lines, while the validation loss is shown as dashed lines. For all models the Adam optimizer has been used.

In figure 4.6 the training and validation loss, as solid and dashed lines, of the same model trained with different learning rate (schedules) is presented.
From this figure it can be seen that the models trained with the CLR schedules, both triangular and exponential, with the maximum learning rate max_lr= $2 \cdot lr_{opt}$ and max_lr= $\gamma^i 2 \cdot lr_{opt}$ respectively, are able to converge to a slightly lower loss value than the model trained with Adam and the optimal learning rate lr_{opt}. However, the larger initial maximum value of rate max_lr= $\gamma^i 10 \cdot lr_{opt}$ leads to even better results for the exponential CLR, both with respect to the final loss value, but also with regards to the speed of convergence at the beginning of the training.

From the graph it can also be seen that the usage of the CLR results in larger fluctuations of the loss between epochs, due to the large learning rates, which is why those fluctuations are still strongly present in the model trained with the triangular CLR later in the training, while the fluctuations diminish for the exponential CLR, as the maximum learning rate decreases in those schedules.

Note that the γ constant has been set to the same value for all the CLR schedules, so that the maximum learning rate max_lr=$\gamma^i 10 \cdot lr_{opt}$ is only 10% of its initial value after training the model for half the number of total epochs, i.e. after 600 epochs, which corresponds to 60 full cycles, since the step size δ is set to 5 epochs, as stated before. For this case γ can be computed from:

$$\begin{gathered}\gamma^{2\cdot 60\delta}10 \text{ max\_lr} = \text{ max\_lr} \\ \gamma = 10^{\frac{-1}{120\delta}}\end{gathered} \tag{4.1}$$

where δ is the number of iterations in 5 epochs, which can be calculated with the batch size and number of data points. In this case the batch size is 1024 and since this model was only trained on cloudy data, the data set only included 1.536.000 data points. The resulting $\delta = \frac{5\cdot 1536000}{1024} = 7500$ leads to $\gamma \approx 0.999997$.

The learning rate for all models in the following sections has been optimized for every neural network individually as described here, using the Adam optimizer together with the exponential CLR with an initial max_lr=$\gamma^i 10 \cdot lr_{opt}$, where lr_{opt} has been estimated as explained earlier.

4.2.2 Input variables and categorization

For the actual computation of the output variables, the reflectivities and transmissivities, only four of the six input variables are used in the `reftra_sw` subroutine. Those are the radiative variables introduced in section 2: the asymmetry factor g, the optical thicknesses τ, the single scattering albedo $\tilde{\omega}$, as well as the cosine of the sun zenith angle $\cos \mu_0$.
Therefore those four input variables are a natural choice of inputs for the neural network.

The logical flags *pcldfmc* and *ac* are only used to distinguish between cloudy and clear sky conditions.
It is thus interesting to test whether the inclusion of one, or both, of those variables as inputs into the neural network help the model to learn about the statistics in the training data and make better predictions of the output variables.
In this section it will therefore be tested whether the inclusion of additional input variables helps the model to recognize the different patterns of cloudy and clear sky cases.

As stated in section 4.1, do the cloudy and clear-sky conditions differ physically, with the variables in the cloudy cases taking on a wider range of possible values.
Another approach is to split the mixed data set into two, one for cloudy and one for clear sky conditions and train one neural network for each of those two categories, separately.
The advantage of this approach is, that if one category turns out to be less complex and easier to be emulated by a neural network, a simpler neural network might suffice to make the predictions for this category, reducing the computational expense.

The input variables of the training data set from figure 4.2 are shown divided into cloudy cases in figure 4.7 and clear sky cases in figure 4.8.
Similarly the output variables of the training data set from figure 4.3 are divided into cloudy conditions in figure 4.9 and clear sky conditions in figure 4.10.

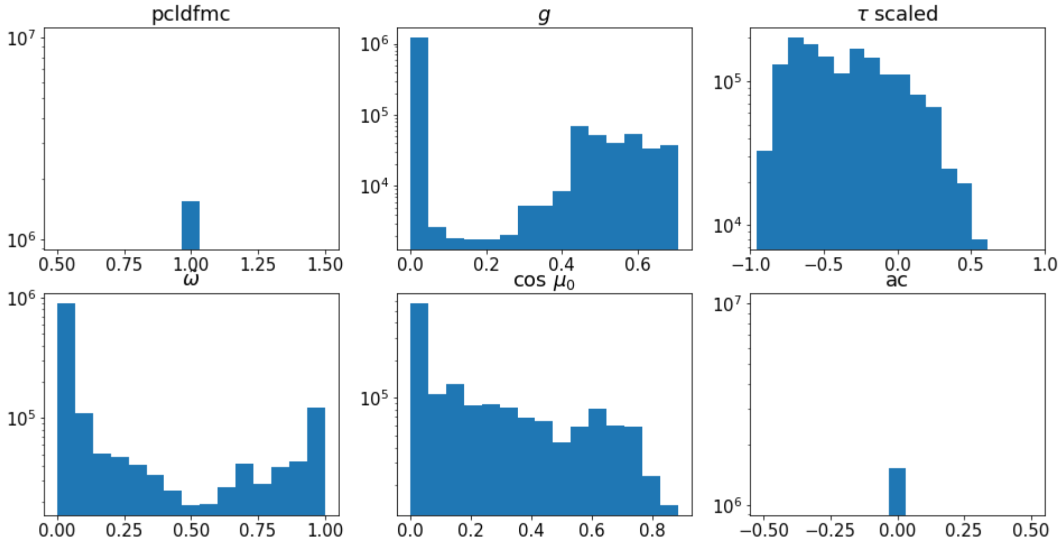


Figure 4.7: Input data of the training data set for cloudy cases (ac=0)

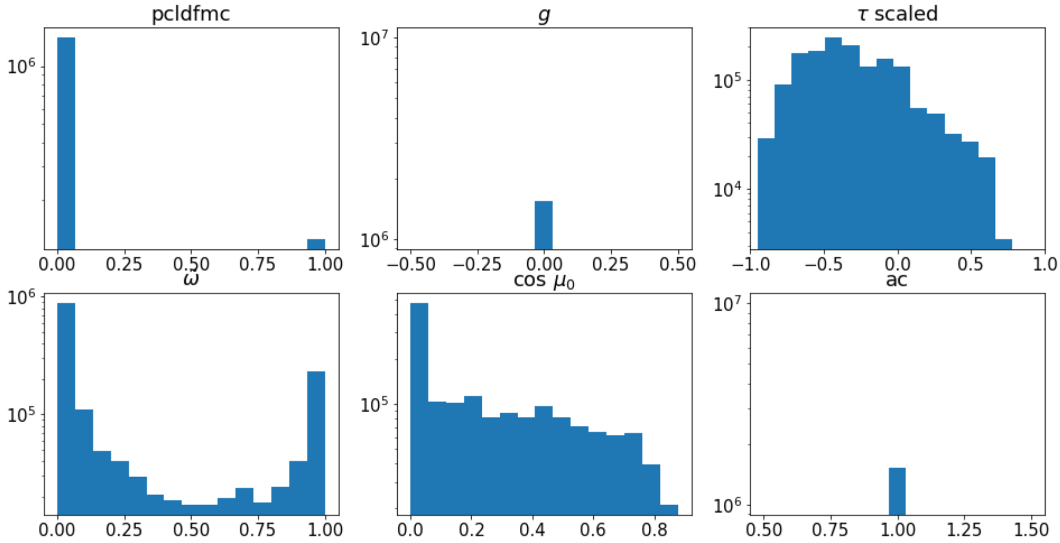


Figure 4.8: Input data of the training data set for clear sky cases (ac=1)

From the distribution of the input variables shown in the figures 4.7 and 4.8, some differences between cloudy and clear sky conditions can be seen.

First, by construction, the *ac* parameter for which the data set has been divided, is now a constant and therefore not of interest for the categorized data sets and their neural networks.

Similarly, since the data for which both $ac = 0$ and $pcldfmc = 0$ has been filtered out from the data set in section 4.1, since these cases are handled by an `if-statement`, the remaining $pcldfmc = 1$ is now also just a constant for cloudy cases.

The *pcldfmc* logical flag shows a strong favouritism towards 0 for clear sky conditions. Meanwhile, the asymmetry factor g is 0 for all clear sky cases, while spreading over a wider range for the cloudy cases.

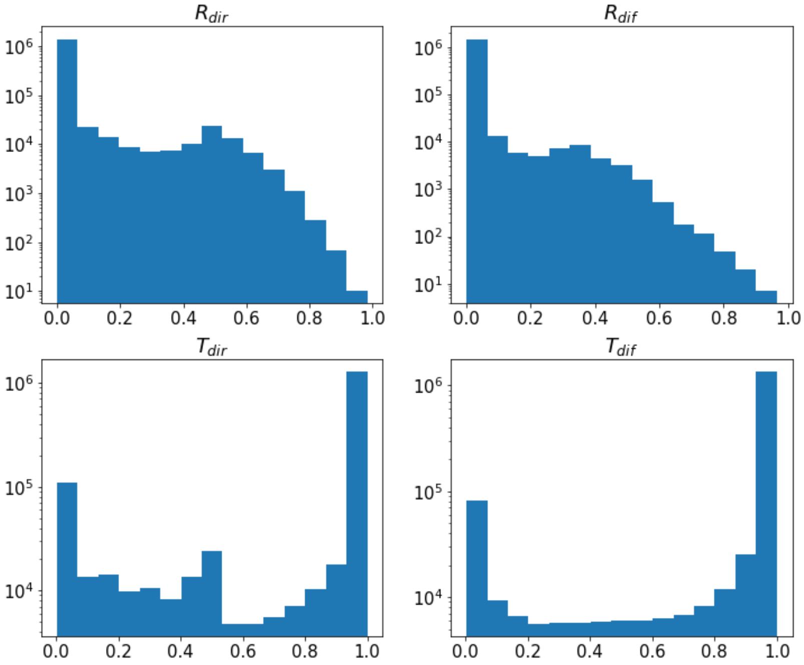


Figure 4.9: Output data of the training data set for cloudy cases (ac=0)

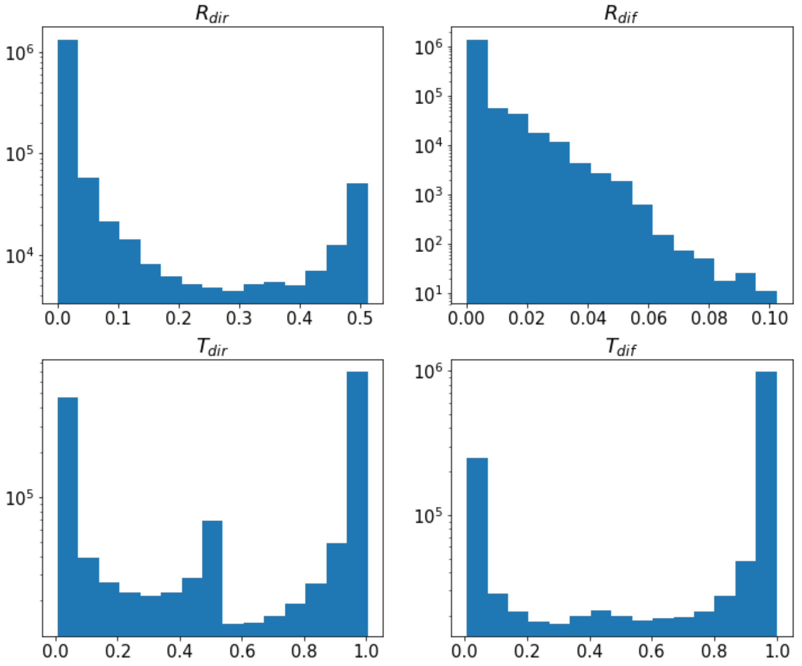
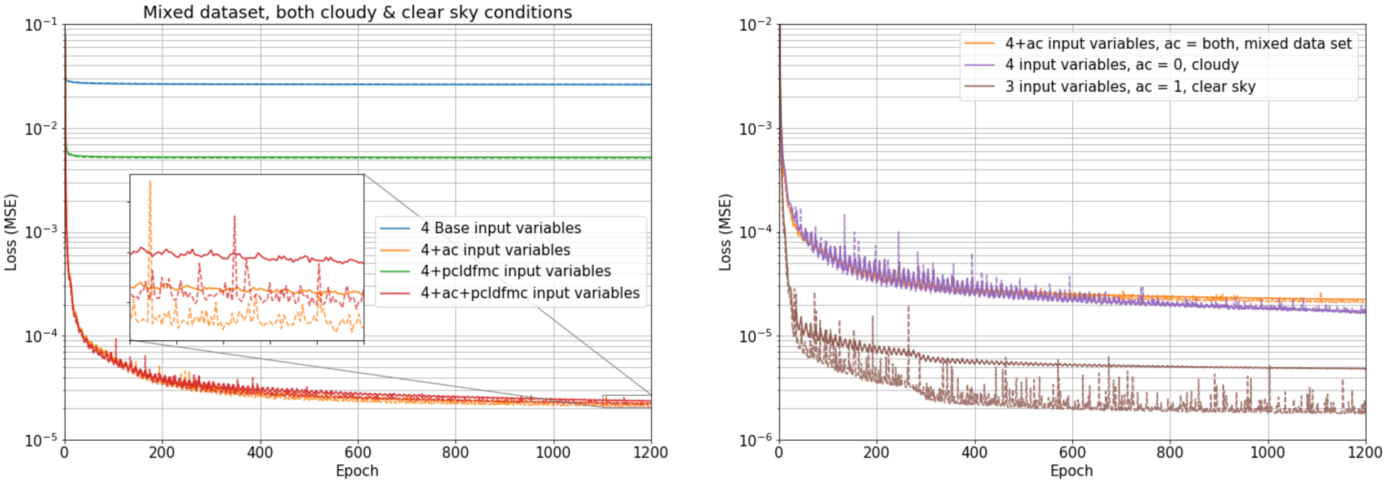


Figure 4.10: Output data of the training data set for clear sky cases (ac=1)

The distribution of the output variables depicted in the histograms in figures 4.9 and 4.10, show some differences between the magnitude of the output for cloudy and clear sky conditions.
It can be seen that the reflectivities, both the direct and especially the diffuse, for clear sky cases are very small compared to the ones in cloudy conditions. While this makes sense physically, since clouds do contribute largely to the reflection and scattering of solar radiation, this difference in magnitudes might make a data categorization into cloudy and clear sky cases useful.

In figure 4.11(a) the same model has been trained on the whole, mixed data set. The only difference is the number of input variables. The four "Base" inputs are the input variables used for the computation of the reflectivities and transmissivities: g, τ, $\tilde{\omega}$ and $\cos\mu_0$. Additionally each or both of ac and $pcldfmc$ were given as input to other neural networks, too. The learning curves show that the adding of an additional variable can lead to a significant reduction of the loss. For the networks where only one of the two logical flags was added as input variable, the model where ac was added performed much better than the one where $pcldfmc$ was added. Interestingly, the model where only ac was added also seems to perform slightly better than the model where both logical flags were added. Thus it appears that the logical flag ac helps the neural network best at detecting cloudy and clear sky conditions.



(a) different input variables for the mixed data set

(b) data categorization, mixed and divided data sets

Figure 4.11: Neural networks with different inputs. In (a) different input variables are used while training on the mixed data set, while in (b) the best performing network from (a) is compared to two networks trained on either only the cloudy or clear sky data points. Training loss is depicted as solid lines, while validation loss is shown as dashed lines.

Figure 4.11(b) shows the learning curve of this best performing model with the ac variable as added input together with two models, each trained on only the cloudy or clear sky data points of the training data set.

While also those networks share the same configurations, it has been chosen to use the same four base inputs for the network for cloudy conditions, while the network for clear sky cases only has three inputs, all of the base inputs apart from the asymmetry parameter g, since it was seen in figure 4.8, that this variable is always 0 for the clear sky data.

It can be seen that the loss of the networks trained on the divided data sets is lower than the one trained on the mixed data set, despite the inclusion of the ac parameter.

However, since the loss is computed on different data set, i.e. only on one half of the training data set for each of the data categorized cases, it is not possible to make final conclusions about the performance of the neural networks based on the learning curves alone.

To make a reasonable comparison between the models trained on different data sets, the third independent data set, the test data is also divided into cloudy and clear sky cases. Afterwards the neural networks are used to make predictions based on these test data sets, i.e. values for the cloudy test cases are predicted by the model trained on the mixed data set, as well as by the model trained only on cloudy data cases. Similarly the clear sky test data is used to make predictions and evaluations of the network trained on the mixed data set and the one trained only on the clear data cases.

Instead of comparing the loss, two other statistical measures will be compared: the Pearson correlation coefficient r and the root mean square error RMSE, defined as :

$$r = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}\left(\hat{y}_i - \overline{\hat{y}}\right)^2}\sqrt{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2}} \tag{4.2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{4.3}$$

where the overbar indicates the average of the variable, the hat denotes the predicted values and y_i denotes the true values as in section 2.3.2. Note however, that $\hat{y}_i$ now indicates the individual i'th output variable, rather than a whole vector, i.e. there will be a measures calculated for each of the four output variables.

Those two measures will be used in the following sections to compare the neural network predictions to the true values of the test data set.

The results for the prediction of the test data with the three models from figure 4.11(b) are shown in table 4.2.

From the table it can be seen that the neural networks trained separately on cloudy or clear sky conditions are better at predicting all output variables in comparison to the model trained on the mixed data set. Thus the data categorization has been used on the training, validation and test data set and two types of neural networks have been trained and optimized from here on: one for cloudy conditions with 4 input variables and one for clear sky conditions with 3 input variables as described earlier.

Neural Network	Stat	R_{dir}	R_{dif}	T_{dir}	T_{dif}
Predictions for cloudy sky (ac = 0) test data:					
4 Inputs +ac	r	0.99764	0.99565	0.99977	0.99994
mixed data set	RMSE	0.00718	0.00517	0.00637	0.00280
4 Inputs cloudy	r	0.99894	0.99774	0.99991	0.99997
(ac = 0)	RMSE	0.00482	0.00373	0.00395	0.00185
Predictions for clear sky (ac = 1) test data:					
4 Inputs +ac	r	0.99976	0.98703	0.99990	0.99999
mixed data set	RMSE	0.00249	0.00097	0.00637	0.00203
3 Inputs clear sky	r	0.99985	0.99896	0.99994	0.99999
(ac = 1)	RMSE	0.00190	0.00028	0.00469	0.00126

Table 4.2: Pearson correlation coefficient (r) and root mean square error (RMSE) for the three neural networks shown in figure 4.11(b) . The predictions have been carried out separately for the clear sky and cloudy sky test data set.

Another interesting discovery that can be made by inspecting the learning curve of the clear sky neural network in figure 4.11(b) is that for this model the validation loss is actually lower than the training loss.
Since the validation loss is measured after each end epoch, in contrast to the training loss which is measured during the epoch, it is possible for the validation loss to be lower than the training loss, however, the validation loss in 4.11(b) is much lower than the training loss which can not be accounted by this.

In section 4.1 the creation of the data sets were described, taking care to randomly sample data points from a huge initial data set, to create independent data sets. One explanation for a much lower validation loss could be that the randomly selected data for the validation data set consists of easier to predict data cases.

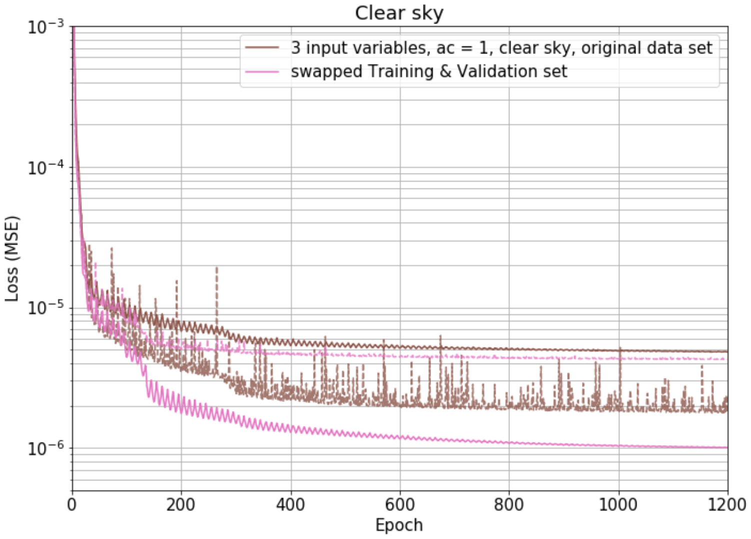


Figure 4.12: Learning curves for neural networks trained on clear sky conditions only, one trained on the original training and validation data set, while for the other the training and validation data set have been swapped.

This theory is supported by the learning curves shown in figure 4.12, which show the same model from before, with 3 input variables trained on the clear sky data points of the training data set, as well as a model with the same configuration, but this time the training and validation data set have been swapped, i.e. the validation data has been used for training, while the training data has been used to validate against. As can be seen in the figure, the losses for this swapped test model are the opposite from before, now the training loss is lower than the validation loss and actually also lower than the validation loss from the model before.

While both scenarios are not desirable, it has been chosen to swap the training and validation data for clear sky cases only from now on. Substituting the two data sets for clear sky cases gives the opportunity to avoid uncontrolled overfitting, as the loss on the more challenging data set is expected to rise if the neural network starts to learn patterns which only are included in the easier data set.
The data distributions of the initial validation data set, similar to those presented for the initial training data set earlier in this section, are included in Appendix B.

4.2.3 Batch size

To test whether the batch size affects the neural network's performance, networks with different batch sizes have been trained for cloudy and clear sky conditions. Figure 4.13 shows the results of the models that have been trained with batch sizes 256, 512, 1024 and 2048.

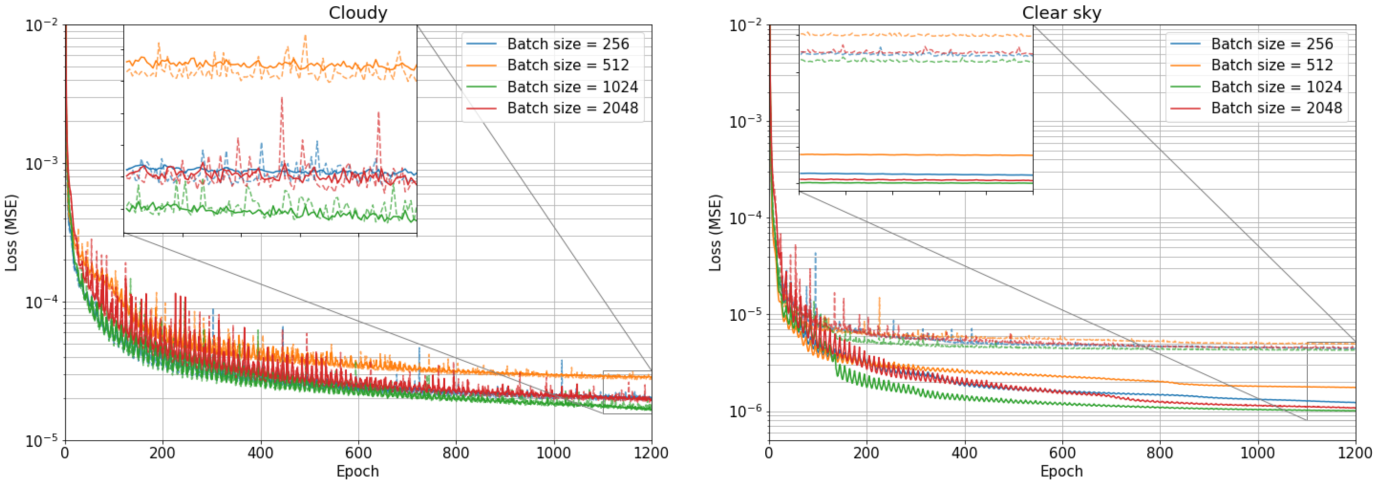


Figure 4.13: Learning curves for neural networks trained with different batch sizes for (left) cloudy cases and (right) clear sky cases. The training loss is depicted as solid lines, while the validation loss is shown as dashed lines.

For both, cloudy and clear sky cases, the batch sizes result in the same ordering of lowest loss on the training and validation data set. There is no systematic relation between batch size and loss, the smallest and largest batch sizes, 256 and 2048, result in similar loss values, while a batch size of 1024 performs slightly better, while a batch size of 512 performs worst for both categories. Since there is no significant difference between the usage of the other three batch sizes, it is continued to use the batch size 1024 as before.

4.2.4 Activation functions

The four commonly used activation functions introduced in section 2.3.1 are tested for clear and cloudy cases as well. It should be noted, however, that the four activation functions only have been tested for the hidden layers of the neural network, while the output layer always uses the sigmoid function, as described in the beginning of section 4.2.

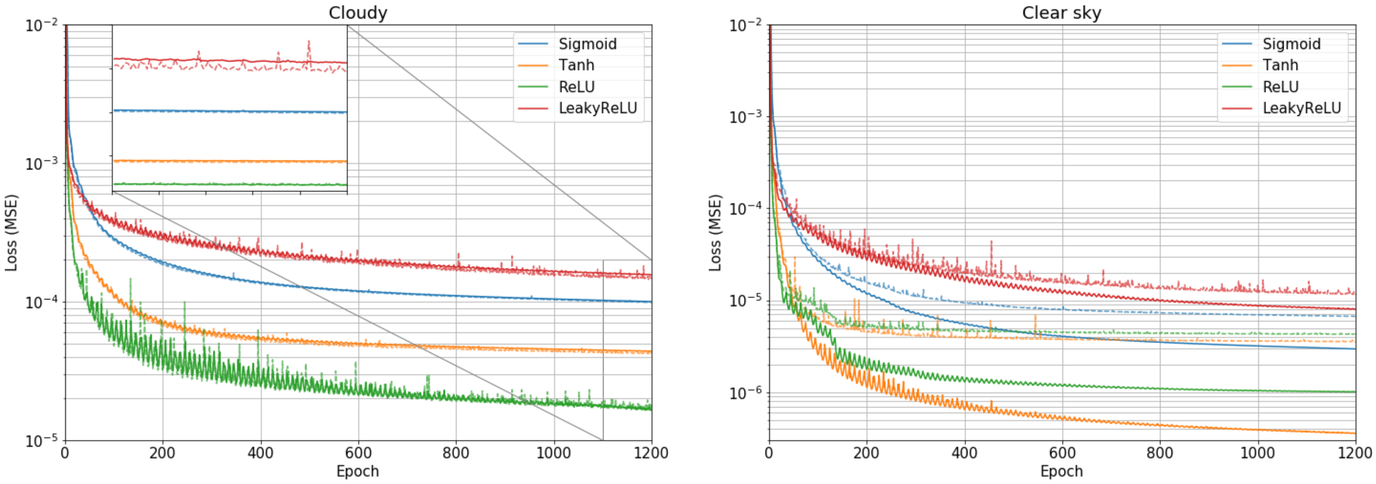


Figure 4.14: Learning curves for neural networks trained with different activation functions for (left) cloudy cases and (right) clear sky cases. The training loss is depicted as solid lines, while the validation loss is shown as dashed lines. For the different activation functions, their default values have been used, e.g. α=0.3 for LeakyReLU.

From the resulting learning curves shown in figure 4.14, some differences can be seen for neural networks trained with cloudy and clear sky data, respectively. While the ReLU activation performed best for cloudy cases, the tanh activation performed better for clear sky conditions. In both cases the leakyReLU and sigmoid activations performed worst. Based on those results, the ReLU function will be continued to be used for the hidden layers for networks trained on cloudy data, while the tanh function will be used for networks trained on clear sky samples.

4.2.5 Small and large Neural Networks in comparison

In total 9 different network sizes were tested, combinations of 1 to 3 layers with either 25, 50 or 100 nodes in each layer, for both cloudy and clear sky cases. For the networks training on cloudy conditions, the ReLU function was used as activation function of the hidden layers, while the tanh activation was used for the networks training on clear sky samples.

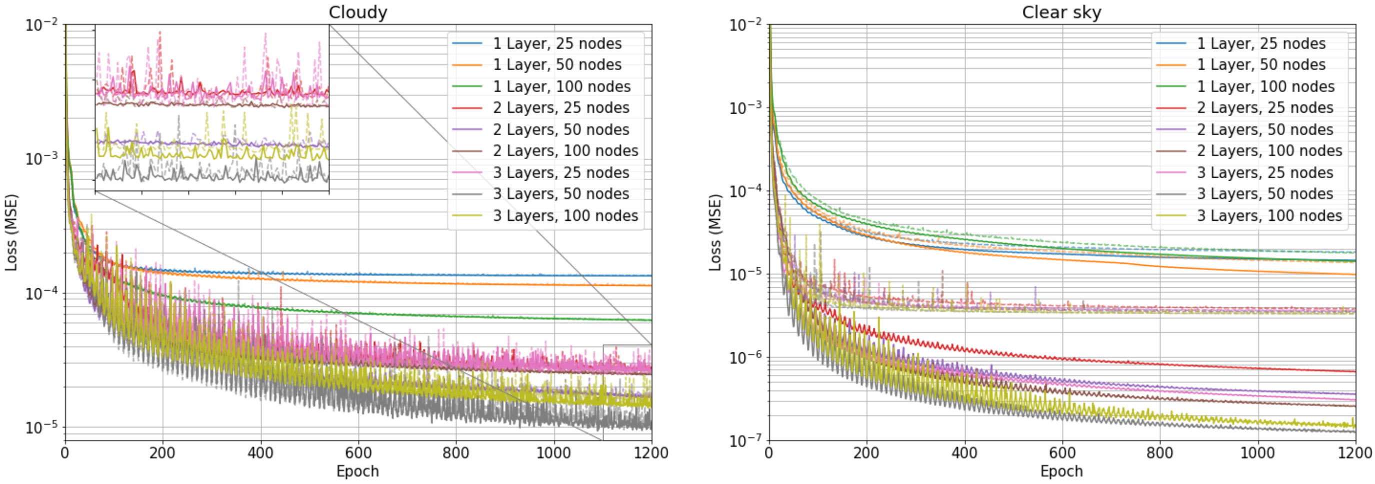


Figure 4.15: Learning curves for neural networks trained with different network sizes for (left) cloudy cases and (right) clear sky cases. The training loss is depicted as solid lines, while the validation loss is shown as dashed lines. Note that for the activation function of the hidden layers ReLU is used for cloudy cases, while tanh is used for clear sky conditions.

From the learning curves in figure 4.15 it can be seen that larger neural networks generally result in smaller loss values, i.e. they perform better on the training and validation data. It does however seem like there is a risk of overfitting by the largest networks. Based on those learning curves good choices seem to be neural networks with 2-3 layers and 25-50 nodes per layer, though the largest combination of 3 layers and 50 nodes might be overfitting.

To decrease the risk of overfitting, the best models for each network size, i.e. the models with the lowest validation loss, are used to make predictions and evaluations with the test data set.

The loss on the test data of these models is shown in figure 4.16, where it has been chosen to plot the loss as function of number of model parameters.

The number of model parameters is not the only factor determining the time needed to make predictions with a neural network, but work nevertheless as a good first indicator.

From figure 4.16 it can be seen that the models with only one layer perform worse than the multi layered networks, even if they have more model parameters, as e.g. is the case for the model with 1 layer and 100 nodes compared to the 2 layer model with 25 nodes in each layer. Interestingly, the two largest models, with many more model parameters than the other models, also perform worse than some of the smaller models.

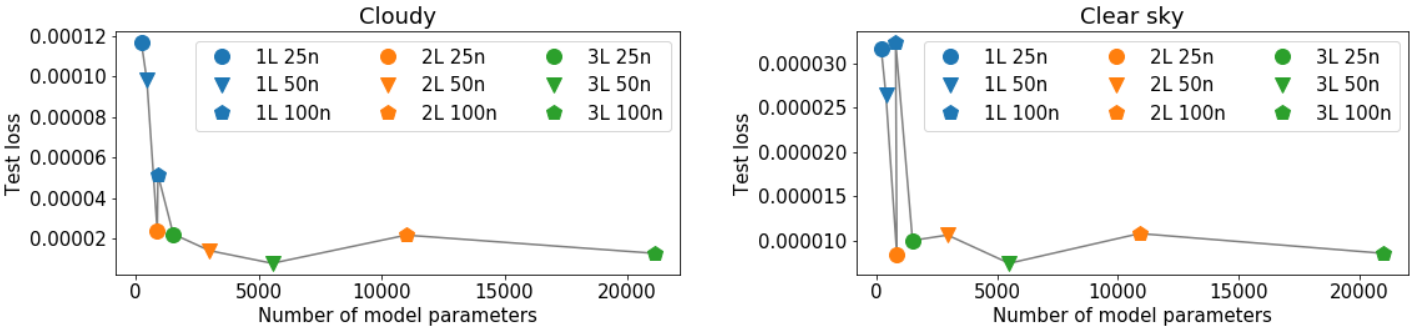


Figure 4.16: The test loss as a function of number of model parameters for neural networks with different network sizes trained for (left) cloudy and (right) clear sky conditions. The number of layers is indicated by color, while the number of nodes per layer is denoted by the symbol. The abbreviation in the legend stands for the number of layers and number of nodes per layer, e.g. 2L50n = 2 layers, 50 nodes per layer.

Based on this figure, it seems like the model with 3 layers and 50 nodes is the best performing for both cloudy an clear sky samples. For clear sky conditions the smaller model with 2 layers and 25 nodes seems to be a good second choice, since it has the second smallest loss, while having less model parameters than the other multi layer models. The performance of the models for cloudy cases seems to be more proportional to the number of model parameters than for the clear sky samples.

The performance of the models shown in figure 4.16 is further investigated in table 4.3 and 4.4, where the predictions of the individual output variables for the test data set are compared. For this the measures from equation 4.2 and 4.3 are used.
The diffuse reflectivity R_{dif} turns out to be the variable, which is predicted worst for both clear sky and especially cloudy data cases.
As alternative choices for the cloudy samples, the second best performing model with 2 layers and 50 nodes as well as the model with 2 layers and 25 nodes is chosen, to investigate the abilities of the smaller models in the WRF model.

Model (ac=0)	Stat	R_{dir}	R_{dif}	T_{dir}	T_{dif}
1 Layer, 25 nodes	r	0.99123	0.98640	0.99902	0.99981
	RMSE	0.01380	0.00911	0.01301	0.00498
1 Layer, 50 nodes	r	0.99263	0.98879	0.99913	0.99990
	RMSE	0.01268	0.00829	0.01225	0.00371
1 Layer, 100 nodes	r	0.99595	0.99260	0.99962	0.99995
	RMSE	0.00940	0.00673	0.00809	0.00264
2 Layers, 25 nodes	r	0.99837	0.99695	0.99981	0.99995
	RMSE	0.00596	0.00433	0.00577	0.00272
2 Layers, 50 nodes	r	0.99894	0.99774	0.99991	0.99997
	RMSE	0.00482	0.00373	0.00395	0.00185
2 Layers, 100 nodes	r	0.99834	0.99619	0.99987	0.99997
	RMSE	0.00605	0.00485	0.00474	0.00207
3 Layers, 25 nodes	r	0.99839	0.99637	0.99985	0.99996
	RMSE	0.00593	0.00475	0.00510	0.00225
3 Layers, 50 nodes	r	0.99940	0.99836	0.99996	0.99999
	RMSE	0.00362	0.00318	0.00254	0.00119
3 Layers, 100 nodes	r	0.99899	0.99735	0.99994	0.99998
	RMSE	0.00470	0.00403	0.00322	0.00147

Table 4.3: Pearson correlation coefficient (r) and root mean square error (RMSE) for neural networks with different network sizes trained on cloudy conditions. The best predictions are highlighted in red, while the worst performances are written in blue.

Model (ac=1)	Stat	R_{dir}	R_{dif}	T_{dir}	T_{dif}
1 Layer, 25 nodes	r	0.99945	0.99622	0.99972	0.9999896
	RMSE	0.00369	0.00053	0.01048	0.0018
1 Layer, 50 nodes	r	0.99957	0.99826	0.99976	0.9999953
	RMSE	0.00327	0.00035	0.00967	0.00119
1 Layer, 100 nodes	r	0.99957	0.99907	0.99970	0.9999973
	RMSE	0.00326	0.00026	0.01087	0.00091
2 Layers, 25 nodes	r	0.99984	0.99879	0.99992	0.999999
	RMSE	0.00198	0.0003	0.00544	0.00054
2 Layers, 50 nodes	r	0.99986	0.99977	0.99990	0.9999996
	RMSE	0.00185	0.00013	0.00624	0.00037
2 Layers, 100 nodes	r	0.99986	0.99978	0.99990	0.9999998
	RMSE	0.00183	0.00013	0.00631	0.00025
3 Layers, 25 nodes	r	0.99983	0.99890	0.99991	0.9999995
	RMSE	0.00205	0.00028	0.00597	0.00037
3 Layers, 50 nodes	r	0.99986	0.99978	0.99993	0.9999998
	RMSE	0.00185	0.00013	0.00512	0.00021
3 Layers, 100 nodes	r	0.99986	0.99984	0.99992	0.9999998
	RMSE	0.00188	0.00011	0.00554	0.00022

Table 4.4: Pearson correlation coefficient (r) and root mean square error (RMSE) for neural networks with different network sizes trained on clear sky conditions. The best predictions are highlighted in red, while the worst performances are written in blue.

4.3 The implemented Neural Networks

Based on the previous described testing and optimization procedures, 3 combinations of neural networks for cloudy and clear sky samples have been selected and are presented in table 4.5.
In this section those three configurations are compared based on their performance on the test data, for cloudy and clear sky cases combined.

	Network size for cloudy cases	Network size for clear sky cases
Model 1	2 layers with 25 nodes each	2 layers with 25 nodes each
Model 2	2 layers with 50 nodes each	2 layers with 25 nodes each
Model 3	3 layers with 50 nodes each	3 layers with 50 nodes each

Table 4.5: Overview of the selected model combinations for to be implemented into WRF. The models for the cloudy cases use the ReLU activation and 4 input variables (g, τ, $\tilde{\omega}$, $\cos\mu_0$), while the models for the clear cases use the tanh activation function and 3 input variables (τ, $\tilde{\omega}$, $\cos\mu_0$).

The statistical measures from equation 4.2 and 4.3 for the predictions of the test data are shown in table 4.6. Since model 3 consists of the two largest and best performing networks from the previous section, it is not surprising that it does perform best on the test data set. Meanwhile model 1 has the smallest number of model parameters, while model 2 serves as compromise between the two others.

Neural Network	Stat	R_{dir}	R_{dif}	T_{dir}	T_{dif}
Model 1	r	0.99915	0.99701	0.99990	0.999983
	RMSE	0.00444	0.00307	0.00560	0.00196
Model 2	r	0.99942	0.99778	0.99993	0.999992
	RMSE	0.00368	0.00264	0.00475	0.00136
Model 3	r	0.99964	0.99839	0.99995	0.999997
	RMSE	0.00288	0.00225	0.00405	0.00085

Table 4.6: Pearson correlation coefficient (r) and root mean square error (RMSE) for the three neural network models shown in table 4.5 . Predictions were made for both clear and cloudy cases.

Additionally, the predictions of the four output variables are presented as function of the true values of the test data set as 2D histograms in figure 4.17. The colors indicate the data point density, note the logarithmic colorbar. These plots make a visual comparison between the predicted output variables possible. The general tendency is that the spread of data points becomes less comparing the results from model 1 to model 2 and model 2 to model 3. A good example of this is the spread for the direct transmissivities T_{dir}, or the underestimation of the diffuse transmissivities T_{dif} by the model 1 and 2, while model 3 has a more even spread for small T_{dif} values.

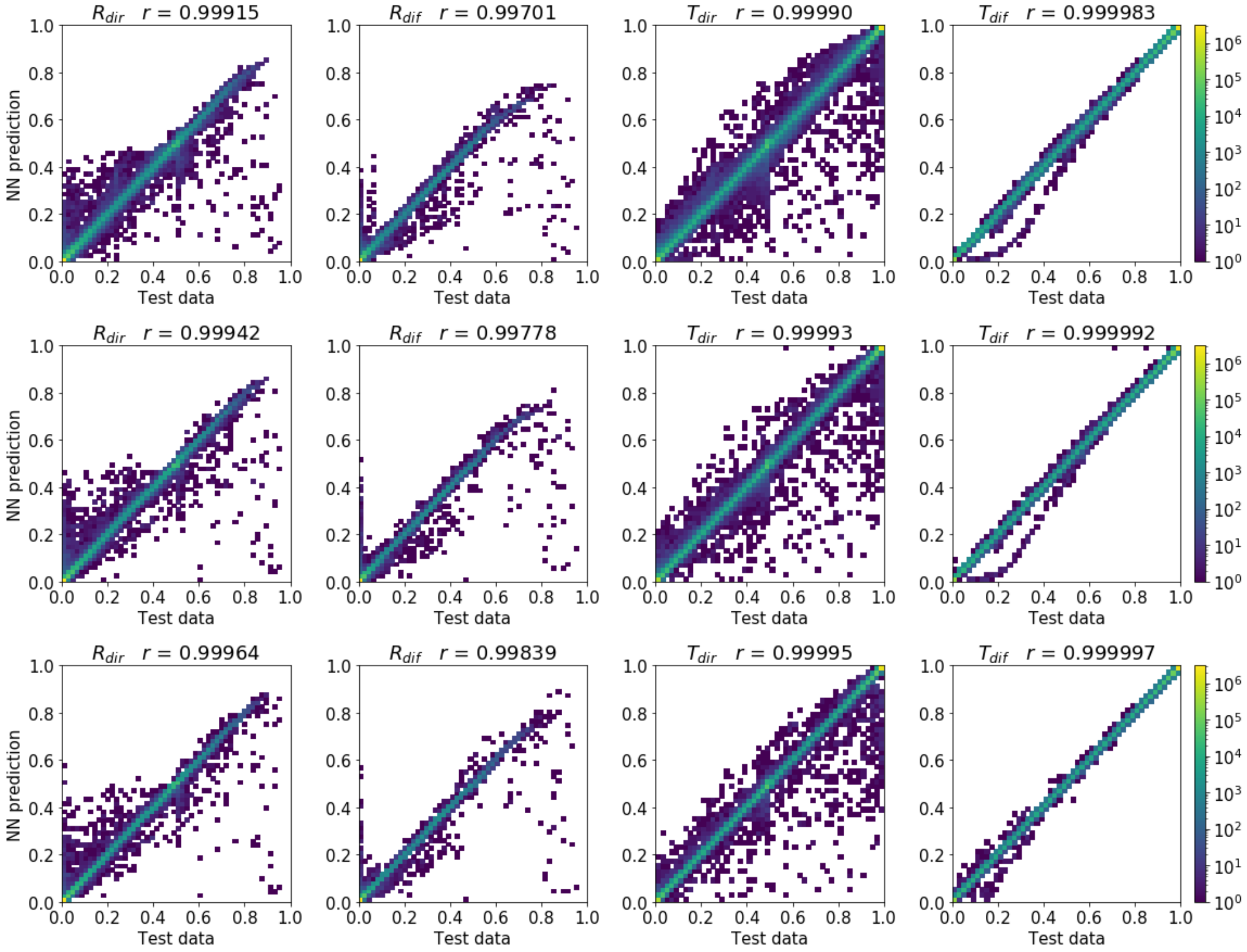


Figure 4.17: 2D histograms showing the correlation between the predictions of the four output variables and the true values of the test data set. The colors show the density of the data points (note the logarithmic colorbars). The first row shows the predictions made by model 1, the second row the predictions by model 2 and the third row the ones made by model 3. The correlation values r are the same as in table 4.6

5 Results

Until now the performance of the neural networks has only been evaluated with the sampled data sets. While the usage of a third independent test data set gives a good first indication of how well the the model can predict the reflectivities and transmissivities, this corresponds to the performance of prediction for a single time step in the WRF model. It is therefore important to test the behaviour of the neural networks in longer simulations, to investigate whether they have systematic errors which can lead to feedback mechanisms and divergence of the model run.

Additionally, since the output of the WRF model does not contain the reflectivities and transmissivities calculated in the shortwave parameterization, but rather the shortwave fluxes computed from those parameters, the comparison of these fluxes and other physical variables in the WRF output will show how strongly those are influenced by the errors of the predicted reflectivities and transmissivities.

The methods used to compare the different shortwave radiation parameterization schemes will be described in section 5.1. In section 5.2 the comparison of the schemes is presented with those methods, while the computational efficiency of the different shortwave schemes is discussed in section 5.3.

5.1 Comparison method and case studies

A total of four different weather scenarios, i.e. case studies for four seasons, have been simulated with seven different shortwave schemes. The simulations done with the original RRTMG-fast scheme will serve as reference for each scenario, where those predictions will represent the true values. While this might not be accurate in reality, it is reasonable for the comparison of performance of the neural networks, since they have been trained with data sets created by the RRTMG-fast scheme and will therefore try to emulate it.

The modified variants of the RRTMG-fast scheme, where the three neural network models, described in section 4.3, replace the computations in the `reftra_sw` subroutine, will be denoted as schemes NN 1, 2 and 3, respectively in this section.

The neural networks were implemented in `reftra_sw` with the Fortran-Keras Bridge (FKB) [Ott et al., 2020], where the `if-statement`, described in section 4.1, has been kept in the code to save computation time. A second condition has been added in the routine with the neural networks, which will scale the computed reflectivities and transmissivities to sum to 1, in case the sum of the two direct, or the two diffuse parameters exceeds 1, which would be a nonphysical result.

In addition to the simulations with those schemes, the schemes RRTMG, New Goddard and Dudhia, previously introduced in section 3.3, will also be used to make predictions for the four case studies.

This is done to be able to identify how much of the observed divergences between the predictions by the original RRTMG-fast and the NN schemes can be accounted to as a result of simulating a chaotic system, and how much might be a result of some systematic errors of the neural networks.

The case studies consist of four 4-day (96 hour) simulations starting at 0 UTC on the 1st of January, April, October and December 2019, one simulation for each season, respectively. Note that time periods have been chosen, which have not been part of the previous sampled data sets from the 12 days of 2018, to make sure that the models are tested on the performance on unknown data.

The same domain as for the creation of the data sets described in section 3.1.1 is used for all simulations.

The most interesting variable from the output of the WRF model to compare is the shortwave radiation flux at the surface, which is calculated from the reflectivities and transmissivities computed by the reftra subroutine, i.e. by the neural networks in the three new NN schemes.

Additionally the 2 meter temperature, as well as the sensible and latent heat fluxes at the surface are compared, since the shortwave radiation influences the heating rates of the atmosphere and surface.
The influence of clouds and the cloud fraction will also be discussed where adequate.

While comparing variables on geographically plots gives a good intuitive picture of the difference in predictions, it is difficult to compare all study cases like this.
In section 5.2 it will therefore be first focused on such plots for the summer cases (additional plots for the other simulations can be seen in Appendix C).

Afterwards the performance of the schemes on the four season cases will be compared through the comparison of the root mean square error (RMSE) and the Pearson correlation coefficient r as function of time for the different variables.

Note that in contrast to the definitions of the RMSE and correlation r given in equation 4.3 and 4.2, those measures are now applied on geographical 2-dimensional data, with latitude and longitude coordinates.
While the RMSE gives an indication of the magnitude of the mean difference between the predicted values of the original RRTMG-fast and the other schemes, the correlation coefficient r will compare the similarity of two fields, so it can be be thought of as pattern correlation.

5.2 WRF predictions with different radiation schemes

As stated in the previous section, it is first focused on the summer case study, a 4-day (96-hour) simulation with the intital start time 1.7.2019 0 UTC, for which spatial differences between the predictions of various variables are investigated.

For all comparisons the predictions made with the RRTMG-fast scheme are shown on the leftmost plot, while the anomaly fields, showing the differences to the predictions of the other six schemes, calculated as $pred_{scheme} - pred_{RRTMG-fast}$, are located on the right. Thus, positive anomalies will indicate that the scheme on the right predicts larger values than the RRTMG-fast scheme.

The first variable to be compared is the downward shortwave flux at the surface. In figure 5.1 the differences between RRTMG-fast and all other schemes is shown, once valid for 12 hours into the simulation in the upper plots, and once valid for 84 hours after the initial time.

This means the predictions are valid for 12 UTC on the first and fourth day of the simulation, respectively. From the variable field on the left it can be seen that at this time the whole domain is exposed to sun light in contrast to e.g. morning hours, indicated by the high values of the flux in red colors.

The blue patterns, showing no or low amounts of incoming shortwave radiation at the surface, indicate the cloud patterns in the atmosphere at the given time, since it is due to those clouds that no or only small amounts of solar radiation reach the surface.

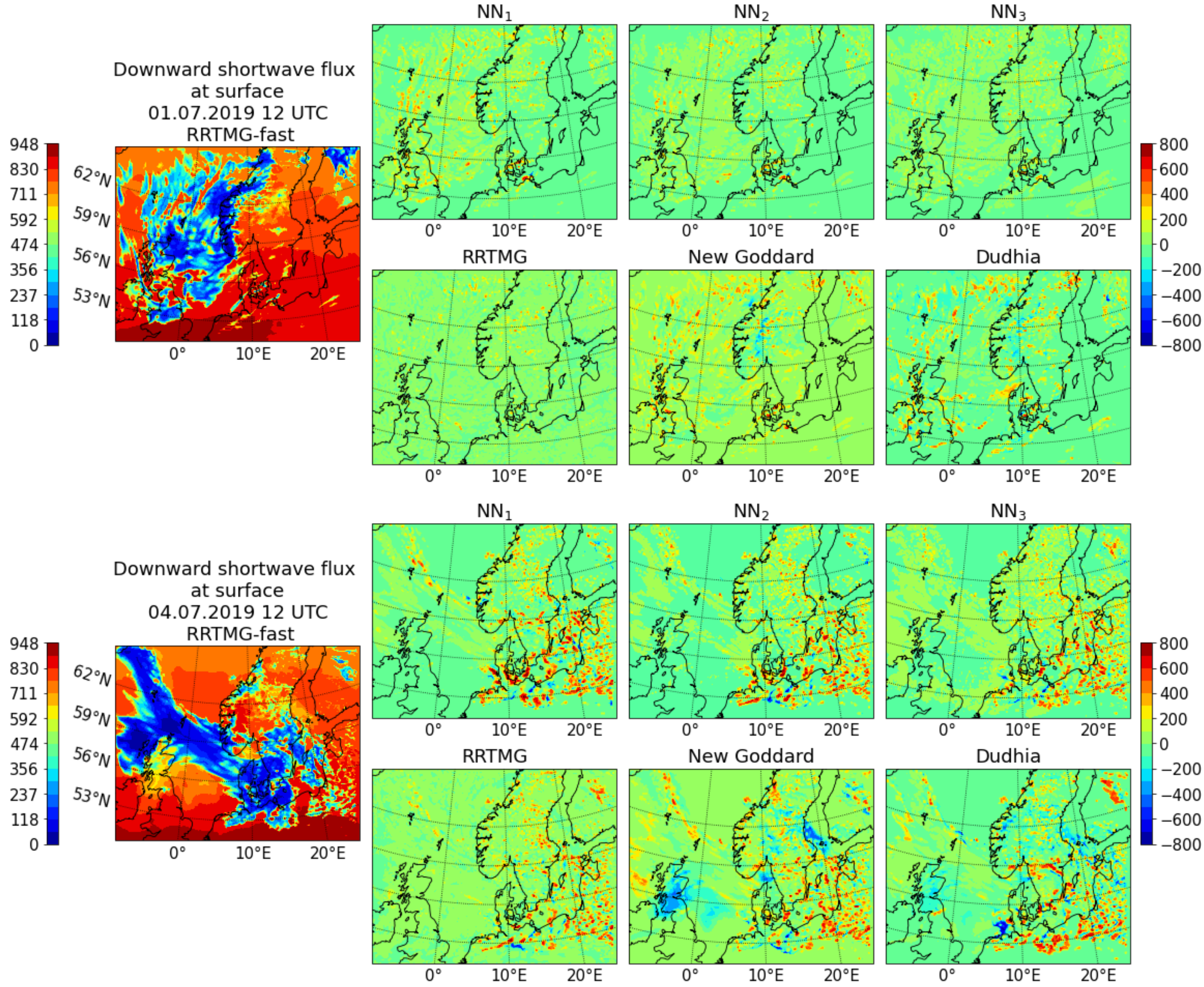

NN$_1$
NN$_2$
NN$_3$
Downward shortwave flux at surface 01.07.2019 12 UTC RRTMG-fast
RRTMG
New Goddard
Dudhia
Downward shortwave flux at surface 04.07.2019 12 UTC RRTMG-fast
948
830
711
592
474
356
237
118
0
62°N
59°N
56°N
53°N
0°
10°E
20°E
800
600
400
200
0
−200
−400
−600
−800

Figure 5.1: Plots showing the predictions made for the downward shortwave radiation at the surface (in W/m^2) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name are depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

For all six compared schemes it can be seen that the difference between their prediction and the prediction of RRTMG-fast scheme increased with the simulation time. The anomalies for the New Goddard and Dudhia scheme are generally larger than the ones for the NN schemes and the RRTMG.

The anomalies of the NN schemes show the same tendency as in their performance evaluation on the test data in section 4.3: NN 3, with the largest neural networks, performs best, while NN 1 with the the smallest neural networks performs worst, but still better than the Dudhia and New Goddard scheme. Note that "better" here means that the scheme's predictions are more similar to the ones of the RRTMG-fast, than it is the case for the other two, but does not necessarily imply that the predictions are more true than the others.

It can also be seen that the spatial distribution of the anomalies is very similar for the NN schemes and the RRTMG scheme, where NN 3 seems to also have the same magnitudes of anomalies as the RRTMG. Since the RRTMG and the RRTMG-fast use the same approximations, this implies that the neural network is good at emulating the RRTMG-fast.

Comparing the locations of the larger anomalies with the plot of the shortwave flux on the left indicates that the largest errors occur mainly in areas, where there are sharp contrast in the magnitude of the flux, i.e. at the boarders of clouds.

Figures 5.2 - 5.4 show the same two times, but for the 2 meter temperature, surface sensible heat flux and surface latent heat flux. For those variables the same tendencies can be recognized, i.e. after 84 hours the anomalies are smaller, than after 84 hours and the NN 3 and RRTMG schemes are the ones that perform best, while the New Goddard and Dudhia scheme show larger differences. Even the spatial location of the larger errors is very similar for all four variables.

Since the location of clouds has such a large impact on the anomalies of all variables, the cloud area fraction has been investigated. This fraction is part of the WRF output and is defined for the full 3-dimensional grid, i.e. for all vertical layers. The (vertical) mean value of this variable is shown in figure 5.5.

It can be noticed, that the areas with large values of the mean cloud area correspond mostly to the the blue areas of small shortwave flux values in figure 5.1, but not all of them, as there are also areas where the mean cloud area fraction is low and the shortwave flux is small as well. Meanwhile, the area with the largest anomalies is located where there is a relatively low mean cloud fraction.

Still, the mean cloud area shows also the same tendencies as before, with the similar spatial distribution of larger anomalies that are bigger for a time later in the simulation.

The largest anomalies after 84 hours in the shortwave flux in figure 5.1 are located in the south-east. It can be noted that the anomalies are not uniformly, but scattered around in smaller packages, however, this is also true for the distribution of the shortwave flux itself, as can be seen in the figure on the left.

The same feature can be seen for the shortwave flux after 36 and 60 hours of simulation time, presented in figure 5.6, while the corresponding 2m temperature is shown in figure 5.7, where the anomalies are not as sharply distributes as for the fluxes, but in the same general area.

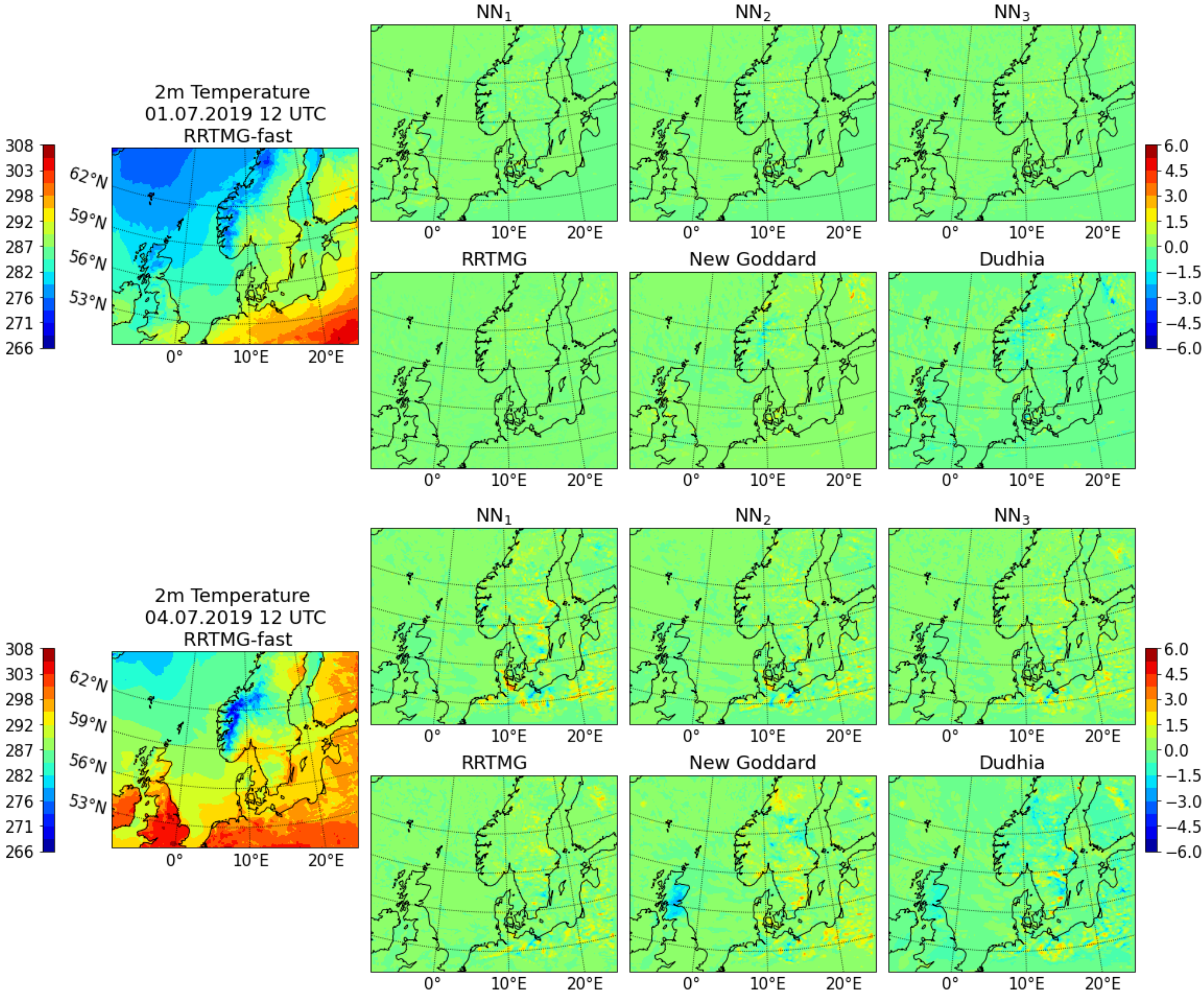


Figure 5.2: Plots showing the predictions made for the 2 meter temperature (in K) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name are depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

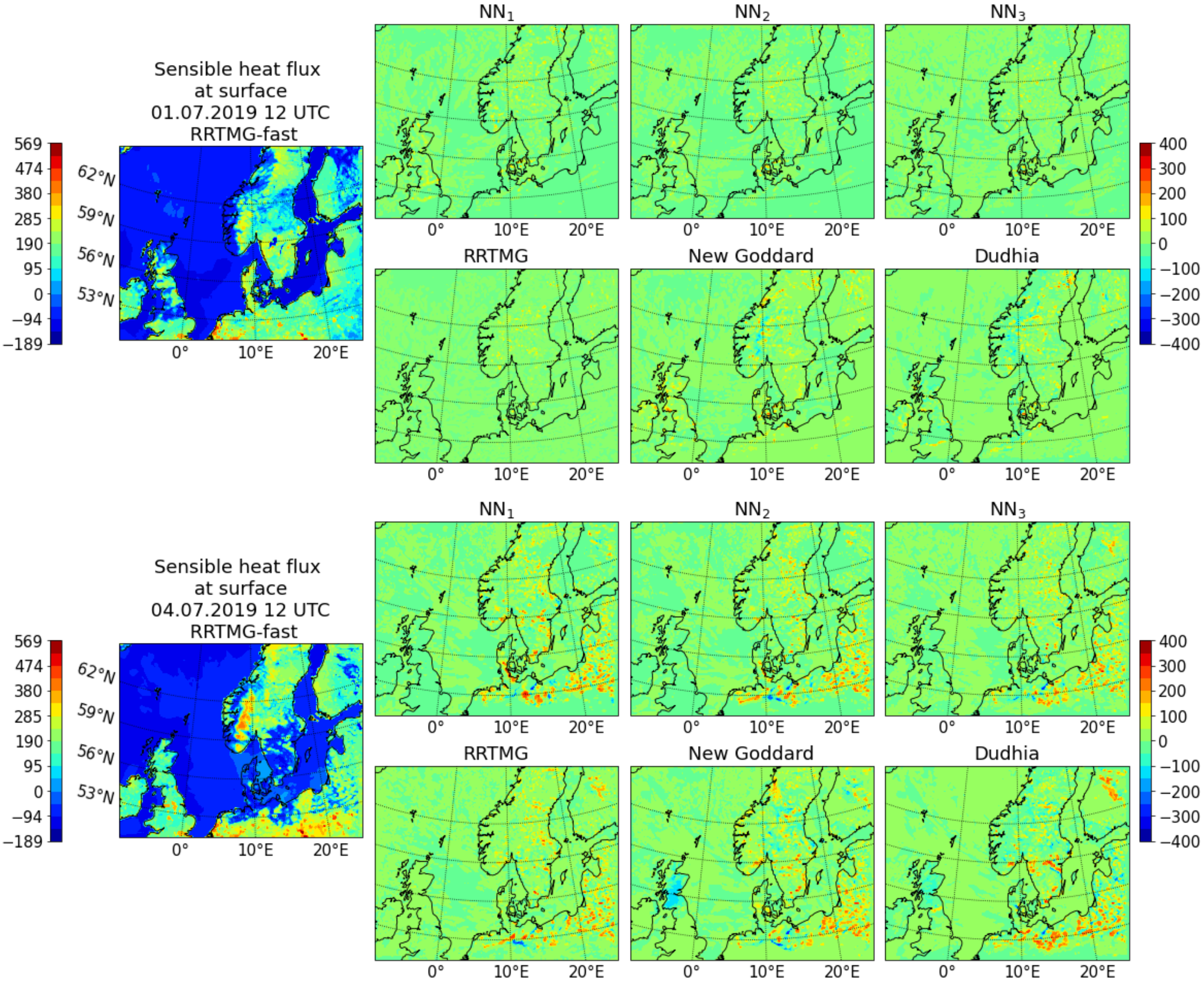


Figure 5.3: Plots showing the predictions made for the surface sensible heat flux (in W/m^2) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name are depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

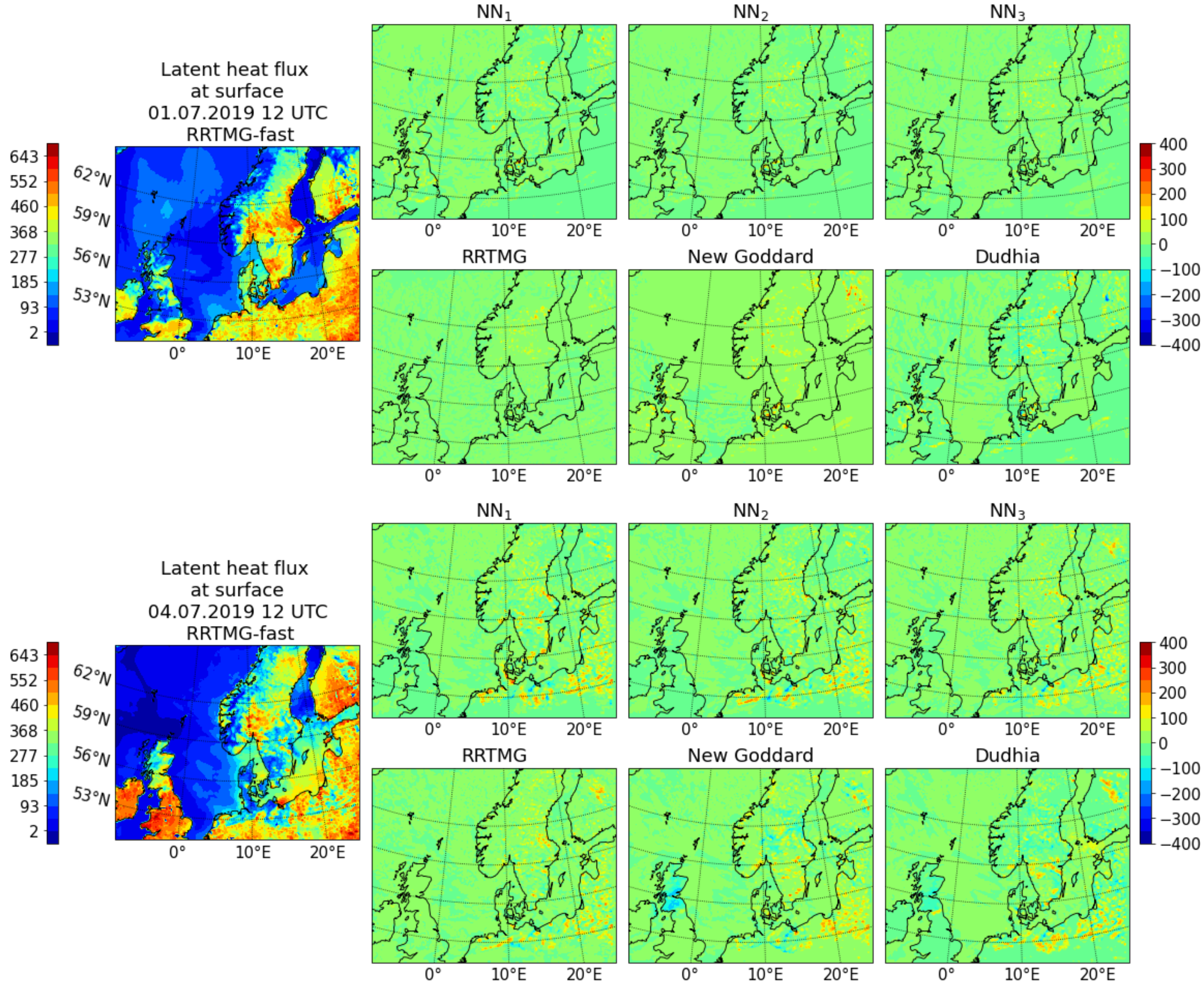


Figure 5.4: Plots showing the predictions made for the surface latent heat flux (in W/m^2) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name are depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

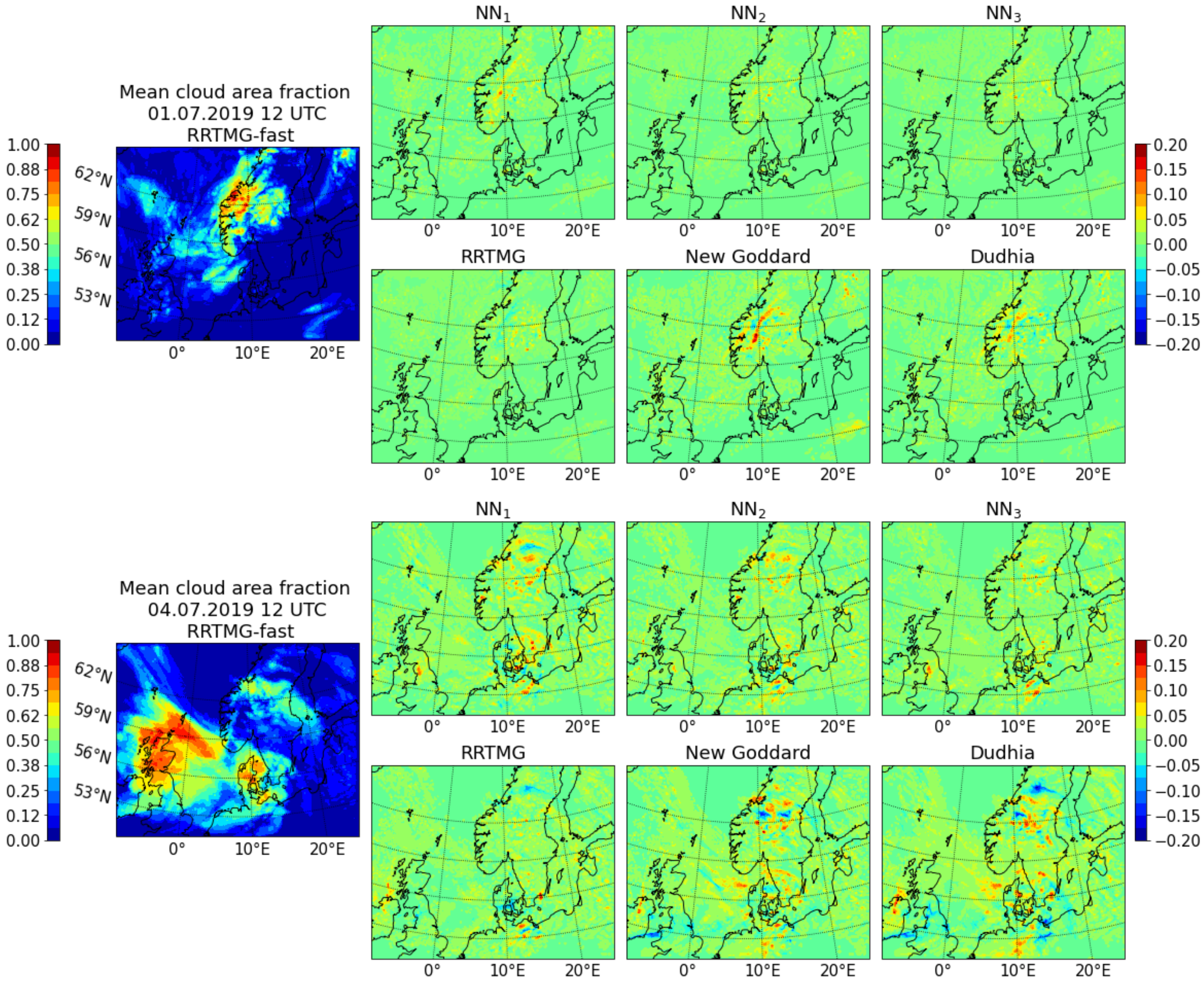


Figure 5.5: Plots showing the predictions made for the cloud area fraction (mean value for all vertical layers, as fraction e.g. 0.2 = 20%) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name are depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

Note how the largest anomalies occur where the fluxes have the least uniformly distribution. A good example of this is the shortwave flux at 12 UTC of 3.7.2019, i.e. 60 hours after the initial simulation time. The largest anomalies occur in the south-east where the fluxes are not the lowest, but not very smoothly distributed, while the blue areas with very small values do not result in such large anomalies. This is the case for all six schemes.

Clouds are generally predicted very similar in the different schemes, especially in the RRTMG and NN schemes, which use the same approximations as the RRTMG-fast. However, if a cloud's exact position in another simulation is predicted to develop in a slightly shifted location, the anomalies showing the difference between individual grid points will yield large anomalies, as it is seen here.

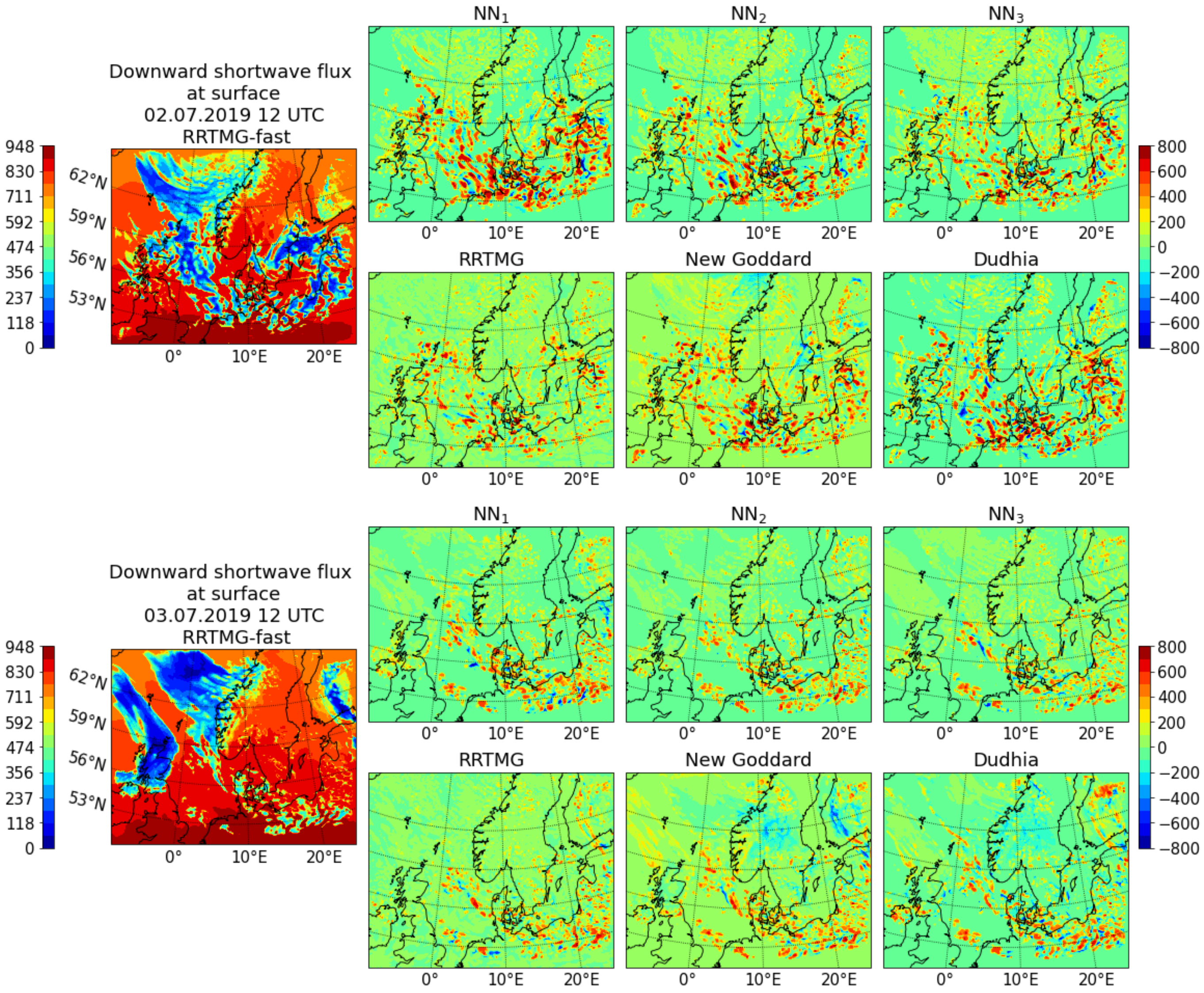


Figure 5.6: Plots showing the predictions made for the downward shortwave radiation at the surface (in W/m^2) after 36 hours (upper figure) and 60 hours (lower graphs). The timestamp and variable name are depicted above the figures on the left, while the anomalies of the other schemes are indicated by the titles on the right. Similar to figure 5.1.

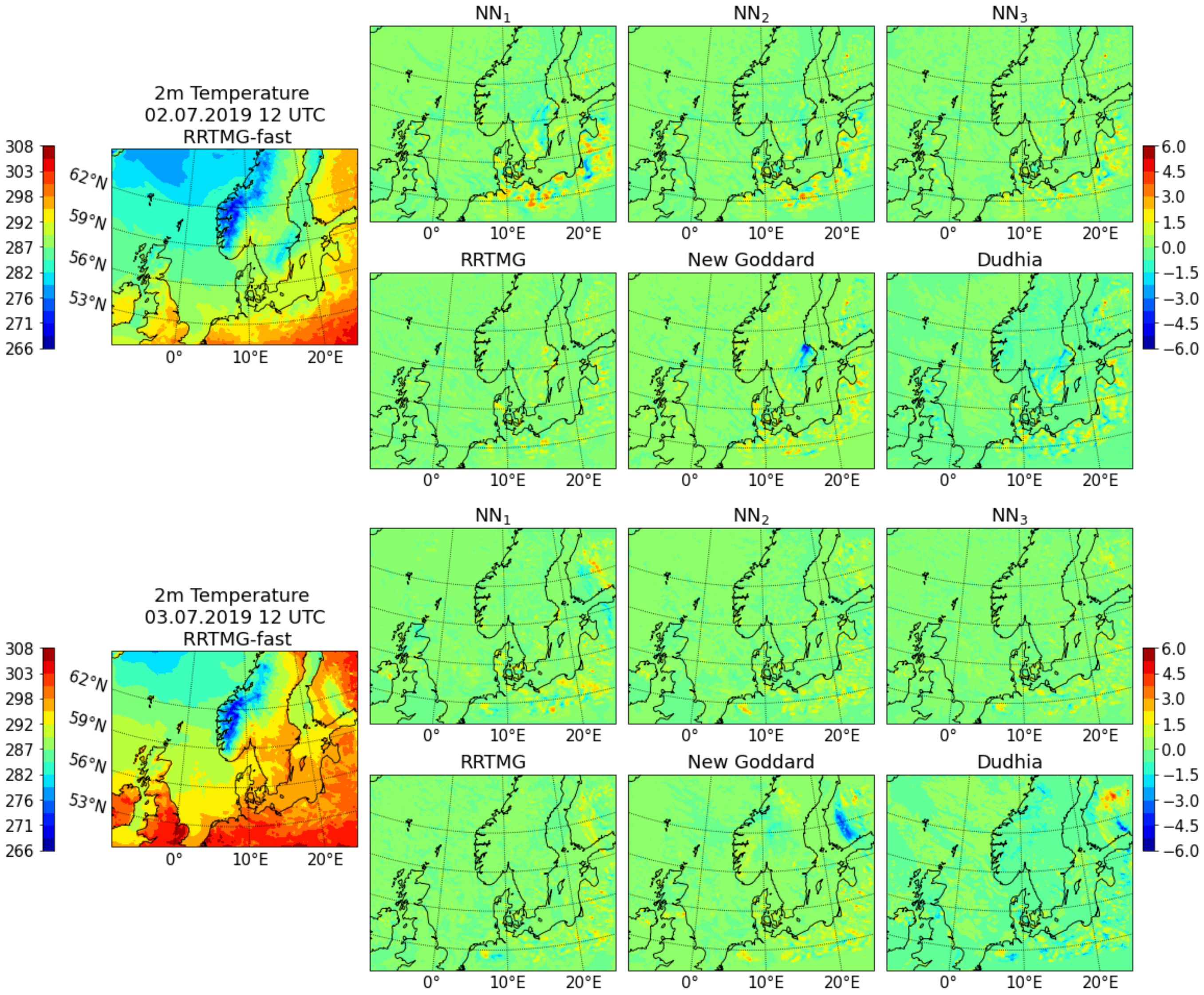


Figure 5.7: Plots showing the predictions made for the 2m temperature (in K) after 36 hours (upper figure) and 60 hours (lower graphs). The timestamp and variable name are depicted above the figures on the left, while the anomalies of the other schemes are indicated by the titles on the right. Similar to figure 5.2, but for the same times as figure 5.6.

Therefore the RMSE and Correlation coefficient r are calculated for the whole domain, for each variable, as described in section 5.1. The results are shown as functions of time since simulation start in figure 5.8.

The graphs in figure 5.8 show the some of the same features as seen in the geographical plots. The correlation is lowest and the RMSE is largest for the Dudhia and New Goddard scheme, which is in accordance with what has been seen before, i.e. their predictions differ most strongly from the RRTMG-fast predictions, while the opposite can be said about the RRTMG and NN 3 schemes.

The dropping correlation and increasing RMSE of the 2m temperature indicates that the anomalies increase with simulation time. This can also be seen for the other variables, but is harder to identify due to the strong diurnal cycle.

The diurnal cycle can also be seen from the plots in figure 5.9, which show the shortwave radiation flux at the surface for the times 6, 12 and 18 UTC on the 4.7.2019, i.e. after 78, 84 and 90 hours from the initial simulation time.

From both figure 5.8 and 5.9 it can be seen that the differences between the prediction of the six schemes and RRTMG-fast first increases and later decreases during the day, when the sunlight is starting to reach and later leave the domain, respectively.

The same analysis of calculating the RMSE and the correlation coefficent r has been carried out for the other three seasonal study cases, which are shown in the figures 5.10 - 5.12.

For all four seasons the same tendencies can be seen, though there are some smaller differences, as e.g. the diurnal cycle is not as strongly present in 5.10 for the winter simulation, which is probably due to the relatively short period of time the domain is exposed to sunlight.

It can also be seen that the magnitude of the errors is largest in the summer case, which is related to the values of the variables, which are generally larger in summer, and the daytime is longer.

It can be noted that the RRTMG and NN 3 scheme generally compete for being the scheme with the smallest difference compared to the RRTMG-fast.

The anomalies of the NN schemes, is rather small and much smaller than those of the New Goddard and Dudhia schemes. Additionally the anomalies increase in a smiliar manner as those of the RRTMG scheme, which uses the same physical approximations as the RRTMG-fast scheme. Thus it can be concluded that the neural networks have been able to learn to emulate the computations of the `reftra_sw` to a high degree.

It should however be noted that the predictions by the NN schemes do have a strange feature of computing occasionally small, negative downward fluxes, which is a non-physical result, as the downward flux is by definition positive. This only happens at a few, apparently random points, usually in the morning in the areas where the sunlight begins to reach the grid points.

For the NN 1 and NN 3 scheme the most negative downward fluxes at the surface were of the magnitude $\sim -0.1\ W/m^2$, for NN 2, however, the minimum was $\sim -50\ W/m^2$.

Since the neural networks do not predict the flux directly, but the reflectivities and transmissivities used for the flux computation, it was not possible to find out what triggered these negative fluxes, since the known constraints of the reflectivities and transmissivities were met, i.e. each of the parameters can only take on a value between 0 and 1 and for their sums it applies: $R_{dir} + T_{dir} \leq 1$ and $R_{dif} + T_{dif} \leq 1$.

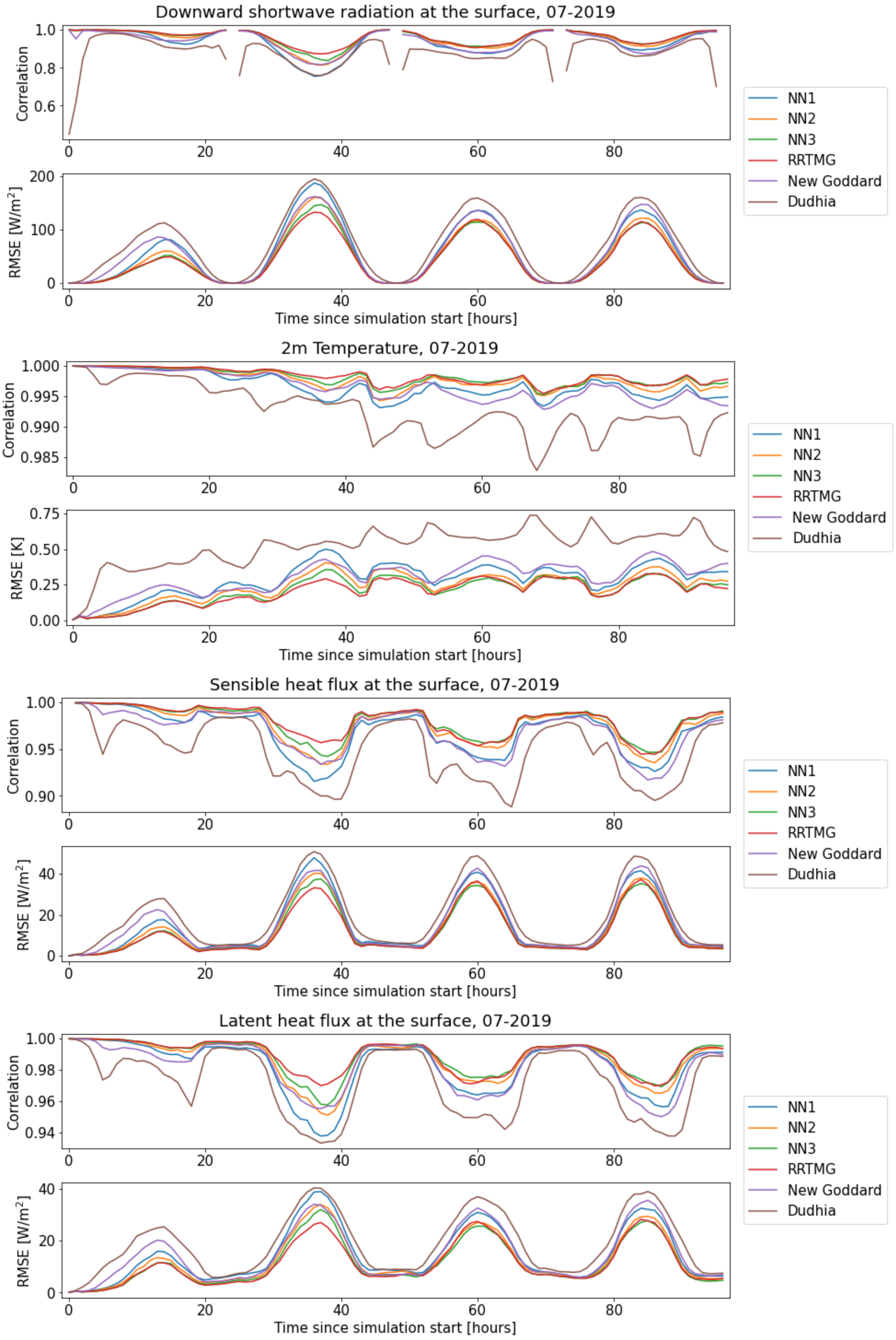


Figure 5.8: Plots showing the correlation r and the root mean square error (RMSE) as function of the number of simulated hours since the initial time. The depicted variable is indicated by the title, as well as for which 4-day (96h) simulation the figure was constructed, here the data shown is based on the simulation with the initial time 1.7.2019 00 UTC. The different radiation parameterization schemes are listed in the legends on the right side.

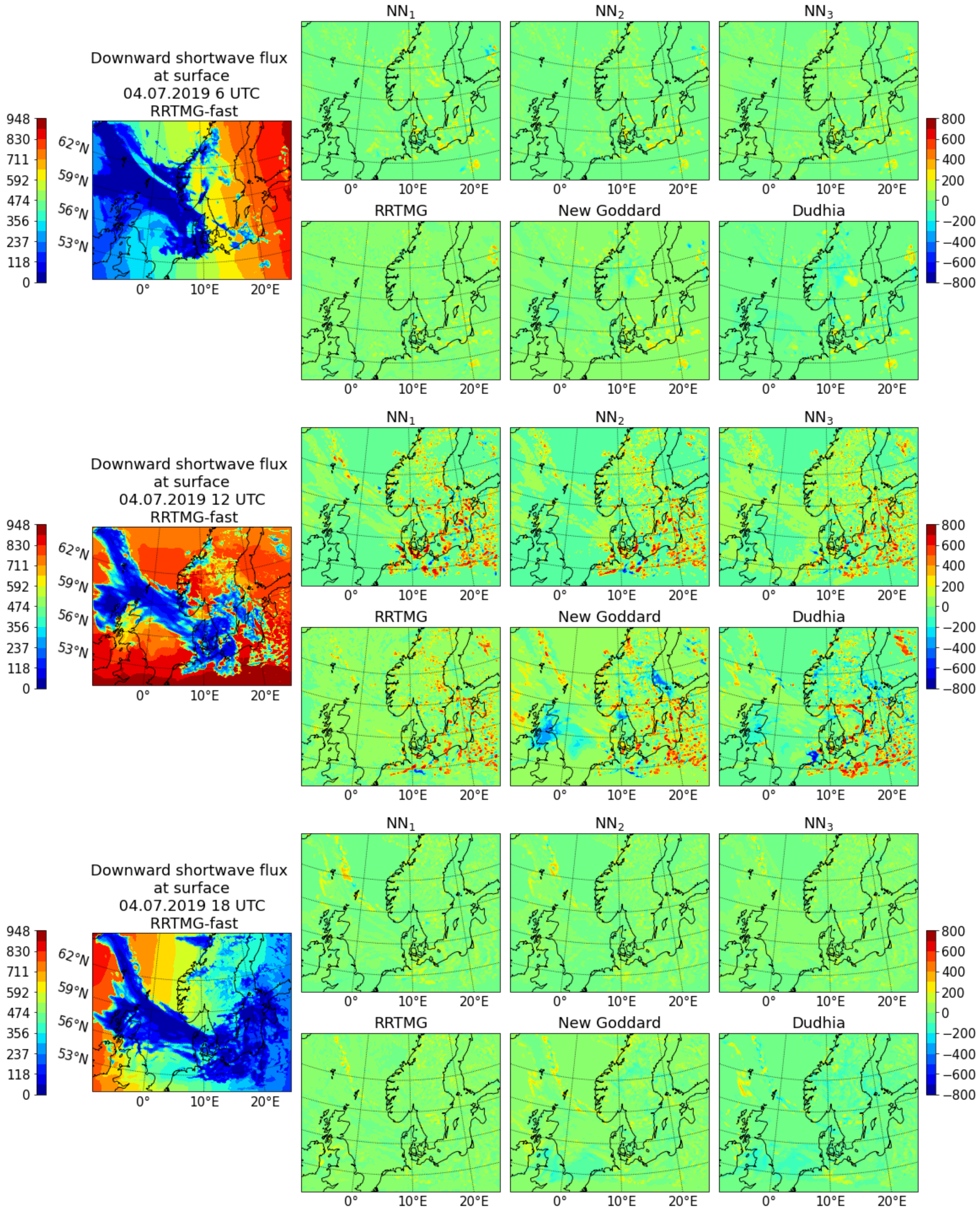
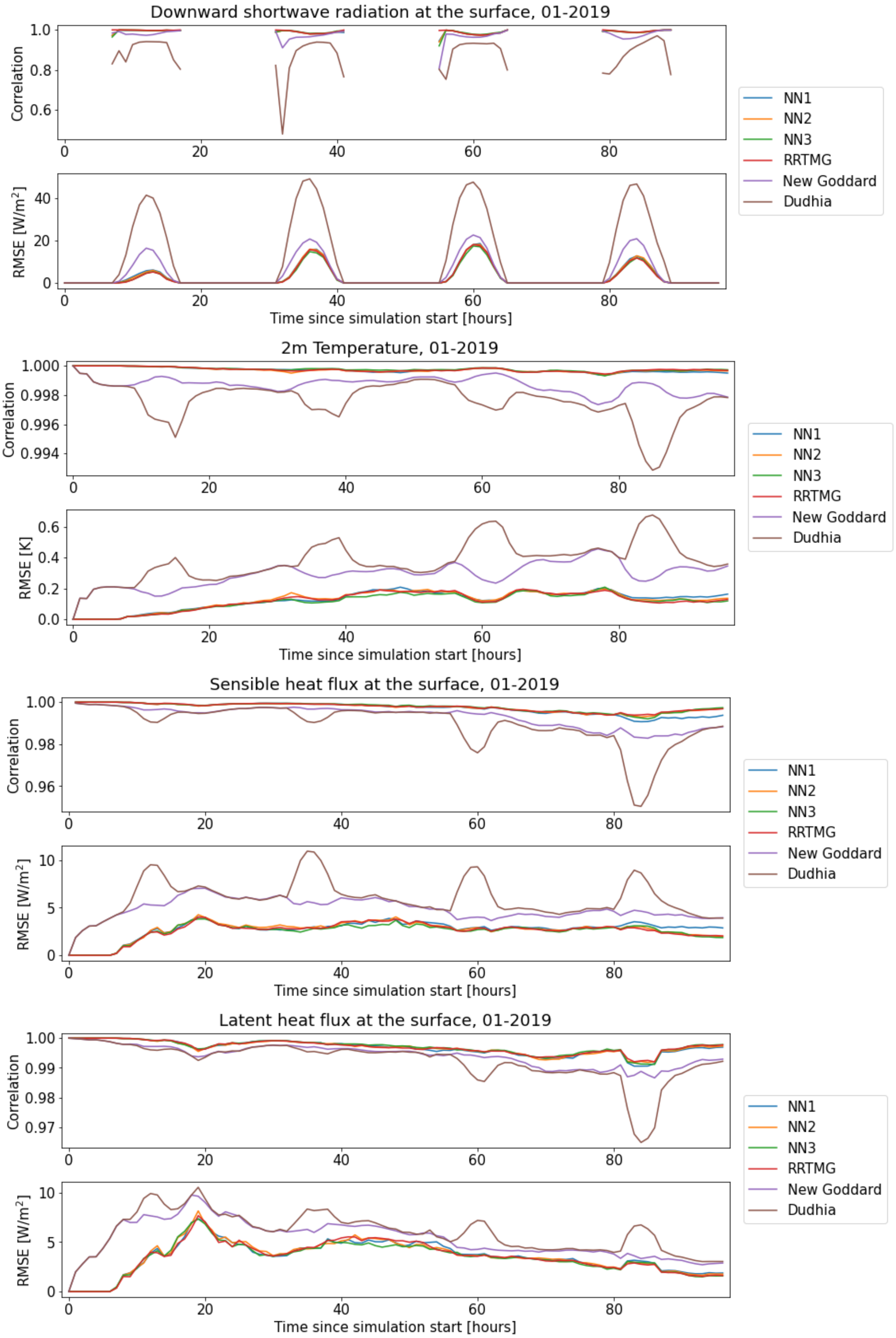


Figure 5.9: Plots showing the predictions made for the downward shortwave radiation at the surface (in W/m^2) after 78 hours (upper figure), 84 hours (middle) and 90 hours (lower graphs). The timestamp and variable name are depicted above the figures on the left, while the anomalies of the other schemes are indicated by the titles on the right. Similar to figure 5.1.



Downward shortwave radiation at the surface, 01-2019

2m Temperature, 01-2019

Sensible heat flux at the surface, 01-2019

Latent heat flux at the surface, 01-2019

Figure 5.10: Plots showing the correlation r and the root mean square error (RMSE) as function of the number of simulated hours since the initial time. The depicted variable is indicated by the title, as well as for which 4-day (96h) simulation the figure was constructed, here the data shown is based on the simulation with the initial time 1.1.2019 00 UTC. The different radiation parameterization schemes are listed in the legends on the right side.

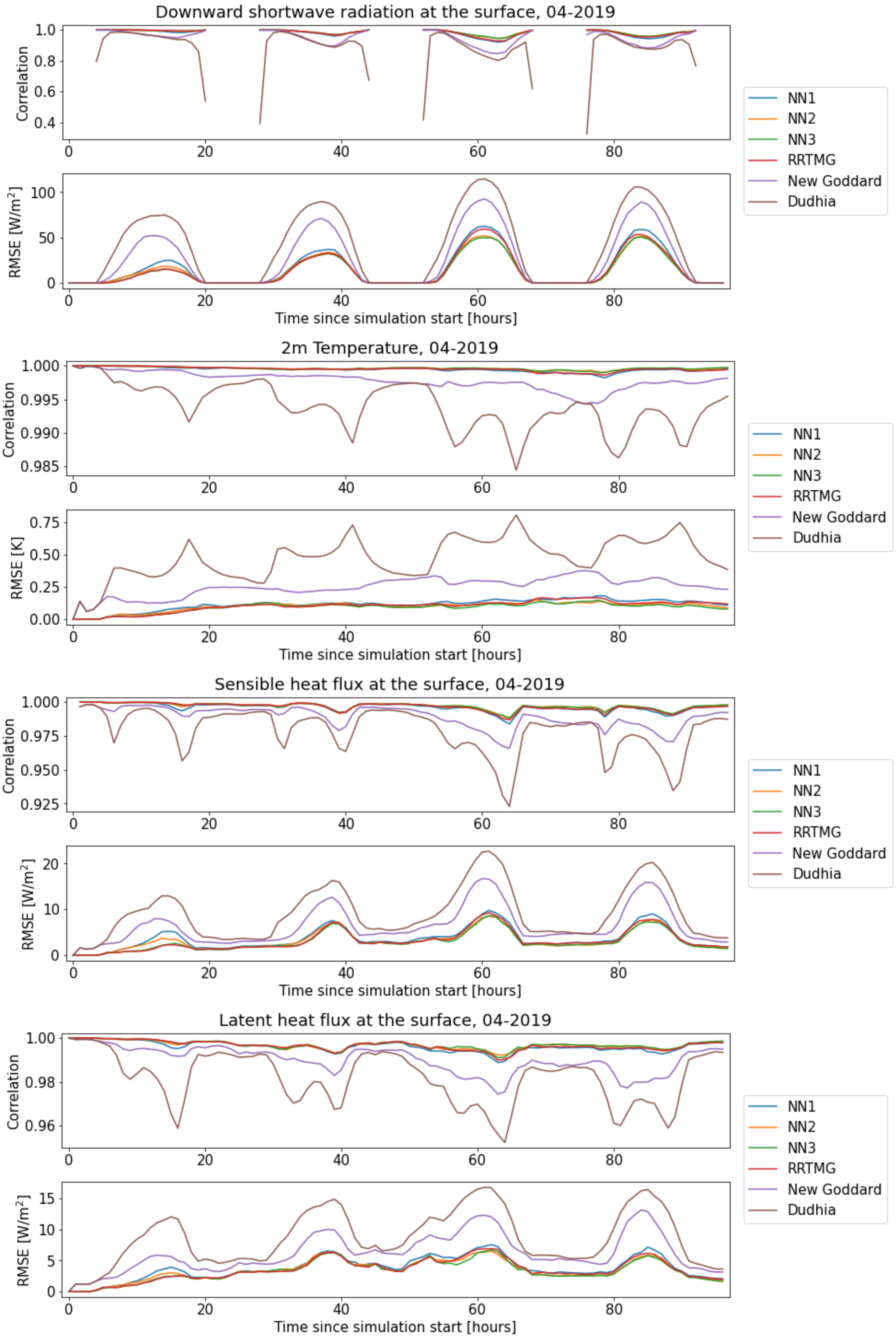


Figure 5.11: Plots showing the correlation r and the root mean square error (RMSE) as function of the number of simulated hours since the initial time. The depicted variable is indicated by the title, as well as for which 4-day (96h) simulation the figure was constructed, here the data shown is based on the simulation with the initial time 1.4.2019 00 UTC. The different radiation parameterization schemes are listed in the legends on the right side.

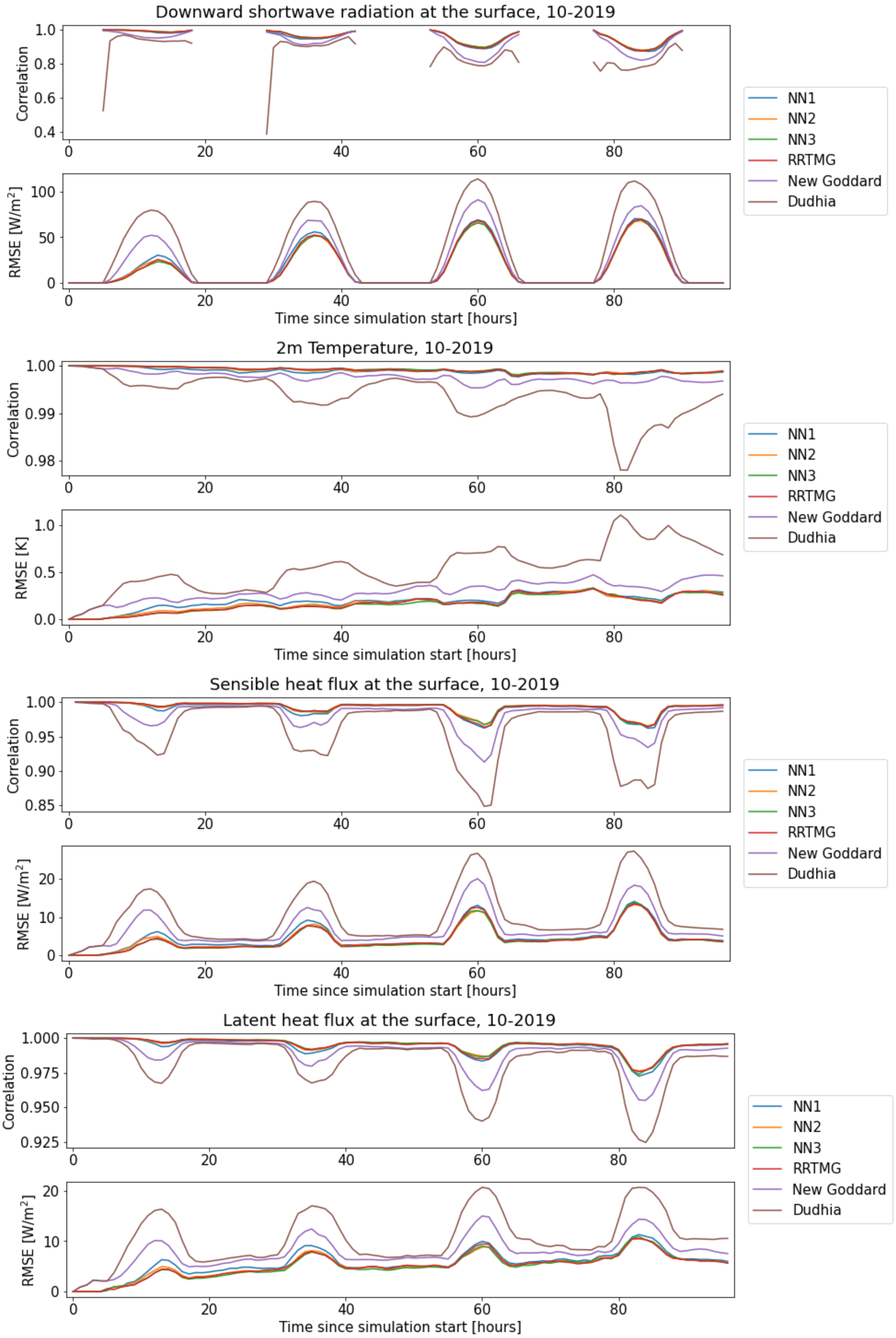


Figure 5.12: Plots showing the correlation r and the root mean square error (RMSE) as function of the number of simulated hours since the initial time. The depicted variable is indicated by the title, as well as for which 4-day (96h) simulation the figure was constructed, here the data shown is based on the simulation with the initial time 1.10.2019 00 UTC. The different radiation parameterization schemes are listed in the legends on the right side.

5.3 Computational efficiency

Since the only difference between all simulations made by the WRF model is the choice of the used shortwave parameterization, the total computation time needed to simulate the same time period for the same domain gives an indication of the computation time of the different shortwave parameterization schemes.
The percentage of the total model run used on the shortwave parameterization depends heavily on the specific time period and domain, since the parameterization is only executed if there is sunlight present, as mentioned earlier in section 3.3.

For a more direct comparison between the computational times of the shortwave schemes and the modified schemes, where the neural networks have been included, it can be chosen to let WRF write out the time spend on calculations in the `radiation driver` at each radiative time step.
A 12-hour simulation was carried out with each of the shortwave parameterizaton schemes used for the test cases in the previous section, where the computation times have been reported by the WRF model.

While the individual computation times of some calculations inside a parameterization can vary by some magnitudes, as e.g. for cloudy and clear sky cases where different neural networks are used for, the total time spend on the parameterization turns out to be very similar for all radiative time steps.

In table 5.1 the typical time spend on the radiation driver is shown for the different schemes, for the cases when the shortwave radiation is executed. Note that the computation time of the longwave radiation has been subtracted from those values, as the same longwave parameterization scheme was used in all simulations and showed very similar execution times in all models, as well as for all radiative time steps.

Shortwave parameterization scheme	**Time spent on radiation driver per radiative time step, when SW is executed**
RRTMG-fast with NN model 3	111.60 s
RRTMG-fast with NN model 2	43.15 s
RRTMG-fast with NN model 1	42.02 s
RRTMG-fast with NN model 1* (ReLu)	22.88 s
RRTMG-fast	6.45 s
RRTMG	4.95 s
New Goddard	1.66 s
Dudhia	0.18 s

Table 5.1: Table showing an example of the typical time spent on the radiation driver for the different shortwave parameterization schemes for one radiative time step, when the shortwave parameterization is executed. Note that the fraction spend on the longwave parameterization by the radiation driver has been subtracted from these times.

From the values in table 5.1 it becomes apparent that all parameterization schemes including neural networks are much slower than the original RRTMG-fast scheme. The best performing, but most complex neural network scheme is more than 17 times slower than the scheme it tries to emulate, while the other two neural network models are around $6 \sim 7$ times slower than the original RRTMG-fast.

There are a few possible reasons for the slow performance of the neural networks. One unclear factor is the implementation using the FKB library, which has not been evaluated on its computational efficiency. For the implementation into WRF, a different approach might be more efficient, e.g. hard coding the weights of the neural networks into a WRF subroutine. But it seems unlikely to be the main cause for the slow performance.

There is, however, another factor that was easy to be tested through the usage of a fourth parameterization with another neural network, which in the table is denoted as model 1^*.

The neural networks in this model are identical to the ones in model 1, however, the neural network trained on clear sky conditions uses the ReLu function instead of the tanh function as activation in its hidden layers.

It is well known that the tanh function is computational more expensive than the ReLu function and the quick test showed a large improvement in computational speed, where the computational time of model 1^* is only $\sim 55\%$ of the time needed by model 1. Note though, that the performance of this model 1^* has yet to be evaluated.

The simpler schemes, Dudhia and New Goddard, are faster than the RRTMG schemes, as one would expect. It is surprising that the optimized version of the RRTMG scheme, RRTMG-fast is slightly slower than the original RRTMG parameterization.

It should be noted though, that all simulations in this study have been performed on CPUs, so it is to expect that both the original RRTMG-fast and its modified variants with neural networks would perform faster when used with GPUs.

6 Discussion

In this section a very brief summary of the main points from the first few sections will be given, before discussing the central findings of the development of the neural networks described in section 4 and the results from the simulations carried out by the WRF model with the parameterization schemes containing neural network, presented in section 5.

The goal of this thesis was to test the usage of neural networks in the radiation parameterization of the WRF model.
Since early tests suggested that the shortwave radiation parameterization was more computationally heavy than the longwave parameterization, it was chosen to focus on the shortwave parameterization schemes.

Of all the available shortwave parameterizations schemes in the WRF model, the RRTMG and its for GPUs optimized version RRTMG-fast are the most complex parameterizations, which also are the ones most similar to the schemes used by modern NWP models.
Despite not having GPUs available in this study, it was chosen to work with the newer RRTMG-fast scheme. If GPUs would be used, it is expected that the RRTMG-fast would perform much faster.

There are different approaches on how to utilize neural networks for parameterizations schemes, in this case it was chosen to substitute the part of the scheme that was computational most expensive: the `reftra_sw` subroutine, where the reflectivities and transmissivities are computed.

After this choice was made, a data set for training, validating and testing the neural networks was created with the original RRTMG-fast parameterization, for which data points were randomly sampled from 12 24-hour simulations, distributed over the year 2018.
With this data set several tests for optimizing the neural networks were carried out.

When the influence of different input variables for the neural networks were investigated, it became apparent that it was beneficial to divide the data set into two categorize and train two separate neural networks for the two cases: cloudy and clear sky conditions.

However, it was also then discovered that the initial training and validation data set for the clear sky conditions was not divided into two completely independent, well distributed data sets, as the initial validation set appeared to contain less challenging data than the training data, leading to a much lower validation error.

Since this would make detecting overfitting and evaluating the training's process very difficult, it was chosen to interchange the training and validation data for clear sky samples.
While this was not an optimal premise, it appears to at least have prevented uncontrolled overfitting, as the later implemented models also performed well in the study cases in the WRF model, which posed as an additional independent test.

During the many procedures to find the optimal hyper parameters it had also

been discovered during the early phases of trial and error, that the scaling of the optical thicknesses τ had a big impact on the predictions of the neural networks.
All input and output variables of the neural network, which are taken from `reftra_sw`, only take on values between 0 and 1, apart from the optical thicknesses values, that have a wide range of values.

Since such large differences in orders of magnitudes of the different input variables have a bad influence on the learning rate and updating of the model weights of the neural network, a normalization was needed.
A logarithmic scaling coupled with the division through a maximum value gave the best results.
Still, all of the implemented neural network models tested in section 5, continued to predict small negative fluxes, which are non-physical.
Despite using constraints in the modified `reftra_sw` routine with the neural networks, where the predicted transmissivities and reflectivites only can range between 0 an 1 and their diffuse and direct sum only can become as large as 1, the negative fluxes still occurred.

The minimum value is very small for two of the three models ($\sim -0.1\ W/m^2$) while larger for the last one ($\sim -50\ W/m^2$), which still is lower than the minimum values of models with different scaling, which could predict values much lower and larger.

The three implemented radiation parameterizations containing neural networks performed well in the four simulations carried out with the WRF model. The NN 3 scheme performed best of all three, with an root mean square error (RMSE) and correlation coefficient r, that was consistently lower than the ones of the other two neural network schemes. The NN 1 scheme performed worst of the three, which is in accordance to the results of the previous evaluation on the test data set of the three neural network models, which implies that the test data set seems to have been well sampled and independent.

The plots showing the RMSE and correlation r as a function of time showed that the predictions by the neural network NN schemes were more similar to the RRTMG-fast than the predictions made by two different radiation parameterizations, the New Goddard and Dudhia scheme.

The only other parameterization that showed similar RMSE and r values was the RRTMG scheme, which is not surprising, since the physical approximations in this scheme are the same as in its optimized version, RRTMG-fast.
It could also be seen that the rate at which the NN schemes and the RRTMG scheme start to diverge further from the RRTMG-fast with increasing simulation time is of the same magnitude.

The largest anomalies were found for the the shortwave fluxes, which also seemed to be less uniformly, than the anomalies of the 2m temperature, the sensible heat flux at the surface and the latent heat flux at the surface.

The large anomalies of the shortwave fluxes are probably due to slight differences in the simulated cloud cover. Small scattered clouds that are located at slightly shifted positions in the different simulations made with the parameterization schemes result by a grid point vs grid point comparison to large fluctuations.

When the computational times were investigated in section 5.3, it was seen that all neural network containing schemes performed much slower than the original RRTMG-fast scheme. The best scheme, NN 3, was $\sim$ 17times slower, while the other two schemes were faster, but still $\sim 6-7$ times slower than the RRTMG-fast.

There are many factors that have yet to be evaluated that can have an influence on the computational efficiency.

One would be the method used to implement the neural networks into the WRF fortran code, for which in this study the Fortran-Keras Bridge (FKB) was used. This fortran library is very useful, since it makes loading different neural networks, trained with keras, into fortran very easy, which was good to test many different models, however, it was not evaluated how computational efficient it is.

However, one major cause for the long computational times was found to be the activation function of the clear-sky cases. The tanh function is much more computational expensive than the ReLU function used for the cloudy cases and a quick test, were a neural network scheme similar to NN 1, but with ReLu instead of tanh as activation, performed much faster only needing $\sim 55\%$ of the time compared to NN 1 with tanh.

The performance of this model has yet to be evaluated, however, since the ReLu function performed as the second best activation during the testing of activation functions for clear sky cases, it seems reasonable to expect, that a good neural network can be optimized with this function, for this problem.

7 Conclusion and outlook

In this study it has been shown that neural networks can be trained well enough to be used as part of the RRTMG-fast shortwave radiation parameterization scheme in the WRF model. The results illustrated how predictions made by parameterization schemes using neural networks gave similar results to the original RRTMG-fast scheme, which they were trained on, as well as to the original RRTMG scheme, which uses the same approximations.

Unfortunately, the parameterization schemes using neural networks turned out to be computationally slower than the original parameterization scheme.
So to build an efficient shortwave radiation parameterization scheme, additional optimization methods need to be considered.

The choice of activation function proved to be computationally expensive and was one of the main contributors to the slow performance. A simple test suggested that the computation time can be strongly reduced by using a cheaper function, e.g. the ReLu instead of the tanh as activation function, though the performance of such a neural network still needs to be evaluated.

All simulations of the WRF model were carried out on CPUs in this study. The RRTMG-fast parameterization scheme is, however, designed for the use with GPUs. Therefore it would be interesting to see how much faster the scheme becomes when utilizing GPUs and how much of an improvement this would lead to for the parameterization schemes containing neural networks.

During the optimization process of the neural networks it was discovered that the random data sampling used for creating the training, validation and test data set had not resulted in completely independent, well divided subsets. Therefore one way of improving the neural network should include a reconsideration of the method to sample data, e.g. to expand the data set by manually selecting extreme values.

Another approach could be to investigate the usage of neural networks for other parts of the radiation parameterization, e.g. the optical gas or aerosol computations, or to try replacing a larger part of the parameterization instead of one individual subroutine.

If available, it could also be interesting to train neural networks on more realistic data, such as outputs from line-by-line models, to test whether this could lead to more accurate predictions of the radiative fluxes.

8 References

References

[Abadi et al., 2015] Abadi, M. et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from `https://www.tensorflow.org/`.

[Aggarwal, 2018] Aggarwal, C. C. (2018). Neural Networks and Deep Learning. Springer, Cham. `https://doi.org/10.1007/978-3-319-94463-0`.

[Chandrasekhar, 1950] Chandrasekhar, S. (1950). Radiative transfer. Oxford Univ. Press.

[Chevallier et al., 1998] Chevallier, F., Chéruy, F., Scott, N. A., and Chédin, A. (1998). A neural network approach for a fast and accurate computation of a longwave radiative budget. Journal of Applied Meteorology, 37(11):1385–1397. `https://doi.org/10.1175/1520-0450(1998)037<1385:ANNAFA>2.0.CO;2`.

[Chollet et al., 2015] Chollet, F. et al. (2015). Keras. `https://keras.io`.

[Chou and Suarez, 1999] Chou, M.-D. and Suarez, M. J. (1999). A solar radiation parameterization for atmospheric studies. NASA Technical Report, NASA/GSFC, 15. `https://ntrs.nasa.gov/search.jsp?R=19990060930` [Accessed: 15-07-2020].

[CLR-github,] CLR-github. Cyclical Learning Rate (CLR) github repository. `https://github.com/bckenstler/CLR` [Accessed: 29-06-2020].

[Coblenz, 2015] Coblenz, J. S. (2015). Using Machine Learning Techniques to Improve Precipitation Forecasting.

[Dudhia, 1989] Dudhia, J. (1989). Numerical study of convection observed during the winter monsoon experiment using a mesoscale two-dimensional model. Journal of The Atmospheric Sciences - J ATMOS SCI, 46:3077–3107. `https://doi.org/10.1175/1520-0469(1989)046<3077:NSOCOD>2.0.CO;2`.

[Dueben, 2020] Dueben, P. (2020). AI and machine learning at ECMWF. ECMWF newsletter nr.163. `https://www.ecmwf.int/en/newsletter/163/news/ai-and-machine-learning-ecmwf`.

[Dueben and Bauer, 2018] Dueben, P. D. and Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. Geoscientific Model Development. doi: 10.5194/gmd-11-3999-2018.

[ECMWF, 2017] ECMWF (2017). Press kit: Bologna to host ECMWF's new data centre. `https://www.ecmwf.int/en/about/media-centre/press-kit-bologna-host-ecmwfs-new-data-centre` [Accessed: 29-04-2020].

[Feng et al., 2019] Feng, J., He, X., Teng, Q., Ren, C., Chen, H., and Li, Y. (2019). Reconstruction of porous media from extremely limited information using conditional generative adversarial networks. Physical Review E, 100. http://doi.org/10.1103/PhysRevE.100.033308.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press. http://www.deeplearningbook.org.

[Hogan et al., 2018] Hogan, R., Bozzo, A., Fielding, M., Barker, H., Vitart, F., Schaefer, S., and Polichtchouk, I. (2018). Challenges for radiation in NWP models. https://www.ecmwf.int/node/18220.

[Hogan and Bozzo, 2018] Hogan, R. J. and Bozzo, A. (2018). 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.

[Hogan and Illingworth, 2000] Hogan, R. J. and Illingworth, A. J. (2000). Deriving cloud overlap statistics from radar. Quarterly Journal of the Royal Meteorological Society, 126(569):2903–2909. https://doi.org/10.1002/qj.49712656914.

[Iacono et al., 2008] Iacono, M. J., Delamere, J. S., Mlawer, E. J., Shephard, M. W., Clough, S. A., and Collins, W. D. (2008). Radiative forcing by long-lived greenhouse gases: Calculations with the aer radiative transfer models. Journal of Geophysical Research: Atmospheres, 113(D13). https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JD009944.

[Ingeniøren, 2016] Ingeniøren (2016). Ingeniøren/version2, Efter 31 år er supercomputeren blevet tavs i DMI's maskinstue. https://www.version2.dk/artikel/efter-31-aar-er-supercomputeren-blevet-tavs-i-dmis-maskinstue-651962 [Accessed: 29-04-2020].

[Inness and Dorling., 2013] Inness, P. and Dorling., S. (2013). Operational Weather Forecasting. Wiley-Blackwell, ISBN: 978-0-470-71158-3 .

[Joseph et al., 1976] Joseph, J. H., Wiscombe, W. J., and Weinman, J. A. (1976). The Delta-Eddington Approximation for Radiative Flux Transfer. Journal of the Atmospheric Sciences, 33(12):2452–2459. https://doi.org/10.1175/1520-0469(1976)033<2452:TDEAFR>2.0.CO;2.

[Kawai et al., 2014] Kawai, H., Yabu, S., and Hagihara, Y. (2014). The evaluation of the vertical structures of marine boundary layer clouds over mid-latitudes. volume 44, pages 0611–0612.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015, http://arxiv.org/abs/1412.6980.

[Krasnopolsky et al., 2005] Krasnopolsky, V. M., Fox-Rabinovitz, M. S., and Chalikov, D. V. (2005). New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. Monthly Weather Review, 133(5):1370–1383. https://doi.org/10.1175/MWR2923.1.

[Krasnopolsky and Lin, 2012] Krasnopolsky, V. M. and Lin, Y. (2012). A neural network nonlinear multimodel ensemble to improve precipitation forecasts over continental us. Hindawi Publishing Corporation - Advances in Meteorology. https://doi.org/10.1155/2012/649450.

[Liou, 1974] Liou, K.-n. (1974). Analytic Two-Stream and Four-Stream Solutions for Radiative Transfer. Journal of the Atmospheric Sciences, 31(5):1473–1475. https://doi.org/10.1175/1520-0469(1974)031<1473:ATSAFS>2.0.CO;2.

[Liou, 2002] Liou, K. N. (2002). An Introduction to Atmospheric Radiation. Second Edition, Academic Press, ISBN:0-12-451451-0 .

[Meador and Weaver, 1980] Meador, W. E. and Weaver, W. R. (1980). Two-Stream Approximations to Radiative Transfer in Planetary Atmospheres: A Unified Description of Existing Methods and a New Improvement. Journal of the Atmospheric Sciences, 37(3):630–643. https://doi.org/10.1175/1520-0469(1980)037<0630:TSATRT>2.0.CO;2.

[MetOffice-website,] MetOffice-website. Our supercomputers. https://www.metoffice.gov.uk/about-us/who/sustainability/environment/supercomputers [Accessed: 29-04-2020].

[Morcrette and Fouquart, 1986] Morcrette, J.-J. and Fouquart, Y. (1986). The Overlapping of Cloud Layers in Shortwave Radiation Parameterizations. Journal of the Atmospheric Sciences, 43(4):321–328. https://doi.org/10.1175/1520-0469(1986)043<0321:TOOCLI>2.0.CO;2.

[NCEP, 2015] NCEP (2015). National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce, NCEP GDAS/FNL 0.25 Degree Global Tropospheric Analyses and Forecast Grids. Research Data Archive at the National Center for Atmospheric Research, Computational and Information Systems Laboratory. https://doi.org/10.5065/D65Q4T4Z [Accessed: 27-09-2019].

[Ott et al., 2020] Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., and Baldi, P. (2020). A fortran-keras deep learning bridge for scientific computing. https://arxiv.org/abs/2004.10652.

[Pincus et al., 2003] Pincus, R., Barker, H. W., and Morcrette, J.-J. (2003). A fast, flexible, approximate technique for computing radiative transfer in inhomogeneous cloud fields. Journal of Geophysical Research: Atmospheres, 108(D13). https://doi.org/10.1029/2002JD003322.

[Randall, 2015] Randall, D. A. (2015). An Introduction to the Global Circulation of the Atmosphere. Princeton University Press, ISBN:978-0-69-1148960 .

[Räisänen, 2002] Räisänen, P. (2002). Two-stream approximations revisited: A new improvement and tests with gcm data. Quarterly Journal of the Royal Meteorological Society, 128(585):2397–2416. https://rmets.onlinelibrary.wiley.com/doi/abs/10.1256/qj.01.161.

[Skamarock et al., 2019] Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Liu, Z., Berner, J., Wang, W., Powers, J. G., Duda, M. G., Barker, D. M., and Huang, X.-Y. (2019). A Description of the Advanced Research WRF Version 4 NCAR Tech. Note 556. . https://doi.org/10.5065/1dfh-6p97.

[Smith, 2015] Smith, L. N. (2015). Cyclical learning rates for training neural networks (revised version 2017). IEEE Winter Conference on Applications of Computer Vision (WACV), pages 464–472. http://arxiv.org/abs/1506.01186.

[Stephens, 1978] Stephens, G. L. (1978). Radiation Profiles in Extended Water Clouds. II: Parameterization Schemes. Journal of the Atmospheric Sciences, 35(11):2123–2132. https://doi.org/10.1175/1520-0469(1978)035<2123:RPIEWC>2.0.CO;2.

[Stergiou et al., 2017] Stergiou, I., Tagaris, E., and Sotiropoulou, R. (2017). Sensitivity assessment of wrf parameterizations over europe. Proceedings, 1:119. https://doi.org/10.3390/ecas2017-04138.

[Thomas and Stamnes, 1999] Thomas, G. E. and Stamnes, K. (1999). Radiative Transfer in the Atmosphere and Ocean. Cambridge University Press, ISBN: 0-521-40124-0 .

[Van Rossum and Drake, 2009] Van Rossum, G. and Drake, F. L. (2009). Python 3 Reference Manual. CreateSpace, Scotts Valley, CA. Available at http://www.python.org.

[Wallace and Hobbs., 2006] Wallace, J. M. and Hobbs., P. V. (2006). Atmospheric Science - An Introductory Survey. Second Edition, Academic Press, ISBN: 978-0-12-732951-2 .

[Weyn et al., 2019] Weyn, J. A., Durran, D. R., and Caruana, R. (2019). Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data. Journal of Advances in Modeling Earth Systems. https://doi.org/10.1029/2019MS001705.

[Wolf et al., 2020] Wolf, K., Ehrlich, A., Mech, M., Hogan, R. J., and Wendisch, M. (2020). Evaluation of ECMWF Radiation Scheme Using Aircraft Observations of Spectral Irradiance above Clouds. Journal of the Atmospheric Sciences, 77(8):2665–2685. https://doi.org/10.1175/JAS-D-19-0333.1.

[WRF-userguide,] WRF-userguide. Weather Research and Forecasting Model - ARW Version 4 Modeling System User's Guide. . https://www2.mmm.ucar.edu/wrf/users/docs/user_guide_v4/contents.html [Accessed: 15-07-2020].

[Yang et al., 2018] Yang, Q., Zhang, F., Zhang, H., Wang, Z., Li, J., Wu, K., Shi, Y., and Peng, Y. (2018). Assessment of two two-stream approximations in a climate model. Journal of Quantitative Spectroscopy and Radiative Transfer, 225. https://doi.org/10.1016/j.jqsrt.2018.12.016.

[Zdunkowski et al., 1980] Zdunkowski, W. G., Welch, R. M., and Korb, G. (1980). An investigation of the structure of typical two-stream methods for the calculation of solar fluxes and heating rates in clouds. Beiträge zur Physik der Atmosphäre, 53:147–166.

[Zhang et al., 2018] Zhang, F., Yan, J.-R., Li, J., Wu, K., Iwabuchi, H., and Shi, Y.-N. (2018). A New Radiative Transfer Method for Solar Radiation in a Vertically Internally Inhomogeneous Medium. Journal of the Atmospheric Sciences, 75(1):41–55. https://doi.org/10.1175/JAS-D-17-0104.1.

A Appendix: WRF Namelist example

Below are examples for the `namelist.wps` and `namelist.input` files shown.
The examples show the configuration for a 4-day (96-hour) simulation, starting at 1.1.2019 00 UTC.

Example of `namelist.wps` :

```
&share
 wrf_core = 'ARW',
 max_dom = 1,                           !specifying number of domains
 start_date = '2019-01-01_00:00:00'     !specifying the start time of the model run
 end_date   = '2019-01-05_00:00:00'     !specifying the end time
 interval_seconds = 21600               !specifying the interval between the boundary condition files
 io_form_geogrid = 2,
/
&geogrid
 parent_id         =   1,
 parent_grid_ratio =   1,
 i_parent_start    =   1,               !specifying the first index for the staggered dimension in x
 j_parent_start    =   1,               !specifying the first index for the staggered dimension in y
 e_we              =  230,              !specifying the last index for the staggered dimension in x
 e_sn              =  170,              !specifying the last index for the staggered dimension in y
 geog_data_res = 'maxsnowalb_ncep+albedo_ncep+default'
 dx = 10000,                            !specifying the grid cell size in x
 dy = 10000,                            !specifying the grid cell size in y
 map_proj = 'lambert',                  !specifying the type of map projection
 ref_lat   =  59,                       !specifying the reference latitude
 ref_lon   =  8,                        !specifying the reference longitude
 truelat1  =  54.0,                     !specifying true latitude 1
 truelat2  =  54.0,                     !specifying true latitude 2
 stand_lon =  8.0,                      !specifying the standard longitude
 geog_data_path = '/Path/to/files'
/
&ungrib
 out_format = 'WPS',
 prefix = 'FILE',
/
&metgrid
 fg_name = 'FILE'
 io_form_metgrid = 2,
/
```

Example of `namelist.input` :

```
 ! First the initial start time, model run duration and end time must be specified corresponding to namelist.wps
 &time_control
 run_days                            = 4,
 run_hours                           = 0,
 run_minutes                         = 0,
 run_seconds                         = 0,
 start_year                          = 2019,
 start_month                         = 01,
 start_day                           = 01,
 start_hour                          = 00,
 end_year                            = 2019,
 end_month                           = 01,
 end_day                             = 05,
 end_hour                            = 00,
 interval_seconds                    = 21600
 input_from_file                     = .true.,
 history_interval                    = 60,
 frames_per_outfile                  = 1000,
 restart                             = .false.,
 restart_interval                    = 360,
 io_form_history                     = 2
 io_form_restart                     = 2
 io_form_input                       = 2
 io_form_boundary                    = 2
 force_use_old_data                  = T
 /
 &domains
 time_step                           = 60,          !specifying the length of the time step (in seconds)
 time_step_fract_num                 = 0,
 time_step_fract_den                 = 1,
 max_dom                             = 1,
 e_we                                = 230,         !number of grid points in x (corresponding to namelist.wps)
 e_sn                                = 170,         !number of grid points in y (corresponding to namelist.wps)
 e_vert                              = 70,          !specifying the number of vertical layers
 p_top_requested                     = 5000,
 num_metgrid_levels                  = 32,
 num_metgrid_soil_levels             = 4,
 dx                                  = 10000,       !specifying horizontal spacial resolution in direction x
 dy                                  = 10000,       !specifying horizontal spacial resolution in direction y
 grid_id                             = 1,
 parent_id                           = 0,
 i_parent_start                      = 1,
 j_parent_start                      = 1,
 parent_grid_ratio                   = 1,
 parent_time_step_ratio              = 1,
 feedback                            = 1,
 smooth_option                       = 0
! numtiles                            = 6
 /
 &physics
 physics_suite                       = 'CONUS'     !selecting the CONUS physics suite, referred to below as '-1'
 mp_physics                          = -1,
 cu_physics                          = -1,
 ra_lw_physics                       = 24,         !overwriting the physics suite's longwave radiation scheme
                                                   !with option 24, which corresponds to the RRTMG-fast scheme
 ra_sw_physics                       = 24,         !overwriting the physics suite's shortwave radiation scheme
 bl_pbl_physics                      = -1,
 sf_sfclay_physics                   = -1,
 sf_surface_physics                  = -1,
 radt                                = 60,         !specifying the radiative time step (in minutes)
 bldt                                = 0,
 cudt                                = 5,
 icloud                              = 1,
 num_land_cat                        = 21,
 sf_urban_physics                    = 0,
 /
 &fdda
 /
 &dynamics
 hybrid_opt                          = 2,
 w_damping                           = 0,
 diff_opt                            = 1,
 km_opt                              = 4,
 diff_6th_opt                        = 0,
 diff_6th_factor                     = 0.12,
 base_temp                           = 290.
 damp_opt                            = 3,
 zdamp                               = 5000.,
 dampcoef                            = 0.2,
 khdif                               = 0,
 kvdif                               = 0,
 non_hydrostatic                     = .true.,
 moist_adv_opt                       = 1,
 scalar_adv_opt                      = 1,
 gwd_opt                             = 1,
 /
```

```
&bdy_control
spec_bdy_width                      = 5,
specified                           = .true.
/
&grib2
/
&namelist_quilt
nio_tasks_per_group = 0,
nio_groups = 1,
/
! Configuration of the DFI, which must be adjusted to the specified initial start time from above
! For the TDFI (dfi_opt=3), this means 1 hour before as backstop and 1/2 hour after as forward stop
&dfi_control
dfi_opt                             = 3
dfi_nfilter                         = 7
dfi_write_filtered_input            = .true.
dfi_write_dfi_history               = .false.
dfi_cutoff_seconds                  = 3600
dfi_time_dim                        = 1000
dfi_bckstop_year                    = 2018
dfi_bckstop_month                   = 12
dfi_bckstop_day                     = 31
dfi_bckstop_hour                    = 23
dfi_bckstop_minute                  = 00
dfi_bckstop_second                  = 00
dfi_fwdstop_year                    = 2019
dfi_fwdstop_month                   = 01
dfi_fwdstop_day                     = 01
dfi_fwdstop_hour                    = 00
dfi_fwdstop_minute                  = 30
dfi_fwdstop_second                  = 00
/
```

B Appendix: Validation data set

Distributions of the initial validation data set (used as training data set for ac=1, clear sky condition from section 4.2.2 onward).
Cloudy:

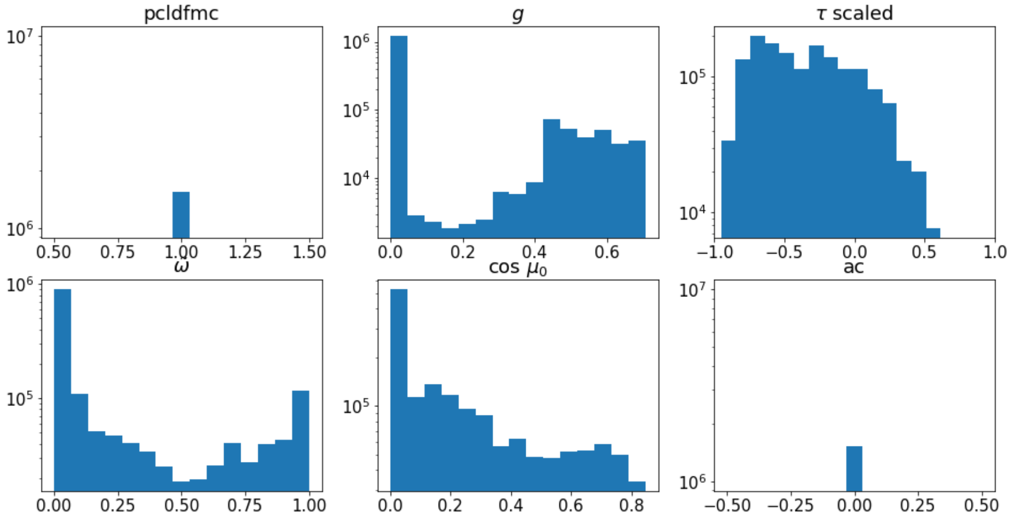


Figure B.1: Input data of the validation data set for cloudy cases (ac=0)

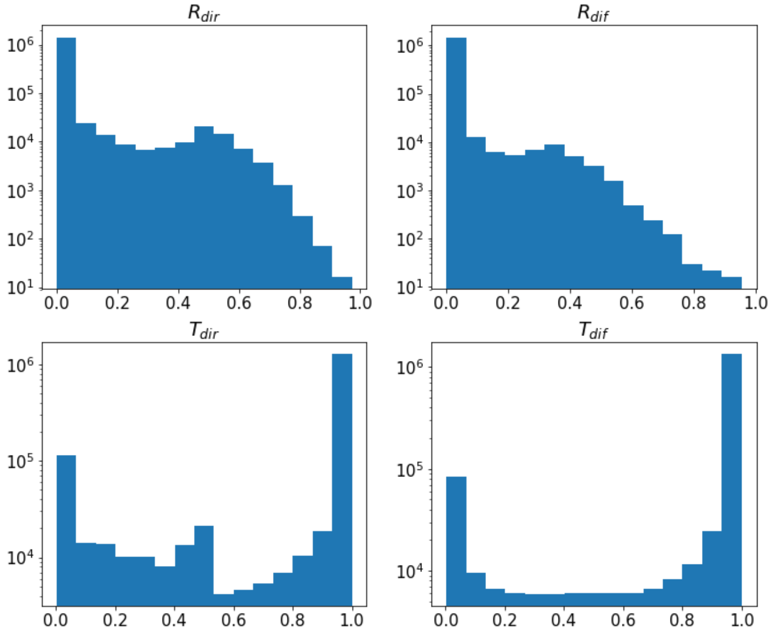


Figure B.2: Output data of the validation data set for cloudy cases (ac=0)

Clear sky:

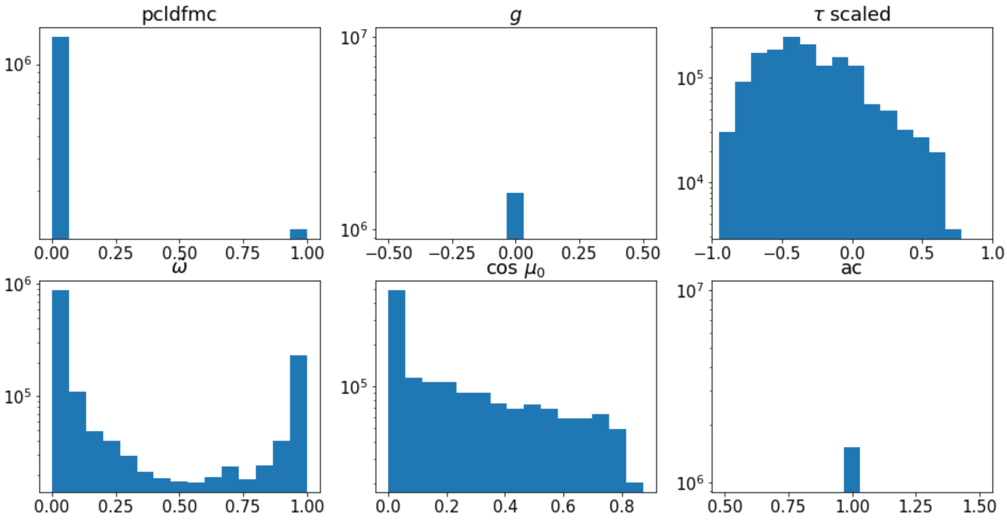


Figure B.3: Input data of the initial validation data set for clear sky cases (ac=1)

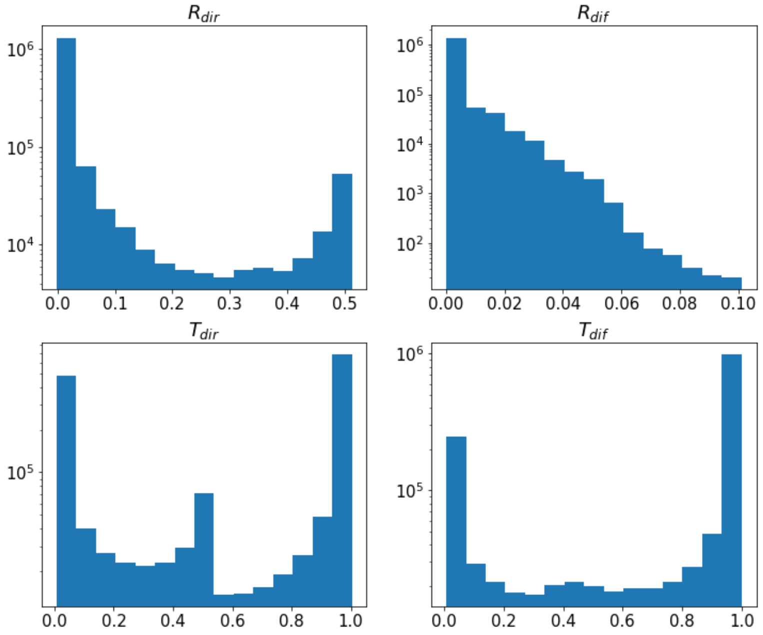


Figure B.4: Output data of the initial validation data set for clear sky cases (ac=1)

C Appendix: Additional plots

Winter simulations (1.1.2019 00 UTC - 5.1.2019 00 UTC):

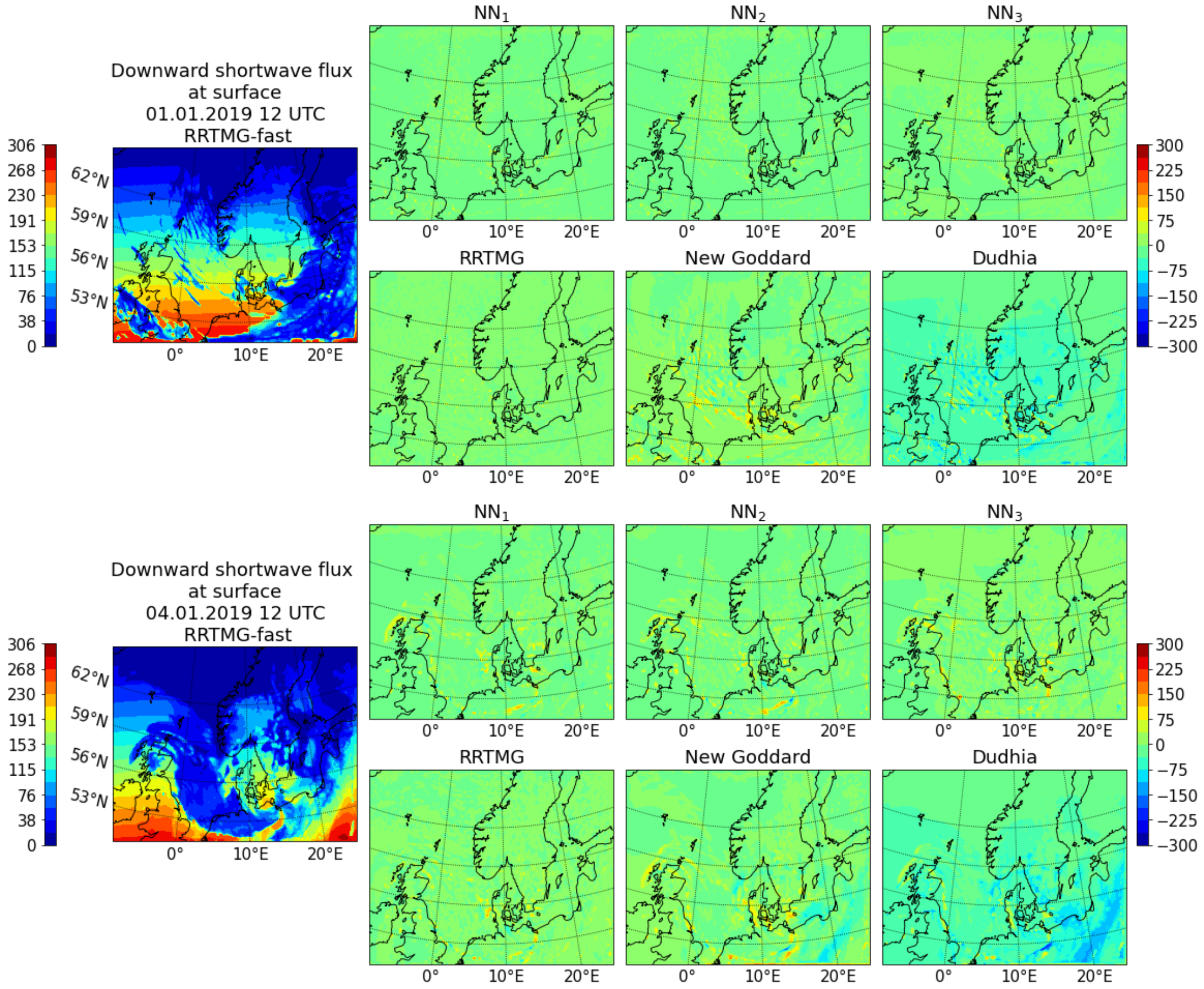


Figure C.1: Plots showing the predictions made for the downward shortwave radiation at the surface (in W/m^2) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name is depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

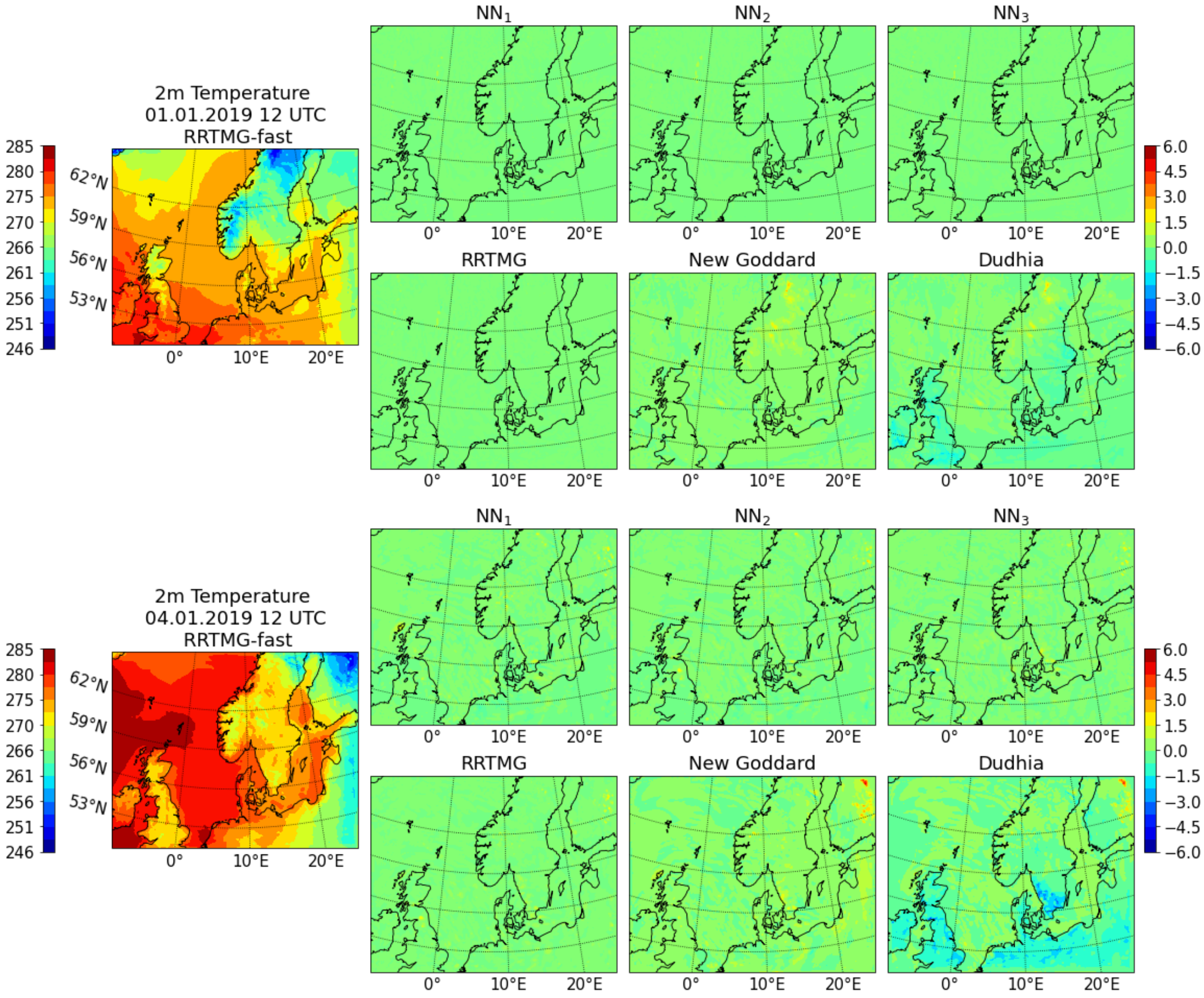


Figure C.2: Plots showing the predictions made for the 2 meter temperature (in K) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name is depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

Spring simulations (1.4.2019 00 UTC - 5.4.2019 00 UTC):

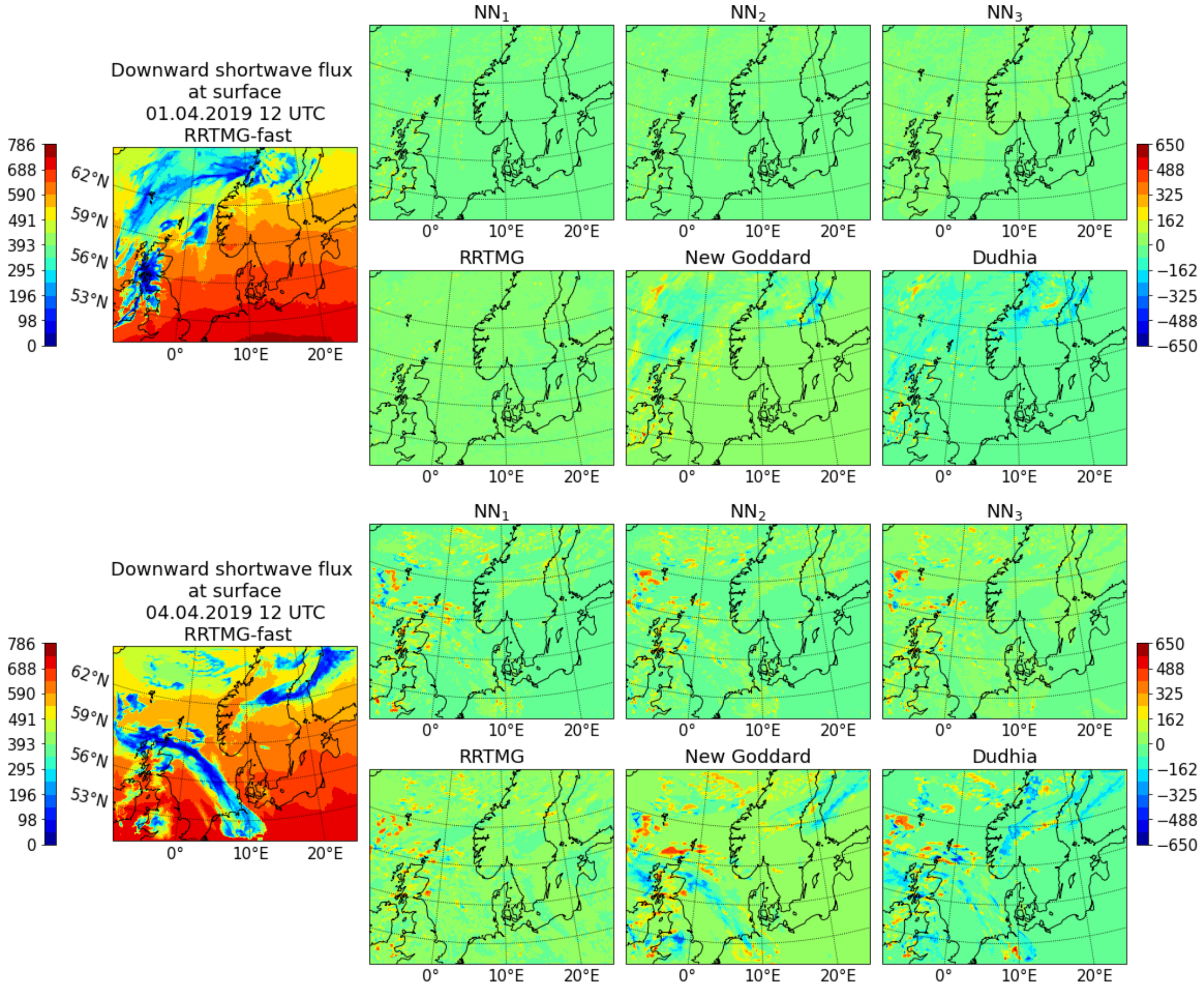


Figure C.3: Plots showing the predictions made for the downward shortwave radiation at the surface (in W/m^2) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name is depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

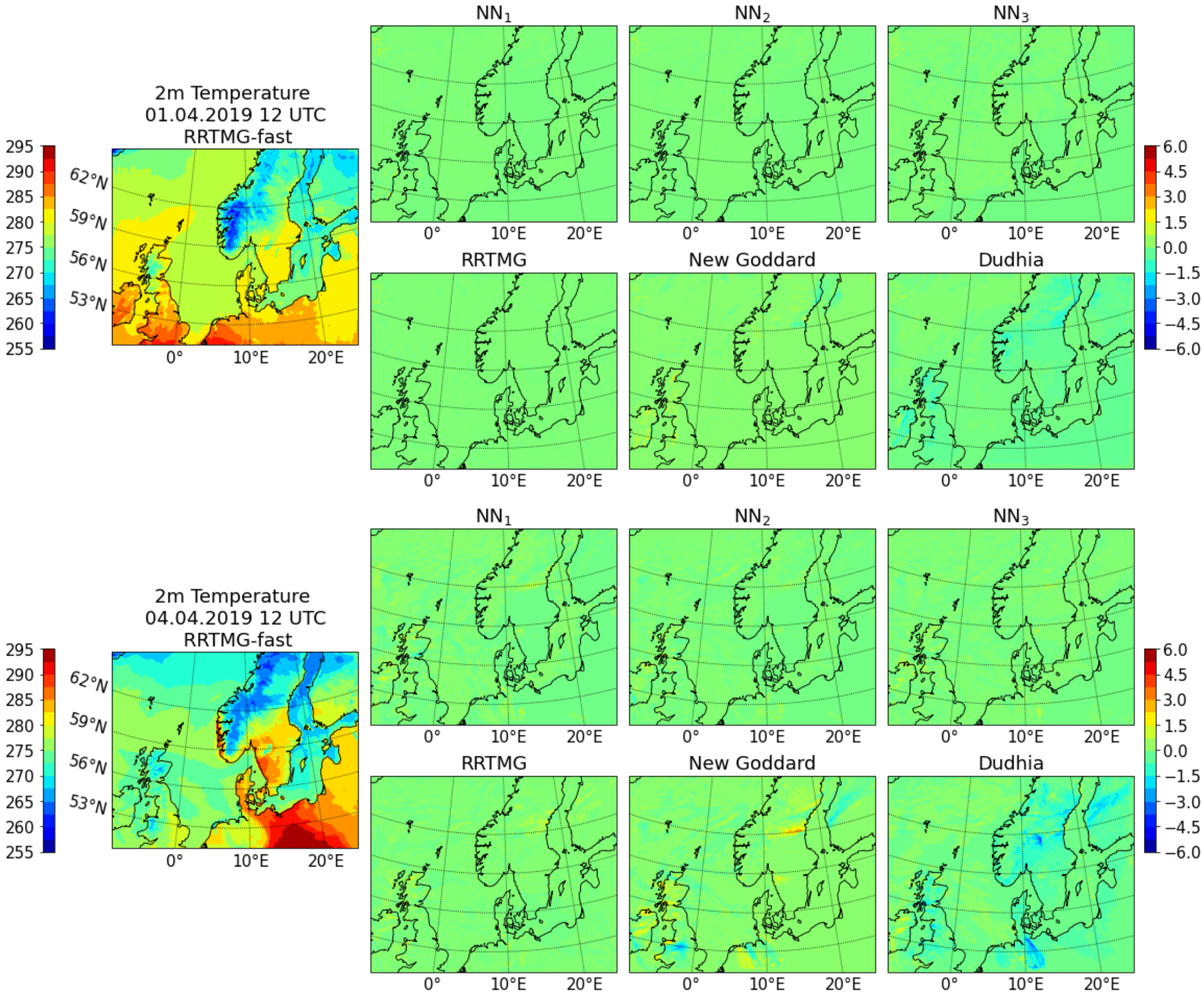


Figure C.4: Plots showing the predictions made for the 2 meter temperature (in K) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name is depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

Autumn simulations (1.10.2019 00 UTC - 5.10.2019 00 UTC):

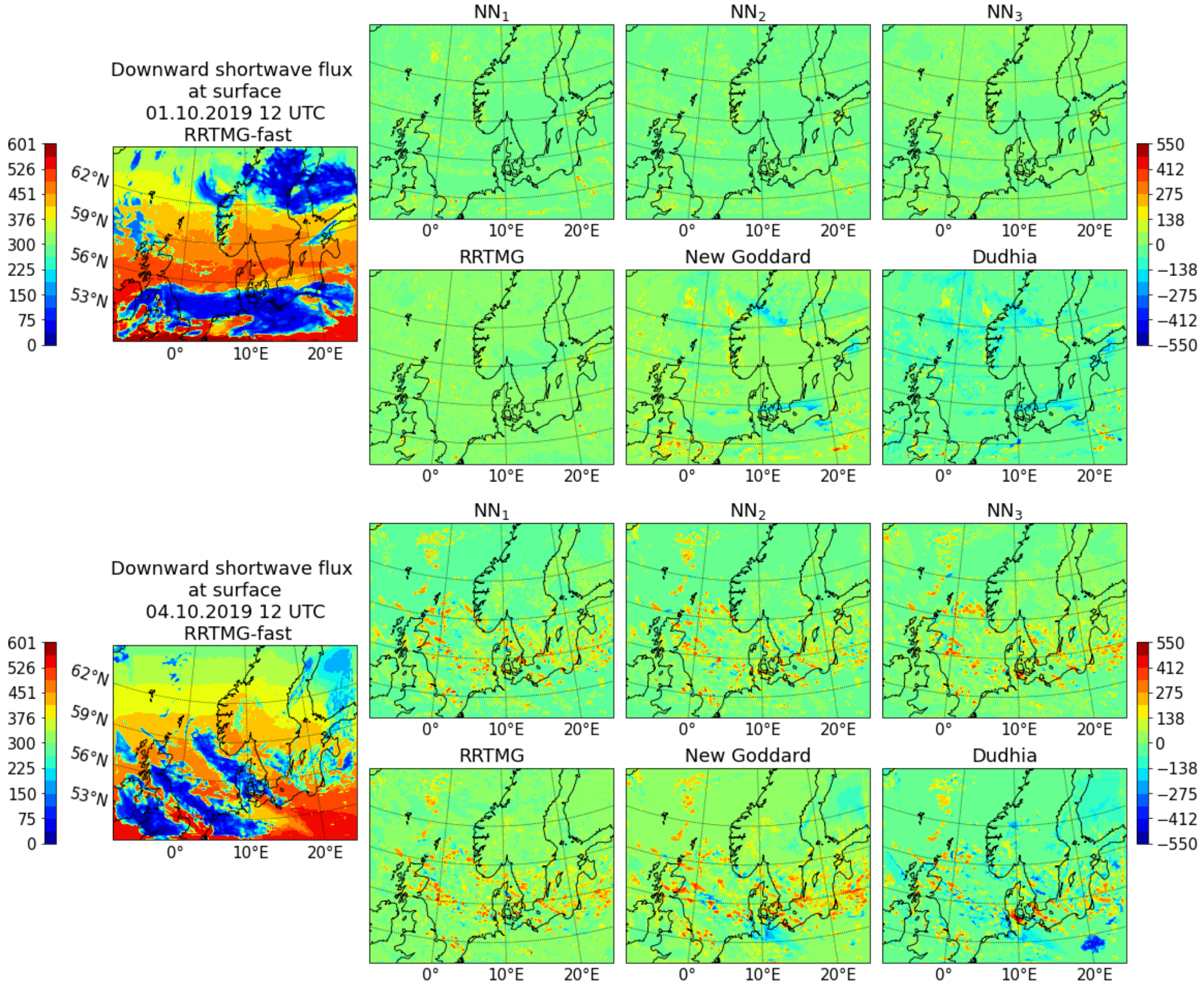


Figure C.5: Plots showing the predictions made for the downward shortwave radiation at the surface (in W/m^2) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name is depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

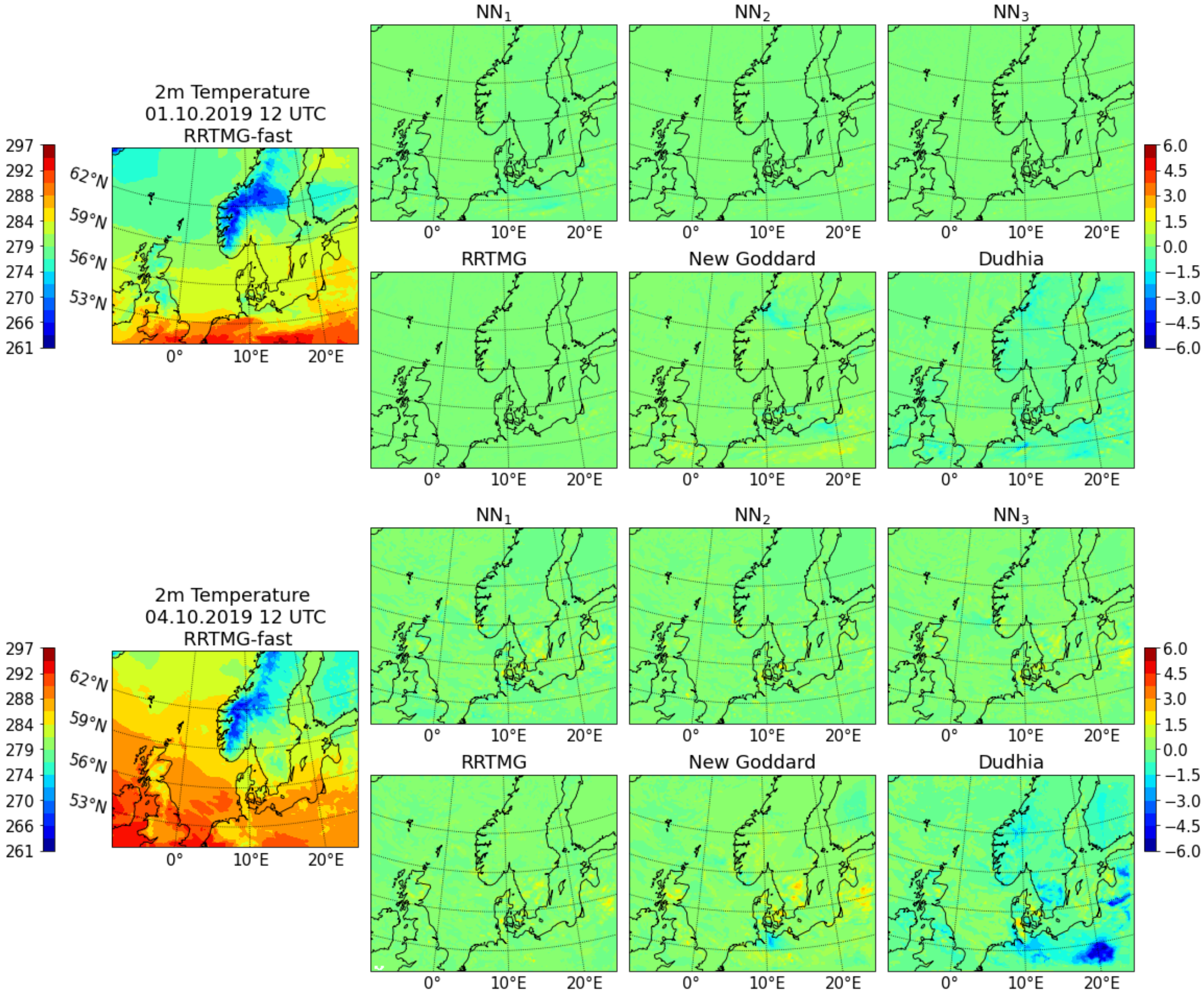


Figure C.6: Plots showing the predictions made for the 2 meter temperature (in K) after 12 hours (upper figure) and 84 hours (lower graphs). The timestamp and variable name is depicted above the figures on the left, showing the predictions made by the RRTMG-fast scheme. The six plots per timestamp on the right show the anomaly fields for the other radiation schemes in contrast to the prediction made by RRTMG-fast, computed as $pred_{scheme} - pred_{RRTMG-fast}$, where the selected scheme is indicated by the figure's title.

List of Figures

List of Tables