Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Adaptive volcanic plume modeling using Discontinuous Galerkin Methods

Michel Bänsch

A thesis submitted to the

Faculty of Mathematics, Informatics and Natural Sciences
Department of Mathematics of Universität Hamburg

in partial fulfillment of the requirements for the degree of

Dr. rer. nat

Hamburg 2022

University:	Universität Hamburg
Faculty:	Faculty of Mathematics, Informatics and Natural Sciences
Department:	Department of Mathematics
First supervisor and reviewer:	Prof. Dr. Jörn Behrens Department of Mathematics Numerical Methods in Geosciences
Second supervisor and reviewer:	Prof. Dr. Matthias Hort Department of Earth System Sciences Institute of Geophysics
Date of disputation:	14.04.2023

The sides of this thesis present parts of an adaptive simulation performed for one of the volcanic jet test cases and can be used as a flip book.

Abstract

Just as the recent Tonga eruption in January 2022 showed, explosive volcanic eruptions can emit large ash clouds that reach high into the atmosphere, which in turn can impact atmospheric behavior over a range of spatial and temporal scales. Since these atmospheric processes occur on many scales, it is of interest to have numerical schemes that can resolve physical processes over all these scales without the need to rely solely on high performance computing.
One way to face this obstacle is to utilize adaptive mesh refinement techniques, which allow to control the (spatial) resolutions so that high spatial resolution is ensured in regions of high interest, while areas of low interest also have low spatial resolution.
Such adaptive meshing tools have not been used in models which are used to model the fluid dynamics of volcanic plumes, yet.

The approach within this thesis is to use a nodal Discontinuous Galerkin Method that is able to model volcanic jets and plumes by discretizing equations that are based on the Euler equations.
While this thesis is just a conceptual study on how much computational cost can be saved for 2D volcanic plume models, the results from the numerical scheme show that at least 42% of CPU time can be saved (compared to uniform runs) if adaptivity is used correctly, while still maintaining results that resemble the uniform runs, qualitatively.

Zusammenfassung

Wie die Tonga-Eruption im Januar 2022 gezeigt hat, können explosive Vulkaneruptionen große Aschewolken emittieren, die auch in hohe Bereiche der Atmosphäre gelangen können. Wenn Vulkanasche in diese Bereiche gelangt kann dies das atmosphärische Verhalten über eine Bandbreite an Raum-, sowie Zeitskalen, beeinflussen. Da atmosphärische Prozesse über viele verschiedene Skalen hinweg auftreten, sind numerische Verfahren, welche all diese Skalen auflösen können ohne nur auf Hochleistungsrechner zurückzugreifen von Interesse.
Eine Möglichkeit all diese Skalen in einem Verfahren aufzulösen ist die Verwendung von adaptiver Gitterverfeinerung, welche die (räumliche) Auflösung steuert und so für hohe räumliche Auflösung in Gebieten von hoher Relevanz sorgt, während Gebiete, welcher weniger von Interesse sind auch weniger hoch aufgelöst sind.
Verfahren, die adaptive Gitter verwenden sind bis dato noch nicht Modellen für vulkanische Plumes, die die Fluiddynamik simulieren, zum Einsatz gekommen.

Der Ansatz, welcher in dieser Dissertation verwendet wird, greift auf eine nodale Diskontinuierliche Galerkin-Methode zurück, die vulkanische Jets und Plumes mit Hilfe eines Modells auf Basis der Euler-Gleichungen simulieren kann.
Obwohl diese Arbeit nur eine konzeptionelle Studie zur Analyse der Rechenzeitersparnis von 2D vulkanischen Plume-Modellen darstellt, ergeben die Resultate, dass mindestens 42% der Rechenzeit (im Vergleich zu uniformen Simulationen) eingespart werden können wenn adaptive Gitterverfahren korrekt benutzt werden können sodass diese qualitativ den uniformen Ergebnissen ähneln.

Acknowledgements

The process of a PhD and writing such a thesis is, of course, no work that is solely done by oneself especially, since at times, I thought that I would not be able to finish this journey and contemplated quitting it all. Consequently I would like to express my deepest gratitude to so many people how helped shape me, guide me, inspire me and support me on my journey. While many people mentioned here may not read that I acknowledged them, I still want to include them.

To Jörn Behrens, my first supervisor, superior and panel member: thank you so much for all the patience you had for me on both professional, as well as personal level. If it were not for your support, I think I would not have finished this thesis.
To Matthias Hort, my second supervisor and panel member: thank you for pushing me to start researching volcanoes in the way that I did from the start of my studies, the different perspective you gave me within the panel meetings and the data that you provided for this thesis.
To Armin Iske, my panel chair: thank you for the thorough focus on details and almost relentlessness that you brought with you as part of the panel which provided me with much vigor.

Thank you to Stefan Vater and Konrad Simon for the ideas and support while developing the stabilization method or implementing the respective approaches - be that within StormFlash or deal.II.
Thank you to the *NumGeo* group and all the lovely people (namely and if not already mentioned: Anusha Sunkisala, Claudine von Hallern, Ezra Rozier, Heena Patel, Maša Avakumović, Mouhanned Gabsi and Yumeng Chen) I was able to meet, with lots of fruitful discussions about work or personal life being just one small part of the life within this group.
The same applies to all the other members of the mathematics department who are too numerous to name them all. I feel like I was lucky to interact with so many colleagues due to my teaching duties and for example while being co-examiner for exams. Of course I also want to express my gratitude towards the University of Hamburg for employing me for the 4.5 years of my PhD position.

Furthermore, I want to specifically extend my gratitude to Ingenuin Gasser, Mathias Schacht and Peywand Kiani who agreed to letting me suggest them as for my doctoral committee.

I want to thank Vit Dolejší and Vadym Aizinger for providing literature, results and answering questions that helped during the development of the stabilization scheme.

Thank you to Martin Kronbichler for all the work on the deal.II example that was used throughout the thesis and providing answers to my questions regarding details of that code.

Obviously, I also want to thank my family for all the support they have given me throughout the years; to you, Arne and Shima who always give me new perspectives, to you, Mom, for being there for me to listen to all my ramblings, to you, Dad, for always helping me if I get stuck and of course for proof-reading and giving so many good suggestions on how to improve parts of my thesis.

In particular I want to thank E. Grobe who started a rather large journey of self-discovery for me.
If it were not for all the input I got from the sessions with Dirk Starcke, I would probably not be where I am today, so I want to express my deepest gratitude for all the support you have given me throughout the last 2.5 years. Thank you for always given me a healthy perspective and accompanying me through this journey that is/was my (PhD) life.
Thank you to Marie: your coaching proved to be invaluable to me and you helped me build up my self-worth much quicker.

Of course, these acknowledgements would be incomplete without mentioning all the other very important people who are or were part of my life and help me keep my sanity:

Thank you, Toni, for showing me what a loving friendship looks like! My life would be so much emptier if I did not have you around and since I met you I have been learning so much with all the helped and compassion that you provided.

Thank you, Kim, for going through all the ups and downs we can face within a competitive sport and tackling all that life throws at us together.

Thank you, Christian, for mirroring me in so many ways that it can sometimes be hard do bear, all the incredibly important discussion about our personalities and your compassion.

Thank you, Suza, for throwing me into a new world, all the insights we discuss, your sensibility and your empathy.

Thank you, Anna-Sophie, for all the special moments that we share and you showing me how much you value us thinking alike in so many ways.

Thank you, Yannic, for teaching me that friendship can still be strong even though we rarely see each other and proving that you have my back in times of need.

Thank you, Maca, for being such a huge moral support, providing a different take on our similar struggles and often times being a voice of reason.

Anya, Faizel, Isy, Kimmi, Lia, Linus, Maike, Melli, Nina, Paul, Petzi, Saif, Tomke and Vera, thank you all for the (oh so many) inspirations you have given me throughout the years, all the incredible learning experiences that you provided me with, trust you placed in me, all the appreciation for me as a human being and teaching me what kind of people I want in my life - those people being the ones who are honest, treat me with respect, goodwill, show appreciation and who push me to new heights.

Thank you, Moni for all the interesting talks we had about everything science-related and the new ideas or perspectives you provided.

Contents

1 Introduction

Throughout history, there have been many large volcanic eruptions which also have had drastic impacts on humanity, such as the latest Toba eruption roughly 74,000 years ago and influenced civilised life on planet Earth in one way or another, such as the Mt. Pinatubo (1991) or Eyjafjallajökull (2010) eruptions (Oppenheimer, 2011). In addition, just at the start of this year, the Tonga eruption has emitted the highest volcanic plume ever recorded (Proud et al., 2022). Consequently, with no shortage of unique volcanic events, the fascination about volcanoes is nothing new for mankind.

In addition to the fascination inspired by volcanoes, the fertile ground in volcanic regions has been a reason why people choose to live close to volcanoes despite the dangers. Especially in Southeast Asia and Central America people tend to live close to regions where volcanism might pose a threat and it is estimated that around 14% of the world population (as of 2015) live in a 100 km radius around a Holocene volcano (see Freire et al., 2019a; Freire et al., 2019b). Today, as the Earth's population keeps increasing, with just over 8 billion people (worldometers, 2022) living on Earth, the amount of people living close to volcanoes will probably keep rising.
Since the probability that people will live close to volcanoes will increase, this will certainly result in an increase of people being potentially influenced by the dangers volcanism poses.
While certain volcano types - such as shield volcanoes - prove to be more of a localized risk and threat to e.g. infrastructure instead of people themselves, more explosive volcanoes can influence people's lives much more drastically. While explosive volcanism is not a topic too prevalent in Western media, the 2010 Eyjafjallajökull eruption has shown that a spontaneous explosive eruption with airborne volcanic ash can disrupt infrastructure and impact (Western) society and economy as well, despite the eruption being only medium sized (VEI 4, Smithsonian Institution, 2022).

Consequences of the 2010 Eyjafjallajökull eruption entail about 100,000 flights being canceled due to a no-fly zone that was established (The Telegraph, 2011; BBC, 2010), a cost of 130 million GBP per day (Guardian, 2010) and an estimated total loss to airlines in the range of 1.7 billion USD (The Telegraph, 2011; BBC, 2010).
This recent example is of course just one of the few, with it being very prominent in the Western world - both in the scientific and journalistic media - for the deep impact such an eruption can have.
Risk assessments can be improved by thorough monitoring and observation of volcanoes, especially if those are in the vicinity of densely populated areas, which is done in part by volcano advisory centers (e.g. Zidikheri et al., 2017a; Zidikheri et al., 2017b). Additionally, there are also satellites used for tracking the development of volcanic clouds that

are erupted into the atmosphere. Ideally, this monitoring, along with precise and accurate forecasting tools can lead to a mitigation of injury to people and possibly prevent damages. The modeling of volcanic clouds bears resemblance to the modeling of climate and weather phenomena which is an essential tools for either everyday life (weather forecasts) or for example within the aviation industry. Usually, these models come with a high computational cost, however.

As was shown in Müller et al. (2013), the computational cost for atmospheric models can be reduced by up to 50% without losing important features if adaptive mesh refinement techniques are utilized. With these techniques, the model resolution can be improved in regions of for example high turbulence and coarsened where very little dynamics occur. Since models for simulating volcanic eruption have a lot in common with climate and weather models, it stands to reason that adaptive mesh refinement could also be an effective tool in a volcanic setup where the resolution of the model could be improved in the presence of volcanic ash, for example.

1.1 Review of the most prominent 3D plume models

Within the community that models volcanic eruption columns, it is of high interest to accurately model the physical behavior and development of the volcanic ash clouds. Models used for this purpose are called plume models. For the case of 3D volcanic eruptive columns, four prominent algorithms have been developed:

- ASHEE (Ash Equilibrium Eulerian model; Cerminara et al., 2016a) which uses a so called dusty (Marble, 1970) or pseudo gas (Cerminara et al., 2016b) model, where a single ash-gas mixture phase is used to model the eruptive column which is treated as a gas.
- ATHAM (Active Tracer High-resolution Atmospheric Model; Herzog et al., 2003) which is also based on pseudo gas approach.
- PDAC (Pyroclastic Dispersal Analysis Code; Neri et al., 2003; Ongaro et al., 2007) which is the only of the four algorithms that relies on a multi-phase approach and considers different (gaseous and ash) phases.
- SK-3D (Suzuki et al., 2005; Suzuki and Koyaguchi, 2009; Suzuki and Koyaguchi, 2013; Suzuki and Koyaguchi, 2015) which, again, is a pseudo gas model.

Throughout the years, additional codes for modeling volcanic plumes have been developed, these four 3D codes are the most prominent ones, however, which resulted in them being compared in the International Association of Volcanology and Chemistry of Earth's Interior (IAVCEI) Commission on Tephra Hazard modeling intercomparison study (for the 3D model comparison see Suzuki et al., 2016). For a more detailed description of each approach, either the separate papers within the IAVCEI intercomparison study or the literature that was provided above can be consulted.

Each plume model uses a different numerical approach. Regarding the spatial discretization, ASHEE and PDAC rely on some use of the Finite Volume Method, while ATHAM and SK-3D are based on Finite Difference Method schemes to solve the respective partial differential equation sets that arise. Except for SK-3D, all plume models are based on the Navier-Stokes equations, while SK-3D uses the Euler equations as basis. Regarding the accuracy, ASHEE and ATHAM are 2^{nd} order accurate while PDAC and SK-3D are 3^{rd} order accurate.
The IAVCEI intercomparison study (Suzuki et al., 2016) also summarizes the numerical approaches fot the respective algorithms:

ASHEE:
ASHEE uses the fluid dynamics open-source code OpenFOAM (`https://openfoam.org`) as a basis. For the time stepping scheme, a Crank-Nicholson time-integration (Crank and Nicolson, 1996) is utilized (in addition to an unlimited, centered linear scheme). The PISO-PIMPLE (Pressure Implicit with Splitting of Operators; Issa, 1986) scheme which uses a semi-implicit solution, which is based on a pressure correction algorith, ensures that the system of discretized partial differential equations can be resolved for low, as well as high Mach numbers.

ATHAM:
ATHAM relies on transport equations, which are formulated in flux form. This results in a mass and momentum conservation. To circumvent over- or undershoots within the solution, a correction term is added to the solution similar to Smolarkiewicz (1984). ATHAM also uses a Crank-Nicholson time-integration; general Crank-Nicholson scheme with a forward weight of 0.25.

PDAC:
Using the limited MUSCL (Monotone Upstream-centred Scheme for Conservation Laws) reconstruction (Leer, 1977) as basis, PDAC solved the partial differential equations by a semi-implicit Finite Volume discretization scheme in conjunction with a pressure-based iterative non-linear solver that was designed for compressible (multiphase) flows.

SK-3D:
SK-3D also utilizes the MUSCL interpolation for spatial integration (Leer, 1977) in addition to the Roe scheme (Roe, 1981), while the time splitting method is used as time stepping scheme.

All algorithms allow for the modeling of water (vapor) and ash particles within the eruptive column which is ejected from the vent into an atmosphere comprised of air.
The plume models that are based on the Navier-Stokes equations all use some form of

Table 1: Maximum and minimum (vertical) grid spacing for the IAVCEI's intercomparison study strong plume case. D_0 denotes the vent diameter for each algorithm, respectively. Additionally, the plume heights and the (rough) grid spacing at that height are provided.

	ASHEE	ATHAM	PDAC	SK-3D
D_0	1408 m	1417 m	1410 m	1400 m
Δz_{min}	$\frac{D_0}{32}$ = 44 m	$\frac{D_0}{13}$ = 109 m	$\frac{D_0}{30}$ = 47 m	$\frac{D_0}{40}$ = 35 m
Δz_{max}	300 m	4 km	1 km	300 m
Plume height H	36.7 km	33.4 km	42.5 km	39.9 km
Δz at plume height	150 m	800 m	500 m	300 m

Large Eddy Simulation (LES) approach (Smagorinsky, 1963) which allow to take sub-grid turbulence mixing effects into account.

All algorithms use non-uniform grids for their calculations. These grids differ in resolution and size. Apart from ASHEE (which uses a cylindrical, non-orthogonal mesh), all codes use Cartesian meshes. The algorithms are usually set up in such a way that the highest resolution and minimal grid spacing is determined by the volcanic vent radius/diameter. Using the strong plume test case from the intercomparison study as an example, it becomes evident that the grid sizes for each code vary a lot compared to the other algorithms. **Table 1** shows the respective vent diameters, as well as maximum and minimum grid spacing. Appendix A of Suzuki et al. (2016) shows how the grid spacing varies with increasing height - for all algorithms, the spacing decreases (and consequently the resolution) with height. For ASHEE, the resolution decreases with a constant factor which is kept close to one. For ATHAM, the resolution decreases only until the lowest resolution/maximum grid spacing is reached which then remains constant. SK-3D and PDAC behave similarly, only that for PDAC, the resolution remains constant until a certain height before it decreases. Within SK-3D, the resolution decreases by 1.02 with each cell in either horizontal or vertical direction.

1.2 Motivation

As **Table 1** shows, the pre-existing, prominent plume models face the problem that the spatial (vertical) resolution gets quite low with increasing height. If the plume heights for the strong plume case from Suzuki et al. (2016) are analyzed, it can be seen that the heights range from at least (roughly) 33 km for ATHAM to a maximum heights of 42.5 km for PDAC. Comparing that with the Appendix figure for the vertical grid size at those heights, it can be observed that the resolutions are rather coarse with ASHEE having the finest (vertical) resolution of roughly 150 m, which would still allow for estimating the plume height to one significant figure while the other spatial resolutions at the respective plume heights raise the question whether those estimates are significant.
Of course, it is still a minor remark and while the plume height does not indicate the layer of neutral buoyancy, having an algorithm that is equipped with a high spatial resolution where the volcanic cloud is present is desirable.

One way to circumvent this resolution problem is to use adaptive mesh refinement techniques that refine (and coarsen) the grid where additional resolution is necessary (while coarsening the grid where the resolution is not needed). This would, in theory, allow to better capture plume heights and layers of neutral buoyancy/umbrella regions. Additionally, while being a minor aspect, using adaptive mesh refinement techniques can also be useful when it comes to file storage since the output produced by algorithms with refined meshes require less storage.

1.3 Goal and structure of the thesis

This thesis builds upon the work done in Bänsch (2017) and follows a very similar outline while improving upon the specific implementations regarding the volcanic setups. Additionally, **Section 6.6** presents a stabilization scheme which is based on both the work done in Bänsch (2017), as well as Bänsch et al. (2022). As of the date of the submission of the thesis, the paper (Bänsch et al., 2022) has not been accepted, yet. For this paper, the stabilization scheme was developed by Stefan Vater and me, the tests were designed by Jörn Behrens and me while conducting the experiments and writing the draft were done just by me whereas everyone contributed to parts of the submitted text.

Within the scope of this thesis, the idea is to implement an algorithm that is able to simulate 2D volcanic plumes, which will utilize adaptive mesh refinement techniques. Additionally, instead of a Finite Difference or Finite Volume Method basis, this algorithm will use a Discontinuous Galerkin Method basis.
It should be stressed that this thesis provides mostly a conceptual approach to analyze the question whether adaptive mesh refinement and the Discontinuous Galerkin Method are viable options for plume models.

Consequently, for this approach, the following research questions arise:

- Can a plume model be implemented using the Discontinuous Galerkin Method?
- Are adaptive grids a viable option for volcanic modeling?
- How much CPU time can be saved by using adaptive mesh refinement?

The thesis will be structured in the following way: After this introduction, the theoretical backgrounds will be presented starting with the underlying equation sets for fluid dynamics in **Section 2**, the spatial discretization alongside boundary and initial conditions, a brief part about slope limiters and grid adaptivity in **Section 3**, the time stepping or time-integration schemes in **Section 4**, an overview over the numerical bases that will be used throughout the thesis - StormFlash and deal.II with verification, as well as validation - in **Section 5**, different stabilization methods for the numerically solved Euler equations with gravity in **Section 6**, the results for these stabilization methods in **Section 7**, a detailed depiction on the approach on the plume model with background, equation sets, different test cases and their respective results in **Section 8** and finally, concluding the thesis, discussion, conclusion and outlook in **Section 9**.
The appendix will feature a list of variables (for the equation sets) and physical parameters, definitions for triangulations, the AMATOS and deal.II library, respectively and finally the results for the adaptive volcanic jet or plume setups.

2 Basic equations

In this chapter, the equations are outlined on which the models are based. The Navier-Stokes equations are introduced as the overarching equation set that describes mass, momentum and energy conservation for a Newtonian fluid and as a result allow to model its behavior while also allowing to capture complex phenomena such as turbulence or boundary layers. In this thesis, none of these phenomena are depicted in detail and more detailed analyses can be found in textbooks such as Hirsch (1988) or Chorin and Marsden (1993).

Throughout this thesis, however, mainly the Euler equations are used which can be derived from the Navier-Stokes equations. This chapter serves as a general approach while the more specific case of a volcanic setup will be discussed in a **Section 8**.

2.1 Compressible Navier-Stokes equations

The compressible Navier-Stokes equations (with gravity as body force/source term) in a space-time domain $\Omega \times [0, t_{\max}] \subset \mathbb{R}^d \times \mathbb{R}$ (with $d = 2, 3$) can be written as the following set of partial differential equations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0, \\ \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + P I_d) &= -\rho \nabla \phi + \nabla \cdot \mathbf{F}_{\mathbf{u}}^{\text{visc}}, \\ \frac{\partial (\rho e)}{\partial t} + \nabla \cdot [(\rho e + P)\mathbf{u}] &= -\rho \mathbf{u} \cdot \nabla \phi + \nabla \cdot \mathbf{F}_{\mathbf{e}}^{\text{visc}}, \end{aligned} \qquad \text{(Navier-Stokes equations)}$$

with $\rho \in \mathbb{R}$ being the density, $\rho \mathbf{u} \in \mathbb{R}^d$ the momentum, $\rho e \in \mathbb{R}$ the total energy density (without potential energy), $P \in \mathbb{R}$ the pressure, $\phi \in \mathbb{R}$ the gravitational potential and $\mathbf{F}_{\mathbf{u}}^{\text{visc}} \in \mathbb{R}^d$, $\mathbf{F}_{\mathbf{e}}^{\text{visc}} \in \mathbb{R}$ the viscous fluxes. For this thesis, the gravitational potential is given by $\phi = g\mathbf{k}\mathbf{x} = gz$ ($\mathbf{k}$: vertical unit vector, $\mathbf{x}$ Cartesian coordinates).

For the operators, $\frac{\partial}{\partial t}(\cdot)$ is the partial time derivative, $\nabla \cdot A(x)$ denotes the divergence of a vector field $A(x) \in \mathbb{R}^d$ while $\nabla B(x)$ is the gradient of $B(x) \in \mathbb{R}$, $\cdot$, $\otimes$ are the scalar and tensor product, respectively and I_d is the identity matrix in $d = 2, 3$ dimensions.

The system of equations needs to be closed by the equation of state which is given by the ideal gas law:

$$P = \rho R T, \tag{1}$$

where $R \in \mathbb{R}$ is the specific gas constant of the gas and $T \in \mathbb{R}$ its temperature. Since an ideal gas is assumed, the energy can be calculated by

$$e = \frac{1}{2}\mathbf{u} \cdot \mathbf{u} + c_v T = \frac{1}{2}\mathbf{u} \cdot \mathbf{u} + \frac{P}{\rho(\gamma - 1)}, \tag{2}$$

with $\gamma = \frac{c_p}{c_v} \in \mathbb{R}$ being the ratio of the specific heats of the gas (or specific heat capacities to be more correct) $c_p \in \mathbb{R}$ (isobaric; at constant pressure) and $c_v \in \mathbb{R}$ (isochoric; at constant

volume) and by making use of the equation of state (**Equation 1**) as well as the fact that $R = c_p - c_v$. Rearranging **Equation 2** then also allows to calculate the pressure in terms of conserved variables:

$$P = (\gamma - 1)\left(\rho e - \frac{1}{2}\frac{(\rho\mathbf{u})\cdot(\rho\mathbf{u})}{\rho}\right) = (\gamma - 1)\left(\rho e - \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right). \tag{3}$$

The viscous fluxes are defined as in Giraldo and Restelli (2008):

$$\mathbf{F}_{\mathbf{u}}^{\text{visc}} = \mu\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^{\top} + \lambda(\nabla\cdot\mathbf{u})I_d\right] \tag{4}$$

and

$$\mathbf{F}_{\mathbf{e}}^{\text{visc}} = \mathbf{u}\cdot\mathbf{F}_{\mathbf{u}}^{\text{visc}} + \frac{\mu c_p}{\text{Pr}}\nabla T, \tag{5}$$

where $\mu \in \mathbb{R}$ is the dynamic viscosity, $\lambda = \frac{2}{3}$ follows from the Stokes hypothesis, $\text{Pr} \in \mathbb{R}$ is the Prandtl number and $\top$ denotes the transpose operator.

Values for the physical constants can be found in **Appendix A** (if not defined otherwise in the respective sections).

2.2 Compressible Euler equations

The compressible Euler equations are a simplification of the compressible Navier-Stokes equation in the inviscid case where viscosity is neglected ($\mu = 0$) which leaves us with the following set of fully hyperbolic partial differential equations:

$$\begin{aligned}
\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) &= 0 \\
\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u} + PI_d) &= -\rho\nabla\phi \qquad \text{(Euler equations)} \\
\frac{\partial(\rho e)}{\partial t} + \nabla\cdot[(\rho e + P)\mathbf{u}] &= -\rho\mathbf{u}\cdot\nabla\phi.
\end{aligned}$$

This form is what will be used as basis for most of the work in this thesis.

2.3 Rewriting the equation sets

For simplicity, the equation sets can be rewritten into flux form, such that:

$$\frac{\partial\mathbf{q}}{\partial t} + \nabla\cdot\mathbf{F}(\mathbf{q}) = \mathbf{S}(\mathbf{q}) + \nabla\cdot\mathbf{F}^{\text{visc}}(\mathbf{q}), \tag{6}$$

for the Navier-Stokes equations where $\mathbf{q}$ are the conserved variables $(\rho, \rho\mathbf{u}^{\top}, \rho e)^{\top}$, while the fluxes and source are given by:

$$\mathbf{F}(\mathbf{q}) = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + PI_d \\ (\rho e + P)\mathbf{u} \end{pmatrix}, \quad \mathbf{S}(\mathbf{q}) = \begin{pmatrix} 0 \\ -\rho\nabla\phi \\ -\rho\mathbf{u}\nabla\phi \end{pmatrix} \text{ and } \mathbf{F}^{\text{visc}}(\mathbf{q}) = \begin{pmatrix} 0 \\ \mathbf{F}_{\mathbf{u}}^{\text{visc}} \\ \mathbf{F}_{\mathbf{e}}^{\text{visc}} \end{pmatrix}.$$

For the Euler equations, the viscous flux is omitted so that the flux form is reduced to

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{S}(\mathbf{q}). \tag{7}$$

The flux forms are introduced here, as they are more convenient for later formulations.

2.4 Boundary and initial conditions

To ensure that physical behavior can be modeled by the equation sets, initial values - i.e. values at the the start of a simulation t_0 (usually with $t = t_0$ s) have to be prescribed for all variables. Depending on the problem the initial conditions might be given in primitive variables or the conserved variables. Additionally, values at the boundary of our computational domain Ω have to be prescribed as well. There are several types of boundaries that are possible and a simulation can feature a combination of different boundary conditions as well. For boundary conditions, very often Dirichlet boundary conditions (where values are prescribed), Neumann boundary conditions (where values for the normal derivative are prescribed and thus usually depend on the values of the element adjacent to the edge) or a combination of both are utilized.

2.4.1 Periodic boundaries

These boundaries can be very useful for testing and verifying or if the computational domain is for example a 2D representation of the Earth's surface (map projection). For periodic boundaries, two edges $\partial\Omega_1$ and $\partial\Omega_2$ are paired and the values at the points of these boundaries $\mathbf{x}_1$ and $\mathbf{x}_2$ from one edge are mapped to the other that $\mathbf{q}(\mathbf{x}_1) = \mathbf{q}(\mathbf{x}_2)$.

2.4.2 Wall type boundaries

These boundaries mimic a wall that the flow cannot penetrate. Two cases of this are the no-flux, free-slip and no-slip boundary conditions. The no-flux and free-slip boundary conditions are enforced by $\mathbf{n} \cdot \mathbf{u} = 0$ (with velocity $\mathbf{u}$ and $\mathbf{n}$ being the normal vector at the edge) while the no-slip condition demands that $\mathbf{u} = 0$. Free-slip boundaries also require the tangential stress to be zero (Tan, 2018).
Depending on whether the Euler or Navier-Stokes equations are used energy fluxes can be neglected or have to be set (for example for the Euler equations, the stress field is neglected and thus free-slip are equivalent to no-flux boundaries). For details regarding these conditions, please refer to Giraldo and Restelli (2008) or Birken (2013).

2.4.3 Inflow/outflow boundaries

For inflow boundaries, Dirichlet boundaries are used so that the values of the variables are prescribed at certain points while the outflow boundaries are of mixed type where the values of the energy fluxes are prescribed but the other variables (density and momentum) depend on the values of the element that shares the boundary.

3 Spatial discretization

To solve sophisticated partial differential equations - such as the Navier-Stokes equations - with an algorithm, it becomes necessary to discretize the equation sets (both in space and time), since a computer is not able to compute analytical solutions for such an involved problem. There are many choices for spatial discretization one could choose from. As a brief overview over the most notably used methods, there is the straight-forward approach of using Finite Difference methods (FDM), Finite Volume methods (FVM) which are well-suited for conservation laws and Galerkin methods such as Finite Element methods (FEM) where the solution is approximated by basis functions and finally the Discontinuous Galerkin Finite Element or just Discontinuous Galerkin methods (DG-FEM/DGM) which are a combination of FEM and FVM methods. A brief overview over the methods will be given now, while the next subsection will focus on the DGM in particular.

For the FDM, the approach is rather simple as the partial derivatives are approximated by Taylor expansion such that for example the forward difference in 1D becomes:

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x+\Delta x) - f(x)}{\Delta x}, \tag{8}$$

where f is the function whose derivative is to be calculated, x is the spatial variable and Δx is the grid size. This method has the advantage of being easily implemented but lacks, for example, strategies on how to cope with complex geometries.

For the FVM, balance laws are solved by taking the integral form of the respective partial differential equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_e} \mathbf{q}(\mathbf{x},t)\ \mathrm{d}\mathbf{x} + \int_{\Gamma_e} \mathbf{F}(\mathbf{q}(\mathbf{x},t))\cdot\mathbf{n}\ \mathrm{d}S = \int_{\Omega_e} \mathbf{S}(\mathbf{q}(\mathbf{x},t))\ \mathrm{d}\mathbf{x}, \tag{9}$$

where Ω_e is the local cell for which the quantities are calculated, Γ_e is the cell's boundary, $\mathbf{n}$ is the outer normal vector (for the cell's boundary) and the rest of the quantities are the same as in **Equation 7**. With this relation, the quantities get discretized by taking their mean values and summation over each cell. For details on this approach, one can refer to, for example, LeVeque (1992) or Birken (2013). The FVM is a very natural way to solve conservation laws which makes it a good candidate to use in Computational Fluid Dynamic (CFD) models. A drawback however is the lack of higher order schemes, as most FVM schemes only allow for second order accuracy.

For the FEM, instead of discretizing the calculation domain (such as cells) the function space is approximated. The functions space can be approximated using basis functions such as continuous, piece-wise polynomials which have compact support:

$$\mathbf{q}_N(x,t) = \sum_i \mathbf{q}_i(t)\psi_i(x), \tag{10}$$

Table 2: Table (taken from Hesthaven and Warburton, 2007) to summarize generic properties of widely used spatial discretization methods: Finite Difference methods (FDM), Finite Volume methods (FVM), Finite Element methods (FEM) and Discontinuous Galkerin finite element methods (DG-FEM). The symbols represent that the respective method has the given property, ✓, is not suited for that property, ×, or to be improved upon to be able to but does not intrinsically fulfill that property, (✓).

	Complex geometries	High-order accuracy and *hp*-adaptivity	Explicit semi-discrete form	Conservation laws	Elliptic problems
FDM	×	✓	✓	✓	✓
FVM	✓	×	✓	✓	(✓)
FEM	✓	✓	×	(✓)	✓
DG-FEM	✓	✓	✓	✓	(✓)

where $\mathbf{q}_i$ are time-dependent coefficients for the variables while ψ_i are the basis functions. A straight-forward approach is to use continuous, piece-wise linear polynomials (so-called hat functions) which have certain values in the local cell/element and are zero elsewhere. The partial differential equations can then be rewritten into a system of equations which can then be solved. For details see Ern and Guermond (2004) or Giraldo (2020).

3.1 Discontinuous Galerkin Method

The Discontinuous Galerkin Method (from now on called DGM) is a spatial discretization method. Essentially, it combines the FVM and FEM (Giraldo and Restelli, 2008), as the integration methods (which it consists of) allow for discontinuities (analogous to FVM) while discretizing the function space itself rather than the calculation domain (analogous to FEM).

Using local operators (per element), local solutions can be approximated - instead of calculating a global solution. The prescribed equations are solved using the DGM, by following an approach similar to Beisiegel et al. (2020) or literature such as Hesthaven and Warburton (2007), Dolejsi and Feistauer (2015) and Giraldo (2020).

Using the DGM approach, the flux from **Equation 7** with the variables $\mathbf{q}$, the flux $\mathbf{F}(\mathbf{q})$ and the source vector $\mathbf{S}(\mathbf{q})$ as previously described can be discretized. The computational domain Ω can be decomposed into conforming elements (either triangles or quadrilaterals in this thesis) Ω_i such that $\Omega = \bigcup_i \Omega_i$. In this case, conforming means that all elements make up the boundary, elements do not overlap, each element boundary neighbors another element or is part of the boundary of the domain. A local approximation for the variables $\mathbf{q}_N(x,t) = \sum_i \mathbf{q}_i(t)\psi_i(x)$ can be calculated, where $\psi_i(x)$ are Lagrange polynomials and $\mathbf{q}_i(t)$ are (time-dependent) coefficients.

Integrating by parts once yields the semi-discrete *weak* form (also refer to the mentioned

literature):

$$\int_{\Omega_e} \left(\frac{\partial \mathbf{q}_{N,e}}{\partial t} - \mathbf{F}_{N,e} \cdot \nabla - \mathbf{S}_{N,e} \right) \psi_{i,e}(\mathbf{x}) \, \mathrm{d}\mathbf{x} = - \int_{\Gamma_e} \psi_{i,e}(\mathbf{x}) \mathbf{n} \cdot \mathbf{F}^*_{N,e} \, \mathrm{d}S, \qquad (\textit{weak} \text{ form})$$

where $\mathbf{q}_{N,e}$, $\mathbf{F}_{N,e}$, $\mathbf{S}_{N,e}$ are the local approximation of the conserved variables $\mathbf{q}_{N,e}$, $\mathbf{F}_{N,e} = \mathbf{F}(\mathbf{q}_{N,e})$ and $\mathbf{S}_{N,e} = \mathbf{S}(\mathbf{q}_{N,e})$ inside each element. Ω_e is the element, Γ_e its boundary, $\mathbf{n}$ its normal unit vector, ψ are the test/basis functions and $\mathbf{F}^*_{N,e}$ is the numerical (Rusanov) flux for $\mathbf{q}^-_{N,e}$ and $\mathbf{q}^+_{N,e}$ (which are the local approximation of the conserved quantities on both sides of the element boundary Γ_e, respectively) with:

$$\mathbf{F}^*(\mathbf{q}^-_{N,e}, \mathbf{q}^+_{N,e}) = \frac{1}{2} \left[\mathbf{F}(\mathbf{q}^-_{N,e}) + \mathbf{F}(\mathbf{q}^+_{N,e}) - |\lambda| (\mathbf{q}^+_{N,e} - \mathbf{q}^-_{N,e}) \mathbf{n} \right], \qquad (11)$$

where $\lambda = \max_{i=L,R}(|U^i| + \sqrt{\gamma P^i / \rho^i})$ is the characteristic speed which is comprised of the normal component of velocity with respect to the edge Γ_e given by $U^i = \mathbf{u} \cdot \mathbf{n}$ and the speed of sound $\sqrt{\gamma P^i / \rho^i}$, where γ is the specific heat ratio.

Regarding the elements, it should be noted that while there are many other choices for elements, triangular and quadrilateral elements are easily implemented (compared to hexagonal element, for example), which is also a standard approach. The property that the elements are to be conforming elements makes sure that every element edge is either part of the domain boundary or neighbors another while none of the elements overlap (Müller, 2012).

Advantages of triangular elements include their flexibility concerning the geometry. Especially curved shapes can be represented well by using triangular grids for which reason they are well suited for complex geometries. On the other hand, quadrilateral elements make calculations easier.

After the domain has been decomposed into the elements, all calculations are performed on reference elements - reference triangle or reference square which are the $[-1, 1]$ right triangle or $[-1, 1]^2$ reference square, respectively (for 2D). As an example, **Figure 1** shows the reference triangle. In theory, an element can have any triangular or quadrilateral shape which makes a mapping from the elements to the reference elements necessary. This mapping is given by $\mathbf{x} = \boldsymbol{\Psi}(\boldsymbol{\xi})$ from the reference triangle coordinates $\boldsymbol{\xi} = (\xi, \eta)^\tau$ to the element with Cartesian coordinates $\mathbf{x} = (x, z)^\tau$. The reference triangle is then defined as $\widehat{\Omega_e} = \{(\xi, \eta) \in \mathbb{R}^2, -1 \leq \xi, \eta \leq 1, \xi + \eta \leq 0\}$ (see Giraldo and Warburton, 2008) while the reference square is defined as $\widehat{\Omega_e} = \{(\xi, \eta) \in [-1, 1]^2\}$ (see Giraldo et al., 2002). For quadrilateral elements, the mapping is similar but a reference rectangle/square is used instead of a reference triangle.

The Lagrange polynomials $\psi(x)$ that are part of the local approximation are the basis functions which are also mapped to the reference elements. In this thesis, this basis is

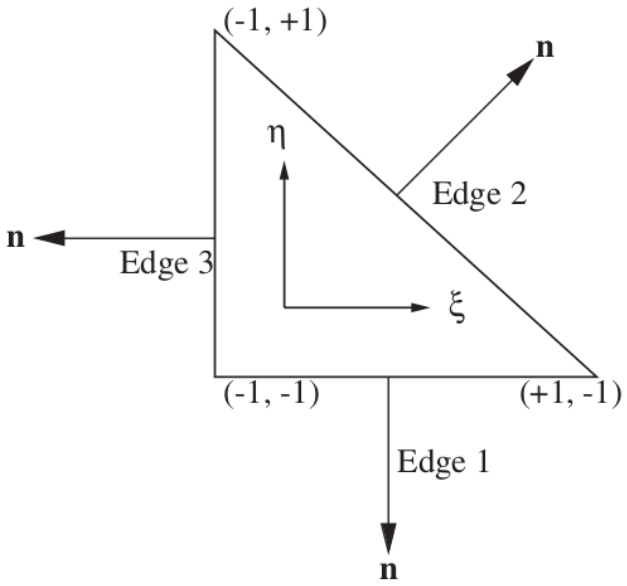

Figure 1: Reference triangle with coordinate directions ξ, η and normal vectors $\mathbf{n}$ (taken from Giraldo and Warburton, 2008).

constructed as a set of nodal (physical space) basis functions.
Using those Lagrange polynomials, we can define the mass and differentiation matrix which we can use for a more convenient formulation of the *weak* form. The mass matrix is defined as

$$M_{ij}^e = \int_{\Omega_e} \psi_{i,e}(\mathbf{x})\psi_{j,e}(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{12}$$

for the interior of the element and as

$$M_{ij}^s = \int_{\Gamma_e} \psi_{i,e}(\mathbf{x})\psi_{j,e}(\mathbf{x})\mathbf{n} \, \mathrm{d}S \tag{13}$$

for the edges. The differentiation matrix is defined as

$$D_{ij}^e = \int_{\Omega_e} \psi_{i,e}(\mathbf{x})\nabla\psi_{j,e}(\mathbf{x}) \, \mathrm{d}\mathbf{x}. \tag{14}$$

The *weak* form can be rewritten into a semi-discrete system:

$$\begin{aligned} \int_{\Omega_e} \psi_{i,e}(\mathbf{x})\psi_{j,e}(\mathbf{x}) \, \mathrm{d}\mathbf{x} \, \frac{\partial \mathbf{q}_j}{\partial t} - \int_{\Omega_e} \psi_{i,e}(\mathbf{x})\nabla\psi_{j,e}(\mathbf{x}) \, \mathrm{d}\mathbf{x} \cdot \mathbf{F}_j - \int_{\Omega_e} \psi_{i,e}(\mathbf{x})\psi_{j,e}(\mathbf{x}) \, \mathrm{d}\mathbf{x} \, \mathbf{S}_j \\ = - \int_{\Gamma_e} \psi_{i,e}(\mathbf{x})\psi_{j,e}(\mathbf{x})\mathbf{n} \, \mathrm{d}S \cdot \mathbf{F}_j^*, \end{aligned} \tag{15}$$

with $\mathbf{F}_j = \mathbf{F}(\mathbf{q}_j)$, $\mathbf{S}_j = \mathbf{S}(\mathbf{q}_j)$ and $\mathbf{F}_j^* = \mathbf{F}^*(\mathbf{q}_j^-, \mathbf{q}_j^+)$ which simplifies to

$$M_{ij}^e \, \frac{\partial \mathbf{q}_j}{\partial t} - (D_{ij}^e)^\top \mathbf{F}_j - M_{ij}^e \, \mathbf{S}_j = -(M_{ij}^s)^\top \, \mathbf{F}_j^*, \tag{16}$$

if the mass and differentiation matrices are utilized. The matrix form allows for a faster overview of what needs to be calculated (also in terms of elements or edges). There is also the possibility to eliminate the mass matrix by substituting the basis functions by basis functions that are multiplied with the inverse mass matrix. For details see Giraldo and Warburton (2008). Please remark that the calculation for both **Equation 13** and **Equation 14** contain vectors while the calculation for **Equation 12** is just a scalar. In the end, the dimensions do match up, however, since the fluxes that get multiplied with the mass matrix for the edges, as well as the differentiation matrix contain vectorized quantities themselves.

As mentioned in Giraldo and Warburton (2008), the DGM form that is used here, is valid for all ψ that are twice integrable and elements of the N dimensional polynomial space: $\psi \in \mathcal{S}$ with

$$\mathcal{S} = \{\psi \in \mathcal{L}_2(\Omega) : \psi|_{\Omega_e} \in P_N(\Omega_e) \forall \Omega_e\},$$

P_N being the polynomial space defined on every element Ω_e and $\Omega = \bigcup_i \Omega_i$ being the whole computational domain.

Additionally, it is necessary to define the (numerical) integration for all the integrals that have been defined in this subsection. The integration can be performed by utilizing the Gauss quadrature. The equations for boundary and area integrals for two functions f and g are defined as follows:

$$\int_{\Omega_e} f(\mathbf{x})g(\mathbf{x})\ \mathrm{d}\mathbf{x} \approx \sum_{i=1}^{N_{q,e}} \lambda_i^e |J^e(\xi_i)| f(\xi_i)g(\xi_i), \tag{17}$$

$$\int_{\Gamma_e} f(\mathbf{x})g(\mathbf{x})\ \mathrm{d}S \approx \sum_{i=1}^{N_{q,g}} \lambda_i^g |J^g(\xi_i)| f(\xi_i)g(\xi_i), \tag{18}$$

where $N_{q,e}$ and $N_{q,g}$ are the numbers of quadrature points for both area and boundary integral, λ_i^e, λ_i^g are the weights associated with each quadrature point ξ_i and J^e, J^g are the respective area of the element or length of the edge (Giraldo and Warburton, 2008).

3.2 Slope limiter

Even though slope limiters are not used extensively in this thesis, they should find a short mention. The name of the Discontinuous Galerkin Method already suggests that it allows or rather is based on allowing for discontinuous solutions at the element boundaries which results in (possibly) different values at the nodes of neighboring elements.
If a so called slope limiter is introduced, it ensures stability and dampens oscillations (such as Gibbs phenomena) around the element edges. The limiter works by adjusting

the gradient between elements to better fit the values at the nodes and reduced over- and undershoots as a result. For a detailed approach on slope limiters, either Cockburn and Shu (1998) or Cockburn and Shu (2001a) are advised.
Following the mentioned literature, the limiter is applied by calculating average values of all quantities in the current element and deviations at the edge midpoints from the averages and then determining a parameter from these deviations that gets added to the average values. If smooth extrema occur, the limiter ensures that the original values (before applying the limiter) are kept.

3.3 Grid adaptivity

Grid adaptivity describes a numerical procedure to control the resolution/accuracy of a numerical scheme by increasing the resolution where it is necessary while decreasing it in regions where it is not as relevant. Adaptive calculations have the advantage of reduced computational cost since the accuracy in irrelevant areas can be decreased which leads to an improvement in CPU time on the one hand, while on the other hand, the resolution can be increased in areas in need of (e.g.) finer grids without having a fine uniform grid. Müller et al. (2013) states that the total CPU time can be reduced by up to 50 % if adaptive grids are utilized, while the error in respect to the uniform calculation becomes insignificant with increasing resolution. Consequently, adaptivity is very useful for prognostic simulations (such as volcanic eruptions).
Adaptive mesh refinement can be achieved in different ways which vary greatly in their procedure. According to Müller (2012), there are three distinct kinds of adaptivity:

- **h-adaptivity**: Here, the (spatial) resolution is increased or decreased by adding or removing grid nodes which leads to a locally refined or coarsened grid. Since array storage is not static for these procedures, a good grid management becomes a necessity. In addition, by adding nodes into the grid, the values at these coordinates have to be calculated (usually via interpolation). Ideally, neighboring nodes should be stored in proximity to one another inside the respective arrays for computational efficiency. This kind of adaptivity is used in this thesis.
- **p-adaptivity**: Here, the grid resolution remains constant while the numerical accuracy is controlled by changing the order of the basis functions (polynomials). For areas where better resolution is required, the polynomial order is increased while it is decreased if that accuracy is longer needed. A description of this kind of adaptivity can be found in Babuška et al. (1981).
- **r-adaptivity**: Here, the number of degrees of freedom remains constant during the calculation. The accuracy of the scheme is varied by shifting around the nodes so that the mesh is refined in areas where higher resolution is needed. Adding or removing nodes is not an issue with this procedure, however, the mesh can be distorted if the node spacing becomes too great in certain areas. As a result, this can lead to an addi-

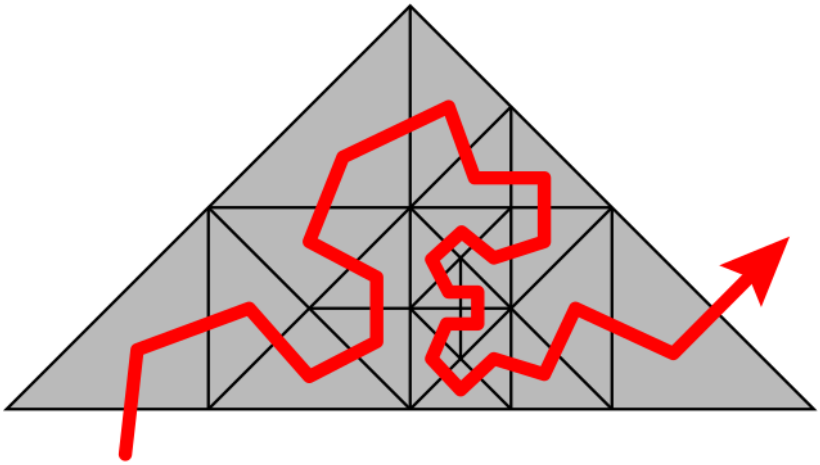

Figure 2: Example of a space filling curve on a triangulation composed of right triangles (taken from Müller, 2012).

tional reduction in accuracy in these areas, especially since many numerical methods are not well-suited to deal with distorted elements. Additionally, the algorithms for r-adaptivity have to take moving nodes into account. This kind of adaptivity is used in Kühnlein et al. (2012).

These kinds of adaptivity can be combined (Lang et al., 2003; Piggott et al., 2005), but in this thesis only h-adaptivity is utilized.

3.3.1 Space filling curves

First discovered by Peano (Peano, 1890), space filling curves provide an algorithm or a method to fill 2D space by a 1D curve with the goal that points in close proximity to one-another in the 2D plane will also be close to each other inside the curve as well. Usually, the idea is to have an iteration that starts with a "coarse" curve and fills up more space as the scheme is iterated. The Hilbert curve presented by David Hilbert in 1891 (Hilbert, 1891) is one prominent example of a space filling curve.
The properties space filling curves have is advantageous and desirable for an h-adaptive mesh, so many mesh refinement methods utilize space filling curves to manage node and element connections. For triangulations, a space filling curve is created in similar ways as the example shown in **Figure 2**.

3.3.2 Grid refinement and coarsening criteria

In order to have a useful grid adaption which means efficient refinement, as well as coarsening, a refinement criterion is required (Behrens et al., 2005). The choice of refinement criterion is a highly debated topic both in the adaptive mesh refinement (AMR) community (e.g. Li, 2010; Fryxell et al., 2000; Berger and Oliger, 1984), as well as in this thesis.

Within this thesis, refinement and coarsening will be performed by using an error estimator or indicator: For each cell, an error is calculated and compared to the maximum error that has been calculated weighted by a certain tolerance. If the error is larger than the maximum error weighted by a refinement tolerance, the cell will be refined and similarly, if the error is smaller than the maximum error weighted by a coarsening tolerance, the cell will be coarsened.
The criteria can also be expressed as follows:

$$\mathbf{e}(\Omega_i) > \sigma_{\text{ref}} \max_{\Omega_i \subset \Omega} \mathbf{e}(\Omega_i), \tag{19}$$

for refinement and

$$\mathbf{e}(\Omega_i) < \sigma_{\text{coarse}} \max_{\Omega_i \subset \Omega} \mathbf{e}(\Omega_i), \tag{20}$$

for coarsening, where $\mathbf{e}(\Omega_i)$ is the error (either calculated by error indicator or error estimator) within each cell Ω_i, σ_{ref} is the tolerance for grid refinement while σ_{coarse} is the tolerance for grid coarsening.
The values for the refinement/coarsening tolerances σ_{ref} and σ_{coarse}, as well as the specifics on how to approximate the errors will be presented in the respective sections where adaptivity will be relevant.

4 Time stepping/time-integration

In addition to having to make a choice how to discretize an equation in space, another choice has to be made on how to advance the solution in time - a time stepping or time-integration scheme.
Once the equations are discretized spatially, one is left with a system of ordinary differential equations (ODEs)

$$\frac{\mathrm{d}\mathbf{q(t)}}{\mathrm{d}t} = L(t, \mathbf{q(t)}),\ \mathbf{q}(t_0) = \mathbf{q}_0,\ t \in [t_0, t_{\max}], \tag{21}$$

where $\mathbf{q}$ are the conserved variables as in the equation section and L is the differential operator of the respective PDE, which are either the Navier-Stokes or Euler equations.
If the differential operator should have no explicit dependence on time, a reformulation into an autonomous system is possible such that

$$\frac{\mathrm{d}\mathbf{q(t)}}{\mathrm{d}t} = L(\mathbf{q(t)}),\ \mathbf{q}(0) = \mathbf{q}_0,\ t \in [t_0, t_{\max}]. \tag{22}$$

In the literature, there exists a large number of methods on how to solve problems of this type. For an overview, please refer to Hairer et al. (2000) or Hairer and Wanner (1996).

In general, to advance the solution (or system of ODEs) in time, a choice has to be made whether an explicit or implicit time stepping scheme is picked. Additionally, a combination of both forms is also possible which in turn leads to solving problems in the form of either

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t f(\mathbf{q}^n),\ \text{or}\ \mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t f(\mathbf{q}^{n+1}),$$

where $\mathbf{q}^n \in \mathbb{R}^m$ is the solution vector at the current time step n with time step size Δt and f is an operator that determines values for the previous time steps (and depends on the differential operator L). Depending on the time stepping method, the operator f can be very different.
The very basic approaches for either explicit or implicit time stepping methods are given by the explicit/implicit Euler method, where

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t L(\mathbf{q}^n),\ \text{or}\ \mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t L(\mathbf{q}^{n+1}).$$

Explicit methods have the advantage that the solution can be computed in a rather straight-forward manner but come with time step size restrictions while implicit methods require an additional system of equations to be solved but have better stability properties. In this thesis, there will be no focus on the whole theory of stability of a time stepping scheme (just certain aspects which will be explained in the next subsection).

4.1 Conditional stability for explicit time stepping methods

For explicit time stepping methods, the time step size needs to be restricted as there is a relationship between the spatial and temporal resolution in conjunction with the

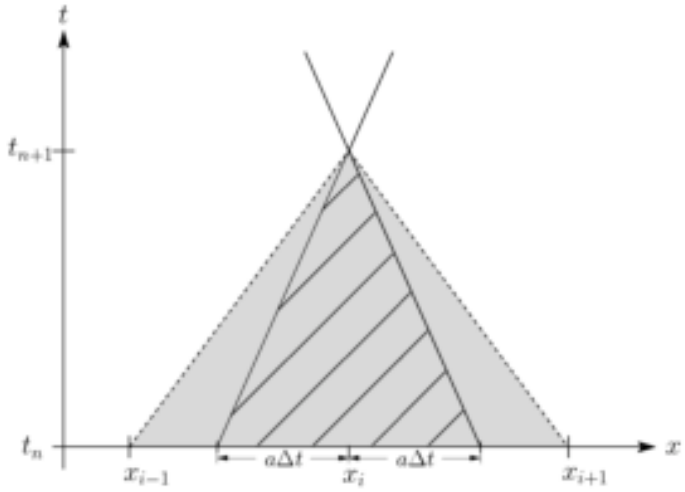

Figure 3: This figure is taken from Birken (2013) and illustrates the CFL condition. In particular, it shows how the exact solution for a linear equation at (x_i, t_{n+1}) can be influenced by points within the shaded area. As a result, the numerical domain (grey; would correspond to Ω from **Section 3.1**) has to contain the shaded area if stability is to be ensured which results in constraints for the ratio of the spatial and temporal resolution $\Delta t/\Delta x$.

propagation speed of the flow. This condition is known as the CFL (or Courant-Friedrichs-Levy) criterion (Courant et al., 1928) which states that explicit time stepping schemes can only be stable up to certain CFL numbers. For elements of order p the restriction for the time step is given by (see Schoeder et al., 2018):

$$\Delta t = \frac{\mathrm{CFL}_{\max}\Delta x}{p^{1.5}c}, \tag{23}$$

where we have the time step size Δt, the spatial resolution Δx, maximum propagation speed of the flow c and the maximum CFL number $\mathrm{CFL}_{\max}$. The propagation speed is given by the eigenvalues of the Jacobian of the inviscid flow (see e.g. Birken, 2013). Implicit methods automatically satisfy this condition which is the reason why they are stable regardless of time step size.

4.2 Runge-Kutta methods

Runge-Kutta are one of the most widely used (one-step) methods to solve ODEs. For Runge-Kutta methods, the so-called stages $\mathbf{k}_i$ are calculated which then are used to update the solution vector. For an s-stage Runge-Kutta method these coefficients can be calculated by (following Hairer and Wanner, 1996)

$$\begin{aligned}\mathbf{k}_i &= L(t^n + c_i\Delta t, \mathbf{q}^n + \Delta t\sum_{j=1}^{s} a_{ij}\mathbf{k}_j),\ i = 1,\dots,s\\ \mathbf{q}^{n+1} &= \mathbf{q}^n + \Delta t\sum_{i=1}^{s} b_i\mathbf{k}_i.\end{aligned} \tag{24}$$

In the case of an autonomous system (**Equation 22**), this can be reduced to:

$$\mathbf{k}_i = L(\mathbf{q}^n + \Delta t \sum_{j=1}^{s} a_{ij}\mathbf{k}_j), \; i = 1, \ldots, s$$
$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \sum_{i=1}^{s} b_i \mathbf{k}_i. \tag{25}$$

For both cases of the Runge-Kutta method, the coefficients a_{ij}, b_i and c_i have to be defined. For simplicity, these coefficients can be collected as $\mathbf{A} = (a_{ij})_{ij} \in \mathbb{R}^{s\times s}$, $\mathbf{b} = (b_1, \ldots, b_s)^\top$ and $\mathbf{c} = (c_1, \ldots, c_s)^\top$ and rewritten into a Butcher tableau (which is the usual way of describing a Runge-Kutta method):

$$\begin{array}{c|ccc} c_1 & a_{11} & \ldots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \ldots & b_s \end{array} = \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^\top \end{array}.$$

A Runge-Kutta method that has a lower triangular matrix $\mathbf{A}$ is an explicit method, while those that have at least one non-zero entry in the upper triangle (so also on the diagonal) has implicit entries.
An explicit Runge-Kutta method then has a Butcher tableau like

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{12} & & & \\ \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & \ldots & a_{s,s-1} & 0 \\ \hline & b_1 & \ldots & \ldots & b_s \end{array}$$

In this thesis, two types of Runge-Kutta methods are used: the strongly stability preserving Runge-Kutta (SSPRK) method (Cockburn and Shu, 2001b) and the low-storage Runge-Kutta method (Kennedy et al., 2000). As an overview, the LSRK needs less distinct values in the Butcher tableau as additional structure in the coefficients is assumed so that the tableau is reduced to:

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_1 & & & & \\ \vdots & b_1 & a_2 & & & \\ \vdots & \vdots & b_2 & a_3 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ c_s & b_1 & \ldots & \ldots & b_{s-1} & 0 \\ \hline & b_1 & \ldots & \ldots & \ldots & b_s \end{array}$$

The SSPRK method enforces stricter stability conditions for flow problems. This method includes a total variation diminishing (TVD) properties which is for example used in TVD Runge-Kutta methods which were introduced in Shu (1988) and Shu and Osher (1988).

TVD schemes need to satisfy that the total variation (TV) does not increase with each timestep such that $\mathrm{TV}(\mathbf{q}^{n+1}) \leq \mathrm{TV}(\mathbf{q}^{n})$ where n denotes the time step or written in another way:

$$\|\mathbf{q}^{n+1}\| \leq \max_{0\leq k\leq s} \|\mathbf{q}^{k}\|.$$

In addition to the scheme being a TVD scheme, SSPRK methods need to satisfy the following conditions:

- if $\beta_{ij} \neq 0$ then $\alpha_{ij} \neq 0$,
- $\alpha_{ij} \geq 0$
- $\sum_{j=0}^{i-1} \alpha_{ij} = 1$,

where α_{ij} and β_{ij} are coefficients as presented in Cockburn and Shu (2001b). It should be noted that another notation for the RK method is used.
The SSPRK is used because of stability properties while the LSRK is useful due to its computational efficiency. In this thesis, schemes of at least order 2 are used. The exact methods (with the respective values for the coefficients) can be found in Cockburn and Shu (2001b) and Kennedy et al. (2000) while Birken (2013) also provides the coefficient values in line with the notation for the RK method used in this thesis.

4.3 Rosenbrock-type methods

Rosenbrock-type (also known as Rosenbrock-Wanner; ROW for short) methods are in a sense related to Runge-Kutta methods. They are similar to so-called diagonally implicit Runge-Kutta (DIRK) methods. For this, an s-stage DIRK method is taken and the operator L is linearized. As a result, accuracy and stability are sacrificed while gaining computational efficiency. Nonetheless, they prove to be a good alternative to DIRK methods and have been found to be competitive (John and Rang, 2010; Jax et al., 2021).

For a derivation and more explanations regarding the Rosenbrock method compared to the (S)DIRK, one should refer to Birken (2013).
For implicit schemes, the linear system that arises is solved using Krylov subspace methods - namely GMRES which is explained in Saad and Schultz (1986). Out of the two numerical bases presented in the next section, deal.II does not allow for implicit time stepping, while capabilities for ROW time stepping are present in StormFlash. Unfortunately, a code restructure led to errors within the ROW implementation and thus it is not usable on an operational level. Consequently, the results that will be shown in this thesis will be results obtained by explicit methods.

5 Numerical bases used in this thesis - StormFlash and deal.II

After getting familiar with the theoretical background, the approaches for modeling fluid flow with the Euler equations in a DG context can be presented.
In this thesis, mainly two implementations for the Euler equations with a DGM are utilized:

- StormFlash, a code that is based on the AMATOS library (see **Appendix B**) and is used in the *Numerical Methods in Geosciences* working group at the University of Hamburg.
- Code adapted from a deal.II tutorial (step-67) which is a very well-developed and maintained library (Arndt et al., 2020).

The main difference between these two libraries for the implementations for this thesis is the use of elements: StormFlash makes use of triangular elements while the deal.II utilizes quadrilateral elements.
StormFlash is implemented in such a way that SSPRK or ROW methods are used for time stepping, whereas the implementation based on the deal.II example uses the LSRK time stepping. As mentioned previously, the ROW within StormFlash is not implemented fully, however.
It should be noted that during my time as PhD candidate StormFlash was restructured which led to changes in the code. All results for the verification and validation were obtained after the code was restructured.
While, in many cases, the results from deal.II do not prove to be as physically plausible as expected (see e.g. **Section 7.3.2** or **Section 8.3.1**), they are still presented here because of the large part they took throughout the PhD.

Furthermore, a 1D DG code similar to StormFlash is used for research of the stabilization method (next main section) but verification and validation will not be shown for this implementation as it only plays a minor role. Additionally, this code did produce better results (as can be seen in a later section).

Both verification and validation are done without gravity source term as this is the topic of the next main section.

5.1 Verification of the implementations

The verification of the implementations is done using a convergence study. For the sake of comparing and quantifying results between different setups (e.g. different resolutions or implementations), we have to calculate the error of an implementation. The analysis of an implementation's error is done by calculating the L_2 and L_∞ errors. The relative

errors are calculated by

$$\|\mathbf{q}\|_{L_2} = \sqrt{\frac{\int_\Omega (\mathbf{q} - \mathbf{q}_{\text{exact}})^2 \, \mathrm{d}\mathbf{x}}{\int_\Omega \mathbf{q}_{\text{exact}}^2 \, \mathrm{d}\mathbf{x}}} \tag{26}$$

for the L_2 errors and

$$\|\mathbf{q}\|_{L_\infty} = \frac{\max_{\mathbf{x}\in\Omega} |\mathbf{q} - \mathbf{q}_{\text{exact}}|}{\max_{\mathbf{x}\in\Omega} |\mathbf{q}_{\text{exact}}|} \tag{27}$$

for the L_∞ errors. If absolute errors are necessary, they can be calculated similarly by omitting the normalization by the division by the denominators.
If the L_2 or L_∞ are mentioned in later sections they are calculated as defined by these equations.

The (experimental) rate of convergence σ can be calculated which allows for a better analysis of the convergence order of the respective method. The rate of convergence is defined by the quotient of errors for a resolution and the next coarser case:

$$\sigma_{\mathbf{q}}(\Delta_i) = \log_2 \left(\frac{\|\mathbf{q}(\Delta_{i-1})\|_k}{\|\mathbf{q}(\Delta_i)\|_k} \right), \tag{28}$$

where $\mathbf{q}$ are the respective variables, Δ_i (or Δ_{i-1}) are the respective resolutions of the case and $k = 2, \infty$. Similarly to the errors, if the rate of convergence is mentioned in later sections, this definition is used.

Since the deal.II example page which the deal.II implementation is based on (https://www.dealii.org/current/doxygen/deal.II/step_67.html) already features a convergence study, only the convergence study for StormFlash will be shown. For convenience, a different test case is used that was already used to analyze the convergence rates and consequently the order of the numerical scheme in my master's thesis (Bänsch, 2017). For the convergence study, a test case similar to the vortex test cases in Vater (2013) and Benacchio (2014) is used. Here, a quasi-stationary vortex is transported along the diagonal on the unit square (with the domain $\Omega = [0\ \mathrm{m}, 1\ \mathrm{m}]^2$). The boundary conditions are chosen to be periodic everywhere. Once the vortex has completed one period and has returned to its starting position, the simulation is stopped. The period and thus the simulation length $t_{\max}$ is set to one second (with velocities of 1 m s^{-1} in both directions as well).
The derivation of this vortex test case is done using the axisymmetric Euler equations (Zheng and Zhang, 1998), where the density $\rho = 1$ kg m^{-3}, the radial velocity v_r is set to 0 m s^{-1}, the tangential velocity v_θ and pressure P will be defined in the following equations. The vortex center (x_m, y_m) m is located in the center of the unit square so that $(x_m, y_m) = (0.5, 0.5)$ m.

The tangential velocity is given by

$$v_\theta(r) = \begin{cases} v_{\max} \dfrac{s \cdot r}{r_m^2 - r^2} \cdot \sqrt{2 \exp\left(\dfrac{1}{r^2 - r_m^2}\right)} & 0 \le r < r_m, \\ 0 & \text{otherwise,} \end{cases}$$

where the distance to the center r can be calculated with

$$r = \sqrt{(x - x_m)^2 + (y - y_m)^2}.$$

$r_m = 0.45$ m is the fixed radius of the vortex, $v_{\max} = 0.5$ m s^{-1} is the maximum tangential velocity and s is a scaling factor that is calculated by

$$s = \frac{|r_m^2 - r_{vm}^2|}{r_{vm} \sqrt{2 \exp\left(\frac{1}{r_{vm}^2 - r_m^2}\right)}},$$

with r_{vm} being the distance at which the tangential velocity reaches its maximum value.

$$r_{vm} = \frac{1}{2} \sqrt{-2 + 2\sqrt{1 + 4 r_m^4}}$$

The velocities are initialized as:

$$u_0(x, y) = u_{\text{bg}} - v_\theta(r) \sin(\theta) \quad \text{and} \quad v_0(x, y) = v_{\text{bg}} + v_\theta(r) \cos(\theta),$$

where $(u_{\text{bg}}, v_{\text{bg}}) = (1, 1)$ m s^{-1} are the background velocities in x- and y-direction respectively (so that the vortex is transported diagonally across the domain). The angle θ can be obtained via $\theta = \arctan((y - y_m)/(x - x_m))$.

The pressure is defined by the following equation:

$$P = \begin{cases} P_{\text{ref}} - 2\rho v_{\max}^2 s^2 \exp\left(\dfrac{1}{r^2 - r_m^2}\right) & 0 \le r < r_m, \\ P_{\text{ref}} & \text{otherwise,} \end{cases}$$

where $P_{\text{ref}} = 1$ Pa. In this convergence study, linear elements are used which should result in a scheme of second order. The L_2 and L_∞ errors for resolutions of $\Delta_i = \Delta x = \Delta y = 2^{-(3+i)}$ m with $i = 1, 2, 3, 4, 5, 6$ are shown **Figure 4** with a time step that is adapted via a fixed CFL number of 0.15. Both plots include an error curve for a scheme of order 2 for easier comparison between the errors and the expected outcome. Upon inspection, it is observable that the errors for all variables seem to fit a scheme of order 2. Additionally, this can be verified by the rates of convergence given in **Table 3** which confirm that the rates of convergence are of order 2. Consequently, this test case verifies that StormFlash is a scheme that is second order accurate with linear elements. While it will not be shown here, StormFlash also allows for higher order elements that also provide schemes with expected order of accuracy.

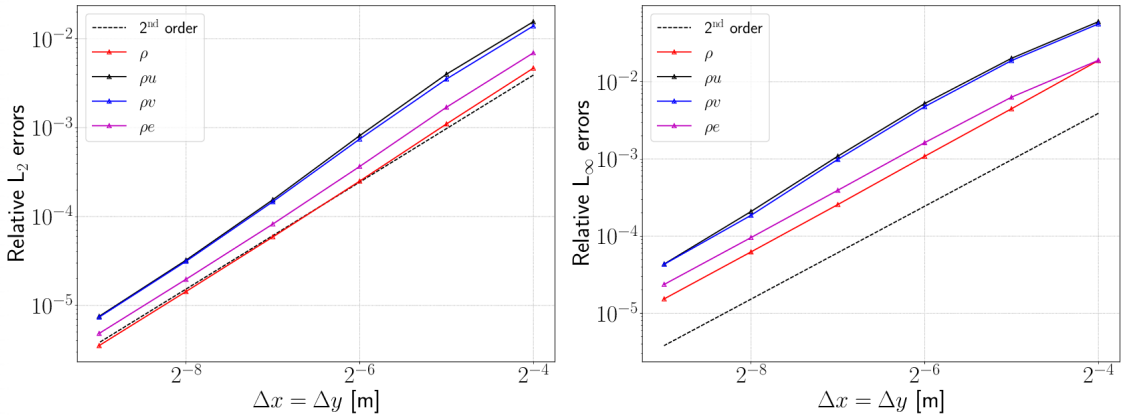

Figure 4: L_2 and L_∞ errors for quasi-stationary vortex for StormFlash. The plot on the left-hand side shows the L_2 errors while the figure on the right shows the L_∞ errors. The errors show density in red, x-momentum in black, y-momentum in blue and the energy in magenta and show the results for resolutions of $\Delta x = \Delta y = 2^{-(3+i)}$ with $i = 1, 2, 3, 4, 5, 6$. Additionally, a line with slope 2 (for the logarithmic plots; dashed black line) is added as reference.

Table 3: Relative errors and rate of convergence for the quasi-stationary vortex for StormFlash.

L_2 error after $t_{\max}$ with	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
$\Delta_1 = 2^{-4}$ m	4.678e-03	-	1.557e-02	-	1.395e-02	-	6.949e-03	-
$\Delta_2 = 2^{-5}$ m	1.102e-03	2.09	4.004e-03	1.96	3.529e-03	1.98	1.699e-03	2.03
$\Delta_3 = 2^{-6}$ m	2.504e-04	2.14	8.136e-04	2.30	7.448e-04	2.24	3.660e-04	2.21
$\Delta_4 = 2^{-7}$ m	5.895e-05	2.09	1.546e-04	2.40	1.471e-04	2.34	8.235e-05	2.15
$\Delta_5 = 2^{-8}$ m	1.432e-05	2.04	3.231e-05	2.26	3.135e-05	2.23	1.968e-05	2.07
$\Delta_6 = 2^{-9}$ m	3.531e-06	2.02	7.536e-06	2.10	7.377e-06	2.09	4.839e-06	2.02
L_∞ error after $t_{\max}$ with	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
$\Delta_1 = 2^{-4}$ m	1.889e-02	-	6.021e-02	-	5.636e-02	-	1.905e-02	-
$\Delta_2 = 2^{-5}$ m	4.481e-03	2.08	2.021e-02	1.57	1.888e-02	1.58	6.331e-03	1.59
$\Delta_3 = 2^{-6}$ m	1.084e-03	2.05	5.238e-03	1.95	4.771e-03	1.98	1.630e-03	1.96
$\Delta_4 = 2^{-7}$ m	2.568e-04	2.08	1.088e-03	2.27	9.904e-04	2.27	3.932e-04	2.05
$\Delta_5 = 2^{-8}$ m	6.247e-05	2.04	2.086e-04	2.38	1.862e-04	2.41	9.581e-05	2.04
$\Delta_6 = 2^{-9}$ m	1.538e-05	2.02	4.351e-05	2.26	4.315e-05	2.11	2.367e-05	2.02

5.2 Validation of the implementations

The implementations are validated using the prominent test case with flow around a cylinder. This is done as to ensure that the two implementations are comparable. This test case is one of the cases that is implemented in the deal.II step-67 tutorial. In 2D, it is implemented in such a way that flow in a channel around a cylinder (without source term) is simulated (see the deal.II example page: https://www.dealii.org/current/doxygen/deal.II/step_67.html).
The channel domain is set up such that $\Omega = [0\text{ m}, 2.2\text{ m}] \times [0\text{ m}, 0.41\text{ m}]$ and the mentioned cylinder is positioned at $[0.2\text{ m}, 0.2]$ m with a diameter of 0.1 m (for 2D).

For the boundaries, an inflow boundary is situated at the left wall. Top, bottom and cylinder walls are imposed with a no-penetration (wall) boundary and finally a subsonic outflow boundary is prescribed for the right wall boundary such that the boundary value $\mathbf{q}^+$ is given by $\mathbf{q}^+ = (\rho^-, (\rho\mathbf{u}^-, (\rho e)_D))^\top$ with density and momentum from values $\mathbf{q}^-$ inside the neighboring element and the prescribed Dirichlet values for energy.

The flow field is initialized as a subsonic field with a Mach number of Ma = 0.307, a constant velocity in x-direction (of 0.4), no velocity in y-direction, constant density (of 1) and an energy that corresponds to 1.3 times the speed of sound compared to the background velocity field. These values are also imposed for the inflow.

For both the deal.II implementation, as well as StormFlash, fixed meshes (in time) are used and are shown in in **Figure 5**. For this test case, the polynomial degree of the elements is set to be quadratic so that the results from the deal.II example page can be used as reference.

To allow for an easier comparison with the "reference" solution shown on the deal.II tutorial example web-page for step-67, the pressure profiles are shown with a color bar from 0.87 to 1.6 Pa. Snapshots for both deal.II and StormFlash are plotted at $t = 0.1$ s and $t = 2.0$ s while for the latter, part of the domain close to the outflow boundary is shown to focus on the waves that get reflected at the boundary. These results are presented in **Figure 7** ($t = 2.0$ s).
Both StormFlash, as well as deal.II yield very similar results. In certain regions, however, deal.II yields results with more prominently outlined pressure profiles which can most likely be explained by higher spatial resolution in these areas since StormFlash has a finely refined grid close to the cylinder but coarser resolution compared to the deal.II case if further away from the cylinder. This can be observed in the comparison of the development of the acoustic wave at $t = 0.1$ s (**Figure 6**) or for the acoustic waves at the back at $t = 2.0$ s (**Figure 7**) where the pressure profile for the simulation with deal.II

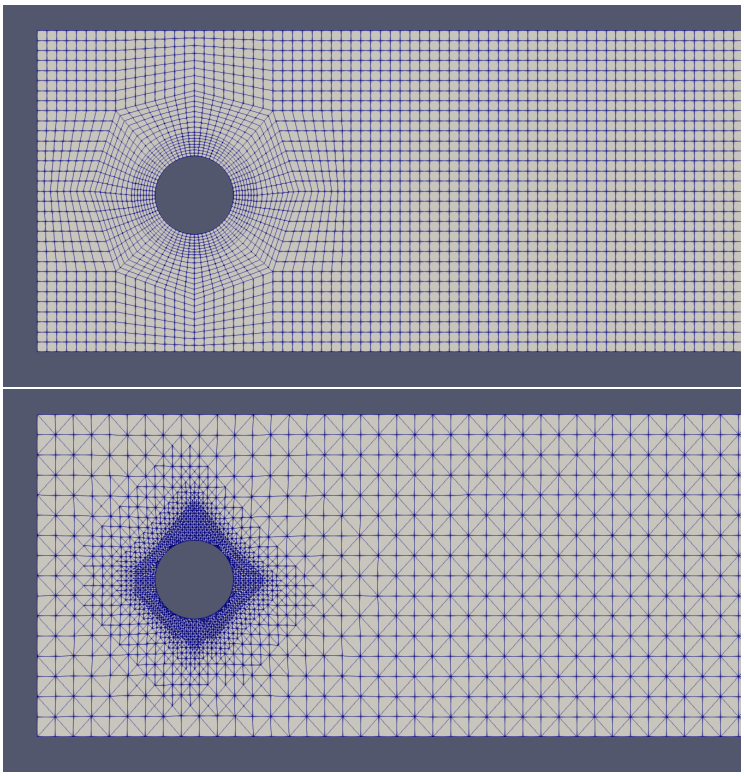

Figure 5: Comparison of the meshes for the channel flow with cylinder test case (around the cylinder). The grid generated with deal.II is shown in the first picture and uses quadrilateral elements whereas the second picture shows the triangular grid used in StormFlash. With a certain distance to the cylinders both grids are more or less uniform in both cases.

seems to be less diffused compared to the results from the StormFlash implementation. Nonetheless, the results from both implementations are very comparable which allows for the conclusion that both codes work reasonably well and can be validated by this experiment.

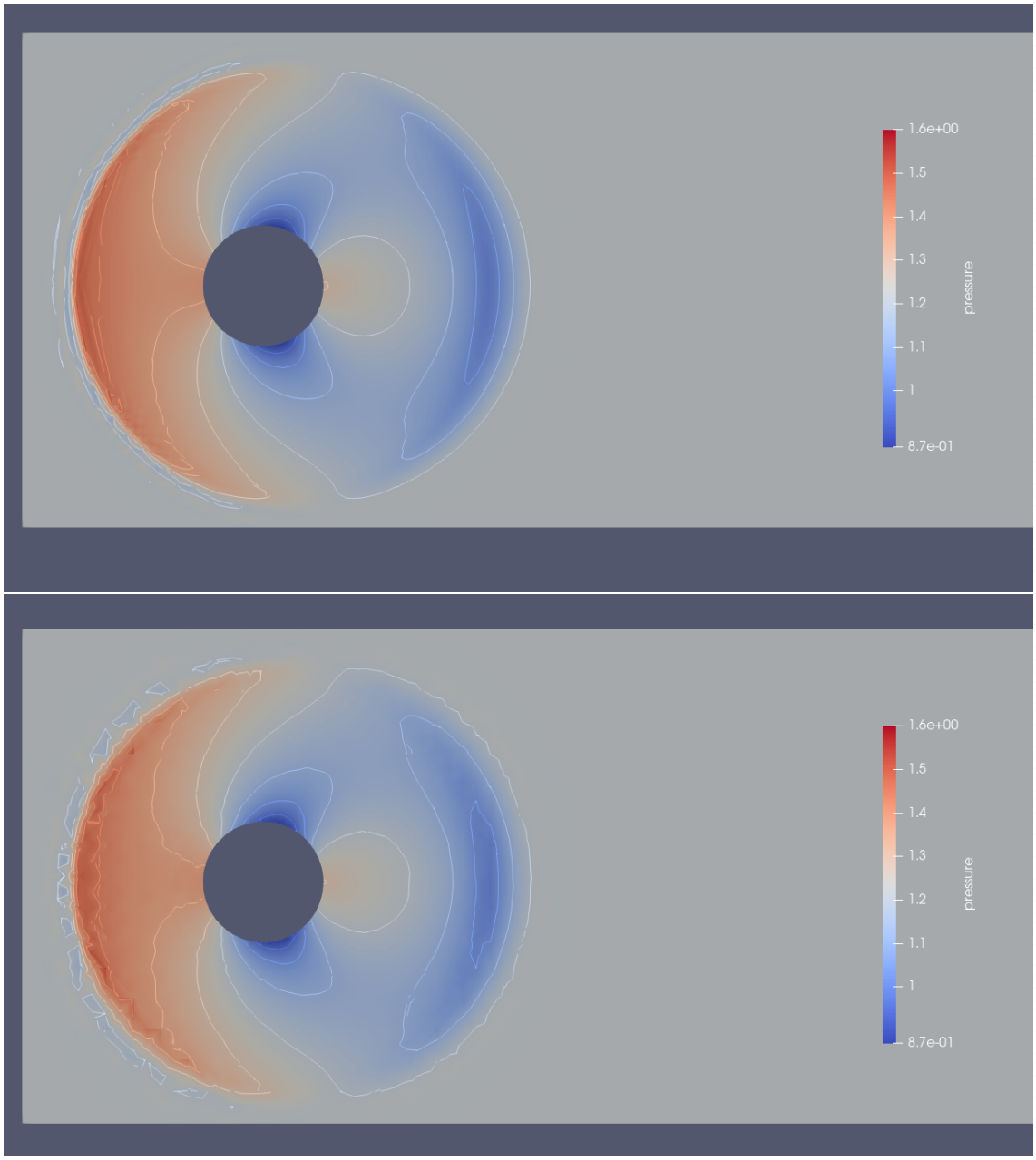

Figure 6: Comparison of the pressure profiles for the channel flow with cylinder test case at time $t = 0.1$ (showing contour lines in 10 steps between 0.87 and 1.6). The results for deal.II are shown on the top while the results for StormFlash are shown at the bottom. The development of the front of the acoustic wave shows a distinct difference since the resolution for StormFlash is coarser in that area. For reference, the solution from the deal.II tutorial (step-67) can be used.

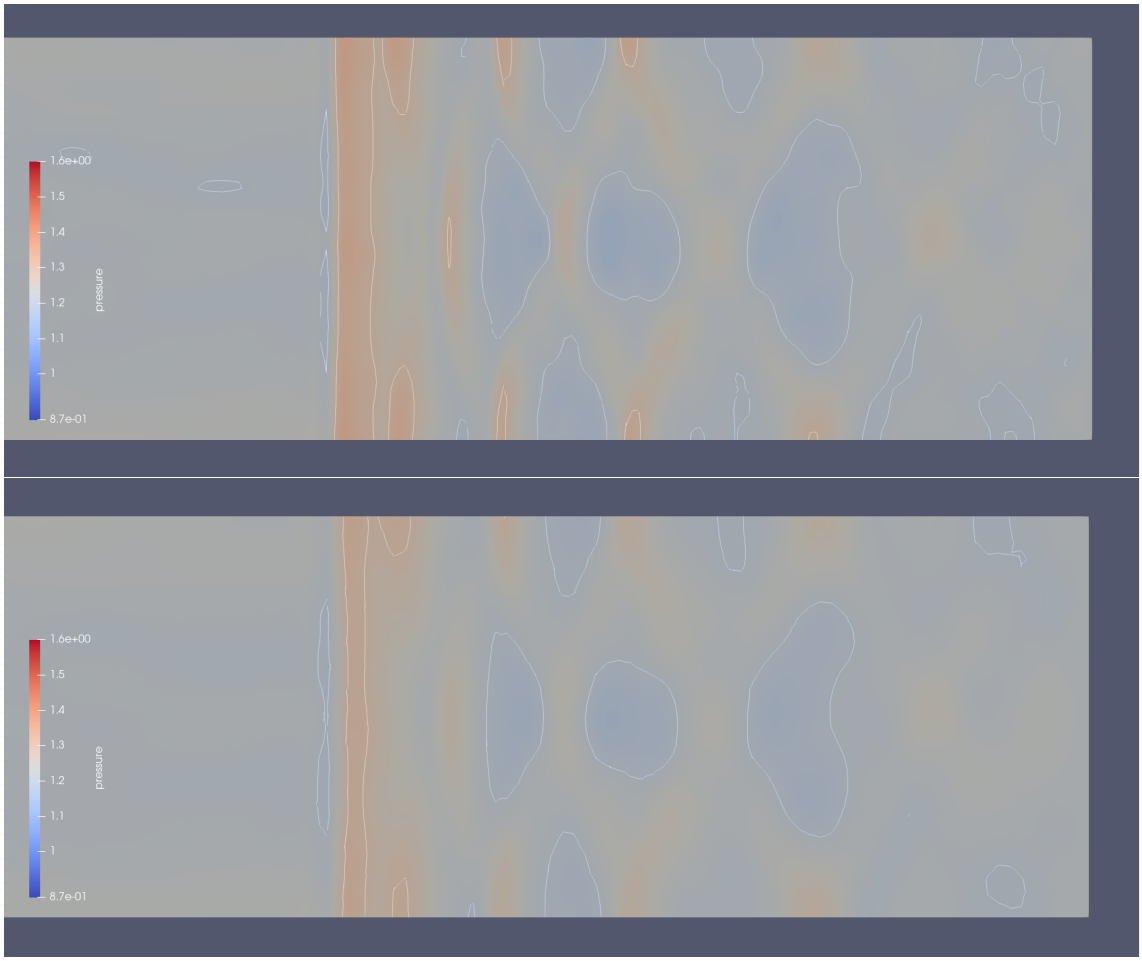

Figure 7: Comparison of the pressure profiles for the channel flow with cylinder test case at time $t = 2.0$ close to the outflow boundary (showing contour lines in 10 steps between 0.87 and 1.6). The results for deal.II are shown on the top while the results for StormFlash are shown at the bottom. The pressure profile for deal.II shows fronts that are more pronounced compared to StormFlash which is likely explained by a difference in spatial resolution.

6 Stabilization of the Euler equations with gravity source term

For the Euler equations, discrete (numerical) solvers for conservation laws suffer from the drawback that steady state solutions might not necessarily be preserved and that those solvers lead to spurious oscillations or otherwise incorrect results, in general (Li and Xing, 2016). Should a scheme, however, preserve steady states without the need for high spatial resolution close to machine precision - by using for example hydrostatic solutions - they are called well-balanced. Counter examples (what happens if a scheme is not well-balanced/stabilized) will be shown in later sections (see e.g. **Figure 13**).

For numerical solvers, problems arise as soon as the flux and gravity source term do not cancel in the hydrostatic case:

$$\nabla P \neq -\rho g \mathbf{k} = \nabla P^h,$$

where ∇P is the calculated pressure gradient for each degree of freedom and $-\rho g \mathbf{k} = \nabla P^h$ is the pressure gradient of a hydrostatic pressure. The hydrostatic relationship ($-\rho g \mathbf{k} = \nabla P^h$) can be obtained from the **Euler equations** - in particular the momentum equation - if $\mathbf{u} = 0$ is assumed.

While **Section 7** shows that not many of the schemes that are presented here produce viable results, all of these methods are shown because they were part of the PhD and should not be omitted.

6.1 SUPG and IDDG

One way to deal with spurious oscillations in compressible or incompressible flows (both viscous and inviscid flows) is to use the so-called streamline-upwind/Petrov-Galerkin (or SUPG in short) method (Aliabadi et al., 1993).
For details on this scheme, resources such as Aliabadi and Tezduyar (1993), Tezduyar (1992) and Kler et al. (2013) should be referred to. The basic idea is to apply a stabilization matrix to the flux term of the equation set which leads to an additional perturbation along streamlines which acts as an additional numerical diffusion. This stabilization matrix is composed of a part due to the advection term, a part term to account for shocks in the flow and a part that handles physical diffusion. While these schemes have shown to be effective for ensuring stable results for these flow problems, they are defined for Finite Element methods.

As a result, the need for a similar methods for the Discontinuous Galerkin method arose. The Implicit Diffusive Discontinuous Galerkin method (IDDG) provides an approach similar to that of the SUPG as it is basically a combination of implementing the idea of the

SUPG in a DG context with the utilization of a Runge-Kutta time stepping as shown in Calle et al. (2005a) or Calle et al. (2005b). As mentioned in these papers, the IDDG is alternative to having a slope limiter for preventing spurious oscillations as it provides local streamline diffusion.
As taken from Calle et al. (2005b), the IDDG replaces a traditional Galerkin method with the following (without source terms):

$$\int_{\Omega_e} \left(\frac{\partial \mathbf{q}_{N,e}}{\partial t} - \mathbf{F}_{N,e} \cdot \nabla \right) \psi_{i,e}(\mathbf{x}) \, \mathrm{d}\mathbf{x} - \int_{\Omega_e} \underbrace{\delta (\nabla \psi_{i,e} \cdot \beta) \nabla \cdot \mathbf{F}_{N,e}}_{\text{diffusion term}} \nabla \psi_{i,e}(\mathbf{x}) \, \mathrm{d}\mathbf{x} = - \int_{\Gamma_e} \psi_{i,e}(\mathbf{x}) \mathbf{n} \cdot \mathbf{F}_{N,e} \, \mathrm{d}S, \tag{29}$$

where δ is an artificial diffusion coefficient while β is a tensor/matrix diffusion term (for 2D or 3D). For example for 2D β is given by

$$\beta = (f_1' \tau, f_2') \tag{30}$$

with f_i' being the derivatives of the flux tensor and τ being

$$\tau = \left[\left| \frac{\partial \xi}{\partial x} f_1' + \frac{\partial \xi}{\partial y} f_2' \right| + \left| \frac{\partial \eta}{\partial x} f_1' + \frac{\partial \eta}{\partial y} f_2' \right| \right]^{-1} \tag{31}$$

As the name of the method suggests, the time stepping is done in an implicit manner where the mentioned papers suggest an implicit Euler scheme.

6.2 Filtering

Another way to ensure stability for numerical schemes is to make use of filtering techniques. One prominent filter is the so-called Boyd-Vandeven filter (Boyd, 1996) which spatially filters the results after each time step. This filter is for example applied in Giraldo and Restelli (2008).
After each completed time step, the filtering matrix $\mathcal{F}$ is applied to the numerical approximation $\mathbf{q}_N$ such that we receive the filtered solution $\mathbf{q}_N^F$:

$$\mathbf{q}_N^F = \mathcal{F} \mathbf{q}_N. \tag{32}$$

According to Giraldo and Restelli (2008), only the highest modes are reduced by 5%. The definition of the filter matrix can be found in Giraldo and Rosmond (2004).

6.3 Switch to equation sets with perturbed variables

The approach to switch to equation sets which are based on perturbed variables is not a way to stabilize the numerical scheme, per se, but it limits the incorrect behavior or oscillations that occur since the variables that are calculated have a much smaller magnitude. Giraldo and Restelli (2008) describes the way that this switch to perturbed variables - by

splitting up the variables - is done: for the **Euler equations**, the density, energy and pressure are split up in such a way that (for 2D) $\rho(\mathbf{x},t) = \overline{\rho(z)} + \rho'(\mathbf{x},t)$, $e(\mathbf{x},t) = \overline{e(z)} + e'(\mathbf{x},t)$ and $P(\mathbf{x},t) = \overline{P(z)} + P'(\mathbf{x},t)$ where $\overline{\mathbf{q}}$ describes mean values (for the respective variable) that are in hydrostatic balance and $\mathbf{q}'$ are the perturbed variables which also represent the deviation towards the hydrostatic background, for example. Using these variable splitting, we yield the following Euler equations:

$$\begin{aligned} \frac{\partial \rho'}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0 \\ \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot \left(\rho \mathbf{u} \otimes \mathbf{u} + P' I_d\right) &= -\rho' \nabla \phi \\ \frac{\partial \rho e'}{\partial t} + \nabla \cdot [(\rho e + P)\mathbf{u}] &= -\rho \mathbf{u} \cdot \nabla \phi. \end{aligned} \tag{33}$$

An advantage of this method is that it requires minimal change to the algorithm used for solving the Euler equations which makes it straight-forward to implement.

6.4 Stabilization via the use of the Navier-Stokes equations

In the literature, there are many sources (e.g. Müller et al., 2013, just to give one example) that rely on so-called "artificial viscosity" to add diffusion to ensure stability and to dampen or get rid of spurious oscillation. This basically leads to a change in equation set as the Navier-Stokes equations are solved - instead of the Euler equations. Still, this shows that using fluid flows where some form of viscosity is added can yield stable results. For the approaches in this thesis, there are two types of methods that were tried to implement ways to model the Navier-Stokes equations as to get stable flows: the Local Discontinuous Galerkin (LDG) method and the Interior Penalty (IP) method. The additional discretization of the elliptic term for the diffusion/viscosity increases the computational cost of the scheme.

As a note, the Navier-Stokes equations - especially in the case of compressible fluids - are notorious for the lack of analytical solutions. Consequently, code verification for the Navier-Stokes equations can be done by utilizing the Method of Manufactured Solutions (MMS). This method produces an analytical solution by including an analytical source term into the equation and using its derivatives for the flux terms, for example. As reference, Bond et al. (2007), Roache (2001) or Roy et al. (2004) can be used.

6.4.1 The Local Discontinuous Galerkin Method - LDG

The Local Discontinuous Galerkin Method (LDG) was proposed in Bassi and Rebay (1997) and is also used in Giraldo and Restelli (2008) to implement the Navier-Stokes equations. The elliptical term in the equation set has to be discretized, which is done by introducing auxiliary variables for the velocity and temperature gradient instead of directly approximating the Laplacian (as these auxiliary quantities are included in the calculation of the

Laplacian) for the viscous fluxes:

$$\mathcal{U} = \nabla \mathbf{u}, \ \mathcal{T} = \nabla T.$$

The auxiliary variables are then discretized by applying the *weak* form which leads to

$$\begin{aligned} \mathcal{U}_N &= -\int_{\Omega_e} \mathbf{u}_{N,e} \nabla \psi_{i,e}(\mathbf{x}) \ \mathrm{d}\mathbf{x} + \int_{\Gamma_e} \psi_{i,e}(\mathbf{x}) \mathbf{n} \cdot \mathbf{u}^* \ \mathrm{d}S, \\ \mathcal{T}_N &= -\int_{\Omega_e} T_{N,e} \nabla \psi_{i,e}(\mathbf{x}) \ \mathrm{d}\mathbf{x} + \int_{\Gamma_e} \psi_{i,e}(\mathbf{x}) \mathbf{n} \cdot T^* \ \mathrm{d}S, \end{aligned} \tag{34}$$

where $\mathcal{U}_N$ and $\mathcal{T}_N$ are the approximations for the auxiliary variables and $\mathbf{u}^*$ or T^* numerical fluxes for the velocity or temperature inside the element. While other numerical fluxes can be chosen, a very straight-forward approach is to use the average values of both sides of an interior face at the element boundary (similar as for the numerical flux in **Section 3.1**) so that

$$\mathbf{u}^* = \frac{\mathbf{u}^-_{N,e} + \mathbf{u}^+_{N,e}}{2}, \ \mathbf{T}^* = \frac{T^-_{N,e} + T^+_{N,e}}{2}.$$

With these variables, the viscous flux $\mathbf{F}^{\text{visc}}$ (with the separate fluxes from **Equation 4** and **Equation 5**) can be expressed as functions in terms of $\mathcal{U}$ and $\mathcal{T}$ with $\mathbf{F}^{\text{visc}} = \mathbf{F}^{\text{visc}}(\mathcal{U}, \mathcal{T})$. The Navier-Stokes equation can then be solved by using the *weak* form

$$\int_{\Omega_e} \left(\frac{\partial \mathbf{q}_{N,e}}{\partial t} - (\mathbf{F}_{N,e} - \mathbf{F}^{\text{visc}}_{N,e}(\mathcal{U}, \mathcal{T})) \cdot \nabla - \mathbf{S}_{N,e} \right) \psi_{i,e}(\mathbf{x}) \ \mathrm{d}\mathbf{x} = -\int_{\Gamma_e} \psi_{i,e}(\mathbf{x}) \mathbf{n} \cdot (\mathbf{F}^*_{N,e} - \mathbf{F}^{\text{visc}}_{N,e}(\mathcal{U}^*, \mathcal{T}^*)) \ \mathrm{d}S, \tag{35}$$

where the numerical fluxes $\mathcal{U}^*$ and $\mathcal{T}^*$ are chosen as the averages again:

$$\mathcal{U}^* = \frac{\mathcal{U}^-_{N,e} + \mathcal{U}^+_{N,e}}{2}, \ \mathcal{T}^* = \frac{\mathcal{T}^-_{N,e} + \mathcal{T}^+_{N,e}}{2}.$$

6.4.2 The Interior Penalty Method - IP

The Interior Penalty method (IP) that is presented here is a method that is similar to the LDG in the sense that the elliptical terms of the Navier-Stokes equations are expressed in terms of an auxiliary variable. The approach that is outlined here is taken from Fehn et al. (2019). For this strategy, let this auxiliary variable be the viscous flux such that

$$\mathcal{Q} = \mathbf{F}^{\text{visc}}(\mathbf{q}, \nabla \mathbf{q}).$$

Applying the usual Galerkin method with integrating the (auxiliary) variable yields the following discretization after rearranging and repeated integration-by-parts:

$$\int_{\Omega_e} \mathcal{Q}_{N,e} \chi_i(\mathbf{x}) \ \mathrm{d}\mathbf{x} = \int_{\Omega_e} \mathbf{F}^{\text{visc}}(\mathbf{q}_{N,e}, \nabla \mathbf{q}_{N,e}) \chi_i(\mathbf{x}) \ \mathrm{d}\mathbf{x} + \int_{\Gamma_e} \mathbf{F}^{\text{visc}}(\mathbf{q}_{N,e}, (\mathbf{q}^*_{N,e} - \mathbf{q}_{N,e}) \otimes \mathbf{n}) \mathbf{n} \chi_i(\mathbf{x}) \ \mathrm{d}S, \tag{36}$$

where χ_i is the basis function for the auxiliary variable, $\mathbf{q}^*_{N,e}$ is a numerical flux again and the tensor product has to be used for the second argument of the viscous flux for the edge integral. Using the *weak* form in conjunction with **Equation 36** for the viscous flux (similar to the LDG), as well as setting the basis function χ to $\chi = \nabla\psi_{i,e}$ leads to the Interior Penalty method:

$$\int_{\Omega_e} \left(\frac{\partial \mathbf{q}_{N,e}}{\partial t} - (\mathbf{F}_{N,e} - \mathbf{F}^{\text{visc}}(\mathbf{q}_{N,e}, \nabla\mathbf{q}_{N,e})) \cdot \nabla - \mathbf{S}_{N,e} \right) \psi_{i,e}(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$
$$= -\int_{\Gamma_e} \psi_{i,e}(\mathbf{x}) \mathbf{n} \cdot (\mathbf{F}^*_{N,e} - \mathcal{Q}^*) \, \mathrm{d}S \; - \int_{\Gamma_e} \mathbf{F}^{\text{visc}}(\mathbf{q}_{N,e}, (\mathbf{q}^*_{N,e} - \mathbf{q}_{N,e}) \otimes \mathbf{n}) \mathbf{n} \nabla\psi_{i,e}(\mathbf{x}) \, \mathrm{d}S. \quad (37)$$

The numerical fluxes $\mathbf{q}^*_{N,e}$ and $\mathcal{Q}^*$ are defined as:

$$\mathbf{q}^*_{N,e} = \frac{\mathbf{q}^-_{N,e} + \mathbf{q}^+_{N,e}}{2}$$
$$\mathcal{Q}^* = \frac{\mathbf{F}^{\text{visc}}(\mathbf{q}^-_{N,e}, \nabla\mathbf{q}^-_{N,e}) + \mathbf{F}^{\text{visc}}(\mathbf{q}^+_{N,e}, \nabla\mathbf{q}^+_{N,e})}{2} - \tau_{\text{IP}} I_d(\mathbf{q}^+ - \mathbf{q}^-)\mathbf{n}. \quad (38)$$

Here, τ_{IP} is the penalty parameter which is given by

$$\tau_{\text{IP}} = \nu(p+1)^2 \frac{A(\Gamma_e \backslash \Gamma_N)/2 + A(\Gamma_e \cap \Gamma_N)}{V(\Omega_e)}, \quad (39)$$

where A is the surface area/edge length, V is the volume, Γ_e is the element edge/face, Γ_N the current face or edge, Ω_e is the element and p is the (polynomial) order of the basis functions. $\nu = \mu/\rho_0$ is the kinematic viscosity which is obtained by dividing the dynamic viscosity μ by a reference density ρ_0 to obtain the correct units.

6.5 Specific well-balancing schemes for the Discontinuous Galerkin method

Since the well-balancing question for the Euler equations with gravity term is a research topic with high interest for (e.g.) atmospheric modeling, there are methods to ensure well-balanced solutions in the case of using Discontinuous Galerkin methods. The methods that will be presented in the following subsections are taken from Li and Xing (2016) and Chandrashekar and Zenk (2017). These methods have the big advantage of actually ensuring a well-balanced solution - so preserving steady states up to machine precision - while they have the drawback of having certain assumptions the flow needs to satisfy. The assumptions presented here are either isothermal flow (which is unrealistic for volcanic modeling) and polytropic flow (which is more realistic but not necessarily fulfilled for ash-gas mixtures).

In general, if a hydrostatic state is to be satisfied, the following equation holds:

$$\nabla P = -\rho\nabla\phi = -\rho g \mathbf{k}, \quad (40)$$

which can be obtained from the **Euler equations** and $\mathbf{u} = 0$ just as mentioned at the start of the stabilization section. The hydrostatic state is the basis for the following schemes. For the following schemes, only the source term approximation for the momentum equation will be presented. The source term for the energy equation is obtained by a scalar product of the velocity and the momentum source term approximation.

6.5.1 Isothermal flows

For this scheme, an ideal gas is assumed so that the pressure is given by **Equation 1** ($P = \rho RT$) as well as isothermy - such that $T = T_c = \text{const}$. Inserting the ideal gas law into **Equation 40** and integrating, the following relationship is obtained:

$$P \exp\left(\frac{\phi}{RT_c}\right) = \text{const}. \tag{41}$$

With this relation, the source term $\mathbf{S}_{\rho\mathbf{u}}$ (for the momentum) can be rewritten in terms of the ideal gas law and the quantities in **Equation 41**, such that

$$\mathbf{S}_{\rho\mathbf{u}} = -\rho\nabla\phi = \rho R\bar{T}_K \exp\left(\frac{\phi}{R\bar{T}_K}\right)\nabla \exp\left(-\frac{\phi}{R\bar{T}_K}\right), \tag{42}$$

where $\bar{T}_K$ is the cell averaged temperature in cell K. This source term can then be discretized like all quantities in the section about the **Discontinuous Galerkin Method**:

$$\mathbf{S}_{\rho\mathbf{u},N} = \rho_N R\bar{T}_K \exp\left(\frac{\phi(\mathbf{x})}{R\bar{T}_K}\right)\nabla \sum_i \exp\left(-\frac{\phi(\mathbf{x}_i)}{R\bar{T}_K}\right)\psi_i(\mathbf{x}). \tag{43}$$

As shown in the mentioned literature, if source terms like this are used instead, a steady state can be preserved for isothermal flows and the scheme is well-balanced.

6.5.2 Polytropic flows

Polytropic flows (a flow in a polytropic equilibrium) satisfy the condition that

$$P\rho^{-\gamma} = \alpha = \text{const}. \tag{44}$$

Similar to the isothermal case, this relation can be used and inserted into **Equation 40** which then yields:

$$\frac{\alpha\gamma}{\gamma-1}\rho^{\gamma-1}(\mathbf{x}) + \phi(\mathbf{x}) = \beta = \text{const}. \tag{45}$$

For this method, a function $H(\mathbf{x})$ is defined for each cell K with

$$H(\mathbf{x}) = \frac{\gamma}{\gamma-1}\ln\left(\frac{\gamma-1}{\alpha_K\gamma}(\beta_K - \phi(\mathbf{x}))\right), \tag{46}$$

where the constants α_K and β_K have to be chosen. This function $H(\mathbf{x})$ can be used to rewrite the source term (for the momentum) as

$$\mathbf{S}_{\rho\mathbf{u}} = -\rho\nabla\phi = \frac{\gamma-1}{\gamma}\rho(\beta_K - \phi(\mathbf{x}))\exp(-H(\mathbf{x}))\nabla\exp(H(\mathbf{x})). \tag{47}$$

Again, similar to the isothermal case, the source term (for the momentum) is discretized with the DGM in such a way that:

$$\mathbf{S}_{\rho\mathbf{u},N} = \frac{\gamma-1}{\gamma}\rho_N(\beta_K - \phi(\mathbf{x}))\exp(-H(\mathbf{x}))\nabla\sum_i \exp(H(\mathbf{x}_i))\psi_i(\mathbf{x}). \tag{48}$$

The parameters α_K and β_K are chosen in such a way as to chose each node where the polytropic condition is maximized inside each cell (for that cell node the notation i^* is used):

$$\beta_K = \max_i\left[\frac{\gamma}{\gamma-1}\frac{P_i}{\rho_i} + \phi(\mathbf{x}_i)\right],\ \alpha_K = P_{i^*}\rho_{i^*}^{-\gamma}. \tag{49}$$

Again, as shown in the literature, this scheme is well-balanced for polytropic flows.

6.6 General linear low-effort stabilization scheme

In contrast to the methods that were already presented, the goal of this general linear low-effort stabilization scheme is to find a numerical approach to stabilize the Euler equations with gravity source term with the use of a low order (linear elements) DGM. The advantage that this scheme has is that it is implemented in a rather straight-forward way since no assumptions to the flow have to be made - unlike the two methods presented in the previous sections. That means that the flow does not have to be polytropic or isothermal to achieve stable results. Unfortunately, the method, as it is implemented right now, has the drawback of not preserving hydrostatic solutions up to machine precision. Hence it is only a stabilization method and not a well-balanced scheme.
We follow the approach as in Bänsch et al. (2022). This general stabilization scheme is similar to Blaise et al. (2016) in a way, as the polynomial order of the source term approximation is reduced by one.
This general linear stabilization scheme is not a well-balanced scheme. Nonetheless, the approach on how well-balancing would be achieved will be presented.

This well-balancing is obtained by exchanging the conventional source term in the **Euler equations** (i.e. $-\rho g\mathbf{k}$) with a new source term which is a quantity that is akin to a hydrostatic pressure. If the *weak* form is used, the source term can be replaced with:

$$\mathbf{S}_N = \nabla\cdot\mathcal{P}, \tag{50}$$

where $\mathcal{P}$ is comprised of $\left(0, P^hI_2, (P^hI_2)\cdot\mathbf{u}\right)^\top$ and $\nabla\cdot\mathcal{P} = (0, \nabla\cdot(P^hI_2),\ \nabla\cdot(P^hI_2)\cdot\mathbf{u})^\top$, where P^h is the quantity that represents a hydrostatic pressure (reconstruction). Following this approach, the polynomial order of the source term (the hydrostatic pressure gradient) is identical to the order of the pressure gradient. This is where the similarity to Blaise et al. (2016) comes into play.

If the DG discretization is applied to **Equation 50**, we get:

$$-\int_{\Omega_e} \mathbf{S}_{N,e}\psi_{i,e}(\mathbf{x})\ \mathrm{d}\mathbf{x} = -\int_{\Omega_e} \nabla\cdot\mathcal{P}\psi_{i,e}(\mathbf{x})\ \mathrm{d}\mathbf{x} = \int_{\Omega_e} \mathcal{P}\nabla\psi_{i,e}(\mathbf{x})\ \mathrm{d}\mathbf{x} - \int_{\Gamma_e} \psi_{i,e}(\mathbf{x})\mathbf{n}\cdot\mathcal{P}^*\ \mathrm{d}S. \quad (51)$$

Using **Equation 51** and inserting it into the *weak* form yields

$$\int_{\Omega_e} \left(\frac{\partial \mathbf{q}_{N,e}}{\partial t} - (\mathbf{F}_{N,e} - \mathcal{P})\cdot\nabla\right)\psi_{i,e}(\mathbf{x})\ \mathrm{d}\mathbf{x} = \int_{\Gamma_e} \psi_{i,e}(\mathbf{x})\mathbf{n}\cdot(\mathcal{P}^* - \mathbf{F}^*_{N,e})\ \mathrm{d}S.$$

(well-balanced *weak* form)

To achieve this well-balancing, the hydrostatic pressure as well as the numerical flux for that hydrostatic pressure (so the numerical hydrostatic flux) $\mathcal{P}^*$ have to be calculated.

6.6.1 Pressure reconstruction

The hydrostatic pressure has to satisfy

$$P^h = (\gamma-1)(\rho e)^h, \quad (52)$$

where $(\rho e)^h$ is the hydrostatic total energy which is obtained from the energy equation (**Equation 2**) when $\mathbf{u} = 0$, as well as the hydrostatic relationship as defined at the very beginning of this section of stabilization of the Euler equations:

$$\frac{\partial P^h}{\partial z} = -\rho g. \quad (53)$$

Here, only the vertical derivative (in z-direction) is considered since the gravitational potential is defined as non-zero only in the vertical direction for the setups in this thesis. Inserting **Equation 52** into **Equation 53** yields

$$(\gamma-1)\frac{\partial(\rho e)^h}{\partial z} = -\rho g. \quad (54)$$

Rearranging and integrating (similar to the mean value theorem for integrals) allows to solve for the hydrostatic energy:

$$(\rho e)^h = \overline{\rho e} - \frac{g\overline{\rho}}{\gamma-1}(z-\overline{z}), \quad (55)$$

where the quantities with bars are the respective element-averages.
This hydrostatic energy representation is quickly calculated for every element which makes it straightforward. Advantageously, the derivation of the (linear) hydrostatic pressure profile now also becomes rather simple:

$$P^h = (\gamma-1)\overline{\rho e} - g\overline{\rho}(z-\overline{z}). \quad (56)$$

Finally, we can define the numerical flux for the hydrostatic pressure $\mathcal{P}^*$. It is defined as:

$$\mathcal{P}^* = \mathbf{F}^*\left((\rho_L, \mathbf{u}_L = 0, P^h_L/(\gamma-1))^\top, (\rho_R, \mathbf{u}_R = 0, P^h_R/(\gamma-1))^\top\right), \quad (57)$$

where we either can, for example, use the Rusanov flux (see **Equation 11**) or a straight-forward approach of a mean value similar to the numerical fluxes for the LDG or IP.

In theory, the pressure reconstruction (along with the numerical flux) preserves hydrostatic initial conditions. While hydrostatic initial conditions are preserved, perturbations of balanced states are not transported correctly, however. This problem was encountered by implementing the second test case from Chandrashekar and Zenk (2017) where a pressure perturbation is added. The results do not differ that much from the initial conditions (which is incorrect).

This raises the question whether a different approach - most likely for the numerical flux - can be found. Instead, linear (hydrostatic) profiles for all variables could be calculated as in Botta et al. (2004). For this method, these profiles could then be used for calculating the numerical flux.
If $\mathcal{Q}$ would be the hydrostatic variables, then the numerical flux should be $\mathcal{P}^* = \mathbf{F}^*(\mathcal{Q}_L, \mathcal{Q}_R)$. However, it should be noted that this approach would result in an additional assumption to the flow field since the initial condition would have to be polytropic (i.e. $P/\rho^\gamma = \text{const.}$) and that the velocity is not set to zero by default in the arguments for the numerical flux. Obviously, this would remove one mentioned advantage for this novel approach.

6.6.2 Stabilization scheme instead of well-balanced method

Instead of another a costly method for calculating the numerical flux for the hydrostatic pressure, we can also follow a very straight-forward approach by setting $\mathcal{P}^* = 0$.
The obvious drawback is that with this approach the well-balancedness of the method is lost. Still, this approach for the method yields stable results (as will be shown in the following sections). To differentiate between the well-balanced method and the choice of $\mathcal{P}^* = 0$, the method with prescribed numerical flux will be called stabilized scheme from now on. Since setting the numerical flux for the hydrostatic pressure to zero leads to a stabilized method, this supports the claim in the previous subsection that different approaches for calculating the numerical flux could lead to a well-balanced method. For correct well-balancing properties, further research is needed but was not done in the scope of this thesis due to time constraints.

Table 4: Overview for the source term approximations

Trivial source term	well-balanced source term	stabilized source term
$\mathbf{S}_N = (0, -\rho g \mathbf{k}, -\rho g \mathbf{k} \cdot \mathbf{u})^\top$	$\mathbf{S}_N = \nabla \cdot \mathcal{P}$, $\mathcal{P}^*$ included	$\mathbf{S}_N = \nabla \cdot \mathcal{P}$, $\mathcal{P}^* = 0$
	with hydrostatic pressure reconstruction	

7 Results for the stabilization problem

In this section, the results (if any usable ones were produced) of the respective methods to stabilize the Euler equations with gravity source term will be presented. In this scope, mainly two test cases will be used: an atmosphere at rest test case which is taken from Benacchio (2014) that is used to verify the stability and order of the scheme and the prominent rising warm air bubble test case (taken from Giraldo and Restelli, 2008). Other test cases will be defined in the sections where they are used. The main test cases are defined in the following two subsections:

7.1 Neutrally stratified atmosphere after Benacchio (2014)

This test case sets up an atmosphere that is neutrally stratified and satisfies hydrostacy. As a result, it serves as a basis to check whether a scheme or an approach presented in the previous section preserves hydrostatic states.
The neutrally stratified, homentropic atmosphere (i.e. constant potential temperature) is implemented as in Benacchio (2014) which has the following initial conditions:

$$\begin{aligned} P(z) &= P_{\text{ref}} \left(1 - \Gamma \frac{g\rho_{\text{ref}}}{P_{\text{ref}}} z\right)^{\frac{1}{\Gamma}}, \\ \rho(z) &= \rho_{\text{ref}} \left(\frac{P(z)}{P_{\text{ref}}}\right)^{\frac{1}{\gamma}}, \\ \rho_{\text{ref}} &= \frac{P_{\text{ref}}}{RT_{\text{ref}}}, \end{aligned} \tag{58}$$

where $\Gamma = (\gamma - 1)/\gamma$ and $R = 287.17$ J kg^{-1} K^{-1} is the gas constant for dry air. $P_{\text{ref}} = 10^5$ Pa is the reference pressure and $T_{\text{ref}} = 300$ K the reference temperature for this atmosphere. The velocities are initialized to be zero. The calculation domain is set to be $\Omega = [0\text{ m}, 10000\text{ m}]^2$ with 33, 65 and 129 nodes in each direction, respectively, which leads to spatial resolutions of $\Delta x = \Delta z = \Delta$ with $\Delta_1 = 312.5$ m, $\Delta_2 = 156.25$ m and $\Delta_3 = 78.125$ m.
The simulation is run for 3600 simulated seconds ($T_{\text{max}} = 3600$ s) where the time step sizes are set up to be $\Delta t_1 = 0.0675$ s, $\Delta t_2 = 0.03375$ s and $\Delta t_3 = 0.016875$ s (which correspond to a CFL number of about 0.15 with the corresponding spatial resolution Δ_i).
The boundaries are set up to be inflow/Dirichlet boundaries.
After T_{max} is reached, the errors can be calculated (in regards to the initial condition) which is done as defined in **Equation 26** and **Equation 27**, respectively. For the momentum, however, instead of the relative error, the absolute error has to be calculated since the initial momentum is zero which does not allow for a relative error analysis. Consequently, the relative errors will only be calculated for density and energy.
Ideally, the initial condition should be preserved (up to machine precision).

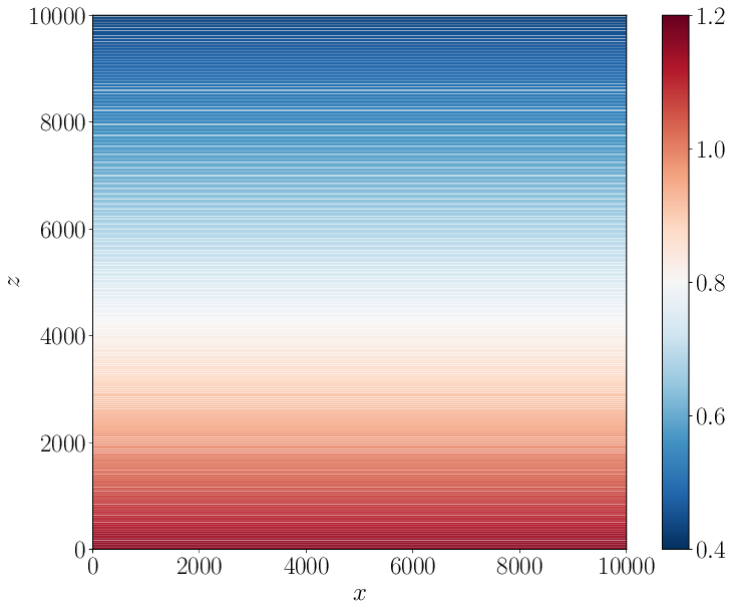

Figure 8: Initial density for the neutrally stratified atmosphere. The values range from 1.1608 kg m^{-3} at the bottom to 0.4342 kg m^{-3} at the top.

In addition to this rather straight-forward approach, a setup where the lower right corner of the domain will be removed will also be shown for StormFlash with stabilization and deal.II. To be more precise, a triangle will be "cut out" starting at $\mathbf{x} = (5000, 0)$ m and ending at $\mathbf{x} = (0, 5000)$ m from the initial domain of $\Omega = [0 \text{ m}, 10000 \text{ m}]^2$. For deal.II this change in domain also leads to a rearranged mesh due to its quadrilateral structure. This leads to the spatial resolution Δ_i being just an approximate resolution in this case.
The time step sizes will be adjusted for the three different spatial resolutions such that $\Delta t_1 = 0.0675$ s, $\Delta t_2 = 0.03375$ s and $\Delta t_3 = 0.016875$ s. All other parameters remain the same.

The goal of this setup is to see whether more complex geometries influence the results.

7.2 Rising warm air bubble as in Giraldo and Restelli (2008)

The rising warm air bubble test case is a prominent test case first introduced by Robert (1993). In this thesis however, the definition of the test case from Giraldo and Restelli (2008) is taken. Here, an atmosphere with constant potential temperature θ_{ref} (which is similar to the previous test case with $\theta_{\text{ref}} = 300$ K) is perturbed by a warm air bubble which is introduced with a cosine-profile. This allows to model the evolution and rise of this warm air bubble.
Similar to the previous test case (neutrally stratified atmosphere), the energy and density are inferred from the potential temperature and pressure profile, while no background velocity is prescribed (hence initial x- and z-momentum are zero).

The potential temperature θ is initialized as:

$$\theta = \begin{cases} \theta_{\text{ref}} & r > r_c, \\ \theta_{\text{ref}} + \dfrac{\theta_c}{2}\left(1 + \cos\left(\dfrac{\pi r}{r_c}\right)\right) & r \leq r_c, \end{cases}$$

where θ_{ref} is the background or reference potential temperature which is set to 300 K. $\theta_c = 0.5$ K is the maximum potential bubble temperature, the distance from the center of the bubble is given by $r = \sqrt{(x - x_c)^2 + (z - z_c)^2}$. The center of the bubble is located at $(x_c, z_c) = (500, 350)$ m. $r_c = 250$ m is the bubble radius and the calculation domain is defined as $\Omega = [0\text{ m}, 1000\text{ m}]^2$ with no-flux/wall boundaries.

The so-called Exner pressure can be calculated with the use of the potential temperature which allows for the calculation of the other variables so that density, pressure and consequently also the energy can be inferred:

$$\begin{aligned} \Pi &= 1 - \frac{gz}{c_p \theta_{\text{ref}}}, \\ \rho &= \frac{P_{\text{ref}}}{R\theta} \Pi^{\frac{1}{\gamma - 1}}, \\ P &= P_{\text{ref}} \left(\frac{R\rho\theta}{P_{\text{ref}}}\right)^{\gamma}, \end{aligned}$$

where $c_p = R\gamma/(\gamma - 1)$ is the specific heat capacity at constant pressure. γ, R and P_{ref} are defined as in the neutrally stratified atmosphere test case. The test case is modeled for 700 s. As for the previous test case, the spatial resolution is set to be equal in horizontal and vertical direction so that $\Delta x = \Delta z = \Delta$. If not otherwise specified, the resolution is set to $\Delta = 15.625$ m with a time step size of $\Delta t = 0.004$ s while the reference solution is shown for a spatial resolution of 20 m (but with 10^{th} order polynomials and filtering).

7.3 Without any stabilization

In this section, the results for StormFlash and deal.II without the use of stabilization will be shown.

7.3.1 StormFlash - with and without slope limiter

For the case of the trivial source term where the source term is just discretized by applying the usual DG discretization (without any stabilization), the results with StormFlash become unstable.

In this subsection, results from before and after the code restructuring for StormFlash will be shown. This should be taken into account because the simulation post-restructure for the finest spatial resolution ($\Delta_3 = 78.125$ m) is stopped after $t = 14$ minutes due to

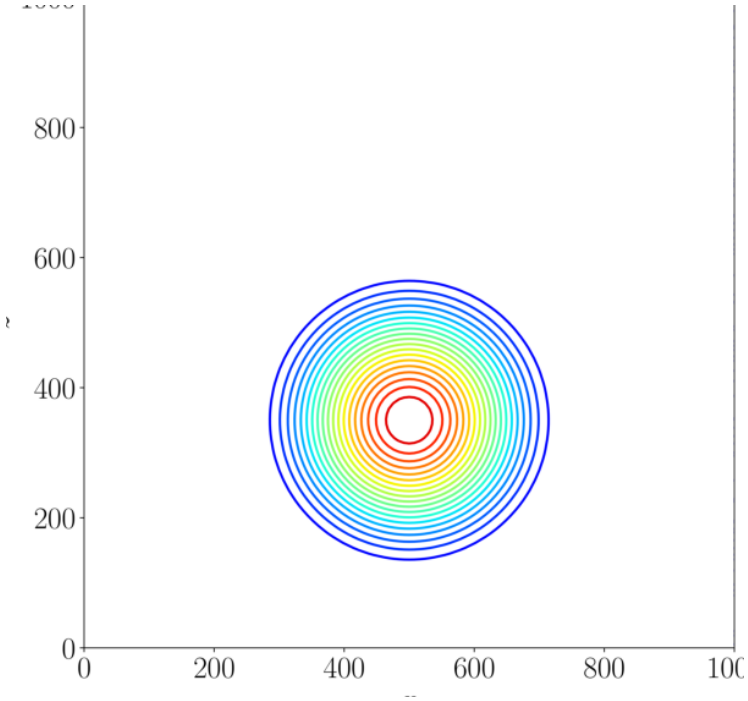

Figure 9: Initial potential temperature perturbation for the rising warm air bubble. The contour lines range from 0 K to 0.525 K with interval sizes of 0.025 K.

instabilities.

In addition to the trivial source term, the simulation is also run with a slope limiter to check whether the application of a limiter can stabilize the trivial source term. While the results with limiter vary significantly (compared with the non-limited case), they still are unstable.

The instability and as a result the scheme's divergence (if the trivial source term is used) is clearly observable in **Figure 10**, **Figure 11** and **Figure 12**.

Figure 10 shows a comparison between limited and non-limited solution after 24 minutes in 6 minute increments until the maximum simulation time (60 minutes) is reached with a spatial resolution of $\Delta_1 = 312.5$ m. Neither approach yields stable results which is very clearly visible as the results should be visually indistinguishable when compared to **Figure 8** if the scheme would be second order accurate.

Figure 11 shows the results for all three different spatial resolutions after $t = 34$ minutes. When looking at this figure, the plots show behavior reminiscent of Kelvin-Helmholtz instabilities which is physically unrealistic for a neutrally stratified atmosphere that satisfies the hydrostatic property. It is observable that the smaller scales of higher spatial resolutions drastically influence the results as they appear to model flows that look very turbulent.

Figure 12 shows the L_∞ errors for the density and vertical momentum as examples for the errors for StormFlash with the trivial source term. It becomes evident that the scheme

Table 5: Errors and rates of convergence for the neutrally stratified atmosphere test case for StormFlash without the stabilization scheme. Note that the momentum errors are absolute errors and not relative errors and that the results for Δ = 78.125 m are shown after t = 840 s because the calculation was interrupted afterwards (due to instability). As a result, the rates of convergence for Δ_3 cannot be shown for this resolution. These results were obtained post-StormFlash restructure.

L_2 error after T_{max} with	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
Δ_1 = 312.500 m	9.213e-02	-	4.172e+03	-	3.526e+03	-	4.681e-02	-
Δ_2 = 156.250 m	1.025e-01	-0.15	1.276e+04	-1.61	8.913e+03	-1.34	6.521e-02	-0.48
Δ_3 = 78.125 m	1.561e-01	-	2.138e+04	-	2.015e+04	-	7.449e-02	-
L_∞ error after T_{max} with	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
Δ_1 = 312.500 m	2.503e-01	-	1.334e+02	-	9.600e+01	-	8.272e-02	-
Δ_2 = 156.250 m	4.063e-01	-0.70	2.137e+02	-0.68	1.249e+02	-0.38	1.712e-01	-1.05
Δ_3 = 78.125 m	1.246	-	2.205e+02	-	2.013e+02	-	1.701e-01	-

is not consistent as the error does not get smaller with an increase in spatial resolution. The errors even oscillate very strongly in time for each respective spatial resolution. For the finest resolution Δ_3 = 78.125 m, the simulation is stopped after after t = 14 minutes, as previously mentioned. This leads to the error plot for that resolution having fewer data points.

Additionally to the figures, **Table 5** (which shows the errors and convergence rates post-code restructure) supports the claim that the scheme is divergent as the errors for the velocities are beyond reason for a scheme that would be stable. It should be noted that calculating the rates of convergence for a scheme that divergences is unreasonable but the rates are shown nonetheless to further the claim that the scheme is nowhere close to being second order accurate if the trivial source term is chosen.

For the simulations of the rising warm air bubble, the results (shown in **Figure 13**) are also physically unreasonable as they are nowhere close to the reference solution from Giraldo and Restelli (2008). The contour lines are chosen to be the same as in **Figure 9** to allow for better comparison and makes it obvious that StormFlash with trivial source term leads to physically unrealistic behavior as the scheme is unstable. Consequently, StormFlash cannot be used for simulations where the gravity term is implemented trivially.

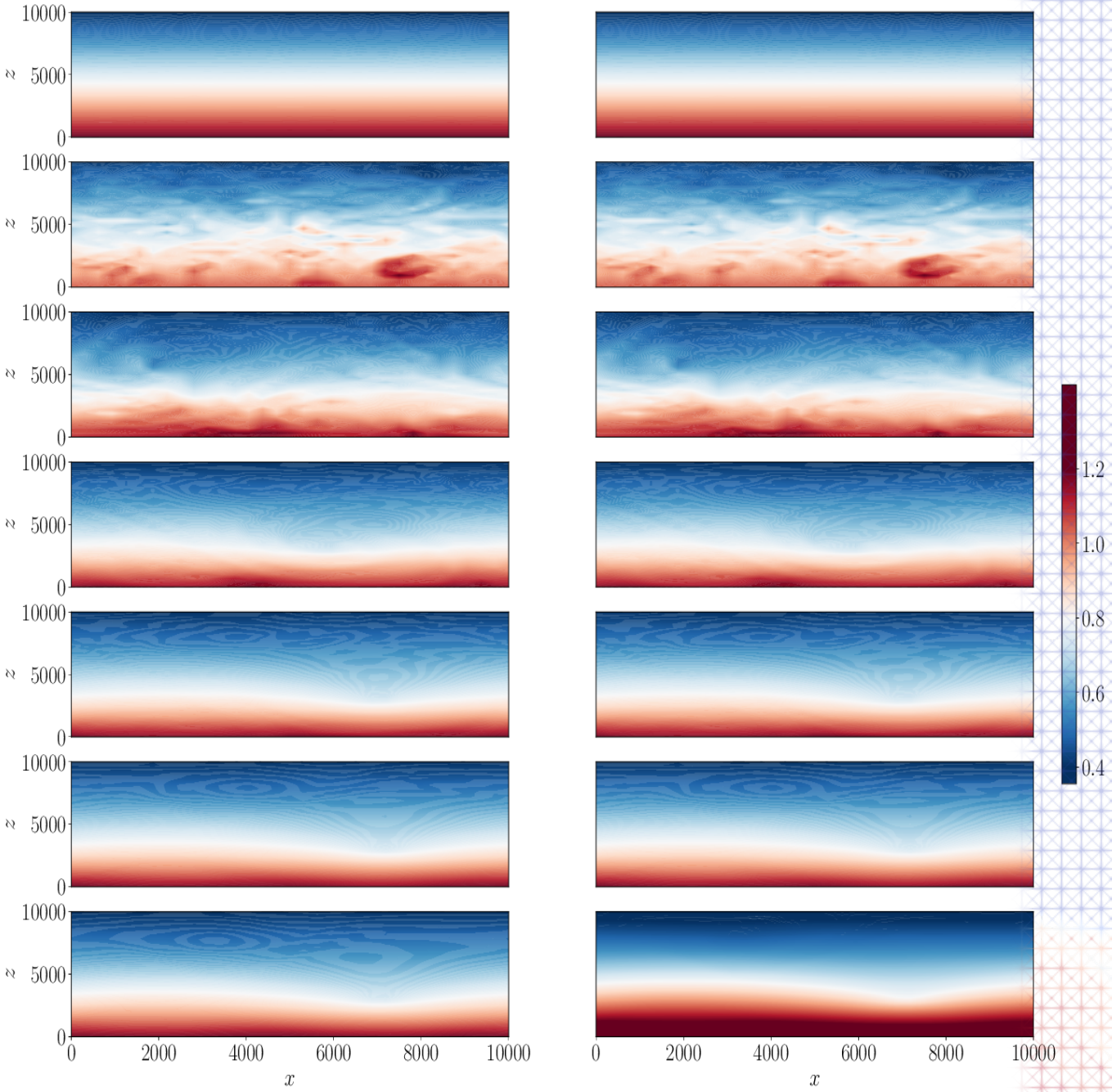

Figure 10: Comparison of non-limited (left) and limited density results (right) for the neutrally stratified atmosphere with the trivial (unstable) source term. The results are shown after 24, 30, 36, 42, 48, 54 and 60 minutes from top to bottom for $\Delta = \Delta_1 = 312.5$ m. The maximum and minimum densities are exceeded in the colormap which was chosen in this way to allow for better comparison with the initial condition in **Figure 8**. These results were obtained pre-StormFlash restructure.

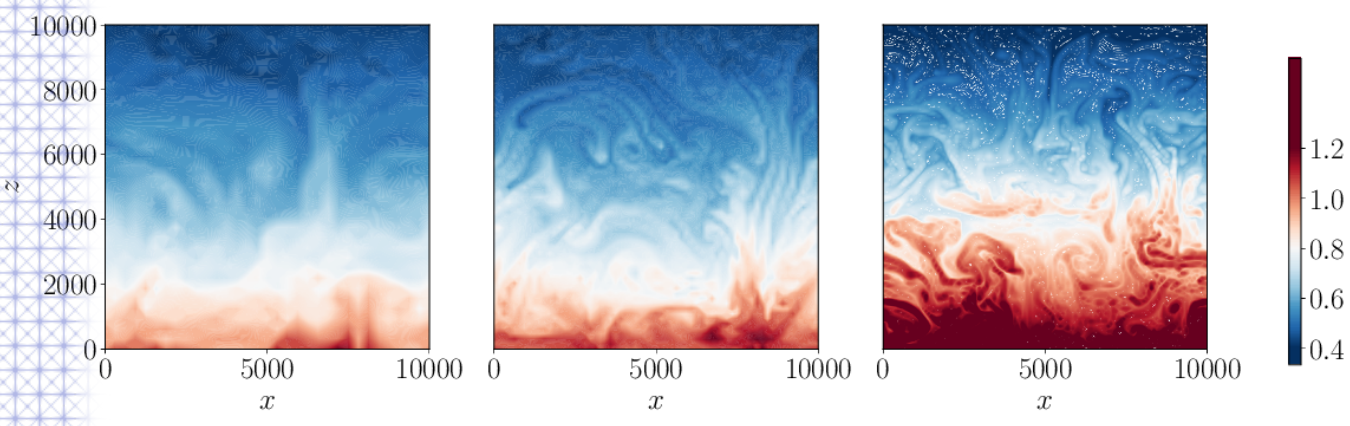

Figure 11: Results for the density after $t = 34$ min for different resolutions for the neutrally stratified atmosphere with the trivial (unstable) source term. The left plot shows the result for $\Delta_1 = 312.5$ m, the plot in the middle for $\Delta_2 = 156.25$ m and the right plot for $\Delta_3 = 78.125$ m. The higher the resolution, the more refined small scale instabilities can occur which leads to the observable structures. These results were obtained pre-StormFlash restructure.

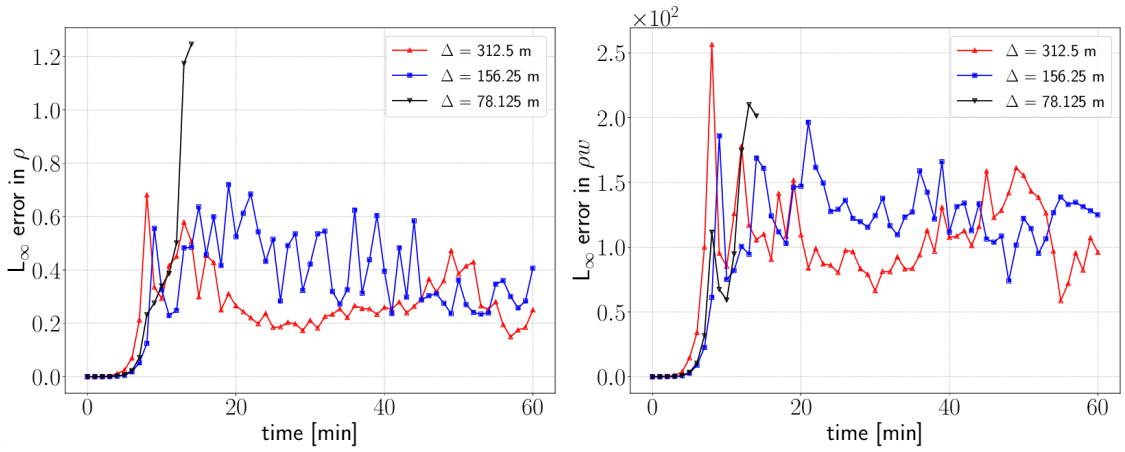

Figure 12: L_∞ errors for the density (relative error) and momentum in vertical direction (absolute error) for StormFlash with the trivial source term. On the left-hand side, the error plot for the density is shown while on the right-hand side the error plot for the vertical momentum is presented. For the finest spatial resolution, the simulation is interrupted (due to instability) after 14 minutes. These results were obtained post-StormFlash restructure.

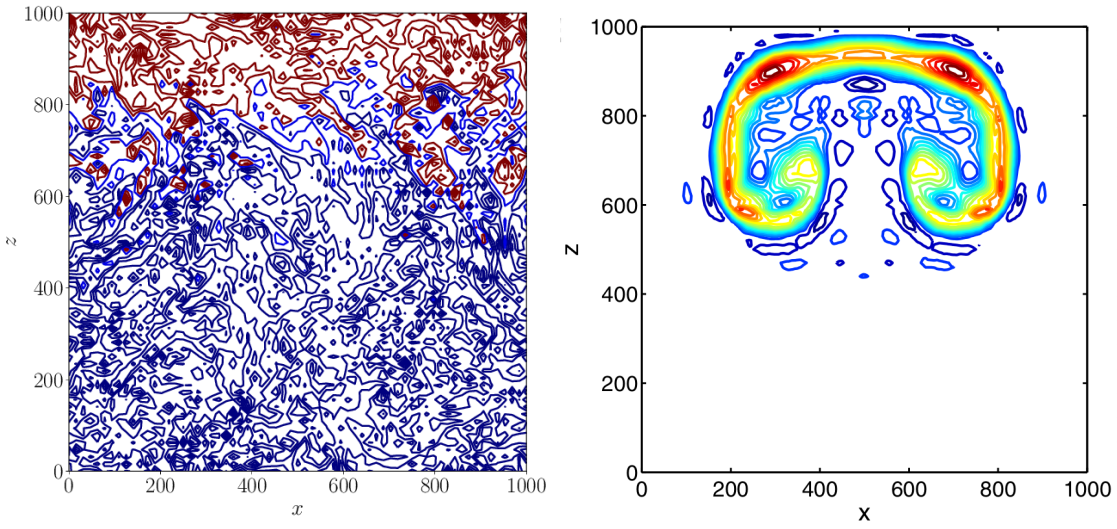

Figure 13: Results for the rising warm air bubble obtained from StormFlash with the trivial source term on the left and a reference solution from Giraldo and Restelli (2008) on the right. The contour lines are chosen to be same as for the initial condition in **Figure 9** (and Giraldo and Restelli, 2008) to allow for a better comparison. The contour lines for StormFlash without stabilization do not resemble the reference solution at all.

7.3.2 deal.II

The results that will be shown for deal.II are done without any stabilization approach. While the attempt was made to implement the stabilization scheme, due to time constraints, it had to be abandoned since more pressing matters had to receive the focus of the work (i.e. the implementation of the plume model itself).

That being said, this section will present the results for the test cases that were obtained using the implementation in deal.II. For the neutrally stratified atmosphere test case, **Figure 14** shows the L_∞ errors for density and vertical momentum (as examples) for the duration of the simulation. The figure suggests that the error - while decreasing with higher spatial resolution (as is expected) - in density increases the longer the simulation is run. This behavior is only observed for the density. The errors for all other variables either remain stable or even decrease with time (plots not shown for the other variables). **Table 6** shows the errors and rates of convergence for this test case after T_{max}. The table shows that the rates of convergence do not allow us to conclude that deal.II is second order accurate for a setup with gravity source term - at least when looking at the L_2 error since the scheme is second order accurate for the L_∞ error.

The results for the neutrally stratified atmosphere with a slope inside the domain yield similar results. **Figure 15** again shows errors for density and vertical momentum. Here, the errors again increase the longer the simulation is run for. For the coarsest spatial

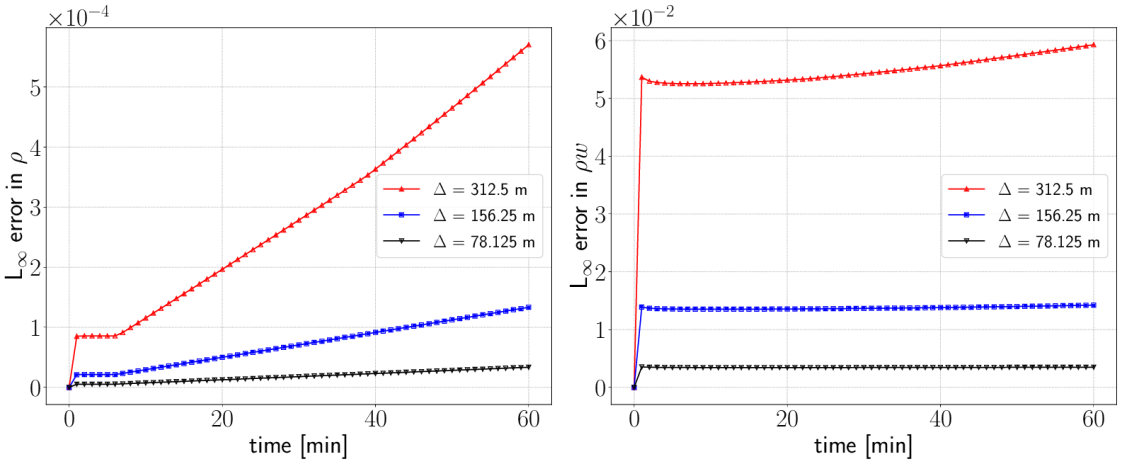

Figure 14: L_∞ errors for the density (relative error) and momentum in vertical direction (absolute error) for deal.II for the neutrally stratified atmosphere. The plot on the left-hand side shows the errors for density, the right-hand side for the momentum. For the density, the errors increase with time while a higher resolution leads to smaller errors for both cases.

resolution the error curves do not follow a clear pattern and do not exert predictable behavior. **Table 7** shows the numerical values for the errors and convergence rates after T_{max}. The values again suggest that the scheme is not second order accurate - this time both for the L_2 and L_∞ error. Hence, a change in geometry seems to influence the results for deal.II.

Finally, the results for the rising warm air bubble from deal.II are presented in **Figure 16**. As is observable in this figure, deal.II produces results that do not fully resemble the reference solution from Giraldo and Restelli (2008). If this observation is combined with the knowledge from the previous test case, it is very likely that the different behavior is due to the scheme not being second order accurate for the use of linear elements which leads to unexpected behavior and causes errors.

7.4 Stabilization scheme

The stabilization scheme was implemented in StormFlash as outlined in **Section 6.6**. Just as in the previous sections, the neutrally stratified atmosphere, as well as rising warm air bubble test case were run using this setup. In addition to the square domain setup, as in the section about deal.II, the atmosphere test case was also run with a domain with slope. All the parameters for the test cases are the same as presented at the start of this result section.

Figure 17 shows the L_∞ errors for the density and vertical momentum for neutrally

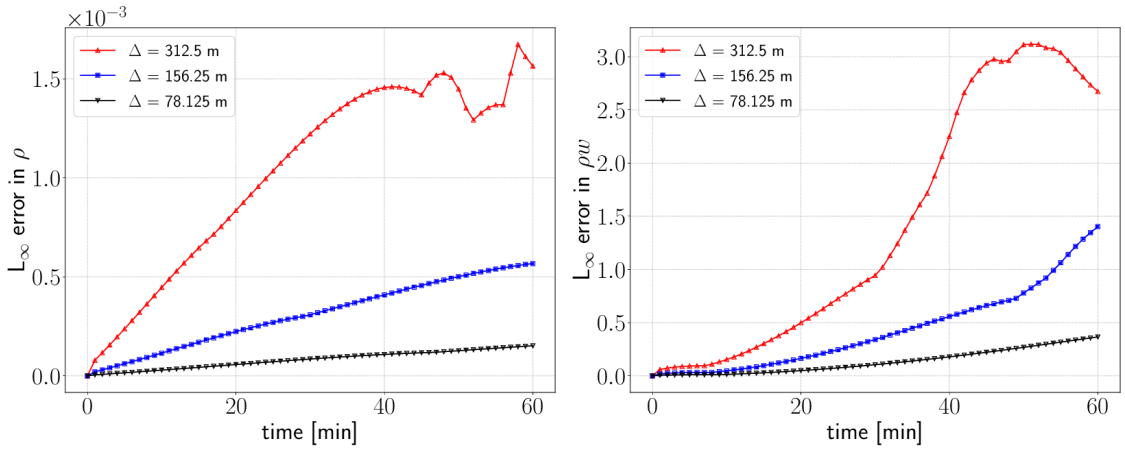

Figure 15: L_∞ errors for the density (relative error) and momentum in vertical direction (absolute error) for deal.II for the neutrally stratified atmosphere case with slope. The plot on the left-hand side shows the errors for density, the right-hand side for the momentum. In both cases, the errors increase with time while a higher resolution leads to smaller errors. For the coarsest resolution the errors do not follow a clear pattern for both density and momentum.

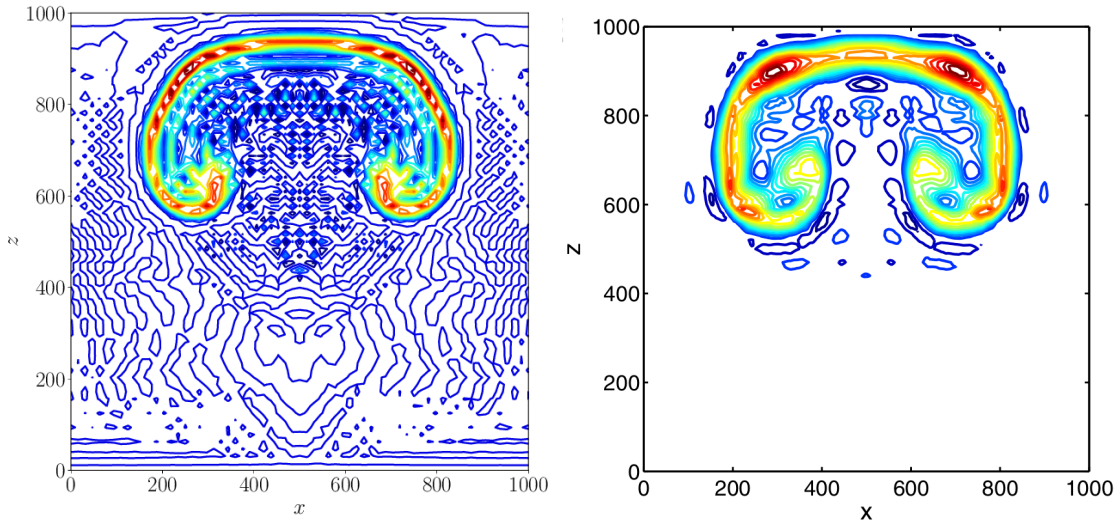

Figure 16: Results for the rising warm air bubble obtained from deal.II (left) and a reference solution from Giraldo and Restelli (2008) on the right. The contour lines are chosen to be the same as for the initial condition in **Figure 9** (and Giraldo and Restelli, 2008) to allow for a better comparison. The results from deal.II resemble the ones from the reference solution but the curvature of the eddies on the outside of the bubble differ compared to the reference solution.

Table 6: Errors and rates of convergence for the atmosphere at rest test case for deal.II. Note that the momentum errors are absolute errors and not relative errors.

$\mathbf{L_2}$ **error after** $T_{\mathbf{max}}$ **with**	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
Δ_1 = 312.500 m	1.596e-04	-	2.724e-01	-	2.356	-	3.952e-05	-
Δ_2 = 156.250 m	2.624e-05	2.61	1.529e-01	0.83	1.186	0.99	1.086e-05	1.86
Δ_3 = 78.125 m	4.712e-06	2.48	8.118e-02	0.91	5.959e-01	0.99	2.890e-06	1.91
$\mathbf{L_\infty}$ **error after** $T_{\mathbf{max}}$ **with**	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
Δ_1 = 312.500 m	5.703e-04	-	1.615e-02	-	5.927e-02	-	7.791e-05	-
Δ_2 = 156.250 m	1.333e-04	2.10	4.114e-03	1.97	1.423e-02	2.06	2.073e-05	1.91
Δ_3 = 78.125 m	3.353e-05	1.99	1.053e-03	1.97	3.492e-03	2.03	5.366e-06	1.95

Table 7: Errors and rates of convergence for the atmosphere at rest test case for deal.II with slope. Note that the momentum errors are absolute errors and not relative errors. Δ is the approximate resolution for this case.

$\mathbf{L_2}$ **error after** $T_{\mathbf{max}}$ **with**	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
$\Delta_1 \approx$ 312.500 m	4.634e-04	-	38.505	-	48.582	-	6.602e-05	-
$\Delta_2 \approx$ 156.250 m	8.573e-05	2.32	18.828	1.43	23.712	1.43	1.083e-05	2.47
$\Delta_3 \approx$ 78.125 m	1.426e-05	2.45	10.014	1.37	7.924	1.73	2.059e-06	2.29
$\mathbf{L_\infty}$ **error after** $T_{\mathbf{max}}$ **with**	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
$\Delta_1 \approx$ 312.500 m	1.565e-03	-	1.942	-	2.674	-	2.469e-04	-
$\Delta_2 \approx$ 156.250 m	5.675e-04	1.66	9.041e-01	1.47	1.402	1.38	7.763e-05	1.85
$\Delta_3 \approx$ 78.125 m	1.521e-04	1.93	5.354e-01	1.30	3.657e-01	1.96	1.511e-05	2.27

stratified atmosphere test case (which are shown as examples). The figure shows that the errors remain constant throughout the simulation. As is expected of a consistent scheme, the error decreases with increase in spatial resolution. **Table 8** shows the corresponding errors after $T_{\max}$. Interestingly, convergence rate for the L_2 error for both momenta is of order 1 instead of 2 while it is of order 2 for density and energy. Additionally, all convergence rates for the L_∞ are of order 2. Similar to Blaise et al. (2016), a reduction in order of the source term approximation stabilizes or balances the Euler equations which is similar to the approach for the stabilization scheme utilized in StormFlash. This however, does not explain the rate of convergence of 1 for the L_2 error while the order remains 2 in the case of the L_∞ error.

In the atmosphere test case with slope, the results also have an increased rate of convergence for the L_2 error of the momenta (about 1.6 instead of 1.0; see **Table 9**). **Figure 19**

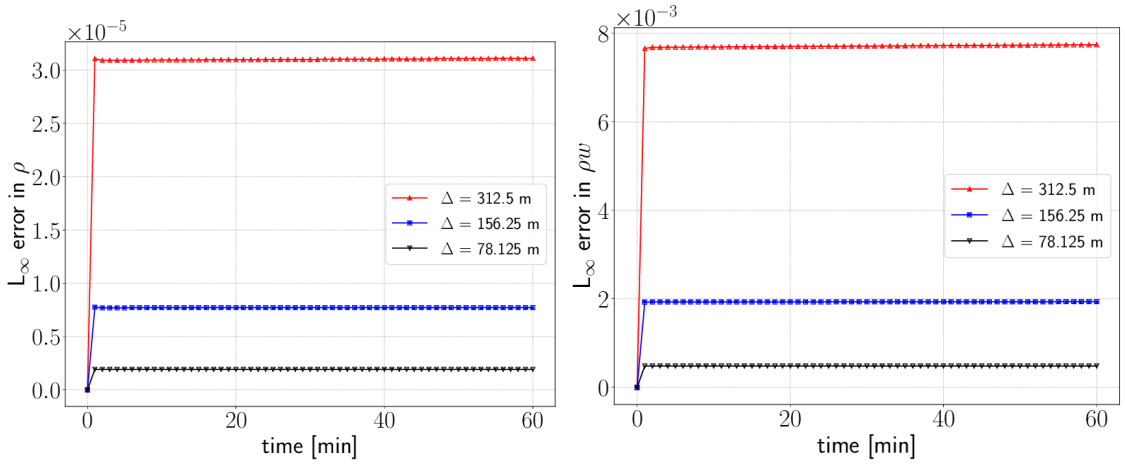

Figure 17: L_∞ errors for the density (relative error) and momentum in vertical direction (absolute error) for the neutrally stratified atmosphere case for StormFlash with stabilization. The plot on the left-hand side shows the errors for density, the right-hand side for the momentum. In both cases, the errors remain constant throughout the simulation while a higher resolution leads to smaller errors.

shows the same errors as **Figure 17** but for the sloped case. In this case, the errors increase with time which is probably due to the more complex geometry or rather the more complex (Dirichlet) boundaries.
Furthermore, **Figure 18** shows a snap shot of the vertical momentum which clearly shows the imprint of the triangular mesh. This imprint is caused by the calculation of the linear hydrostatic pressure reconstruction and might be related to the issue regarding the rate of convergence for both horizontal and vertical momentum. It has to be noted that further research into this matter is needed.

Regarding the rising warm air bubble test case, the results for StormFlash with stabilization are presented in **Figure 20**. For this case, the results resemble the reference solution quite well with an exception of the eddy structure "inside" of the bubble where the eddies with StormFlash reach higher into the inside. But especially if compared to the results obtained with deal.II (**Figure 16**), StormFlash yields results that are more accurate (when compared to the reference solution).
While there still is an open question regarding the order of the scheme when it comes to the momentum, the stabilization scheme is at least able to model the physical behavior of the rising warm air bubble - and consequently fluid flow - reasonably well.

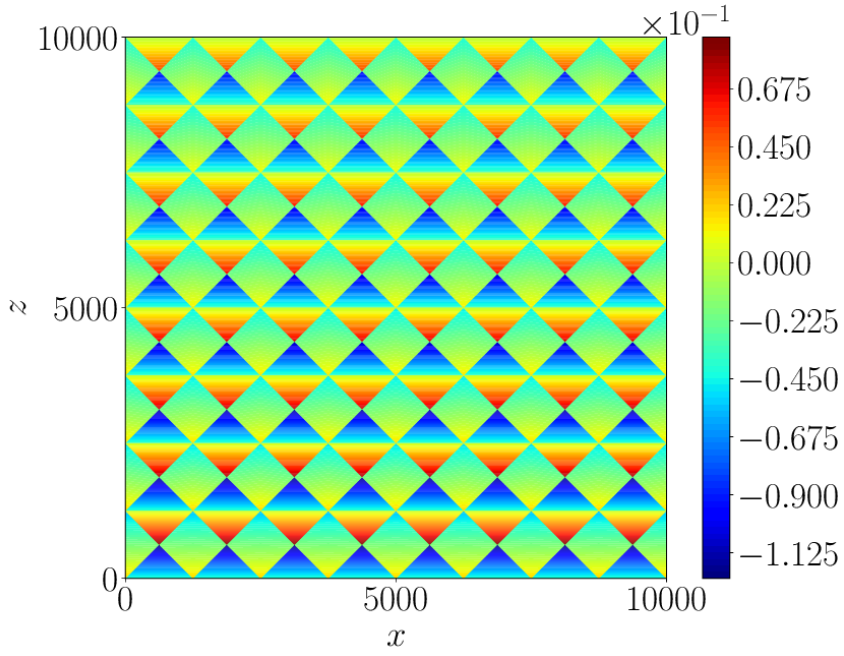

Figure 18: Vertical momentum for the stabilized StormFlash for the atmosphere at rest test case (which is the same for all time steps after $t = 0$ s). For demonstration purposes, the results for a resolution of $\Delta = 1250$ m are presented but the structure remains the same for all finer resolutions with the exception that the magnitude of error in momentum decreases with finer grids. A grid imprint from the triangular mesh is clearly visible. The imprint is caused by the linear hydrostatic pressure reconstruction.

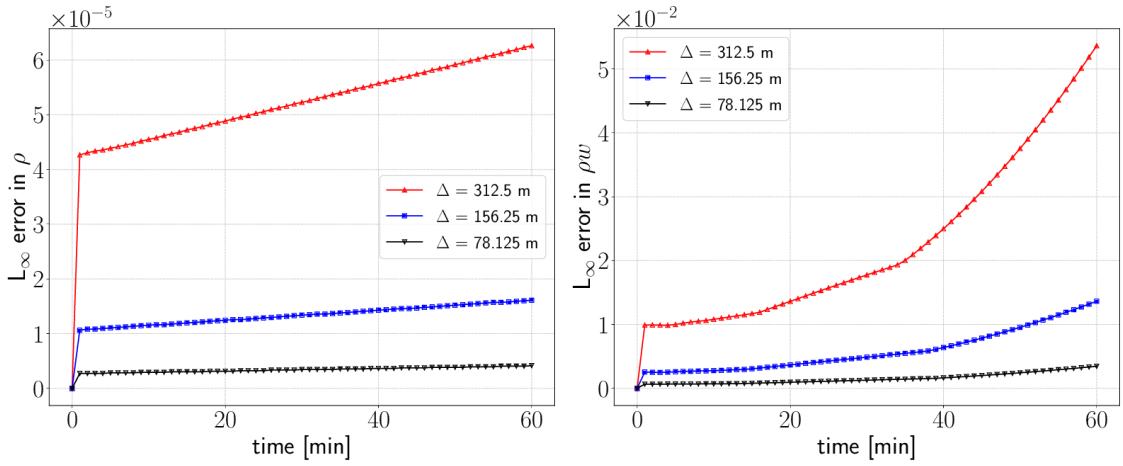

Figure 19: L_∞ errors for the density (relative error) and momentum in vertical direction (absolute error) for the neutrally stratified atmosphere case with slope for StormFlash with stabilization. The plot on the left-hand side shows the errors for density, the right-hand side for the momentum. In both cases, the errors increase with time while a higher resolution leads to smaller errors.

Table 8: Errors and rates of convergence for the atmosphere at rest test case for StormFlash with stabilization. Note that the momentum errors are absolute errors and not relative errors.

$\mathbf{L_2}$ **error after** $T_{\mathbf{max}}$ **with**	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
Δ_1 = 312.500 m	2.254e-05	-	2.924e-01	-	3.200e-01	-	3.578e-05	-
Δ_2 = 156.250 m	6.117e-06	1.88	1.463e-01	1.00	1.595e-01	1.00	8.949e-06	2.00
Δ_3 = 78.125 m	1.601e-06	1.93	7.320e-02	1.00	7.957e-02	1.00	2.239e-06	2.00
$\mathbf{L_\infty}$ **error after** $T_{\mathbf{max}}$ **with**	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
Δ_1 = 312.500 m	3.110e-05	-	6.584e-03	-	7.744e-03	-	4.256e-05	-
Δ_2 = 156.250 m	7.724e-06	2.00	1.650e-03	2.00	1.938e-03	2.00	1.073e-05	1.99
Δ_3 = 78.125 m	1.925e-06	2.00	4.139e-04	2.00	4.847e-04	2.00	2.714e-06	1.98

Table 9: Errors and rates of convergence for the atmosphere at rest test case for StormFlash with stabilization and slope. Note that the momentum errors are absolute errors and not relative errors.

$\mathbf{L_2}$ **error after** $T_{\mathbf{max}}$ **with**	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
Δ_1 = 312.500 m	3.226e-05	-	7.495e-01	-	8.613e-01	-	4.717e-05	-
Δ_2 = 156.250 m	8.374e-06	1.96	2.941e-01	1.60	3.610e-01	1.54	1.180e-05	2.00
Δ_3 = 78.125 m	2.147e-06	1.97	1.202e-01	1.56	1.482e-01	1.56	2.951e-06	2.00
$\mathbf{L_\infty}$ **error after** $T_{\mathbf{max}}$ **with**	ρ	σ_ρ	ρu	$\sigma_{\rho u}$	ρw	$\sigma_{\rho w}$	ρe	$\sigma_{\rho e}$
Δ_1 = 312.500 m	6.260e-05	-	5.010e-02	-	5.363e-02	-	5.835e-05	-
Δ_2 = 156.250 m	1.611e-05	1.97	1.274e-01	1.98	1.364e-02	1.98	1.474e-05	1.99
Δ_3 = 78.125 m	4.153e-06	1.97	3.229e-02	1.99	3.459e-03	1.99	3.701e-06	2.00

7.5 Methods that did not work or did not produce any significant results

7.5.1 Slope limiter in combination with the stabilization scheme

To see whether the slope limiter improves the results for the stabilization scheme, the slope limiter was utilized for the neutrally stratified atmosphere test case as well. Initially, the idea was to present a difference plot to show the error between the use of a slope limiter and the approach without but in fact, the results with the slope limiter in conjunction with the stabilization scheme are identical to simulations with just the stabilization scheme. Consequently, the slope limiter will also not be used further in this thesis. Additionally, no difference plot will be shown.

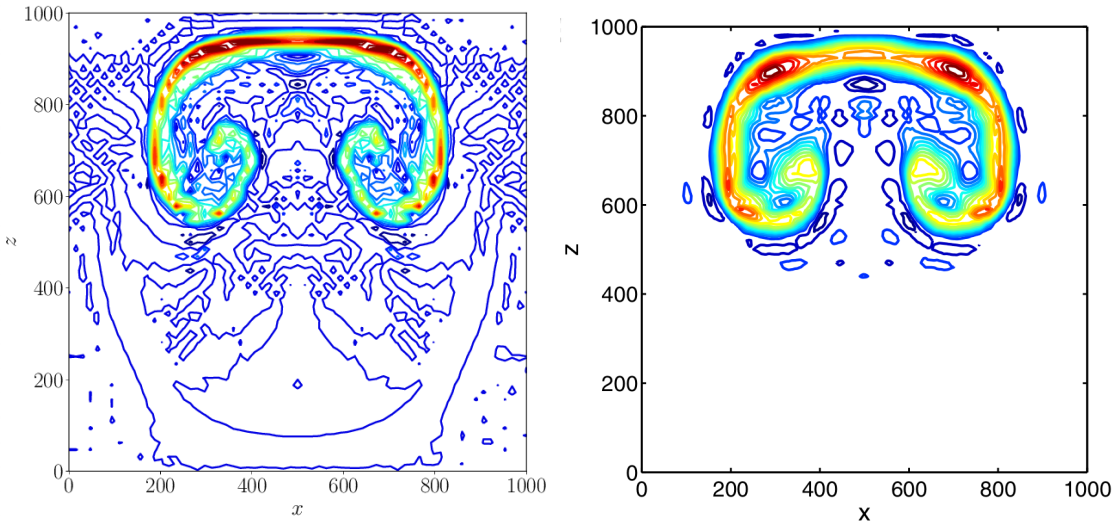

Figure 20: Results for the rising warm air bubble obtained from StormFlash with stabilization (left) and a reference solution from Giraldo and Restelli (2008) on the right. The contour lines are chosen to be the same as for the initial condition in **Figure 9** (and Giraldo and Restelli, 2008) to allow for a better comparison. The results from StormFlash resemble the ones from the reference solution with the eddies on the "inside" of the bubble differing from the reference solution a little.

7.5.2 IDDG

Unfortunately, implementing the IDDG was unsuccessful and only led to erroneous or unstable simulations. As a result, nothing from those simulations will be shown here.

7.5.3 Filtering

The idea for the filtering approach was to use a Boyd-Vandeven filter using StormFlash for runs on the rising warm air bubble to dampen the oscillations that occur. Unfortunately, the filter could not be implemented correctly in StormFlash. The work that would have been necessary did not warrant the time investment for this method, however, so after the attempts to implement the filter, the idea was dismissed.

7.5.4 Background/perturbation equation set

For the Euler equations with background or perturbation approach, the initial idea was the same as for the results with slope limiter in addition to the stabilization scheme where a difference should be plotted. However, just as with the slope limiter approach, the difference between the different equation sets was zero for which reason no results will be shown and this approach to solving the Euler equations will also not be used any further.

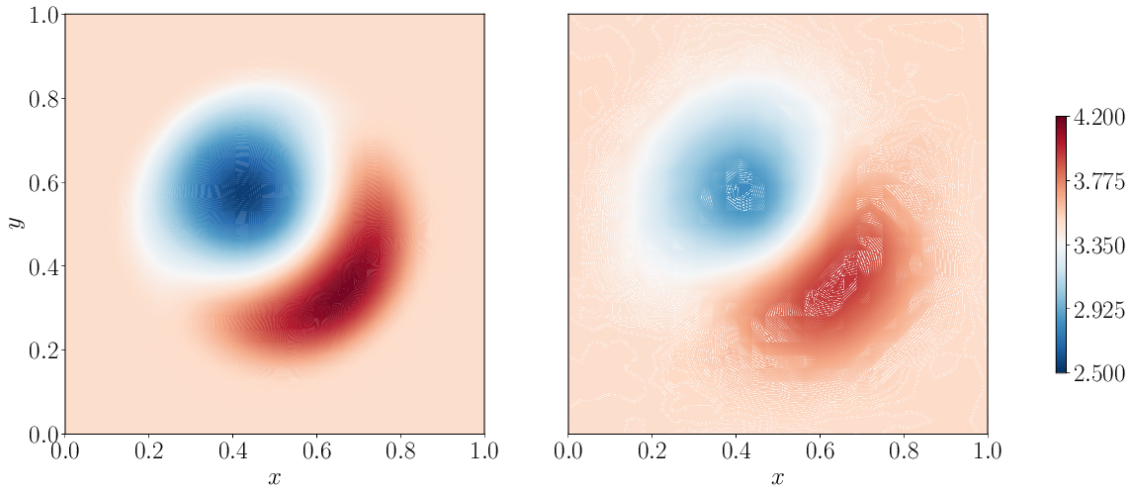

Figure 21: Results for the quasi-stationary vortex with a viscosity of $\mu = 0.01$ Pa s. The figure shows the initial energy on the left and the results for the energy after one period ($T_{\max} = 1$ s) on the right. Both energies are shown in J. The results after one second are diffused compared to the initial values for energy.

7.5.5 Navier-Stokes equations

The Navier-Stokes equations were implemented in StormFlash using the LDG method and seemed to work reasonably well. While the results look promising visually, a verification via the MMS was inconclusive - probably due to incorrect implementations.
To showcase the visual reasonableness, the quasi-stationary vortex test case (from **Section 5.1**) was used. **Figure 21** shows the results for this vortex test case where a viscosity of $\mu = 0.01$ Pa s was used. The results after one period are clearly more diffused which is the reason why the argument was made that the results look visually promising.

Additionally to the quasi-stationary vortex test case, results for the rising warm air bubble were calculated with the implementation for the Navier-Stokes equations. For this, a viscosity of $\mu = 0.1$ Pa s was used. The results are presented in **Figure 22**, where the rising warm air bubble test case was run with spatial resolutions of $\Delta x = \Delta z = \Delta = 7.8125$ m and $\Delta x = \Delta z = \Delta = 3.90625$ m, respectively, which are compared with reference solutions from Giraldo and Restelli (2008) with spatial resolutions of 10 m and 5 m. With a viscosity of $\mu = 0.1$ Pa s, the results for StormFlash do not quite resemble the reference solution (both in curvature of the warm air bubble from the center at the front of the bubble and the eddy structure for the run with a resolution of $\Delta = 3.90625$ m) but the plots suggest that utilizing viscous flux terms both stabilizes and removes oscillations from the flow. Were it not for the code restructure of StormFlash, runs with smaller viscosities could have been performed which might have led to better results.

While the exact reason is unknown, the code restructure for StormFlash led to problems

with the Navier-Stokes implementation. Consequently, afterwards the results looked very differently to what was calculated beforehand. As an example, **Figure 23** shows the results that were obtained for a run of the rising warm air bubble with a spatial resolution of $\Delta x = \Delta z = \Delta = 31.25$ m. If the general structure of the results is compared with those of **Figure 22**, it becomes evident that at the code restructure of StormFlash caused some issues since the latest results show an asymmetric structure which is not physically plausible. This was one of the reasons why the further development of the plume model with StormFlash as basis was abandoned and a switch to deal.II was made.

In the scope of using deal.II for a plume model, the effort was made to implement both an LDG, as well as an IP method approach using the deal.II step-67 tutorial as basis. Unfortunately, the implementation was unsuccessful and then also abandoned in favor of implementing a plume model based on the Euler equations (see **Section 8**).

7.5.6 Polytropic/isothermal well-balancing schemes

While it was already mentioned in the section where the models were presented, using an isothermal approach for well-balancing is not really realistic for the goal of modeling volcanic plumes. Still, the attempt to implement this well-balancing approach was made, in addition to the attempt to implement the polytropic well-balancing scheme.
Unfortunately, neither of the methods could be implemented correctly. In both cases, the simulations became unstable and did not yield results. The attempts were made for StormFlash in 2D and a simpler approach with a 1D python model which is very similar to StormFlash.
Since no usable results were produced, no results can be presented here.

7.5.7 Stabilization scheme with well-balanced source term

The attempt to correctly well-balance the Euler equations was the first focus of this thesis (and the one which was focused on most as well).
For this, the part that was researched the most was the edge integral part of the well-balanced source term approach. Within StormFlash, the neutrally stratified atmosphere test case was used as basis to analyze the numerical flux for the hydrostatic pressure reconstruction. With the numerical flux being calculated as in **Equation 57** the results could not be improved much more than with $\mathcal{P}^* = 0$ which is one of the reasons why this approach was used for the stabilization method.

As a second test case, the first 1D hydrostatic test case from Chandrashekar and Zenk (2017) (isothermal case with pressure perturbation) with gravitational potential $\phi = x$ was implemented.

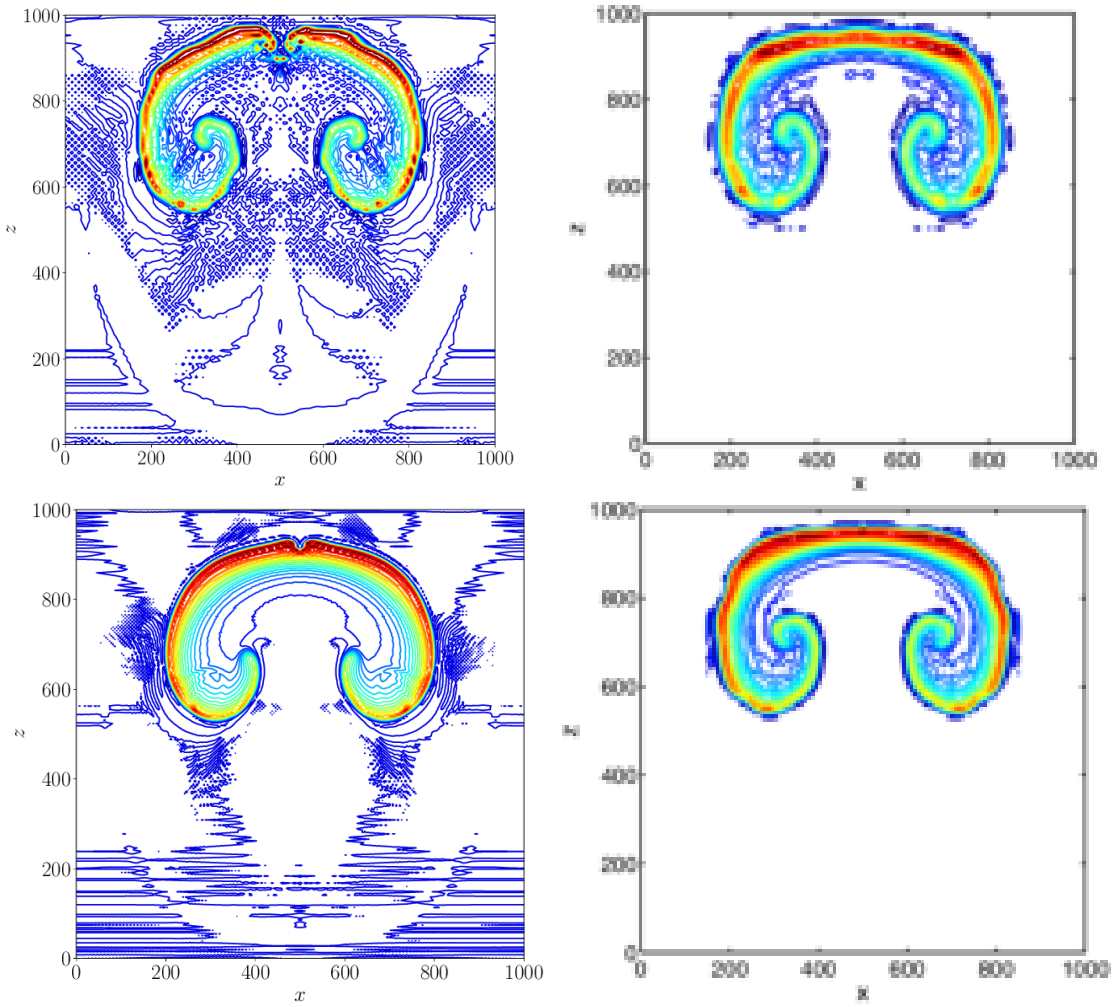

Figure 22: Results for the rising warm air bubble with the Navier-Stokes implementation before the code restructure of StormFlash. The plots on the left-hand side show the results for StormFlash with a viscosity of $\mu = 0.1$ Pa s (top: $\Delta = 7.8125$ m, bottom: $\Delta = 3.90625$ m) while the reference solutions are shown on the right (top: $\Delta = 10$ m, bottom: $\Delta = 5$ m). The contour lines are chosen to be the same as for the initial condition in **Figure 9** (and Giraldo and Restelli, 2008) to allow for a better comparison. The results obtained with StormFlash do not quite resemble the reference solutions - gaps form at the front of the bubble, the curvature of the bubble is different and the eddies inside the bubble are dampened for the bottom left plot.

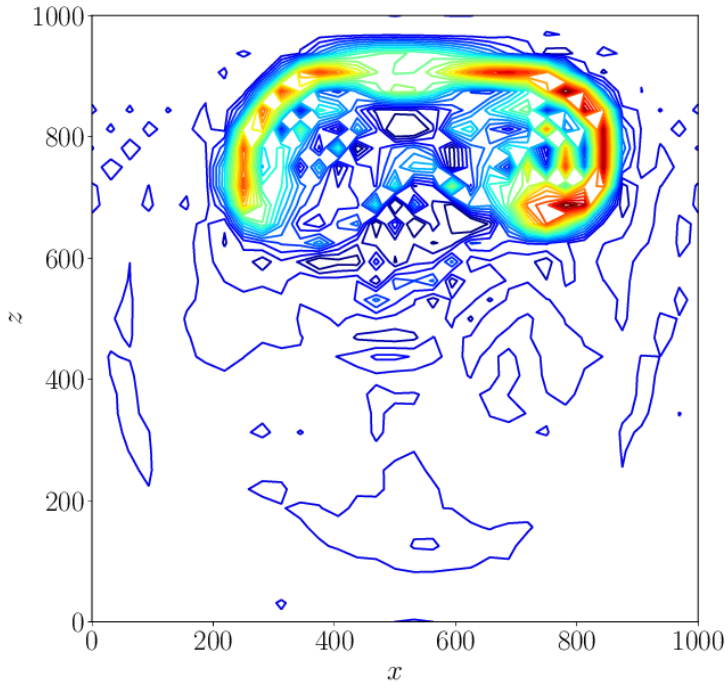

Figure 23: Results for the rising warm air bubble with the Navier-Stokes implementation after the code restructure of StormFlash. The contour lines are chosen to be the same as for the initial condition in **Figure 9** (and Giraldo and Restelli, 2008) to allow for a better comparison. The results show asymmetric behavior. Here, no reference solution is shown.

This test case was used within StormFlash (as a pseudo 1D case) and a 1D python model that is similar to StormFlash in regards to structure and implementation.
Unfortunately, in both cases the implementations did not provide meaningful results. With StormFlash, the results for this test case are comparable to the results for the neutrally stratified atmosphere test case (for that reason they are not shown here). With the 1D python model, after the well-balanced source term was implemented, the results are conserved up to machine precision but apparently there was an error in the flux calculations. Consequently, the initial condition only changed in the first time step and remained constant afterwards. Since the 1D python code was used to get an idea what to change in StormFlash to get the scheme well-balanced, this approach unfortunately did not yield conclusive results.
Consequently, at some point, answering the well-balancing question was postponed as implementing the actual plume model became more important (due to time constraints).

Following this section about the results for the stabilization problem, in the next section about the plume model, both StormFlash (with the stabilization scheme), as well as deal.II will be used. Although deal.II did not show promise regarding the physical behavior when it comes to model fluid flow with gravity, it may still be of interest to see the behavior for the plume model. Since StormFlash (with stabilization scheme) showed the most promising results, it is also an obvious choice for the basis of a plume model.

8 Plume model

Plume models are (geo)physical models that can be used to simulate eruptions that produce clouds or volcanic plume which occur in explosive volcanic eruptions - such as Vulcanian, sub-Plinian, Plinian or ultra-Plinian eruptions (Parfitt and Wilson, 2008). **Figure 24** shows the Walker classification scheme for different eruption types. As the figure shows, the volcanic eruptions that are relevant for plume models are eruptions with large ash dispersal.
These models should allow to model all three important regions of volcanic ash cloud development: the inertia dominated *gas thrust region*, the *convective region* and the *umbrella region* which are shown as sketch in **Figure 25**.
The *gas thrust region* characterizes the first part of a volcanic eruption where erupted material leaves the volcanic vent at potentially high velocities which can create acoustic or pressure waves.
In the *convective region*, the initial inertia becomes less significant and the entrainment of the surrounding air (which is usually less dense) is of greater importance. This entrainment leads to a decompression in the eruptive column which then accelerates the ash cloud once more.
Once the plume reaches a level of neutral buoyancy, since the cloud has about the same density as the surrounding air, the plume starts to spread horizontally while vertical movement of the cloud is of lesser importance (there is still some inertia left so that the plume overshoots the level of neutral buoyancy). At this stage, the cloud is also called umbrella cloud because of the horizontal spreading and thus also the term *umbrella region* is used (Parfitt and Wilson, 2008).

Plume models usually aim to simulate eruptions to estimate the plume height which for example can be input data for volcanic ash dispersion models which in turn can be used to determine no-fly zones for air traffic. For this purpose, usually 1D plume models are used. Most plume models that are currently used in the plume modeling community have been compared in the IAVCEI's eruptive column model inter-comparison study (Costa et al., 2016). In this study, nine 1D plume models are presented: Puffin (e.g. Bursik, 2001), Degruyter (e.g. Degruyter and Bonadonna, 2012), PlumeMoM (e.g. de' Michieli Vitturi et al., 2015), Devenish (e.g. Devenish, 2013), FPluMe (e.g. Cai et al., 2016), PPM (e.g. Girault et al., 2014), Plumeria (e.g. Mastin, 2007), PlumeRise (e.g. Woodhouse et al., 2013a) and ASH1D (e.g. Cerminara, 2015).

The goal for plume models with higher dimension is to model the plume height more accurately and especially to model physical processes which can then be used to analyze, for example, the dynamics of a volcanic eruption. The most prominent 3D models (which are also presented in the inter-comparison study) are ATHAM (e.g. Herzog et al., 2003),

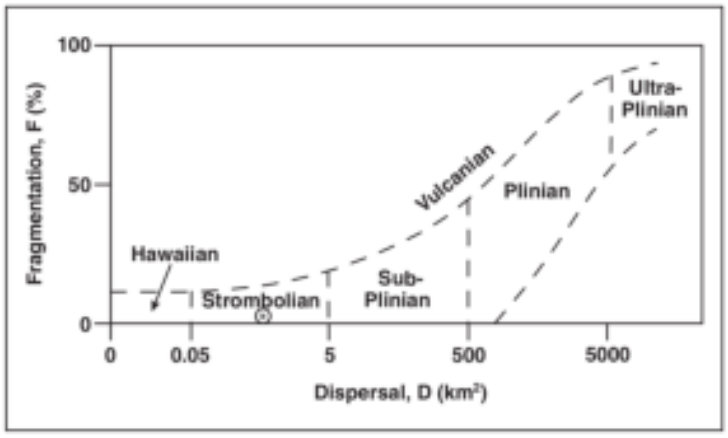

Figure 24: Walker classification scheme (Walker, 1973) where the different types of volcanic eruptions are classified according to ash dispersion or "Dispersal" and fragmentation of the magmatic material. The larger the ash dispersion, the more relevant volcanic clouds become and as a result the more applicable plume models become. This graphic was taken from Parfitt and Wilson (2008).

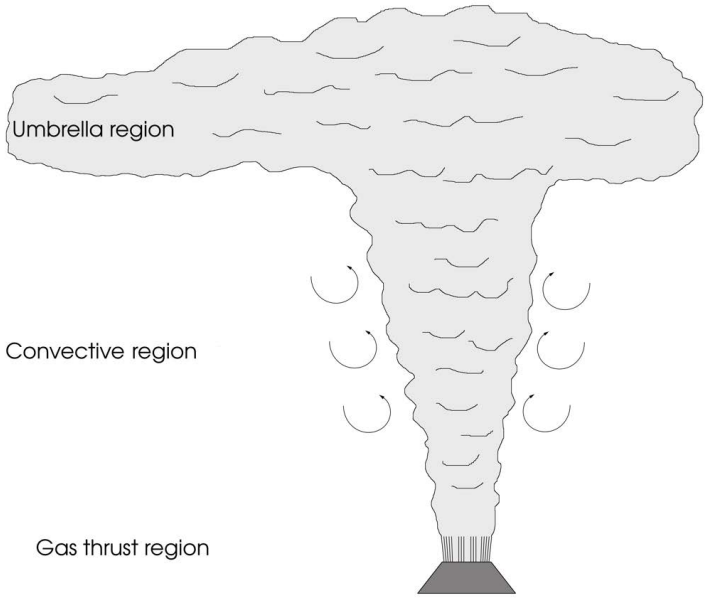

Figure 25: Schematic overview of an eruption column with its three distinct regions.

SK-3D (e.g. Suzuki et al., 2005), ASHEE (e.g. Cerminara et al., 2016a) and PDAC (e.g. Neri et al., 2003).

Since the correct height determination and the modeling of correct physics is of interest for plume models that are not 1D models, it stands to reason that a high spatial resolution is desirable. Most of the 3D models have a fixed grid size with fine resolution close to the vent that gets coarser the farther away a cell is from the vent (usually with a certain scaling factor).
In this thesis, the approach to circumvent this resolution problem is to utilize h-adaptive mesh refinement (as presented in **Section 3.3**).

8.1 Equation set for volcanic scenarios and setup

In this thesis, the same approach as for the SK-3D model is taken. This means that the underlying physics are modeled using the **Euler equations** with an additional equation for advection of the erupted material and a slightly changed calculation for the equation of state.
This approach is called pseudo-gas or dusty gas approach. The assumptions that allow for the use of this approach are outlined concisely in Cao et al. (2021), which is also based on the approach from Suzuki et al. (2005). The assumptions are quoted as follows:

- Molecular viscosity and heat conduction are neglected since turbulent energy and momentum exchange are dominant.
- Erupted material consisting of solid with different sizes and the mixture of gases is assumed to be well-mixed and behaves like a single-phase fluid (phase 2), which is valid for eruptions with fine particles and ash.
- Air, which is assumed to be a well-mixed mixture of different gases, is assumed to be another phase (phase 1).
- Assume thermodynamic equilibrium and dynamic equilibrium between the two phases. As a result, both phases share the common energy equation and momentum equations.
- All other microphysical processes (such as the phase changes of H_2O, aggregation, disaggregation, absorption of gas on the surface of solids, solution of gas into a liquid) and chemical processes are not considered in this model.
- The effect of wind is also not currently considered in this model.

Using the **Euler equations** as basis (with all its variables that have been introduced), together with these assumptions, the following set of equations is obtained:

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0, \\
\frac{\partial (\rho \xi)}{\partial t} + \nabla \cdot (\rho \xi \mathbf{u}) &= 0, \qquad \text{(volcanic setup)} \\
\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + P I_d) &= -\rho g \mathbf{k}, \\
\frac{\partial (\rho e)}{\partial t} + \nabla \cdot [(\rho e + P)\mathbf{u}] &= -g \rho \mathbf{u} \cdot \mathbf{k}.
\end{aligned}
$$

As already mentioned, compared to the Euler equations, an additional equation is added for the advection of erupted material and the equation of state is adjusted to account for an ash-gas mixture. The mass fraction of erupted material ξ is taken into account with the additional equation so that material that is added to the simulated environment gets distributed. The calculation for the equation of state is basically unchanged such that:

$$
P = (\gamma_m - 1)\left(\rho e - \frac{1}{2}\rho \mathbf{u} \cdot \mathbf{u}\right), \tag{59}
$$

where the specific heat ratio γ is changed to γ_m to make sure the erupted material is considered. This assumption is valid for pressures close to atmospheric pressure (as stated in Suzuki et al., 2005). While this does not necessarily describe all volcanic eruptions, this approach is used for ease of implementation and similarity with SK-3D. Due to consideration of erupted material in the mixture, the following relations are taken into account:

$$
\gamma_m = \frac{R_m}{c_{vm}} + 1, \tag{60}
$$

$$
R_m = \xi_g R_g + \xi_a R_a, \tag{61}
$$

$$
c_{vm} = \xi_s c_{vs} + \xi_g c_{vg} + \xi_a c_{va}, \tag{62}
$$

$$
\xi_a = 1 - \xi, \tag{63}
$$

$$
\xi_g = \xi \cdot \xi_{g0}, \tag{64}
$$

$$
\xi_s = \xi - \xi_g = \xi(1 - \xi_g), \tag{65}
$$

where the physical parameters are the same as in **Section 2** with the addition that the indices indicate mixture (m), surrounding (atmospheric) air (a), erupted gas (g) and erupted solids or ash (s). As a result the mass fractions with index also represent the mass fraction of the respective portions. ξ_{g0} is the mass fraction of erupted gas at the vent. In this thesis, the erupted gas will just be water vapor (without any other gases).

At the volcanic vent, the following (constant) quantities have to be provided:

- Atmospheric (reference) pressure P_{ref} (at $z = 0$ m) which is also assumed at the vent (no overpressure),
- Vertical exit velocity w_0, whereas the horizontal exit velocity is set to zero,

- Temperature of the erupted material T_m,
- Initial mass fraction of volcanic gas (water vapor) ξ_{g0}.

Using these quantities, the density of erupted material at the vent $\rho_{m,0}$ can be calculated using the equation of state (ideal gas law):

$$\rho_{m,0} = \frac{P_{\text{ref}}}{R_m T_m}, \tag{66}$$

where it should be noted that $R_m = \xi_{g,0} R_g$ at the vent since $\xi = 1$.

For 3D volcanic plumes, Morton et al. (1956) found a relationship between the height of the volcanic plume and the mass flux. Within this study, a proportionality was found that the plume height H (in km) is proportional to the mass flux at the vent M_f to a power of 1/4. Parfitt and Wilson (2008) gives the following relationship for a "standard atmosphere" on Earth:

$$H = \alpha M_f^{1/4}, \tag{67}$$

where α is a parameter that contains the fit to this "standard atmosphere". This parameter is comprised of a constant factor c, a density factor for air ρ_{a_0} (units: kg m^{-3}), the gravitational acceleration g (units: m s^{-2}) and the buoyancy frequency N (units: s^{-1}), where $\alpha = c\rho_{a_0}^{-1/4} g^{1/4} N^{-3/4}$ (see Woodhouse et al., 2013b). For the "standard atmosphere", this factor α is 236 m kg$^{-1/4}$ s$^{1/4}$ or 0.236 km kg$^{-1/4}$ s$^{1/4}$. As a remark, instead of M_f, many pieces of literature also use either Q or Q_0 as notation for the mass flux. It should be noted that the plume model used in this thesis will just be used in 2D for now (in line with the previous chapters). This is mainly to showcase the possibility that adaptive mesh refinement offers for the modeling of volcanic eruptive columns and easier implementation in 2D compared to a full 3D model.

8.2 General overview of volcanic test cases

Following the approach from Suzuki et al. (2005), in general, the domain is chosen in such a way that the symmetry of the (Cartesian) 2D setup is used. Since a (Cartesian) 2D plume model basically models a line volcano, the flow is axisymmetric (with respect to the vertical axis) which allows to model just one half of the volcanic eruption. As a result, the volcano is chosen to be placed at the bottom left-hand corner of the domain which corresponds to the origin. At the volcanic vent, the necessary quantities (see previous section) are given as constant inflow conditions.

The vertical axis or axis of the flow which corresponds with the left boundary of the domain is prescribed with a no-flux/free-slip boundary condition to preserve the symmetry of the flow.

Differing from Suzuki et al. (2005), the bottom boundary (other than the volcanic vent) is prescribed with a no-slip boundary condition while the top and right boundaries are also

set up to be Dirichlet boundary conditions.
For the no-slip boundary, all variables on the one side of the boundary $\mathbf{q}^+$ are set to the values on the other side $\mathbf{q}^-$ except for the velocities (and consequently the momenta) which are set to 0.
For the Dirichlet boundary, the boundary values are the initial conditions for **Section 7.1**.

Regarding the different numerical bases, the plume model has been implemented in both StormFlash, as well as deal.II. Unfortunately, at the time of writing this thesis, a bug within deal.II occurred, which did not allow for adaptive mesh refinement for the array structure for which the CFD approach with deal.II step-67 was implemented. Consequently, no adaptive runs will be shown for deal.II. For all simulations with just air, the CPU times that will be given were performed on an Intel Core i7-7500U. For the simulations with water vapor, an Intel i5-8350U was used as well. It should be noted that the CPU times for the run with deal.II will not be provided. The CPU times just serve as a way to calculate speed up for the adaptive mesh refinement cases, which is not necessary for the case with deal.II.

In total, four test cases will be used:

- A volcanic jet where the erupted material is comprised of dry air that has the same temperature as the surrounding atmosphere,
- a volcanic jet where the erupted material is comprised of hot dry air,
- a volcanic jet where hot water vapor is erupted,
- a volcanic plume that is comprised of hot water vapor and ash.

With each setup, the implementation will be improved incrementally. Consequently, ways to solve issues that occur will often be implemented in conjunction with making the setup more complicated (but also realistic).

8.3 Volcanic jet without water vapor and ash

For the first test case, the neutrally stratified atmosphere from **Section 7.1** is used as a basis (with the exception that $P_{\text{ref}} = 101325$ Pa) with just air being erupted from the volcano. Hence, no water vapor or solids are erupted and $\xi = 0$ everywhere.
As previously mentioned, a volcanic vent is added in the bottom left corner. The domain extends 10 km in each direction, respectively, and the vent has a radius of 625 m. For this test case, the maximum simulation length t_{max} is set to be 50 s. For this simplified volcanic jet, only the vertical exit velocity and temperature have to be provided. The exit velocity is set to $w_0 = 275$ m s^{-1} while the temperature will be set to the same as the atmospheric air such that $T_m = T_{\text{ref}} = 300$ K for the first jet case and $T_m = 1053$ K for the second jet case.

Most of those parameters are taken from the *strong plume* test case from Costa et al. (2016). As will be roughly estimated later in this section, the plume height would be much larger than the domain size. For this reason especially, the maximum simulation length is chosen to be 50 s as to just simulate the initial phase of a volcanic setup. Consequently, the plume or jet does not reach large heights and as a result, no boundary effects from the right or top boundaries on the eruptive column occur.

The idea of this first test case is to show the general behavior of the fluid flow and analyze the adaptive mesh refinement for a simple case first before a "full" volcanic setup is used.

For the following runs, the finest resolution will be set to 39.0625 m. In the case of uniform calculations, this is both the vertical and horizontal resolution while in the adaptive case, the resolution can range from $\Delta_{\text{finest}} = 39.0625$ m to $\Delta_{\text{coarsest}} = 625$ m. All of these simulations are run with a time step size constraint due to the CFL number which is set to 0.015. This small CFL number is chosen to ensure stability also for higher stiffness due to the more complex problem with the volcanic setups.

Initially, while running adaptive test cases for StormFlash, runs were performed where no coarsening was used. Unfortunately, these runs became unstable after several seconds of simulated time. These instabilities are likely caused by incorrect values for quantities (density and velocities) close to the vent. One way to solve these issues was to switch to different boundary conditions for the bottom edge of the domain (which represents the ground). This approach was used for the jet where the jet temperature is the same as for the surrounding atmosphere.
Another idea on how to deal with the instabilities close to the vent is to use a dampening factor f_{damp} close to the outer rim of the volcanic vent to enable a smooth transition from all ejected quantities to the atmospheric ones and have a less pronounced discontinuity at the junction from volcanic vent to ground. The following dampening factor is used:

$$f_{\text{damp}} = \sin\left(\frac{\pi|x - r_v|}{2L_{\text{damp}}}\right), \tag{68}$$

where $x \in \Gamma_{\text{vent}}$ are the coordinates for the nodes that make up the volcanic vent, r_v is the radius of the vent (625 m) and L_{damp} is the length over which the quantities are dampened (which is set to 200 m). Consequently, the dampening factor is applied for $x \in [425\text{ m},\ 625\text{ m}]$. It should be noted that using such a dampening factor reduces the mass flux at the vent compared to a discontinuous profile, if nothing else is changed (such as vent radius, for example).
With these implementations (different boundary conditions and/or including a dampening factor), adaptive runs with coarsening are stable.

Without the dampening factor, the mass flux for three-dimensional plumes can be calculated via:

$$M_f = \rho_{m,0} w_0 \pi r_v^2. \tag{69}$$

Even though this is not realistic for this plume model (since it is two-dimensional), this would lead to a mass flux of roughly $3.97 \cdot 10^8$ kg s^{-1} with the parameters for the jet with the same temperature as the surrounding atmosphere, which in turn would lead to a jet height of about 33.3 km and a mass flux of roughly $1.13 \cdot 10^8$ kg s^{-1} with corresponding height of 24.3 km for the jet with hot air. Since this plume model is just 2D, this plume height is just a very rough estimate, which is incorrect since the mass flux and plume height estimations rely on the three-dimensionality of the problem. However, this shows that a plume (and jet) should not collapse at very low heights.

Regarding the figures for the jets/plumes, only the plots for deal.II will be explicitly labeled as such, all other plots will feature results obtained with StormFlash. The results for the uniform runs will be shown in the next sections while most plots for the adaptive runs can be found in **Appendix E** (for readability).

8.3.1 Results for deal.II

As already mentioned, no adaptive runs could be performed with deal.II due to problems within the library. Hence, only the results for a uniform run will be shown.

The detailed results can be found in **Appendix D**, while **Figure 26** shows the density after $t_{\max}$. The plot shows that the jet basically collapses after reaching a height of approximately 2 km despite having a large vertical momentum. This behavior is not an expected behavior of a jet that exits the vent with a velocity of 275 m s^{-1} and has a mass flux of about $3.97 \cdot 10^8$ kg s^{-1} which should results in jet heights of several kilometers at least (even if the 33.3 km approximation from **Equation 67** is not precise for a 2D model).

This incorrect behavior might be a result of a missing stabilization scheme and is in line with the results for the warm air bubble test case, where the simulations with deal.II differed from the reference solution. Additionally, since deal.II does not allow for correct adaptive mesh refinement as of the time of writing this thesis (at least for the approach used in deal.II step-67), deal.II will not be used for the purpose of modeling volcanic eruptions until this issues have been addressed.

8.3.2 Results for StormFlash

For the test case with StormFlash, both uniform and adaptive simulations are run. These simulations are set up as stated earlier.

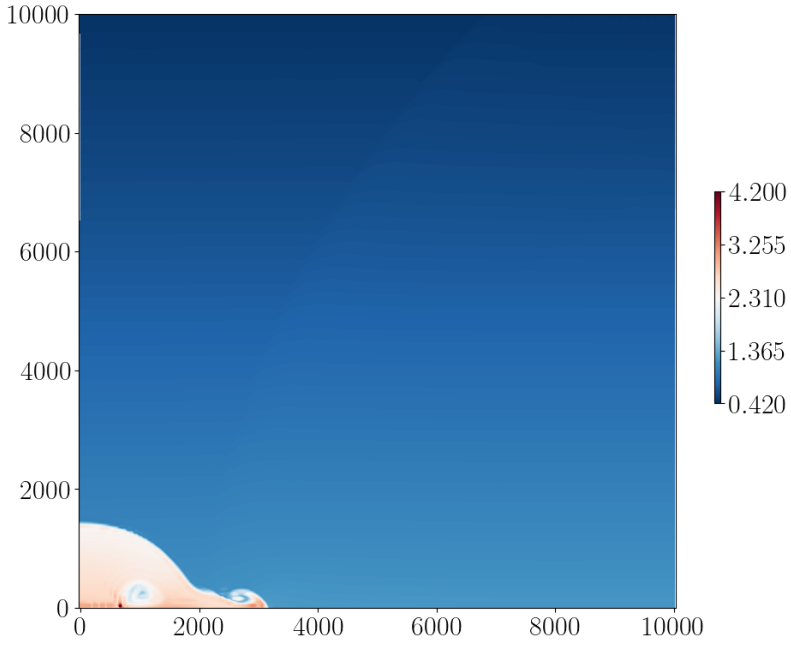

Figure 26: Density for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere for deal.II after 50 seconds of simulated time. The colorbar shows values in kg m^{-3}. As the plot shows, the jet reaches a maximum height of around 2 km and then collapses.

Jet with $T_m = T_{\text{ref}} = 300$ K

As stated in **Section 8.3**, instabilities start to develop. These instabilities bear resemblance to Kelvin-Helmholtz instabilities and seem to appear at the boundary between volcanic vent and ground and are transported along the trajectory of the jet. Interestingly, these instabilities are also somewhat similar to *von Kármán vortex streets* rising from where the quantities above the junction from volcanic vent to ground appear to have discontinuities.

As previously mentioned, different boundary conditions for the bottom boundary which represents the ground are used. The uniform results for the no-slip boundary are presented in **Figure 27**, **Figure 28**, **Figure 29** and **Figure 30** while the results for the Dirichlet boundary are shown in **Figure 31**, **Figure 32**, **Figure 33** and **Figure 34**.

These figures show that the jet starts to develop large vortex structures which reach a height of roughly 3000 m after t_{max} is reached. The plots also suggest that if the simulation were to keep on running the jet would keep on rising due to the large vertical momentum and no sign of collapse, as in the deal.II run. With a mass flux of $3.97 \cdot 10^8$ kg s^{-1}, the jet should reach an approximate height of 33.3 km so a continued ascent of

the jet would be realistic. In the future, simulations with larger domain and thus longer simulation will be performed to verify this behavior.

Regarding the adaptivity, in this scope, the following combinations of tolerances for refinement σ_{ref} and coarsening σ_{coarse} are used, respectively:

- Case A: $\sigma_{\text{ref}} = 0.01$, $\sigma_{\text{coarse}} = 0.005$, no-slip boundary;
- Case B: $\sigma_{\text{ref}} = 0.001$, $\sigma_{\text{coarse}} = 0.0005$, no-slip boundary;
- Case C: $\sigma_{\text{ref}} = 0.01$, $\sigma_{\text{coarse}} = 0.005$, Dirichlet boundary.

Regarding the refinement and coarsening, the procedure as presented in **Section 3.3.2** is used. For Case A and Case B, the error indicator $\mathbf{e}$ is set to be the gradient of the vertical momentum such that $\mathbf{e} = \nabla(\rho w)$. For Case C, the error indicator is switched to be the maximum of the gradient of either of the momenta such that $\mathbf{e} = \max|\nabla(\rho \mathbf{u})|$.
The results for these cases are presented in **Appendix E**.
For Case C, the switch in boundary conditions was necessary as to enable runs with better setups for the initial mesh. With the adaptive simulations for Case A and Case B (see **Appendix E**), the initial grid was not refined around the vent region until roughly one second of simulated time. Consequently, the results after $t_{\max}$ look different due to the non-linearity of the equation set.
The idea for Case C then was to ensure a region close to the vent was refined to the finest resolution. In this case, this region was chosen to be a square with side length 625 m (which is the vent radius) directly above the vent. Without the switch in boundary conditions, the adaptive results became unstable after a certain amount of time (which differed depending on the parameters for grid refinement).

Table 10 shows the CPU times for the uniform runs for the setups where the volcanic jet has the same temperature as the surrounding atmosphere. As the table shows, both Case B and C have a speed of factor of about 8 which is quite substantial. However, it should be noted that the adaptive results after $t_{\max}$ differ from those obtained with the uniform simulation. This is due to the initial adaptive mesh being not refined enough in regions where high resolution would be required (e.g. regions directly above the volcanic vent). Consequently, a mismatch between small and large scale effects leads to different results down the line due the non-linearity of the equation set. **Figure 35** shows the comparison between the adaptive and the uniform simulations for the no-slip boundary for the ground while **Figure 36** shows the comparison between the adaptive and the uniform simulations for the Dirichlet boundary for the ground. In both figures, just the horizontal momentum is shown since this is the variable that presents the difference between the runs the best. All three adaptive cases show a distinct mismatch with the uniform simulations which is due to the already mentioned effects within the (initial) grid. Both for Case A and Case C, the grids also do not capture the complete structure of the jet in the top regions. Since the

Table 10: CPU times for all setups for the volcanic jet without water vapor and ash with $T_m = T_{\text{ref}} = 300$ K. Additionally, the percentage of the CPU time of the uniform runs is provided, as well as speed up and the refinement/coarsening tolerances.

boundary type for the ground		CPU time [s]	percentage of uniform run	speed up	σ_{ref}	σ_{coarse}
no-slip	uniform	158800				
	Case A	19660	12.380	8.078	0.01	0.005
	Case B	84590	53.268	1.877	0.001	0.0005
Dirichlet	uniform	164100				
	Case C	20310	12.380	8.078	0.01	0.005

adaptive results differ from the uniform ones, the mesh refinement needs to be adjusted. This will be done within the next test case setup (volcanic jet with just hot air).

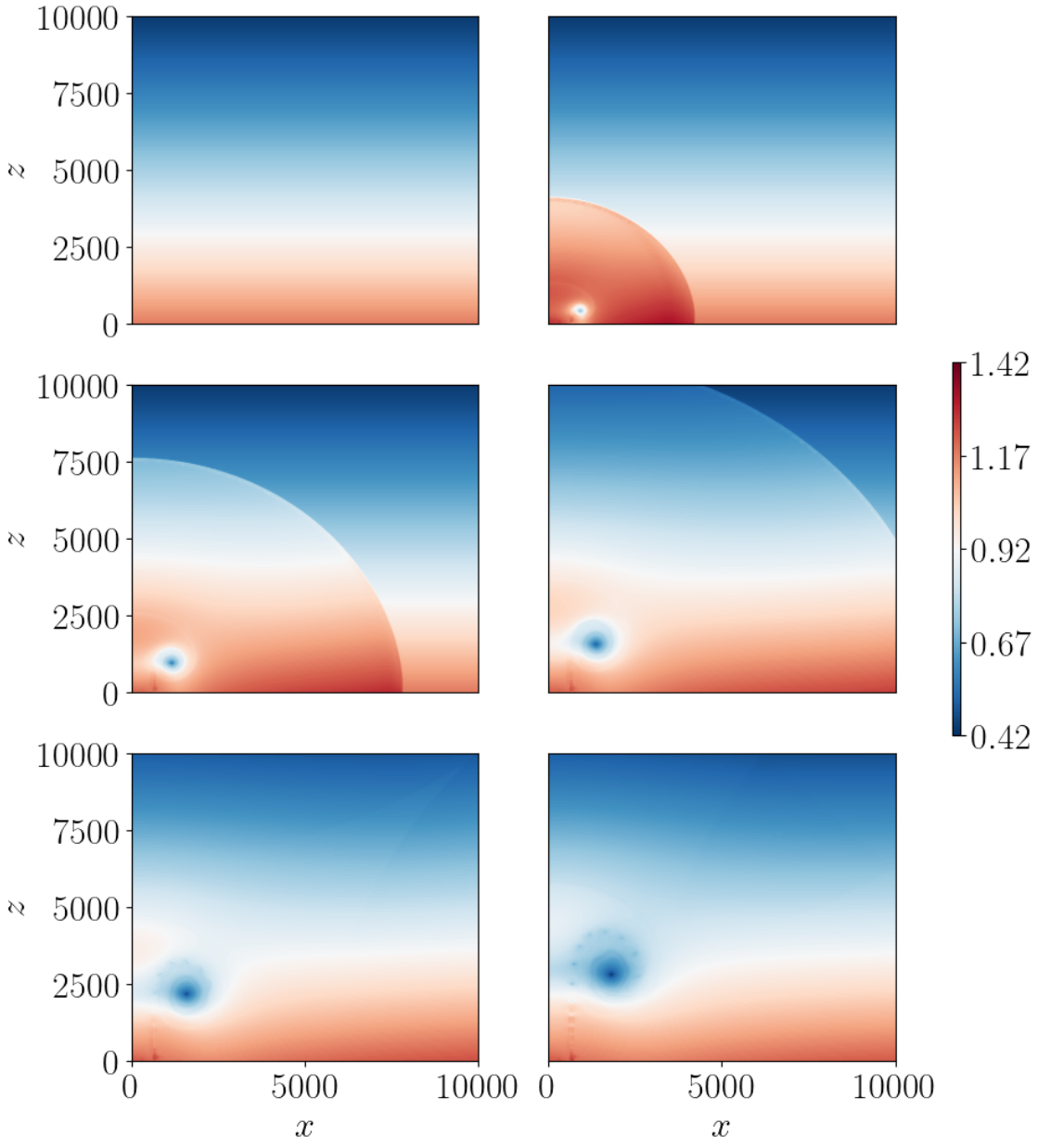

Figure 27: Density for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with no-slip boundary for the ground. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

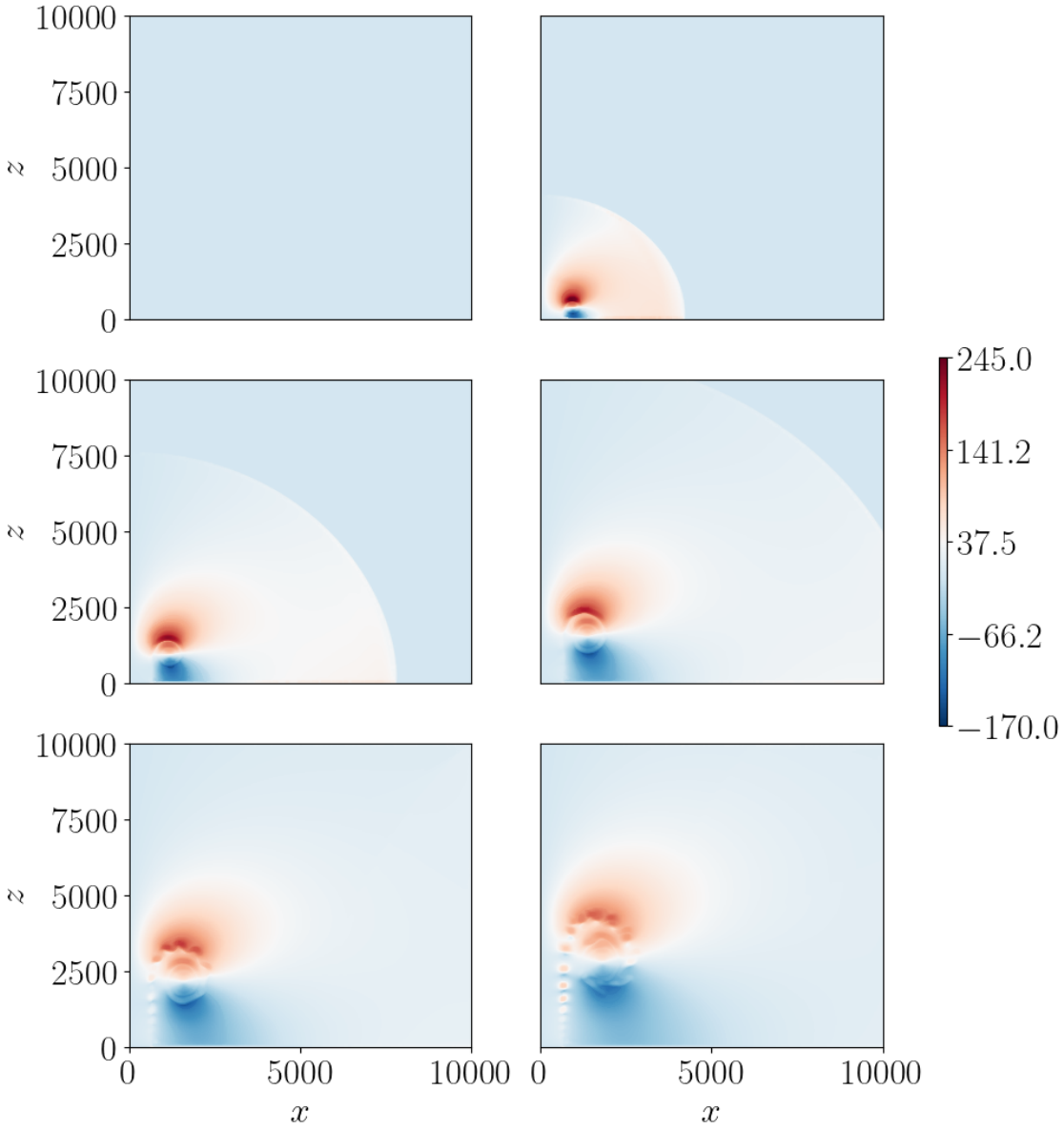

Figure 28: Horizontal momentum for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with no-slip boundary for the ground. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, areas of high and low (negative) momentum, respectively, develop which are situated around a center that is roughly located at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

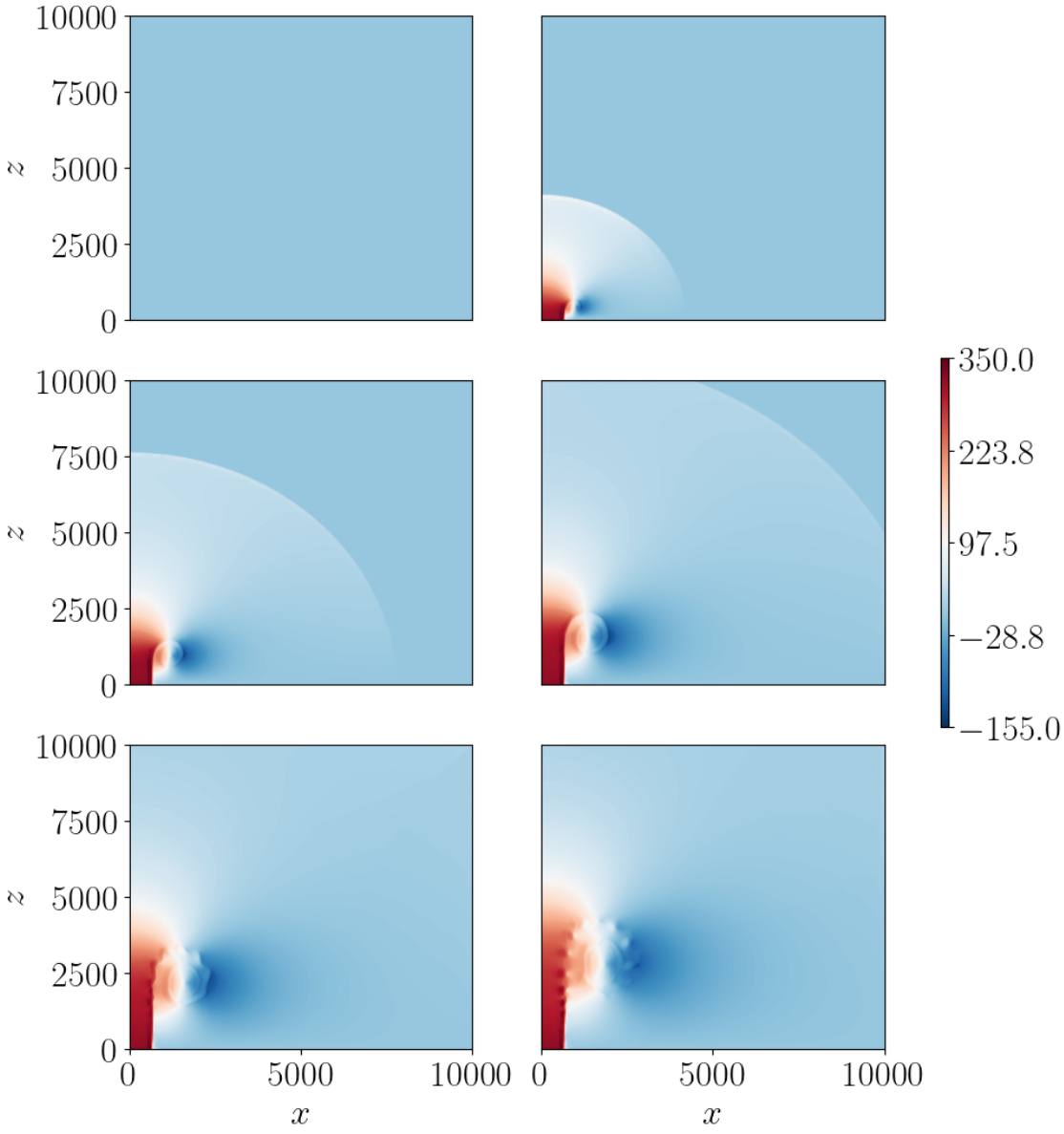

Figure 29: Vertical momentum for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with no-slip boundary for the ground. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, areas of high and low (negative) momentum, respectively, develop which are situated around a center that is roughly located at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

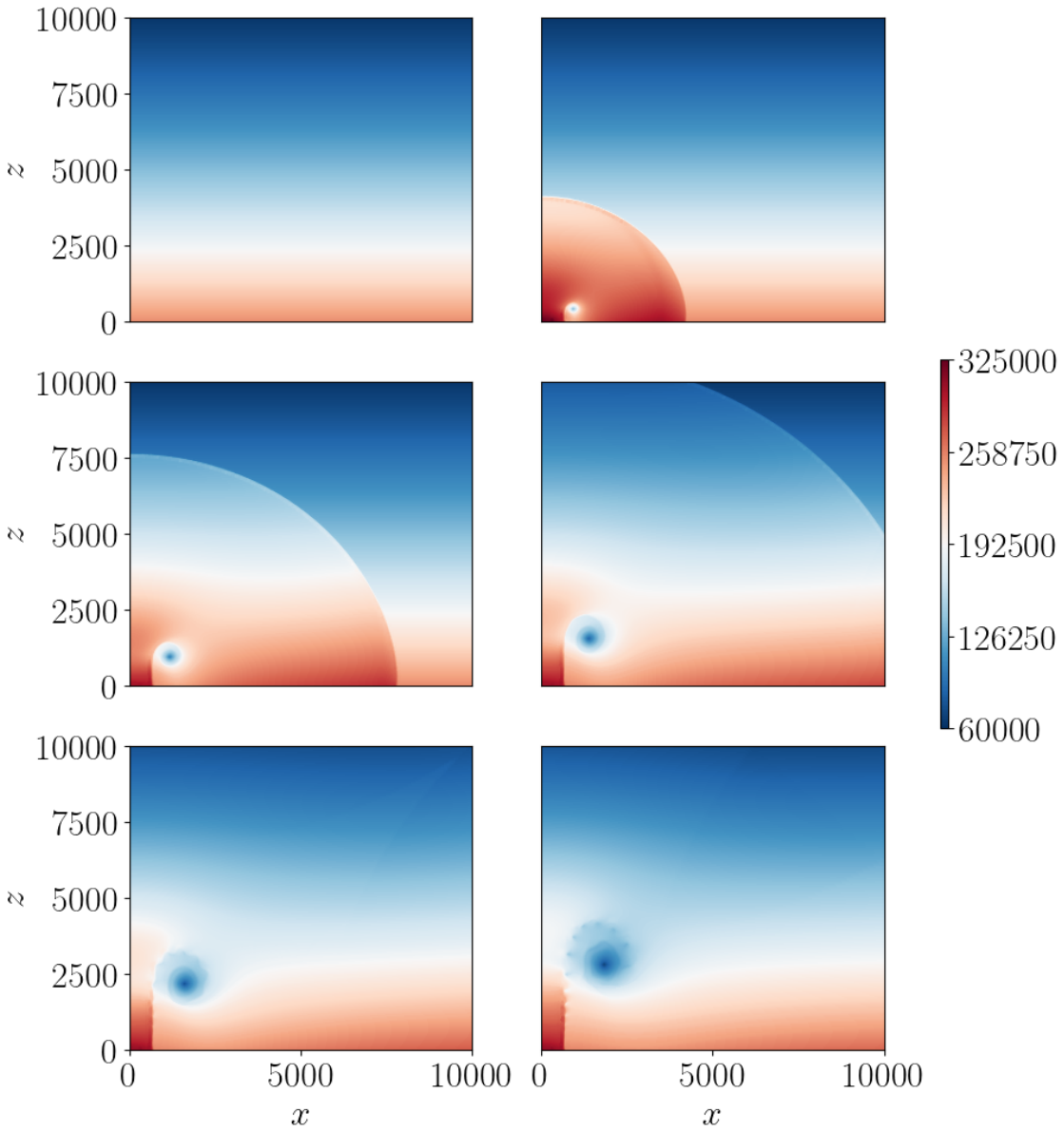

Figure 30: Energy density for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with no-slip boundary for the ground. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower energy density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

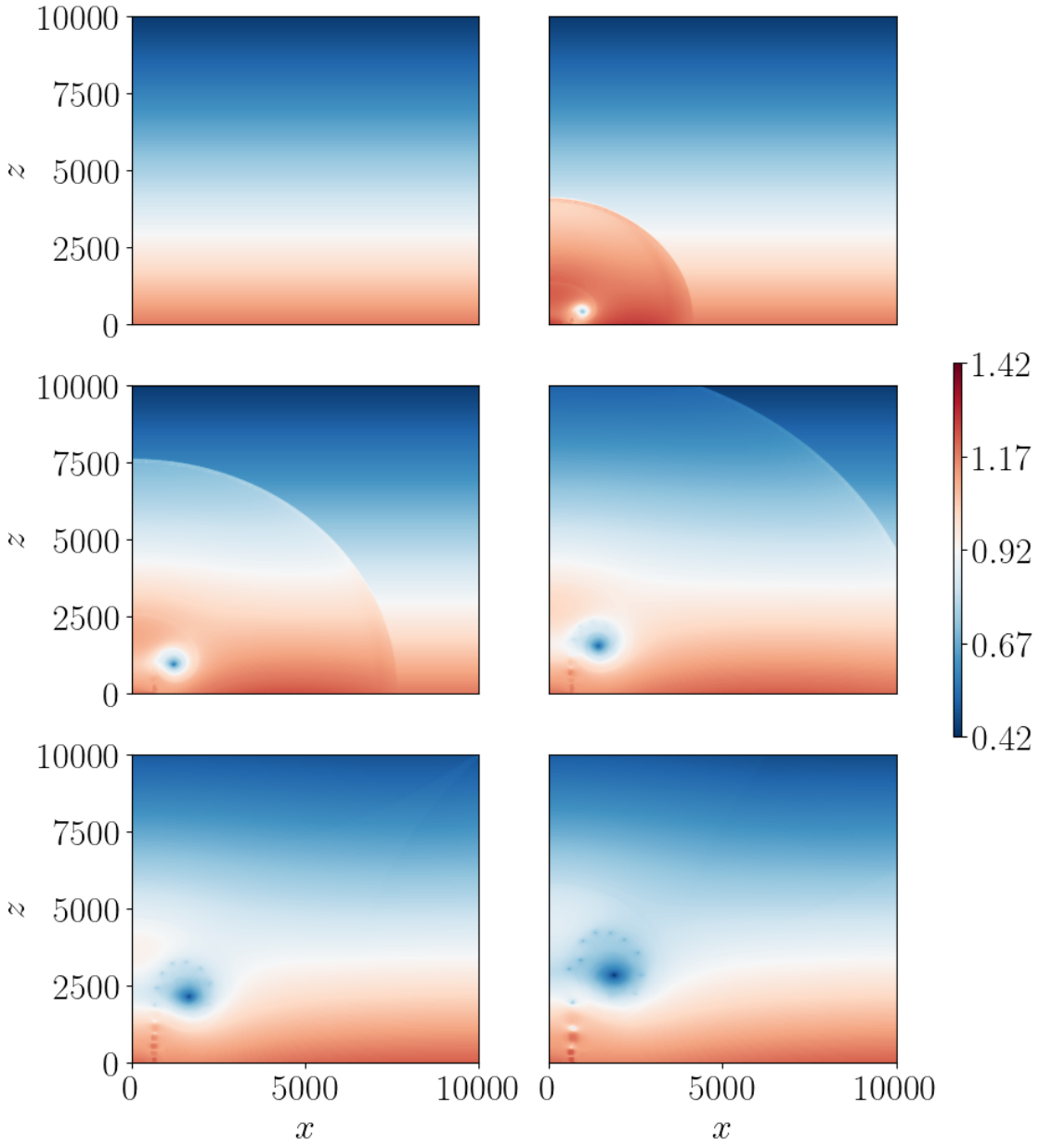

Figure 31: Density for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with Dirichlet boundary for the ground. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

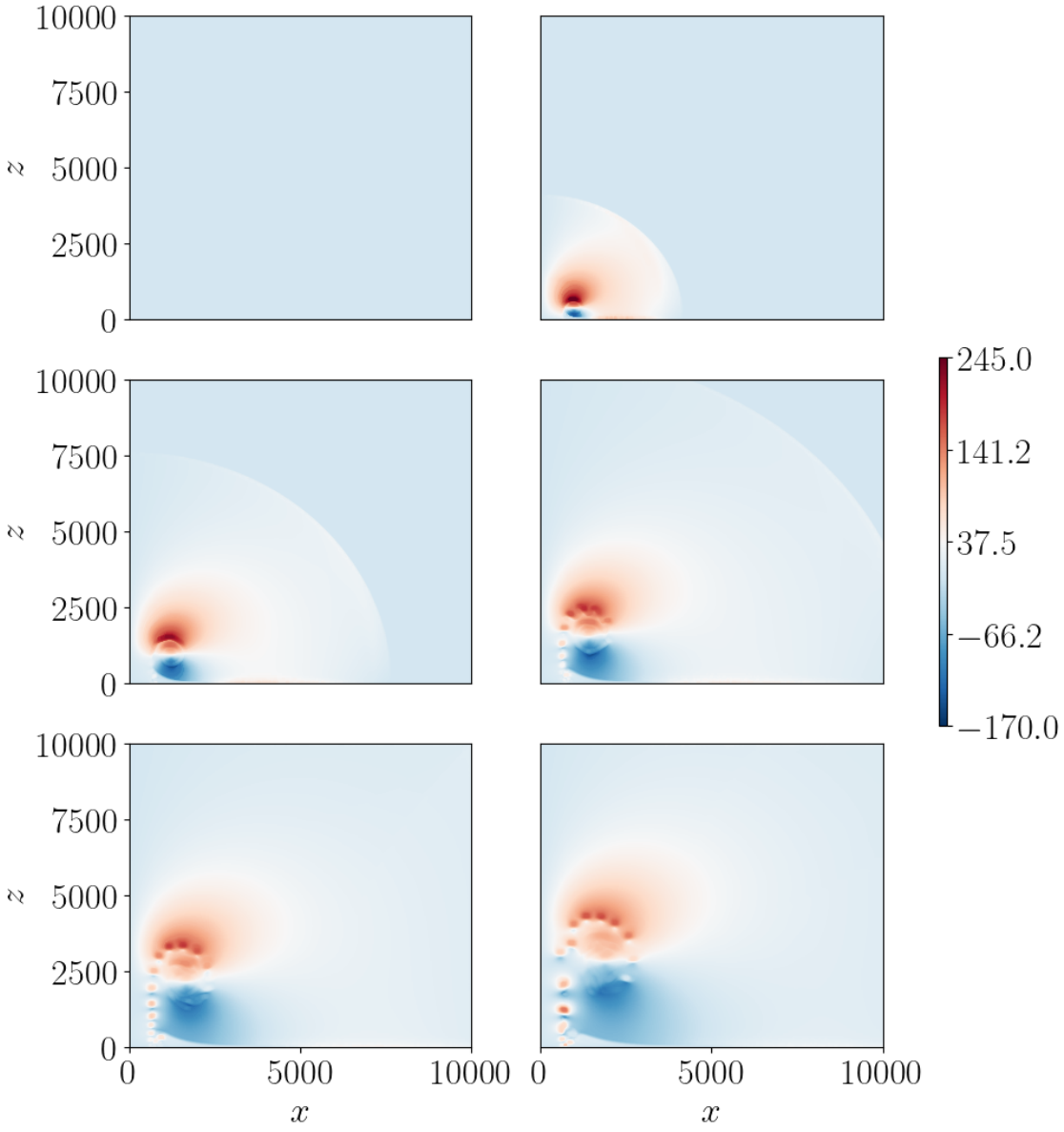

Figure 32: Horizontal momentum for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with Dirichlet boundary for the ground. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, areas of high and low (negative) momentum, respectively, develop which are situated around a center that is roughly located at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

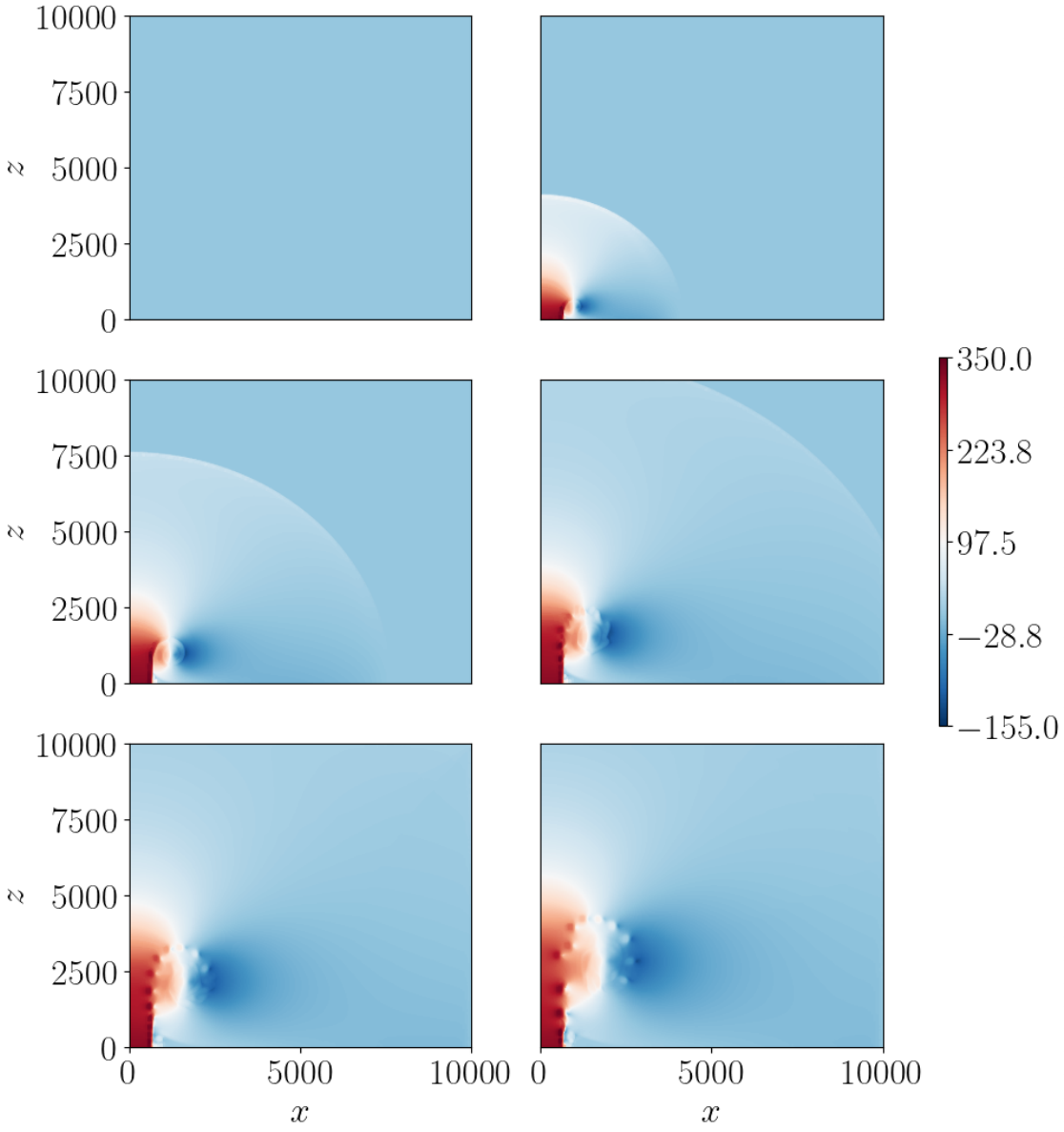

Figure 33: Vertical momentum for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with Dirichlet boundary for the ground. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, areas of high and low (negative) momentum, respectively, develop which are situated around a center that is roughly located at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

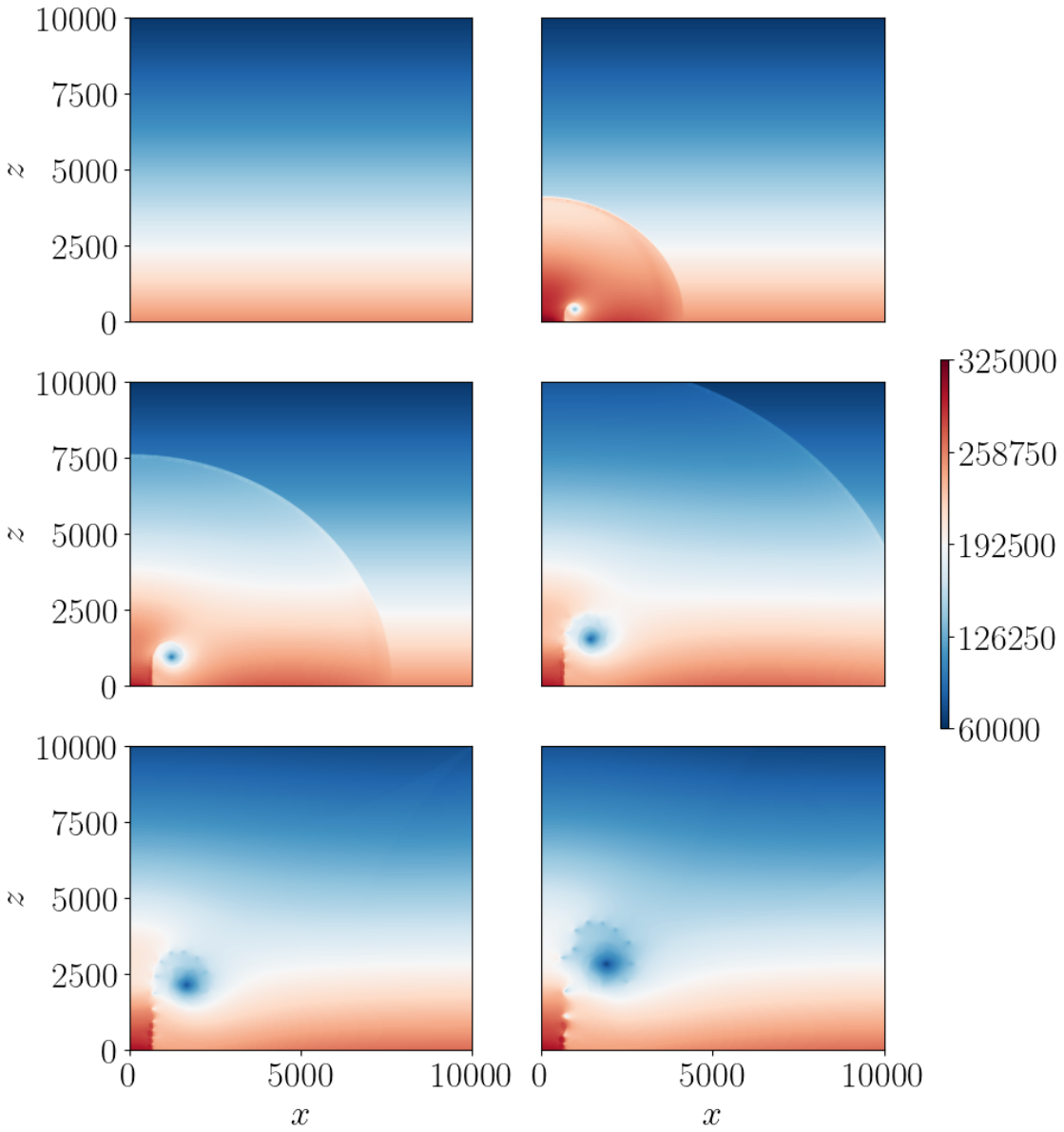

Figure 34: Energy density for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with Dirichlet boundary for the ground. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower energy develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

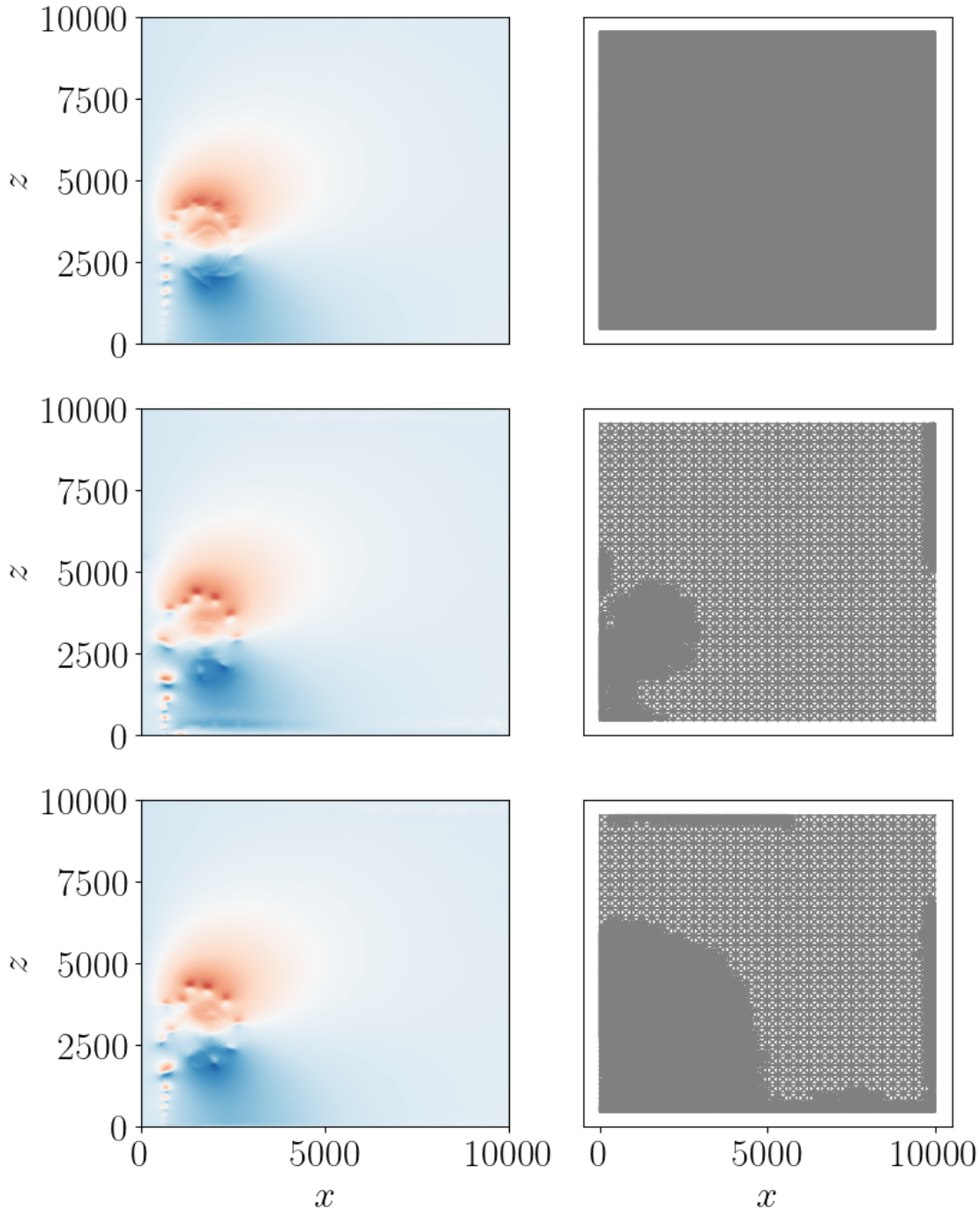

Figure 35: Comparison between uniform and adaptive runs for the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with no-slip boundary for the ground. On the left, the horizontal momentum is shown after $t_{\max}$ while the right shows the corresponding grid. The colormap is setup just as in the previous (uniform) figures. The top rows shows the uniform results, the middle row the results for Case A and the bottom row shows the results for Case B. The adaptive plots differ visually from the uniform plots with different behavior regarding the instabilities.

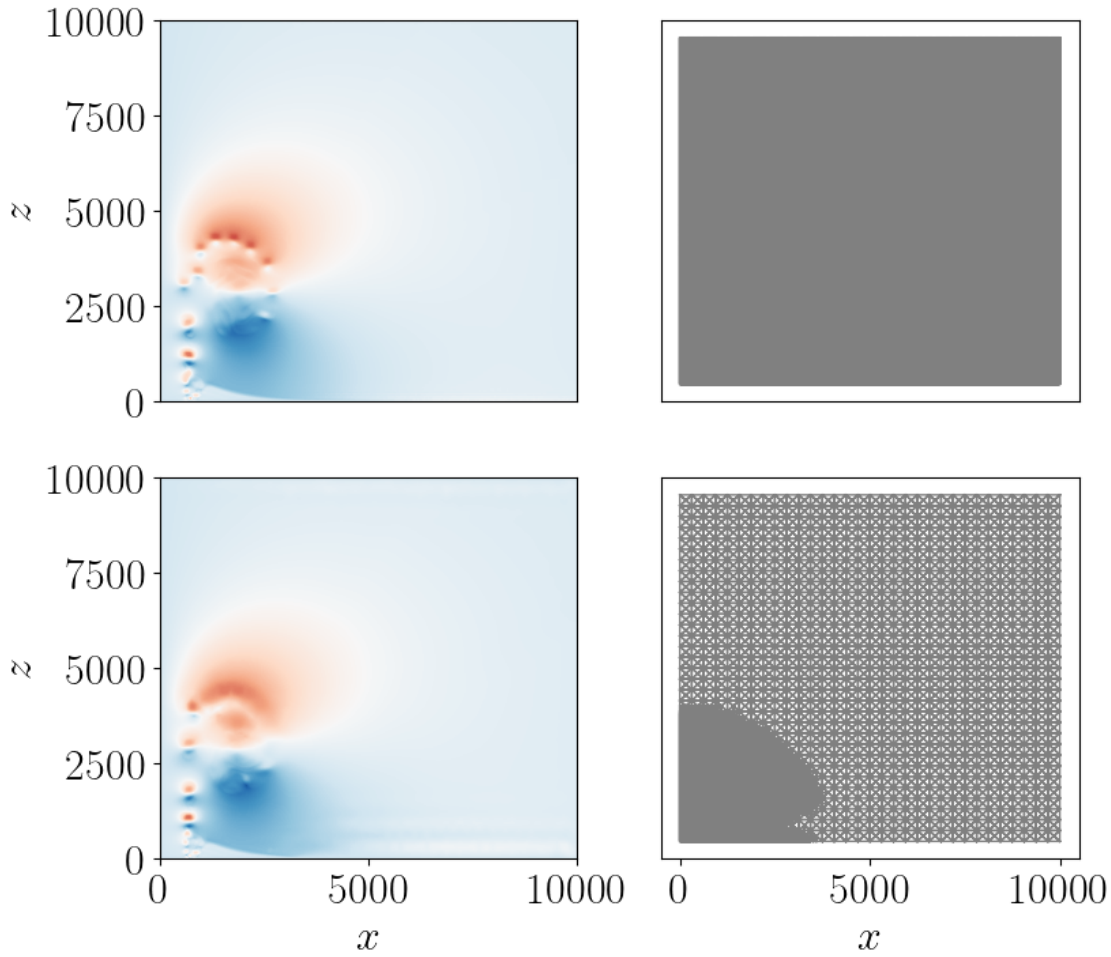

Figure 36: Comparison between uniform and adaptive runs for the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere with Dirichlet boundary for the ground. On the left, the horizontal momentum is shown after $t_{\max}$ while the right shows the corresponding grid. The colormap is setup just as in the previous (uniform) figures. The top rows shows the uniform results and the bottom row shows the results for Case C. For the adaptive run, the instabilities are less pronounced in the upper part of the jet. The grid is not refined to capture the whole jet structure.

Jet with T_m = 1053 K

For this setup, the dampening factor that was introduced earlier is applied to all quantities at the vent. As a result, the results from the simulations no longer show instabilities reminiscent of Kelvin-Helmholtz instabilities within the jet.
Because of improved stability, Dirichlet boundary conditions are used for the ground for all simulations for this test case.

Another issue within this scope is that physically unrealistic densities have to be tackled. Within several seconds of the simulation, a vacuum within a forming vortex would develop which is due to the scheme not having diffusion or viscosity. While the approach for other numerical implementations - as for example the one with SK-3D - might not lead to such problems, the approach with how the DGM is implemented within StormFlash leads to a scheme that does not provide enough numerical diffusion, for example.
The lack of friction within the system then leads to a cascade where the densities get very low which in turn leads to very high velocities that again influences time step sizes, for example.
Consequently, an idea on how to deal with these low density is needed. As a very heuristic and almost trivial approach, the quantities get limited to a certain range so that this approach basically works like a filter. Limiting the quantities obviously raises the question where they should be cut-off. As a simple approach, a low to medium vacuum is assumed which ranges from atmospheric pressure to 10^{-2} Torr which would correspond to about 1.33 Pa (Roth, 2012). As first, rather arbitrary value, a pressure of 15000 Pa (for a low vacuum) is used to calculate the corresponding vacuum density ρ_v that would occur within dry air and the jet temperature of T_m = 1053 K. Using the ideal gas law, $\rho_v \approx 0.05$ kg m^{-3} is obtained. In addition, to ensure stable simulations, the energy is limited to be positive and the momenta are set up to lie between $\pm$1500 kg m^{-2} s^{-1} to somewhat limit the maximum velocities that can occur.

Figure 37, **Figure 38**, **Figure 39** and **Figure 40** show the plots for the uniform run. These plots clearly show the vertical rise of the volcanic jet. Compared to the previous setups, the jet rises higher within the simulated 50 seconds which is to be expected. Again, the plots for the adaptive runs are presented in **Appendix E**.
For the adaptive setups, the same error indicator as in Case C is used (which is the maximum gradient for the momenta). In addition to the refinement carried out due to the refinement criteria, for these cases, extra layers of refined cells are added that the most refined region is extended to allow to better capture the areas of interest (where the gradients are higher).

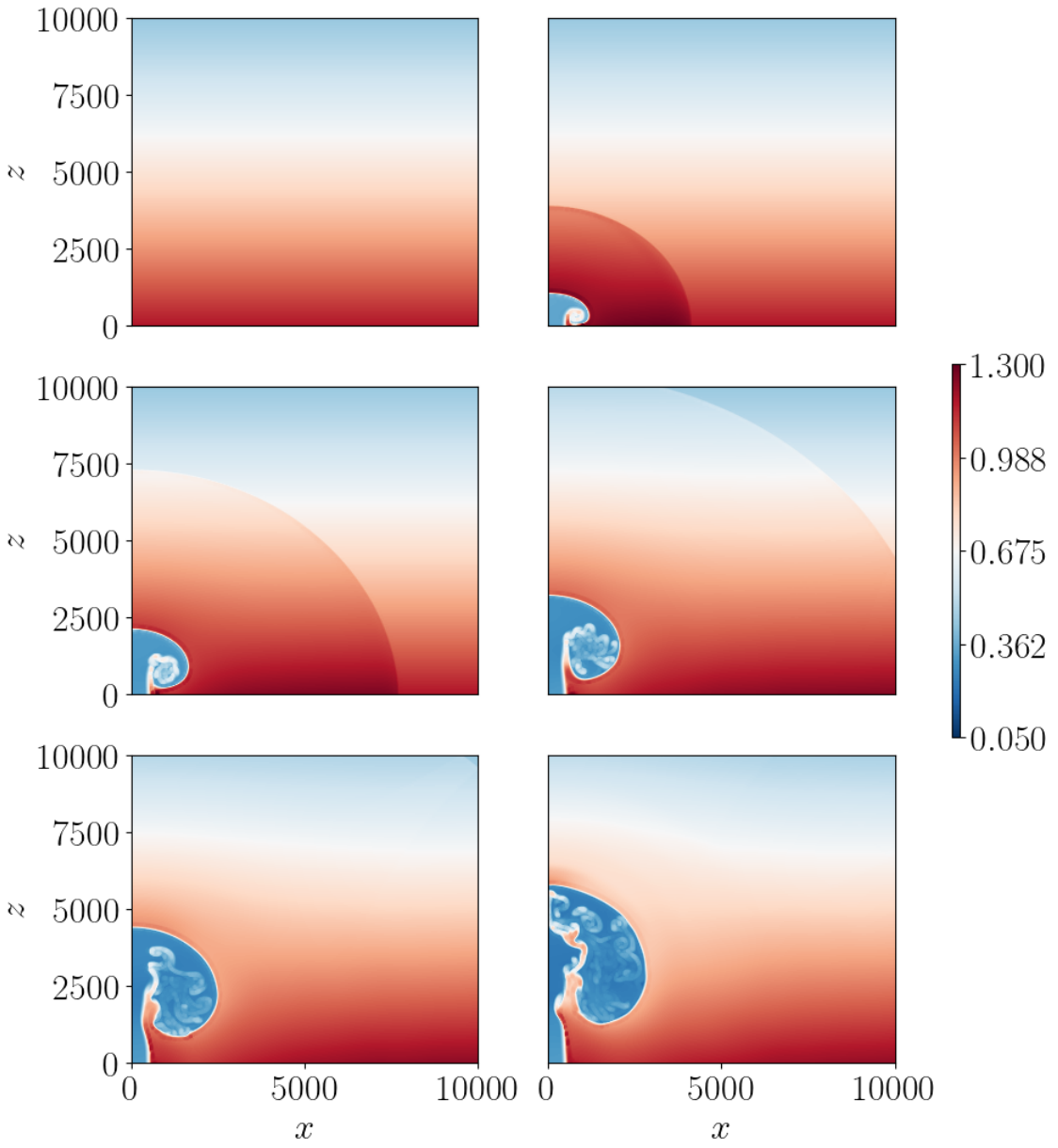

Figure 37: Density for the uniform run of the volcanic jet without water vapor and ash with T_m = 1053 K. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

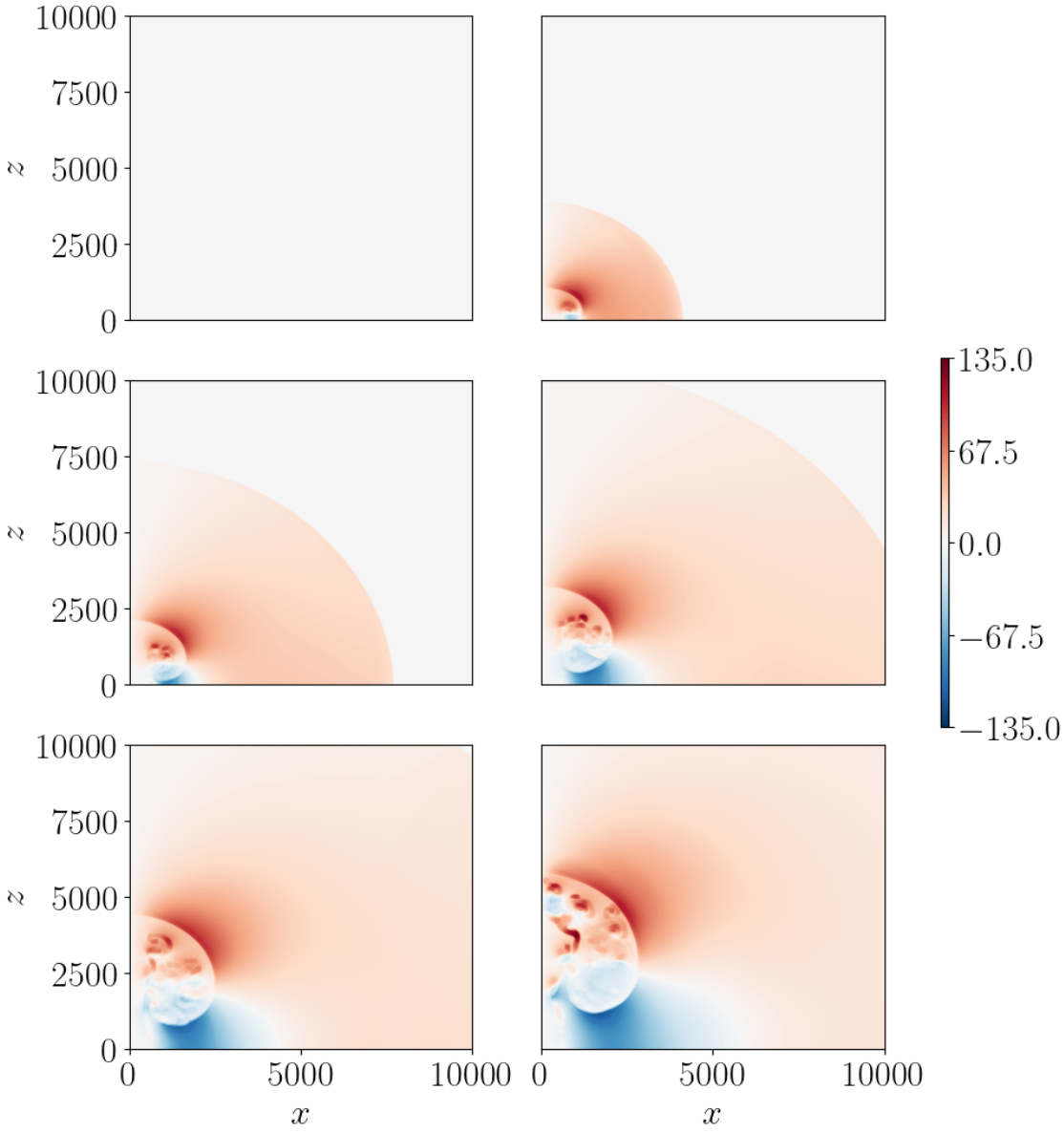

Figure 38: Horizontal momentum for the uniform run of the volcanic jet without water vapor and ash with T_m = 1053 K. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

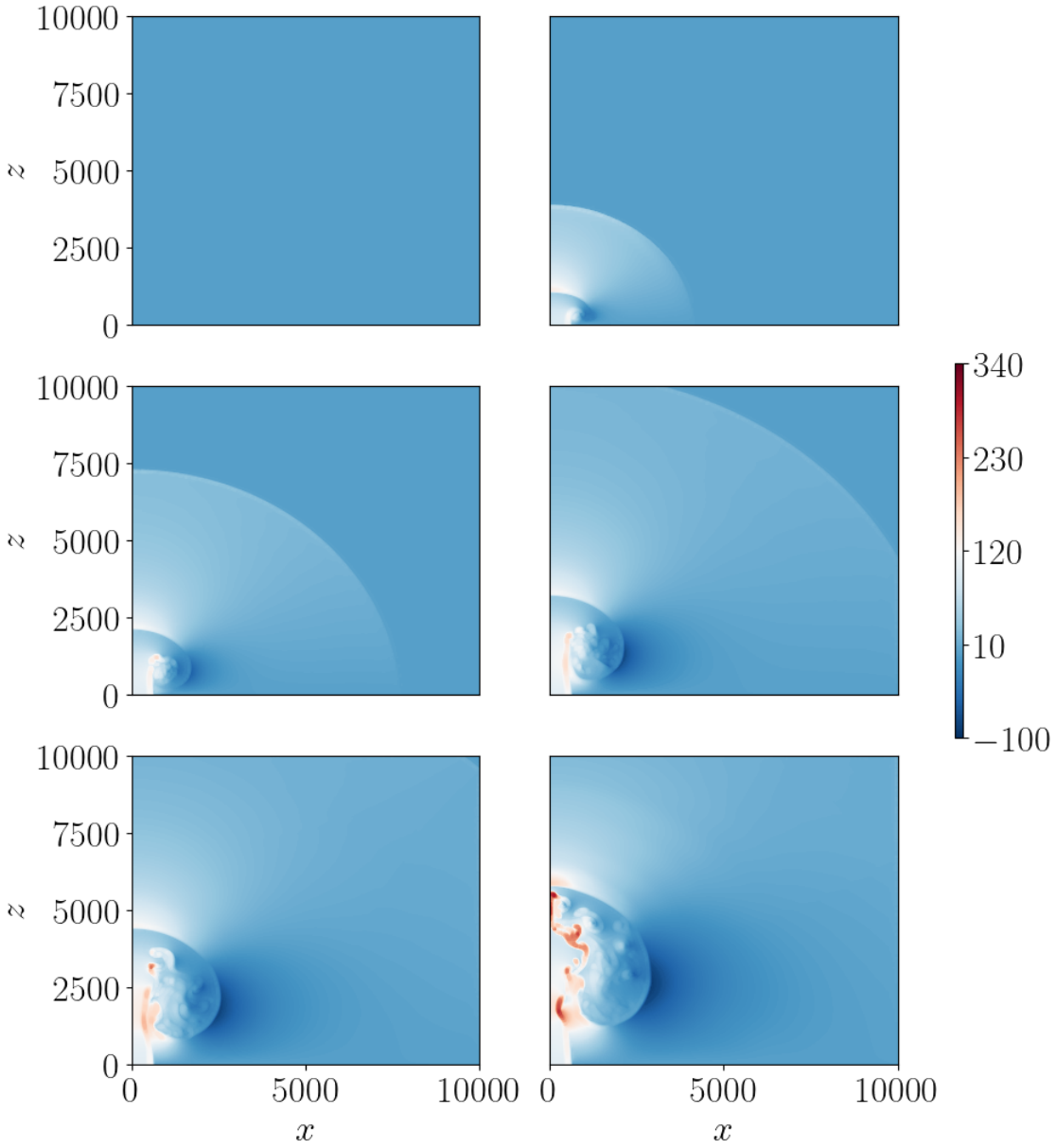

Figure 39: Vertical momentum for the uniform run of the volcanic jet without water vapor and ash with T_m = 1053 K. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

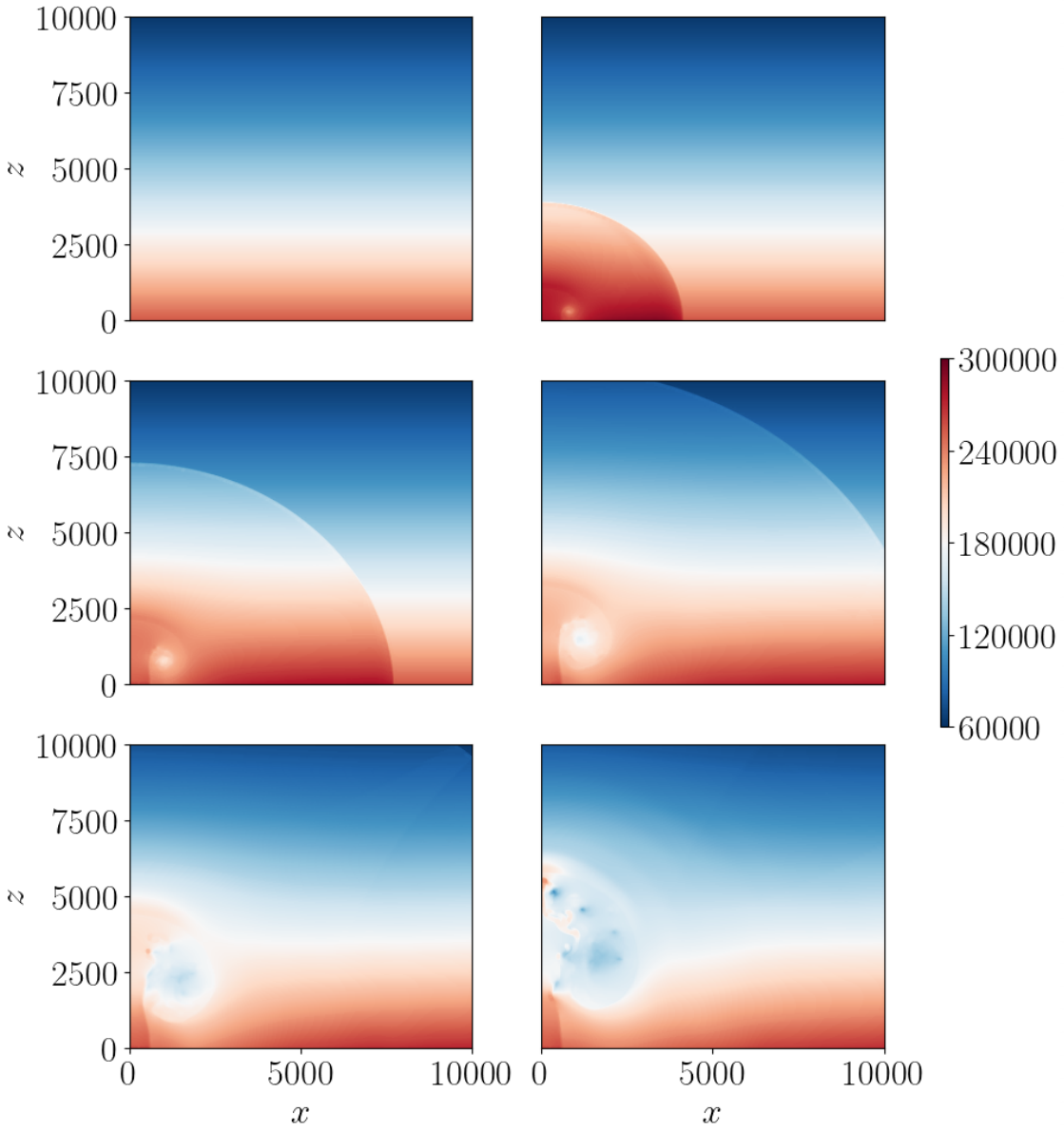

Figure 40: Energy density for the uniform run of the volcanic jet without water vapor and ash with T_m = 1053 K. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

Table 11: CPU times for all setups for the volcanic jet without water vapor and ash with $T_m = T_{\text{ref}} = 1053$ K. Additionally, the percentage of the CPU time of the uniform runs is provided, as well as speed up, the refinement/coarsening tolerances and additional refinement layers.

	CPU time [s]	percentage of uniform run	speed up	σ_{ref}	σ_{coarse}	additional layers of refinement
uniform	303700					
Case D	183900	60.553	1.651	0.01	0.005	3
Case E	175400	57.754	1.731	0.01	0.01	2

The following cases are used:

- Case D: $\sigma_{\text{ref}} = 0.01$, $\sigma_{\text{coarse}} = 0.005$, 3 extra layers of refined cells;
- Case E: $\sigma_{\text{ref}} = 0.01$, $\sigma_{\text{coarse}} = 0.01$, 2 extra layers of refined cells.

For the adaptive cases, similar to the previous setup, a square area directly above the vent is refined to the highest spatial resolution. In this case however, the square has a side-length of 1000 m (instead of 625 m).

As **Table 11** shows, the adaptive cases for the jet with hot air has a much smaller speed up (with factors of 1.651 and 1.731, respectively) than the previous case where the jet has the same temperature as the surrounding atmosphere (where the factors are 8.078 and 1.877, respectively). This is due to the fact that the adaptive grid is refined much more compared to the previous case. While the speed up is smaller, the simulations with the adaptive grid resemble the uniform solutions which, ultimately, is the desired result which is still achieved much faster than the uniform solution.
Figure 41 shows the comparison between horizontal momentum for the uniform run, Case D and Case E. As previously mentioned, the adaptive results resemble the uniform results. Just as previously, the horizontal momentum is used for the comparisons because deviations are most visible in this variable. While there are minimal deviations between the adaptive and uniform results, this is likely explained with the non-linearity of the equation set. Compared to the jet where $T_m = 300$ K, however the deviations between the adaptive and uniform run are negligible for this case of hot air. Since Case D and Case E do not really differ drastically, using the setup that has the better speed up factor is more advantageous. Consequently, for this jet setup, a minimum speed up factor of at least 1.731 can be expected.

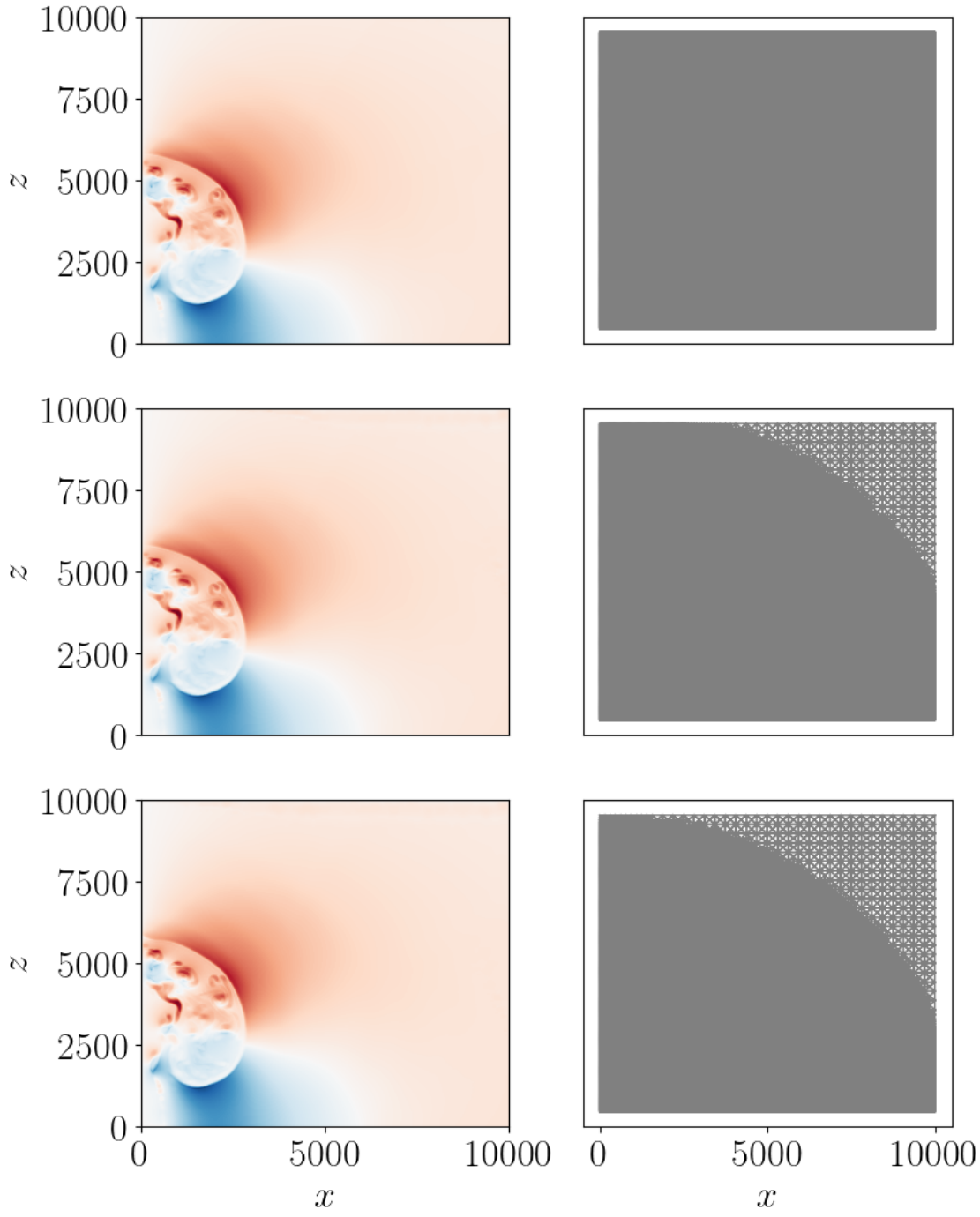

Figure 41: Comparison between uniform and adaptive runs for the volcanic jet without water vapor and ash with T_m = 1053 K. On the left, the horizontal momentum is shown after $t_{\max}$ while the right shows the corresponding grid. The colormap is setup just as in the previous (uniform) figures. The top rows shows the uniform results, the middle row shows the results for Case D and the bottom row shows the results for Case E. The adaptive results resemble the uniform closely while there are minimal deviations. The adaptive grid fully captures the jet.

8.4 Volcanic jet with water vapor but without ash

For this setup, most of the previous case without water vapor and ash is taken over with the only exception being that the erupted material mass fraction ξ is set to one. For this case, the fraction of erupted gas at the vent ξ_{g0} is set to 1 since no ash will be erupted for this setup. In the following, two different limits for the density will be used. The limiting/filtering for the simulation will be performed as previously but with this setup, the density will be cut off at 0.05 kg m^{-3} on the one hand, while it will be cut off at 0.1 kg m^{-3} on the other hand.
The results for this setup are obtained using an Intel i5-8350U CPU (instead of an Intel Core i7-7500U).

Limit 1: density limited at 0.05 kg m^{-3}
Figure 42, **Figure 43**, **Figure 44**, **Figure 45** and **Figure 46** show the results for the uniform run for **Limit 1**.
As these figures show, the results look different if the erupted material is comprised of water vapor instead of (dry) air. The water vapor-air mixture for **Limit 1** spreads out over a larger area than the results for dry air. Interestingly, the plots show the development of shock fronts after 40 seconds. This is likely caused by very high velocities (about 900 m s^{-1}) which are physically implausible and is probably a result of the lack of diffusion - be that large numerical or physical diffusion. Apart from that, the behavior of the erupted cloud seems plausible.

For the adaptive runs, several combinations of criteria are used with the error indicator being the same as before ($\mathbf{e} = \max|\nabla(\rho\mathbf{u})|$). Unfortunately, for this setup, the error indicator is not a good choice since for some reason, the adaptivity does not really work that well and runs sometimes become either unstable or the grid remains almost constant throughout the run (similar to Case C). The runs are stable for two setups where the coarsening was disabled and the refinement criterion σ_{ref} was set to 0.001. The setups differ in the additional layers of refinement that are added but in the end, after roughly 4 seconds of simulated time, no further grid refinement is performed. Consequently, only one case will be shown here:

- Case F: $\sigma_{\text{ref}} = 0.001$, no coarsening, 20 layers of refined cells.

The comparison between uniform and adaptive run for **Limit 1** are shown in **Figure 47** while the CPU times and speed up are shown in **Table 12**. The adaptive results are, as per usual, shown in **Appendix E** While a calculation of speed up is possible, in this case, it is not really of use since the adaptive results do not capture the volcanic jet because parts of the plume have resolution where the grid has its coarsest possible resolution (625 m), similar to Case C. The results show that for a more realistic setup (such as this jet with water vapor), a better refinement criterion is required.

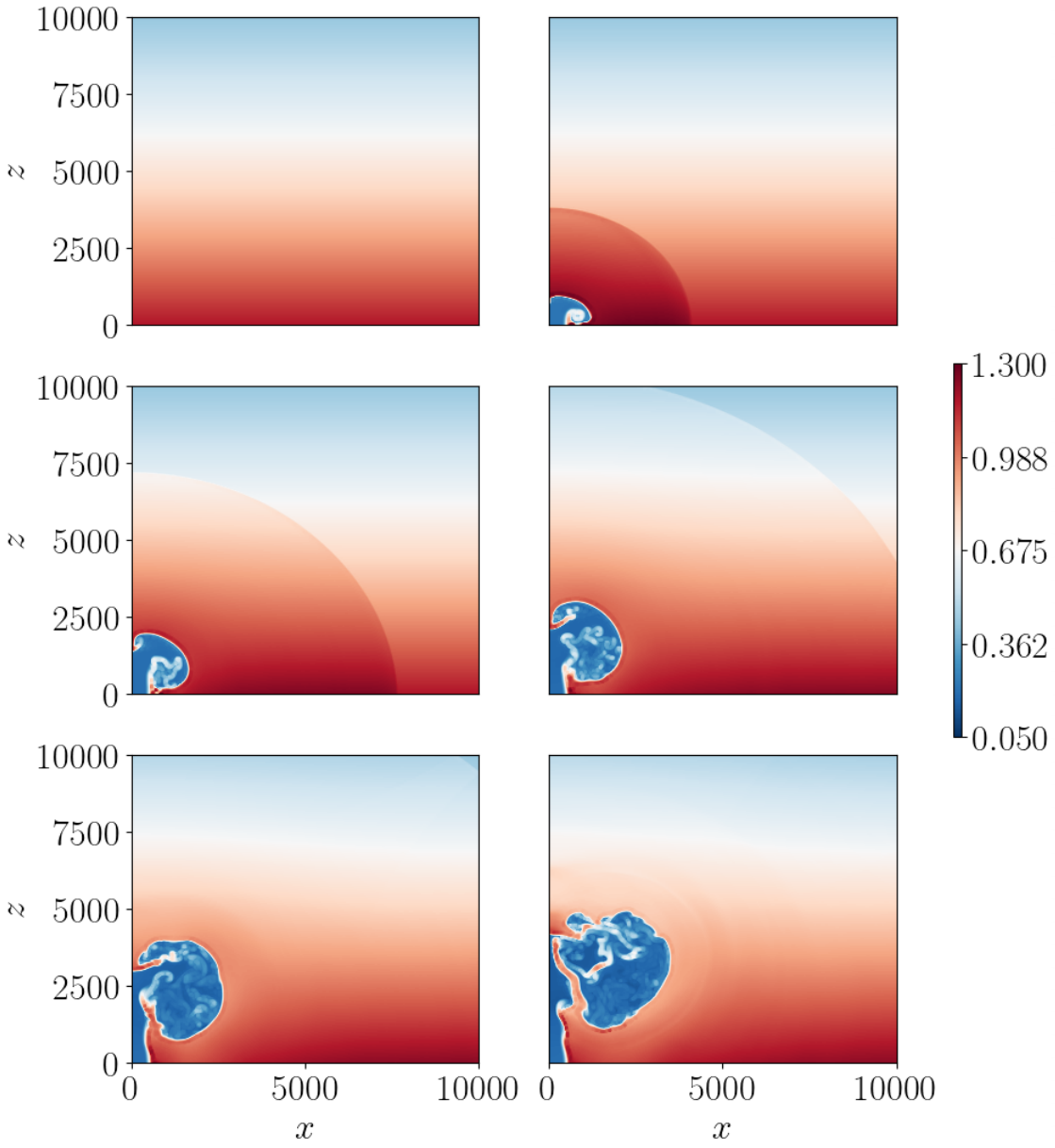

Figure 42: Density for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 1). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

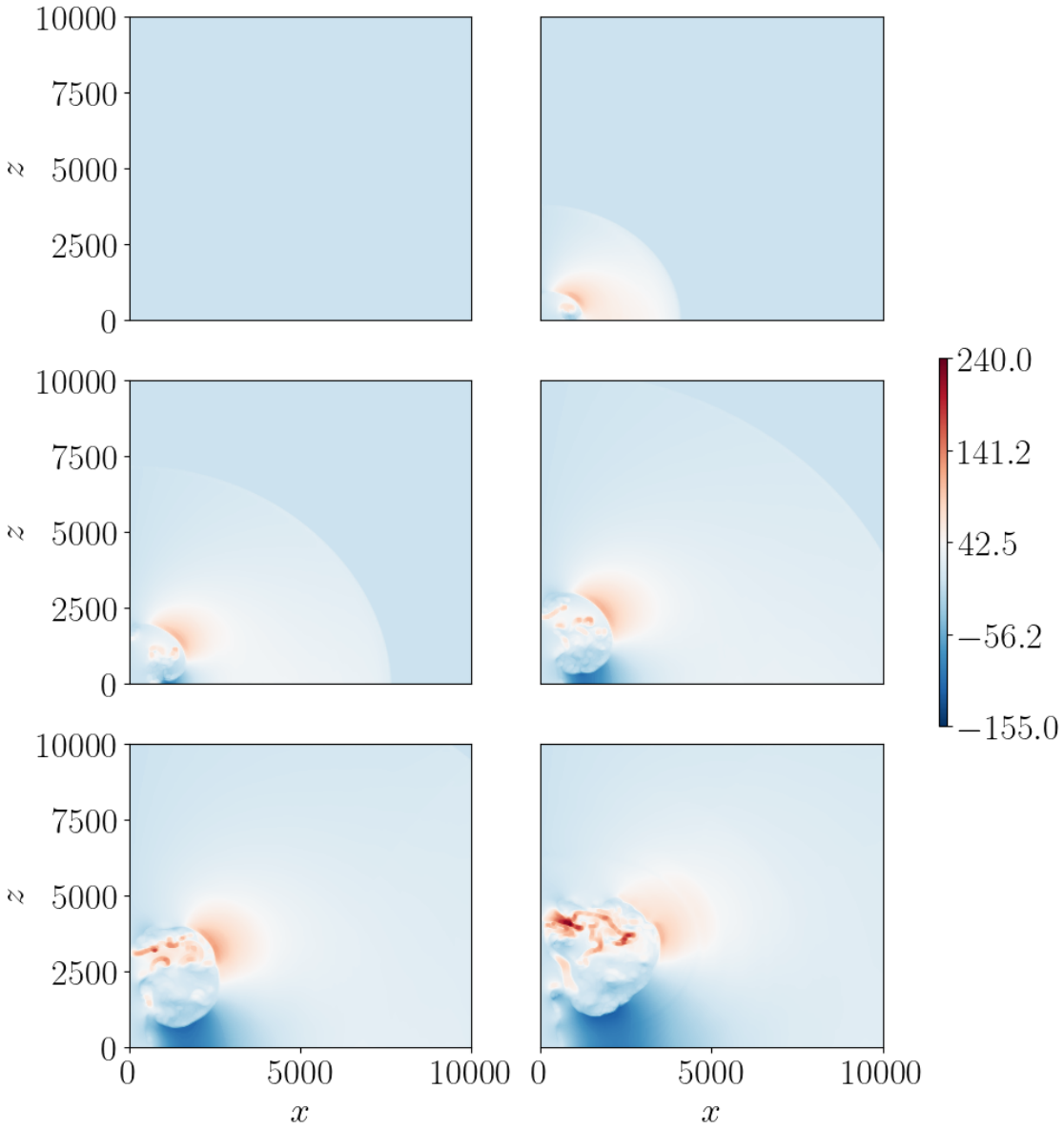

Figure 43: Horizontal momentum for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 1). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

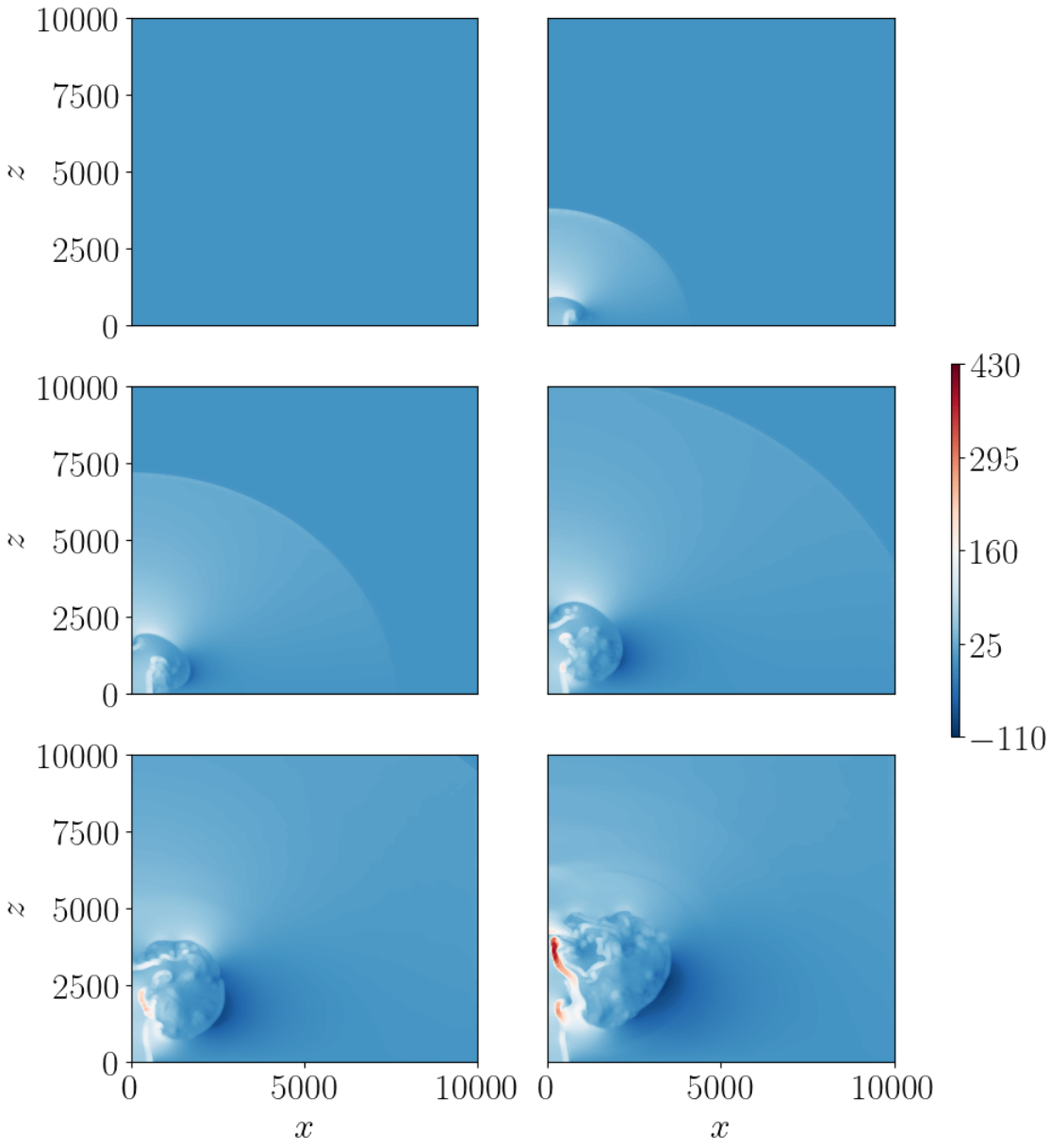

Figure 44: Vertical momentum for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 1). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

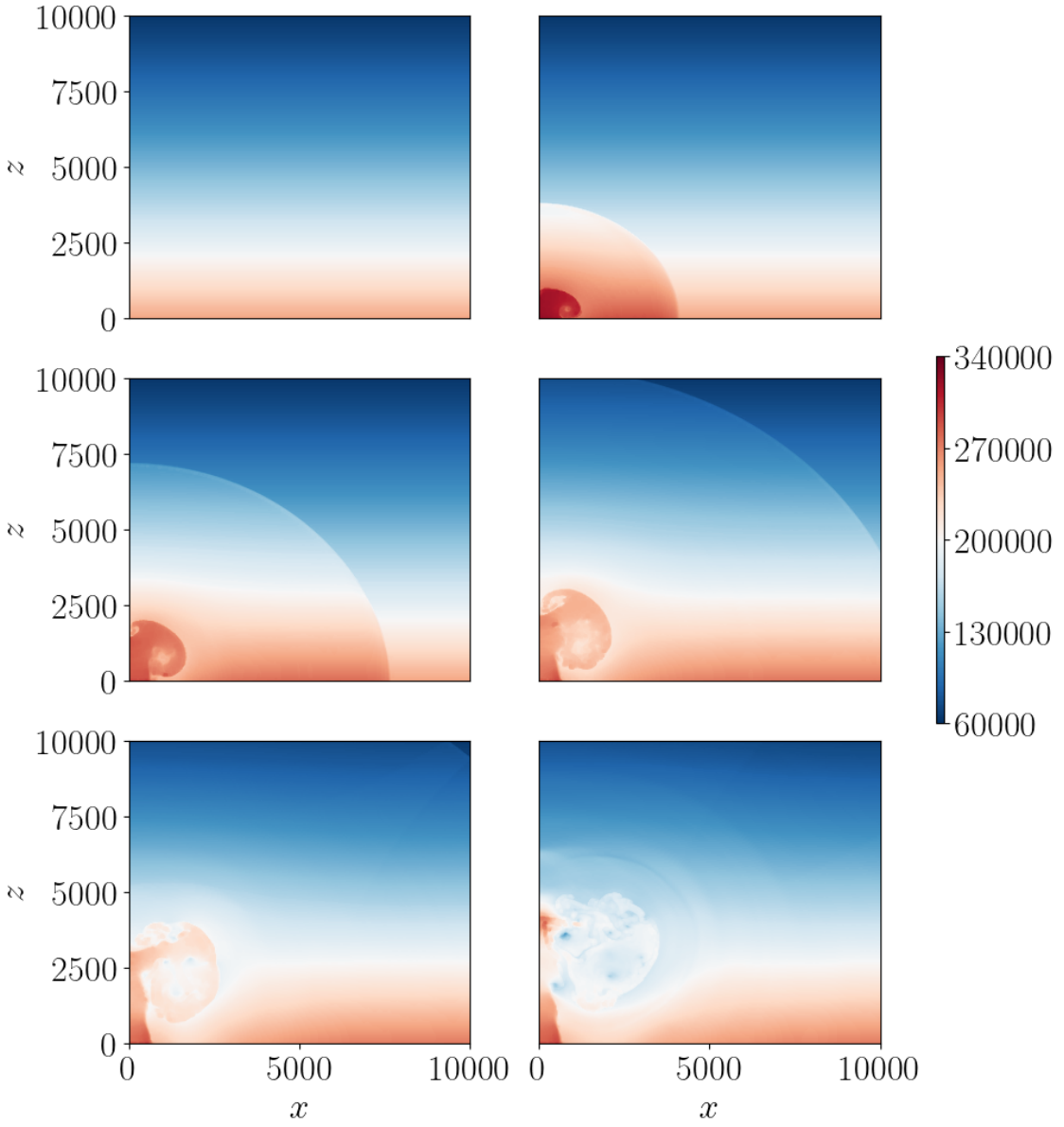

Figure 45: Energy density for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 1). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

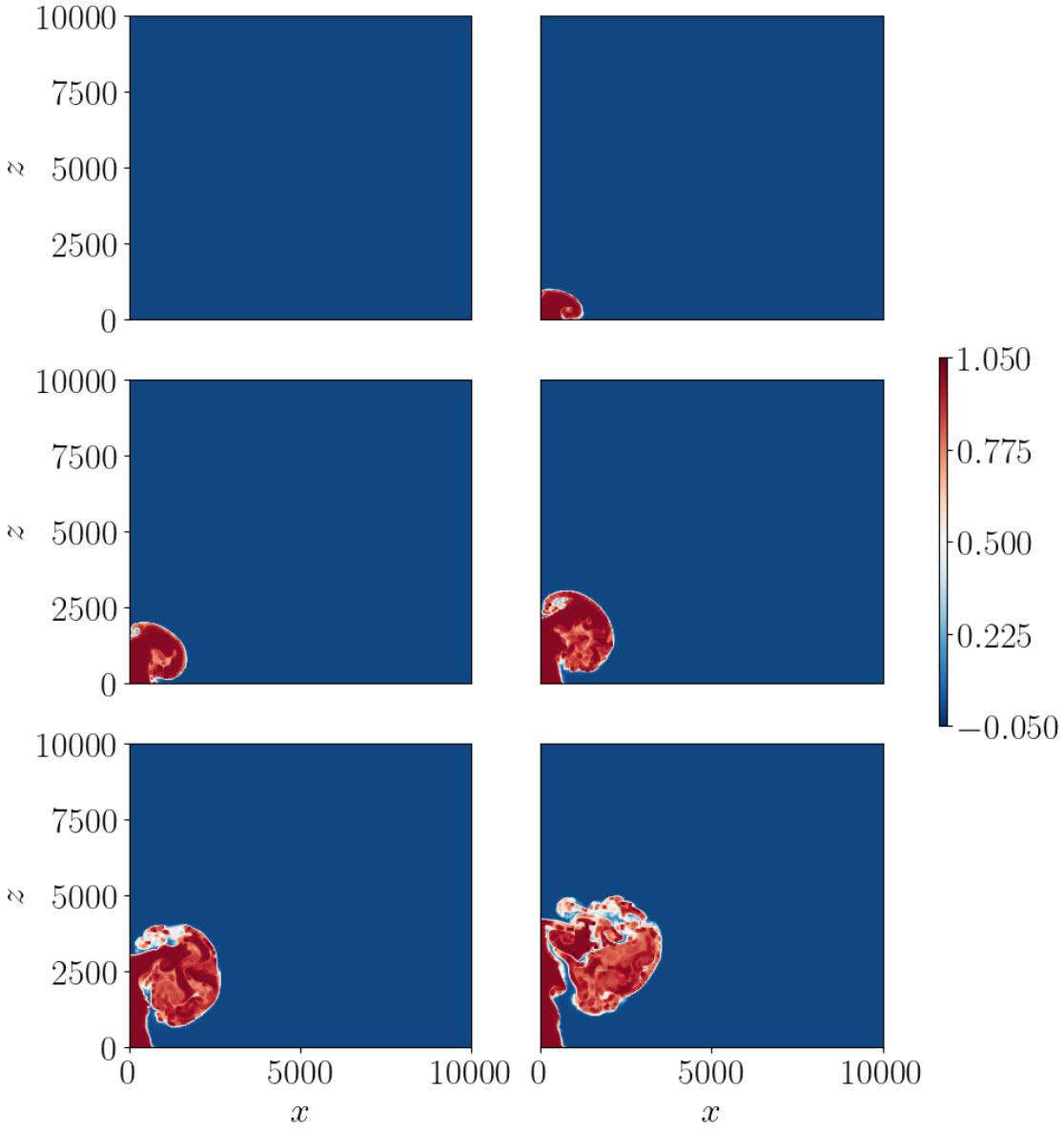

Figure 46: Mass fraction ξ of erupted material for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 1). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally.

Table 12: CPU times for the volcanic jet with water vapor and no ash with $T_m = T_{\text{ref}} = 1053$ K (Limit 1). Additionally, the percentage of the CPU time of the uniform runs is provided, as well as speed up, the refinement/coarsening tolerances and additional refinement layers. Note that the values are obtained using an Intel i5-8350U CPU.

	CPU time [s]	percentage of uniform run	speed up	σ_{ref}	σ_{coarse}	additional layers of refinement
uniform	285500					
Case F	18410	6.448	15.508	0.001	-	20

Limit 2: density limited at 0.1 kg m^{-3}

Figure 48, **Figure 49**, **Figure 50**, **Figure 51** and **Figure 52** show the results for the uniform run for **Limit 2**.
The plots show similar behavior to **Limit 1** as the development of the water vapor jet is similar. While some areas exhibit deviations if the results are studied in detail, the general shape of the clouds that the jets produce are comparable qualitatively. However, the occurrence and development of shock fronts for **Limit 2** seems to be not as pronounced as for **Limit 1**. Still, shock appear for **Limit 2** as the velocities are too large in certain areas which is physically implausible.

For the adaptive setups, the previous error indicator is changed since **Limit 1** showed that the maximum of the gradient of the momenta was not well suited. Since the jet with water vapor is set up to have mass fraction of ejected material ξ that is larger than 0, an idea for tracking the jet correctly with adaptive meshes is to use the mass fraction as indicator since regions where ξ is larger than 0 mark regions where the jet is present. For that, two cases - Case G and Case H - are used. In both cases, the grid is refined if $\rho\xi > 0.2$, while additionally, $\mathbf{e} = \nabla(\rho\xi)$ is used as error indicator for Case G while $\mathbf{e} = \nabla(\xi)$ is the error indicator for Case H. Regarding the tolerances, Case G and Case H are set up as follows:

- Case G: $\sigma_{\text{ref}} = 0.75$, $\sigma_{\text{coarse}} = 0.0001$, 8 layers of refined cells.
- Case H: $\sigma_{\text{ref}} = 0.1$, $\sigma_{\text{coarse}} = 0.05$, 8 layers of refined cells.

This setup for the adaptive mesh refinement very quickly refines the grids for both cases. Interestingly, the grid is quickly refined very finely in regions where the jet is not present at all. While an initial worry was that the mesh refinement would not track the acoustic wave, this behavior is not observed for the grid. This results, however, in a rather large computational time. Still, a speed ups of about 1.8 (Case G) and 1.7 (Case H) are achieved, as can be seen in **Table 13**.

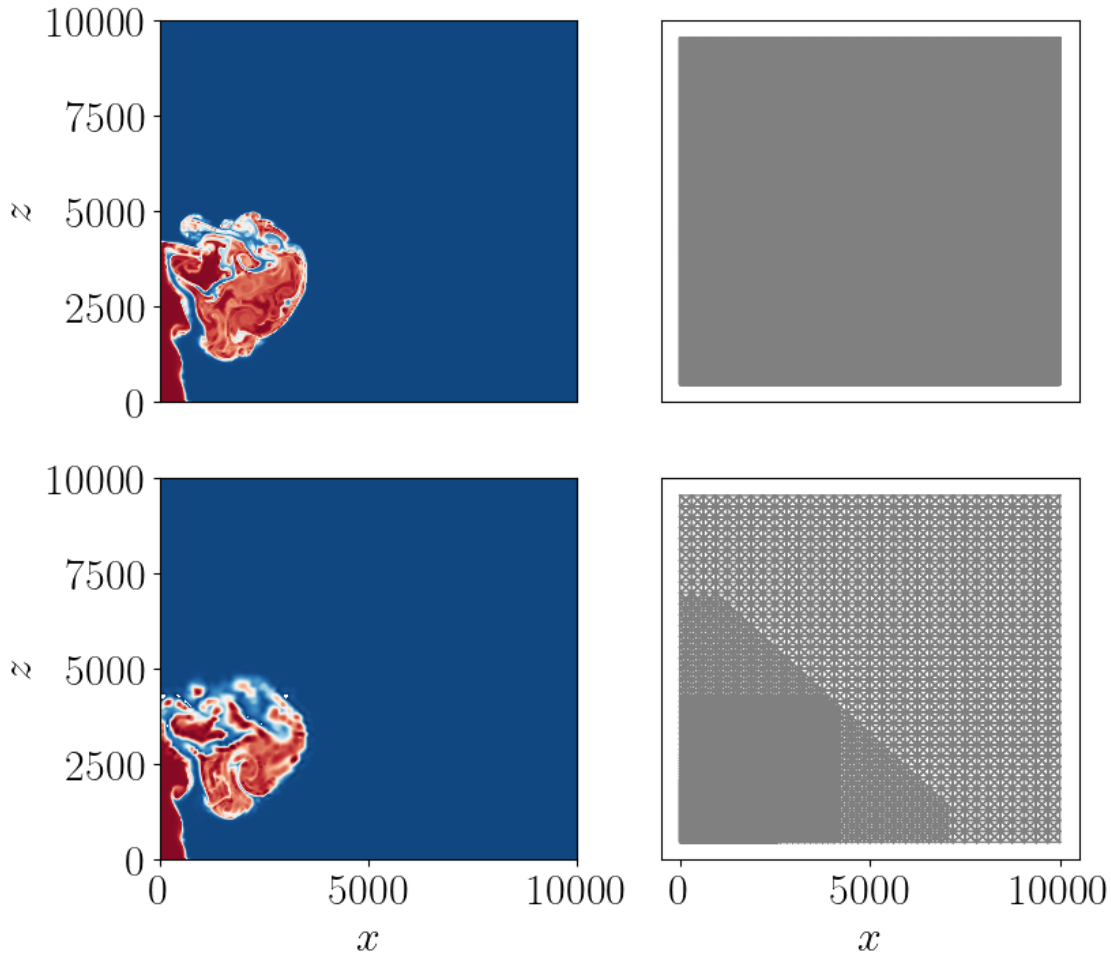

Figure 47: Comparison between uniform and adaptive runs for the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 1). On the left, the mass fraction ξ is shown after $t_{\max}$ while the right shows the corresponding grid. The colormap is setup just as in the previous (uniform) figures. The top rows shows the uniform results and the bottom row shows the results for Case F. The adaptive results differ from the uniform results since the grid does not capture the jet fully.

Table 13: CPU times for the volcanic jet with water vapor and no ash with $T_m = T_{\text{ref}} = 1053$ K (Limit 2). Additionally, the percentage of the CPU time of the uniform runs is provided, as well as speed up, the refinement/coarsening tolerances and additional refinement layers.

	CPU time [s]	percentage of uniform run	speed up	σ_{ref}	σ_{coarse}	additional layers of refinement
uniform	285500					
Case G	158500	55.517	1.801	0.75	0.0001	8
Case H	164600	57.653	1.735	0.1	0.05	8

Figure 53 shows the comparison between the uniform and adaptive runs. The results from Case H resemble the uniform results better than the results from Case G because the mesh for Case G has coarser regions close to the jet. Still, both adaptive cases do not match the uniform results completely which is a similar result to the jet with hot air. The non-linearity of the equation set can lead to different results if the grid for the adaptive cases is not as finely refined as it would be in the uniform run. However, qualitatively, the results from the adaptive runs compare to the ones from the uniform run.

8.5 Volcanic plume which includes water vapor and ash

For the plume test case, all of the parameters from **Limit 2** from the previous jet with water vapor are used, with the only exception that now, the fraction of erupted gas at the vent $\xi g0$ is set to a quantity other than 1.
The first tries for more realistic volcanic plumes with large ash content (over 90% of ash) did not yield stable results, so the idea was to increase the ash content in the eruption with each consecutive setup. For this, first a plume with a mass fraction of 25% ash, 75% water vapor was run, followed by a simulation with 50% ash, 50% water vapor. Unfortunately, in both cases the simulation did not yield very plausible results. While the simulations are stable, the results - especially for the density - after $t_{\text{max}} = 50$ s do not seem realistic, with the maximum density being about 1150 kg m^{-3} and 780 kg m^{-3}, respectively. Additionally, the other variables exhibit behavior that does not seem physically plausible.
Figure 54 presents the mass fraction after t_{max} for both plumes which shows the physically implausible results. Due to this implausibility, no other results from these runs will be shown.

Since the densities for both these plumes have very large maxima, similar to the previous jet test cases, the idea for a very primitive approach arose to just give an upper limit

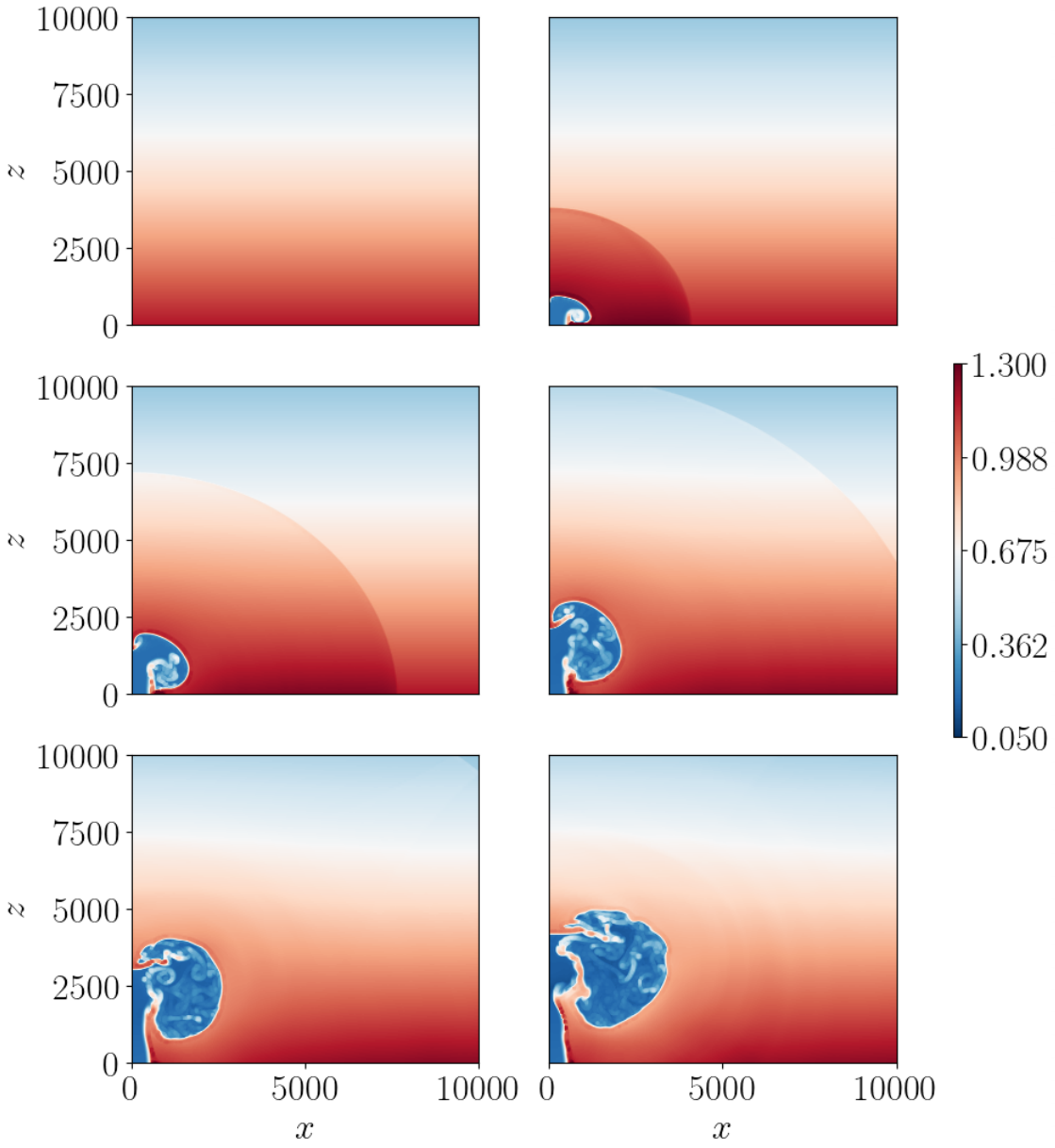

Figure 48: Density for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 2). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

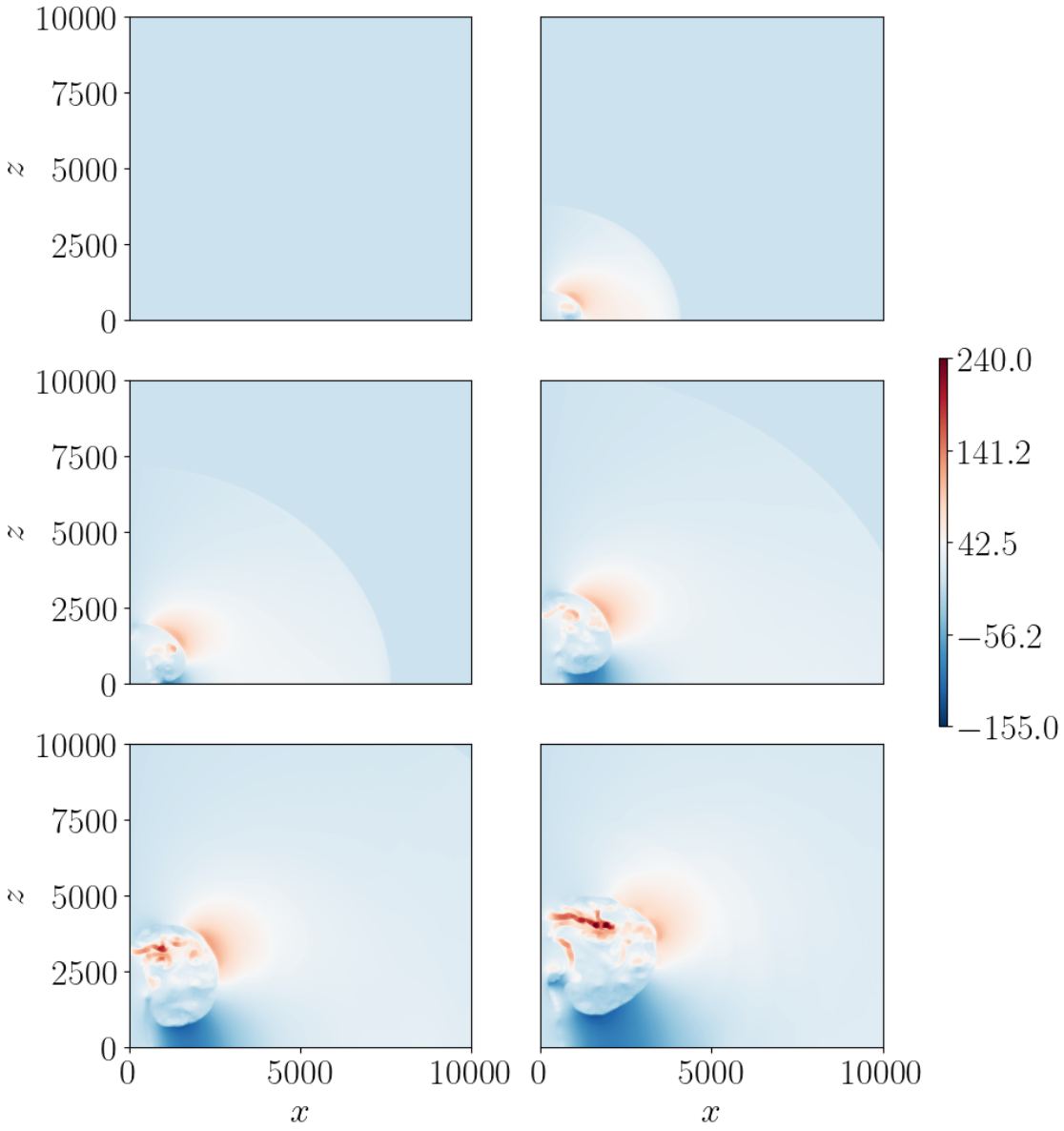

Figure 49: Horizontal momentum for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 2). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

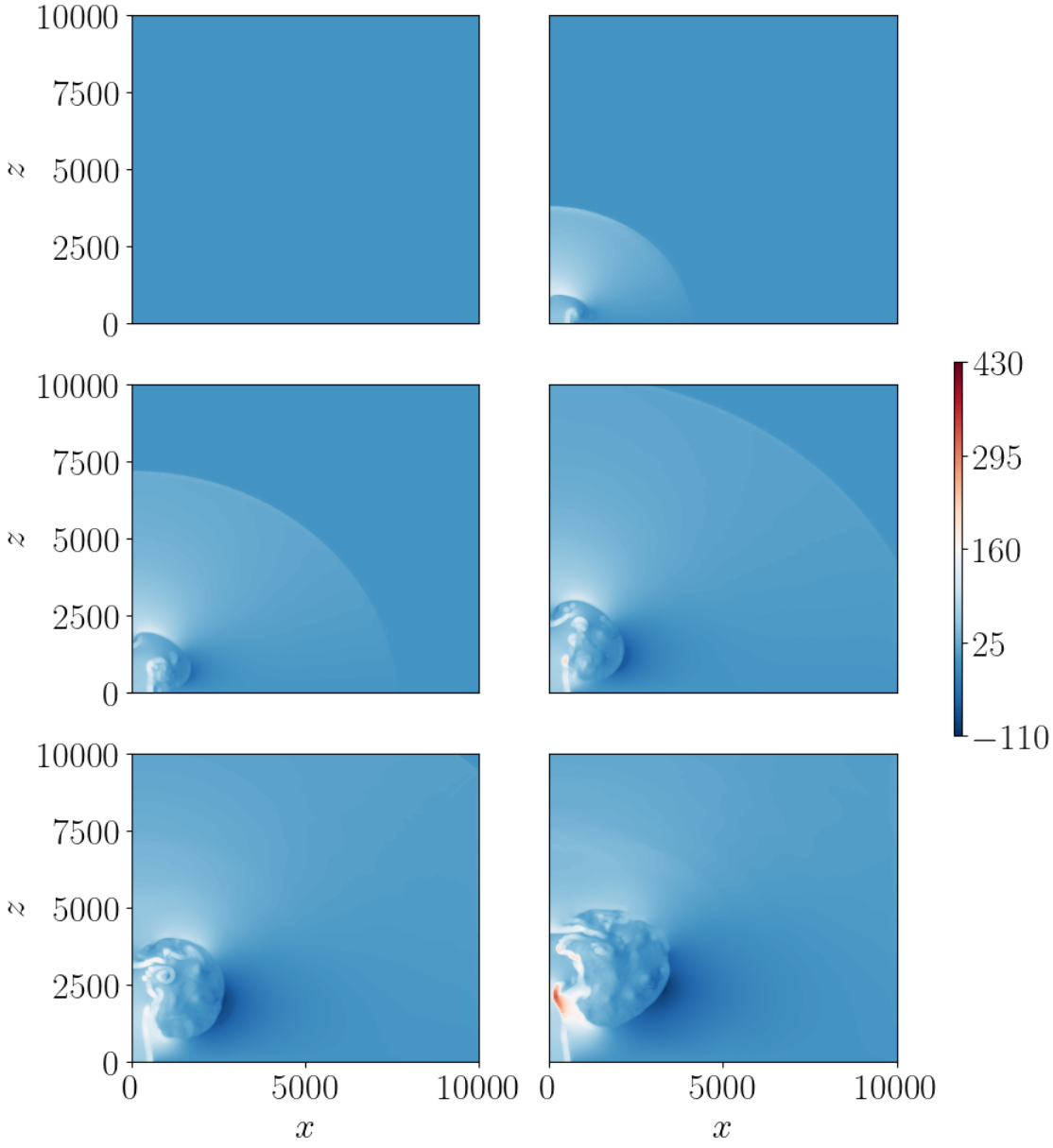

Figure 50: Vertical momentum for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 2). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

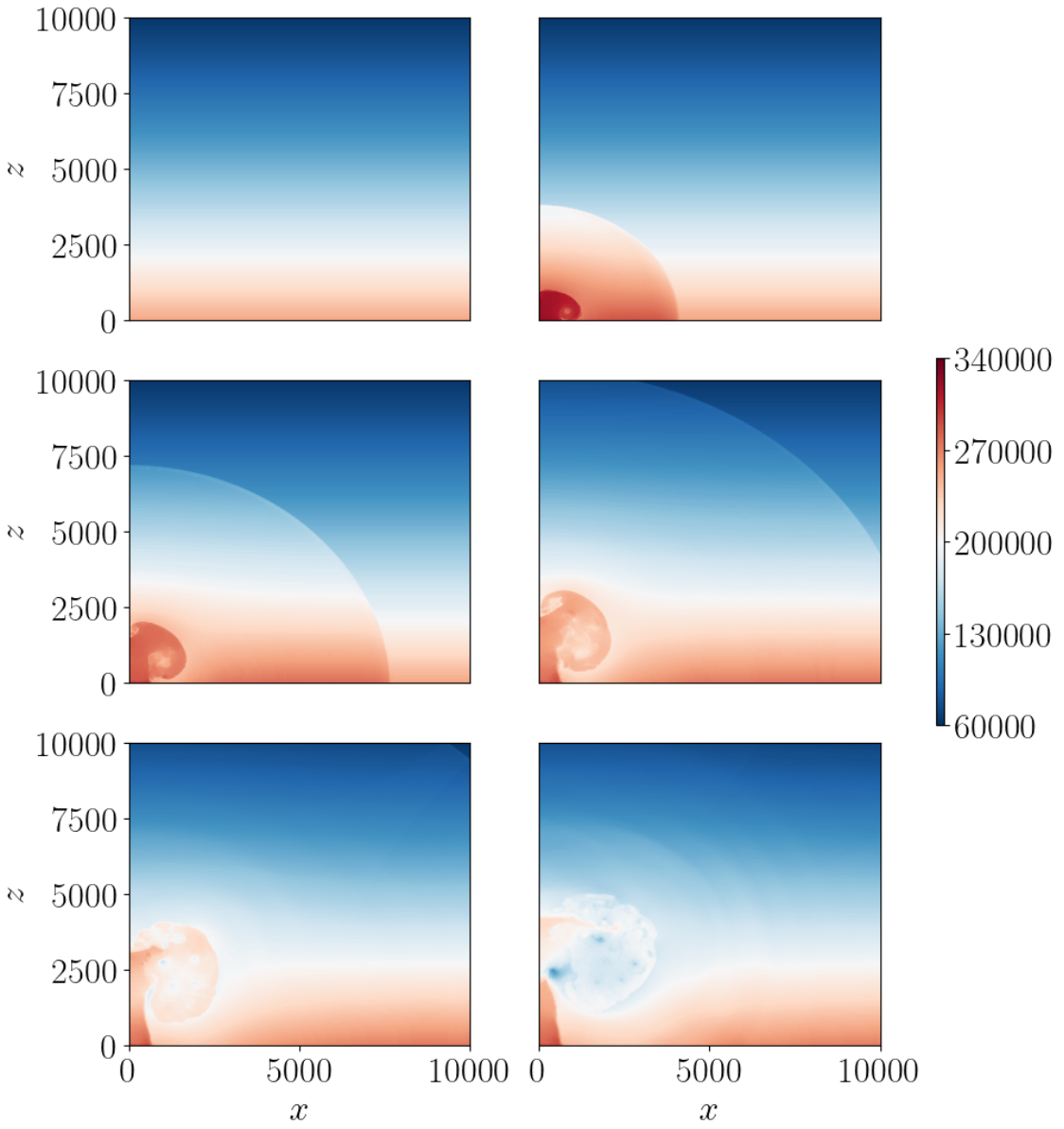

Figure 51: Energy density for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 2). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

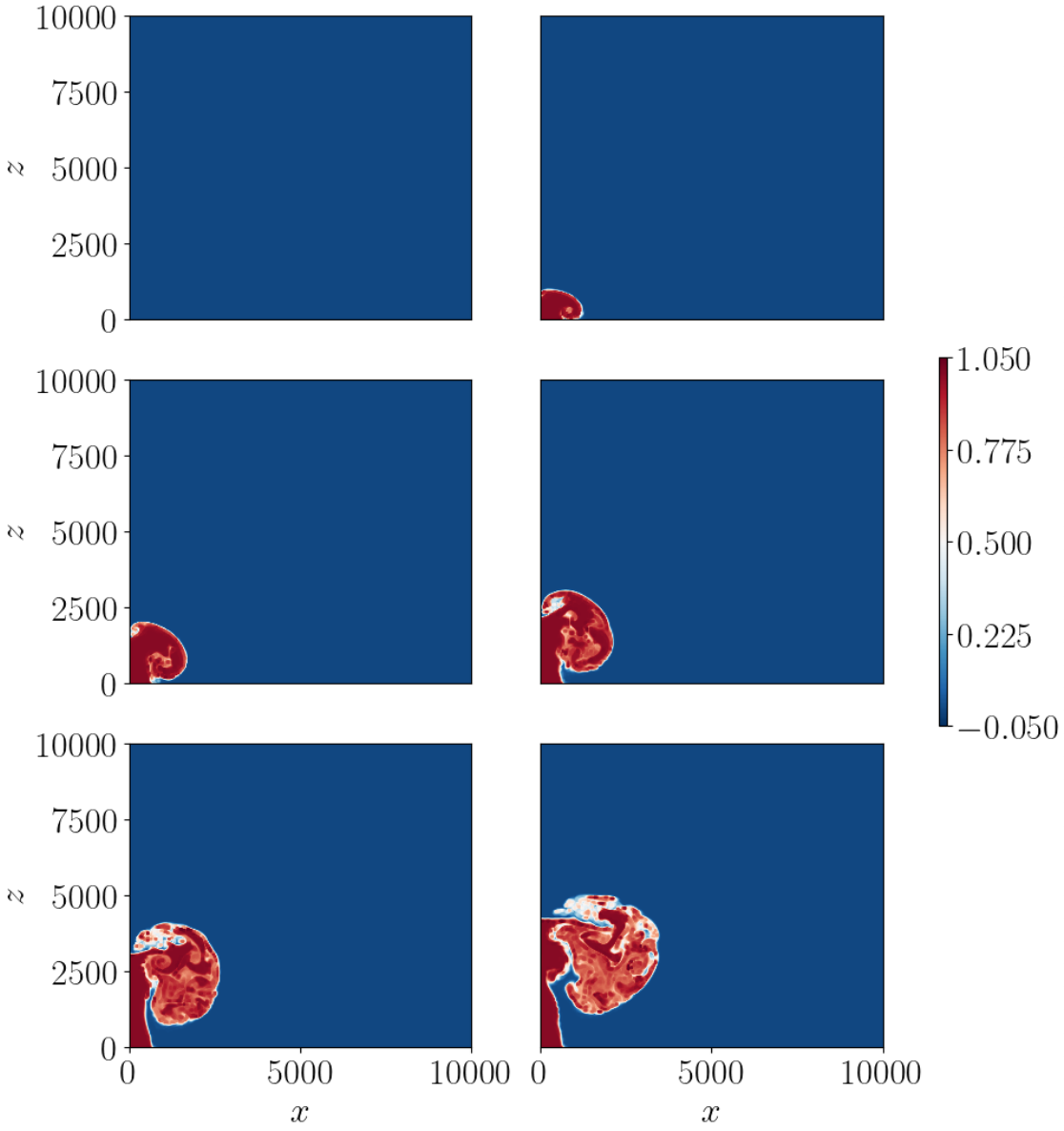

Figure 52: Mass fraction of erupted material for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 2). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally.

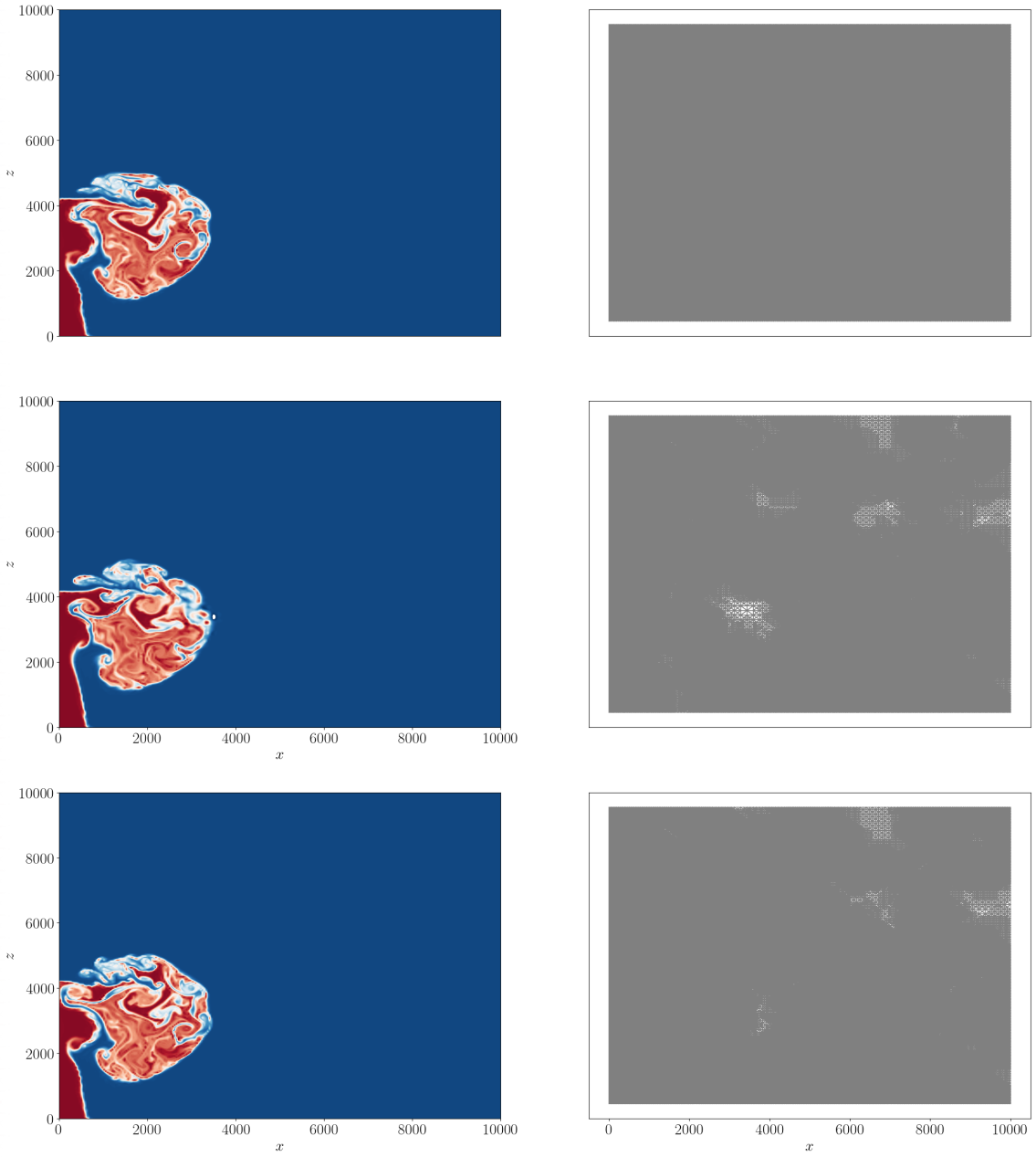

Figure 53: Comparison between uniform and adaptive runs for the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 2). On the left, the horizontal momentum is shown after $t_{\max}$ while the right shows the corresponding grid. The colormap is setup just as in the previous (uniform) figures. The top rows shows the uniform results, the middle row shows the results for Case G and the bottom row shows the results for Case H. The adaptive results resemble the uniform closely while there are minimal deviations. The adaptive grids almost fully capture the jet but especially for case G some areas have a coarser resolution.

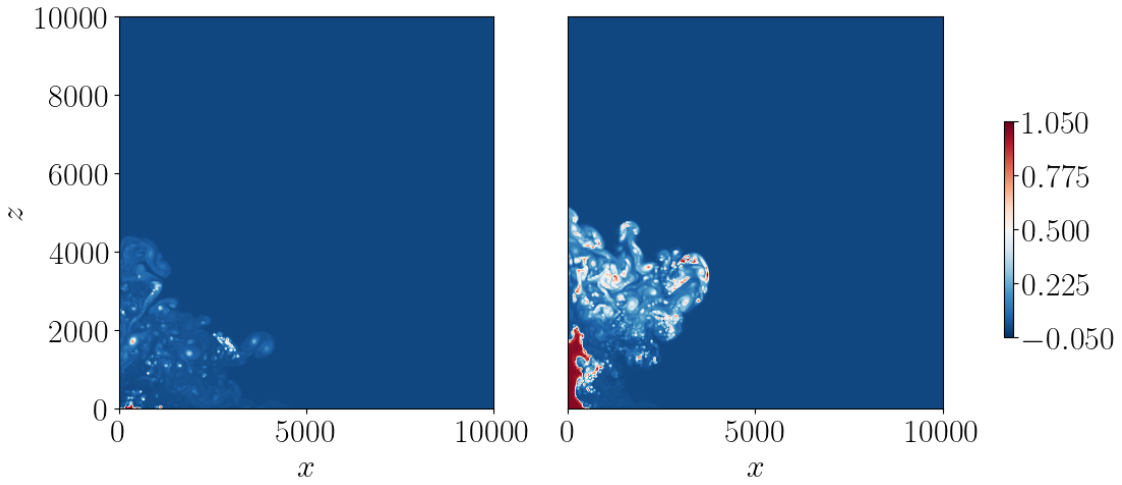

Figure 54: Mass fraction for both unrealistic plumes after $t_{\max}$. The plot on the left shows the result for the plume with 50% ash, 50% water vapor, while the plot on the right shows the result for the plume with 25% ash, 75% water vapor. In both cases, the ash-gas mixture is spread out over a large part of the domain.

for the density as well so that in the end $\rho \in [0.1, 10]$ kg m^{-3}. **Figure 55**, **Figure 56**, **Figure 57**, **Figure 58** and **Figure 59** show the results for this plume run.
As these figures show, the plume starts to ascend into the atmosphere and the entrainment of air after 20 seconds leads to a distribution of the erupted material which is characterized by very localized vortex/eddy structures. The behavior almost seems chaotic and resembles vortex shedding for most variables except for maybe the density even though, due to the equation set, the flow is set up to be inviscid.
The density, on the other hand, does not appear to have such a localized structure and distributes more evenly. However, right at the boundary next to the volcanic vent and above, the density increases drastically. For the plumes cases without upper bound for the density, this is also the region where the density was largest. This leads to the conclusion that in conjunction with a need for more diffusion within the scheme, the behavior at the bottom boundary needs to be investigated. A primitive and temporary fix might be to give an even stricter upper limit for the density which, however, does not tackle the underlying issue of the implementation. Then again, this might allow for runs with larger ash content to be stable.

Regarding the adaptive setup, the previous cases showed that the error indicators did not capture the jets as precisely as expected. Consequently, for the plume setup, the mesh refinement in this case is performed as follows:

- Case I: refine if $\xi > 0.8$, coarsen if $\xi < 0.1$, 15 layers of refined cells.

As usual, the results for the adaptive runs can be found in **Appendix E**. The refinement strategy for Case I shows a great increase in speed up, as shown in **Table 14**, while the

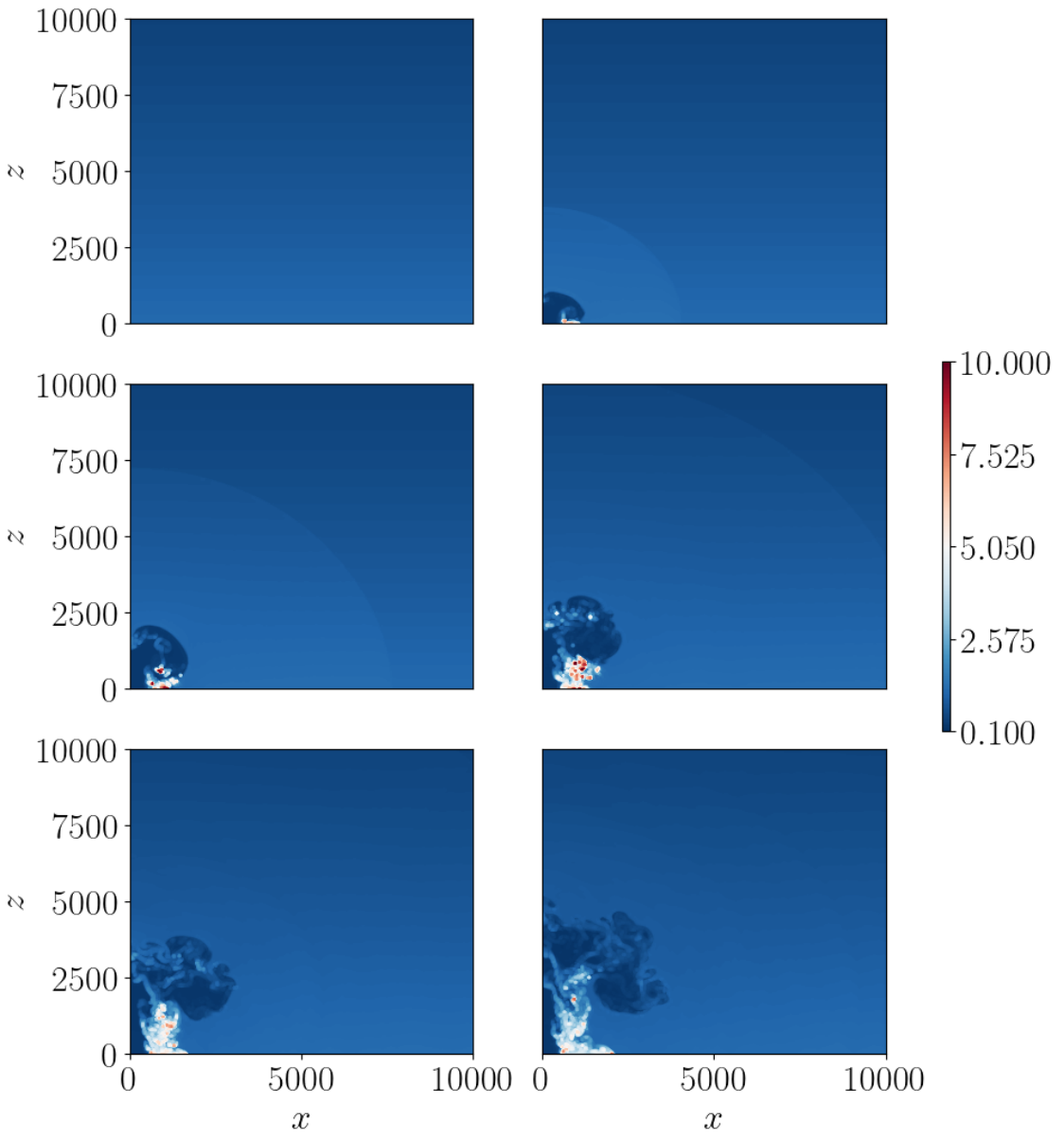

Figure 55: Density for the uniform run of the volcanic plume with T_m = 1053 K. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 5 km. Additionally, there are also portions of the plume that spread out horizontally and that sink down. After 40 seconds, the development of shock fronts around the jet is observable.

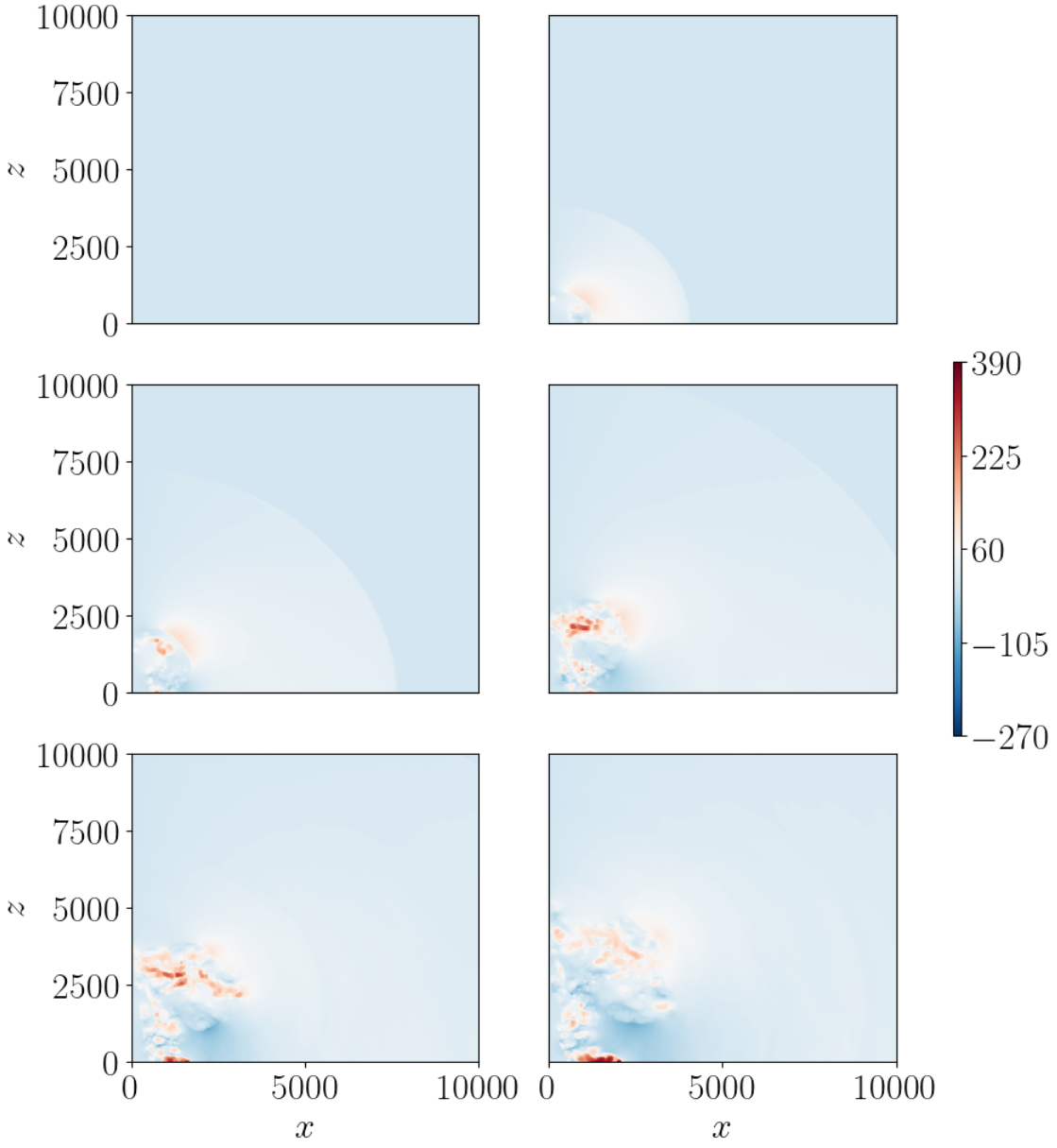

Figure 56: Horizontal momentum for the uniform run of the volcanic plume with T_m = 1053 K. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

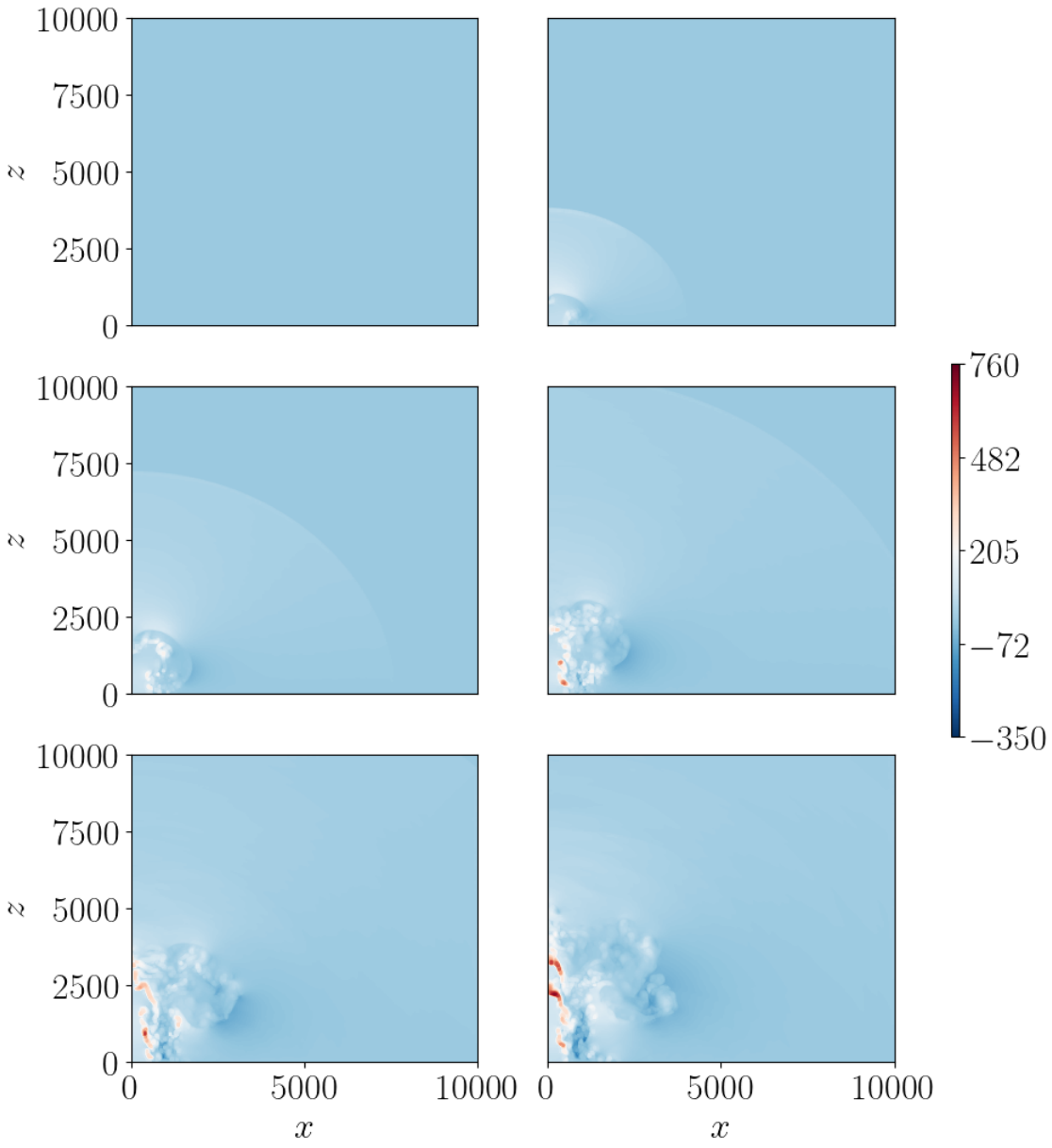

Figure 57: Vertical momentum for the uniform run of the volcanic plume with $T_m = 1053$ K. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

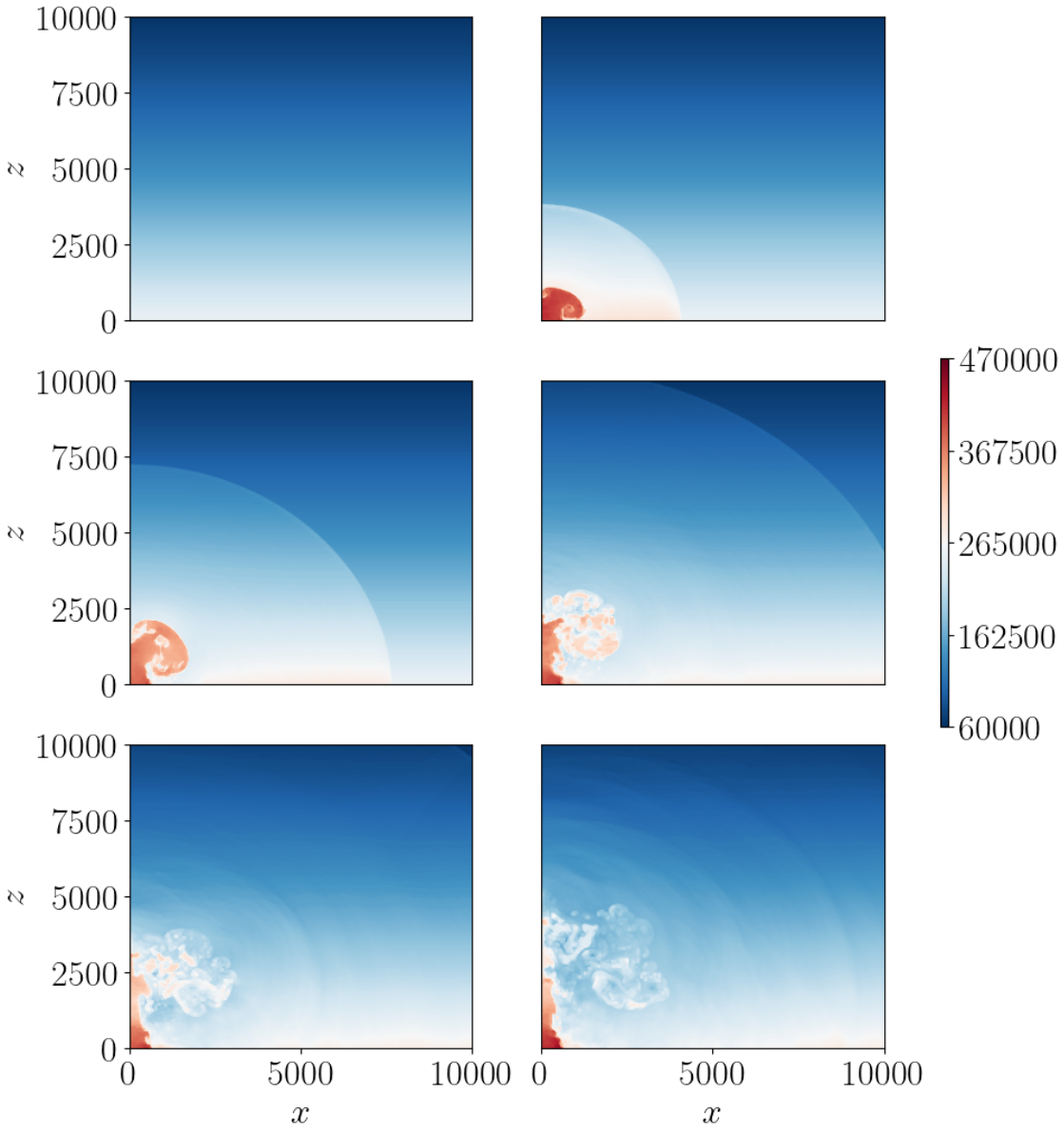

Figure 58: Energy density for the uniform run of the volcanic plume with T_m = 1053 K. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

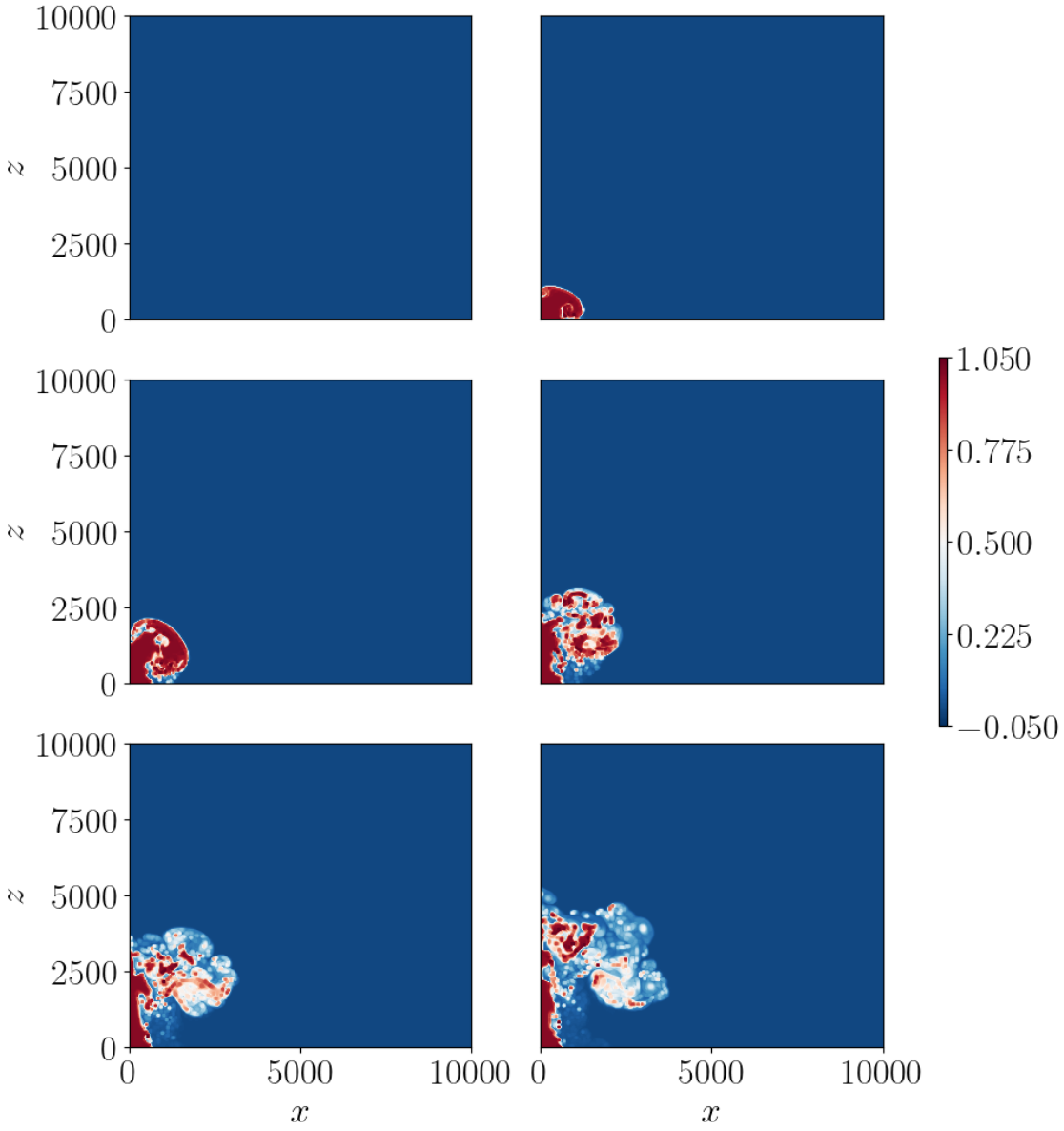

Figure 59: Mass fraction of erupted material for the uniform run of the volcanic jet with water vapor and no ash with T_m = 1053 K (Limit 2). The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally.

Table 14: CPU times for the volcanic plume with $T_m = T_{\text{ref}} = 1053$ K. Additionally, the percentage of the CPU time of the uniform runs is provided, as well as speed up, the refinement/coarsening conditions and additional refinement layers.

	CPU time [s]	percentage of uniform run	speed up	refine if	coarsen if	additional layers of refinement
uniform	440400					
Case I	174400	39.600	2.525	$\xi > 0.8$	$\xi < 0.1$	15

results for the adaptive run differs somewhat to the uniform results (see **Figure 60**). Yet again, the adaptive results are qualitatively comparable to the ones obtained with the uniform setup, which makes this refinement a good option as runs performed with this setup are sped up significantly. To improve the resemblance of adaptive and uniform runs, lowering the refinement threshold for ξ would likely be a valid approach.

8.6 Comparison of the StormFlash results with runs produced with ATHAM

In addition to all the simulations run with StormFlash, the jet with water vapor and both 25% ash, 75% water vapor, as well as 50% ash, 50% water vapor plume cases were run with ATHAM as well. The inputs or parameters for the setups are similar to the ones for the StormFlash cases with the most noticeable difference being the grid and volcano setup. The domain is also a 10 km by 10 km square domain (with some "slices" of a third spatial dimension since this is necessary in order for ATHAM to run) but the volcano is located at the center of the domain, instead of the bottom left corner as for StormFlash. At the vent, the horizontal spatial resolution is 50 m, while the vertical resolution is about 33 m. The vent radius (for these parameters) can only be set to 300 m. For bigger vent radii the simulations become unstable, unfortunately (Matthias Hort, personal communication). For ATHAM, atmospheric parameters have to be provided as to resemble real volcanoes. For this, the atmospheric setup is chosen to resemble the atmosphere for the Pinatubo 1991 eruption. As a reminder, it should be noted that the atmosphere for StormFlash is a rather simple (neutrally stratified) setup.

The simulated eruption emits material for 50 seconds with a "ramp up" and "dampening" over 1 second, respectively, and is run for about 150 seconds in total with an initial time step size of 1 second, where ATHAM then adapts the time step size if necessary.

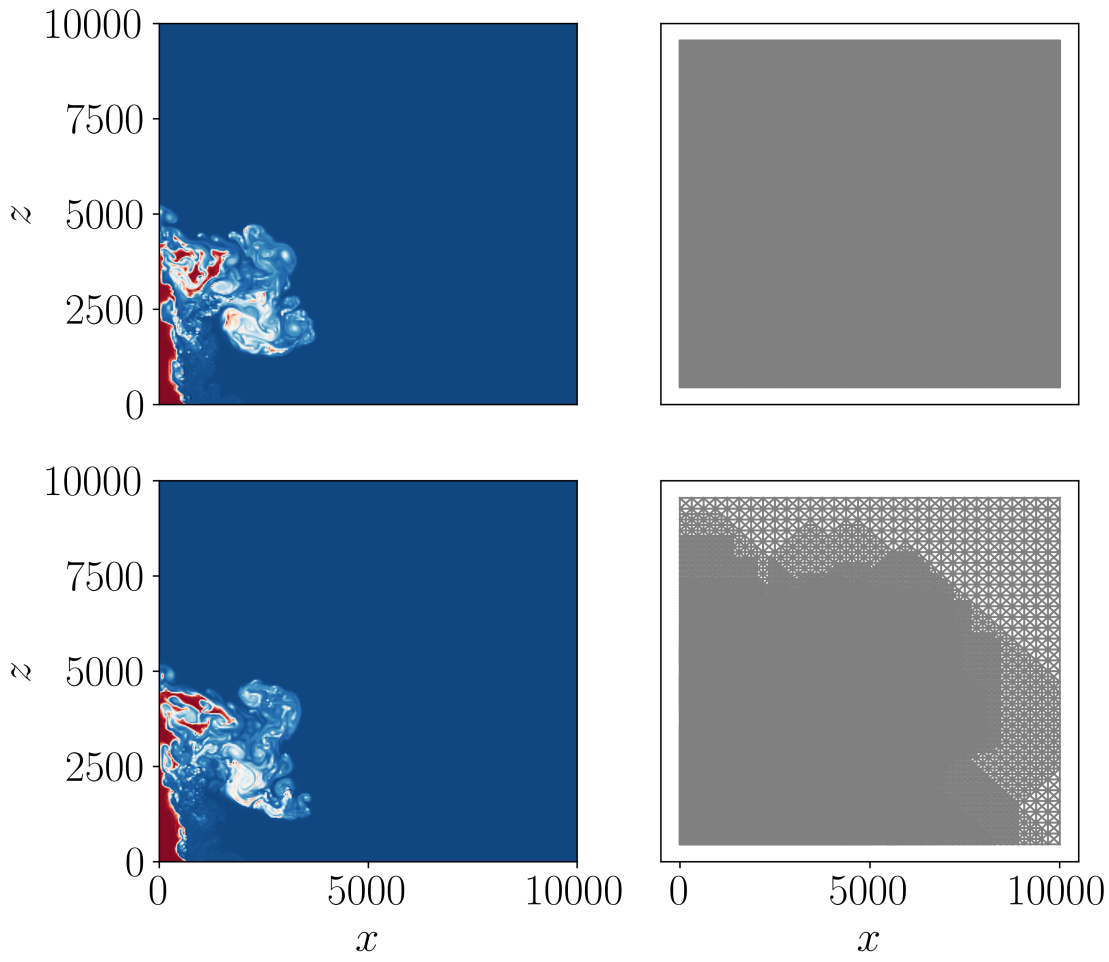

Figure 60: Comparison between uniform and adaptive runs for the volcanic plume and no ash with T_m = 1053 K. On the left, the horizontal momentum is shown after $t_{\max}$ while the right shows the corresponding grid. The colormap is setup just as in the previous (uniform) figures. The top rows shows the uniform results, while the bottom row shows the results for Case I. The adaptive results resembles the uniform while there are small deviations at the top of the plume. The adaptive grid fully captures the plume.

While the output files provide information about many different quantities (e.g. temperature, velocities, concentration of the active tracers), most of the output has a similar shape. Consequently, only the densities from all different runs will be compared. **Figure 61** shows the densities for all three ATHAM runs after (approximately) 50 seconds. While the densities of the jet or the plumes differ slightly (larger ash content leads to a decrease in jet/plume density), the results are very comparable for each of the three ATHAM runs. Additionally, in stark contrast to the results obtained with StormFlash, the density data is very smooth for all three runs.
Since ATHAM is based on some form of the Navier-Stokes equations and utilizes a turbulence closure, a dampening is introduced. This dampening smooths out the solution and is, of course, absent within StormFlash.

In **Figure 61**, it can be observed that the eruptive columns simulated within ATHAM all reach the top boundary of the calculation domain. Since the parameters are chosen mimic the strong plume test case from Suzuki et al. (2016), the eruptive columns having a large height (and consequently the columns reaching the top boundary) is of no surprise. While the velocity data is not shown here, it should be noted that also the runs with ATHAM have velocities that become very large (e.g. about 900 m s^{-1} after 50 s for the 50% ash, 50% water vapor plume case) but these large velocities do not produce incorrect shocks.

Since the runs with ATHAM hit the boundary of the domain at some point, the comparison between the jets and plumes obtained from both ATHAM, as well as StormFlash, are shown for 25 seconds of simulated time (instead of showing the results for $t_{\max}$ = 50 s). **Figure 62** and **Figure 63** show the comparison of the (uniform) StormFlash runs of the jet with water vapor with the one obtained from ATHAM and the comparison of the (uniform) StormFlash runs for the 25% ash, 75% water vapor plume with the one obtained from ATHAM, respectively. Even though the volcanic vent is only half as wide for ATHAM (which leads to smaller mass flux and should consequently lead to a smaller column height), these figures clearly display that the runs with ATHAM produce eruptive columns that are at least 2.5 times higher than the ones obtained with StormFlash. Additionally, the ATHAM results do not show small scale features while the StormFlash results obviously do. This is also very likely part of the explanation why StormFlash produces eruptive columns that have a significantly lower height: the energy cascades down into the smaller scales instead of remaining in the larger scales and driving the upward motion of the column. The primitive limiting of the solution space probably also furthers this procedure.
Both the ATHAM results presented here, as well as all of the results in Suzuki et al. (2016) show that the high plume densities that StormFlash produces are not realistic. Consequently, it is necessary to take a more sophisticated approach for managing the incorrect densities.

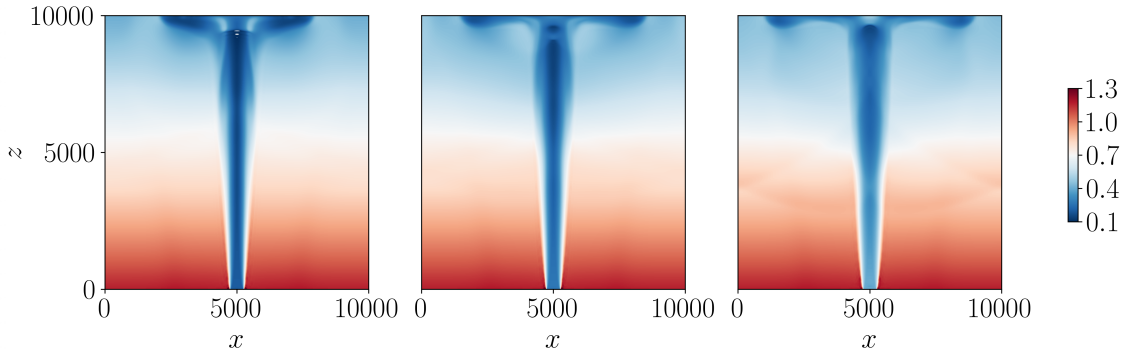

Figure 61: Densities after roughly 50 s for all three ATHAM runs. The plots increase in ash content from left to right, with the left plot being just erupted gas, the middle plot having 25% ash, 75% water vapor and the right plot having 50% ash, 50% water vapor. The colorbar shows values in kg m^{-3}. The eruptive columns all reach the top boundary of the domain while the minimum for the density decreases the larger the ash content.

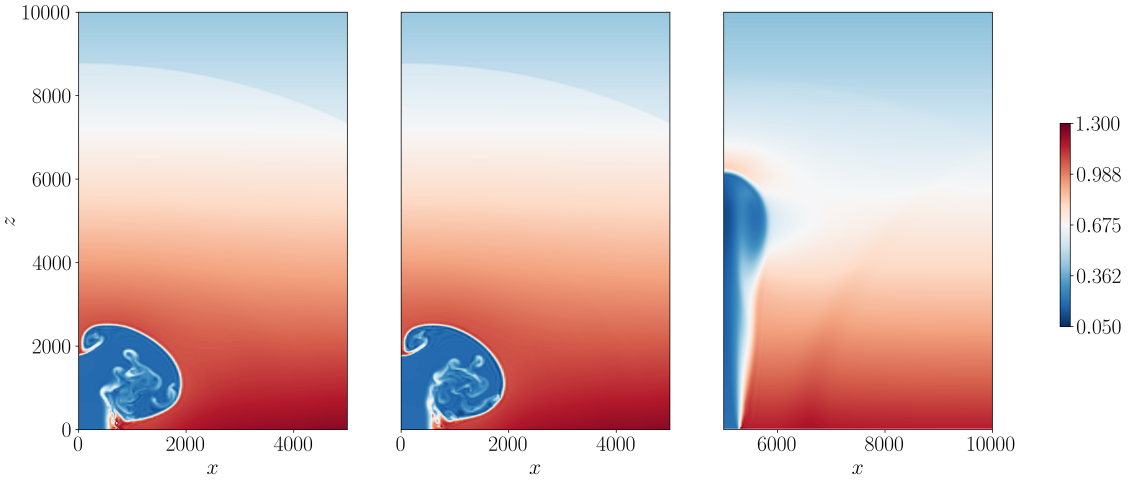

Figure 62: Comparison of the density of the jet with water vapor for both (uniform) StormFlash cases and ATHAM after 25 seconds. The plot on the left shows **Limit 1** for the StormFlash jet, the one in the middle shows **Limit 2** and the one on the right shows the results from ATHAM. The colorbar shows values in kg m^{-3}. The StormFlash jets show clear smaller scale features and reach only a maximum height of about 2.5 km while the ATHAM jet shows no small scale features and reaches a height over 6 km.

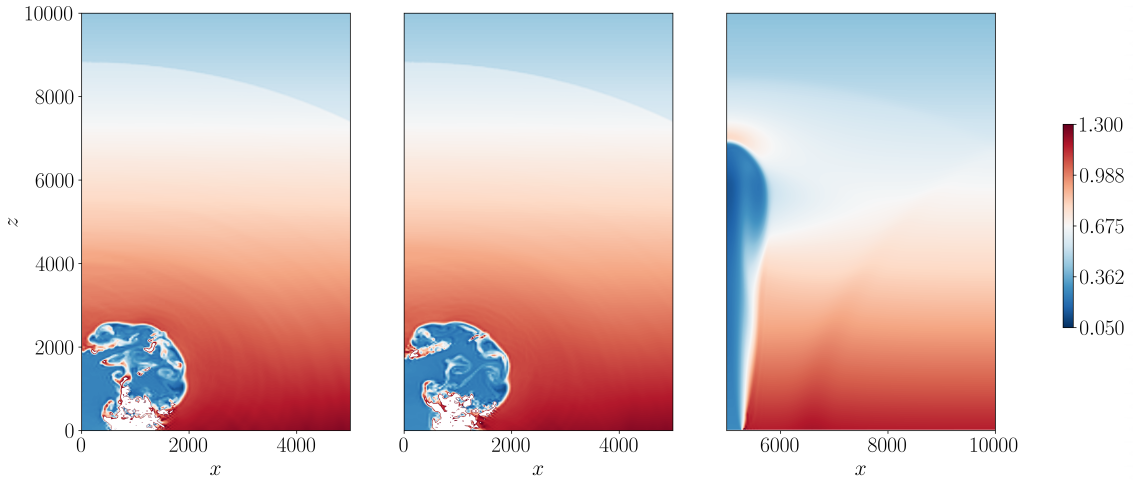

Figure 63: Comparison of the density of the volcanic plumes with 75% water vapor and 25% ash for both (uniform) StormFlash cases and ATHAM after 25 seconds. The plot on the left shows the unlimited StormFlash plume, the one in the middle shows the limited StormFlash plume and the one on the right shows the results from ATHAM. The colorbar shows values in kg m^{-3}. The StormFlash plumes show clear smaller scale features and reach only a maximum height of about 2.5 km while the ATHAM plume shows no small scale features and reaches a height of roughly 7 km. The white color within the StormFlash plots corresponds to areas where the density exceeds 1.3 kg m^{-3}.

9 Discussion, conclusion and outlook

All (StormFlash) results for volcanic jets or volcanic plumes in the previous section only show the initial phase of an eruption (50 simulated seconds), which is done as to save time and computational costs since a small domain of 10 km by 10 km can be used. Logically, using a larger domain where the volcanic jet or plume can fully develop and reach the layer of neutral buoyancy is one of the next steps. Consequently, the runs will also be longer - the simulations in Suzuki et al. (2016) are run for at least 1000 (simulated) seconds.
Since only the initial phase is shown, no correct plume height can be shown (since the neutral buoyancy is not yet reached), yet. As discussed in the previous section, the comparisons with ATHAM show, however, that the results obtained with StormFlash exhibit behaviors where the erupted material is accelerated too much which is likely due to a lack of friction.

Focusing on the plume model itself leads to discussing and answering the first research question: *Can a plume model be implemented using the Discontinuous Galerkin Method?* The results show that a plume model - in theory - can be implemented using the Discontinuous Galerkin Method. What the previous section also shows, however, is that the implementation in the current state does not produce results that would be expected (see Suzuki et al., 2016). Especially the highly primitive approach of limiting the range of values for the variables is something that needs to be exchanged for something that is more plausible.

While the simple atmospheric setup (neutrally stratified atmosphere) might play a role, the much more important fact is that the implementation appears to lack diffusion, which in turn accelerates the erupted material inside the eddy structures too much. Consequently, a very likely problem that arises, is that all the energy is transported into the smaller scales where the energy cannot be dissipated (due to a lack of diffusion). If, for example, an LES would be implemented, this energy within the smallest (Kolmogorov) scales could very likely be dissipated.

Probably due to the energy cascading into the smaller scales, the jet or plume heights are smaller - compared to the ATHAM runs. Additionally, the large velocities produce physically incorrect shocks. While shock fronts are something to be expected for volcanic eruptions, they occur close to the vent rather than several kilometers into the atmosphere (as is the case for the results obtained with StormFlash).

Consequently, the current implementation would probably benefit from a switch to Navier-Stokes equations as basis (instead of the Euler equations) or another way to add friction - and as a result introducing necessary dampening - within the flow. Also, just as **Section 7** shows, the first results for the Navier-Stokes implementation look promising, the code restructure for StormFlash lead to incorrect results, however.

Another possible error source for the incorrect physics might be the incorrect order of the numerical scheme if gravity is introduced (**Section 7.4**). In this case, exchanging the stabilization scheme with another scheme that works should also improve results.

One way this could be done is by utilizing entropy stable methods, which have also been developed for the Euler equations and other conservation laws (see e.g. Waruszewski et al., 2022; Renac, 2021; Ranocha, 2018a,b; Ranocha et al., 2021).

Of course, all these possible explanations and possible approaches to improve the plume model have nothing to do with the Discontinuous Galerkin Method itself, necessarily - it is more of a problem regarding the model or the implementation - and consequently, using the DGM as a basis is possible.

The second research question,
Are adaptive grids a viable option for volcanic modeling?,
can be answered with: yes, adaptive grids are a viable option for volcanic modeling. While the results from the previous section show that uniform and adaptive runs differ slightly (even if the adaptive grids at some point in time are almost uniform), qualitatively, they are very comparable. Additionally, the results between the different plume models differ in the specific results for each plume, as is shown in Suzuki et al. (2016). However, this is not meant as an excuse to say that the adaptive results should not match the uniform results (ideally). An obvious idea regarding the adaptive mesh refinement is to find better criteria and parameters for refinement and coarsening tolerances as to improve the grid adaptation. Additionally, adaptive grids are also viable to reduce the file sizes which leads to less storage space being used up by data. For Case H, for example, about 32% of storage space can be saved (with 30.2 GB of adaptive data versus 44.2 GB of uniform data). With better refinement criteria this can likely be improved as well.

Regarding the final research question,
How much CPU time can be saved by using adaptive mesh refinement?,
with correct setup, the use of an adaptive mesh can at least save 42% (Case H) while the results are comparable to the uniform ones (but only qualitatively and not exactly). The difference between the uniform and adaptive results becomes evident when the results are shown right next to each other. If mesh refinement is performed as with Case I, an even greater speed up can be achieved.
Furthermore, while saving almost half the CPU time for a simulation is incredibly useful, it should be noted that this is only viable for the initial phase of the eruption. As the results in the previous section show, the grids for the adaptive runs after 50 simulated seconds are almost uniform and for longer simulations, the save in CPU time will very likely be minor. Still, improving the mesh refinement with better refinement criteria will also very likely yield better results for capturing the volcanic jet or plume while reducing the amount of degrees of freedom which have to be solved.
Additionally, since the results in this thesis are obtained for just 2D, the save in CPU time will probably be even better for 3D runs.

When it comes to the future development of the plume model, the following ideas should be considered: The plume model for StormFlash is just implemented for 2D so it is evident that the extension into a full 3D model is something that needs to be done.
For further improvement and speed up, the implementation of faster time stepping schemes (such as finishing the implementation of the ROW schemes) is probably very useful since the time step size was very restricted due to the CFL condition for explicit Runge-Kutta schemes.
As mentioned previously, the runs for StormFlash use a simple, neutrally stratified atmosphere. Consequently, changing the initialization of the atmospheric data to one that is more realistic (i.e. based on atmospheric data) could be an additional source for improving the physics of the implementation within StormFlash.
When it comes to the implementation within deal.II, finding the cause for the physical implausible behavior and improving the code could lead to it being a viable option as basis for a plume model. As discussed in previous sections, such an improvement could include some form of stabilization, which might lead to better results for deal.II. Unfortunately, the adaptive mesh refinement could not be used within this thesis due to bugs in the library but once these bugs are dealt with, using deal.II would be very desirable since the library is incredibly well maintained and offers powerful high performance computing capabilities.

References

Aliabadi, Shahrouz K; Ray, Stephen Edward and Tezduyar, Tayfun E (1993). *SUPG finite element computation of viscous compressible flows based on the conservation and entropy variables formulations. Computational Mechanics*, 11(5):300–312
URL https://doi.org/10.1007/BF00350089

Aliabadi, S.K. and Tezduyar, Tayfun (1993). *Space-time finite element computation of compressible flows involving moving boundaries and interfaces. Computer Methods in Applied Mechanics and Engineering*, 107:209–223
URL https://doi.org/10.1016/0045-7825(93)90176-X

Arndt, Daniel; Bangerth, Wolfgang; Blais, Bruno; Clevenger, Thomas C.; Fehling, Marc; Grayver, Alexander V.; Heister, Timo; Heltai, Luca; Kronbichler, Martin; Maier, Matthias; Munch, Peter; Pelteret, Jean-Paul; Rastak, Reza; Tomas, Ignacio; Turcksin, Bruno; Wang, Zhuoran and Wells, David (2020). *The deal.II library, Version 9.2. Journal of Numerical Mathematics*, 28(3):131–146
URL https://doi.org/10.1515/jnma-2020-0043

Babuška, Ivo; Szabo, B.A. and Katz, I.N. (1981). *The p-version of the finite element method. SIAM J. Numer. Anal.*, 18:515–545
URL https://doi.org/10.1137/0718033

Bassi, F. and Rebay, S. (1997). *A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier–Stokes Equations. Journal of Computational Physics*, 131(2):267–279
URL https://www.sciencedirect.com/science/article/pii/S0021999196955722

BBC (2010). *Ash cloud chaos: Airlines face huge task as ban ends, retrieved 20.5.2017*
URL http://news.bbc.co.uk/2/hi/uk_news/8633892.stm

BBC (2010). *Flight disruptions cost airlines $1.7bn, says IATA, retrieved 20.5.2017*
URL http://news.bbc.co.uk/2/hi/business/8634147.stm

Behrens, Jörn; Rakowsky, Natalja; Hiller, Wolfgang; Handorf, Dörthe; Läuter, Matthias; Päpke, Jürgen and Dethloff, Klaus (2005). *amatos: Parallel adaptive mesh generator for atmospheric and oceanic simulation. Ocean Modelling*, 10(1–2):171 – 183. The Second International Workshop on Unstructured Mesh Numerical Modelling of Coastal, Shelf and Ocean Flows
URL http://www.sciencedirect.com/science/article/pii/S1463500304000599

Beisiegel, Nicole; Vater, Stefan; Behrens, Jörn and Dias, Frédéric (2020). *An adaptive discontinuous Galerkin method for the simulation of hurricane storm surge. Ocean Dynamics*, 70
URL https://doi.org/10.1007/s10236-020-01352-w

Benacchio, Tommaso (2014). *A blended semi-implicit numerical model for weakly compressible atmospheric dynamics.* PHD thesis, Freie Universität Berlin
URL http://www.diss.fu-berlin.de/diss/receive/FUDISS_thesis_000000097149

Berger, Marsha J and Oliger, Joseph (1984). *Adaptive mesh refinement for hyperbolic partial differential equations. Journal of Computational Physics*, 53(3):484 – 512
URL http://www.sciencedirect.com/science/article/pii/0021999184900731

Birken, Philipp (2013). *Numerical Methods for the Unsteady Compressible Navier-Stokes Equations*. PHD thesis, Kassel, Universität, FB 10, Mathematik und Naturwissenschaften, Institut für Mathematik
URL http://www.urn.fi/urn:nbn:de:hebis:34-2013013042485

Blaise, Sébastien; Lambrechts, Jonathan and Deleersnijder, Eric (2016). *A stabilization for three-dimensional discontinuous Galerkin discretizations applied to nonhydrostatic atmospheric simulations. International Journal for Numerical Methods in Fluids*, 81(9):558–585
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4197

Bond, Ryan; Ober, Curtis; Knupp, Patrick and Bova, Steve (2007). *Manufactured Solution for Computational Fluid Dynamics Boundary Condition Verification. Aiaa Journal - AIAA J*, 45:2224–2236
URL https://doi.org/10.2514/1.28099

Botta, N.; Klein, R.; Langenberg, S. and Lützenkirchen, S. (2004). *Well balanced finite volume methods for nearly hydrostatic flows. Journal of Computational Physics*, 196(2):539 – 565
URL http://www.sciencedirect.com/science/article/pii/S0021999103006077

Boyd, JP (1996). *The erfc-log filter and the asymptotics of the Euler and Vandeven sequence accelerations*. In *Proceedings of the Third International Conference on Spectral and High Order Methods*, pp. 267–276. Citeseer

Bursik, M. (2001). *Effect of wind on the rise height of volcanic plumes. Geophysical Research Letters*, 28(18):3621–3624
URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2001GL013393

Bänsch, Michel (2017). *A simplified model for volcanic plumes and its solution by an adaptive numerical scheme*

Bänsch, Michel; Behrens, Jörn and Vater, Stefan (2022). *A linear low effort stabilization method for the Euler equations using Discontinuous Galerkin Methods. submitted*

Cai, X.; Yang, Z.-L.; Fisher, J. B.; Zhang, X.; Barlage, M. and Chen, F. (2016). *Integration of nitrogen dynamics into the Noah-MP land surface model v1.1 for climate and environmental predictions. Geoscientific Model Development*, 9(1):1–15
URL https://gmd.copernicus.org/articles/9/1/2016/

Calle, Jorge; Devloo, Philippe and Gomes, Sônia (2005a). *Wavelets and adaptive grids for the discontinuous Galerkin method. Numerical Algorithms*, 39:143–154
URL https://doi.org/10.1007/s00466-012-0712-z

Calle, Jorge LD; Devloo, Philippe RB and Gomes, Sônia M (2005b). *Stabilized discontinuous Galerkin method for hyperbolic equations. Computer Methods in Applied Mechanics and Engineering*, 194(17):1861–1874
URL https://doi.org/10.1016/j.cma.2004.06.036

Cao, Zhixuan; Bursik, Marcus; Yang, Qingyuan and Patra, Abani (2021). *Simulating the Transport and Dispersal of Volcanic Ash Clouds With Initial Conditions Created by a 3D Plume Model. Frontiers in earth science (Lausanne)*, 9
URL https://doi.org/10.3389/feart.2021.704797

Cerminara, M.; Esposti Ongaro, T. and Berselli, L. C. (2016a). *ASHEE-1.0: a compressible, equilibrium–Eulerian model for volcanic ash plumes. Geoscientific Model Development*, 9(2):697–730
URL http://www.geosci-model-dev.net/9/697/2016/

Cerminara, Matteo (2015). *The Multiphase Buoyant Plume Solution of the Dusty Gas Model. arXiv: Geophysics*
URL https://doi.org/10.48550/arXiv.1506.01638

Cerminara, Matteo; Ongaro, Tomaso Esposti and Neri, Augusto (2016b). *Large Eddy Simulation of gas–particle kinematic decoupling and turbulent entrainment in volcanic plumes. Journal of Volcanology and Geothermal Research*, 326:143 – 171. Numerical models of volcanic eruption plumes: inter-comparison and sensitivity
URL http://www.sciencedirect.com/science/article/pii/S0377027316301688

Chandrashekar, Praveen and Zenk, Markus (2017). *Well-Balanced Nodal Discontinuous Galerkin Method for Euler Equations with Gravity. Journal of Scientific Computing*, 71(3):1062–1093
URL https://doi.org/10.1007/s10915-016-0339-x

Chorin, Alexandre J. and Marsden, Jerrold E. (1993). *A Mathematical Introduction to Fluid Mechanics*. Springer-Verlag, New York. ISBN 978-0-387-97918-2
URL https://doi.org/10.1007/978-1-4684-0082-3

Cockburn, Bernardo and Shu, Chi-Wang (1998). *The Runge–Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems. Journal of Computational Physics*, 141(2):199 – 224
URL http://www.sciencedirect.com/science/article/pii/S0021999198958922

Cockburn, Bernardo and Shu, Chi-Wang (2001a). *Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. Journal of Scientific Computing*, 16(3):173–261
URL https://doi.org/10.1023/A:1012873910884

Cockburn, Bernardo and Shu, Chi-Wang (2001b). *Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. Journal of Scientific Computing*, 16(3):173–261
URL https://doi.org/10.1023/A:1012873910884

Costa, A.; Suzuki, Y.J.; Cerminara, M.; Devenish, B.J.; Ongaro, T. Esposti; Herzog, M.; Van Eaton, A.R.; Denby, L.C.; Bursik, M.; de' Michieli Vitturi, M.; Engwell, S.; Neri, A.; Barsotti, S.; Folch, A.; Macedonio, G.; Girault, F.; Carazzo, G.; Tait, S.; Kaminski, E.; Mastin, L.G.; Woodhouse, M.J.; Phillips, J.C.; Hogg, A.J.; Degruyter, W. and Bonadonna, C. (2016). *Results of the eruptive column model inter-comparison study. Journal of Volcanology and Geothermal Research*, 326:2–25. Numerical models of volcanic eruption plumes: inter-comparison and sensitivity
URL https://www.sciencedirect.com/science/article/pii/S0377027316000366

Courant, R.; Friedrichs, K. and Lewy, H. (1928). *Über die partiellen Differenzengleichungen der mathematischen Physik. Mathematische Annalen*, 100:32–74
URL https://ui.adsabs.harvard.edu/abs/1928MatAn.100...32C

Crank, J. and Nicolson, P. (1996). *A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. Advances in Computational Mathematics*, 6(1):207–226
URL https://doi.org/10.1007/BF02127704

Degruyter, W. and Bonadonna, Costanza (2012). *Improving on mass flow rate estimates of volcanic eruptions. Geophysical Research Letters*, 39:16308–
URL https://doi.org/10.1029/2012GL052566

Devenish, B.J. (2013). *Using simple plume models to refine the source mass flux of volcanic eruptions according to atmospheric conditions. Journal of Volcanology and Geothermal Research*, 256:118–127
URL https://www.sciencedirect.com/science/article/pii/S0377027313000668

Dolejsi, Vit and Feistauer, Miloslav (2015). *Discontinuous Galerkin Method*, vol. 48. ISBN 978-3-319-19266-6

Ern, A. and Guermond, J.L. (2004). *Theory and Practice of Finite Elements*. Applied Mathematical Sciences. Springer New York. ISBN 9780387205748
URL https://books.google.de/books?id=CCjm79FbJbcC

Fehn, Niklas; Wall, Wolfgang A. and Kronbichler, Martin (2019). *A matrix-free high-order discontinuous Galerkin compressible Navier-Stokes solver: A performance comparison of compressible and incompressible formulations for turbulent incompressible flows. International Journal for Numerical Methods in Fluids*, 89(3):71–102
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4683

Freire, Sergio; Florczyk, Aneta J.; Pesaresi, Martino and Sliuzas, Richard (2019a). *An Improved Global Analysis of Population Distribution in Proximity to Active Volcanoes, 1975–2015. ISPRS International Journal of Geo-Information*, 8(8)
URL https://www.mdpi.com/2220-9964/8/8/341

Freire, Sergio; Florczyk, Aneta J.; Pesaresi, Martino and Sliuzas, Richard (2019b). *An Improved Global Analysis of Population Distribution in Proximity to Active Volcanoes, 1975–2015. ISPRS International Journal of Geo-Information*, 8(8)
URL https://www.mdpi.com/2220-9964/8/8/341

Fryxell, B.; Olson, K.; Ricker, P.; Timmes, F. X.; Zingale, M.; Lamb, D. Q.; MacNeice, P.; Rosner, R.; Truran, J. W. and Tufo, H. (2000). *FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes. The Astrophysical Journal, Supplement*, 131:273–334
URL https://dx.doi.org/10.1086/317361

Giraldo, F. X. and Warburton, T. (2008). *A high-order triangular discontinuous Galerkin oceanic shallow water model. International Journal for Numerical Methods in Fluids*, 56(7):899–925
URL http://dx.doi.org/10.1002/fld.1562

Giraldo, Francis (2020). *An Introduction to Element-Based Galerkin Methods on Tensor-Product Bases: Analysis, Algorithms, and Applications*. ISBN 978-3-030-55068-4

Giraldo, Francis X. and Rosmond, Thomas E. (2004). *A Scalable Spectral Element Eulerian Atmospheric Model (SEE-AM) for NWP: Dynamical Core Tests*. *Monthly Weather Review*, 132(1):133 – 153
URL https://journals.ametsoc.org/view/journals/mwre/132/1/1520-0493_2004_132_0133_asseea_2.0.co_2.xml

Giraldo, F.X.; Hesthaven, J. S. and Warburton, T. (2002). *Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations*. The article of record as published may be located at http://dx.doi.org/10.1006/jcph.2002.7139
URL http://hdl.handle.net/10945/25524

Giraldo, F.X. and Restelli, M. (2008). *A study of spectral element and discontinuous Galerkin methods for the Navier–Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases*. *Journal of Computational Physics*, 227(8):3849 – 3877
URL http://www.sciencedirect.com/science/article/pii/S0021999107005384

Girault, Frédéric; Carazzo, Guillaume; Tait, Steve; Ferrucci, Fabrizio and Édouard Kaminski (2014). *The effect of total grain-size distribution on the dynamics of turbulent volcanic plumes*. *Earth and Planetary Science Letters*, 394:124–134
URL https://www.sciencedirect.com/science/article/pii/S0012821X14001654

Guardian, The (2010). *Ash cloud costing airlines £130m a day, retrieved 20.5.2017*
URL https://www.theguardian.com/business/2010/apr/16/iceland-volcano-airline-industry-iata

Hairer, E.; Nørsett, S.P. and Wanner, G. (2000). *Solving Ordinary Differential Equations I Nonstiff problems*. Springer, Berlin, second ed.
URL https://www.bibsonomy.org/bibtex/26bfd1a0356243229b8d30cb296e19f48/brouder

Hairer, Ernst and Wanner, Gerhard (1996). *Solving ordinary differential equations. II, Vol. 14 of. Springer Series in Computational Mathematics (Springer Berlin Heidelberg, Berlin, Heidelberg, 1996)*, 10:978–3

Herzog, Michael; Oberhuber, Josef M. and Graf, Hans-F. (2003). *A Prognostic Turbulence Scheme for the Nonhydrostatic Plume Model ATHAM*. *Journal of the Atmospheric Sciences*, 60(22):2783–2796
URL http://dx.doi.org/10.1175/1520-0469(2003)060<2783:APTSFT>2.0.CO;2

Hesthaven, Jan S. and Warburton, Tim (2007). *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Publishing Company, Incorporated, 1st ed. ISBN 0387720650
URL https://doi.org/10.1007/978-0-387-72067-8

Hilbert, David (1891). *Ueber die stetige Abbildung einer Line auf ein Flächenstück*. *Mathematische Annalen*, 38(3):459–460
URL https://doi.org/10.1007/BF01199431

Hirsch, Charles (1988). *Numerical Computation of Internal & External Flows: Fundamentals of Numerical Discretization*. John Wiley & Sons, Inc., USA. ISBN 0471917621
URL https://dl.acm.org/doi/abs/10.5555/63653

Issa, R.I (1986). *Solution of the implicitly discretised fluid flow equations by operator-splitting*. *Journal of Computational Physics*, 62(1):40 – 65
URL http://www.sciencedirect.com/science/article/pii/0021999186900999

Jax, Tim; Bartel, Andreas; Ehrhardt, Matthias; Günther, Michael and Steinebach, Gerd (Editors) (2021). *Rosenbrock–Wanner–Type Methods*. Mathematics Online First Collections. Springer, Cham. ISBN 978-3-030-76809-6
URL https://doi.org/10.1007/978-3-030-76810-2

John, Volker and Rang, Joachim (2010). *Adaptive time step control for the incompressible Navier–Stokes equations*. *Computer Methods in Applied Mechanics and Engineering*, 199(9):514–524
URL https://www.sciencedirect.com/science/article/pii/S0045782509003417

Kennedy, Christopher A.; Carpenter, Mark H. and Lewis, R.Michael (2000). *Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations*. *Applied Numerical Mathematics*, 35(3):177–219
URL https://www.sciencedirect.com/science/article/pii/S0168927499001415

Kler, Pablo; Dalcin, Lisandro; Paz, Rodrigo and Tezduyar, Tayfun (2013). *SUPG and discontinuity-capturing methods for coupled fluid mechanics and electrochemical transport problems*. *Computational Mechanics*, 51:171–185
URL https://doi.org/10.1007/s00466-012-0712-z

Kühnlein, Christian; Smolarkiewicz, Piotr K. and Dörnbrack, Andreas (2012). *Modelling atmospheric flows with adaptive moving meshes*. *Journal of Computational Physics*, 231(7):2741 – 2763
URL http://www.sciencedirect.com/science/article/pii/S0021999111007224

Lang, Jens; Cao, Weiming; Huang, Weizhang and Russell, Robert D. (2003). *A two-dimensional moving finite element method with local refinement based on a posteriori error estimates*. *Applied Numerical Mathematics*, 46(1):75–94
URL https://www.sciencedirect.com/science/article/pii/S0168927403000138

Leer, Bram Van (1977). *Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow*. *Journal of Computational Physics*, 23(3):263 – 275
URL http://www.sciencedirect.com/science/article/pii/0021999177900948

LeVeque, Randall J. (1992). *Numerical Methods for Conservation Laws*. Birkhäuser Basel, Basel. ISBN 978-3-7643-2723-1 978-3-0348-8629-1
URL http://link.springer.com/10.1007/978-3-0348-8629-1

Li, Gang and Xing, Yulong (2016). *Well-Balanced Discontinuous Galerkin Methods for the Euler Equations Under Gravitational Fields*. *Journal of Scientific Computing*, 67(2):493–513
URL https://doi.org/10.1007/s10915-015-0093-5

Li, Shengtai (2010). *Comparison of refinement criteria for structured adaptive mesh refinement. Journal of Computational and Applied Mathematics*, 233(12):3139 – 3147. Finite Element Methods in Engineering and Science (FEMTEC 2009)
URL http://www.sciencedirect.com/science/article/pii/S037704270900586X

Marble, F E (1970). *Dynamics of Dusty Gases. Annual Review of Fluid Mechanics*, 2(1):397–446
URL http://dx.doi.org/10.1146/annurev.fl.02.010170.002145

Mastin, Larry G. (2007). *A user-friendly one-dimensional model for wet volcanic plumes. Geochemistry, Geophysics, Geosystems*, 8(3)
URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2006GC001455

de' Michieli Vitturi, M.; Neri, A. and Barsotti, S. (2015). *PLUME-MoM 1.0: A new integral model of volcanic plumes based on the method of moments. Geoscientific Model Development*, 8(8):2447–2463
URL https://gmd.copernicus.org/articles/8/2447/2015/

Morton, B. R.; Taylor, Geoffrey Ingram and Turner, John Stewart (1956). *Turbulent gravitational convection from maintained and instantaneous sources. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 234(1196):1–23
URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1956.0011

Müller, A.; Behrens, J.; Giraldo, F. X. and Wirth, V. (2013). *Comparison between adaptive and uniform discontinuous Galerkin simulations in dry 2D bubble experiments. Journal of Computational Physics*, 235:371 – 393
URL http://www.sciencedirect.com/science/article/pii/S0021999112006419

Müller, Andreas (2012). *Untersuchungen zur Genauigkeit adaptiver unstetiger Galerkin-Simulationen mit Hilfe von Luftblasen-Testfällen.* PHD thesis, Johannes Gutenberg-Universität
URL http://doi.org/10.25358/openscience-2068

Neri, Augusto; Esposti Ongaro, Tomaso; Macedonio, Giovanni and Gidaspow, Dimitri (2003). *Multiparticle simulation of collapsing volcanic columns and pyroclastic flow. Journal of Geophysical Research: Solid Earth*, 108(B4). 2202
URL http://dx.doi.org/10.1029/2001JB000508

Ongaro, T. Esposti; Cavazzoni, C.; Erbacci, G.; Neri, A. and Salvetti, M.V. (2007). *A parallel multiphase flow code for the 3D simulation of explosive volcanic eruptions. Parallel Computing*, 33(7–8):541 – 560
URL http://www.sciencedirect.com/science/article/pii/S0167819107000634

Oppenheimer, Clive (2011). *Eruptions that Shook the World.* Cambridge University Press
URL https://doi.org/10.1017/CBO9780511978012

Parfitt, E.A. and Wilson, Lionel (2008). *Fundamentals of physical volcanology. Bull. Volcanol.*, 72
URL https://eprints.lancs.ac.uk/id/eprint/31095

Peano, G. (1890). *Sur une courbe, qui remplit toute une aire plane. Mathematische Annalen*, 36(1):157–160
URL https://doi.org/10.1007/BF01199438

Piggott, M.D.; Pain, C.C.; Gorman, G.J.; Power, P.W. and Goddard, A.J.H. (2005). *h, r, and hr adaptivity with applications in numerical ocean modelling. Ocean Modelling*, 10(1):95 – 113. The Second International Workshop on Unstructured Mesh Numerical Modelling of Coastal, Shelf and Ocean Flows
URL http://www.sciencedirect.com/science/article/pii/S1463500304000587

Proud, Simon R.; Prata, Andrew T. and Schmauß, Simeon (2022). *The January 2022 eruption of Hunga Tonga-Hunga Ha’apai volcano reached the mesosphere. Science*, 378(6619):554–557
URL https://www.science.org/doi/abs/10.1126/science.abo4076

Ranocha, Hendrik (2018a). *Comparison of Some Entropy Conservative Numerical Fluxes for the Euler Equations. Journal of Scientific Computing*, 76(1):216–242
URL https://doi.org/10.1007/s10915-017-0618-1

Ranocha, Hendrik (2018b). *Generalised Summation-by-Parts Operators and Entropy Stability of Numerical Methods for Hyperbolic Balance Laws*. PHD thesis, TU Braunschweig
URL https://cuvillier.de/de/shop/publications/7743

Ranocha, Hendrik; Schlottke-Lakemper, Michael; Chan, Jesse; Rueda-Ramírez, Andrés M; Winters, Andrew R; Hindenlang, Florian and Gassner, Gregor J (2021). *Efficient implementation of modern entropy stable and kinetic energy preserving discontinuous Galerkin methods for conservation laws*

Renac, Florent (2021). *Entropy stable, robust and high-order DGSEM for the compressible multicomponent Euler equations. Journal of Computational Physics*, 445:110584
URL https://www.sciencedirect.com/science/article/pii/S0021999121004794

Roache, Patrick J. (2001). *Code Verification by the Method of Manufactured Solutions . Journal of Fluids Engineering*, 124(1):4–10
URL https://doi.org/10.1115/1.1436090

Robert, André (1993). *Bubble Convection Experiments with a Semi-implicit Formulation of the Euler Equations. Journal of the Atmospheric Sciences*, 50(13):1865–1873
URL https://doi.org/10.1175/1520-0469(1993)050<1865:BCEWAS>2.0.CO;2

Roe, P.L (1981). *Approximate Riemann solvers, parameter vectors, and difference schemes. Journal of Computational Physics*, 43(2):357 – 372
URL http://www.sciencedirect.com/science/article/pii/0021999181901285

Roth, A. (2012). *Vacuum Technology*. Elsevier Science. ISBN 9780444598745
URL https://books.google.de/books?id=oBqs3sr9r48C

Roy, C. J.; Nelson, C. C.; Smith, T. M. and Ober, C. C. (2004). *Verification of Euler/Navier-Stokes codes using the method of manufactured solutions. International Journal for Numerical Methods in Fluids*, 44(6):599–620
URL https://ui.adsabs.harvard.edu/abs/2004IJNMF..44..599R

Saad, Youcef and Schultz, Martin H. (1986). *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869
URL https://doi.org/10.1137/0907058

Schoeder, S.; Kormann, K.; Wall, W. A. and Kronbichler, M. (2018). *Efficient Explicit Time Stepping of High Order Discontinuous Galerkin Schemes for Waves. SIAM Journal on Scientific Computing*, 40(6):C803–C826
URL https://doi.org/10.1137/18M1185399

Shu, Chi-Wang (1988). *Total-Variation-Diminishing Time Discretizations. SIAM Journal on Scientific and Statistical Computing*, 9(6):1073–1084
URL https://doi.org/10.1137/0909073

Shu, Chi-Wang and Osher, Stanley (1988). *Efficient implementation of essentially non-oscillatory shock-capturing schemes. Journal of Computational Physics*, 77(2):439–471
URL https://www.sciencedirect.com/science/article/pii/0021999188901775

Smagorinsky, J. (1963). *General circulation experiments with the primitive equations. Monthly Weather Review*, 91(3):99–164
URL https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2

Smithsonian Institution (2022). *Global Volcanism Program retrieved 16.11.2022*
URL https://volcano.si.edu/volcano.cfm?vn=372020&vtab=Eruptions

Smolarkiewicz, Piotr K (1984). *A fully multidimensional positive definite advection transport algorithm with small implicit diffusion. Journal of Computational Physics*, 54(2):325 – 362
URL http://www.sciencedirect.com/science/article/pii/0021999184901219

Suzuki, Y. J. and Koyaguchi, T. (2009). *A three-dimensional numerical simulation of spreading umbrella clouds. Journal of Geophysical Research: Solid Earth*, 114(B3):n/a–n/a. B03209
URL http://dx.doi.org/10.1029/2007JB005369

Suzuki, Y.J.; Costa, A.; Cerminara, M.; Ongaro, T. Esposti; Herzog, M.; Eaton, A.R. Van and Denby, L.C. (2016). *Inter-comparison of three-dimensional models of volcanic plumes. Journal of Volcanology and Geothermal Research*, 326:26 – 42. Numerical models of volcanic eruption plumes: inter-comparison and sensitivity
URL http://www.sciencedirect.com/science/article/pii/S0377027316301391

Suzuki, Yujiro J. and Koyaguchi, Takehiro (2013). *3D numerical simulation of volcanic eruption clouds during the 2011 Shinmoe-dake eruptions. Earth, Planets and Space*, 65(6):10
URL http://dx.doi.org/10.5047/eps.2013.03.009

Suzuki, Yujiro J. and Koyaguchi, Takehiro (2015). *Effects of wind on entrainment efficiency in volcanic plumes. Journal of Geophysical Research: Solid Earth*, 120(9):6122–6140. 2015JB012208
URL http://dx.doi.org/10.1002/2015JB012208

Suzuki, Yujiro J.; Koyaguchi, Takehiro; Ogawa, Masaki and Hachisu, Izumi (2005). *A numerical study of turbulent mixing in eruption clouds using a three-dimensional fluid dynamics model.*

Journal of Geophysical Research: Solid Earth, 110(B8). B08201
URL http://dx.doi.org/10.1029/2004JB003460

Tan, Hua (2018). *Applying the free-slip boundary condition with an adaptive Cartesian cut-cell method for complex geometries. Numerical Heat Transfer, Part B: Fundamentals*, 74(4):661–684
URL https://doi.org/10.1080/10407790.2018.1562770

Tezduyar, Tayfun (1992). *Stabilized Finite Element Formulations for Incompressible Flow Computations. Advances in Applied Mechanics*, 28:1–44
URL https://doi.org/10.1016/S0065-2156(08)70153-4

The Telegraph (2011). *How the 2010 ash cloud caused chaos: facts and figures, retrieved 20.5.2017*
URL http://www.telegraph.co.uk/finance/newsbysector/transport/8531152/How-the-2010-ash-cloud-caused-chaos-facts-and-figures.html

Vater, Stefan (2013). *A multigrid-based multiscale numerical scheme for shallow water flows at low froude number*. PHD thesis, Freie Universität Berlin
URL http://www.diss.fu-berlin.de/diss/receive/FUDISS_thesis_000000093897

Walker, George P. L. (1973). *Explosive volcanic eruptions — a new classification scheme. Geologische Rundschau*, 62(2):431–446
URL https://doi.org/10.1007/BF01840108

Waruszewski, Maciej; Kozdon, Jeremy E.; Wilcox, Lucas C.; Gibson, Thomas H. and Giraldo, Francis X. (2022). *Entropy stable discontinuous Galerkin methods for balance laws in non-conservative form: Applications to the Euler equations with gravity. Journal of Computational Physics*, 468:111507
URL https://www.sciencedirect.com/science/article/pii/S0021999122005691

Woodhouse, M. J.; Hogg, A. J.; Phillips, J. C. and Sparks, R. S. J. (2013a). *Interaction between volcanic plumes and wind during the 2010 Eyjafjallajökull eruption, Iceland. Journal of Geophysical Research: Solid Earth*, 118(1):92–109
URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2012JB009592

Woodhouse, M. J.; Hogg, A. J.; Phillips, J. C. and Sparks, R. S. J. (2013b). *Interaction between volcanic plumes and wind during the 2010 Eyjafjallajökull eruption, Iceland. Journal of Geophysical Research: Solid Earth*, 118(1):92–109
URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2012JB009592

worldometers (2022). *World population retrieved 15.11.2022*
URL http://www.worldometers.info/world-population/

Zheng, Y. and Zhang, T. (1998). *Axisymmetric Solutions of the Euler Equations for Sub-Square Polytropic Gases. ArXiv Mathematics e-prints*
URL http://adsabs.harvard.edu/abs/1998math......7167Z

Zidikheri, Meelis J.; Lucas, Christopher and Potts, Rodney J. (2017a). *Estimation of optimal dispersion model source parameters using satellite detections of volcanic ash. Journal of Geophysical Research: Atmospheres*, 122(15):8207–8232. 2017JD026676
URL http://dx.doi.org/10.1002/2017JD026676

Zidikheri, Meelis J.; Lucas, Christopher and Potts, Rodney J. (2017b). *Toward quantitative forecasts of volcanic ash dispersal: Using satellite retrievals for optimal estimation of source terms. Journal of Geophysical Research: Atmospheres*, 122(15):8187–8206. 2017JD026679
URL http://dx.doi.org/10.1002/2017JD026679

List of Figures

List of Tables

A Appendices

A List of variables and parameters

Table 15: List of variables and parameters

Variable	Value and units	Meaning	Reference
ρ	kg m^{-3}	density of the gas or the ash-gas mixture	
$\mathbf{u} = (u, w)^\tau$	m s^{-1}	velocity in x- and z-direction respectively	
ρe	J m^{-3}	total energy density excluding potential energy	**Equation 2**
P	Pa = N m^{-2}	pressure	**Equation 1**, **Equation 3**, **Equation 59**
R or R_a	287 J kg^{-1} K^{-1}	specific gas constant for dry air	
R_g	461 J kg^{-1} K^{-1}	specific gas constant for water vapor	
c_p	1004.67 kg m^2 K^{-1} s^{-2}	specific heat capacity of dry air at constant pressure	
c_v or $c_{v,a}$	717.5 kg m^2 K^{-1} s^{-2}	specific heat capacity of dry air at constant volume	
$c_{v,g}$	1340 kg m^2 K^{-1} s^{-2}	specific heat capacity of water vapor at constant volume	
$c_{v,s}$	1100 kg m^2 K^{-1} s^{-2}	specific heat capacity of volcanic ash at constant volume	
γ	1.4	specific heat ratio for air	
γ_m		specific heat ratio for ash-gas mixture	

B AMATOS

The AMATOS library is a Fortran 95 library that was developed by Jörn Behrens (Behrens et al., 2005) that allows for the creation and management of adaptive meshes. The library was used for many cases in this thesis. For the simulations, initial triangulation files with information about nodes, elements, edges and boundary conditions have to be provided. Usually, these files start with a small amount of (triangular) elements for which the grid can be refined - adaptively if needed.
For the code basis that was used, right isosceles triangles are utilized that can be refined by section (Behrens et al., 2005).
During a simulation - if adaptive mesh refinement is desired - the grid is adapted if enough elements have been flagged for either coarsening or refinement. With this procedure, the user has the advantage that they do not have to worry about grid creation or management since this is taken care of by the library. The unknowns that are solved for during the simulation are stored in arrays whose sizes can vary throughout the run. To ensure quick memory access, the data is always sorted by means of space filling curves so that values for neighboring cells are also stored close to each other in memory which allows for faster access.
AMATOS takes care of grid management in two steps: the data is provided by a gather functionality (where the algorithm receives a data array from AMATOS) in step one where the data is scattered to the grid again (after numerical calculations such as solving the respective PDEs) so that AMATOS can take care of grid adaption and manipulation as the second step, if necessary (see 64). This procedure is repeated for each time step.

As mentioned in **Section 3.3.2**, to ensure correct mesh refinement and coarsening, the algorithm needs an error indicator/estimator or refinement criterion (Behrens et al., 2005).

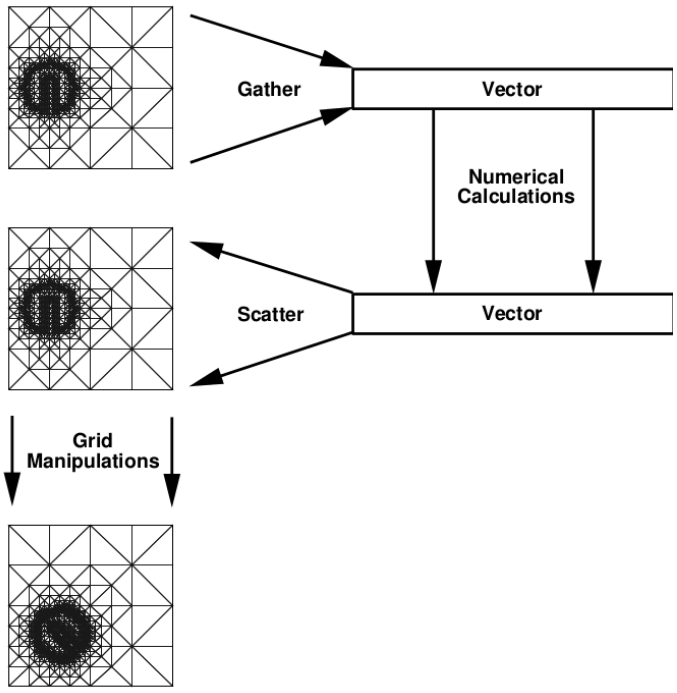

Figure 64: Schematic calculation procedure of AMATOS (taken from Behrens et al., 2005)

C deal.II

As stated on its homepage (https://www.dealii.org/about.html), deal.II is the successor to the Differential Equations Analysis Library. It is a C++ library that started off from work done by the Numerical Methods Group at Universität Heidelberg, Germany. As of today, there are many international contributors.

The goal of the library is to provide opportunities to create modern finite element codes that can easily be supported by an array of different tools such as adaptive mesh refinement, routines for automatic differentiation and preconditioners to name just a few.

Similar to the AMATOS library, the deal.II library can be used to take care of grid management. Different (adaptive) finite element approaches that solve partial differential equations can be implemented with this library and there already exist a plethora of examples or codes for many applications.

D Plots for the jet runs with deal.II

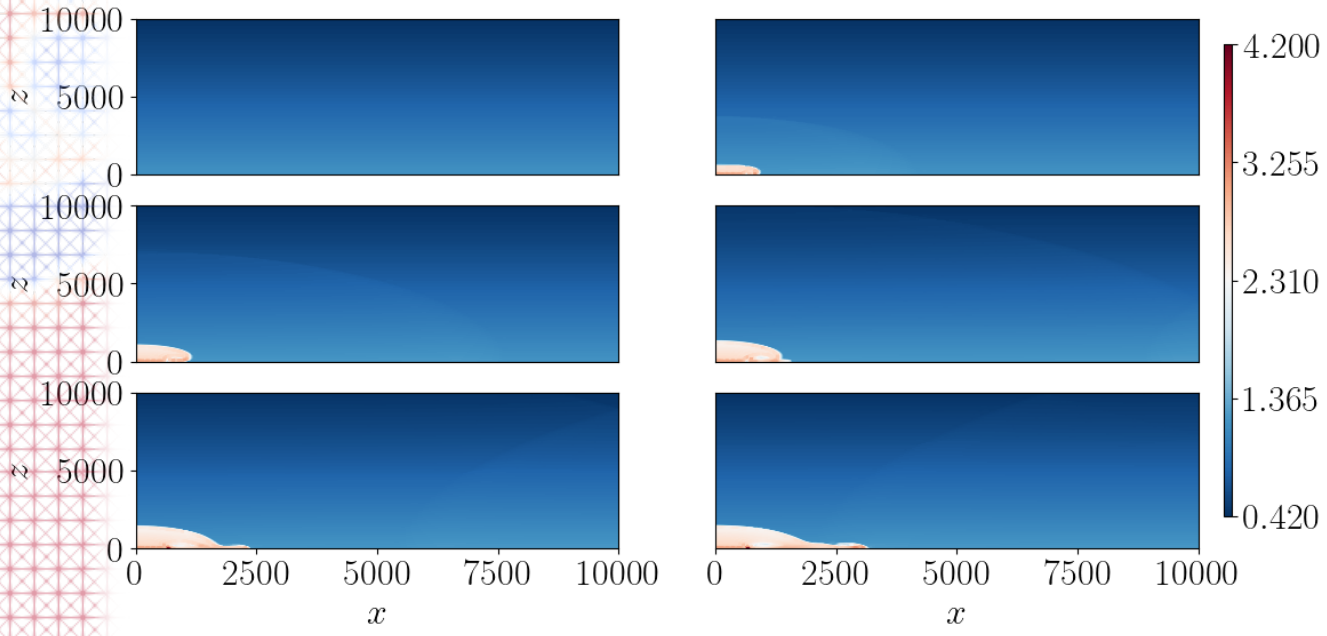

Figure 65: Density for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere for deal.II. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. Within the plots, an acoustic wave travelling outwards (from the vent) is barely visible. The jet reaches a maximum height of around 2 km and then collapses.

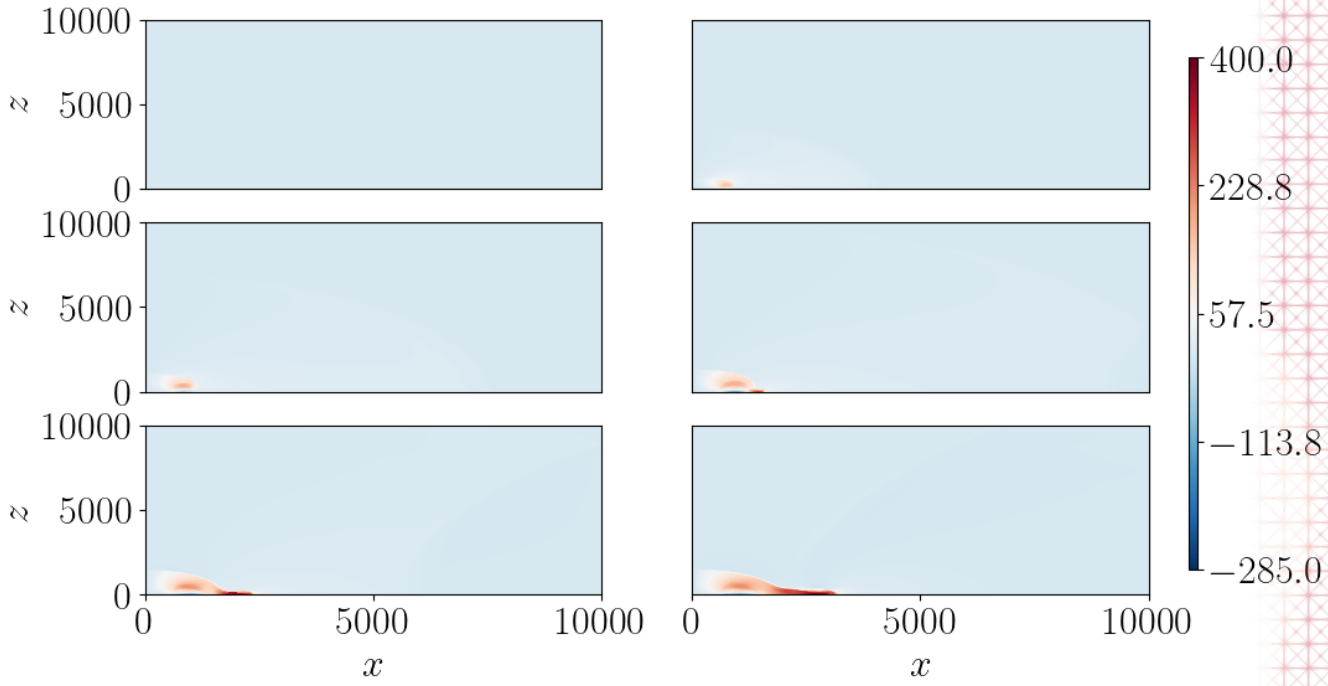

Figure 66: Horizontal momentum for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere for deal.II. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. Within the plots, an acoustic wave travelling outwards (from the vent) is barely visible. As with the density plot, a collapse of the jet is visible with increasing time.

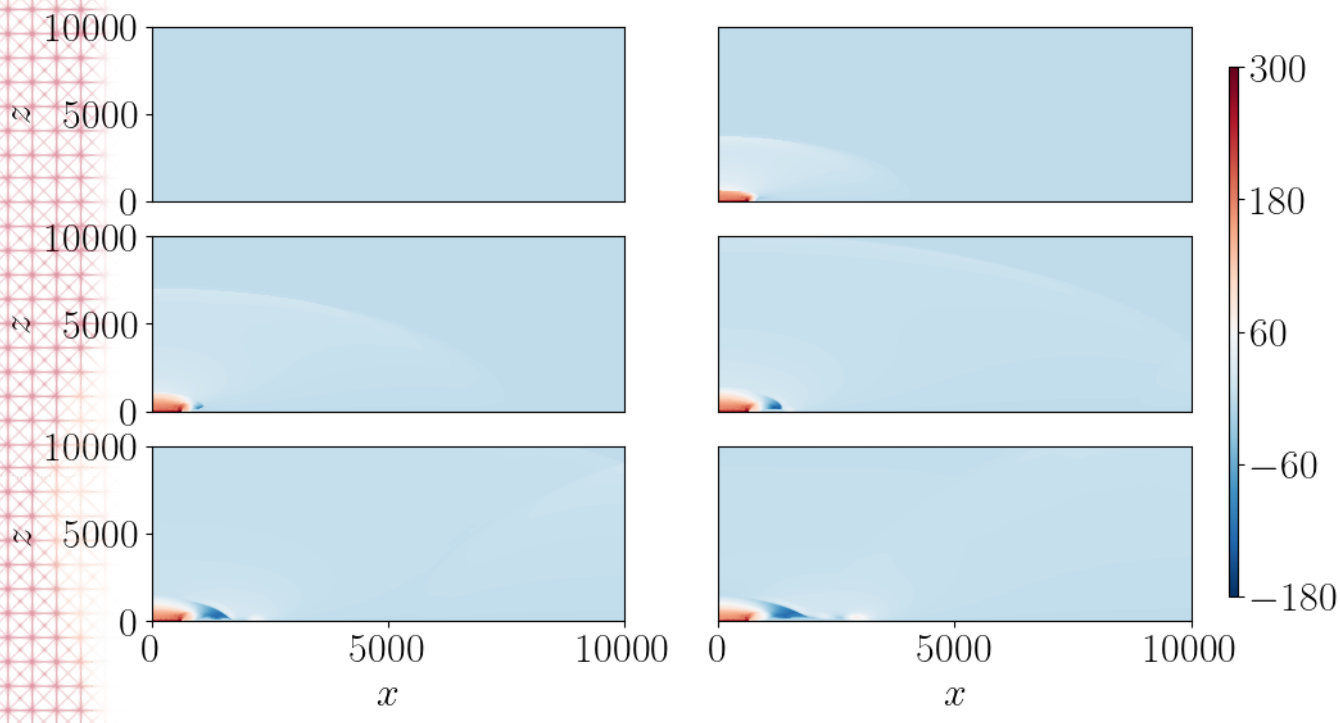

Figure 67: Vertical momentum for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere for deal.II. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. Within the plots, an acoustic wave travelling outwards (from the vent) is barely visible. As with the density plot, a collapse of the jet is visible with increasing time

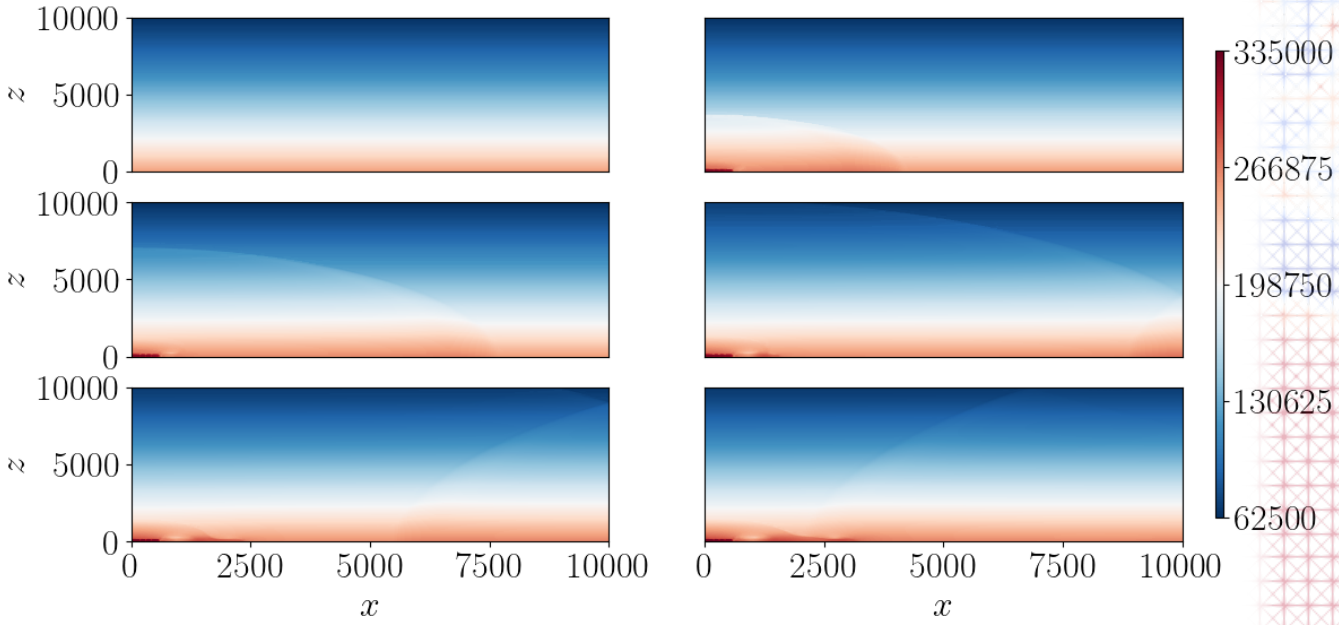

Figure 68: Energy density for the uniform run of the volcanic jet without water vapor and ash with the same temperature as the surrounding atmosphere for deal.II. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. Within the plots, an acoustic wave travelling outwards (from the vent) is barely visible. As with the density plot, a collapse of the jet is visible with increasing time

E Plots for the adaptive jet or plume setups

This appendix show the plots for the results of the volcanic plumes with adaptive mesh refinement. To ensure a quicker overview, the following sections are labeled with T_m which states the temperature of the ejected material, σ_{ref} or σ_{coarse} for refinement or coarsening tolerances and **e** for the error indicator.

In the following, single-phase (air) run means that no water vapor and ash are simulated and the ejected material is just comprised of air. The cases are labeled as in **Section 8.3**. Furthermore, two-phase (water) run means that water vapor is included in the simulation and the erupted material is comprised completely of water vapor that gets injected into an atmosphere of air which then is mixed. The cases are labeled as in **Section 8.4**.

Finally, plume run means that both ash and water vapor are present in the eruption column while the atmosphere is still comprised of (dry) air. The cases are labeled as in **Section 8.5**.

E.1 Case A: Adaptive single-phase (air) run with T_m = 300 K, σ_{ref} = 0.01, σ_{coarse} = 0.005, e = $\nabla(\rho w)$ and no-slip boundary for the ground

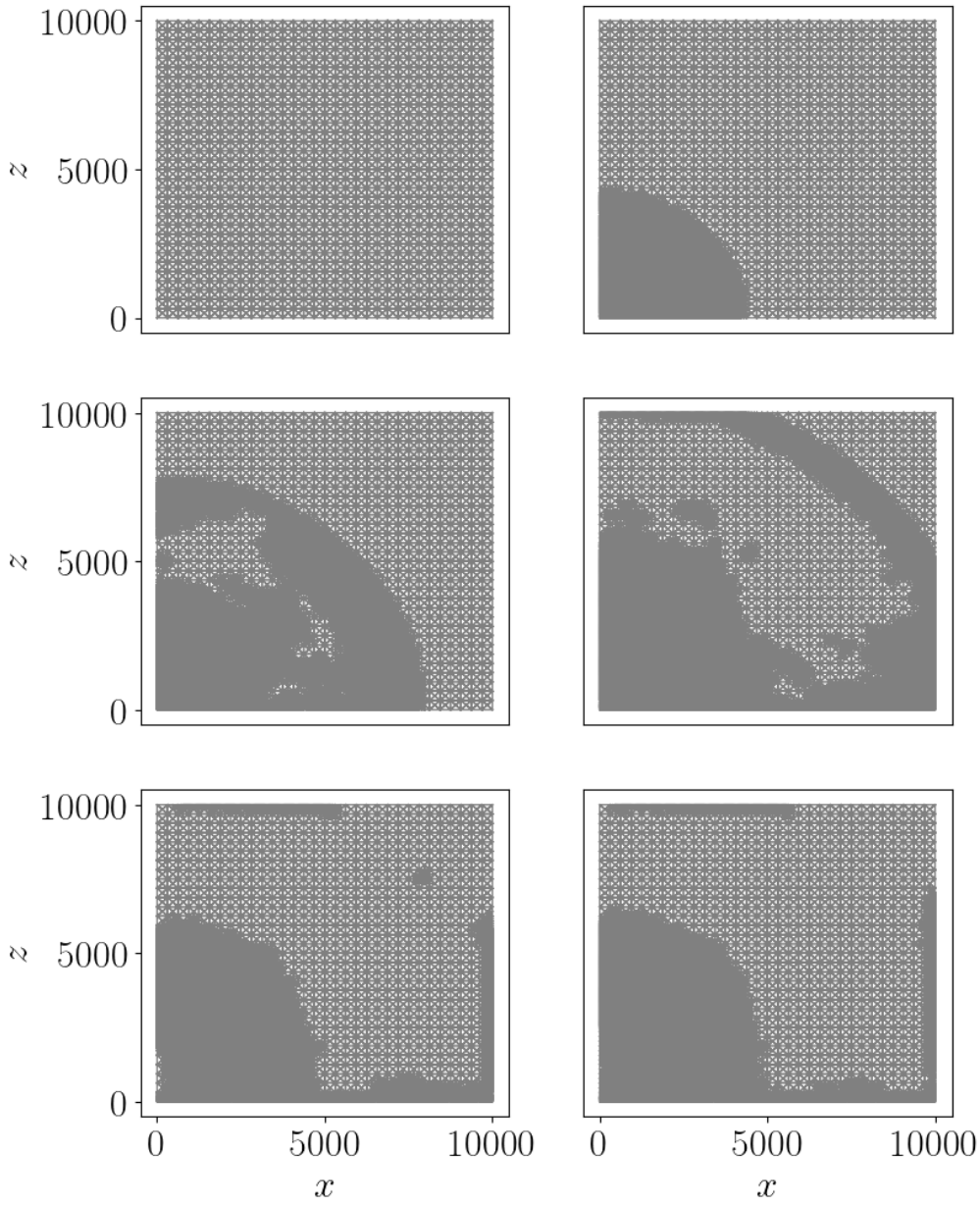

Figure 69: Grid for the adaptive run of the volcanic jet for Case A. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The plots show the development of the adaptive mesh for the respective times. For this setup, the gradient for the vertical momentum was used as error indicator.

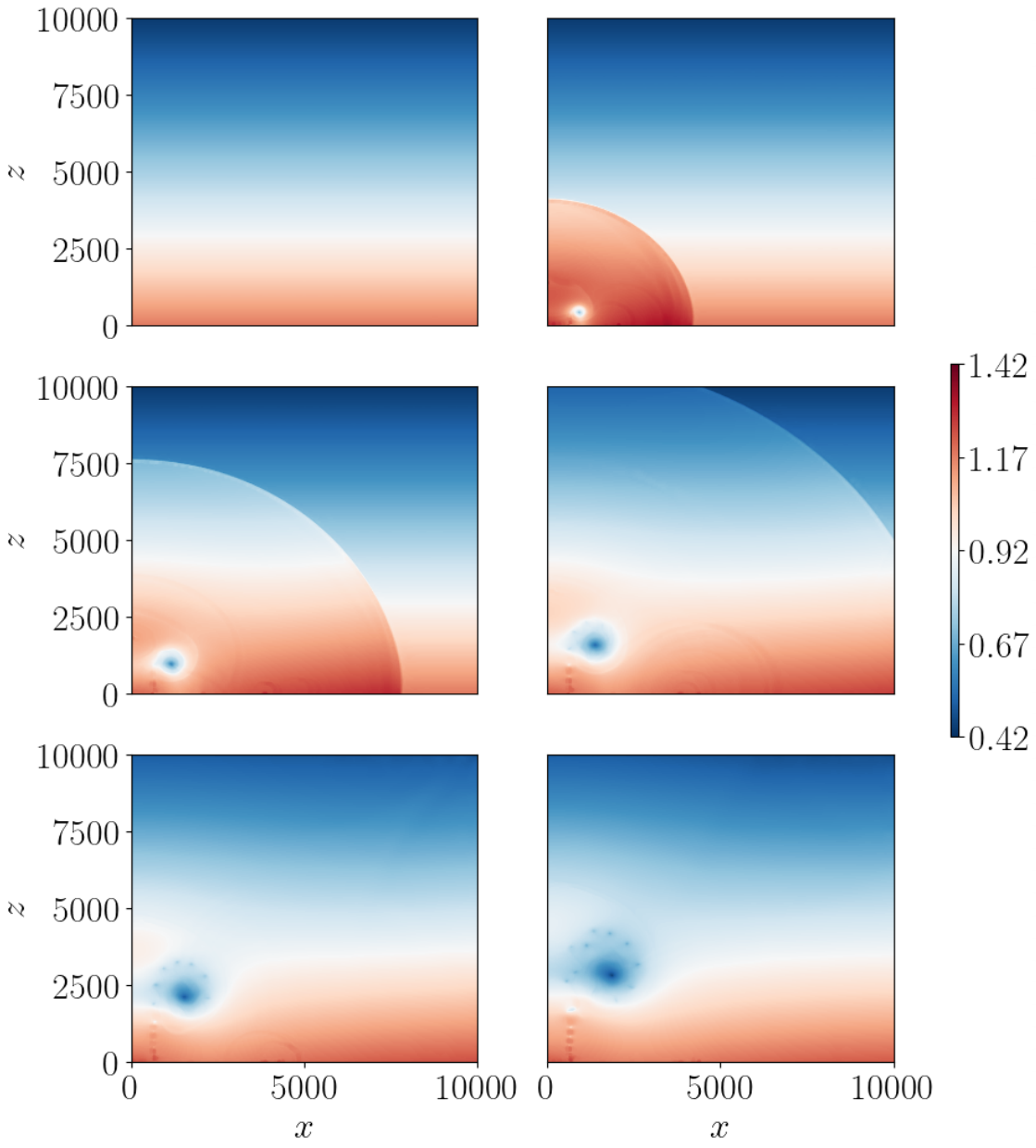

Figure 70: Density for the adaptive run of the volcanic jet for Case A. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

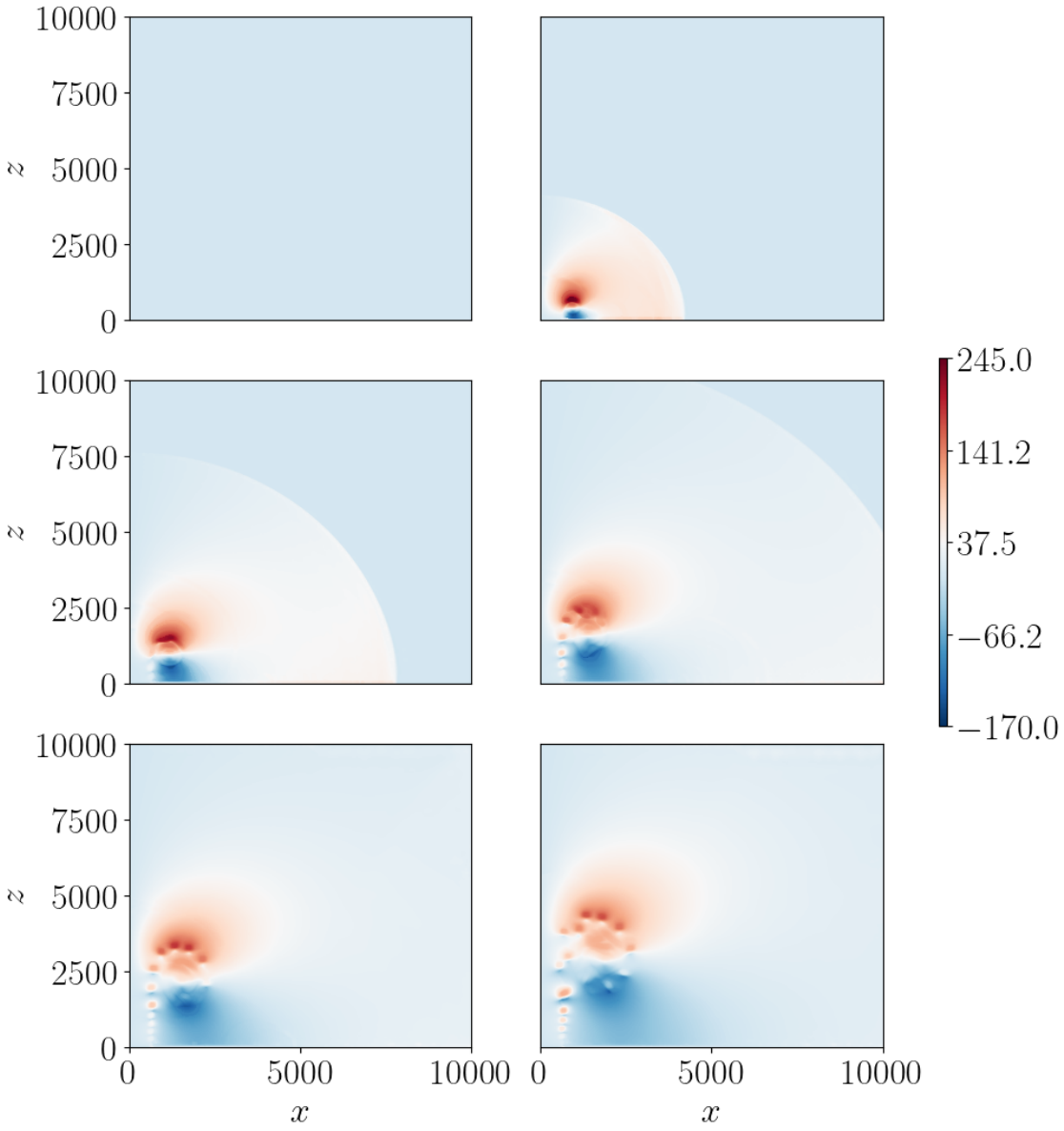

Figure 71: Horizontal momentum for the adaptive run of the volcanic jet for Case A. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

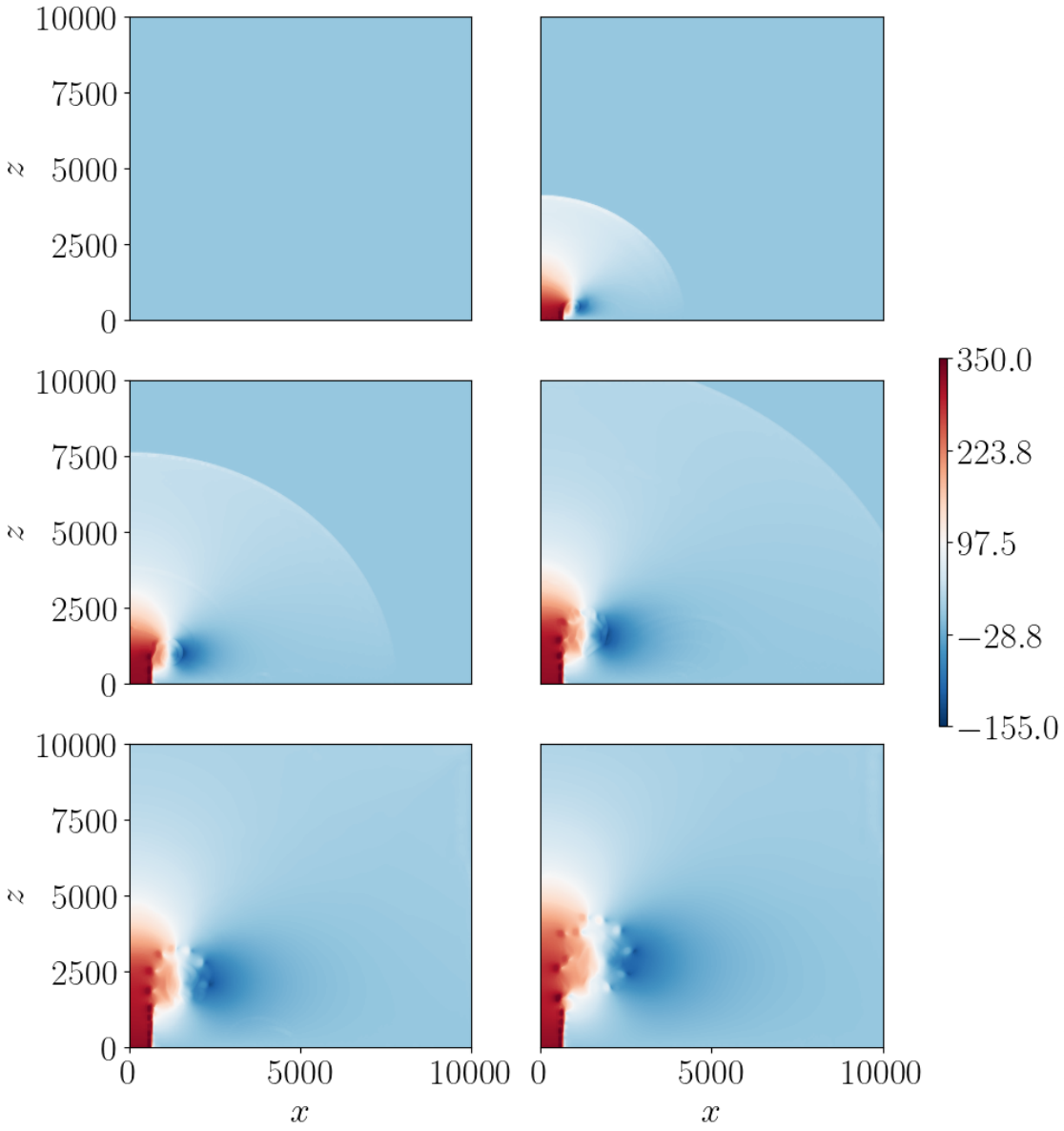

Figure 72: Vertical momentum for the adaptive run of the volcanic jet for Case A. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

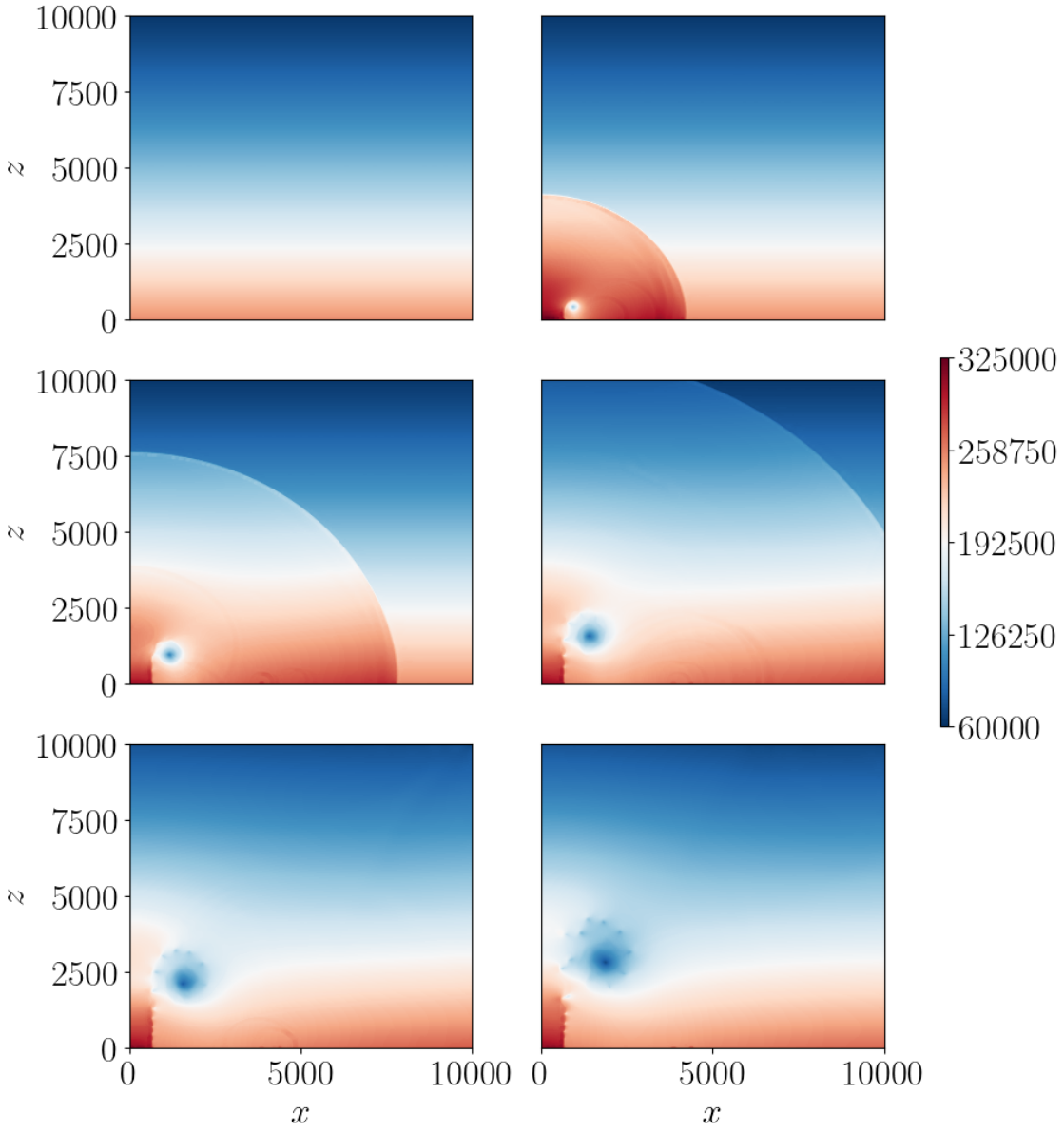

Figure 73: Energy density for the adaptive run of the volcanic jet for Case A. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

E.2 Case B: Adaptive single-phase (air) run with T_m = 300 K, σ_{ref} = 0.001, σ_{coarse} = 0.0005, $\mathbf{e} = \nabla(\rho w)$ and no-slip boundary for the ground

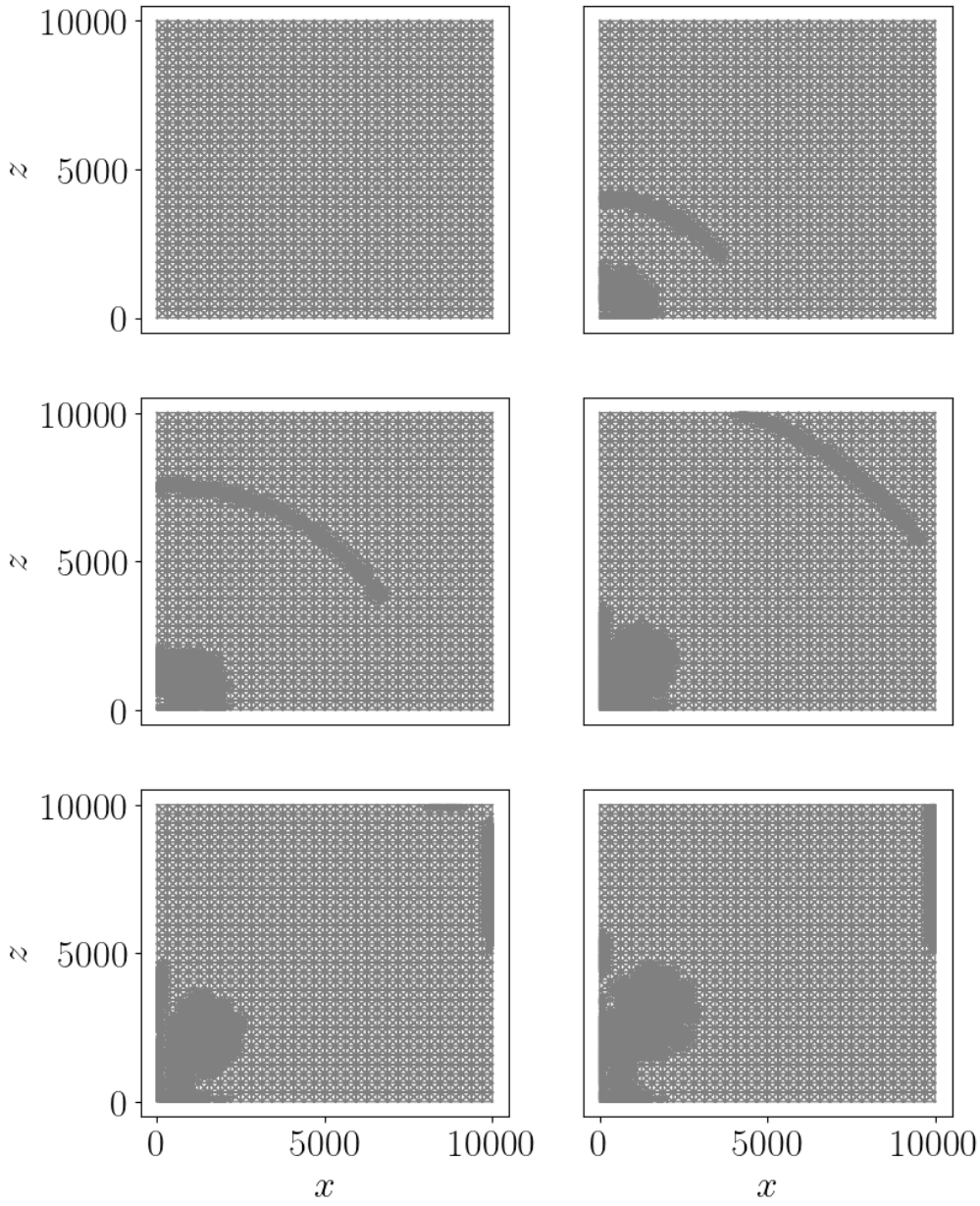

Figure 74: Grid for the adaptive run of the volcanic jet for Case B. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The plots show the development of the adaptive mesh for the respective times. For this setup, the gradient for the vertical momentum was used as error indicator.

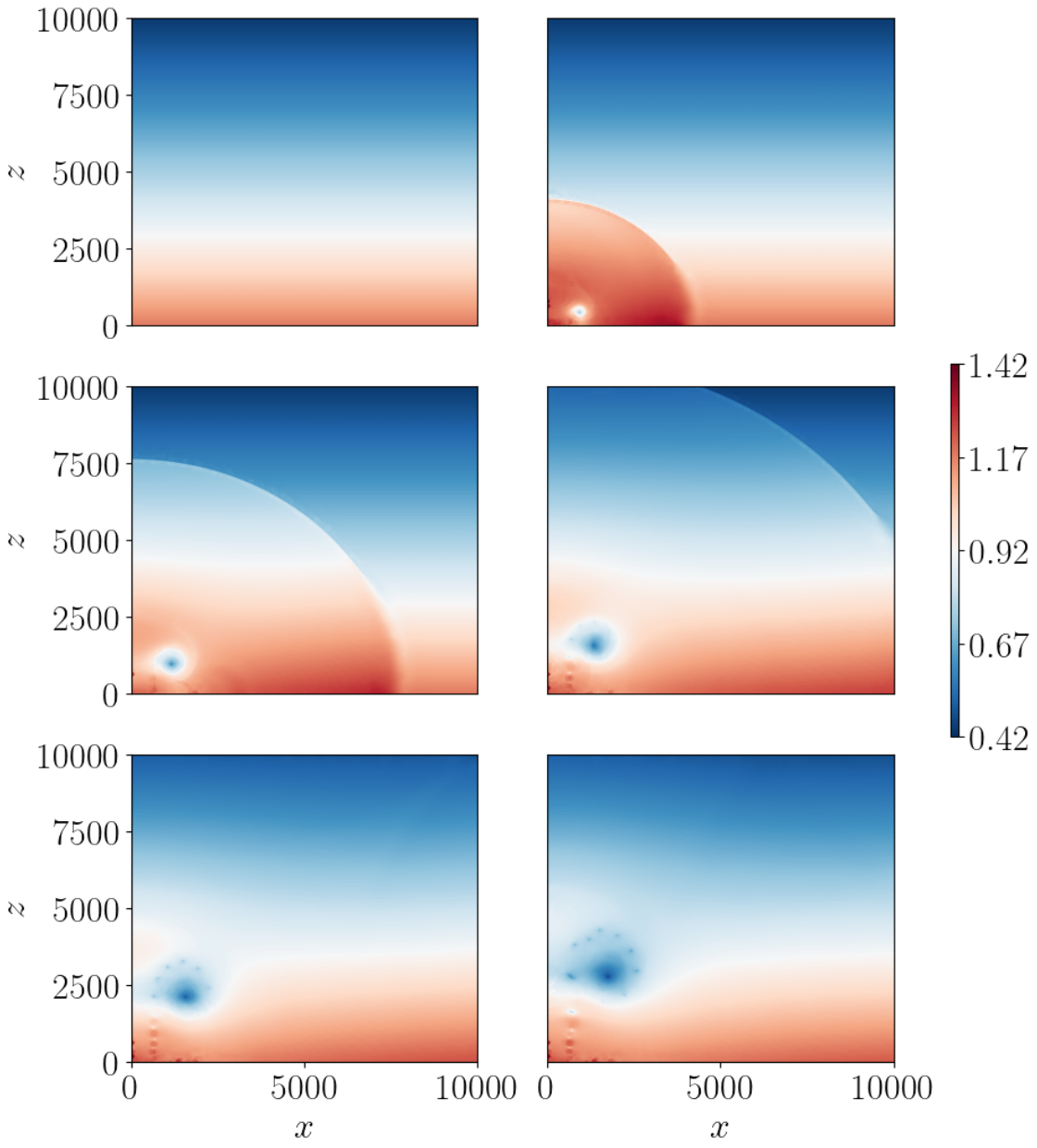

Figure 75: Density for the adaptive run of the volcanic jet for Case B. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

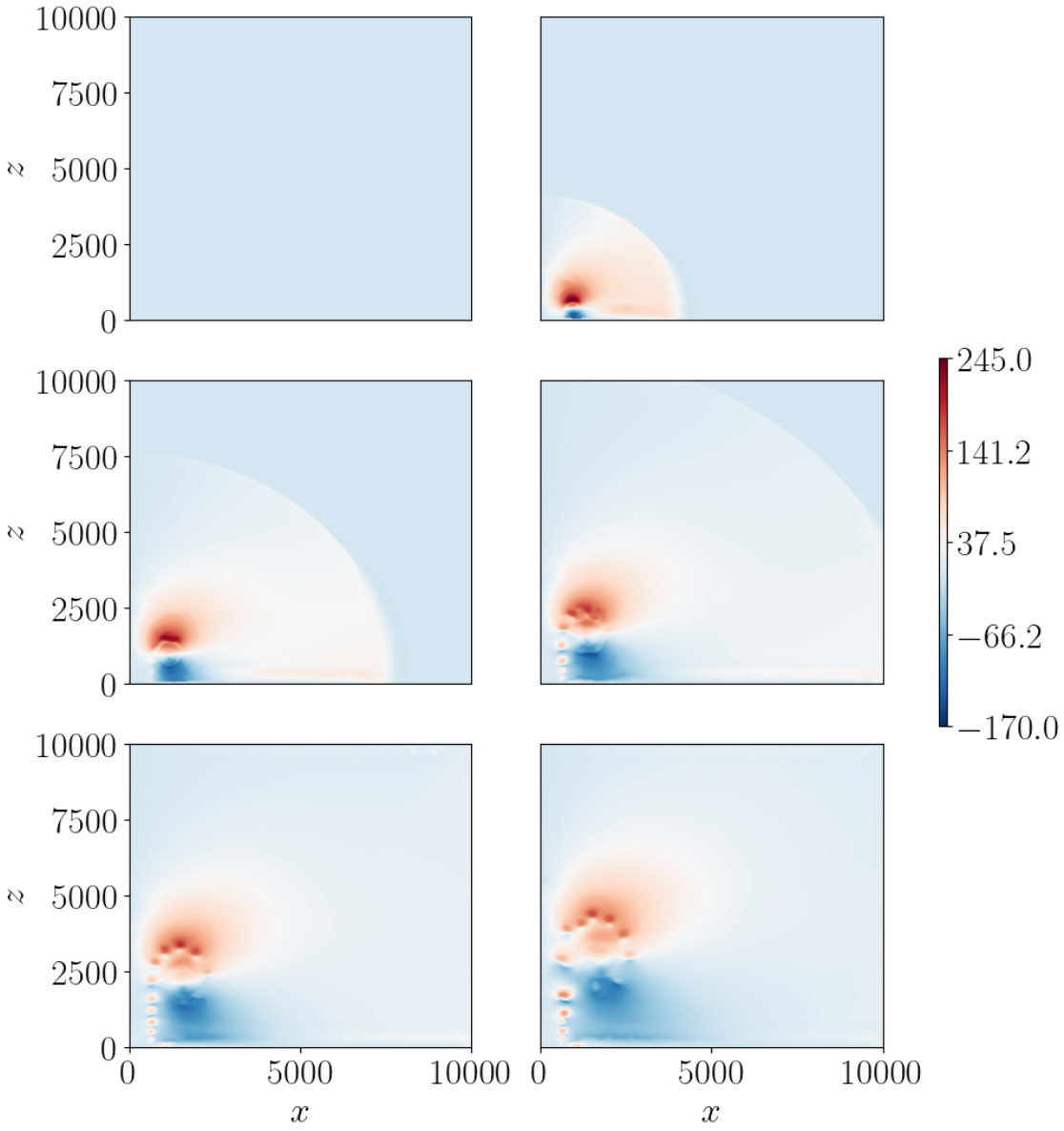

Figure 76: Horizontal momentum for the adaptive run of the volcanic jet for Case B. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

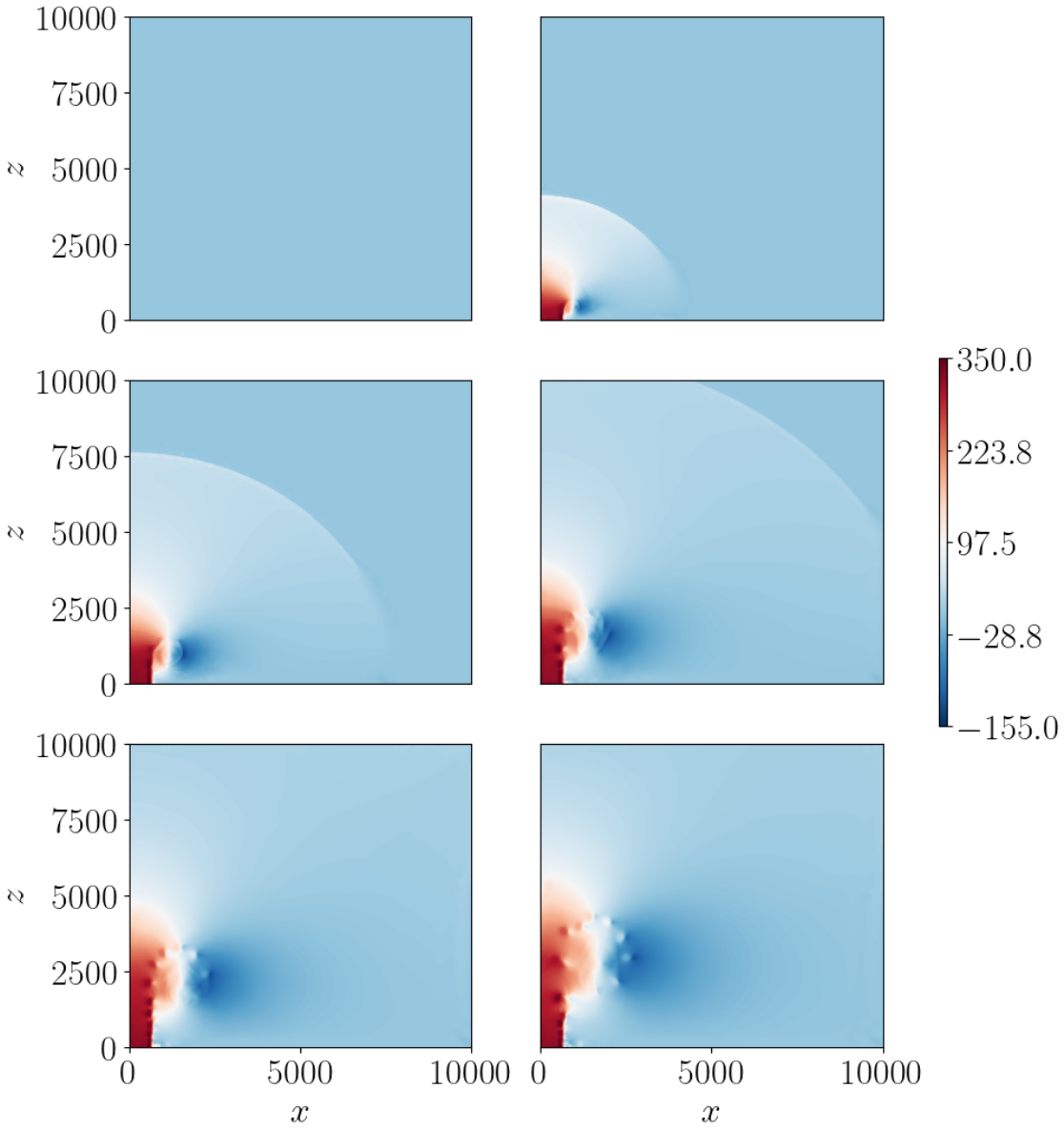

Figure 77: Vertical momentum for the adaptive run of the volcanic jet for Case B. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

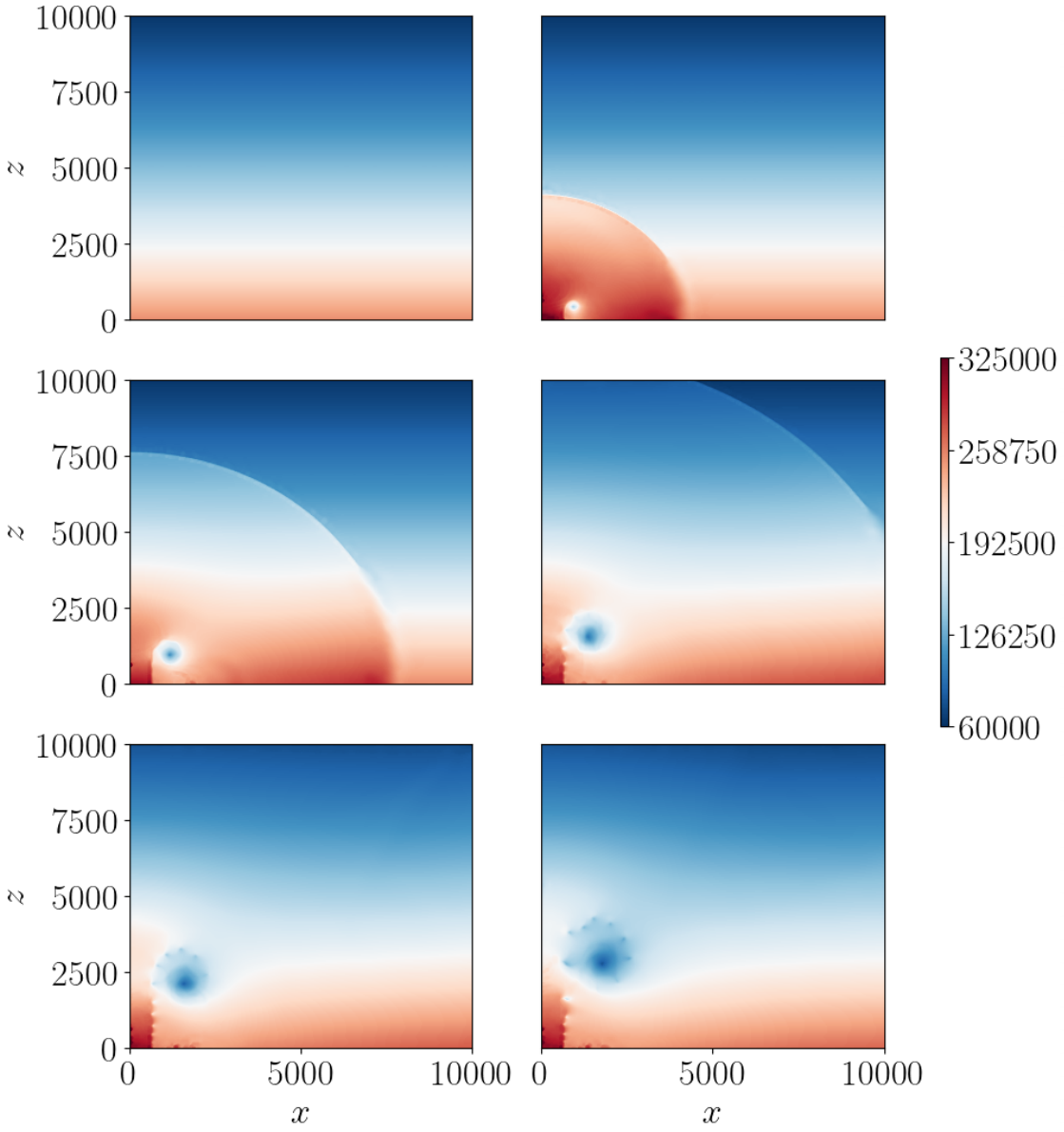

Figure 78: Energy density for the adaptive run of the volcanic jet for Case B. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

E.3 Case C: Adaptive single-phase (air) run with T_m = 300 K, σ_{ref} = 0.01, σ_{coarse} = 0.005, $\mathbf{e} = \max|\nabla(\rho\mathbf{u})|$ and Dirichlet boundary for the ground

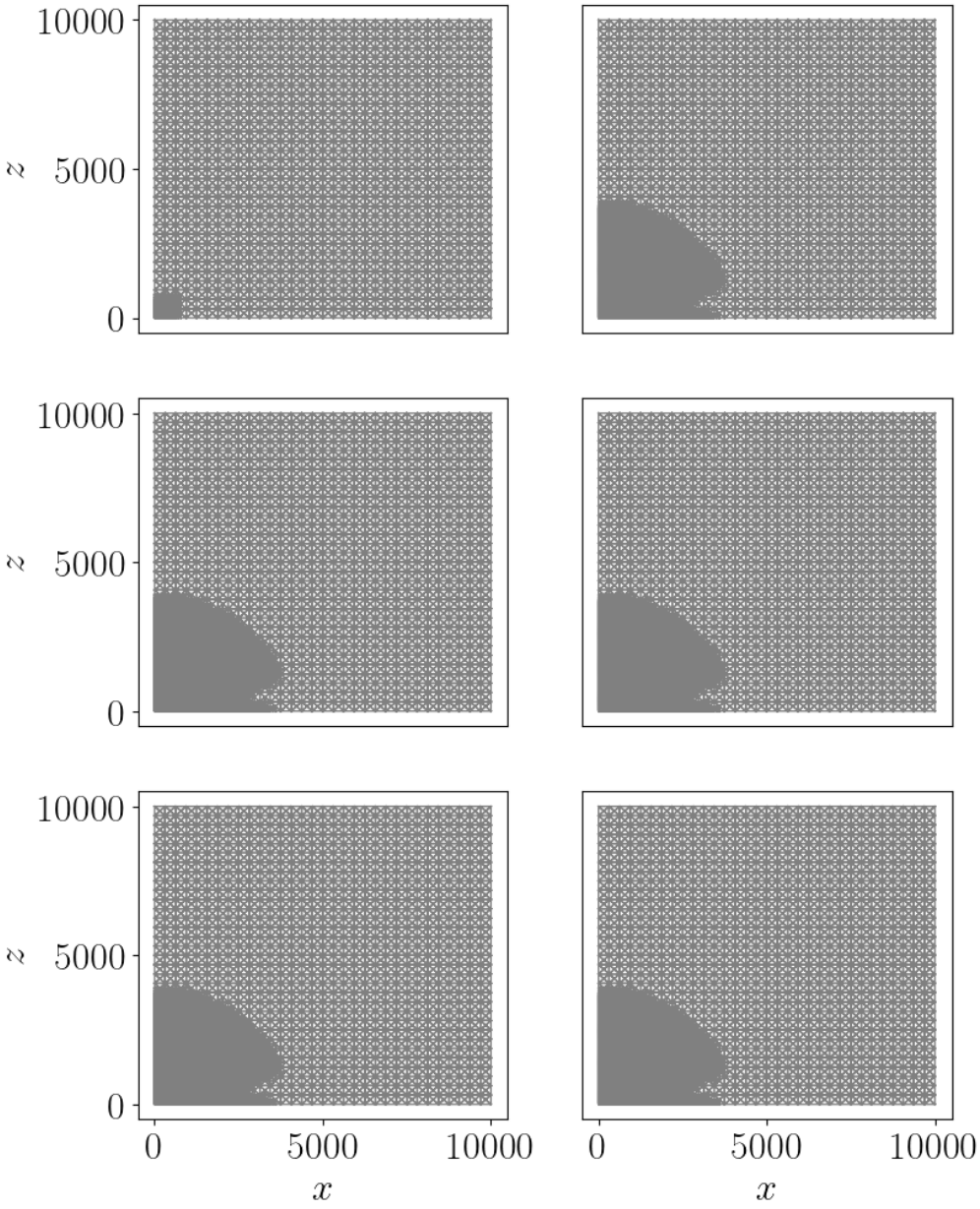

Figure 79: Grid for the adaptive run of the volcanic jet for Case C. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The plots show the development of the adaptive mesh for the respective times. For this setup, the maximum gradient of the momenta was used as error indicator.

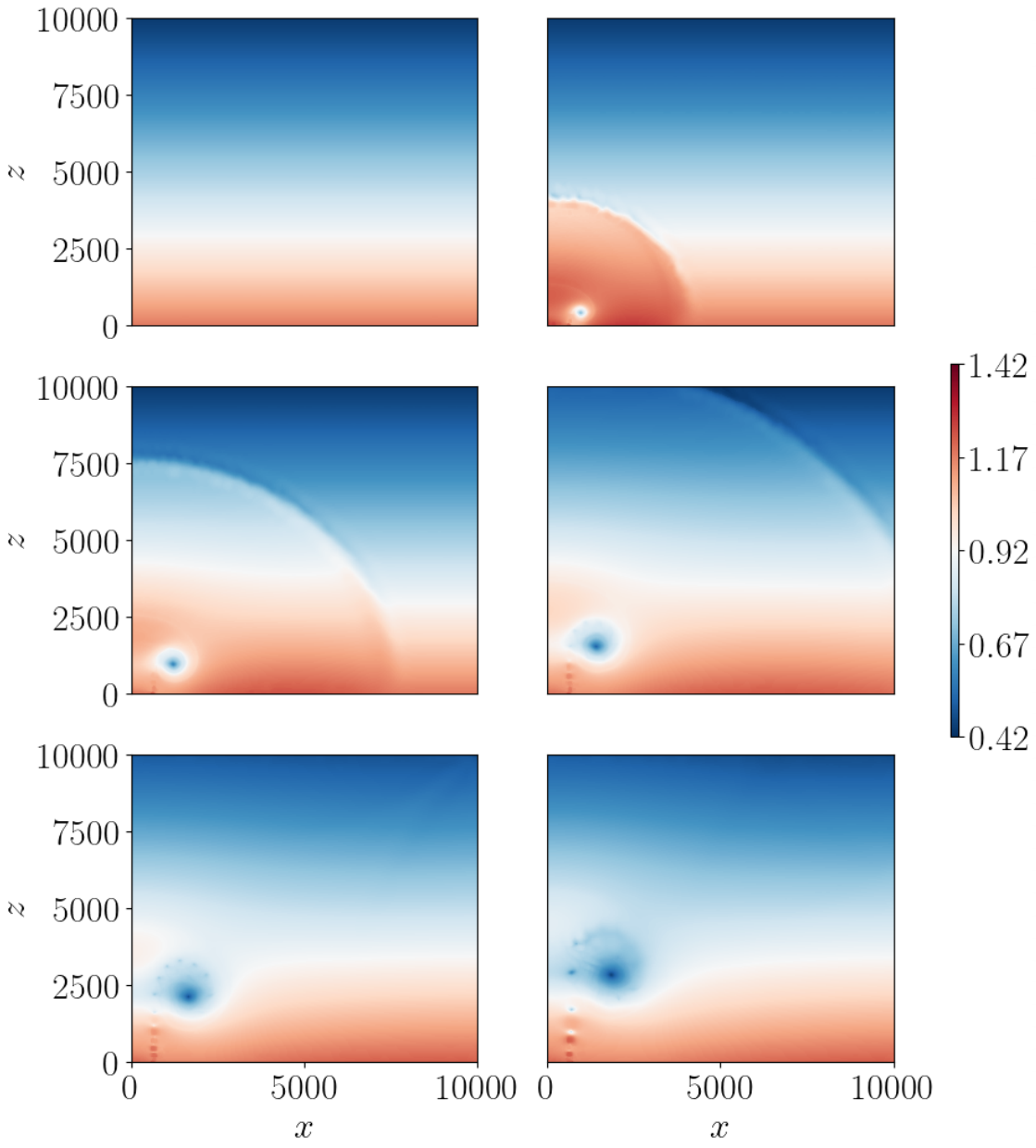

Figure 80: Density for the adaptive run of the volcanic jet for Case C. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

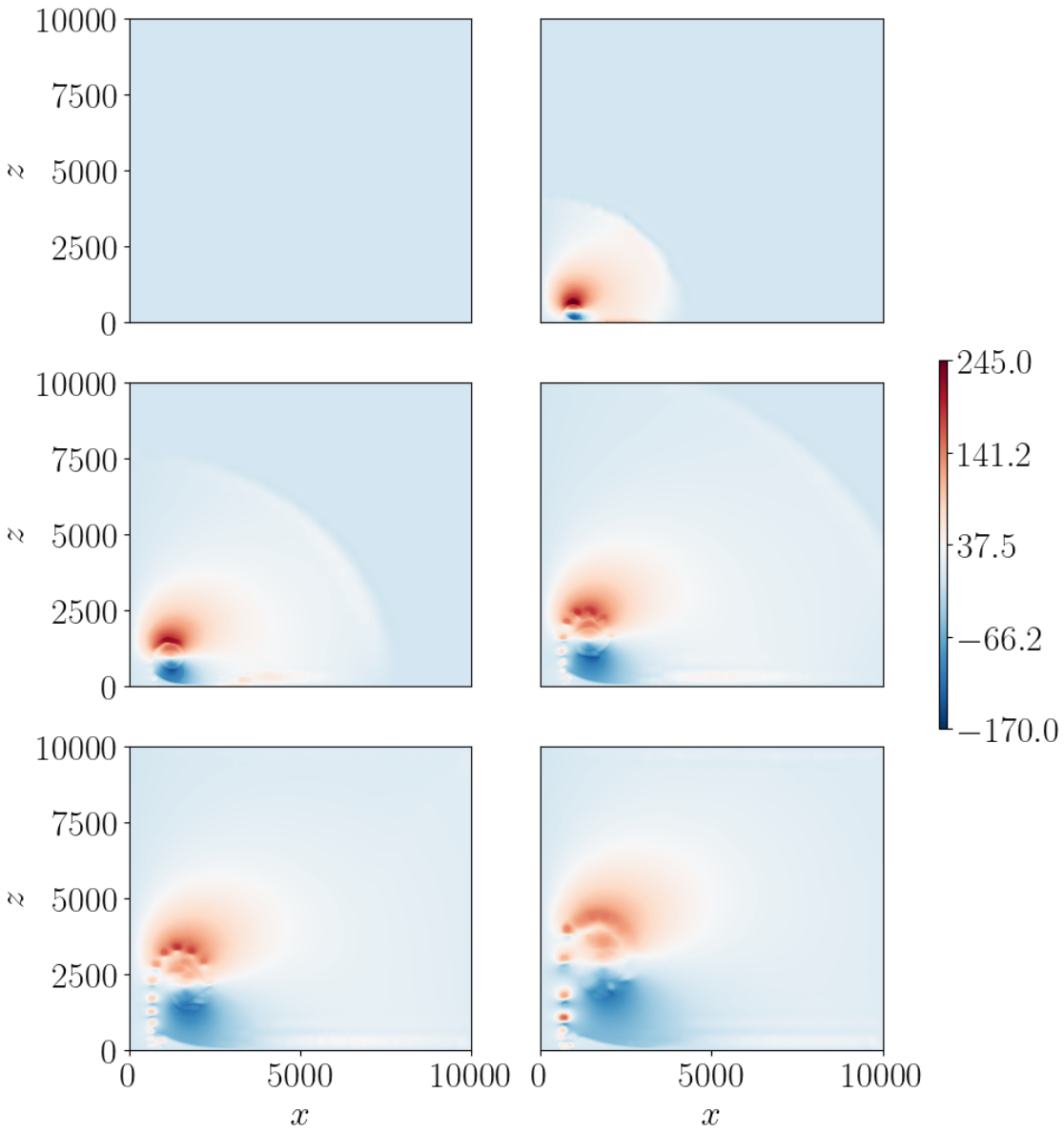

Figure 81: Horizontal momentum for the adaptive run of the volcanic jet for Case C. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, areas of high and low (negative) momentum, respectively, develop which are situated around a center that is roughly located at (2000 m, 3000 m) after $t_{\max}$ is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

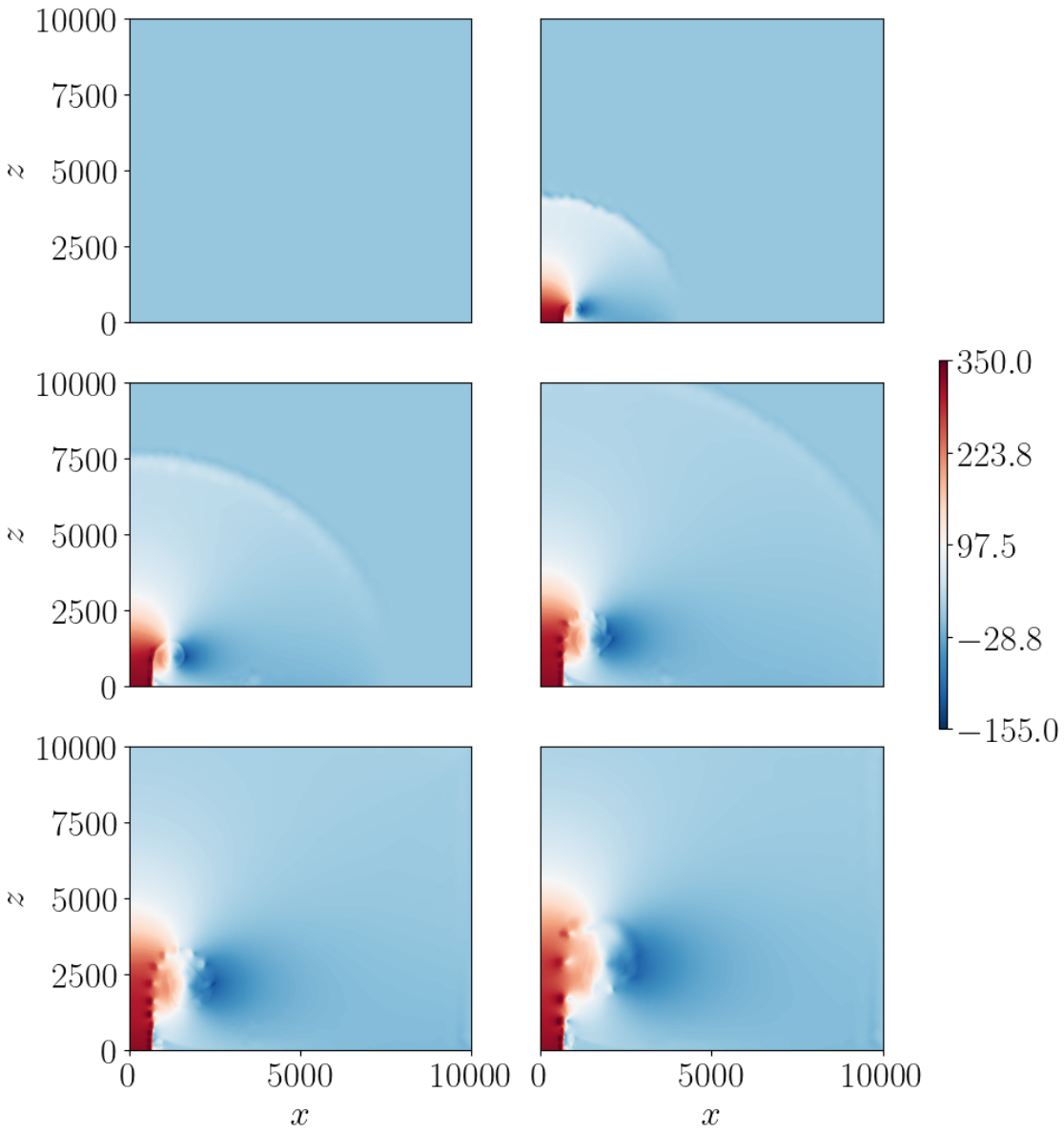

Figure 82: Vertical momentum for the adaptive run of the volcanic jet for Case C. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, areas of high and low (negative) momentum, respectively, develop which are situated around a center that is roughly located at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

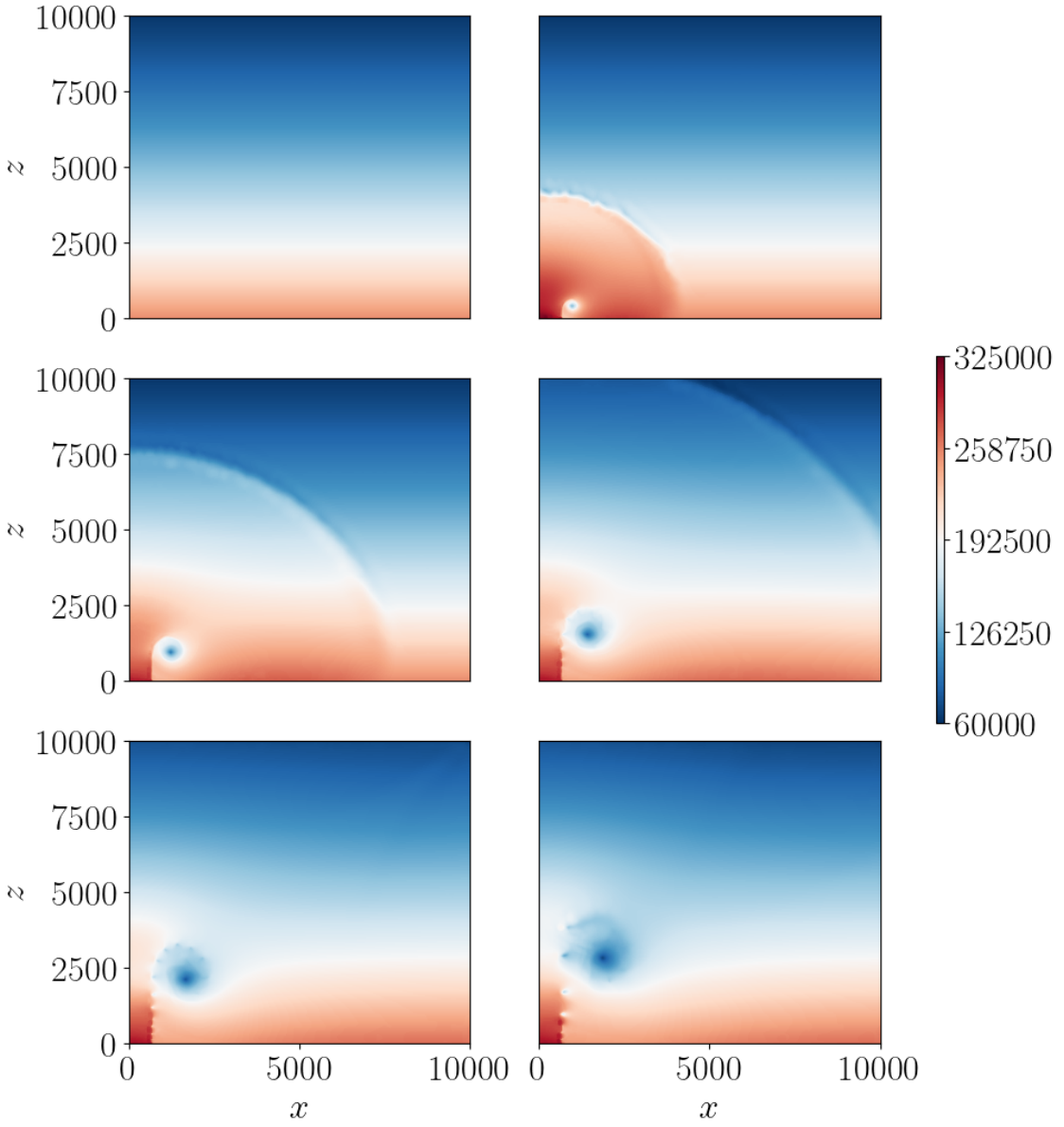

Figure 83: Energy density for the adaptive run of the volcanic jet for Case C. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. With time, an area of lower energy density develops which is roughly centered at (2000 m, 3000 m) after t_{max} is reached. Effects which are similar to Kelvin-Helmholtz instabilities occur along the path which appears to be the trajectory of the air parcels that make up the large eddy structure.

E.4 Case D: Adaptive single-phase (air) run with T_m = 1053 K, σ_{ref} = 0.01, σ_{coarse} = 0.005, $\mathbf{e} = \max|\nabla(\rho\mathbf{u})|$ and three additional layers of refinement

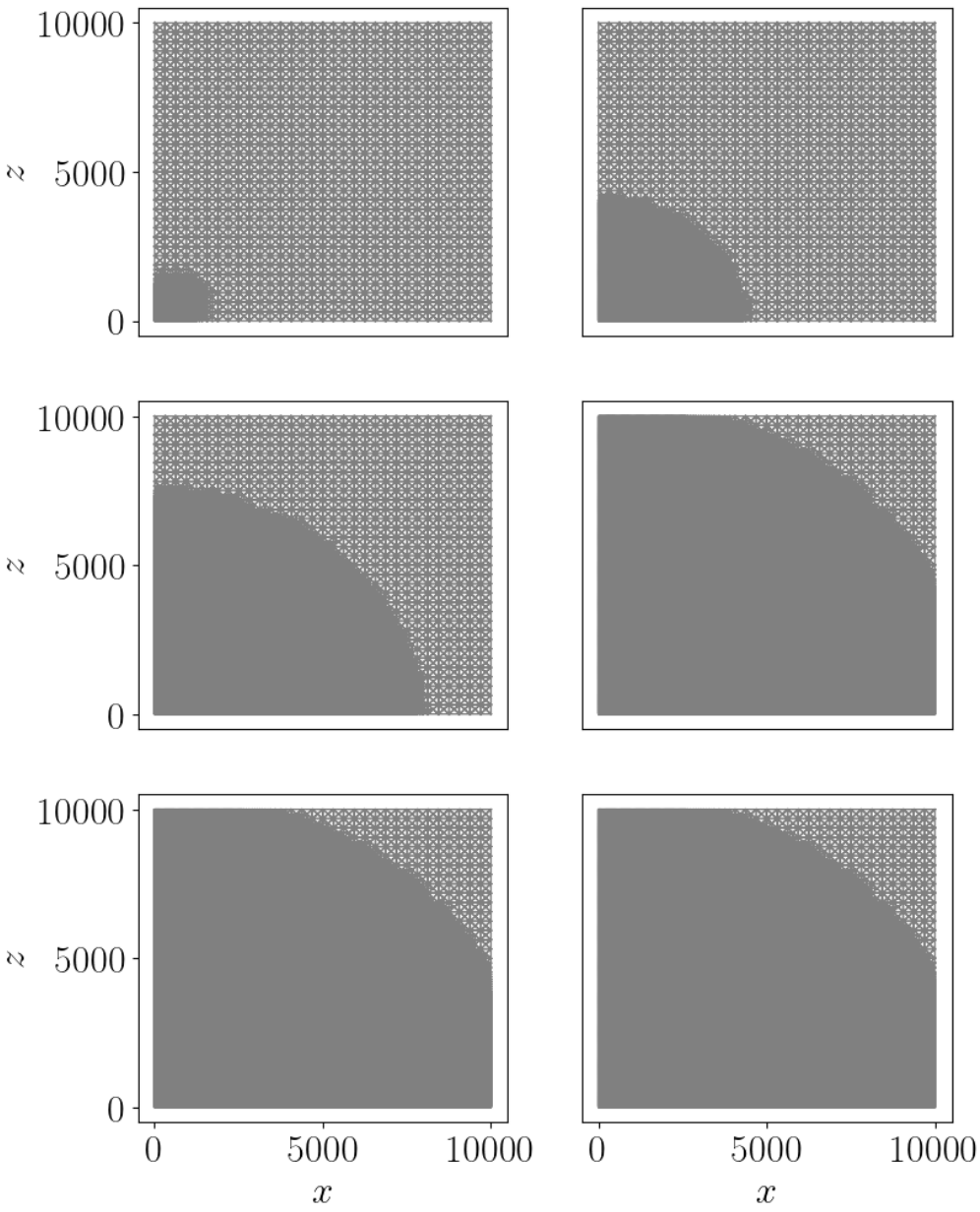

Figure 84: Grid for the adaptive run of the volcanic jet for Case D. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The plots show the development of the adaptive mesh for the respective times. For this setup, the maximum gradient of the momenta was used as error indicator.

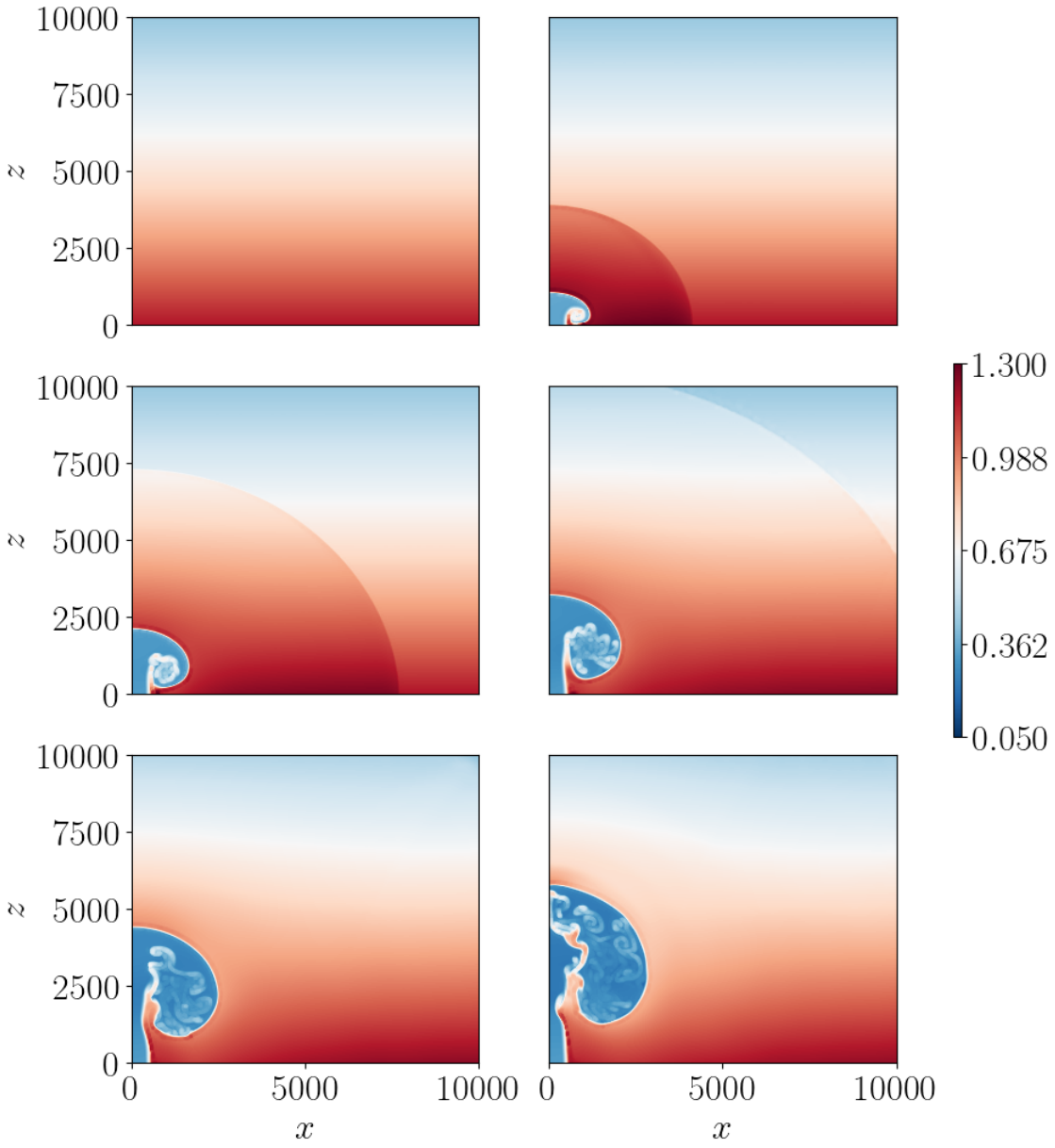

Figure 85: Density for the adaptive run of the volcanic jet for Case D. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

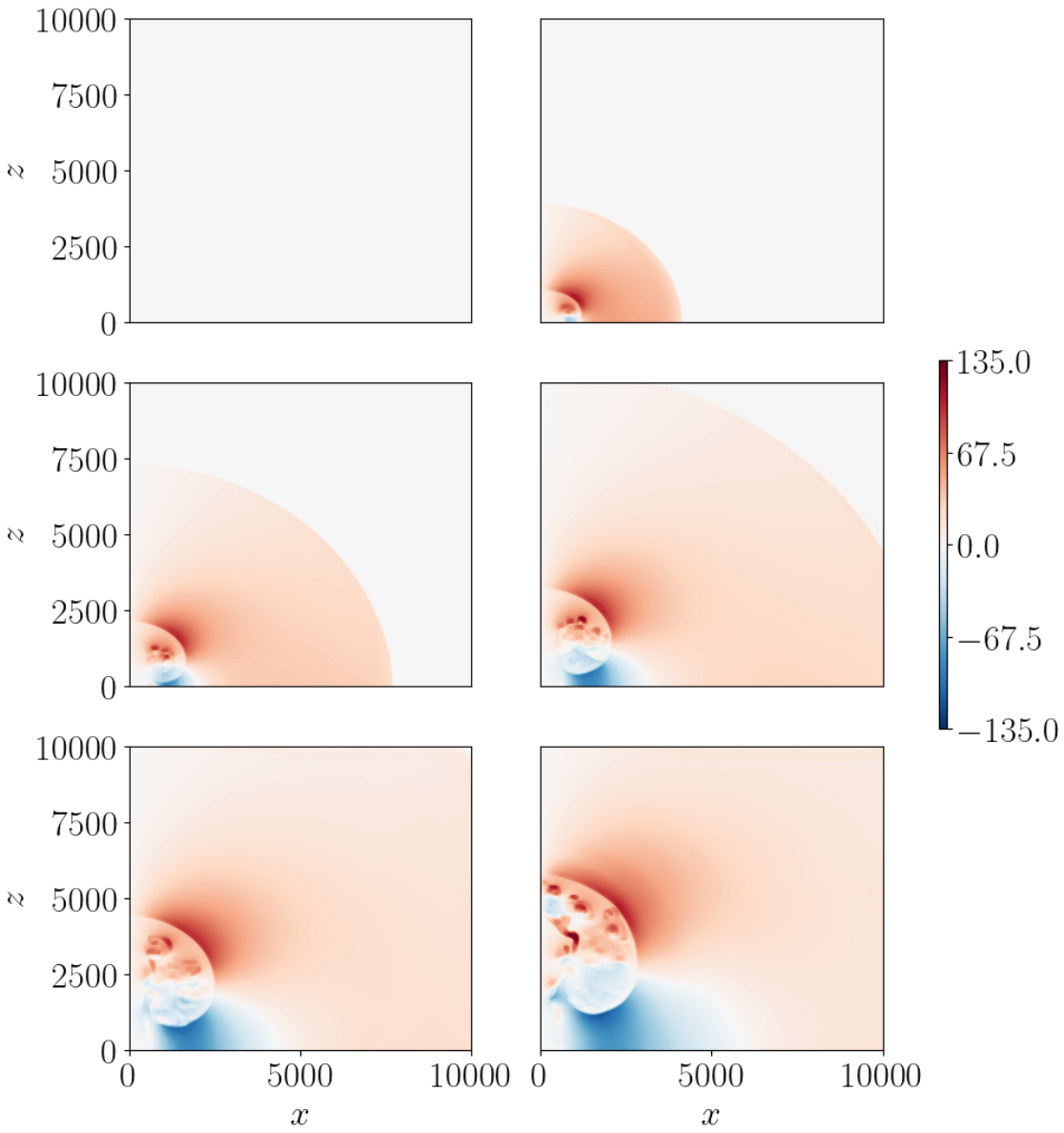

Figure 86: Horizontal momentum for the adaptive run of the volcanic jet for Case D. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

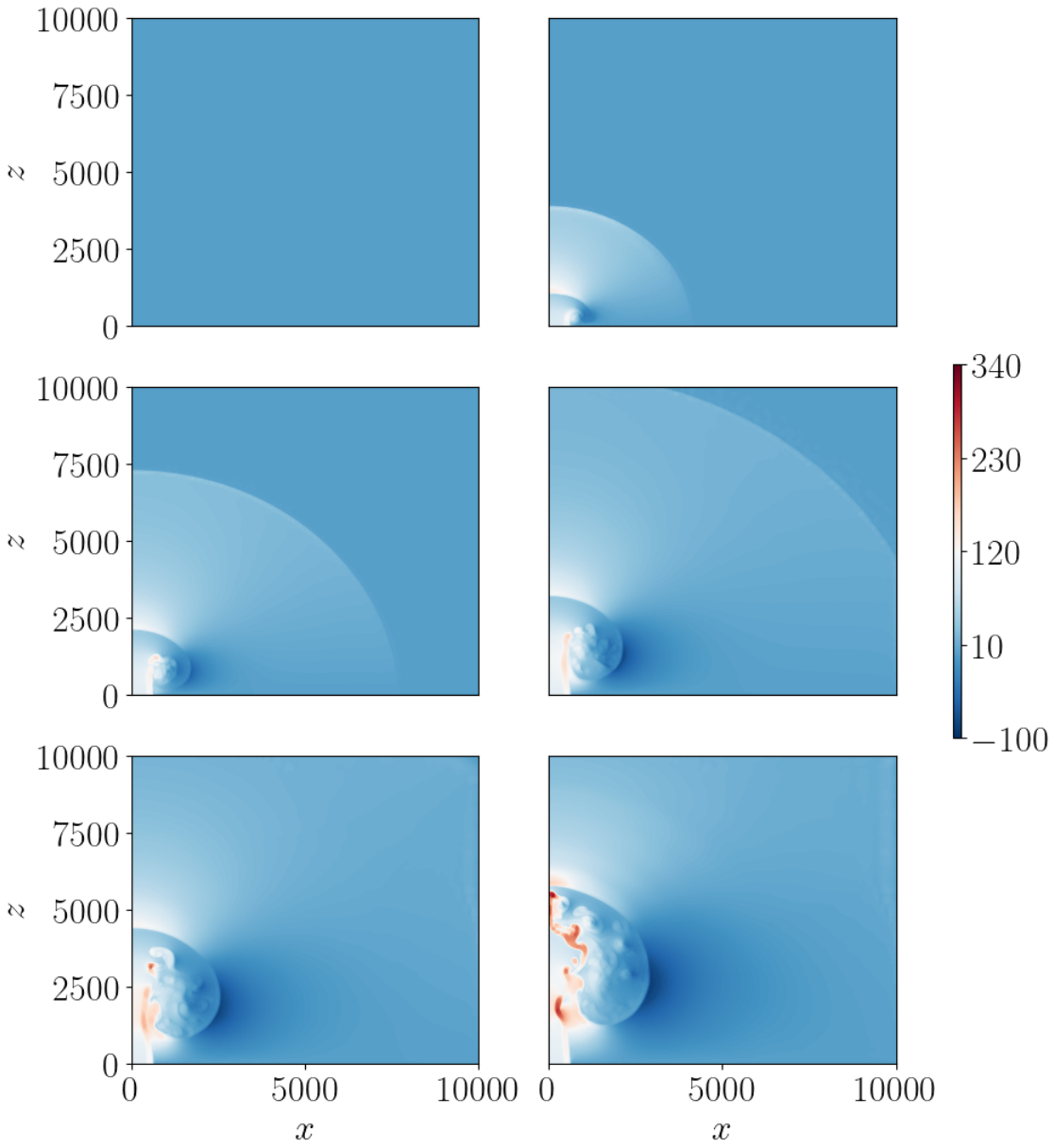

Figure 87: Vertical momentum for the adaptive run of the volcanic jet for Case D. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

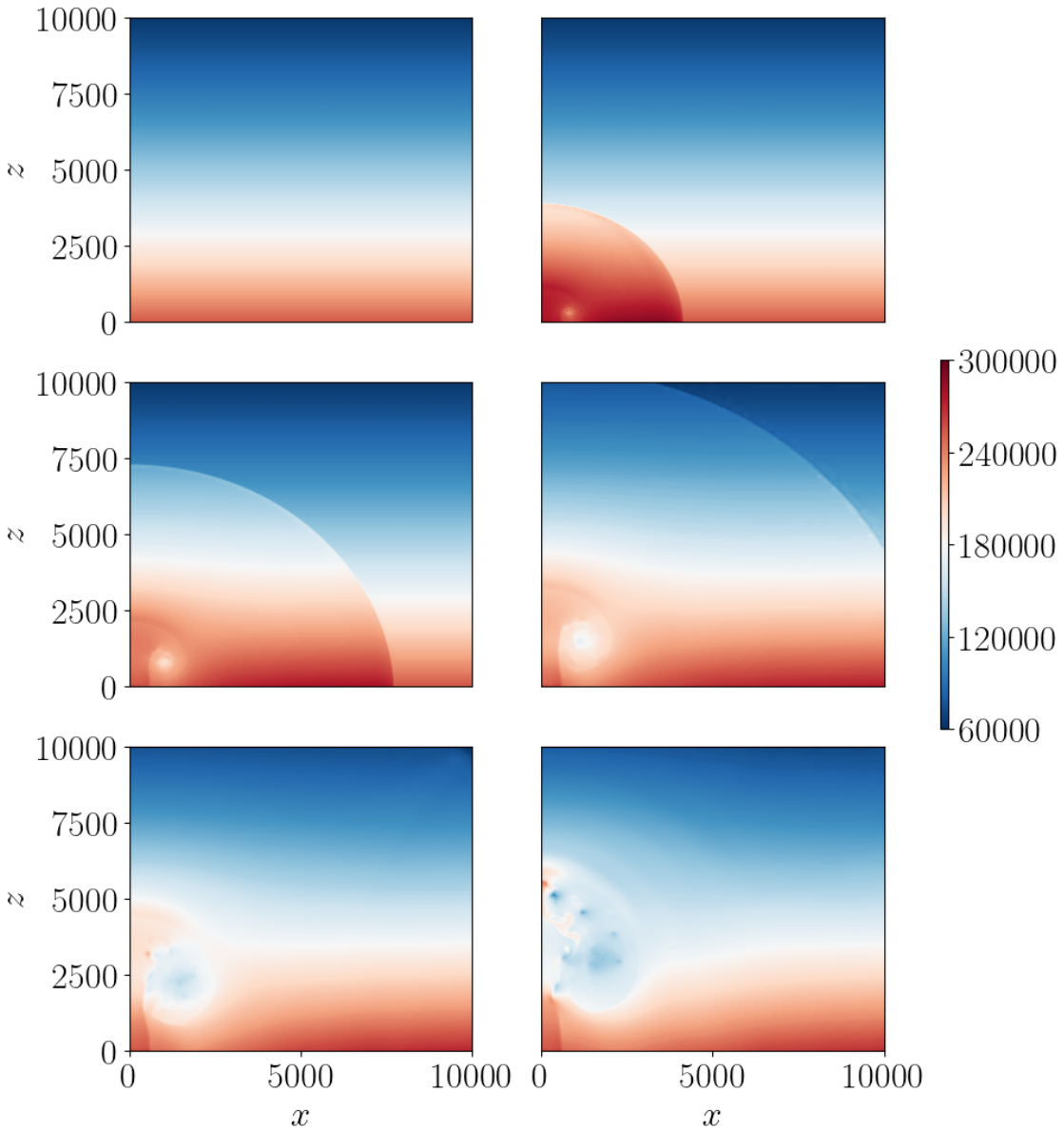

Figure 88: Energy density for the adaptive run of the volcanic jet for Case D. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

E.5 Case E: Adaptive single-phase (air) run with T_m = 1053 K, σ_{ref} = 0.01, σ_{coarse} = 0.01, $\mathbf{e} = \max|\nabla(\rho\mathbf{u})|$ and two additional layers of refinement

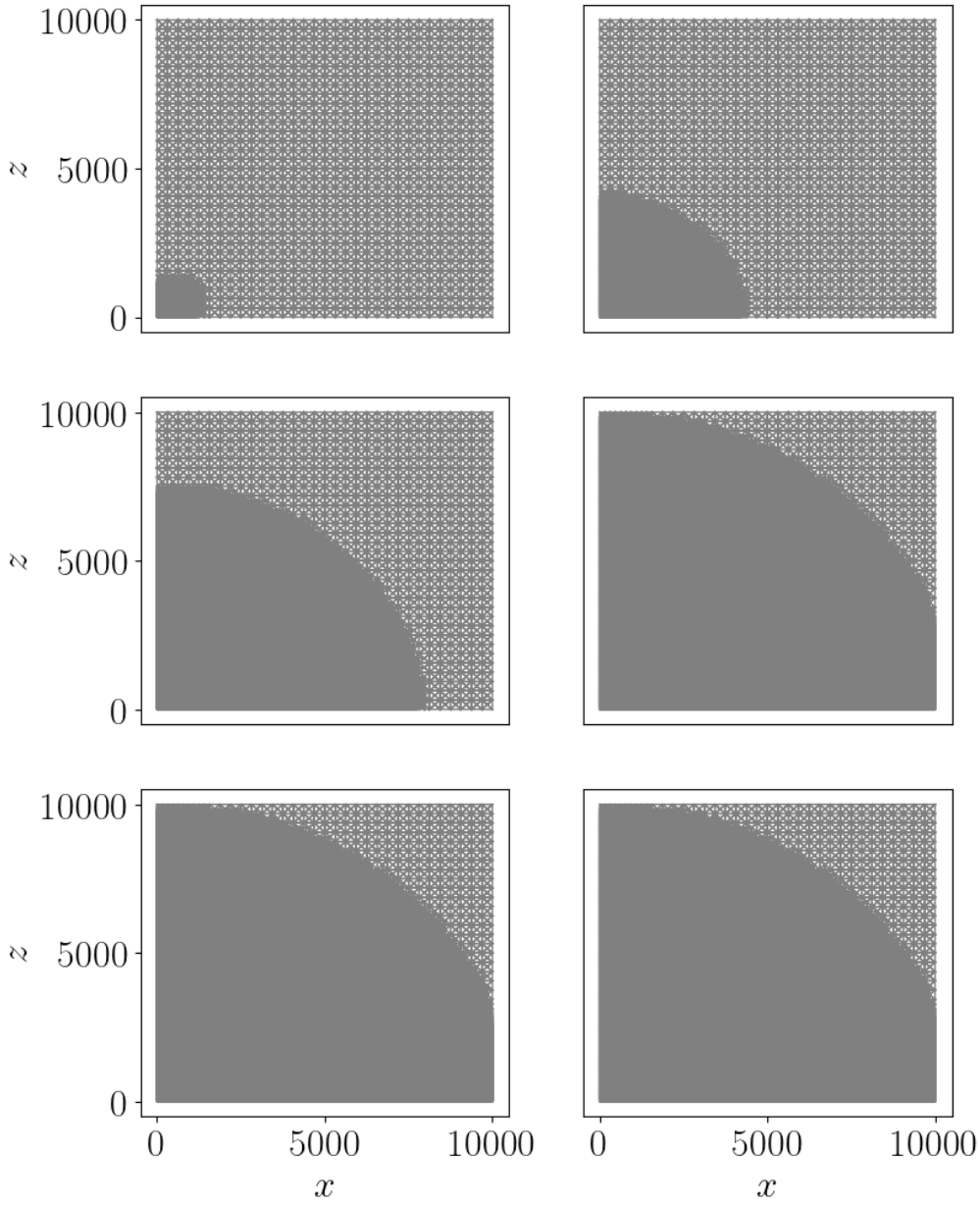

Figure 89: Grid for the adaptive run of the volcanic jet for Case E. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The plots show the development of the adaptive mesh for the respective times. For this setup, the maximum gradient of the momenta was used as error indicator.

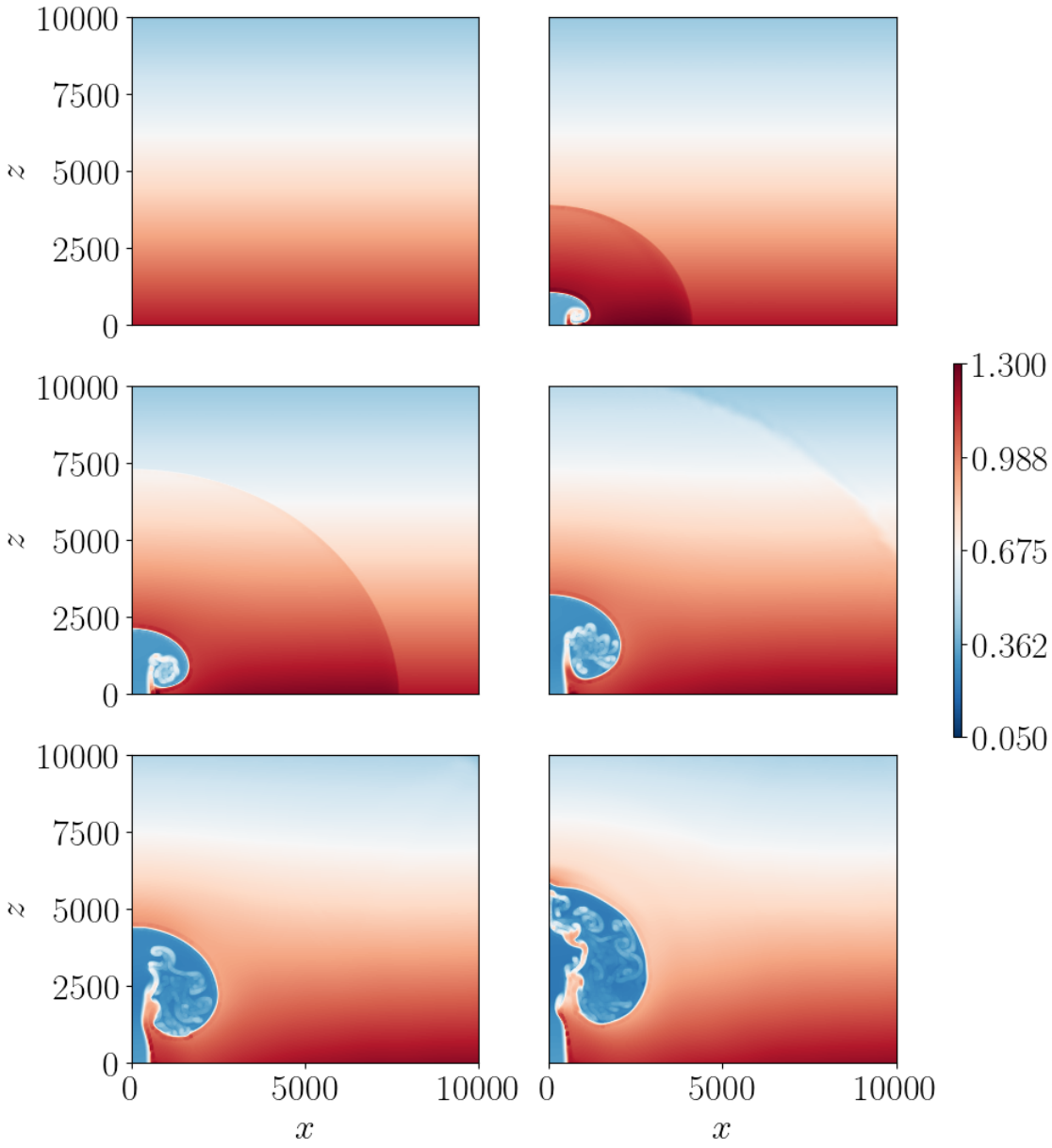

Figure 90: Density for the adaptive run of the volcanic jet for Case E. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

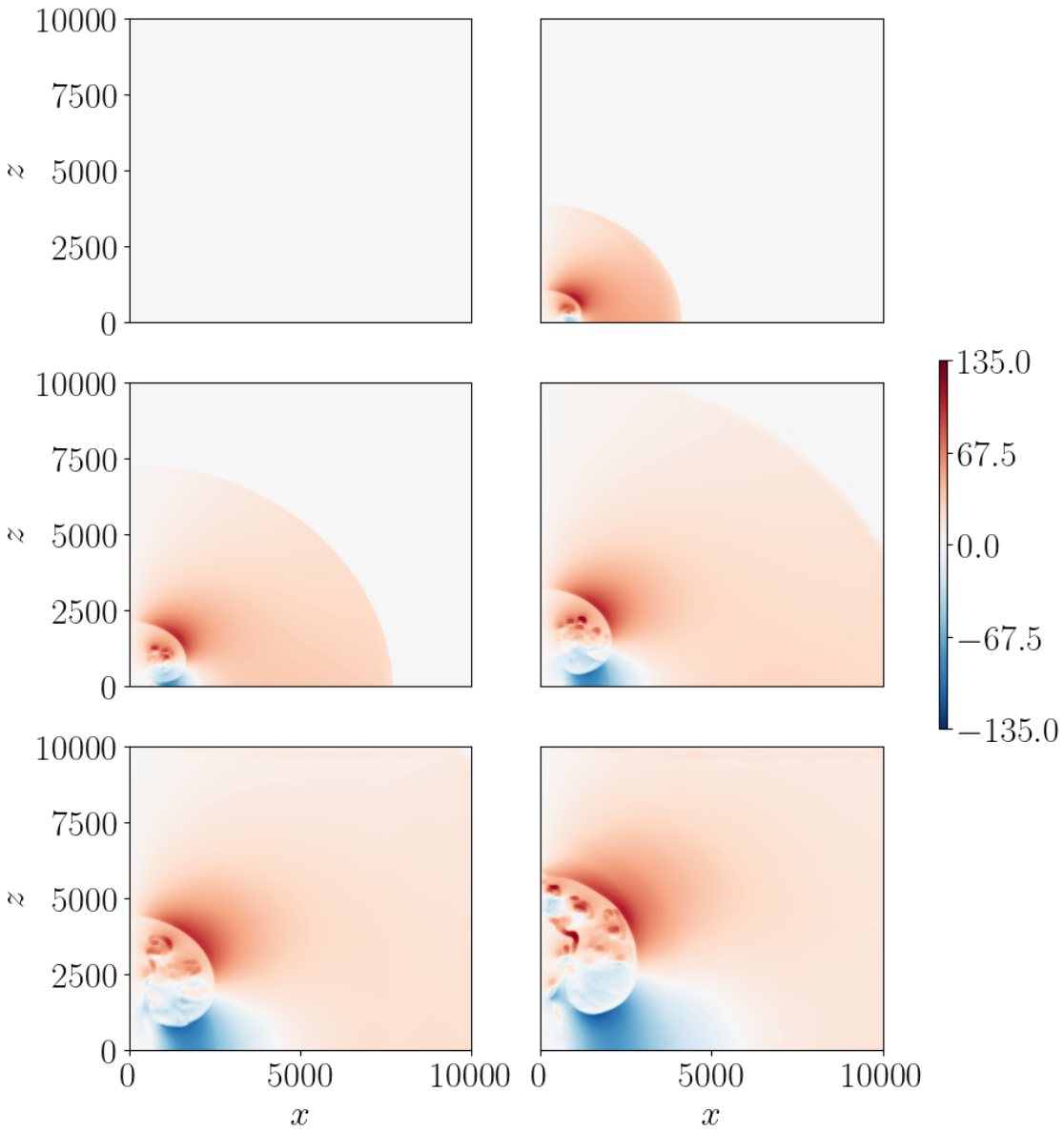

Figure 91: Horizontal momentum for the adaptive run of the volcanic jet for Case E. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

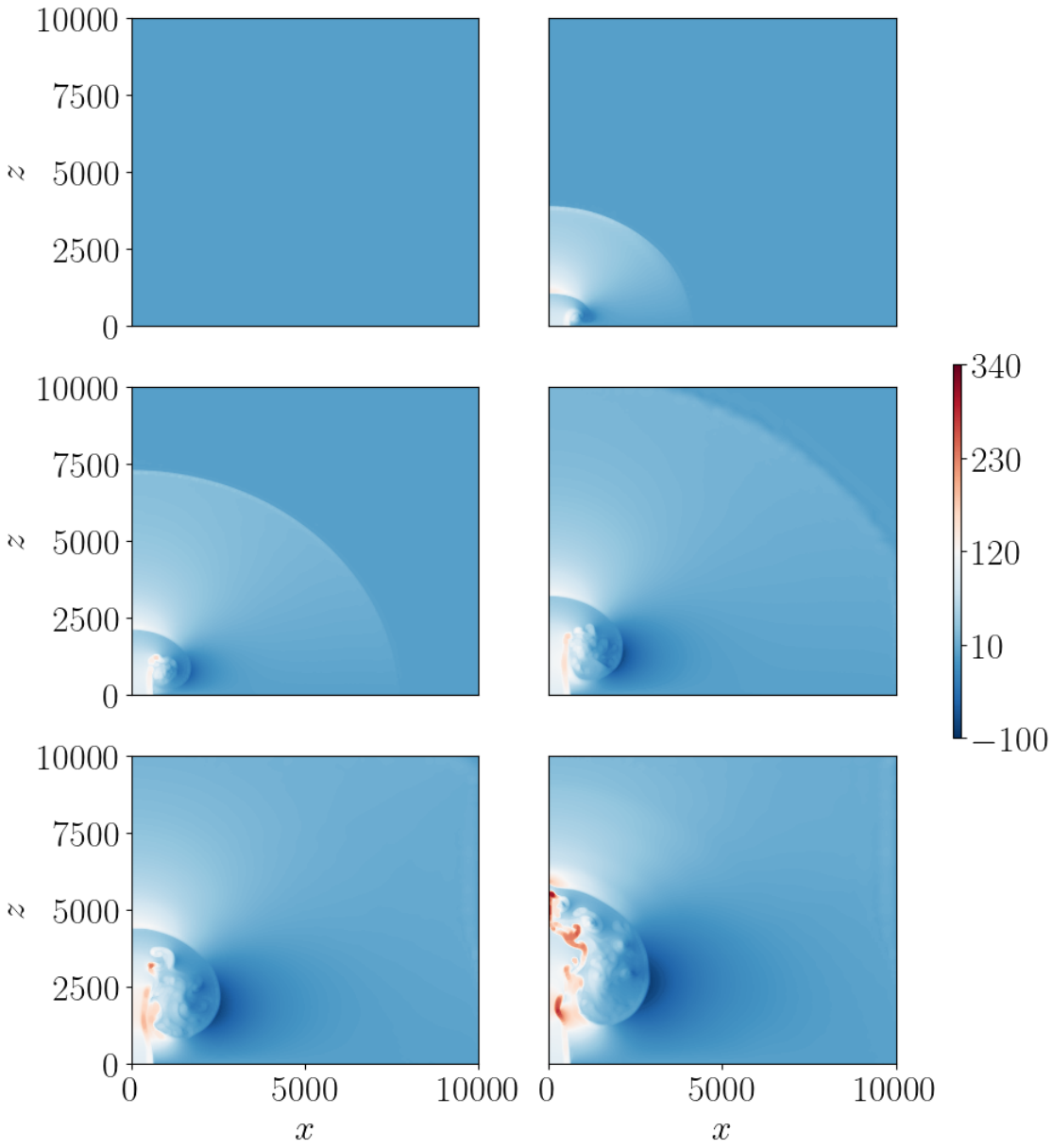

Figure 92: Vertical momentum for the adaptive run of the volcanic jet for Case E. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

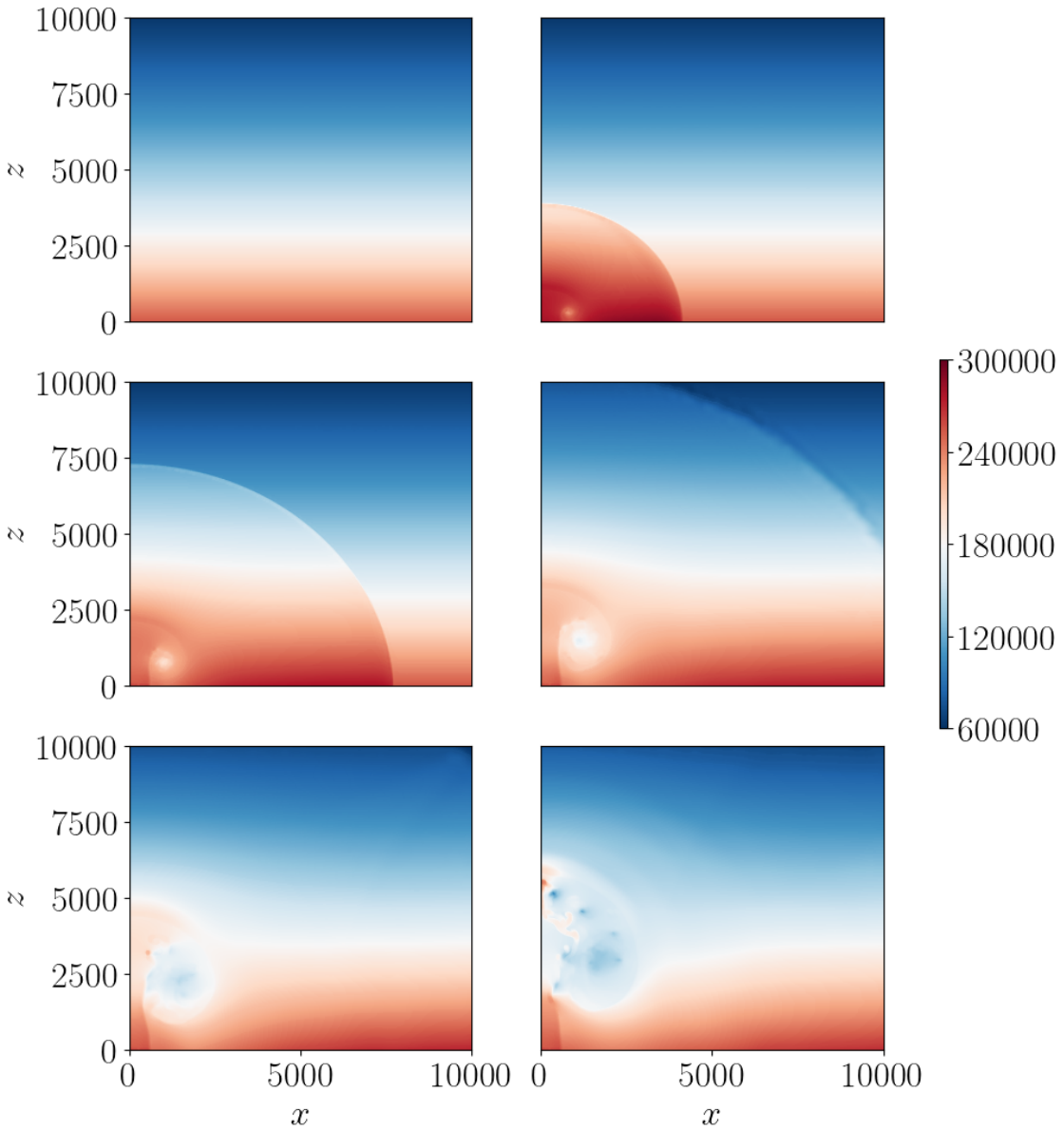

Figure 93: Energy density for the adaptive run of the volcanic jet for Case E. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 6 km while a portion of the jet also spreads out horizontally.

E.6 Case F: Adaptive two-phase (air and water vapor) run with T_m = 1053 K, σ_{ref} = 0.001, no coarsening, e = $\max|\nabla(\rho \mathbf{u})|$ and 20 additional layers of refinement for Limit 1

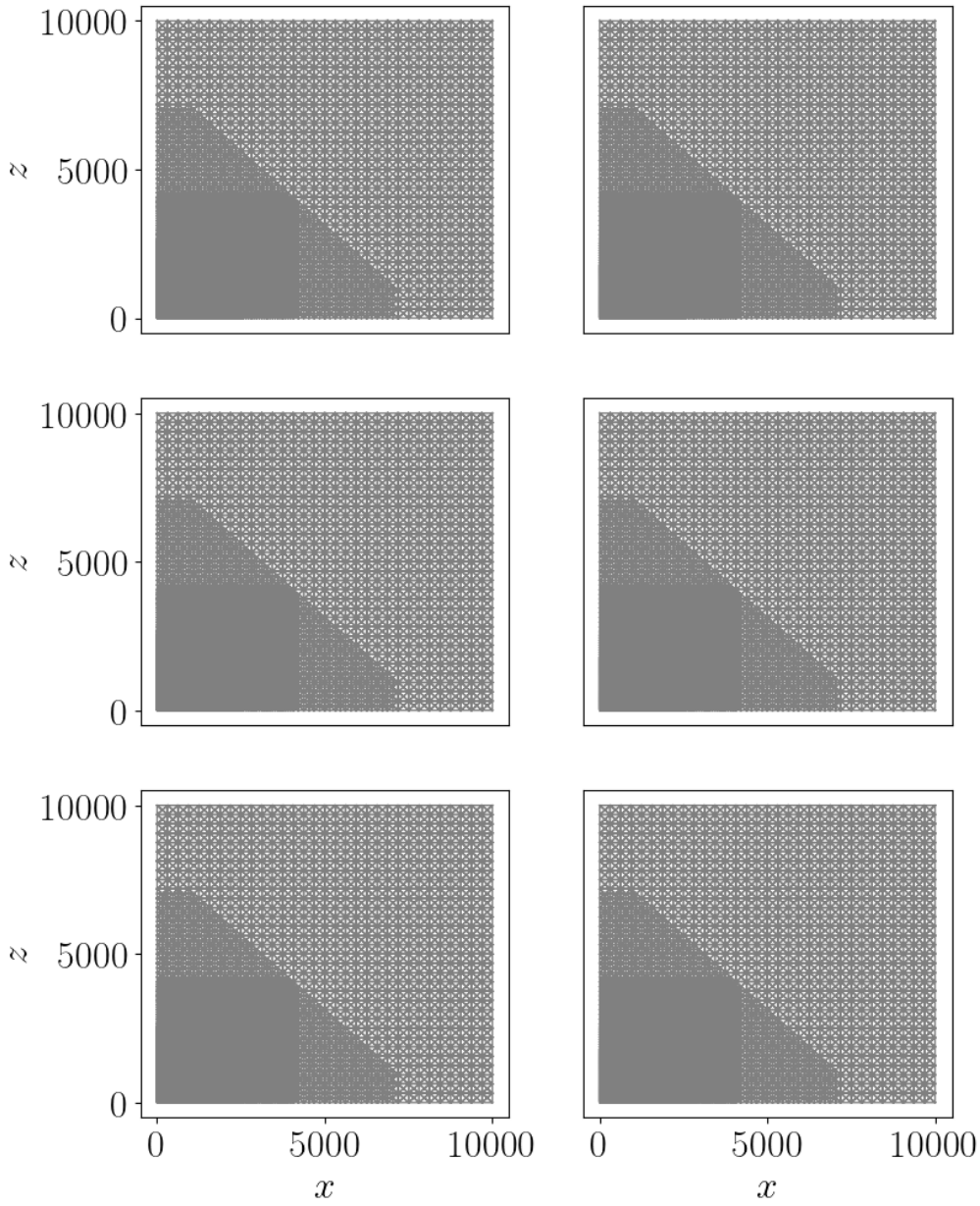

Figure 94: Grid for the adaptive run of the volcanic jet for Case F. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The plots show the development of the adaptive mesh for the respective times. For this setup, the maximum gradient of the momenta was used as error indicator.

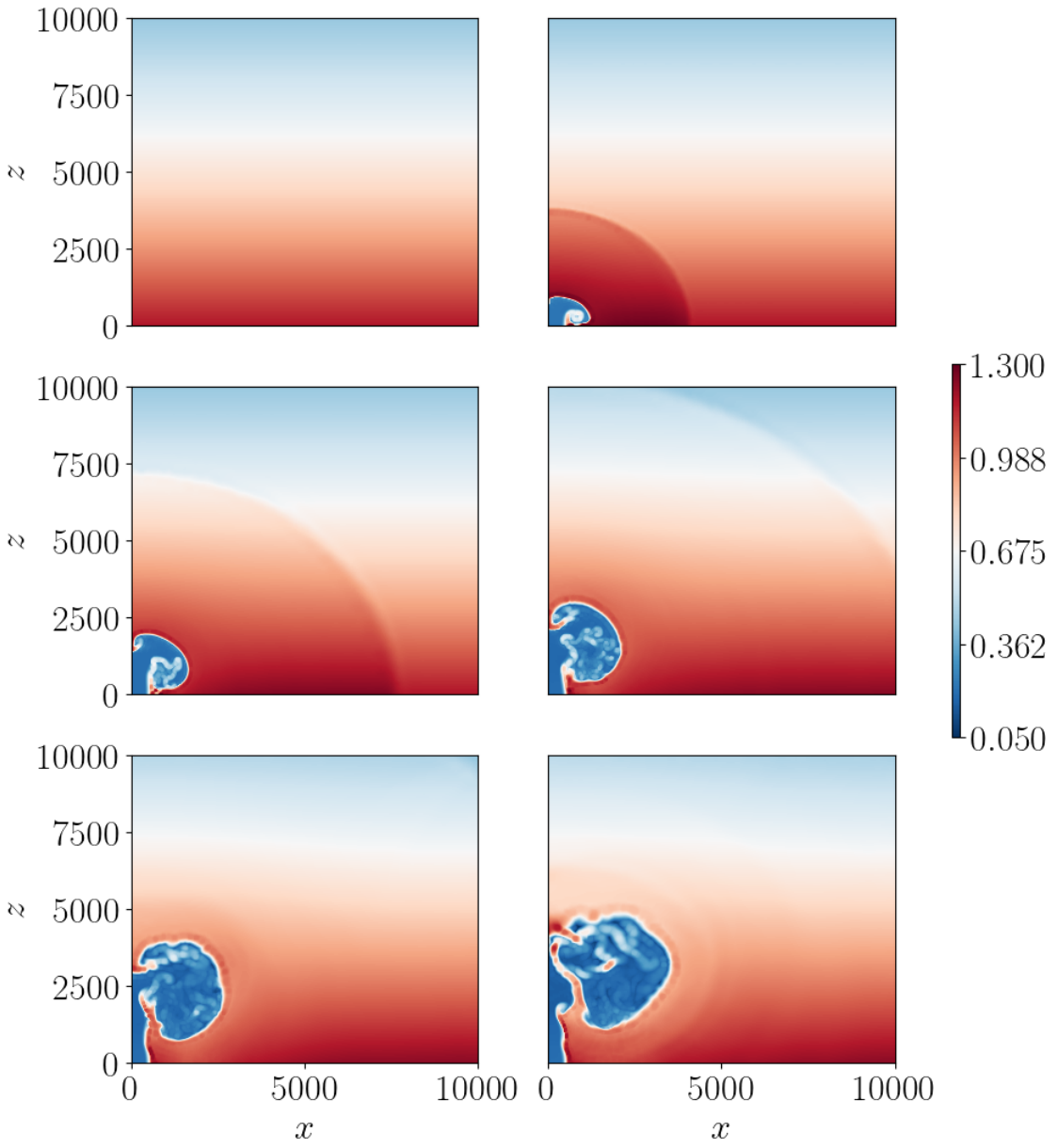

Figure 95: Density for the adaptive run of the volcanic jet for Case F. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

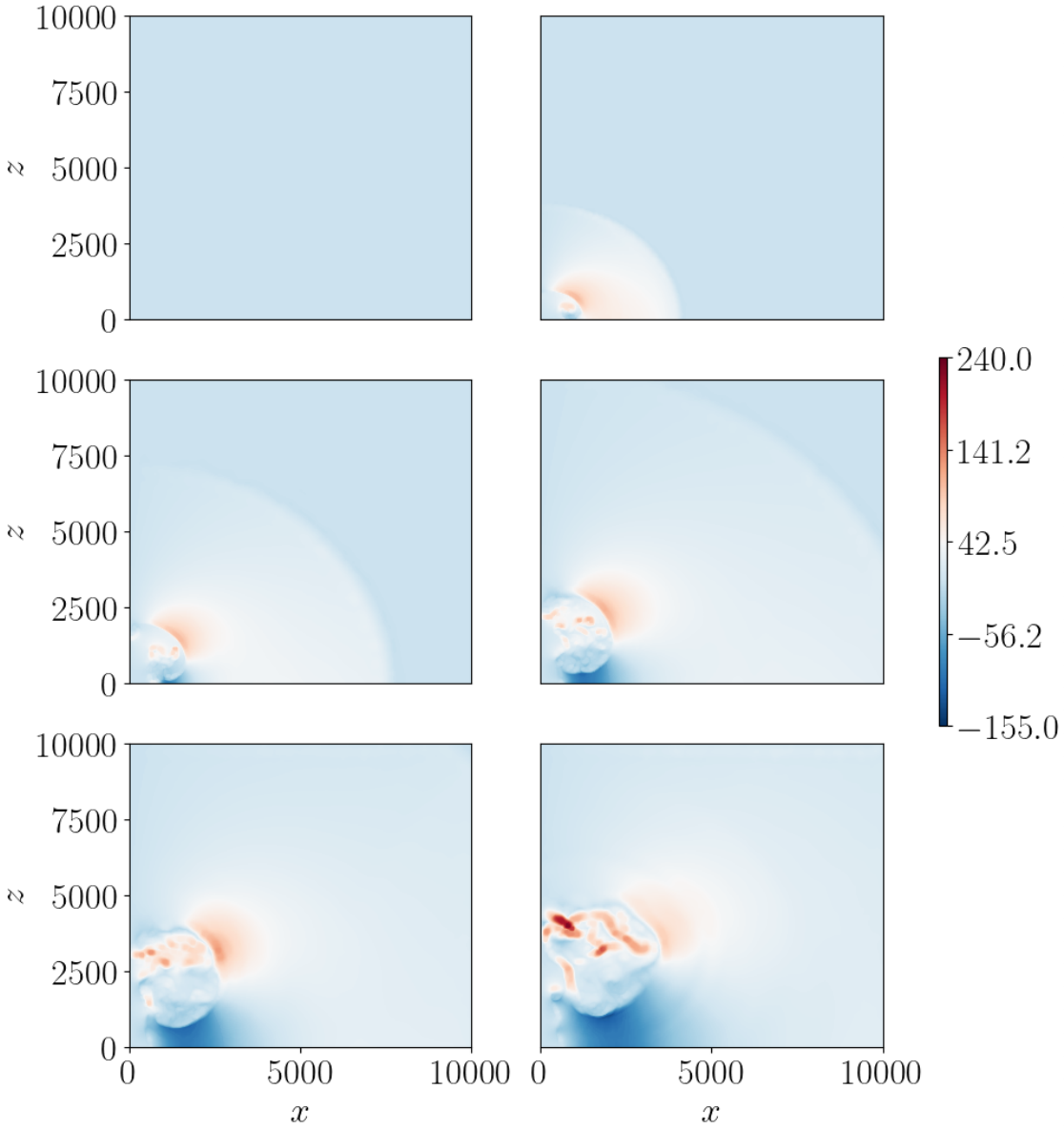

Figure 96: Horizontal momentum for the adaptive run of the volcanic jet for Case F. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

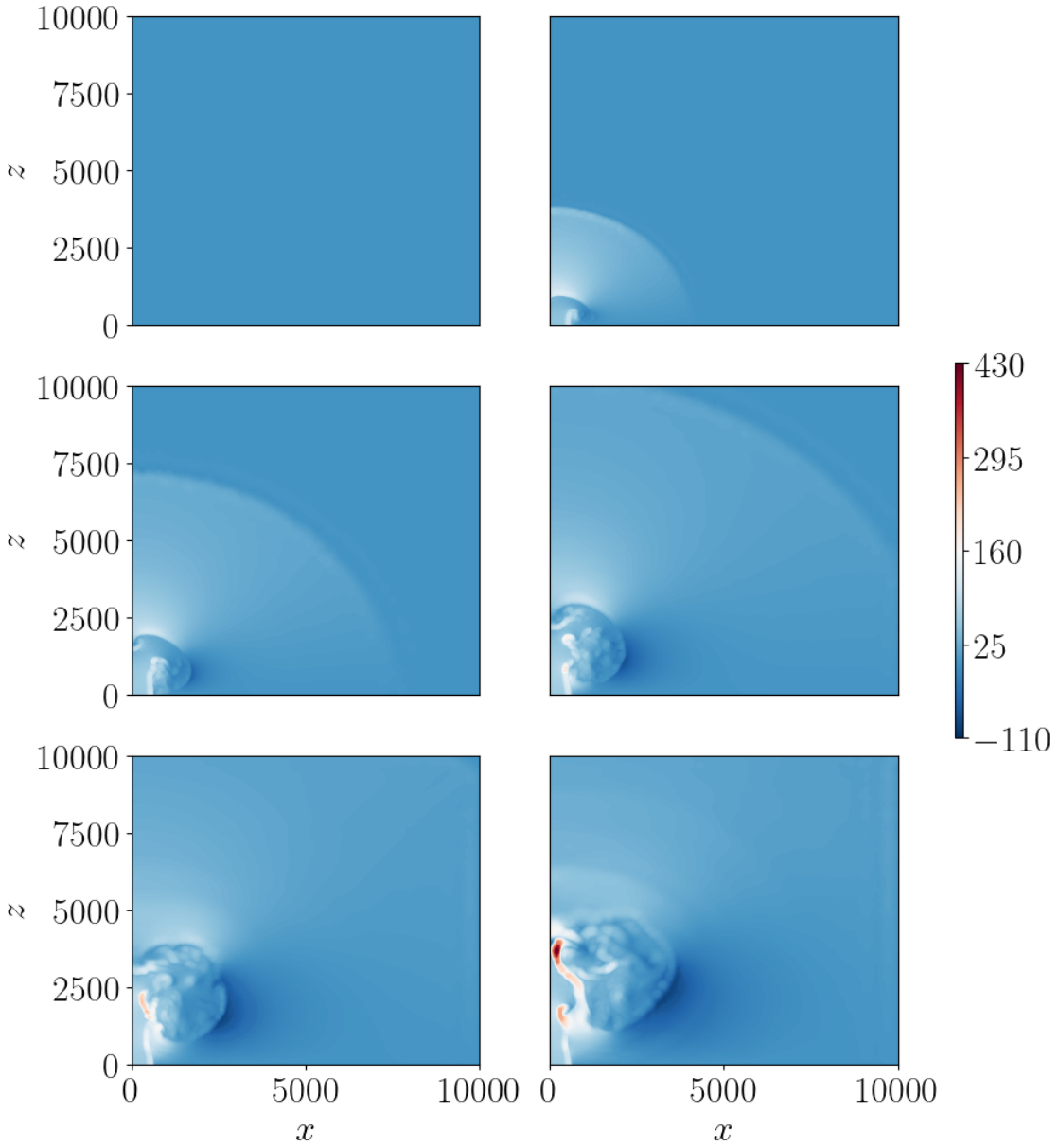

Figure 97: Vertical momentum for the adaptive run of the volcanic jet for Case F. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

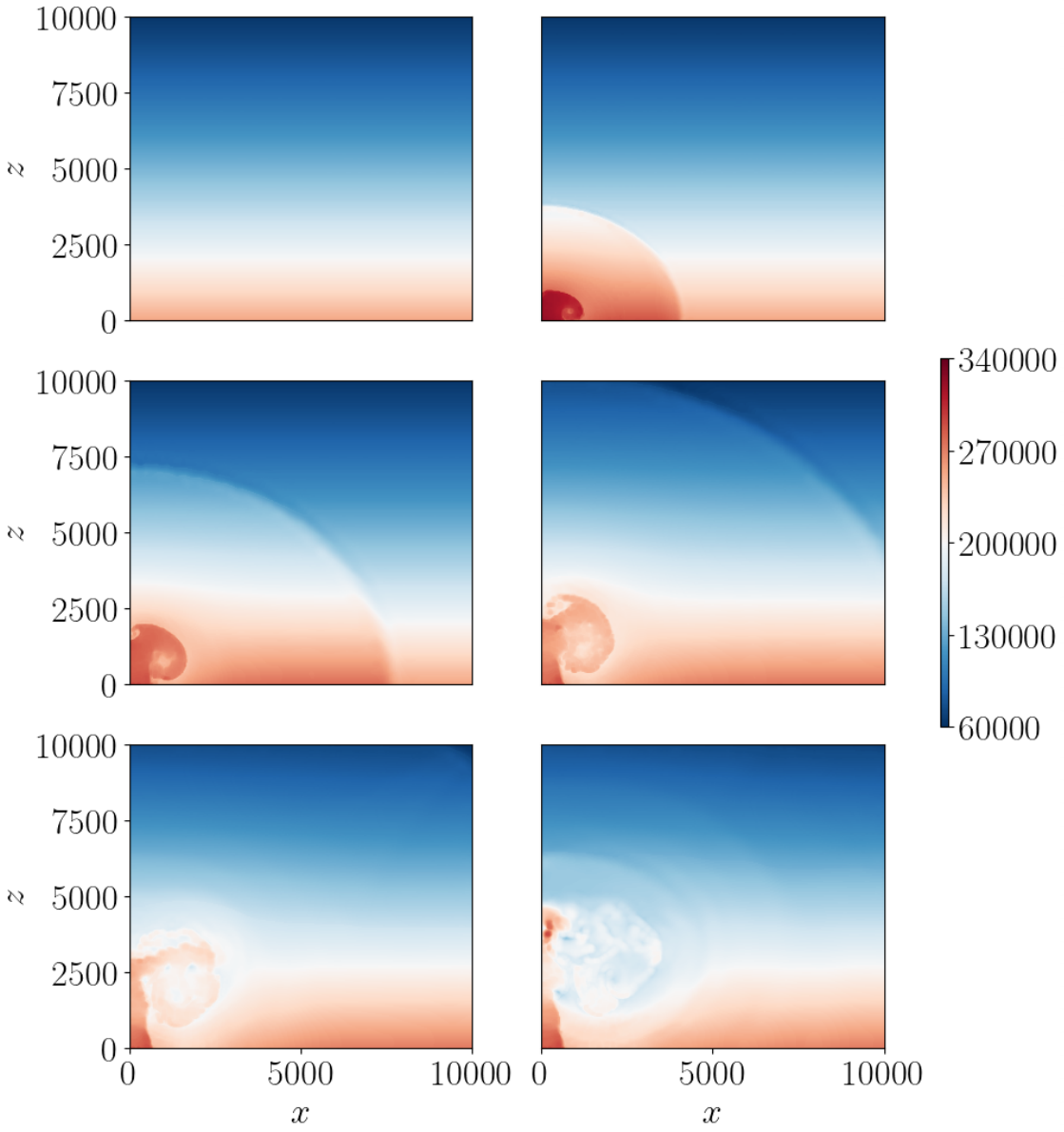

Figure 98: Energy density for the adaptive run of the volcanic jet for Case F. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

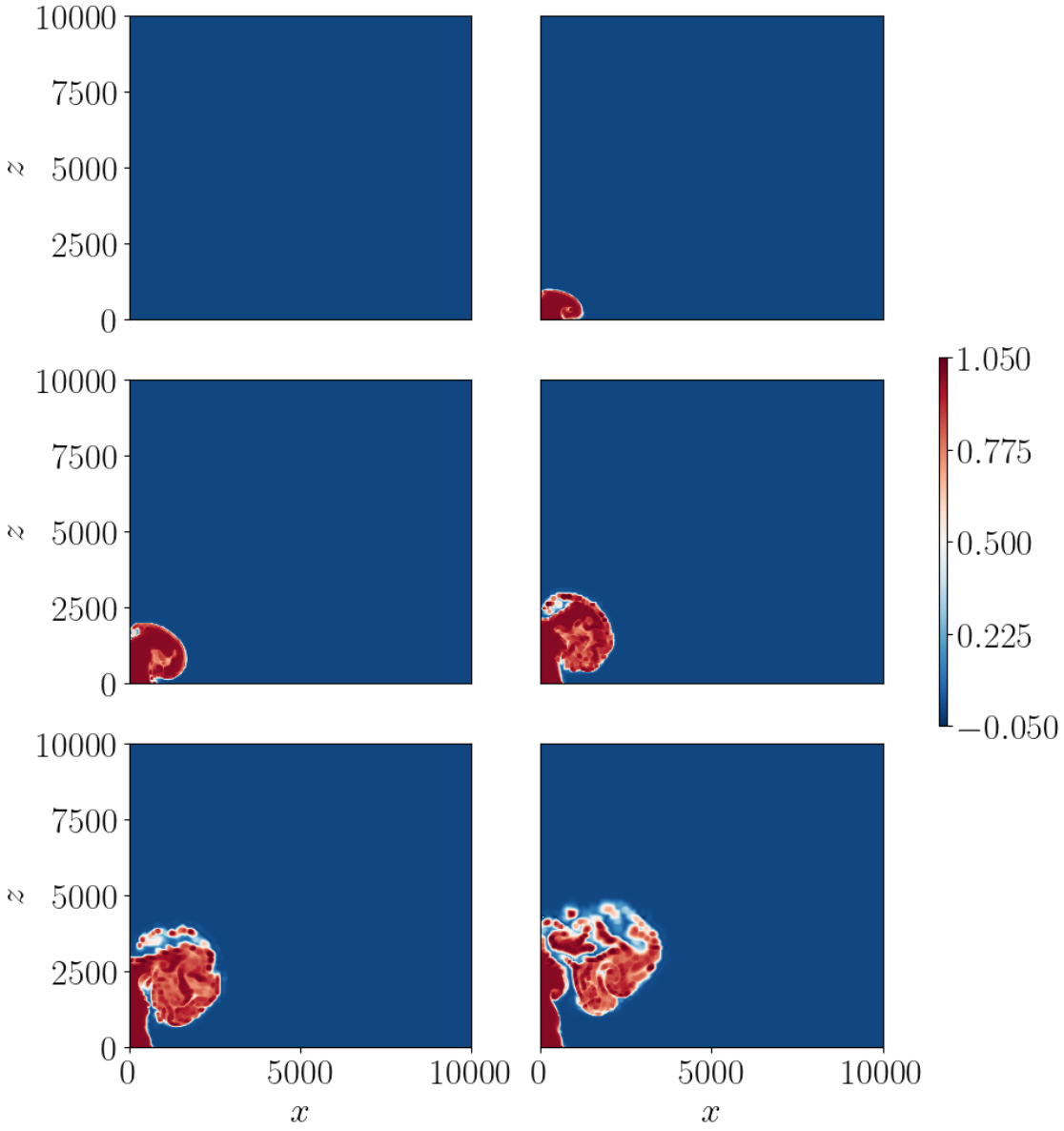

Figure 99: Mass fraction ξ of erupted material for the adaptive run of the volcanic jet for Case F. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally.

E.7 Case G: Adaptive two-phase (air and water vapor) run with T_m = 1053 K, σ_{ref} = 0.75, σ_{coarse} = 0.0001, e = $\nabla(\rho\xi)$, refinement where $\rho\xi > 0.2$ and 8 additional layers of refinement for Limit 2

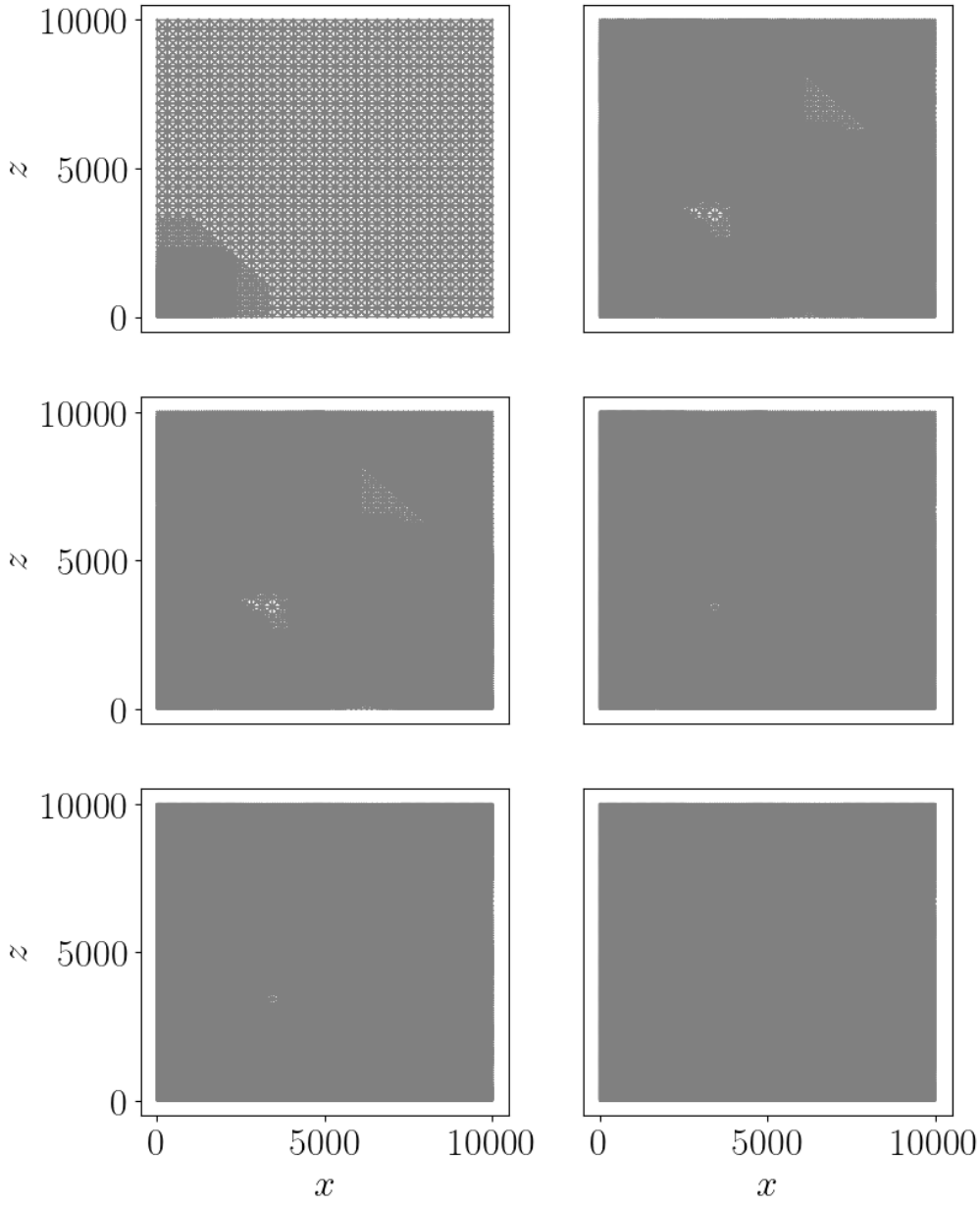

Figure 100: Grid for the adaptive run of the volcanic jet for Case G. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The plots show the development of the adaptive mesh for the respective times. For this setup, the gradient of the mass fraction density was used as error indicator.

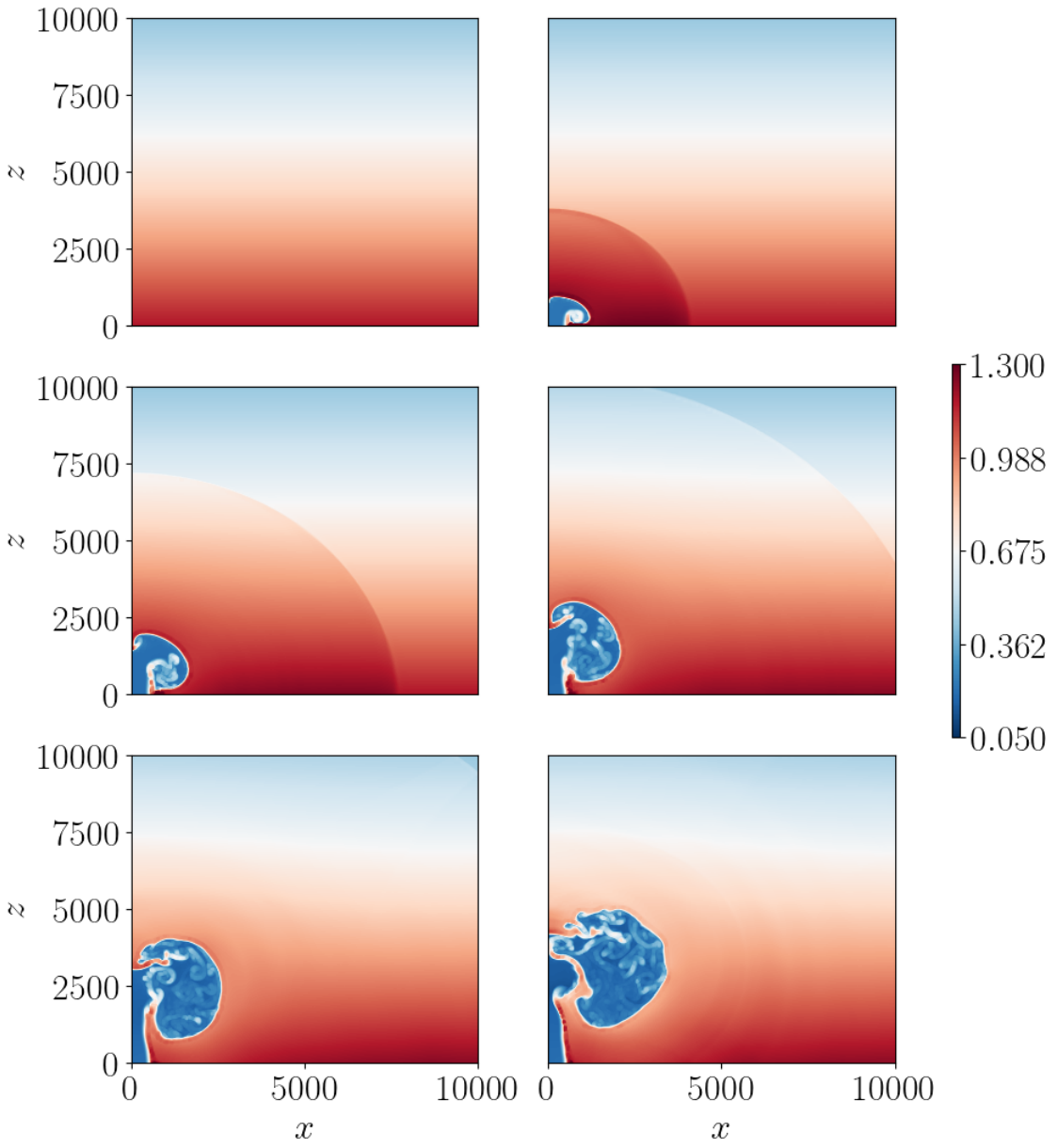

Figure 101: Density for the adaptive run of the volcanic jet for Case G. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

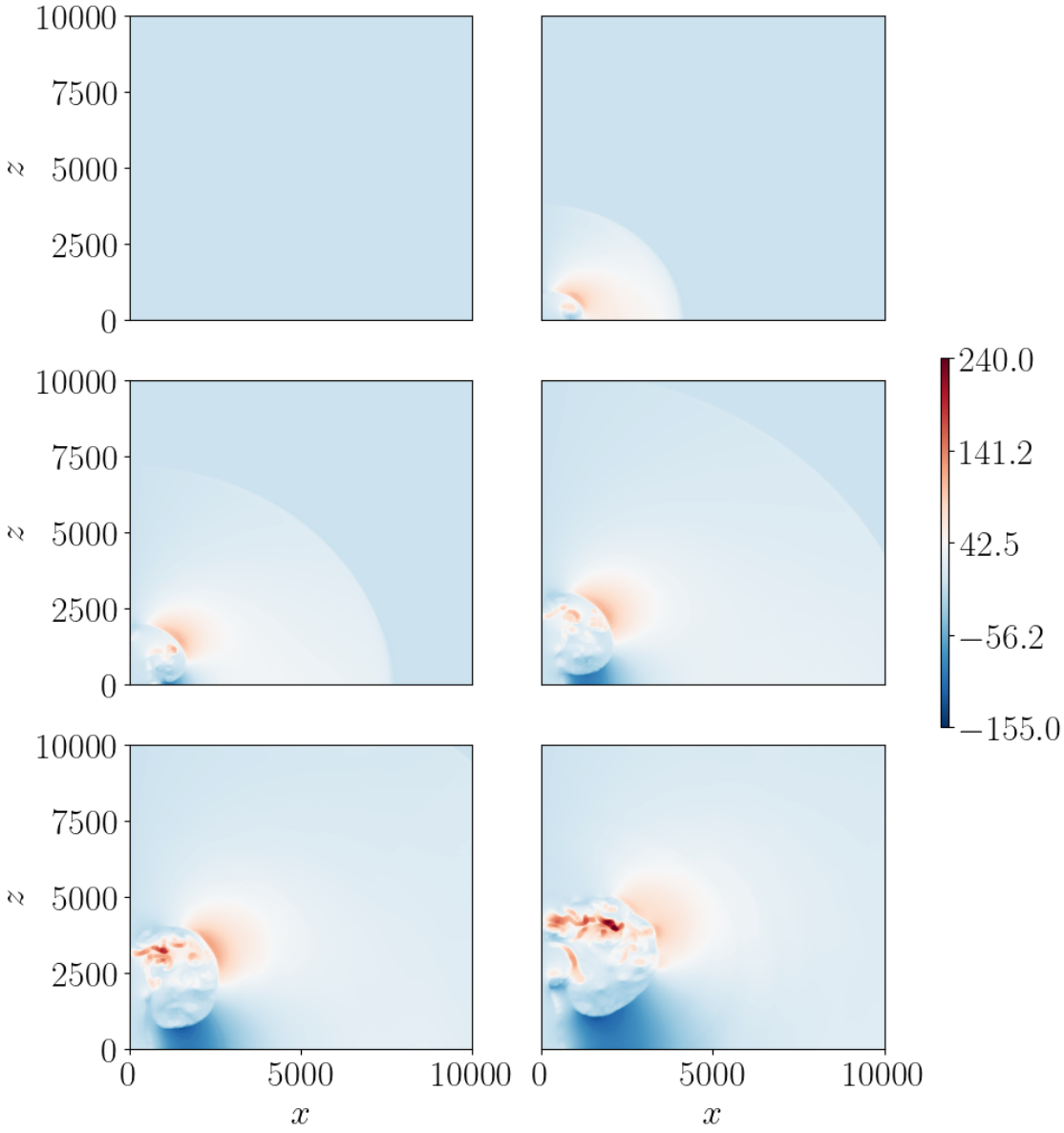

Figure 102: Horizontal momentum for the adaptive run of the volcanic jet for Case G. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

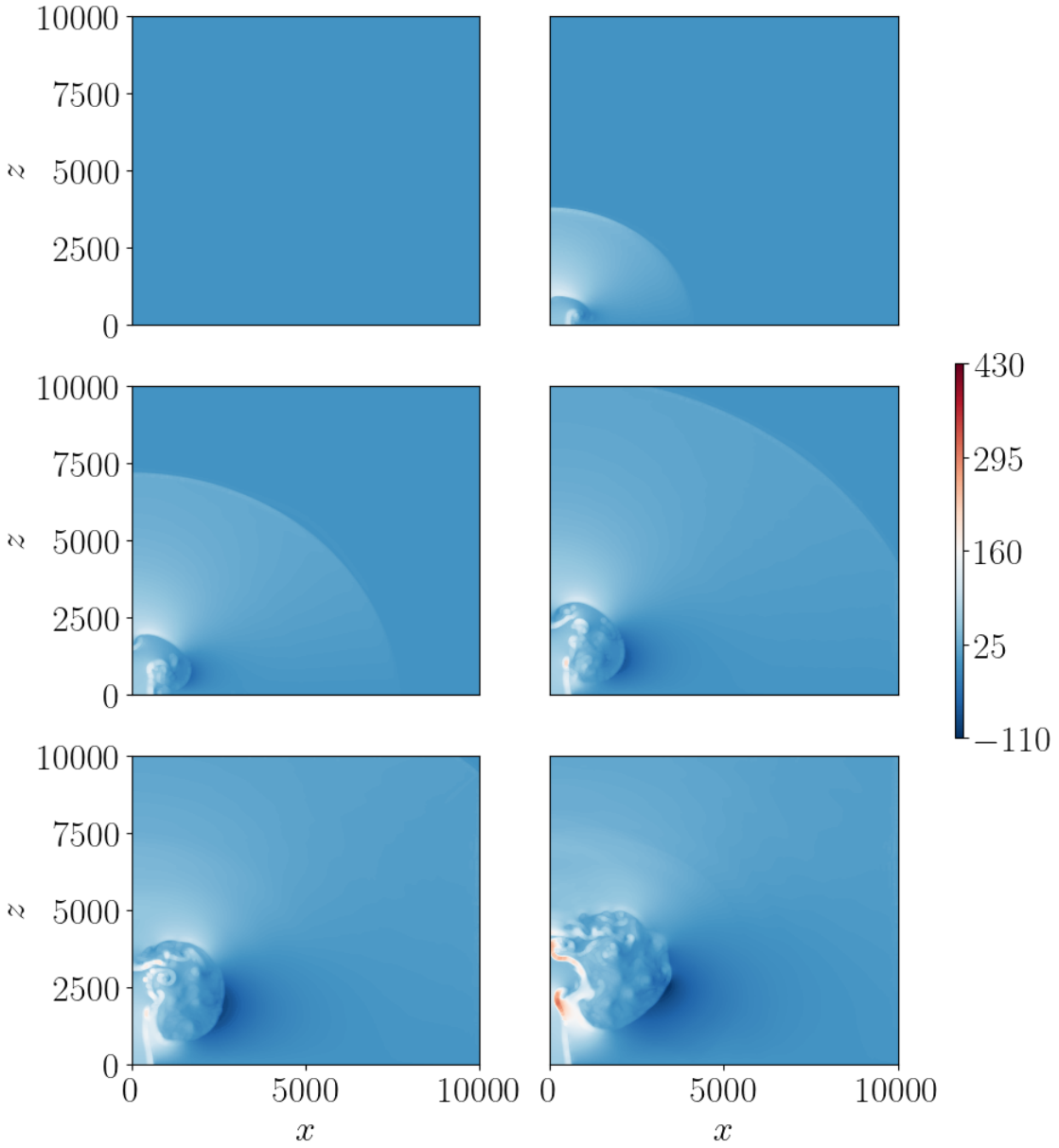

Figure 103: Vertical momentum for the adaptive run of the volcanic jet for Case G. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

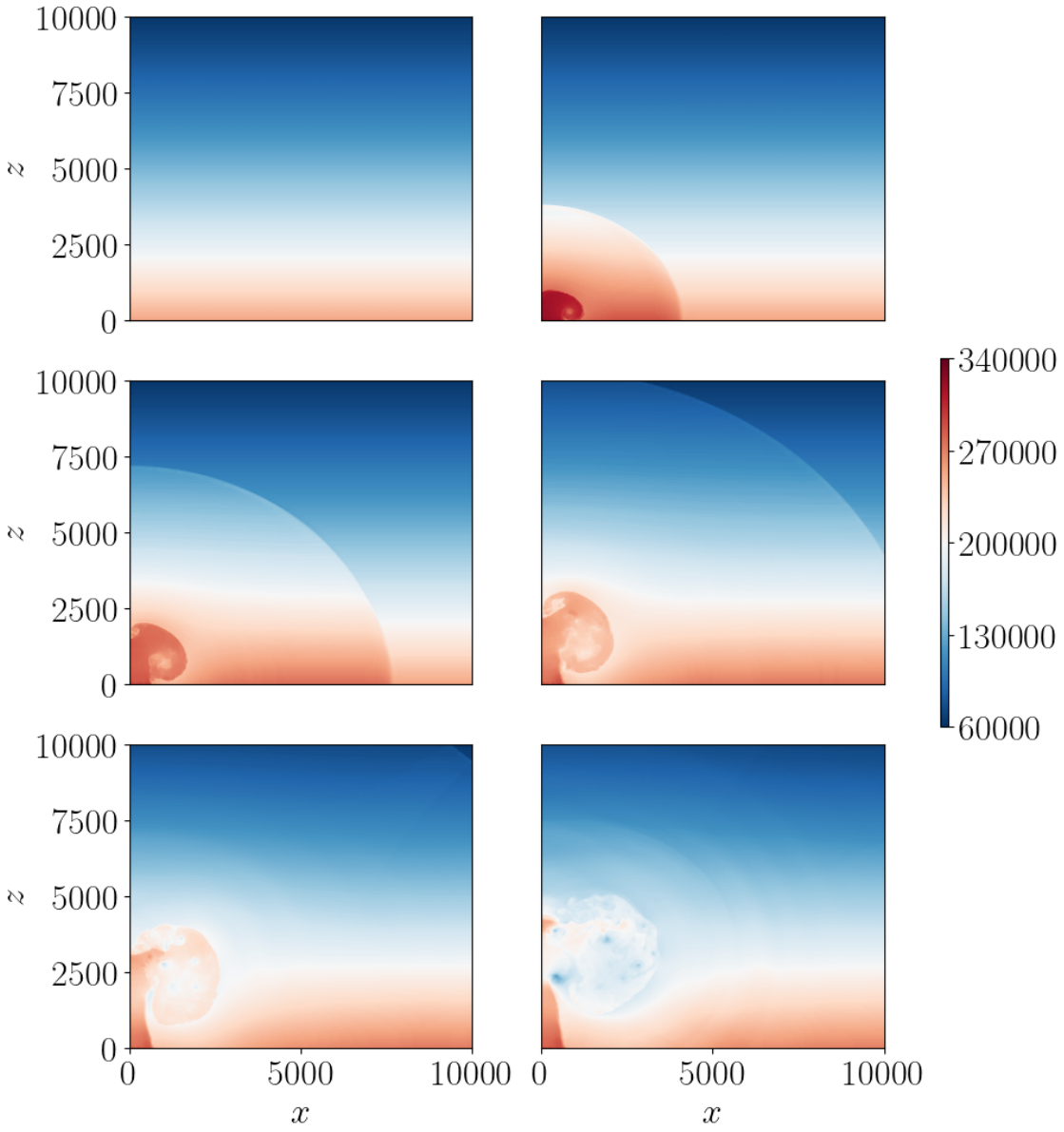

Figure 104: Energy density for the adaptive run of the volcanic jet for Case G. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

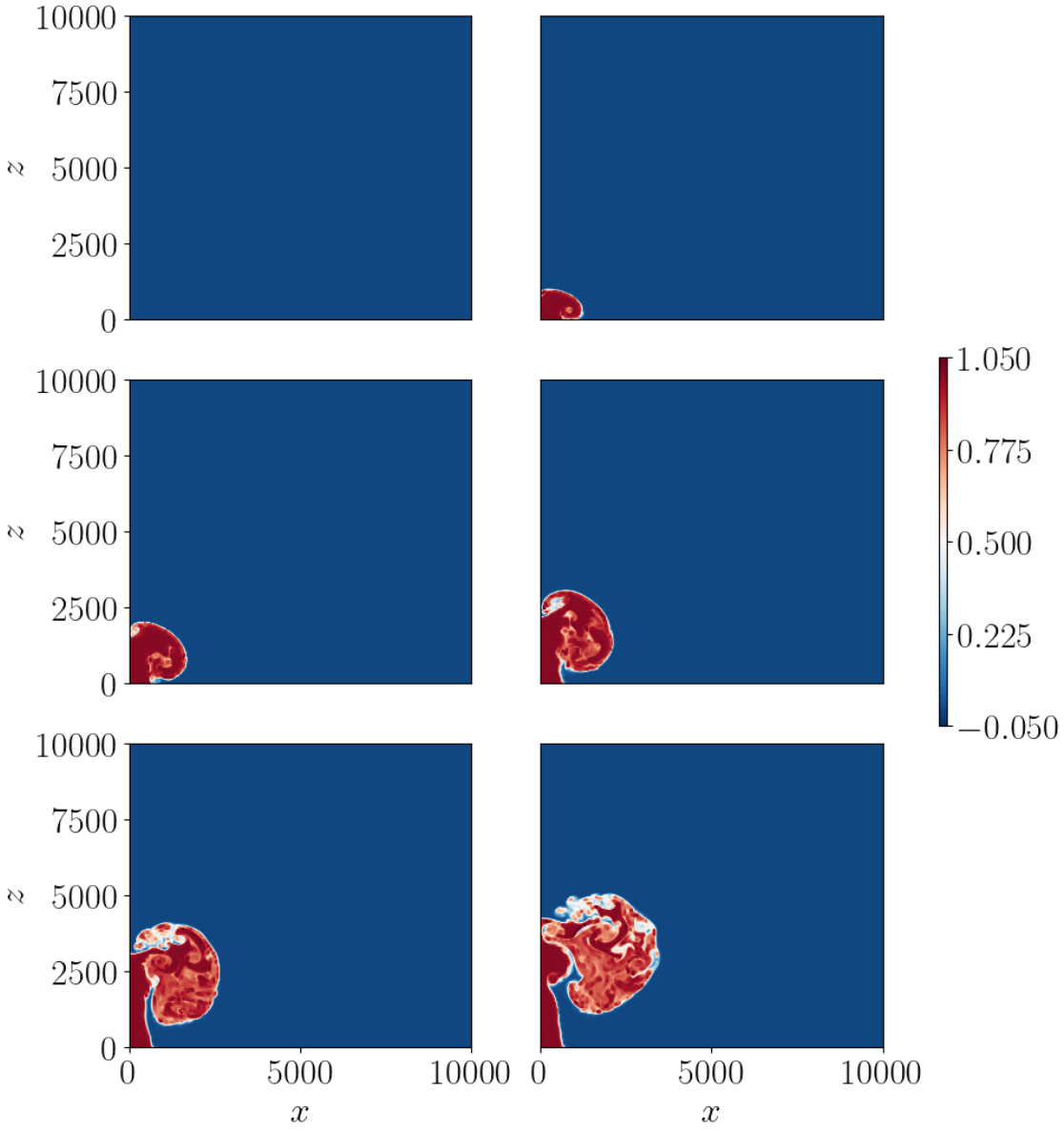

Figure 105: Mass fraction ξ of erupted material for the adaptive run of the volcanic jet for Case G. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally.

E.8 Case H: Adaptive two-phase (air and water vapor) run with T_m = 1053 K, σ_{ref} = 0.1, σ_{coarse} = 0.05, e = $\nabla(\xi)$, refinement where $\rho\xi > 0.2$ and 8 additional layers of refinement for Limit 2

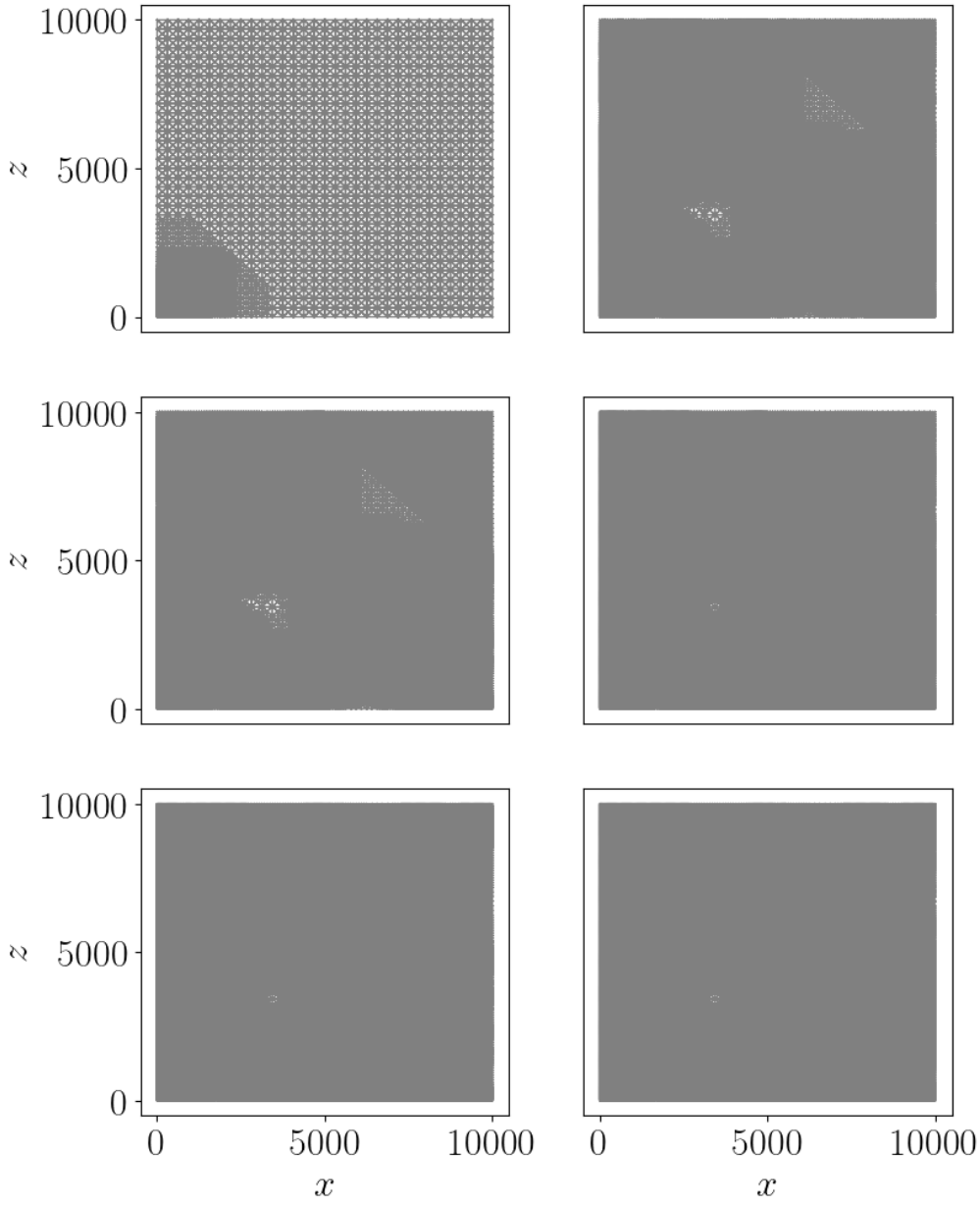

Figure 106: Grid for the adaptive run of the volcanic jet for Case H. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The plots show the development of the adaptive mesh for the respective times. For this setup, the gradient of the mass fraction was used as error indicator.

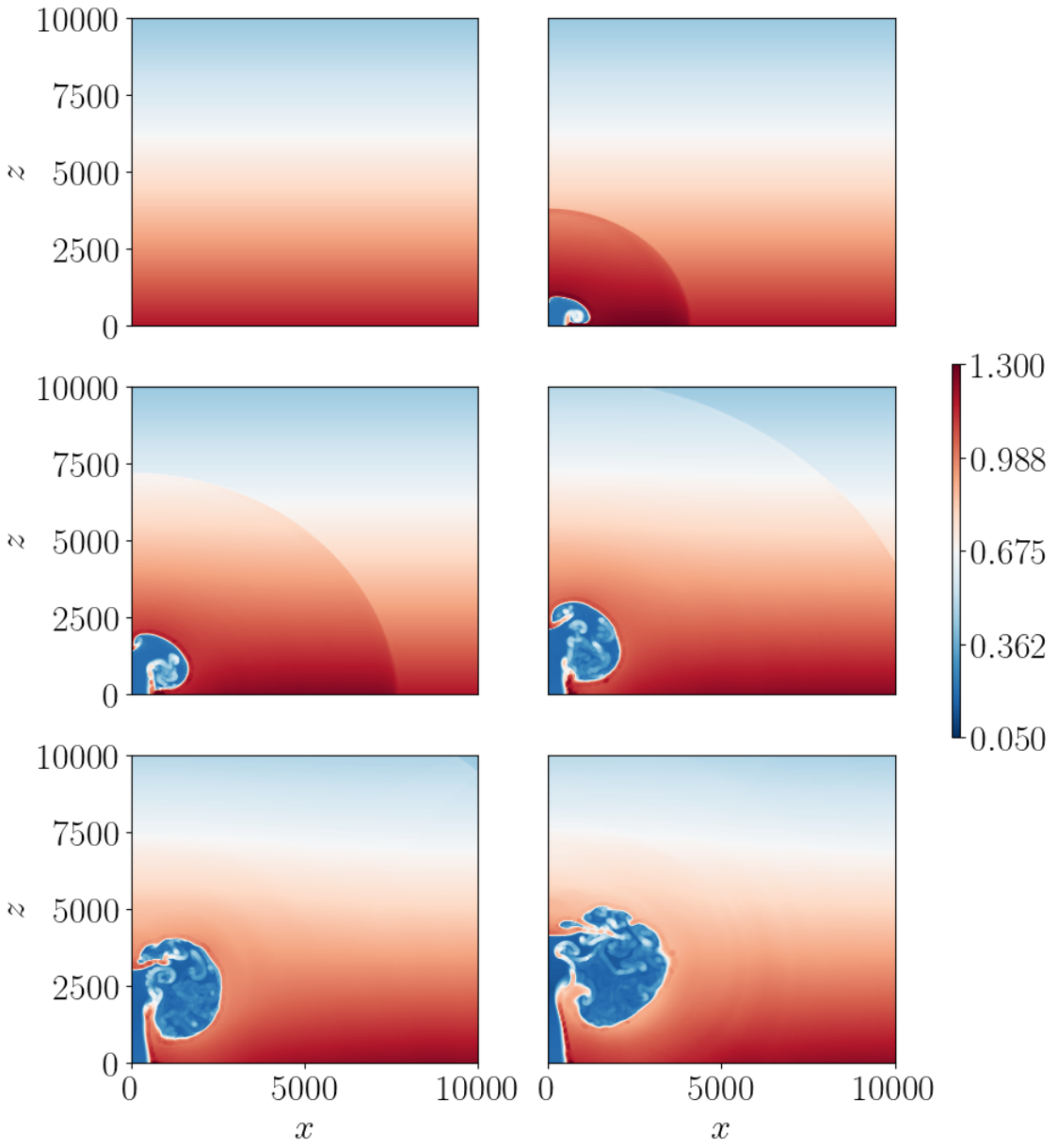

Figure 107: Density for the adaptive run of the volcanic jet for Case H. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

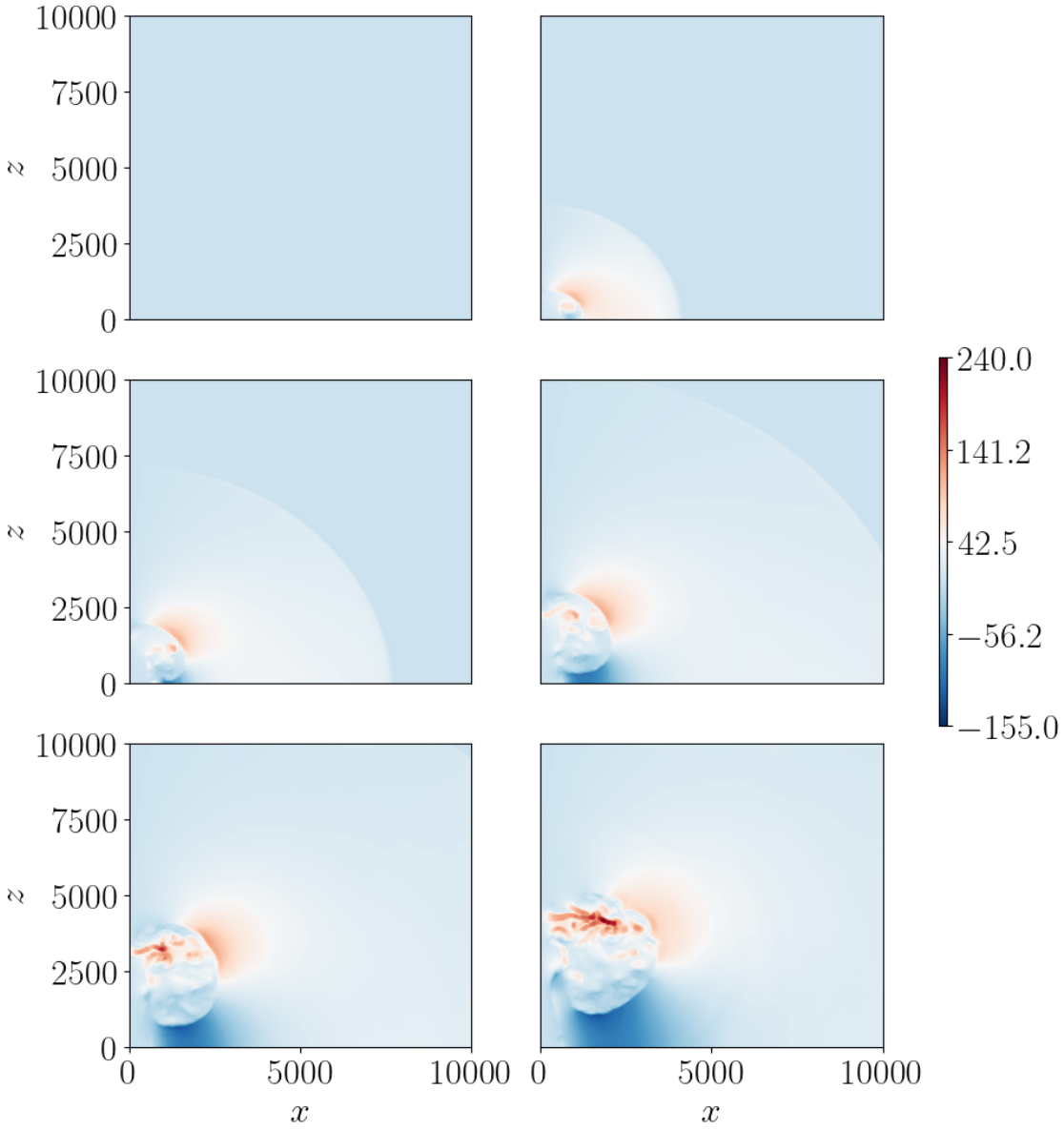

Figure 108: Horizontal momentum for the adaptive run of the volcanic jet for Case H. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

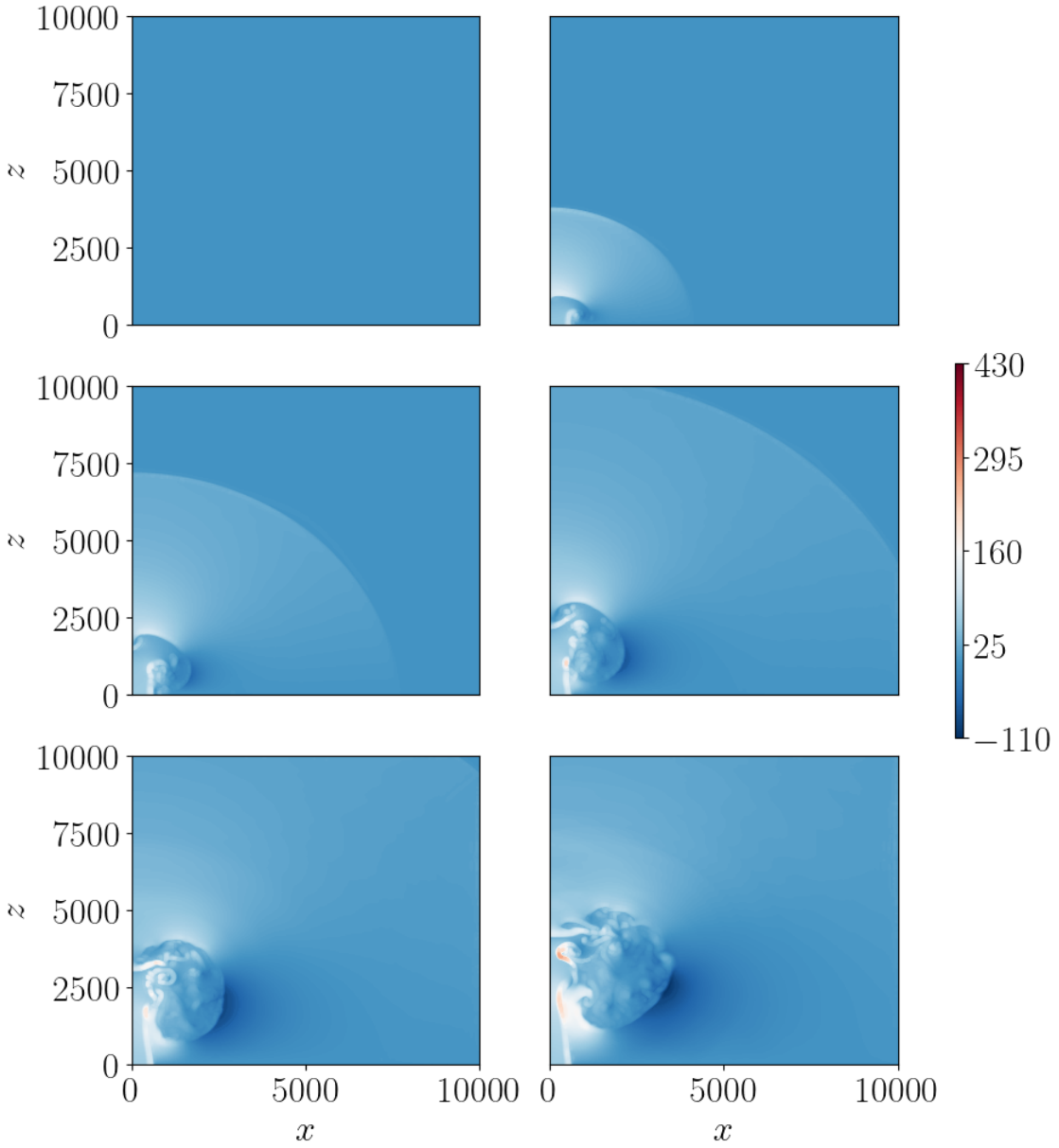

Figure 109: Vertical momentum for the adaptive run of the volcanic jet for Case H. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

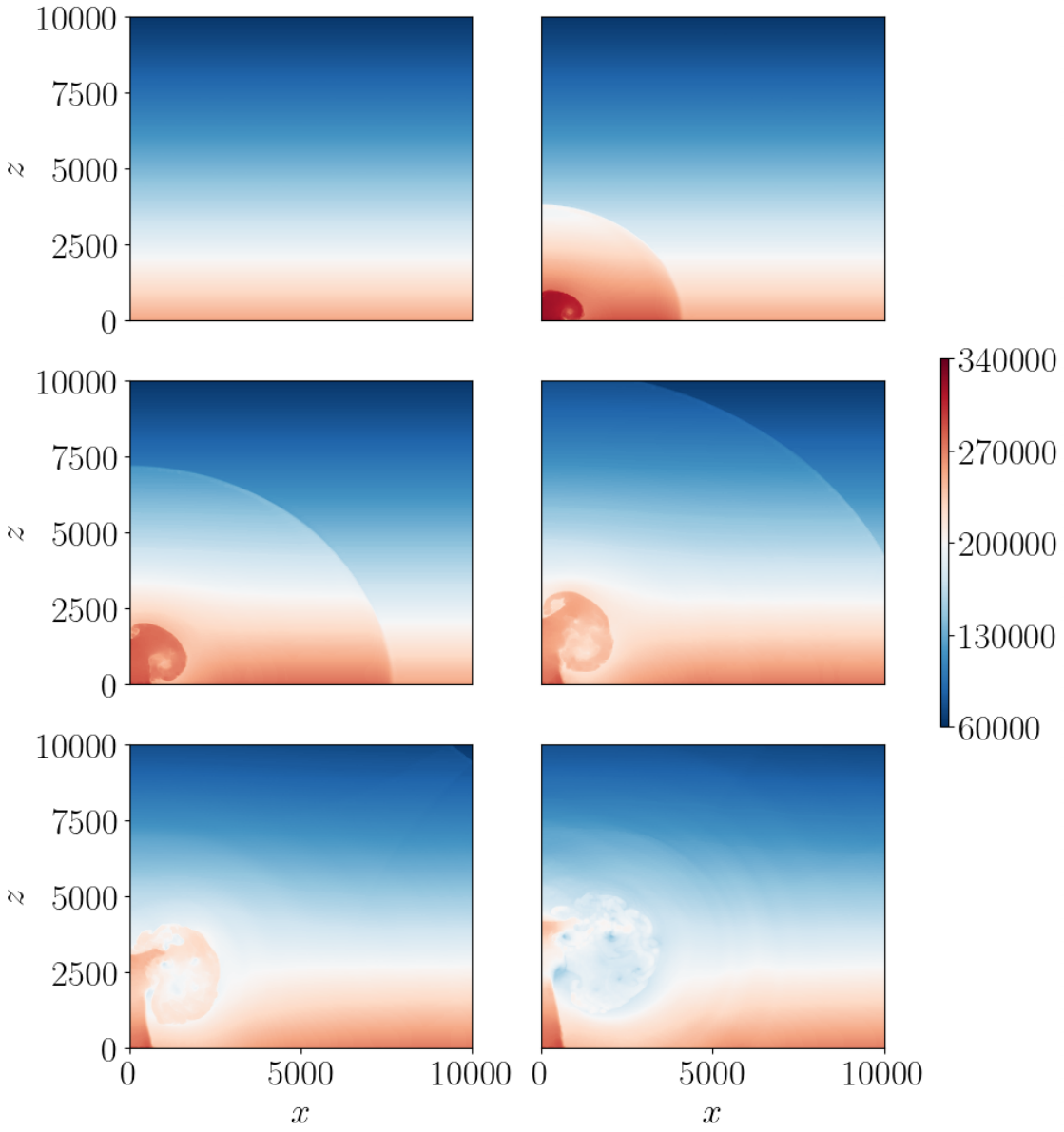

Figure 110: Energy density for the adaptive run of the volcanic jet for Case H. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

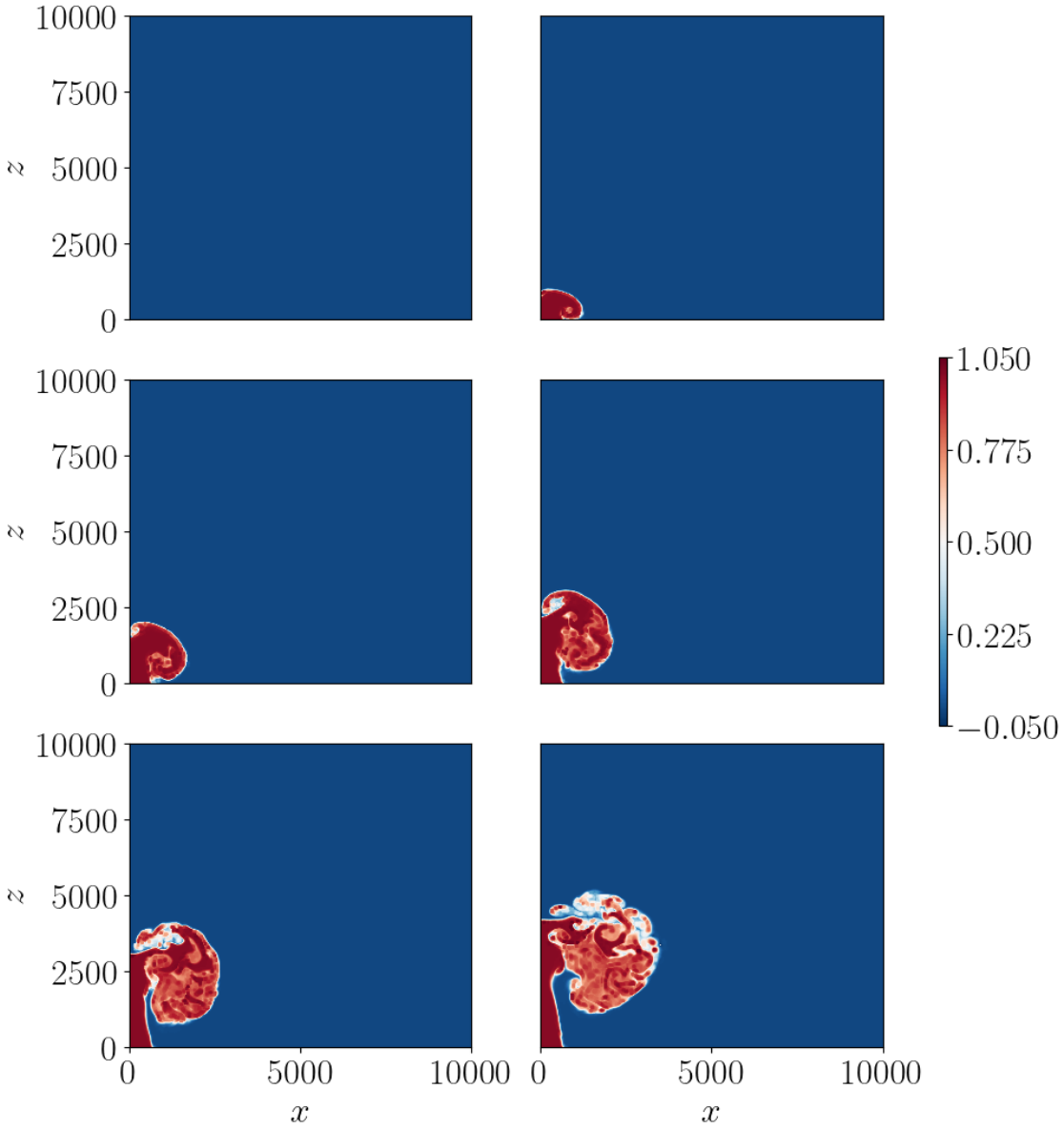

Figure 111: Mass fraction ξ of erupted material for the adaptive run of the volcanic jet for Case H. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally.

E.9 Case I: Adaptive plume run with T_m = 1053 K, refinement if $\xi > 0.8$, coarsening if $\xi < 0.1$ and 15 additional layers of refinement

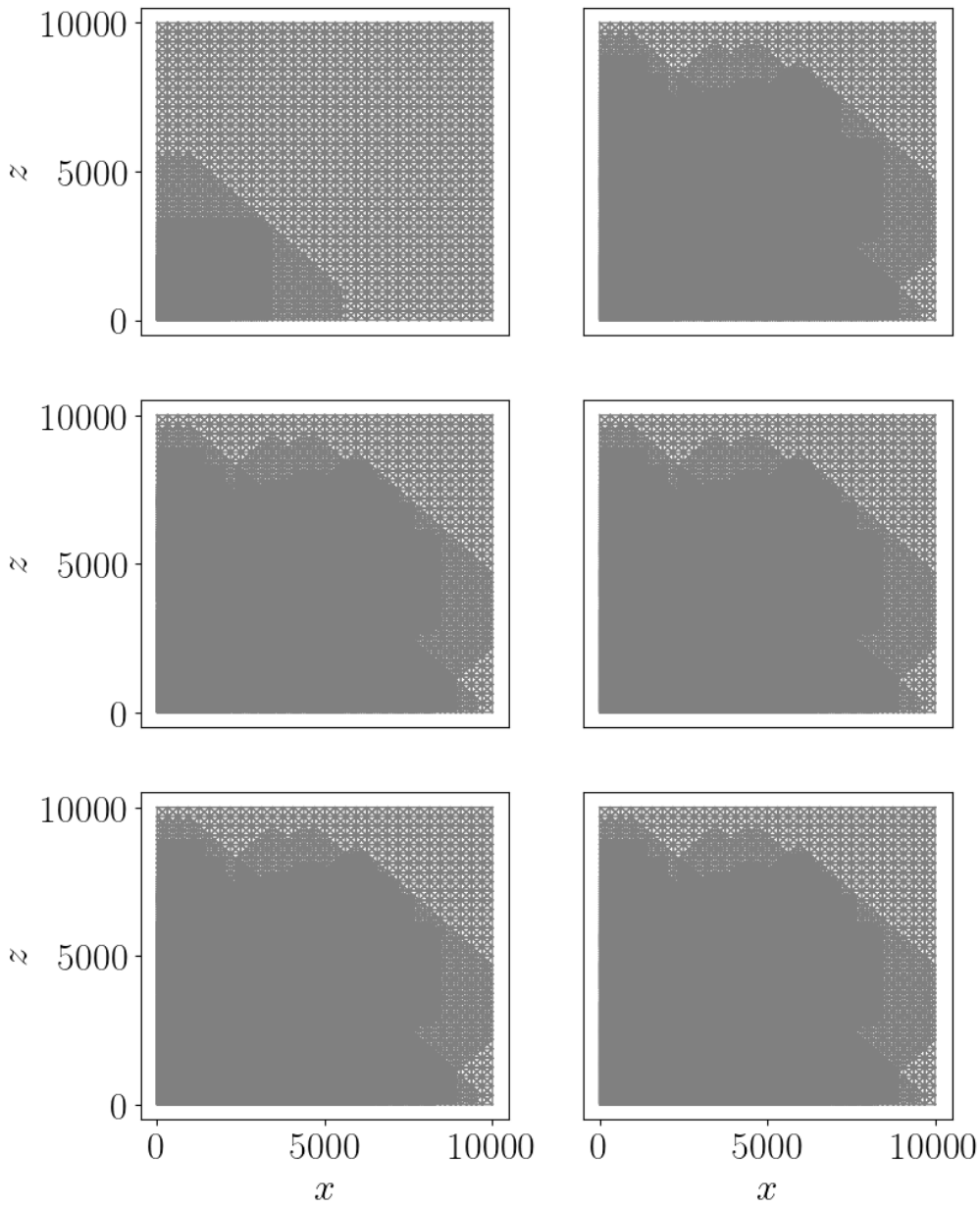

Figure 112: Grid for the adaptive run of the volcanic jet for Case I. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The plots show the development of the adaptive mesh for the respective times. For this setup, no error indicator was used.

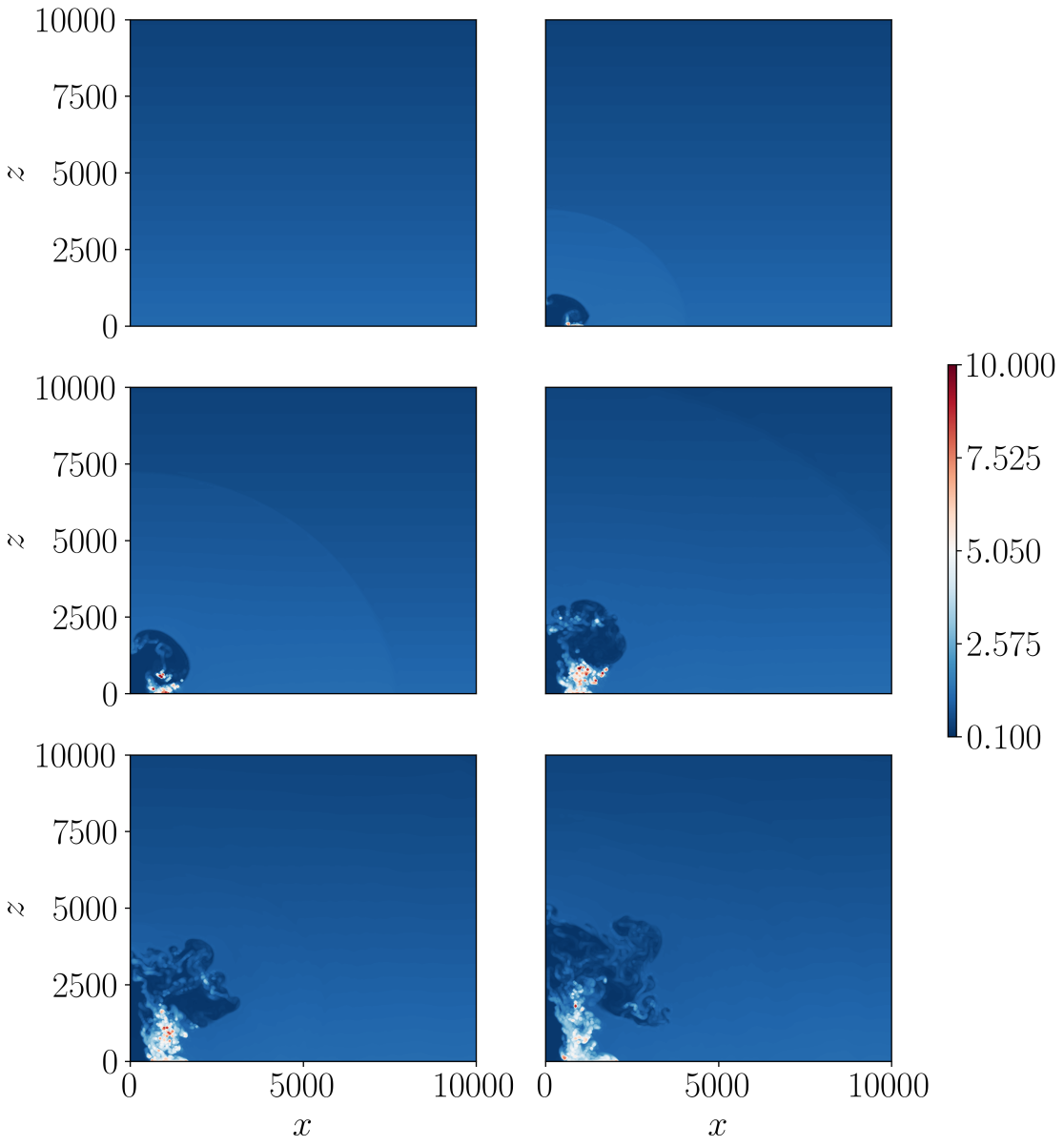

Figure 113: Density for the adaptive run of the volcanic jet for Case I. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet of low density rises upwards from the vent whose highest point reaches a height of roughly 5 km. Additionally, there are also portions of the plume that spread out horizontally and that sink down. After 40 seconds, the development of shock fronts around the jet is observable.

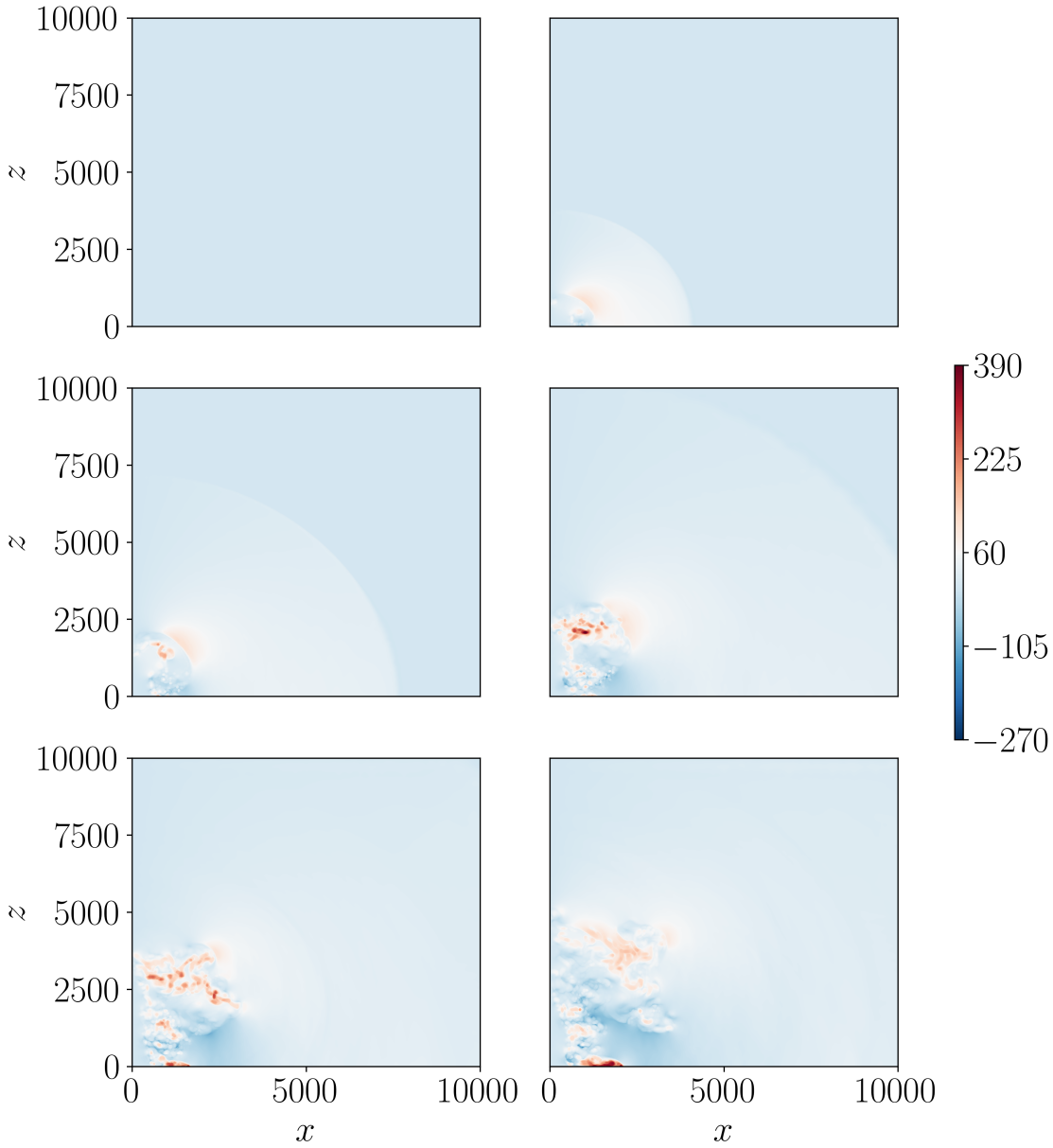

Figure 114: Horizontal momentum for the adaptive run of the volcanic jet for Case I. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

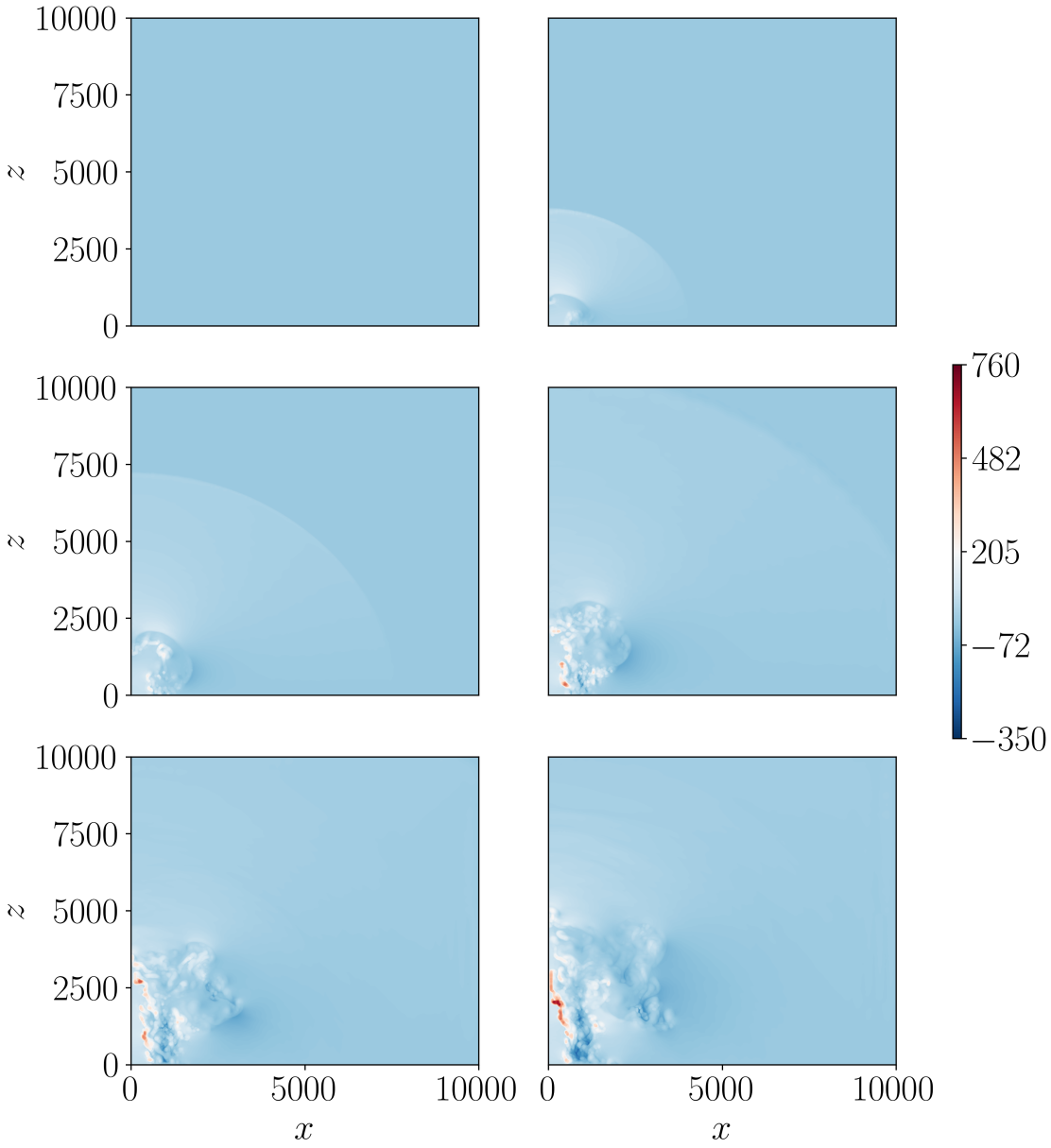

Figure 115: Vertical momentum for the adaptive run of the volcanic jet for Case I. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in kg m^{-2} s^{-1}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

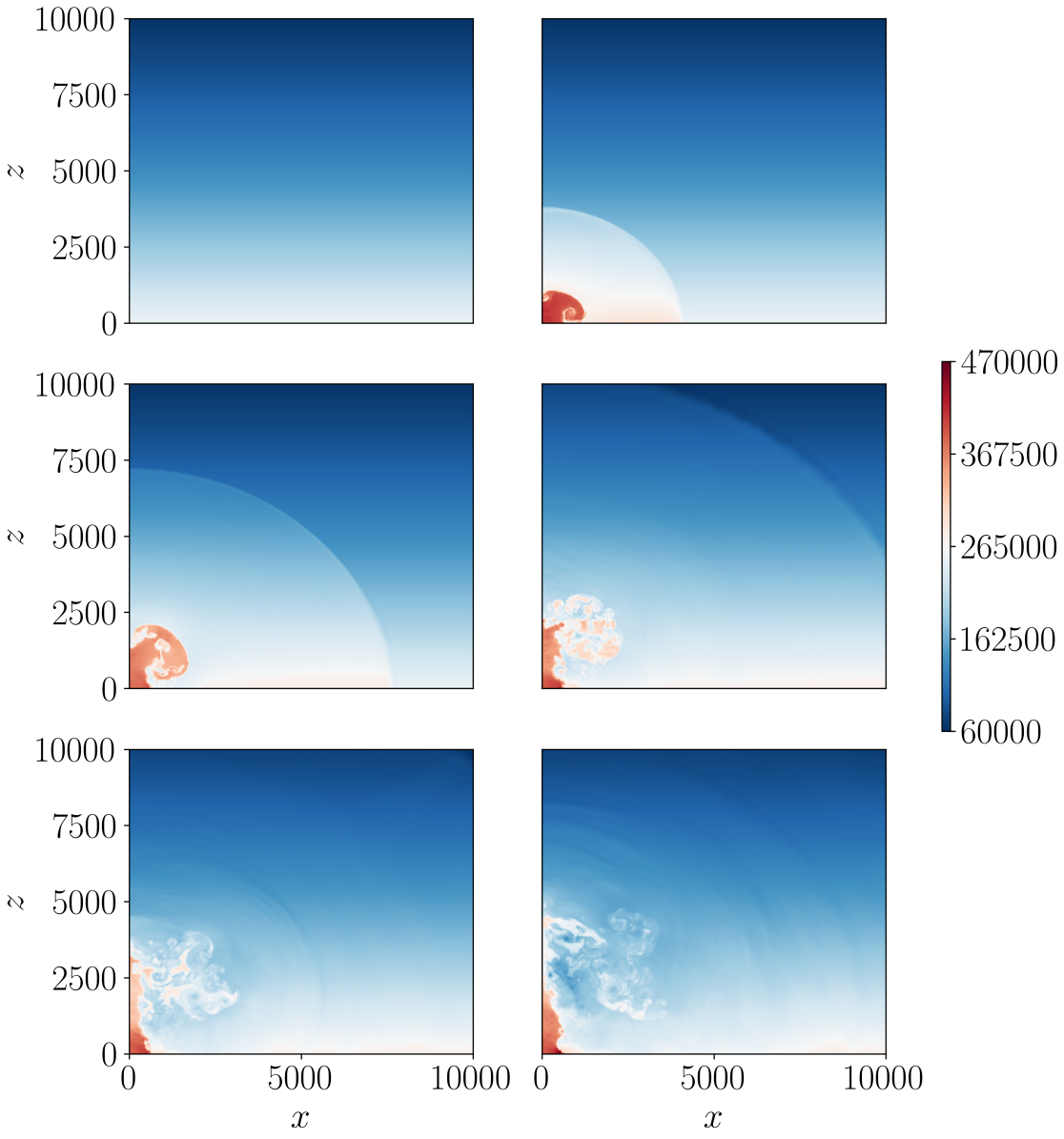

Figure 116: Energy density for the adaptive run of the volcanic jet for Case I. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally. After 40 seconds, the development of shock fronts around the jet is observable.

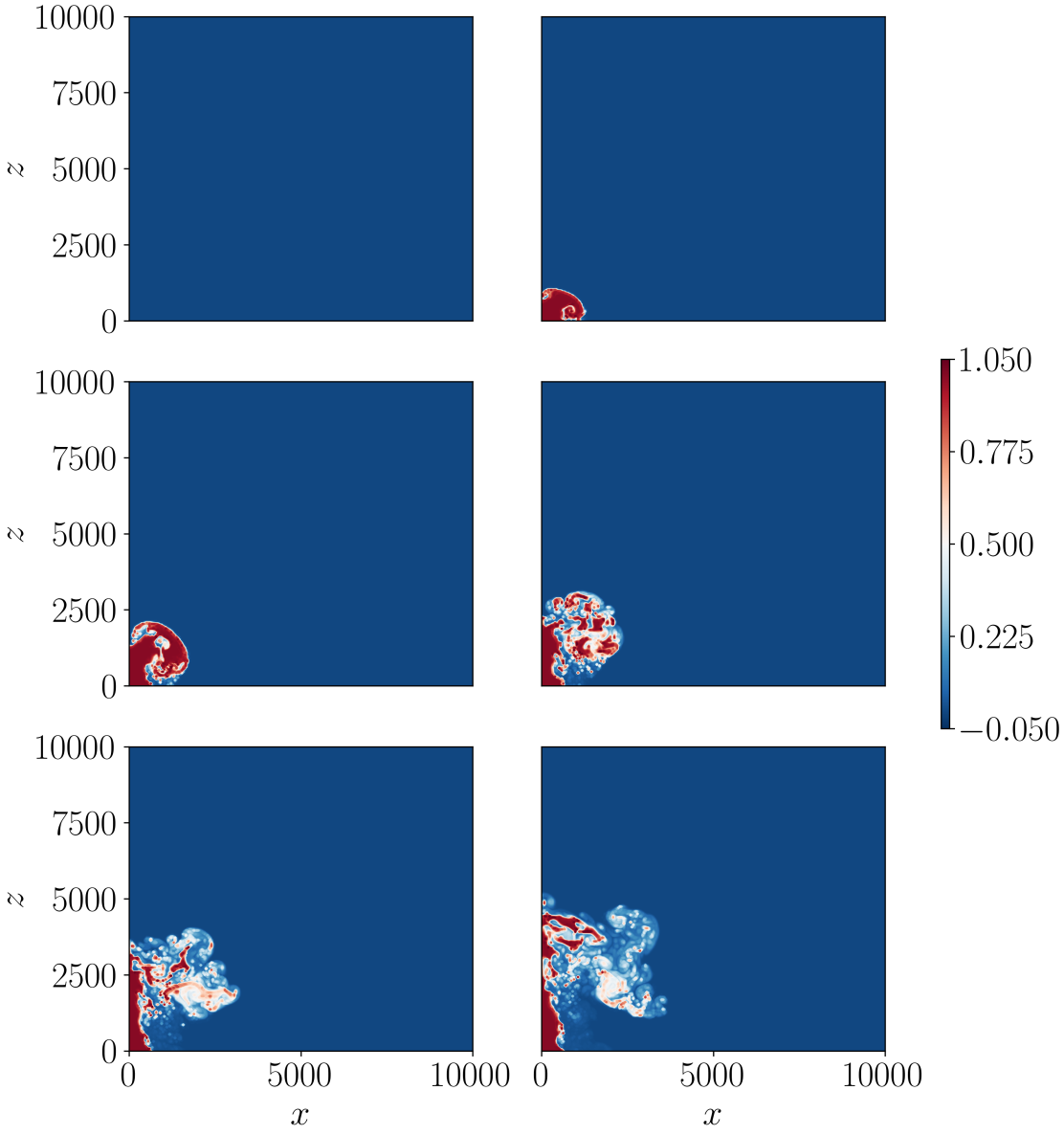

Figure 117: Mass fraction ξ of erupted material for the adaptive run of the volcanic jet for Case I. The plots are shown for 0 (top-left), 10 (top-right), 20 (mid-left), 30 (mid-right), 40 (bottom-left) and 50 (bottom-right) seconds of simulated time. The colorbar shows values in J m^{-3}. The plots show the development of an acoustic wave that travels through and leaves the domain after 30 seconds. The jet rises upwards from the vent whose highest point reaches a height of roughly 5 km while a portion of the jet also spreads out horizontally.

Hiermit versichere ich an Eides statt, dass ich die vorliegende Promotion im Fachbereich Mathematik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe.

Michel Bänsch

Ort, Datum