Stormscapes: Simulating Cloud Dynamics in the Now

TORSTEN HÄDRICH, KAUST
MIŁOSZ MAKOWSKI, UAM
WOJTEK PAŁUBICKI, UAM
DANIEL T. BANUTI, UNM
SÖREN PIRK, Google AI
DOMINIK L. MICHELS, KAUST

Fig. 1. Three types of thunderstorm supercells simulated with our framework: low precipitation supercell (left), a classical supercell (middle), and a high-precipitation supercell (right).

The complex interplay of a number of physical and meteorological phenomena makes simulating clouds a challenging and open research problem. We explore a physically accurate model for simulating clouds and the dynamics of their transitions. We propose first-principle formulations for computing buoyancy and air pressure that allow us to simulate the variations of atmospheric density and varying temperature gradients. Our simulation allows us to model various cloud types, such as cumulus, stratus, and stratoscumulus, and their realistic formations caused by changes in the atmosphere. Moreover, we are able to simulate large-scale cloud super cells – clusters of cumulonimbus formations – that are commonly present during thunderstorms. To enable the efficient exploration of these stormscapes, we propose a lightweight set of high-level parameters that allow us to intuitively explore cloud formations and dynamics. Our method allows us to simulate cloud formations of up to about 20 km × 20 km extents at interactive rates. We explore the capabilities of physically accurate and yet interactive cloud simulations by showing numerous examples and by coupling our model with atmosphere measurements of real-time weather services to simulate cloud formations in the now. Finally, we quantitatively assess our model with cloud fraction profiles, a common measure for comparing cloud types.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Cloud Simulation, Fluid Dynamics, Physics-based Modeling and Simulation, Stormscapes, Weather Simulation.

Authors' addresses: Torsten Hädrich, KAUST, Visual Computing Center, Thuwal 23955, KSA; Miłosz Makowski, UAM, Umultowska 87, 61-614 Poznań, Poland; Wojtek Pałubicki, UAM, Umultowska 87, 61-614 Poznań, Poland; Daniel T. Banuti, UNM, Mechanical Engineering, Building #122, Albuquerque, NM 87131, USA; Sören Pirk, Google AI, 1600 Amphitheatre Parkway, Mountain View, CA, 94043, USA; Dominik L. Michels, KAUST, Visual Computing Center, Thuwal 23955, KSA.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2020 Copyright held by the owner/author(s).
0730-0301/2020/12-ART175
https://doi.org/10.1145/3414685.3417801

ACM Reference Format:
Torsten Hädrich, Miłosz Makowski, Wojtek Pałubicki, Daniel T. Banuti, Sören Pirk, and Dominik L. Michels. 2020. Stormscapes: Simulating Cloud Dynamics in the Now. *ACM Trans. Graph.* 39, 6, Article 175 (December 2020), 16 pages. https://doi.org/10.1145/3414685.3417801

1 INTRODUCTION

Clouds are among the most common weather phenomena that one can observe. Ranging from dense layers of ground fog or during overcast skies, to the well known 'cotton-like' appearance at lower altitudes, and even their absence during blue skies, clouds are the result of the complex interplay of various physical and meteorological phenomena that define the atmosphere. Due to their enthralling visual appearance, cloud formations evoke fascination and fear alike, which makes clouds an important visual asset for the content creation in visual media, such as games or movies.

In computer graphics, existing methods for clouds either focus on simulating specific cloud types [Goswami and Neyret 2017; Harris et al. 2003], their interactive generation [Dobashi et al. 2008], representations to enable cloud formations [Vimont et al. 2020], or the realistic rendering [Bitterli et al. 2018; Novák et al. 2018]. Moreover, it has even been recognized that clouds and other weather phenomena can be used as a means for modeling urban landscapes [Garcia-Dorado et al. 2017]. However, despite these advances the degree of realism and the variety of simulated clouds is still limited and the physically accurate simulation of cloud dynamics has only been partially covered by the visual computing community.

Due to their presence in our daily lives and due to their importance for weather prediction, the motivation to understand clouds and their dynamics also extends far beyond common applications in visual computing. Existing methods in meteorology, engineering, and environmental sciences, aim to realistically simulate atmospheric conditions to enable more reliable predictions of weather

phenomena – often over large spatial extents and based on complex and physically accurate models [Houze 1994; Yau and Rogers 1996]. Most of these models are computationally demanding and therefore prevent the interactive exploration of cloud dynamics. We argue that fast and yet realistic simulations of clouds play a significant role for further understanding weather phenomena, also for the data generation of learning-based approaches [Kim et al. 2019; Ummenhofer et al. 2020; Wiewel et al. 2019].

In this paper, we advance simulations of clouds by introducing a new physical model that allows expressing various cloud types and their transitions, as well as more dynamic phenomena such as the formation of thunderheads, mesocyclones, and super-cell thunderstorms, that we refer to as *stormscapes*. We enable these simulations by introducing novel formulations for buoyancy and temperature distributions based on first-principles. Buoyancy is defined so as to consider variations of density in the atmosphere, while our formulation for temperature profiles allows for varying temperature gradients, including temperature inversion profiles. Unlike existing models, this enables us to create conditions necessary to simulate cumulonimbus cloud formations and even more complex phenomena like the Kelvin-Helmholtz instability. Figure 1 shows three supercell thunderstorms simulation with our framework.

To provide users with an intuitive way to interact with cloud formations, we identify a lightweight set of high-level physical parameters that determine a weather scenario. Specifically, we find that ground humidity controls the cloud base altitude, while the atmospheric temperature lapse rate independently controls the cloud top altitude. Our model is implemented as a grid-based fluid solver to allow for the efficient simulation of complex cloud formations over large spatial extents. To further increase the realism of our simulations we couple the fluid dynamics with detailed representations of terrain obtained from real map data. Together this enables physically accurate simulations and the interactive exploration of large-scale stormscapes.

In summary, our key contributions are: (1) we propose novel first-principle formulations for computing buoyancy and air pressure that allow us to simulate the variations of atmospheric density and varying temperature gradients; (2) we incorporate these formulations into a grid-based fluid dynamics model allowing for the physically accurate simulation of various cloud types and transitions between them, including cumulus, stratus, fog, stratocumulus, and cumulonimbus; (3) we introduce a lightweight set of parameters that allow us to intuitively explore and model cloud formations; (4) we successfully address the simulation of several complex weather phenomena, such as thunderheads, mesocyclones, and various types of super-cell thunderstorms; (5) we quantitatively assess our model using cloud fraction profiles, a common measure for comparing cloud types; (6) finally, we couple our model with data of real-time weather services to interactively explore cloud formations in the now.

2 RELATED WORK

Modeling and rendering complex natural phenomena such as fog [Blinn 1982], fluids [Kass and Miller 1990; Stam 1999], or clouds [Kajiya and Von Herzen 1984], has a long tradition in computer graphics. Today, a large body of work exists to model and simulate fluid dynamics with their many intricate details [Bridson 2008]. While this spans a breadth of work that we cannot conclusively discuss, here we provide an overview of fluid dynamics, weather phenomena, and cloud simulation models related to our work.

Fluid Dynamics. Since the seminal work of Stam [1999] many methods have been introduced to improve solvers for two way coupling [Lu et al. 2016; Teng et al. 2016; Zhu and Bridson 2005], based on reduced order methods [Gupta and Narasimhan 2007; Jones et al. 2016; Treuille et al. 2006], Smoothed Particle Hydrodynamics [Ihmsen et al. 2014; Koschier et al. 2019], or with an emphasis on compute and memory efficiency [Ferstl et al. 2014; Losasso et al. 2004; McAdams et al. 2010; Setaluri et al. 2014; Zehnder et al. 2018]. Numerous methods focus on even more intricate features of fluid flows, for example based on eigenfunctions [Cui et al. 2018], momentum transfer and regional projections [Zhang et al. 2016], style-transfer [Sato et al. 2018b], optimization [Inglis et al. 2017], or based on narrow band representations [Ferstl et al. 2016]. More recently, it has also been recognized that neural networks provide a powerful means to represent details of fluids, for example with an emphasis on temporal coherency [Xie et al. 2018], liquid splash modeling [Um et al. 2018], Lagrangian simulations [Ummenhofer et al. 2020], or even style-transfer [Kim et al. 2020].

Cloud Rendering. While early methods for rendering clouds specifically focus on the efficient rendering based on impostors [Harris and Lastra 2001], more advanced methods explicitly simulate light scattering based on estimating particles present in clouds [Nishita et al. 1996] or by approximating the energy transport and scattering of light [Bouthors et al. 2006, 2008]. Recent methods rely on more accurate algorithms, such for computing non-exponential free-flight distributions [Bitterli et al. 2018], fractional Gaussian fields [Guo et al. 2019], radiative transfer [Miller et al. 2019], or – more generally – Monte Carlo Integration [Novák et al. 2018] for modeling light transport in volumetric and participating media. While all these methods aim to improve the visual fidelity of clouds or other volumetric media, we focus on simulating cloud formations by modeling the atmospheric effects that generate clouds.

Cloud Modeling and Artistic Control. A number of methods aim to realistically generate clouds along with their transitions and temporal evolution; prominent examples include fractals [D. Ebert, F. Musgrave, D. Peachey, K. Perlin, and S. Worley. 2002; Peitgen and Saupe 1988; Voss 1988], noise [Lagae et al. 2010], and cellular automata [Dobashi et al. 2000]. Furthermore, due to the importance for content creation a number of methods aim to model clouds while facilitating artistic control through user-defined sketches [Stiver et al. 2010; Wither et al. 2008], morphing [Yu and Wang 2011], keyframing [Webanck et al. 2018], or animation [Schpok et al. 2003]. Dobashi et al. [2008] and Sato et al. [2018a] introduce sketch-based interfaces for cloud-shape control with plausible simulations of clouds, while other methods focus on extracting visual features from images to correlate them with simulated clouds [Dobashi et al. 2010; Yuan et al. 2014]. Garcia-Dorado [2017] go even further and employ a weather model along with urban procedural modeling of cities. Their approach specifically allows for sketching terrain properties that are used as initial conditions for weather simulations. While these methods provide ways to control the modeling and

authoring of clouds, we introduce a physically accurate model for cloud simulations along with an intuitive set of parameters to control their formations.

Cloud and Weather Simulations. Physics-based methods aim at simulating clouds as meteorological phenomena in the atmosphere. Among the first methods is the work of Kajiya and Von Herzen [1984] who introduce a dynamical model for generating cloud animations. A number of methods rely on Eulerian fluid solvers for the interactive simulation of clouds that carefully balance physical plausibility and compute limitations [Harris et al. 2003; Miyazaki et al. 2002; Overby et al. 2002]. Due to the complexity of cloud simulations, also considering the often very large spatial extend, numerous methods introduce refined representations for simulating clouds. This ranges from geometric- and particle-based representations [Bouthors and Neyret 2004; Gardner 1985; Neyret 1997], coupled map lattices for extended cellular automata [Miyazaki et al. 2001], position-based dynamics [Ferreira Barbosa et al. 2015] to layer-based approaches [Vimont et al. 2020]. Due to these representations a wide range of cloud phenomena can be simulated realistically and efficiently. Goswami and Neyret [2017] introduce a Lagrangian approach for a convective cloud simulation that by definition allows a more convenient way to express boundary and atmospheric conditions. On another trajectory, Duarte and Gomes [2017] introduce a real-time approach for simulating cumulus clouds by exploiting real atmospheric measurements represented as sounding curves.

Similar to many of these approaches, we aim to simulate clouds in a physically accurate manner; however, unlike them our cloud model is based on first-principles and implemented as an integrated fluid solver that does not require a specialized representation, such as particles or layers. This offers the advantage that our method can more realistically simulate cloud various and cloud phenomena that are not captured by existing methods.

3 OVERVIEW

The main motivation for our approach is to enable the realistic simulation of cloud types and their dynamic transitions. From a meteorology perspective we are interested in simulating the atmospheric conditions in the troposphere (i.e. ground level up to 13 km). Clouds form when a parcel of humid air is cooled below the saturation point of the contained water vapor. This includes 'warm' clouds such as cumulus, stratocumulus, stratus, altocumulus, altostratus, and cumulonimbus and nimbostratus, as well as 'cold' clouds that form from ice, such as cirrus, cirrostratus, and cirrocumulus. Each cloud type can have further variations of so called cloud species, such as cumulonimbus capillatus and cumulonimbus incus clouds. Figure 2 shows an illustration of the various cloud types and at which altitudes they appear. For a more thorough overview on cloud dynamics we refer the interested reader to the books of Yau and Rogers [1996] and Houze [1994].

In this paper, we restrict ourselves to 'warm' clouds for which the transition is limited to liquid water. Specifically, we focus on simulating cumulus, stratocumulus, cumulonimbus, and stratocumulus clouds, as well as their transitions. Additionally, we simulate advanced cloud phenomena that are not captured by previous methods, such as different types of supercells, thunderheads, mesocyclones,

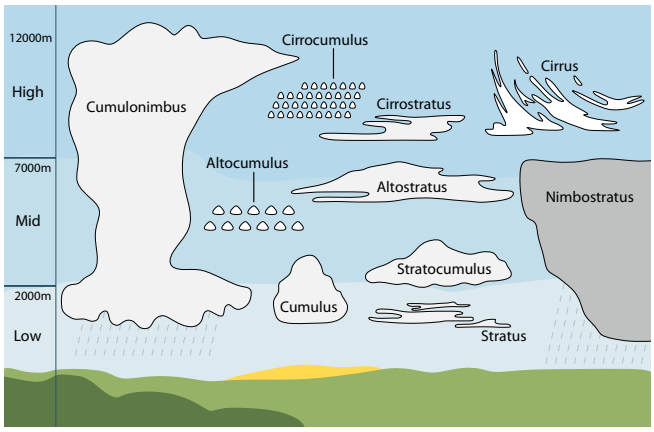


Fig. 2. Illustration of the tropospheric cloud classification: altocumulus, altostratus, cumulonimbus, cirrocumulus, cirrus, cirrostratus, cumulonimbus, nimbostratus, stratocumulus, and stratus. Illustration inspired by Valentin de Bruyn (CC BY-SA 3.0).

and precipitation in the form of rain. Cloud supercells [Grant and Van Den Heever 2014] are clusters of cumulonimbus formations commonly present during thunderstorms resulting in the formation of complex stormscapes.

The realistic simulation of these phenomena includes complex processes such as the exchange of water, vapor and heat between parcels of air and the terrain. These dynamics can lead to fast, turbulent air flows, which are challenging to compute numerically. We introduce height maps and noise functions for vapor and heat as light-weight means to represent the water and temperature exchange between terrain and air. Further, we allow users to specify wind conditions as an external velocity field acting on the fluid simulation. We develop a model derived from thermo- and fluid dynamics that is comprised by two sets of parameters, one for the dynamic control of cloud formation and another for describing the atmospheric background. This enables us to express the transitions between different cloud types and stormscapes.

In the following sections we introduce our cloud dynamics model and showcase its capabilities through a number of experiments. We simulate various cloud types, the transitions between clouds, and the formation of complex stormscapes. To validate the plausibility of our model we use cloud fraction profiles to quantitatively assess cloud formation. Finally, to explore the capabilities of physically accurate and yet interactive simulations we couple our model with atmospheric measurements of real-time weather services to simulate cloud formations in the now.

4 MODEL

In this section we provide an overview of our cloud dynamics model, which can be divided into an atmospheric model that describes temperature and pressure changes as a function of altitude, a 0D thermodynamics model that determines local forces and the formation of clouds, and the fluid dynamic model that determines the motion of humid air in the atmosphere.

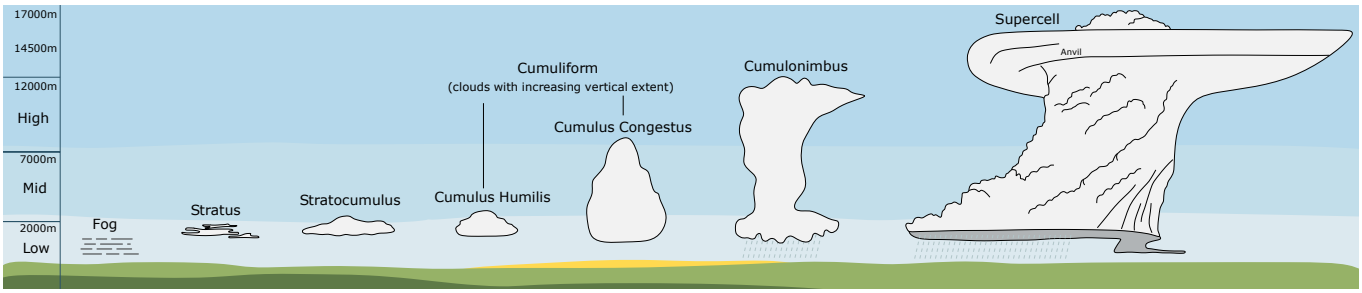


Fig. 3. Left to right: illustration of the development of a cumulonimbus cloud further becoming a part of a supercell (right).

4.1 Atmosphere

The background atmosphere is determined by the temperature distribution and the fundamental hydrostatic equation [Houze 1994] for compressible fluids.

4.1.1 Atmospheric Pressure. The hydrostatic equation relates the density ρ of a fluid to the pressure gradient,

$$\mathrm{d}_z p = -\rho g\,, \tag{1}$$

in which $g \approx 9.81\ \mathrm{m/s^2}$ denotes the acceleration due to gravity. In an ideal gas, pressure p is related to density and temperature T as $\rho = p/(RT)$, where $R = \mathcal{R}/M$ is the specific gas constant, i.e. the ratio of the general gas constant $\mathcal{R}$ and the molar mass M. We introduce a temperature distribution analogous to the 'International Standard Atmosphere' (ISA; see Appendix A) using the temperature lapse rate $\mathrm{d}_z T = \Gamma$, so that Eq. (1) can be integrated to

$$\int_0^z \frac{dp}{p} = \frac{g}{R}\int_0^z \frac{dz}{T(z)} = \frac{g}{R}\int_0^z \frac{dz}{T_0 + \Gamma z}\,, \tag{2}$$

yielding an equation for pressure as a function of altitude,

$$p(z) = p_0\left(1 - \frac{\Gamma z}{T_0}\right)^{\frac{g}{R\Gamma}}\,. \tag{3}$$

4.1.2 Temperature Profile. Interesting weather phenomena occur at an inversion layer at high altitudes, for which we introduce a second lapse rate Γ_1, valid beyond an altitude of z_1, so that the temperature profile is described as

$$\begin{aligned} 0 \le z \le z_1 &: \quad T(z) = T_0 + \Gamma_0 z\,, \\ z_1 < z &: \quad T(z) = T_{z1} + \Gamma_0 z = T_0 + \Gamma_0 z_1 + \Gamma_1(z - z_1)\,. \end{aligned} \tag{4}$$

An interesting stratification is given for $\Gamma_0 < 0$, $\Gamma_1 > 0$, then the temperature inversion at z_1 would act as an obstacle for the rising thermal, allowing the distinct flattened top (the 'anvil') of a cumulonimbus to form. Usually, we employ a simple inversion by making use of a single lapse rate Γ setting $\Gamma_0 := \Gamma$ and $\Gamma_1 := -\Gamma$.

4.2 Thermodynamics

The thermodynamics model based on first-principles provides local descriptions of air-water mixture properties, the temperature change of the rising thermal, the buoyancy and thus resulting local acceleration, and the phase transitions between vapor, cloud, and rain.

4.2.1 Generalities. In the present model, the background air is assumed to be dry. Then its molar mass is constant with

$$\mathcal{M}_{\mathrm{air}} = 28.96\ \mathrm{g/mol}\,. \tag{5}$$

The thermal (i.e. a column of rising air) is an air-water mixture, with a water vapor mole fraction X_V, and the water molar mass

$$\mathcal{M}_W = 18.02\ \mathrm{g/mol}\,, \tag{6}$$

so that, using the water vapor mole fraction X_V, an average molar mass for the humid air in the thermal can be calculated as

$$\mathcal{M}_{\mathrm{th}} = X_V \mathcal{M}_W + (1 - X_V)\mathcal{M}_{\mathrm{air}}\,. \tag{7}$$

Mole fractions X and mass fractions Y are related through

$$Y_V = X_V \frac{\mathcal{M}_W}{\mathcal{M}_{\mathrm{th}}}\,. \tag{8}$$

Following Kessler [Kessler 1969], the amount of water in the atmosphere can be expressed in terms of the mass ratio q_i of water per mass unit of dry air, specifically vapor q_v, cloud q_c, and rain q_r. The mole fractions X_i are related to the mass ratios q_i via

$$X_i = \frac{q_i}{q_i + 1}\,. \tag{9}$$

4.2.2 Temperature in the Rising Thermal. The warm and humid air in a thermal rises when it has a lower density than the surrounding air. The temperature profile of the atmosphere is prescribed and ultimately a function of the weather, whereas the rising thermal changes its local temperature because of its expansion to lower pressures at higher altitudes. Heat exchange with the surrounding atmosphere can be neglected, and the process is slow enough to assume an isentropic expansion to take place.

The change in pressure directly determines the change in temperature through the classical isentropic relations [Anderson 2003]

$$T_{\mathrm{th}}(z) = \hat{T}\left(\frac{p(z)}{\hat{p}}\right)^{\frac{\gamma_{\mathrm{th}}-1}{\gamma_{\mathrm{th}}}}\,, \tag{10}$$

where $\hat{T}$ and $\hat{p}$ are the conditions at the ground. The isentropic exponent γ for water vapor and air are, respectively $\gamma_{\mathrm{air}} = 1.4$ and $\gamma_V = 1.33$, so that the isentropic exponent of the humid thermal is approximately

$$\gamma_{\mathrm{th}} = Y_V \gamma_V + (1 - Y_V)\gamma_{\mathrm{air}}\,, \tag{11}$$

where Y_V is the mass fraction of water in the thermal.

4.2.3 Buoyancy and Acceleration of the Thermal. A difference in density between atmosphere and thermal will result in a vertical acceleration following Archimedes' principle. The humid thermal air parcel of volume V and density ρ_{th} experiences an upward lift force in air of density ρ_{air} of

$$L = Vg(\rho_{\text{air}} - \rho_{\text{th}})\,. \tag{12}$$

Then, the buoyant acceleration on this thermal parcel, from Newton's second law $F = ma$, is given by

$$B = \frac{L}{m} = \frac{Vg(\rho_{\text{air}} - \rho_{th})}{V\rho_{\text{th}}} = g\left(\frac{\rho_{\text{air}}}{\rho_{\text{th}}} - 1\right)\,. \tag{13}$$

We can treat thermal and air as ideal gases, for which

$$\rho = \frac{p}{RT} = \frac{p}{(R/M)T}\,, \tag{14}$$

with the universal gas constant $\mathcal{R} = 8314$ J/g K and the gas molar mass $\mathcal{M}$. The pressure at every altitude is equilibrated between the thermal and the surrounding air, so that Eq. (13) can be simplified to

$$B = g\left(\frac{\mathcal{M}_{\text{air}}}{\mathcal{M}_{\text{th}}}\frac{T_{\text{th}}}{T_{\text{air}}} - 1\right)\,. \tag{15}$$

The air temperature $T_{\text{air}} = f(z)$ is specified by our atmospheric model, Eq. (4), and the thermal temperature from the cooling due to expansion, Eq. (10), the molar masses $\mathcal{M}_i$ as introduced in Sec. 4.2.1.

4.2.4 Condensation and Cloud Formation. The initial water content rises with the thermal. However, the rising humid air in the thermal cools down as it gains altitude and loses capacity to keep the water vapor solved. When the partial pressure on the water vapor in the air drops below the saturation pressure at the local temperature, the excess water vapor that cannot be solved in the air anymore condenses into tiny droplets – clouds. This saturation mixing ratio $q_{vs}(T,p)$ can be approximated for the relevant temperature range as

$$q_{vs}(T,p) = \frac{380.16}{p}\exp\left(\frac{17.67T}{T + 243.50}\right) \tag{16}$$

for given temperature and pressure in Celsius respectively Pascal [Yau and Rogers 1996].

4.2.5 Heat Release from Condensation. In order to account for the energy released by water vapor condensing into droplets when saturation is reached at the base of the cloud, the temperature of the thermal, Eq. (4) has to be extended. The energy release per condensed water mass fraction X_C is the latent heat L. The associated temperature increase ΔT_V then depends on the heat capacity, so that

$$T_{\text{th}}(z) = \hat{T}\left(\frac{p(z)}{\hat{p}}\right)^{\frac{\gamma_{\text{th}}-1}{\gamma_{\text{th}}}} + \Delta T_V = \hat{T}\left(\frac{p(z)}{\hat{p}}\right)^{\frac{\gamma_{\text{th}}-1}{\gamma_{\text{th}}}} + \frac{L}{c_p}X_C\,. \tag{17}$$

Now, the heat capacity c_p of the air in the thermal is required. This is with respect to only the gaseous part of the thermal, i.e. dry air and water vapor X_V,

$$c_p^{\text{th}} = \frac{\gamma_{\text{th}}\mathcal{R}}{\mathcal{M}_{\text{th}}(\gamma_{\text{th}} - 1)}\,, \tag{18}$$

with the thermal air molar mass $\mathcal{M}_{\text{th}}$ from Eq. (7), and the thermal air ratio of specific heats γ_{th} from Eq. (11).

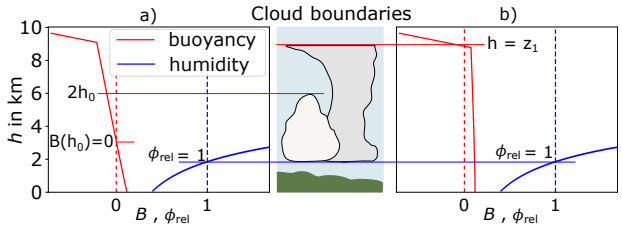
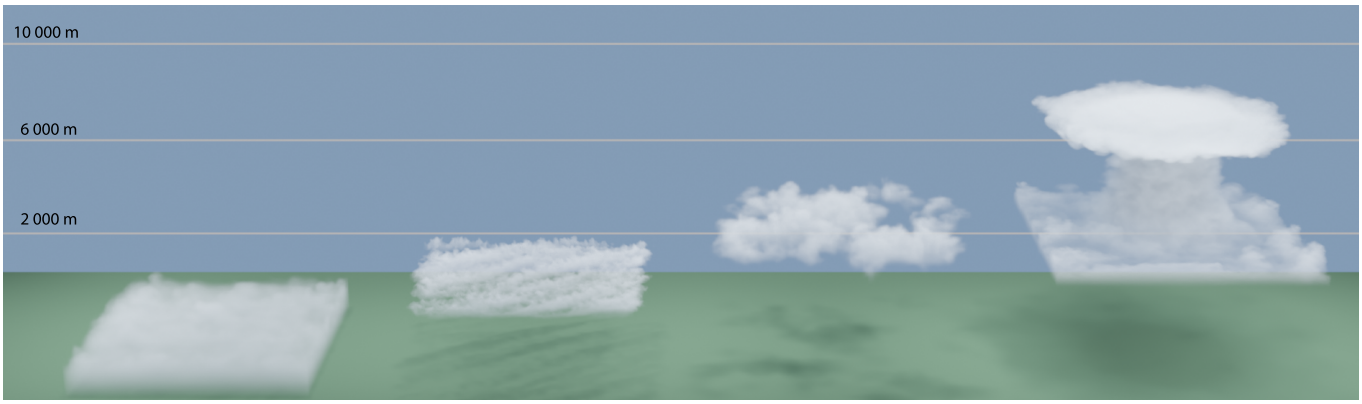


Fig. 4. Illustration of relation between atmospheric profile data and cloud boundaries. A warm/humid thermal will form on the ground and rise up, cooling down in the process. The cloud base will form at the point where the relative humidity ϕ_{rel} has increased to unity due to the reduced temperature. (a) the thermal will be accelerated upwards to neutral buoyancy $B(h_0) = 0$, and will be decelerated beyond. For the linear buoyancy profiles found here, the top of the cloud will be found where the thermal has come to rest at approximately $h = 2h_0$. (b) An inversion layer will stop the upward draft of the air that has not been decelerated, creating the characteristic flat anvil shape of the cumulonimbus cloud top.

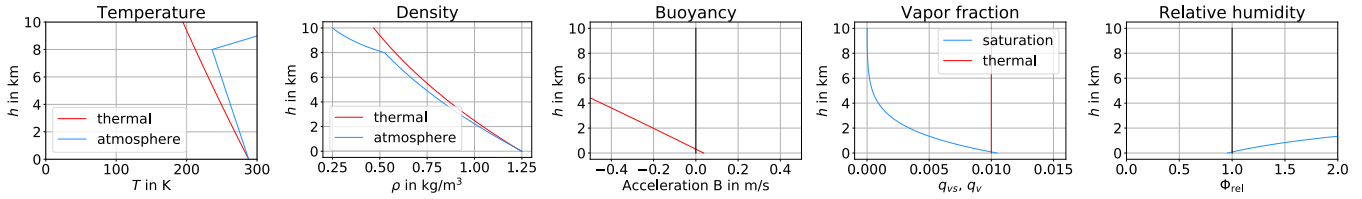
4.2.6 Prediction of Cloud Boundaries. This static model already allows for parameter studies and identification of relevant variables. Figure 4 shows how the relative humidity $\phi_{\text{rel}} = q_v/q_{vs}$ and the buoyancy B determine the base and the top of the cloud. As an approximation for this initial exploration, the effect of latent heat is neglected. Specifically, the buoyancy is determined by the temperature gradient; the relative humidity in the thermal is controlled by the temperature and the vapor mass ratio q_v. Figure 5 shows a number of atmospheric profiles of temperature, density, buoyancy, vapor fraction, and relative humidity, to illustrate representative profiles for fog, stratocumulus, cumulus, and cumulonimbus clouds that can be created with this model.

Figure 5 further reveals the physical parameters that can be used to control the cloud dimensions. The cloud base is determined from the point where the relative humidity exceeds unity and the vapor condenses. The top is reached where the upward velocity of the rising thermal vanishes, with the altitude of vanishing buoyancy marking the mid-point altitude.

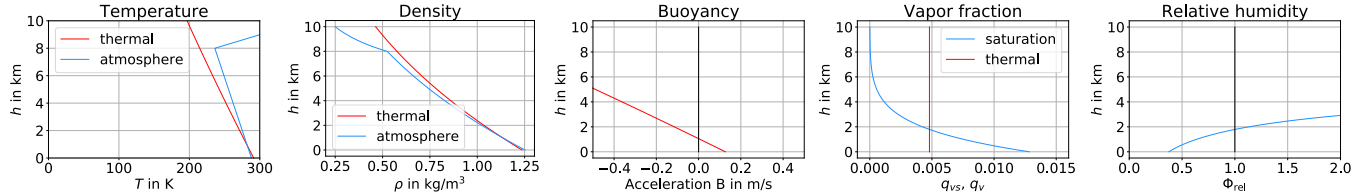
The temperature and density plots are mostly similar across fog, stratocumulus, and cumulus clouds. They differ for cumulonimbus as the temperature values of the thermal are higher relative to the atmospheric temperature values, which can be controlled by the temperature lapse rate Γ. The intersection point of the graphs indicates the altitude of the top of the clouds, which is several dozen meters for fog, 2 000 meters for stratocumulus clouds, 2 500 meters for cumulus clouds and 8 000 meters for cumulonimbus clouds. Alternatively, the top of the clouds is shown by the buoyancy graph. Relative humidity ϕ_{rel} defines the base altitude at which a cloud forms. Fog forms close to the ground surface, while cumulus and cumulonimbus clouds form at an altitude of 1 500 meters and stratocumulus at 1 800 meters. Analogously, this relationship can be expressed by the proportions of vapor fraction q_v and saturation mixing ratio q_{vs}. Varying the parameter values over time allows us to describe the temporal development of weather, such as the transition of fog to stratocumulus clouds. By additionally applying an external (rotating) wind field to the simulation we can transition to a typical summer afternoon thunderstorm.



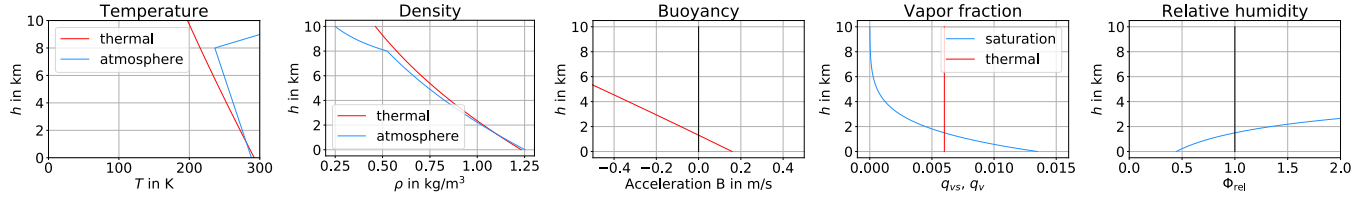
(a) Simulated fog, stratocumulus, cumulus, and cumulonimbus clouds (left to right). The corresponding altitude-temperature, -density, -buoyancy, -vapor fraction and -relative humidity profiles can be found below.



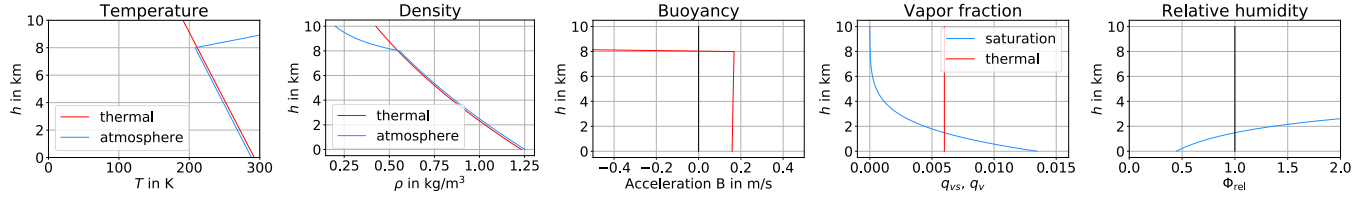
(b) Fog: Vanishing buoyancy at ground level prevents thermals from rising, a relative humidity of one at ground level causes formation of fog.



(c) Stratocumulus: Finite buoyancy at ground causes thermal to rise, with a top of 2000 m, and a base with $\phi_{\text{rel}} = 1$ at 1800 m.



(d) Cumulus: Buoyancy at ground causes thermal to rise, the higher humidity increases the top to 2500 m, and reduces the base with $\phi_{\text{rel}} = 1$ to 1500 m.



(e) Cumulonimbus: a higher temperature lapse rate causes a constant upward acceleration of the thermal to the inversion layer at 8000 m, while maintaining a base of 1500 m.

Fig. 5. Simulation of the development of different cloud types (5a) and corresponding altitude-temperature, -density, -buoyancy, -vapor fraction and -relative humidity profiles: fog (5b), stratocumulus (5c), cumulus (5d), and cumulonimbus (5e) clouds. In accordance with Fig. 4, the cloud base is determined by the vapor reaching saturation conditions with the relative humidity ϕ_{rel} reaching unity; the top of the cloud is where the upward motion of the rising thermal has vanished.

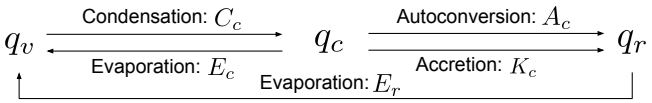


Fig. 6. Illustration of Kessler's scheme [1969] for modeling the transport between vapor, cloud water, and rain.

4.3 Fluid Dynamics

The fluid dynamics model solves the motion of air in the atmosphere, as caused by external wind and differences in density.

4.3.1 Transport Equations. The system of equations for fluid transport is developed from the principles of conservation of mass, energy, and Newton's second law. Following Boussinesq's classical assumption, we account for variable density in the source terms, but assume an approximately constant density in the flow dynamics due to the very low Mach numbers involved. From this follows directly that the momentum and energy equations are decoupled and can be solved independently. Our model is closed by using the water phase transport equations for vapor q_v, condensed cloud q_c, and rain q_r based on Kessler's classical model [1969].

The simulation of fluids is an established and ongoing research topic within the computer graphics community. Among others, Bridson and Müller provided a detailed introduction to fluid simulation from a computer graphics' perspective in their SIGGRAPH course [2007].

As usual, in our model, the fluid's state is described by a velocity field $\boldsymbol{u} : (\boldsymbol{x}, t) \mapsto \boldsymbol{u}(\boldsymbol{x}, t)$ which for given time $t \in \mathbb{R}_0^+$ and position $\boldsymbol{x} \in \mathbb{R}^3$ returns the corresponding local flow $\boldsymbol{u}(\boldsymbol{x}, t) \in \mathbb{R}^3$.

4.3.2 Mass. The conservation of mass prescribes the continuity equation of a solenoidal velocity field as

$$\nabla \cdot \boldsymbol{u} = 0\,. \tag{19}$$

4.3.3 Momentum. The change in momentum according to Newton's second law is expressed in terms of the Navier-Stokes equations, allowing for the solution of the temporal evolution of the velocity field,

$$\partial_t \boldsymbol{u} = -\boldsymbol{u} \cdot \nabla \boldsymbol{u} - \rho^{-1} \nabla p + \nu \nabla \cdot \nabla \boldsymbol{u} + \boldsymbol{b} + \boldsymbol{f}\,, \tag{20}$$

in which the density is denoted by ρ, the pressure by p, and the kinematic viscosity by ν. The first term on the right side of Eq. (20) describes phenomena caused by advection followed by a pressure and a viscosity term. Buoyancy is taken into account by the force $\boldsymbol{b} \in \mathbb{R}^3$ and other external forces are combined and described by an additional external net force $\boldsymbol{f} \in \mathbb{R}^3$. Please note, that the vector Laplacian in Eq. (20) is similarly defined as its scalar counterpart and simply acts component-wise.

4.3.4 Energy. The conservation of energy in temperature form can be written [Houze 1994] as

$$\partial_t \theta + (\boldsymbol{u} \cdot \nabla)\theta + \frac{L}{c_p \Pi} C_c = 0\,, \tag{21}$$

in which $C_c = \partial_t q_c + (\boldsymbol{u} \cdot \nabla)\, q_c$ denotes the rate of condensation. In Eq. (21), the latent head is given by $L \approx 2.5$ J/kg, and the ratio $\Pi := T(t_0)/\theta$ of the absolute and the potential temperature (see Appendix B) is written as the Exner function used in atmospheric modeling. It can be considered as a non-dimensionalized pressure.

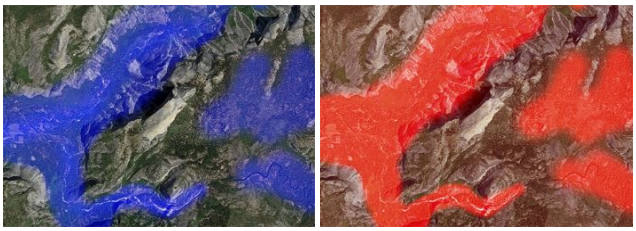


Fig. 7. Visualization of the distribution of vapor (left; blue intensity map) and heat (right; red intensity map) emission used for simulating high fog around Half Dome in Yosemite National Park (see Figure 15, top). The corresponding satellite image is shown in the background. While vapor emission is only present in the valley, a small amount of heat is also emitted at its borders.

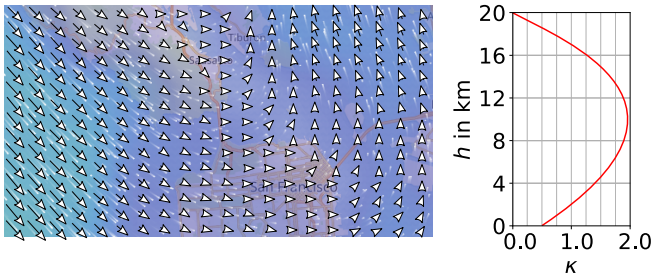


Fig. 8. Illustration of a layered wind field profile (left) extracted by tracking the motion of streamlines from image data provided by an online weather service. The streamlines visualize the wind flow around the San Francisco Bay Area on Thursday, May 21, 2020 at 12noon local time. A scaling function $\kappa : h \mapsto \kappa(h)$ (right) inspired from Limpinsel et al. [2018] is applied to scale the velocities at different altitudes.

4.3.5 Water Continuity. The simplest model for rain, based on Kessler's methodology as illustrated in Figure 6, includes a third transport equation for a rain phase q_r in addition to vapor and cloud, along with source terms that couple the transport equations. Using the material derivative $\mathrm{D}_t \varphi = \partial\varphi/\partial t + \boldsymbol{v} \cdot \nabla\varphi$ [Kundu et al. 2012], we can write

$$\mathrm{D}_t q_v = \quad -C_c + E_c + E_r\,, \tag{22}$$
$$\mathrm{D}_t q_c = \quad C_c - E_c - A_c - K_c\,, \tag{23}$$
$$\mathrm{D}_t q_r = \quad A_c + K_c - E_r\,, \tag{24}$$

with the source terms C_c describing condensation, E_c describing the evaporation of clouds, E_r describing the evaporation of rain, A_c describing autoconversion of raindrops from clouds, and K_c describing the accretion of cloud water due to falling drops. Kessler suggests models in the form $A_c = \alpha_A(q_c - a_T)$ with $a_T = 0.001$ kg/kg, and $K_c = \alpha_K q_c q_r$. Rain drops technically fall and are accelerated, but they reach a constant terminal velocity almost right away (≈ 10 m/s). We employ a constant vertical terminal velocity and the local lateral wind velocity.

5 IMPLEMENTATION

The atmospheric and fluid dynamic models described in the previous section provide the basis for the implementation of our cloud simulation framework. In the following, we provide details for the numerical integration procedure including the setup of the boundary conditions, different terrain properties allowing for variations in vapor and heat emission, as well as external wind fields. Furthermore, we introduce a lightweight parameter set to allow for effortless user control of our interactive simulator, and briefly describe the rendering process.

5.1 Numerical Integration

Our simulator performs the numerical integration of a coupled system of differential equations containing the Navier-Stokes equation (20), the incompressibility equation (19), the first law of thermodynamics (21), and the water continuity equations (22–24).

We discretize a spatial domain $\Omega \subset \mathbb{R}^3$ using a uniform grid scale Δx in which we store the current state of the system which consists of the masses of water vapor q_v, condensed water q_c, and rain q_r next to the potential temperature θ and the velocity field $\boldsymbol{u}$. In most simulations, we employ a rectangular domain Ω but allow for a non-flat terrain defined by a height map in order to include interesting terrains as described in Section 5.1.

An overview of the numerical integration procedure is provided in Algorithm 1 in Appendix C. First, the velocity field is advected followed by the integration of viscosity-related effects by solving the corresponding diffusion process [Stam 1999]. For the advection process, we employ no-slip conditions at the bottom and free-slip conditions at the ceiling. The vertical velocity is set identically to zero at the side boundaries and if an external wind field is specified, the horizontal velocities are computed as the corresponding wind speeds.

While controllable physical damping stabilizes the simulation, the non-controllable and non-physical damping caused by numerical dissipation in these previous steps removes interesting turbulent flow features on different scales, which are essential to obtain realistic cloud patterns. By employing a vorticity confinement force $\boldsymbol{f}_\omega$ as introduced by Steinhoff and Underhill [1994], we inject this energy back into the system. The strength of $\boldsymbol{f}_\omega$ can by controlled by a scale parameter ϵ [Fedkiw et al. 2001]. In the upcoming steps, buoyancy and external net forces (e.g., caused by wind effects or external user input) are computed and integrated numerically. This is followed by the enforcement of incompressibility due to pressure projection by solving the Poisson pressure equation $\nabla^2 p = \Delta t^{-1} \nabla \cdot \boldsymbol{u}$ and subtracting the pressure gradient form the velocity field [Bridson and Müller 2007]. We employ Jacobi pressure projection using 20 iterations on average. Pure Neumann boundary conditions are employed. Finally, θ, q_v, q_c, and q_r, whose temporal evolutions are described by Eq. (21) and Eq. (22–24), are solved by advecting these quantities and updating their mixing ratios. At the top and at the side boundaries, θ is set to the ambient temperature. For q_v, periodic side boundary conditions are used and it is set to zero at the ceiling. The setup of θ on the ground is defined by heat emission as q_v is defined by the emission of water vapor. This is described in Section 5.1. At the whole boundary $\partial\Omega$, the values of q_c and q_r are set identically to zero.

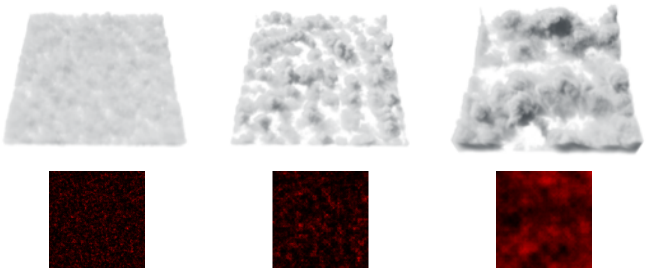


Fig. 9. Cloud formation (top) caused by different noise patterns modeling the terrain (bottom) of the domain Ω. It is sampled from 128×128 noise patterns. The color-maps visualize texture values from 0 (black) to 1 (red). The terrain is completely flat at zero altitude and parameters $E = 7.000$ and $\phi_{\text{rel}} = 26.9\%$ as well as mixture parameters $\gamma_{\text{heat}} \in \{0.120, 0.332, 0.300\}$ respectively $\gamma_{\text{vapor}} \in \{0.267, 0.221, 0.295\}$ are used (from left to right).

5.2 Terrain

To conveniently include a variety of interesting landscapes we make use of a height map $\mathcal{H} : (x, y) \mapsto \mathcal{H}(x, y)$ defining the lower boundary $\partial\Omega_{\text{bottom}} := \{(x, y, \mathcal{H}(x, y))^\mathsf{T} \in \Omega\}$ of the domain Ω.

We allow for temperature variations on $\partial\Omega_{\text{bottom}}$. The temperature θ at a specific position $\boldsymbol{x} = (x, y, h)^\mathsf{T}$ on the ground is computed as the sum of the temperature T_{ISA} (according to ISA, see Appendix A) for a given altitude $h := \mathcal{H}(x, y)$ and the specific heat emitter value E at position $\boldsymbol{x}$, i.e. $\theta(\boldsymbol{x}) = T_{\text{ISA}}(h) + E(\boldsymbol{x})$. We set $E(\boldsymbol{x}) = E$ as a constant global parameter within the whole scene and allow for scaling local heat emission by sampling from a terrain heat emission map $(x, y) \mapsto \text{heat}(x, y) \in [0, 1]$ for all $(x, y, \mathcal{H}(x, y)) \in \partial\Omega_{\text{bottom}}$ as illustrated in Figure 7. The temperature θ at a specific position $\boldsymbol{x}$ is then computed as

$$\theta(\boldsymbol{x}) = T_{\text{ISA}}(h) + E \cdot (\gamma_{\text{heat}} (m \cdot \text{heat}(x, y) - 1) + 1) \,, \qquad (25)$$

in which a mixture parameter $\gamma_{\text{heat}} \in [0, 1]$ is used to scale the influence of the noise pattern. In particular, if $\gamma_{\text{heat}} = 0$, the noise pattern is not influencing the temperature reducing Eq. (25) to $\theta(\boldsymbol{x}) = T_{\text{ISA}}(h) + E$. Instead, if $\gamma_{\text{heat}} = 1$, Eq. (25) is reduced to $\theta(\boldsymbol{x}) = T_{\text{ISA}}(h) + m \cdot \text{heat}(x, y) \cdot E$, i.e. for maximum value of $\text{heat}(x, y) = 1$, the heat emitter value E is scaled by a factor of m. We set $m = 2$ in our simulations. From the temperature θ and the pressure p at altitude h according to ISA, the corresponding static water vapor q_{vs} is computed. Similarly to the mixture parameter γ_{heat} in Eq. (25), an additional mixture parameter $\gamma_{\text{vapor}} \in [0, 1]$ is introduced scaling the influence of another emission map $(x, y) \mapsto \text{vapor}(x, y) \in [0, 1]$ for all $(x, y, \mathcal{H}(x, y)) \in \partial\Omega_{\text{bottom}}$ modeling the distribution of vapor emission on the ground as illustrated in Figure 7. The emitted water vapor q_v at a specific position $\boldsymbol{x}$ is then computed as

$$q_v(\boldsymbol{x}) = \phi_{\text{rel}} \cdot q_{vs}(\theta(\boldsymbol{x}), p_{\text{ISA}}(h)) \cdot \left(\gamma_{\text{vapor}} (m \cdot \text{vapor}(x, y) - 1) + 1\right) \,, \qquad (26)$$

in which the $q_{vs}(\theta, p)$ is estimated using Eq. (16) for the saturation mixing ratio. Eq. (26) also takes the relative humidity ϕ_{rel} as a scaling parameter into account since the approximation (16) presupposes that the air is saturated.

Fig. 10. Variations of different cloud types (top) and corresponding photo comparisons (bottom): cumulus (a, b), cumulus humilis (c, d), stratus (e), stratocumulus (f, g), and cumulus congestus (h).

5.3 Wind Fields

While the general structure of heat and vapor emission maps is mainly designed based on relevant features of the specific terrain as illustrated Figure 7, our framework also supports the use of noise patterns as emission maps. The influence of different noise patterns for heat and vapor emission on cloud formations is illustrated in Figure 9. Moreover, we support the use of Perlin noise [1985] in order to add random perturbations to predefined emission maps. Next to the number of octaves, specifically the noise parameters frequency ν_{Perlin} and its persistence p_{Perlin} can be controlled by the user.

Supporting the use of realistic wind fields is an essential feature for simulating complex weather scenes. In theory, a wind field can be incorporated by defining the external force $\boldsymbol{f}$. However, when approaching different sources for weather data, we found that three-dimensional wind field profiles taking different altitude levels into account are usually very sparse. In fact, mostly two-dimensional layered profiles are available. Also, when employing user-designed wind fields, generating them is more convenient when the task is reduced to specifying layered profiles in two dimensions. Since for most scenes, horizontal wind fields are particularly relevant, we usually employ a single layer that specifies the wind field profile at a given altitude and scale it with a function $\kappa : h \mapsto \kappa(h)$. An example of a wind field layer and a scaling function is shown in Figure 8.

6 RESULTS

In this section, we present a variety of results simulated with our C++/CUDA framework implemented as described in the previous section. Table 1 provides an overview of the different scenes presented throughout this section including relevant parameters. Most scenes can be simulated interactively or even in real-time. The computation times listed in Table 1 are measured on an up-to-date desktop computer running our simulation framework on a NVIDIA® GeForce ® GTX 1080. Double precision floating point arithmetic is used.

To render these scenes, we implemented volume ray casting [Pharr et al. 2016] using OpenGL/GLSL. Rays of light are evaluated as they pass through the cloud volume. The opacity and color for each pixel intersecting the volume is returned and visualized on the screen. This technique is suitable for real-time computing allowing for exploring our simulations interactively within our framework. Unless otherwise stated, the final results shown throughout the paper are rendered offline using the Cycles renderer integrated in Blender.

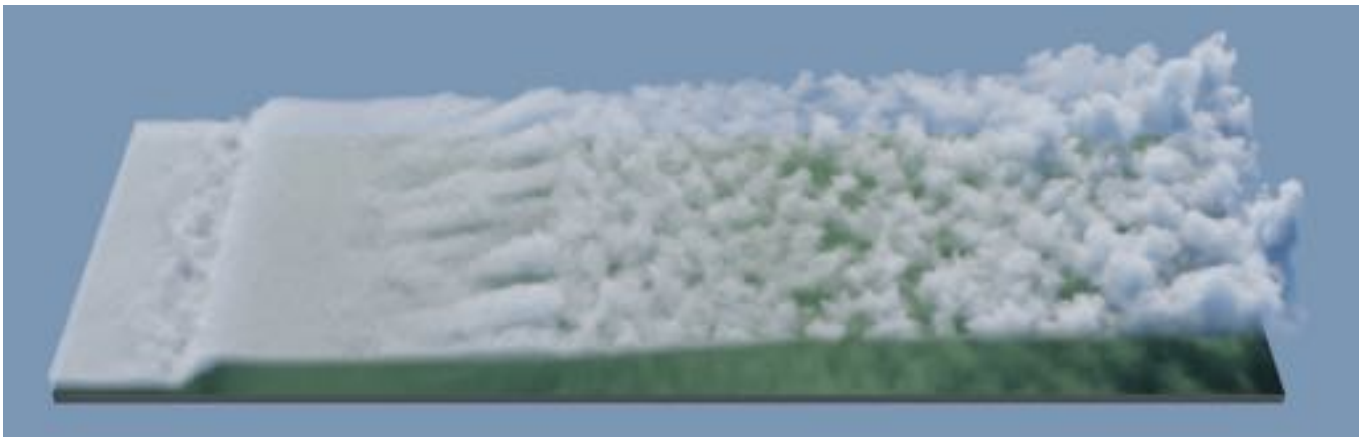


Fig. 11. Visualization of cloud transitions from fog (left) over stratocumulus to cumulus (right). The parameters listed in Table 1 are applied, and to allow for the presentation of transitions in between, we interpolated the parameters linearly from left to right.

Fig. 12. Temporal evolution of a cumulonimbus cloud emerging from cumulus clouds.

Fig. 13. Simulation of the Kelvin-Helmholtz instability [North et al. 2014] generated with our framework.

6.1 Cloud Types and Transitions of Clouds

To support the effortless control of our simulation framework, we introduce a lightweight parameter set (see Appendix D) which contains the heat emitter value E and the scaling parameter γ_{heat}, the relative humidity ϕ_{rel} on the ground, the scaling parameter γ_{vapor}, and the Perlin noise frequency ν_{Perlin} and persistence p_{Perlin}. For the convenient use of the simulator, all other parameters defining the atmospheric background can stay constant, and the aforementioned parameters can be manipulated interactively during the simulation to control the development of cloud formations and transitions.

While γ_{heat} and γ_{vapor} control the heat and the vapor emission variance, E and ϕ_{rel} directly influence the altitude of the upper and lower part of the clouds, respectively. In particular, increasing E leads to an increase of the altitude of the clouds' ceilings, while increasing ϕ_{rel} results in a lower altitude of the clouds' bases (Figure 10, a, b). The pattern of cloud structure can be controlled by changing ν_{Perlin} and p_{Perlin} (Figure 10, c, d). Some examples of photorealistically rendered clouds generated with our framework are shown in Figure 10 including photo comparisons.

We can simulate transitions of different cloud types, e.g., as illustrated in Figure 3 from fog over stratus, stratocumulus and cumulus to cumulonimbus. This is demonstrated in Figures 11 and 12. Starting form an 'empty' scene, fog can be generated by adjusting $E > 0$, $\gamma_{\text{heat}} = \gamma_{\text{vapor}} = 0$, and $\phi_{\text{rel}} = 100\%$. Please note, that noise is not necessarily needed at this point. To transition the fog into stratus, E has to be slightly increased and ϕ_{rel} decreased. Stratocumulus can then be generated by increasing γ_{heat} and γ_{vapor} slightly. Noise should be enabled, and ν_{Perlin} and p_{Perlin} adjusted depended on the desired cloud pattern. To achieve the typical mixture of closed and separated structures, the parameter p_{Perlin} should be explored. To further transition to cumulus clouds, E should be increased and ϕ_{rel} decreased, γ_{heat} and γ_{vapor} should be increased, and ν_{Perlin} should be slightly reduced compared to the stratocumulus setup. By increasing γ_{heat}, the cumulus clouds can then be further transitioned into the cumulonimbus shape. Please note, that the appearance of cumulonimbus clouds is also significantly influenced by the lapse rate Γ and the inversion point z_1 marking the change in the temperature gradient. In particular, the temperature inversion at z_1 acts

as an obstacle for the rising thermal, causing the characteristic flat anvil top of a cumulonimbus to form.

6.2 Complex Scenes

As shown in Figure 15, our framework is capable of simulating complex scenes, such as the characteristic high fog around Half Dome in Yosemite National Park and clouds forming over the Atlantic Ocean moving towards the Irish coast rising up the mountains. Detailed height maps and satellite textures of these regions are used. For the Half Dome scene, heat and vapor emission maps are set up as shown in Figure 7. While vapor emission is only present in the valley, a small amount of heat is also emitted at its outside. A uniform wind field is used to move the fog through the valley.

In the Irish coast scene, we employ different heat emission maps distinguishing between a day and a night scene. Since the sea has a larger heat capacity compared to the land part, during the day, the land warms up more quickly compared to the sea. During the night, the water keeps the heat from the day significantly longer compared to the land. This results in the emerging of a sea breeze during the day and a land breeze during the night. We set up the heat emission maps for the two scenes accordingly while vapor is mostly emitted from the sea part.

6.3 Rain

Figure 16 illustrates the simulation of rain. Rain clouds are formed as a consequence of cold air compressing multiple tiny water droplets together to larger rain drops. The transition of humid air, clouds, and water is expressed by Kessler's scheme [1969] in our method. We illustrate the effect of rain in our simulation with the example of a scene of a cumulonimbus cloud forming over the Alps (Figure 16, left). In Figure 16 (right), we visualize rain as a color map that shows low amounts of rain in blue and high amounts of rain in orange. To better visualize the effect, clouds have been removed from both the cloud base and the cumulonimbus cloud toward the camera. Heavy down pour can be observed at the right side of the cumulonimbus cloud adding to the realism of the scene.

6.4 Kelvin-Helmholtz Instability

The Kelvin-Helmholtz instability [North et al. 2014] can be observed in several cases when a velocity difference across the interface between two fluids is present. In the context of clouds, it can sometimes be observed during sunrise or sunset when the lower part of the clouds cools down and becomes significantly more dense compared to the upper part. As a consequence, the air in the upper part moves faster resulting in the emergence of a pattern comparable to ocean waves when wind is blowing over water. We can reproduce this instability as demonstrated in Figure 13 by applying a wind field characterized by greater velocities in upper simulation grid cells compared to lower grid cells. The heat emitter parameter E is then adjusted to a value that allows cloud condensation to produce a wave-like cloud pattern.

6.5 Cloud Supercells

Supercells form if mesocyclones (i.e. continuously rotating updrafts) are present, usually within clusters of cumulonimbus formations.

Table 1. The table provides an overview of the lightweight parameter set used in the scenes presented in this paper. Moreover, resolution (R) and computation time (T) measured in seconds per frame are listed. A constant time step size of $\Delta t = 1$ min is used. The background parameters defining the atmosphere are set to $\Gamma = -6.5$ K/km and $z_1 = 8$ km.

Fig.	Scene	E	γ_{heat}	ϕ_{rel}	γ_{vapor}	ν_{Perlin}	p_{Perlin}	R	T
11	Fog ↦ Stratus	0.5	0.0	1.00	0.00	-	-	128×48×128	0.04
11	Stratus ↦ Stratocu.	1.6	0.0	0.73	0.00	-	-	128×48×128	0.04
11	Stratocumulus ↦ Cu.	1.6	0.3	0.70	0.00	0.07	0.50	128×48×128	0.04
11	Cumulus ↦ Cumulonib.	2.2	1.5	0.54	0.14	0.15	0.75	128×48×128	0.04
15	Fog around Half Dome	1.1	-	0.86	-	-	-	369×144×288	0.12
15	Irish Coast Day	2.2	1.6	0.54	0.14	0.15	0.75	410×160×320	0.15
15	Irish Coast Night	2.3	1.6	0.54	0.16	0.15	0.75	410×160×320	0.15
16	Rain over the Alps	1.7	0.3	0.65	0.0	0.07	0.50	348×62×256	0.17
13	K.-H. Instability	0.5	0.0	1.00	0.0	-	-	128×32×16	0.02
17	Low P. Supercell	15	0.0	0.70	0.00	-	-	192×96×96	0.12
17	C. Supercell	15	0.0	0.70	0.00	-	-	384×192×192	0.25
17	High P. Supercell	15	0.0	0.80	0.00	-	-	288×48×288	0.19

Fig. 14. Real-time exploration of cloudy weather in the San Francisco Bay Area on May 21, 2020 at 12noon (left) and on October 11, 2019 at 12noon local time. We use in-time weather-data streamed to our framework to simulate realistic cloud formations.

These rotating thunderstorms result in the formation of complex stormscapes. The literature usually considers three different types of supercells [Grant and Van Den Heever 2014]: next to classical supercells, low and high precipitation supercells are studied. To understand their differences, we have to specifically consider the (external) wind relative to the storm itself. If high-speed (> 110 km/h), storm-relative winds are present in the upper layers, this leads to a separation of updraft and downdraft regions resulting in a low precipitation supercell. In contrast, classical supercells typically occur in weather conditions with storm-relative wind speeds of about 70 to 110 km/h. Moreover, since they are devoid of precipitation, their mesoscale features are clearly visible. In environments with storm-relative wind speed below 70 km/h, a small separation of updraft and downdraft regions usually results in high precipitation supercells.

To simulate different types of supercells, we apply a rotationally symmetric layered wind profile modeling the rotating upstream to a cumulonimbus formation. We then add strom-relative winds of varying speeds based on the specific supercell type. Our framework allows for the dynamic simulation of the three types of large-scale cloud supercells as shown in Figure 17. Especially, classical supercells are popular footage among photographers and film makers showing close-up photos and time-lapse videos of their structure and temporal evolution. Inspired from such pieces of art, we present visualisations of our simulations of different supercells in Figure 1.

6.6 Weather Nowcasting

To further showcase the capabilities of a physically accurate cloud simulation model, we coupled our framework with atmospheric

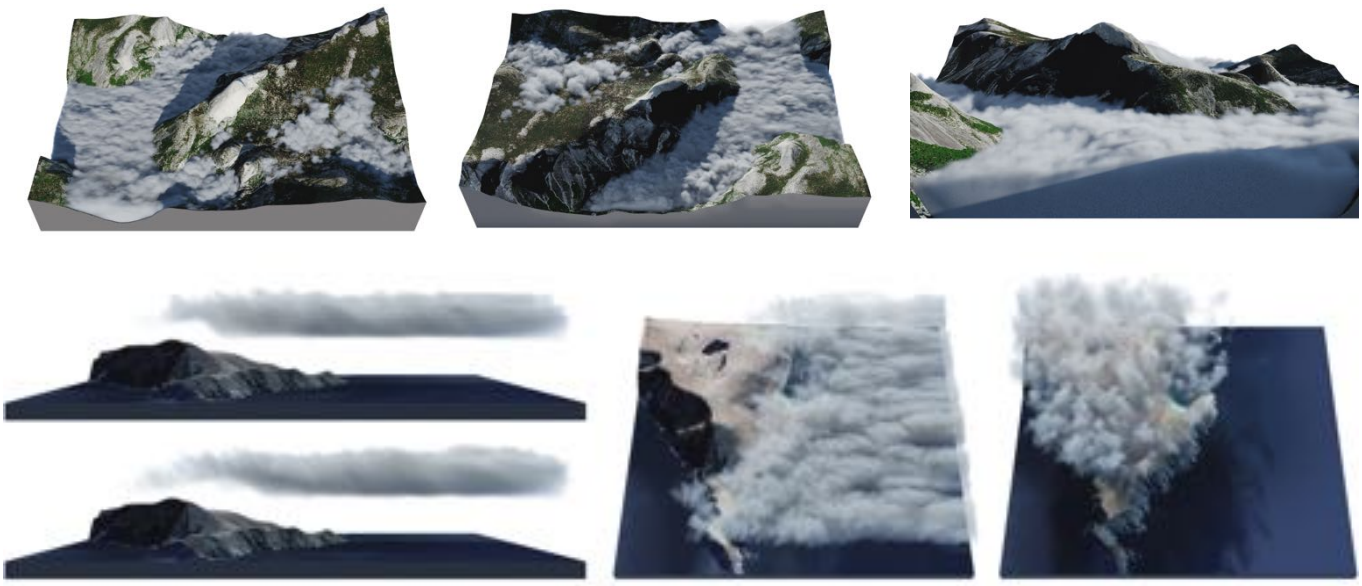


Fig. 15. Simulations of high fog around Half Dome in Yosemite National Park with two top-down (top, left and center), and one in-scene viewpoint (top, right), and clouds forming at the Irish coast (bottom). During the day, a sea breeze (middle) moves the clouds built up over the Atlantic Ocean to the coast rising up the mountains (left). During the night, the opposite effect can usually be observed. Clouds formed over the land (right) are moved to the sea as a result of a land breeze.

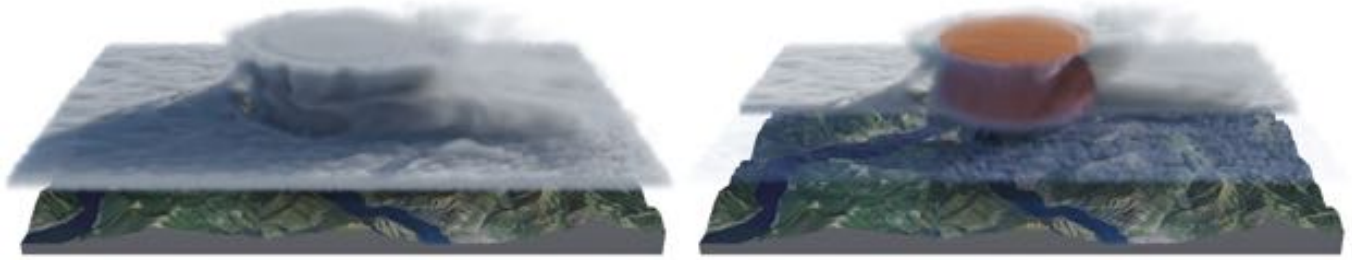


Fig. 16. Heavy rain over the Alps caused by a cumulonimbus cloud (left). The amount of rain is visualized using a color map (right) from blue (low amount) to orange (high amount). To better visualize the effect of rain, clouds have been removed from both the cloud base and the cumulonimbus cloud toward the camera in the right image.

measurements of weather services to simulate cloud formation in real-time enabling weather nowcasting. We manually select a geographic location and query atmospheric conditions via temperature, humidity and wind maps as provided by a weather service.[1] Temperature and humidity maps can directly be used in our framework instead of noise maps (see Section 5.1). A wind map is a dynamic streamline plot visualizing real-time wind velocities projected on satellite imagery in 2D. We use a sequence of wind maps to approximate a wind field for our simulation by tracking the motion of streamlines with optical flow [Itseez 2015]. We then define a wind field by interpolating the generated motion vectors across a regular lattice and use the wind field in our framework to sample wind speeds at different altitudes by scaling the interpolated values with the function κ. This is illustrated in Figure 8. The wind velocities are used as an external wind field as described in Section 5.3.

In Figure 14 we show the result of weather nowcasting for the San Francisco Bay Area on May 21, 2020 at 12pm local time (left) that can be compared to weather conditions on a different date, e.g. October 11, 2019 at 12pm local time (right). To allow for an interactive exploration of the scene in real-time, we use a volume ray casting-based online renderer. Along with real-time simulations for cloud dynamics, weather nowcasting can provide a more immersive experience for exploring urban landscapes. Furthermore, we argue that coupling real-time simulations with in-time observations provides a new way of interacting with real world phenomena, especially when used in virtual and augmented reality applications.

[1] http://www.ventusky.com

7 EVALUATION, DISCUSSION, AND LIMITATIONS

In addition to the qualitative visual assessment in Fig. 10, we provide two quantitative evaluation strategies. The first is the 0D thermodynamics model illustrated in Fig. 5, which can be used to (1) confirm that our CFD simulations do in fact match the theoretical predictions of cloud base and top, and (2) can also be used in an *a priori* exploratory way to determine the set of parameters required to create a cloud with certain characteristics. The second evaluation

Fig. 17. Temporal evolution of different types of cloud supercells. Strong storm-relative winds (> 110 km/h) cause an interruption of the formation of the anvil resulting in a low precipitation supercell (top) while low (< 70 km/h) storm-relative winds result in a high precipitation supercell (bottom). Moderate wind speeds (70 to 110 km/h) lead to the classical supercell (middle).

method for our modeling approach is to show cloud fraction profiles for various cloud morphologies. Cloud fraction is a widely used metric in meteorology and environmental sciences to demarcate different cloud types. As inspiration to model scenes for which we calculated cloud fraction profiles we used reconstructions from Lidar and infrared satellite observations [Cesana et al. 2019], as well as generated data by established simulations in meteorology [Shen et al. 2020]. Cloud fraction is defined as the percentage of grid cells rendered as a cloud ($q_c > 0$) in the whole grid space. We visualize the summed frequencies at different altitudes in Figure 18. There, we show the cloud fraction profiles for exemplary stratus, stratocumulus and cumulus cloud scenes. Stratocumulus have a limited vertical extent [Houze 1994], resulting in a narrower cloud fraction over altitude compared to cumulus or stratus clouds [Cesana et al. 2019, Figure 16]. We simulate this phenomenon by lowering emitter heat values E for the stratocumulus cloud scene. Cumulus clouds occur over a wider range of altitudes compared to other cloud types, which was simulated by increasing the γ_{heat} parameter value. Figure 18 also depicts the temporal evolution of cloud fraction profiles of a rising cumulonimbus cloud which is characterized by the emergence of the characteristic anvil-like top of the cloud at simulation time t_2 [Shen et al. 2020, Figure 3].

7.1 Discussion and Limitations

Our focus was on devising a physically accurate model to enable the realistic simulation of water-based cloud dynamics. This ranges from common cloud types, such as cumulus or stratus, to more complex patterns, such as stormscapes. Stormscapes are weather phenomena that happen over a large range of altitudes, i.e. the whole troposphere. To capture these phenomena, this requires a model to simulate the fluid dynamics continuously over the whole domain, instead of relying on spatial abstractions, such as layers. Moreover, introducing an explicit inversion point allows us to intuitively model the formation of cumulonimbus clouds, which is not possible with existing methods. Simulating stormscapes is especially complicated due to the high wind velocities, underlying cloud dynamics and is addressed in our method by explicitly modeling buoyancy.

Unlike Harris et al. [2003], we generalize the atmospheric temperature profile for cases in which buoyancy vanishes or dramatically reverses. In addition, our formulation of buoyancy does not depend on pressure as our derivation indicates that the impact of the pressure field is negligible on parameters relevant for cloud formation: a layer on top of the given atmosphere merely compresses all underlying layers, and does not noticeably change the resulting buoyancy important for cloud formation and dynamics.

On the other hand, recent work in cloud modeling captures the formation of other cloud types, beyond the scope of this work, by adopting a higher scale of abstraction [Vimont et al. 2020]. The results we show indicate that our method can simulate various plausible cloud morphologies while expressing accurately seminal

physical hypotheses of cloud formation. Furthermore, our simulations of cloud types can be quantitatively compared to data reported in meteorology based on cloud fraction profiles.

While our model is able to capture the various water-based cloud types, we did not conclusively explore their variations, which are commonly described as cloud species. An example thereof is the cumulonimbus capillatus cloud species which is distinguished from the cumulonimbus incus species. The capillatus species exhibits a more rounded top of the cloud instead of the pronounced anvil shape of the incus species. Simulation results for the incus species are shown in Figure 12. Furthermore, our model does currently not allow simulating ice cloud formations, such as cirrus clouds, which are determined by mixing ratios of ice, snow and hail. In principle, the Kessler methodology adopted in our approach could be extended to also account for these additional quantities. The specific mechanisms underlying ice cloud formation are still an active topic of research in the environmental science community. Compared to cloud models used in climatology, e.g. Large Eddy Simulations (LES) [Cesana et al. 2019; Seeley et al. 2019] our method does not represent the feedback of heat transfer between the ground and air. For example, as large clouds rise into the upper troposphere they can shade and subsequently cool the ground surface which can influence cloud dynamics in the lower troposphere.

8 CONCLUSION AND FUTURE WORK

We have presented a novel framework for the physically accurate simulation of clouds based on first-principles. In particular, we proposed novel formulations for computing buoyancy and air pressure that account for variations of atmospheric density and temperature gradients. Using our approach, we can simulate several different cloud types, their transitions, and complex cloud phenomena that we refer to as stormscapes. We introduce heat emitter, relative humidity on the ground, and Perlin noise function parameters as lightweight means for simulating cloud formations, while maintaining interactive rates for rendering. Together this enables the intuitive exploration of large-scale and realistic cloud formations. We have shown the capabilities of our model through numerous experiments of advanced and intricate cloud formations, including rare phenomena such as the Kelvin-Helmholtz instability and different types of supercells. Furthermore, we have coupled our cloud model with underlying terrain and connected it to real-time weather services simulating realistic cloud formations in the now.

A framework for the physically accurate simulation of cloud dynamics opens multiple avenues for future work. For one, it would be interesting to extend our model to simulate clouds that form from ice instead of water. This would allow us to simulate an even wider range of cloud types and their transitions. Furthermore, while our model already simulates rain, extending it to other forms of precipitation, such as snow, sleet, or hail, would allow us to more accurately simulate weather. Moreover, it seems promising to further explore the two-way coupling of terrain and cloud formations or other weather phenomena. While we have shown the interaction of clouds and realistic landscapes, the simulation of various terrain types is likely to result in more realistic simulations of clouds. Finally, it would be interesting to explore the connection of our model to climatology and to further investigate the dynamics of cloud morphologies.

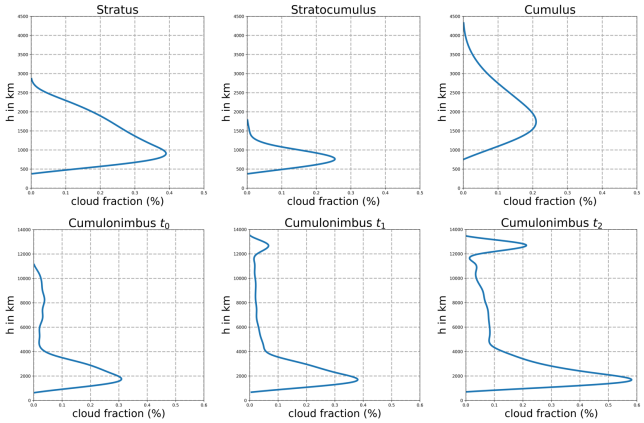


Fig. 18. Fraction profiles of stratus, stratocumulus, and cumulus (top), and of a developing cumulonimbus cloud simulated with our framework at time points t_0, t_1, and t_2 (bottom).

ACKNOWLEDGEMENTS

This work has been supported by KAUST (individual baseline funding). We acknowledge the reviewers' valuable comments.

REFERENCES

J. D. Anderson. 2003. *Modern compressible flow with historical perspective* (3rd ed.). McGraw Hill.
B. Bitterli, S. Ravichandran, T. Müller, M. Wrenninge, J. Novák, S. Marschner, and W. Jarosz. 2018. A Radiative Transfer Framework for Non-Exponential Media. *ACM Trans. Graph.* 37, 6, Article Article 225 (Dec. 2018), 17 pages.
J. F. Blinn. 1982. Light Reflection Functions for Simulation of Clouds and Dusty Surfaces. *SIGGRAPH Comput. Graph.* 16, 3 (1982), 21–29.
A. Bouthors and F. Neyret. 2004. Modeling clouds shape. In *Eurographics 2004 - Short Presentations*, M. Alexa and E. Galin (Eds.). Eurographics Association.
A. Bouthors, F. Neyret, and S. Lefebvre. 2006. Real-Time Realistic Illumination and Shading of Stratiform Clouds. In *Proceedings of the Second Eurographics Conference on Natural Phenomena (NPH'06)*. Eurographics Association, 41–50.
A. Bouthors, F. Neyret, N. Max, E. Bruneton, and C. Crassin. 2008. Interactive Multiple Anisotropic Scattering in Clouds. In *I3D (2008)*. 173–182.
R. Bridson. 2008. *Fluid Simulation for Computer Graphics*. A K Peters, CRC Press.
R. Bridson and M. Müller. 2007. Fluid simulation: SIGGRAPH 2007 course notes. (2007), 1–81.
G. Cesana, A. D. Del Genio, and H. Chepfer. 2019. The Cumulus And Stratocumulus CloudSat-CALIPSO Dataset (CASCCAD). *Earth Syst. Sci. Data* 11 (2019), 1745–1764.
D. Ebert, F. Musgrave, D. Peachey, K. Perlin, and S. Worley. 2002. *Texturing and Modeling: A Procedural Approach* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
Q. Cui, P. Sen, and T. Kim. 2018. Scalable Laplacian Eigenfluids. *ACM Trans. Graph.* 37, 4, Article 87 (2018), 12 pages.
Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, and T. Nishita. 2000. A Simple, Efficient Method for Realistic Animation of Clouds. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. 19–28.
Y. Dobashi, K. Kusumoto, T. Nishita, and T. Yamamoto. 2008. Feedback Control of Cumuliform Cloud Formation Based on Computational Fluid Dynamics. In *ACM SIGGRAPH 2008 Papers (SIGGRAPH '08)*. Association for Computing Machinery, Article Article 94, 8 pages.
Y. Dobashi, Y. Shinzo, and T. Yamamoto. 2010. Modeling of Clouds from a Single Photograph. *CGF* 29, 7 (2010), 2083–2090.
R. P. M. Duarte and A. J. P. Gomes. 2017. Real-Time Simulation of Cumulus Clouds through SkewT/LogP Diagrams. *Comput. Graph.* 67, C (2017), 103–114.
R. Fedkiw, J. Stam, and H. W. Jensen. 2001. Visual Simulation of Smoke. *Proc. of ACM SIGGRAPH* (2001), 15–22.
C. W. Ferreira Barbosa, Y. Dobashi, and T. Yamamoto. 2015. Adaptive Cloud Simulation Using Position Based Fluids. *Comput. Animat. Virtual Worlds* 26, 3–4 (2015), 367–375.
F. Ferstl, R. Ando, C. Wojtan, R. Westermann, and N. Thürey. 2016. Narrow Band FLIP for Liquid Simulations. In *Proceedings of the 37th Annual Conference of the European*

Association for Computer Graphics (EG '16). Eurographics Association, 225–232.
F. Ferstl, R. Westermann, and C. Dick. 2014. Large-Scale Liquid Simulation on Adaptive Hexahedral Grids. *TVCG* 20, 10 (2014), 1405–1417.
I. Garcia-Dorado, D. G. Aliaga, S. Bhalachandran, P. Schmid, and D. Niyogi. 2017. Fast Weather Simulation for Inverse Procedural Design of 3D Urban Models. *ACM Trans. Graph.* 36, 2, Article Article 21 (2017), 19 pages.
G. Y. Gardner. 1985. Visual Simulation of Clouds. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85)*. Association for Computing Machinery, 297–304.
P. Goswami and F. Neyret. 2017. Real-Time Landscape-Size Convective Clouds Simulation and Rendering. In *Proceedings of the 13th Workshop on Virtual Reality Interactions and Physical Simulations (VRIPHYS '17)*. Eurographics Association, 1–8.
L. D. Grant and S. C. Van Den Heever. 2014. Microphysical and dynamical characteristics of low-precipitation and classic supercells. *Journal of the Atmospheric Sciences* 71, 7 (2014), 2604–2624.
J. Guo, Y. Chen, B. Hu, L.-Q. Yan, Y. Guo, and Y. Liu. 2019. Fractional Gaussian Fields for Modeling and Rendering of Spatially-Correlated Media. *ACM Trans. Graph.* 38, 4, Article Article 45 (July 2019), 13 pages.
M. Gupta and S. G. Narasimhan. 2007. Legendre Fluids: A Unified Framework for Analytic Reduced Space Modeling and Rendering of Participating Media. In *Symposium on Computer Animation (SCA '07)*. Eurographics Association, 17–25.
M. J. Harris, W. V. Baxter, T. Scheuermann, and A. Lastra. 2003. Simulation of Cloud Dynamics on Graphics Hardware. In *ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware (HWWS '03)*. Eurographics Association, 92–101.
M. J. Harris and A. Lastra. 2001. Real-Time Cloud Rendering. *CGF* 20, 3 (2001), 76–85.
R. A. Houze. 1994. *Cloud Dynamics*. Elsevier Science.
M. Ihmsen, J. Orthmann, B. Solenthaler, A. Kolb, and M. Teschner. 2014. SPH Fluids in Computer Graphics. In *Eurographics 2014 - State of the Art Reports*.
T. Inglis, M.-L. Eckert, J. Gregson, and N. Thürey. 2017. Primal-Dual Optimization for Fluids. *CGF* 36, 8 (2017), 354–368.
ISO. 1975. *Standard Atmosphere*. Technical Report ISO 2533:1975. International Organization for Standardization.
Itseez. 2015. Open Source Computer Vision Library. https://github.com/itseez/opencv.
A. D. Jones, P. Sen, and T. Kim. 2016. Compressing Fluid Subspaces. In *Symposium on Computer Animation (SCA '16)*. Eurographics Association, 77–84.
J. T. Kajiya and B. P. Von Herzen. 1984. Ray Tracing Volume Densities. *SIGGRAPH Comput. Graph.* 18, 3 (1984), 165–174.
M. Kass and G. Miller. 1990. Rapid, Stable Fluid Dynamics for Computer Graphics. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '90)*. Association for Computing Machinery, 49–57.
E. Kessler. 1969. *On the Distribution and Continuity of Water Substance in Atmospheric Circulations*. American Meteorological Society, Boston, MA, 1–84.
B. Kim, V. C. Azevedo, M. Gross, and B. Solenthaler. 2020. Lagrangian Neural Style Transfer for Fluids. *ACM Transaction on Graphics (SIGGRAPH)* (2020).
B. Kim, V. C. Azevedo, N. Thürey, T. Kim, M. Gross, and B. Solenthaler. 2019. Deep Fluids: A Generative Network for Parameterized Fluid Simulations. *CGF* 38, 2 (2019), 59–70.
D. Koschier, J. Bender, B. Solenthaler, and M. Teschner. 2019. Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. In *Eurographics 2019 - Tutorials*. The Eurographics Association.
P. K. Kundu, I. M. Cohen, and D. R. Dowling. 2012. *Fluid Mechanics*. Elsevier Science.
A. Lagae, S. Lefebvre, R. Cook, T. DeRose, G. Drettakis, D.S. Ebert, J.P. Lewis, K. Perlin, and M. Zwicker. 2010. A Survey of Procedural Noise Functions. *CGF* 29, 8 (2010), 2579–2600.
M. Limpinsel, D. Kuo, and A. Vijh. 2018. SMARTS Modeling of Solar Spectra at Stratospheric Altitude and Influence on Performance of Selected III-V Solar Cells.
F. Losasso, F. Gibou, and R. Fedkiw. 2004. Simulating Water and Smoke with an Octree Data Structure. In *ACM SIGGRAPH 2004 Papers (SIGGRAPH '04)*. Association for Computing Machinery, 457–462.
W. Lu, N. Jin, and R. Fedkiw. 2016. Two-Way Coupling of Fluids to Reduced Deformable Bodies. In *Symposium on Computer Animation (SCA '16)*. Eurographics Association, 67–76.
A. McAdams, E. Sifakis, and J. Teran. 2010. A Parallel Multigrid Poisson Solver for Fluids Simulation on Large Grids. In *Symposium on Computer Animation (SCA '10)*. Eurographics Association, 65–74.
B. Miller, I. Georgiev, and W. Jarosz. 2019. A Null-Scattering Path Integral Formulation of Light Transport. *ACM Trans. Graph.* 38, 4, Article Article 44 (July 2019), 13 pages.
R. Miyazaki, Y. Dobashi, and T. Nishita. 2002. Simulation of Cumuliform Clouds Based on Computational Fluid Dynamics. In *Eurographics*.
R. Miyazaki, S. Yoshida, T. Nishita, and Y. Dobashi. 2001. A Method for Modeling Clouds Based on Atmospheric Fluid Dynamics. In *PG*. IEEE Computer Society, USA, 363.
F. Neyret. 1997. Qualitative Simulation of Convective Cloud Formation and Evolution. In *Computer Animation and Simulation '97*, D. Thalmann and M. van de Panne (Eds.). Springer Vienna, Vienna, 113–124.
T. Nishita, Y. Dobashi, and E. Nakamae. 1996. Display of Clouds Taking into Account Multiple Anisotropic Scattering and Sky Light. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. Association for Computing Machinery, 379–386.
G. R. North, J. A. Pyle, and F. Zhang. 2014. *Encyclopedia of Atmospheric Sciences*. Number v. 1-6. Elsevier Science.
J. Novák, I. Georgiev, J. Hanika, J. Křivánek, and W. Jarosz. 2018. Monte Carlo Methods for Physically Based Volume Rendering. In *ACM SIGGRAPH 2018 Courses (SIGGRAPH '18)*. Article Article 14, 1 pages.
D. Overby, Z. Melek, and J. Keyser. 2002. Interactive physically-based cloud simulation. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.* 469–470.
H.-O. Peitgen and D. Saupe (Eds.). 1988. *The Science of Fractal Images*. Springer-Verlag, Berlin, Heidelberg.
K. Perlin. 1985. An Image Synthesizer. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85)*. Association for Computing Machinery, 287–296.
M. Pharr, W. Jakob, and G. Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation* (3rd ed.). Morgan Kaufmann Publishers Inc.
S. Sato, Y. Dobashi, T. Kim, and T. Nishita. 2018b. Example-Based Turbulence Style Transfer. *ACM Trans. Graph.* 37, 4, Article 84 (2018), 9 pages.
S. Sato, Y. Dobashi, and T. Nishita. 2018a. Editing Fluid Animation Using Flow Interpolation. *ACM Trans. Graph.* 37, 5, Article 173 (2018), 12 pages.
J. Schpok, J. Simons, D. S. Ebert, and C. Hansen. 2003. A Real-Time Cloud Modeling, Rendering, and Animation System. In *Symposium on Computer Animation (SCA '03)*. Eurographics Association, 160–166.
J. T. Seeley, N. Jeevanjee, W. Langhans, and D. M. Romps. 2019. Formation of Tropical Anvil Clouds by Slow Evaporation. *Geophysical Research Letters* 46, 1 (2019), 492–501.
R. Setaluri, M. Aanjaneya, S. Bauer, and E. Sifakis. 2014. SPGrid: A Sparse Paged Grid Structure Applied to Adaptive Smoke Simulation. *ACM Trans. Graph.* 33, 6, Article 205 (2014), 12 pages.
Z. Shen, K. G. Pressel, Z. Tan, and T. Schneider. 2020. Statistically Steady State Large-Eddy Simulations Forced by an Idealized GCM: 1. Forcing Framework and Simulation Characteristics. *Journal of Advances in Modeling Earth Systems* 12, 2 (2020), e2019MS001814.
J. Stam. 1999. Stable Fluids. *Proc. of ACM SIGGRAPH* (1999), 121–128.
J. Steinhoff and D. Underhill. 1994. Modification of the Euler equations for "vorticity confinement": Application to the computation of interacting vortex rings. *Phys. Fluids* 6, 8 (1994), 2738–2744.
M. Stiver, A. Baker, A. Runions, and F. Samavati. 2010. Sketch Based Volumetric Clouds. In *Proceedings of the 10th International Conference on Smart Graphics (SG'10)*. Springer-Verlag, 1–12.
Y. Teng, D. I. W. Levin, and T. Kim. 2016. Eulerian Solid-Fluid Coupling. *ACM Trans. Graph.* 35, 6, Article 200 (Nov. 2016), 8 pages.
A. Treuille, A. Lewis, and Z. Popović. 2006. Model Reduction for Real-Time Fluids. *ACM Trans. Graph.* 25, 3 (2006), 826–834.
K. Um, X. Hu, and N. Thürey. 2018. Liquid Splash Modeling with Neural Networks. *CGF* 37, 8 (2018), 171–182.
B. Ummenhofer, L. Prantl, N. Thürey, and V. Koltun. 2020. Lagrangian Fluid Simulation with Continuous Convolutions. In *ICLR*.
U. Vimont, J. Gain, M. Lastic, G. Cordonnier, B. Abiodun, and M.-C. Cani. 2020. Interactive Meso-scale Simulation of Skyscapes. *Eurographics* (2020).
R. F. Voss. 1988. *Fractals in Nature: From Characterization to Simulation*. Springer-Verlag, Berlin, Heidelberg. 21–70 pages.
A. Webanck, Y. Cortial, E. Guérin, and E. Galin. 2018. Procedural Cloudscapes. *CGF* 37, 2 (2018), 431–442.
S. Wiewel, M. Becher, and N. Thürey. 2019. Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow. *CGF* 38, 2 (2019), 71–82.
J. Wither, A. Bouthors, and M.-P. Cani. 2008. Rapid Sketch Modeling of Clouds. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling (SBM'08)*. Eurographics Association, 113–118.
Y. Xie, E. Franz, M. Chu, and N. Thürey. 2018. TempoGAN: A Temporally Coherent, Volumetric GAN for Super-Resolution Fluid Flow. *ACM Trans. Graph.* 37, 4, Article 95 (2018), 15 pages.
M. K. Yau and R. R. Rogers. 1996. *A Short Course in Cloud Physics*. Elsevier Science.
C.-M. Yu and C.-M. Wang. 2011. An Effective Framework for Cloud Modeling, Rendering, and Morphing. *J. Inf. Sci. Eng.* 27 (05 2011), 891–913.
C. Yuan, X. Liang, S. Hao, Y. Qi, and Q. Zhao. 2014. Modelling Cumulus Cloud Shape from a Single Image. *CGF* 33, 6 (2014), 288–297.
J. Zehnder, R. Narain, and B. Thomaszewski. 2018. An Advection-Reflection Solver for Detail-Preserving Fluid Simulation. *ACM Trans. Graph.* 37, 4, Article 85 (2018), 8 pages.
X. Zhang, M. Li, and R. Bridson. 2016. Resolving Fluid Boundary Layers with Particle Strength Exchange and Weak Adaptivity. *ACM Trans. Graph.* 35, 4, Article 76 (July 2016), 8 pages.
Y. Zhu and R. Bridson. 2005. Animating Sand as a Fluid. *ACM Trans. Graph.* 24, 3 (2005), 965–972.

ALGORITHM 1: Numerical time integration procedure.

Input:

$q_v \leftarrow (q_v(\boldsymbol{x}, t_0))_{\boldsymbol{x}\in\Omega}$, $q_c \leftarrow (q_c(\boldsymbol{x}, t_0))_{\boldsymbol{x}\in\Omega}$, $q_r \leftarrow (q_r(\boldsymbol{x}, t_0))_{\boldsymbol{x}\in\Omega}$
$\theta \leftarrow (\theta(\boldsymbol{x}, t_0))_{\boldsymbol{x}\in\Omega}$, $\boldsymbol{u} \leftarrow (\boldsymbol{u}(\boldsymbol{x}, t_0))_{\boldsymbol{x}\in\Omega}$, $t \leftarrow t_0$

Procedure:

$\boldsymbol{u} \leftarrow$ advect($\boldsymbol{u}$, Δt) //advect velocity field
$\boldsymbol{u} \leftarrow$ diffuse($\boldsymbol{u}$, ν, Δt) //include viscosity as diffusion process
$\boldsymbol{\omega} \leftarrow \nabla \times \boldsymbol{u}$ //compute vorticity field
$\boldsymbol{f}_\omega \leftarrow \epsilon \cdot \Delta x \cdot (\text{normalize}(\nabla \|\boldsymbol{\omega}\|) \times \boldsymbol{\omega})$ //vorticity confinement
$\boldsymbol{u} \leftarrow \boldsymbol{u} + \boldsymbol{f}_\omega \cdot \Delta t$ //apply vorticity confinement force
$\boldsymbol{b} \leftarrow (0, 0, B)^\mathsf{T}$ //compute buoyancy according to Eq. (15)
$\boldsymbol{f} \leftarrow$ getExternalForce(t) //compute external net force
$\boldsymbol{u} \leftarrow \boldsymbol{u} + (\boldsymbol{b} + \boldsymbol{f}) \cdot \Delta t$ //apply buoyancy and external force
$\nabla p \leftarrow$ project($\nabla \cdot \boldsymbol{u}$, Δt) //pressure projection
$\boldsymbol{u} \leftarrow \boldsymbol{u} - \nabla p \cdot \Delta t$ //pressure update
$[\theta, q_v, q_c, q_r] \leftarrow$ advect(θ, q_v, q_c, q_r, $\boldsymbol{u}$, Δt) //advect quantities
$q_{vs} \leftarrow$ getSaturationRatio(getAbsoluteTemperature(θ), p_∞) //see Eq. (16) and Eq. (28)
$q_v \leftarrow q_v + \min(q_{vs} - q_v, q_c) + E_r$
$q_c \leftarrow q_c - \min(q_{vs} - q_v, q_c) - A_c - K_c$
$q_r \leftarrow q_r + A_c + K_c - E_r$
$X_v \leftarrow$ getMoleFraction(q_v) //see Eq. (9)
$\mathcal{M}_{\text{th}} \leftarrow$ getAverageMolarMass(X_v) //see Eq. (7)
$\gamma_{\text{th}} \leftarrow$ getIsentropicExponent($X_v \mathcal{M}_\text{W} / \mathcal{M}_{\text{th}}$) //see Eq. (11)
$c_p^{\text{th}} \leftarrow$ getHeatCapacity(γ_{th}, $\mathcal{M}_{\text{th}}$) //see Eq. (18)
$\theta \leftarrow \theta + L/c_p^{\text{th}} \cdot$ getMoleFraction($-\min(q_{vs} - q_v, q_c)$) //see Eq. (9)
$t_1 \leftarrow t_0 + \Delta t$

Output:

$q_v \leftarrow (q_v(\boldsymbol{x}, t_1))_{\boldsymbol{x}\in\Omega}$, $q_c \leftarrow (q_c(\boldsymbol{x}, t_1))_{\boldsymbol{x}\in\Omega}$, $q_r \leftarrow (q_r(\boldsymbol{x}, t_1))_{\boldsymbol{x}\in\Omega}$
$\theta \leftarrow (\theta(\boldsymbol{x}, t_1))_{\boldsymbol{x}\in\Omega}$, $\boldsymbol{u} \leftarrow (\boldsymbol{u}(\boldsymbol{x}, t_1))_{\boldsymbol{x}\in\Omega}$, $t \leftarrow t_1$

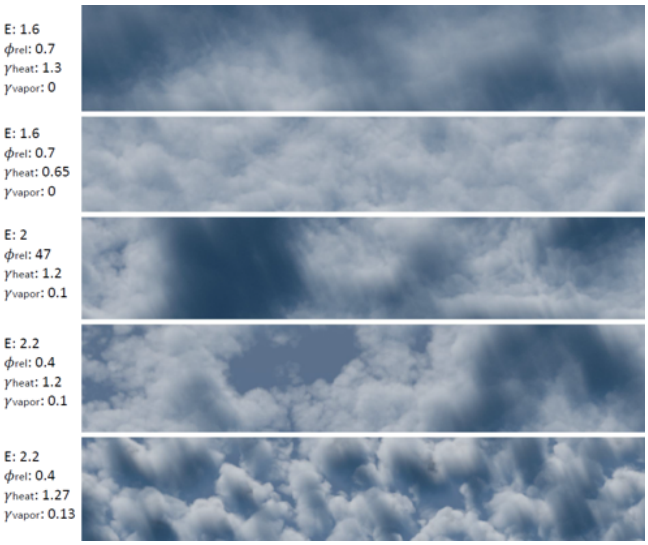


Fig. 20. Several examples of clouds and the parameters used for the emission of heat and vapor on the ground.

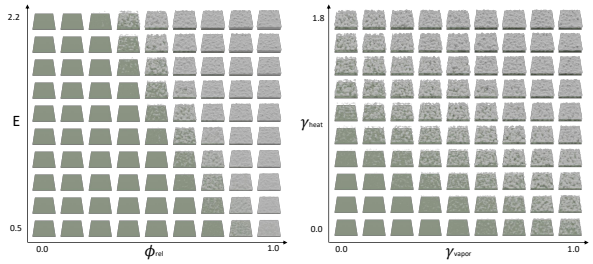


Fig. 21. Parameter space exploration varying the heat emitter value E and relative humidity ϕ_{rel} (left), and scaling parameters γ_{heat} and γ_{vapor} (right). For the non-varying values, parameter settings for cumulus clouds are used: $E = 2.2$, $\phi_{\text{rel}} = 0.4$, $\gamma_{\text{heat}} = 1.7$, and $\gamma_{\text{vapor}} = 0.14$.

A STANDARD ATMOSPHERE

Generally, atmospheric data are evaluated using the 'International Standard Atmosphere' [ISO 1975] which is abbreviated as ISA. The temperature and pressure profile from 0 to 40 km altitude are plotted in Figure 19. Within the troposphere, i.e. below 11 km, the temperature reduces by 6.5 K per 1 km altitude, the pressure approximately halves for every 5 km gained:

$$T_\infty(h) = T_{\infty,0} - 0.0065h\,, \tag{27}$$

$$p_\infty(h) = p_{\infty,0}\left(1 - 0.0065\frac{h}{T_{\infty,0}}\right)^{5.2561}. \tag{28}$$

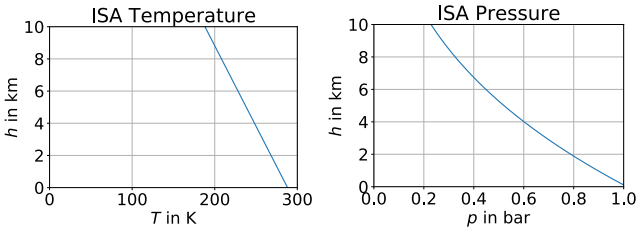


Fig. 19. Temperature and pressure in the troposphere taken from the ISA.

B POTENTIAL TEMPERATURE

Adiabatic changes of temperature and air pressure can be taken into account within the concept of potential temperatures [Yau and Rogers 1996]. Assume, an air parcel is moved adiabatically (i.e. without a change of heat) from temperature $T(t_0)$ and pressure $p(t_0)$ to pressure $p(t_1)$. The final temperature $\theta := T(t_1)$ is given by the potential temperature $\theta = (p(t_1)/p(t_0))^\kappa T(t_0)$ in which $\kappa = R_d/c_p$ is the ratio of the gas constant $R_d = 287\,\text{J/(kg K)}$ for dry air and the specific heat capacity c_p.

C TIME INTEGRATION

The integration procedure is summarized in Algorithm 1.

D PARAMETER SPACE EXPLORATION

A parameter space exploration is presented in Figures 20 and 21.