České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská
Katedra fyzikální elektroniky

Diplomová práce

Ivan Tarant

Praha – 2022

Vzor titulní strany na pevných deskách
Jméno autora a rok ukončení práce také na hřbetní straně

Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering
Department of Physical Electronics

Cell-Centered Lagrangian Code for Simulations in Laser Plasma Physics

Master's thesis

Author: **Bc. Ivan Tarant**
Supervisor: **Ing. Pavel Váchal, Ph.D.**
Consultant: **prof. Ing. Richard Liska, Ph.D.**
Year: **2021/2022**

Před svázáním místo této stránky **vložíte zadání práce** s podpisem děkana (v jedné kopii práce bude list s originálem podpisu).

Toto bude jediný oboustranný list ve Vaší práci!

Acknowledgements:
I would like to express utmost gratitude to my supervisor, Pavel Váchal, for his patience, firm guidance and unrelenting support. Without his knowledge, help and kindness, this thesis would never exist. I would also like to thank my consultant, prof. Richard Liska, for his advice and helpful comments.

Last but definiely not least, I would like to express gratitude to my family, who have supported me unwaveringly throughout my studies.

Declaration

I declare that I wrote this master's thesis independently and exclusively with the use of cited sources.

Prague, May 2, 2022 Ivan Tarant

Contents

Symbols

Symbol	Meaning
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$	Vectors
A, B, C	Matrices
∂G	Boundary of the set G
$\mathbf{a} \otimes \mathbf{b}$	Tensor product of vectors $\mathbf{a}$, $\mathbf{b}$

List of Figures

Název práce:

Cell-Centered lagrangeovský kód pro simulace ve fyzice laserového plazmatu

Autor: Bc. Ivan Tarant

Obor: Laserová fyzika a technika

Druh práce: Diplomová práce

Vedoucí práce: Ing. Pavel Váchal Ph.D., Katedra fyzikální elektroniky, Fakulta jaderná a fyzikálně inženýrská ČVUT v Praze

Abstrakt: Tato práce je věnována rozšíření stávajícího nestrukturovaného cell-centred lagrangeovského kódu pro hydrodynamické simulace, doposud vyvíjeného v kartézských souřadnicích, do cylindrických souřadnic. Součástí je implementace vybraného vhodného numerického schématu pro diskretizaci Eulerových rovnic, společně s řadou testovacích úloh, na kterých je provedena analýza schématu, především z hlediska robustnosti a zachování symetrie. Dalším bodem je návrh a implementace modelu absorpce laserového záření na kritické ploše.

Klíčová slova: cell-centred schéma, lagrangeovská hydrodynamika, laserové plazma, absorpce na kritické ploše

Title:

Cell-Centered Lagrangian Code for Simulations in Laser Plasma Physics

Author: Bc. Ivan Tarant

Abstract: The primary focus of this thesis is the development of a Lagrangian cell-centred unstructured code for hydrodynamic simulations, which was developed in Cartesian coordinates up until this point, and its extension into cylindrical coordinates. One key element is the description and implementation of a selected numerical scheme for the discretization of the Euler equations, accompanied by a variety of test problems, which are used for the analysis of said numerical scheme, especially in terms of robustness and symmetry preservation. Furthermore, a simple model for laser absorption on critical surface will be designed, implemented and tested on a sample simulation.

Key words: cell-centred scheme, Lagrangian hydrodynamics, laser plasma, absorption on critical surface

Introduction

Numerical simulations have always been an integral part of modern physics, allowing to research the character of an unimaginable amout of complex physical phenomena, ranging from mechanics to quantum physics. The study of hydrodynamics is no different in this regard, as CFD (*Computational Fluid Dynamics*) is a discipline with an esteemed reputation, ever since the dawn of modern numerical methods and computers in the late 1940s.

Lagrangian hydrodynamical methods have enjoyed widespread use for many decades now, with numerous practical applications. There are two main types of Lagrangian numerical schemes employed in computing, the *staggered* schemes and *cell-centred* schemes. The designations come from the means of discretization of field variables in the compuational mesh. Whereas staggered schemes define velocity in nodes of the mesh and the rest of the field variables in cell centres, the cell-centred schemes define all field variables in cell centres. Both approaches have their pitfalls and merit.

The staggered schemes have historically been more popular, because storing velocity in mesh nodes naturally solves the issue of Lagrangian mesh movement. However, this comes at a price of loss in consevativity, which has to be compensated by the design of the numerical scheme [8]. In addition, because the staggered discretization corresponds to inviscid flow, it is necessary to ensure proper conversion of kinetic energy into internal in non-smooth flows. This is achieved by the means of *artificial viscosity* [15], which dissipates energy across the shock waves and ensures the validity of the second law of thermodynamics. The staggered schemes have thus been a subject of great interest and considerable research was done on the subject of artificial viscosity, making staggered schemes a very powerful tool in the frame of Lagrangian hydrodynamics.

The cell-centred schemes, on the other hand, are naturally conservative. There is also no need to employ artificial viscosity, because in cell-centred schemes based on the Godunov method, the conversion of kinetic energy into internal is natively provided by the solution of the Riemann problem. The issue of mesh movement however requires resolution, establishing a need to construct solvers for the nodal velocity in order to realize Lagrangian mesh movement, on the condition that they are compatible with the so-called GCL (*Geometric Conservation Law*). Construction of such a nodal solver has historically been somewhat challenging, however, in the recent past, numerous cell-centred schemes have been successfully developed, such as [5] or [9] and have also become a popular subject of research.

However, the use of hydrodynamical simulations is not limited to mere hydrodynamics itself. It is additionally possible to employ them in the framework of plasma simulations, since plasma under certain conditions can be described using a hydrodynamical model. Therefore Lagrangian hydrodynamical methods are also used in the simulations such as ICF (*Inertial Confinement Fusion*), where numerical modeling is an absolutely essential part of research, since perpetually conducting physical experiments would be extremely wasteful, ineffective and expensive. A somewhat less extreme alternative to ICF simulations is its simpler variant, general interaction of radiation with matter, for example interaction of a laser radiation with foils of various materials. Such simulations are one of the foci of the staggered hydrodynamical code PALE [6] (*Prague*

ALE[1]), developed in our institution.

Thus arises an interest of also possessing a cell-centred code, constituting an alternative to the PALE code. Although our code is purely Lagrangian cell-centred, the primary intention is to study the effects generated by the difference in discretization methods.

The primary concern of this thesis is to extend our existing code CCLUS (*Cell-Centred Lagrangian Unstructured Suite*) into cylindrical coordinates, assess its functionality on a variety of reference test problems and implement a simple model of laser absorption on critical surface to simulate interaction of radiation with matter.

In section 1 we shall introduce the main aspects of Lagrangian hydrodynamics in 2D cylindrical coordinates, notably the governing equations and specifics related to the geometry.

Section 2 will be dedicated to the construction of a 2D cylindrical Lagrangian numerical scheme, which was selected for the extension of the CCLUS code into cylindrical geometry.

In section 3 we will briefly describe the physics related to absorption of radiation in plasma and formulate the basic principles of the model of absorption on critical surface.

The focus of section 4 will be to provide necessary information about the structure and functional capabilities of the developed code, discussing some of its notable features and certain practical aspects.

In section 5 we shall present the results obtained using the developed code. We will demonstrate its performance on typical test problems such as Sod shock tube, Noh implosion and Sedov blast wave and finally, we shall present the result of the simulation of laser absorption on critical surface.

1 Lagrangian hydrodynamics in 2D cylindrical coordinates

In this section we shall briefly recapitulate some crucial information and results regarding Lagrangian hydrodynamics which will be relevant in context of this thesis. See publications [9] or [12] for more details.

The primary feature of Lagrangian hydrodynamics is the assumption that the computational volumes $V(t)$ (i.e. computational cells) represent the movement of the fluid. The conserved variables, namely momentum and total energy, are subsequently computed under said assumption. The task at hand is to discretize the governing equations, in this case, the Lagrangian form of the Euler equations, in 2D cylindrical coordinates,

[1] Arbitrary Lagrangian-Eulerian

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \rho \, \mathrm{d}V = 0, \tag{1.1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \mathrm{d}V - \int_{V(t)} \nabla \cdot \mathbf{U} \, \mathrm{d}V = 0, \tag{1.1b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \rho\mathbf{U} \, \mathrm{d}V - \int_{V(t)} \nabla P \, \mathrm{d}V = \mathbf{0}, \tag{1.1c}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \rho E \mathrm{d}V - \int_{V(t)} \nabla \cdot (P\mathbf{U}) \, \mathrm{d}V = 0, \tag{1.1d}$$

where ρ represents density, P pressure, $\mathbf{U}$ velocity, E specific total energy and $\frac{\mathrm{d}}{\mathrm{d}t}$ is the material derivative. Equations (1.1a) to (1.1d) have their physical interpretation, namely (1.1a) is the mass conservation equation, (1.1b) represents the GCL (geometric conservation law), (1.1c) is the conservation of momentum and (1.1d) is conservation of total energy. Note that the GCL (1.1b) captures the time evolution of a computational volume and is equivalent to the local kinematic equation

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \mathbf{U}, \qquad \mathbf{X}|_{t=0} = \mathbf{x}_0, \tag{1.2}$$

where $\mathbf{X}(t)$ is an arbitrary point located on the surface $S(t)$ of the control volume $V(t)$ and $\mathbf{x}_0$ is its starting position.

Thermodynamic closure of the set (1.1) is achieved by an equation of state of the form

$$P = P\left(\rho, \mathcal{E}\right), \tag{1.3}$$

where $\mathcal{E} = E - \frac{1}{2}\left\|\mathbf{U}\right\|^2$ is the specific internal energy.

Now we shall introduce the specifics related to the geometry of 2D cylindrical coordinates. In our particular situation, the use of 2D cylindrical coordinates assumes symmetry in the azimuthal direction, in other words, rotational symmetry around the z-axis. For demonstration purposes we shall stress the dimensionality of defined geometric objects in this section.

It is useful to note here, that in computational fluid dynamics in 2D Cartesian coordinates, it is customary to use the terms *area* and *volume* interchangeably, which can be seen for example in publications [5] or [12]. However, it is crucial to point out that in 2D cylindrical coordinates, these two terms must be properly distunguished, see Figs. 1 - 2, where the distinction is made clear. The importance of proper distinction of these two terms shall become clearer when defining vector operators prior to discretization of our governing equations (1.1). All geometric quantities are computed at a fixed time level, therefore we shall omit the time dependence in the following text (i.e. $V(t) \to V$). Let us have a 2D set Ω localized in the r-z plane and let $\partial\Omega$ be its boundary. Thus, Ω is a 2D surface and $\partial\Omega$ is a curve in 2D, also localized in r-z plane.

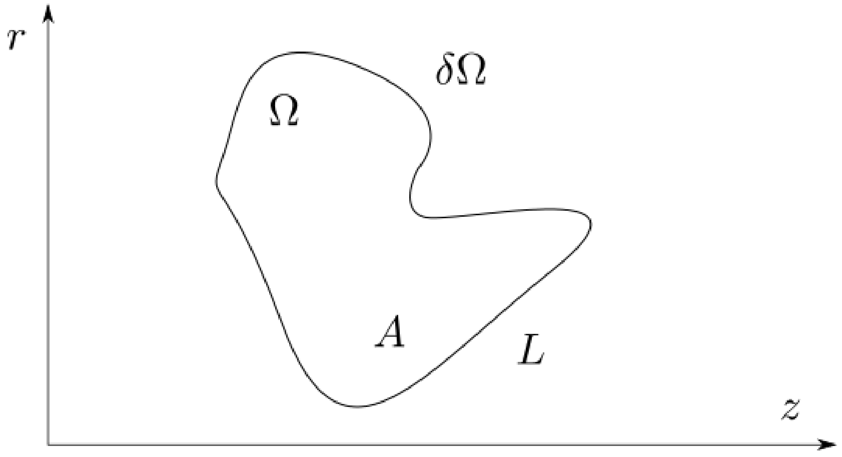


Figure 1: Cylindrical area.

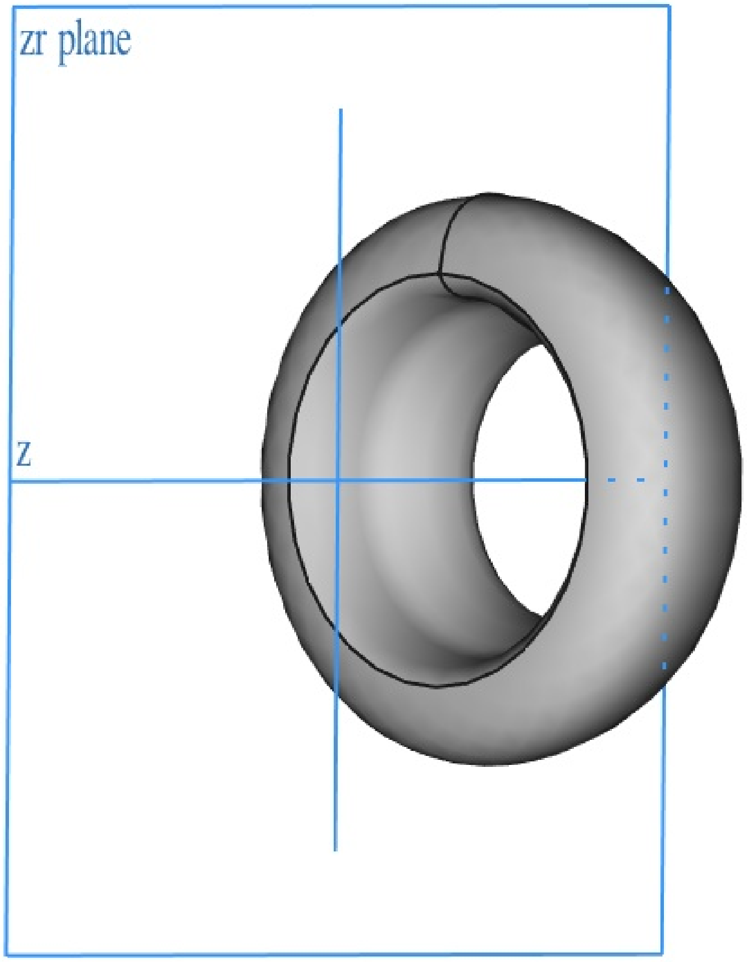


Figure 2: Cylindrical volume of a set Ω located in the $z-r$ plane.

The area A of the set Ω is analogous to 2D Cartesian surface in the sense that

$$A = \int_\Omega \mathrm{d}A = \int_\Omega \mathrm{d}z\ \mathrm{d}r. \tag{1.4}$$

The volume is then calculated by rotating the area A about the z-axis, namely

$$\tilde{V} = \int_{\Omega\times[0,2\pi]} r\ \mathrm{d}A\ \mathrm{d}\varphi, \tag{1.5}$$

however, it is customary to define all quantities per unit angle (radian), in the sense that $V = \frac{\tilde{V}}{2\pi}$, therefore the formula for volume reduces to

$$V = \int_\Omega r\ \mathrm{d}A. \tag{1.6}$$

The transition between (1.5) and (1.6) might be the source of severe confusion for the uninitiated reader, although the process itself is obviously trivial, because it effectively discards the integration in the angular coordinate, thus leading to a certain distortion in comprehension, since the result of the formula (1.6) poses as a 2D object.

In similar fashion, L is the boundary of area A and the surface S is the boundary of the volume V, therefore

$$S = \int_{\partial\Omega} r\ \mathrm{d}L \tag{1.7}$$

Lastly, we define the orthonormal basis

$$\mathbf{e}_z = (1{,}0)\,; \qquad \mathbf{e}_r = (0{,}1)\,,$$

and a normal $\mathbf{N}$ to the curve $L(t)$, which will be used in the following text. The introduced notation is visualized in Fig. 3.

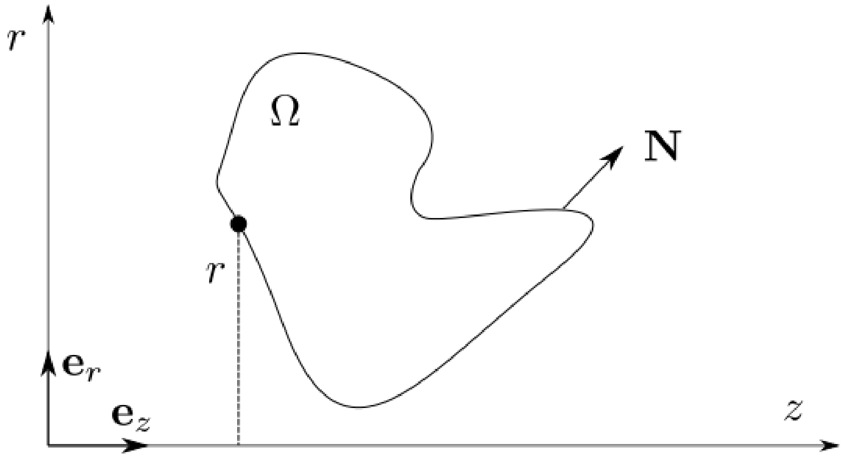


Figure 3: Visualization of notation in cylindrical geometry.

1.1 Discretization of gradient and divergence operators

Similarly to the discretization process undertaken in [12] for 2D Cartesian geometry, the volume integrals also have to be expressed with the use of certain theorems of vector calculus, in this particular case, the Green theorem. The gradient and divergence operators are given by

$$\nabla \cdot \mathbf{U} = \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} + \frac{v}{r}, \tag{1.8}$$

$$\nabla P = \left(\frac{\partial P}{\partial z}, \frac{\partial P}{\partial r} \right) = \frac{\partial P}{\partial z}\mathbf{e}_z + \frac{\partial P}{\partial r}\mathbf{e}_r. \tag{1.9}$$

Firstly, we shall express the second integral in (1.1b) using the expression for the divergence operator (1.8)

$$\int_V \nabla \cdot \mathbf{U} = \int_A \left[\frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} + \frac{v}{r} \right] r \, \mathrm{d}A = \int_L (\mathbf{U} \cdot \mathbf{N}) \, r \, \mathrm{d}L. \tag{1.10}$$

Thus we have obtained the Green theorem (or the 2D form of the divergence theorem), allowing us to express the aforementioned volume integral with a line[2] integral

$$\int_V \nabla \cdot \mathbf{U} = \int_L (\mathbf{U} \cdot \mathbf{N}) \, r \, \mathrm{d}L. \tag{1.11}$$

Now will we similarly rewrite the second integral in (1.1d). With the use of the well-known vector identity

$$\nabla \cdot (\varphi \mathbf{A}) = \mathbf{A} \cdot \nabla \varphi + \varphi \nabla \cdot \mathbf{A}, \tag{1.12}$$

valid for a scalar function φ and a vector $\mathbf{A}$, the second integral of (1.1d) yields

$$\int_L P\mathbf{U} \cdot \mathbf{N} \; r \, \mathrm{d}L = \int_V \mathbf{U} \cdot \nabla P \, \mathrm{d}V + \int_A P \nabla \cdot \mathbf{U} \; r \, \mathrm{d}A. \tag{1.13}$$

If the velocity $\mathbf{U}$ is assumed constant in space, (1.13) takes the form

$$\int_V \nabla P \, \mathrm{d}V = \int_L P\mathbf{N} r \, \mathrm{d}L - \mathbf{e}_r \int_A P \, \mathrm{d}A. \tag{1.14}$$

Contrary to the 2D Cartesian case, the volume integral of the gradient operator is expressed not only by a line integral (1.14), but there is also an additional source term, essentially meaning that in 2D cylindrical case, the surface integral over a closed surface is no longer equal to zero. This term assures the compatibility of the volume integral of the gradient operator with the surface integral of the divergence operator, leading to the first approach to discretization of (1.1), which is commonly referred to as the *control volume* formulation.

[2] It is technically a surface integral, because $r \, \mathrm{d}L$ is actually the surface element dS.

There exists also an alternative approach, the so-called *area-weighted* formulation. This approach rests upon initially setting

$$\int_V \nabla P \mathrm{d}V = \int_A \nabla P \ r \ \mathrm{d}A = \overline{r} \int_A \nabla P \mathrm{d}A, \tag{1.15}$$

where $\overline{r}$ is the average radius of the area A calculated using the mean value theorem

$$\overline{r} = \frac{\int_A r \ \mathrm{d}A}{\int_A \mathrm{d}A}. \tag{1.16}$$

Thus have we reformulated the volume integral of the gradient operator in a manner closely resembling its Cartesian form, save for the additional multiplication by the average radius $\overline{r}$

$$\int_V \nabla P \ \mathrm{d}V = \overline{r} \int_L P\mathbf{N} \ \mathrm{d}L. \tag{1.17}$$

The use of the area-weighted formulation (1.17) primarily leads to the loss of compatibility of the surface integrals of the defined differential operators gradient and divergence. However, it has certain well-desired positive attributes, for example related to symmetry preservation of discretization schemes, as will be discussed and demonstrated later.

We point out that in the 2D Cartesian case both formulations (1.14) and (1.17) coincide, since $r = \overline{r} = 1$ and the source term in (1.14), $\mathbf{e}_r \int_A P \mathrm{d}A$, vanishes.

1.2 Reformulation of the Euler equations for discretization

In the following text, we will rewrite the Euler equations (1.1) in Lagrangian form using the identities involving previously defined differential operators gradient and divergence. We will do so using both *control volume* and *area-weighted* formulations.

1.2.1 Control volume

We will begin with the control volume formulation. Firstly we have to define some new objects related to the Lagrangian formulation, which will be used in this section.

We define a mass density average for an arbitrary fluid variable φ by the expression

$$\langle \varphi \rangle = \frac{\int_V \rho\varphi \ \mathrm{d}V}{\int_V \rho \ \mathrm{d}V} = \frac{\int_V \rho\varphi \ \mathrm{d}V}{m}, \tag{1.18}$$

where mass m of the volume V is defined as

$$m = \int_V \rho \ \mathrm{d}V.$$

Using the aforementioned results and most importantly (1.14), the system (1.1) yields

$$m\frac{\mathrm{d}}{\mathrm{d}t}\left\langle\frac{1}{\rho}\right\rangle - \int_L \mathbf{U}\cdot\mathbf{N}\ r\ \mathrm{d}L = 0, \tag{1.19a}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} - \int_L \mathbf{U}\cdot\mathbf{N}\ r\ \mathrm{d}L = 0, \tag{1.19b}$$

$$m\frac{\mathrm{d}}{\mathrm{d}t}\left\langle\mathbf{U}\right\rangle + \int_L P\mathbf{N}\ r\ \mathrm{d}L - \mathbf{e}_r\int_A P\ \mathrm{d}A = \mathbf{0}, \tag{1.19c}$$

$$m\frac{\mathrm{d}}{\mathrm{d}t}\left\langle E\right\rangle + \int_L P\mathbf{U}\cdot\mathbf{N}\ r\mathrm{d}L = 0. \tag{1.19d}$$

The source term in the momentum conservation equation (1.19c) can still be rewritten in a more favorable form, especially in context of the imminent spatial discretization. Because

$$\int_L \mathbf{N}r\ \mathrm{d}L = A\mathbf{e}_r, \tag{1.20}$$

the source term assumes the form

$$\mathbf{e}_r\int_A P\ \mathrm{d}A = \int_L \overline{P}\mathbf{N}\ r\ \mathrm{d}L, \qquad \overline{P} = \frac{\int_A P\ \mathrm{d}A}{\int_A \mathrm{d}A}, \tag{1.21}$$

where $\overline{P}$ is an averaged pressure over the area A. Therefore we can reformulate the momentum conservation equation (1.19c) in the form of a flux over the outer boundary line elements $\mathrm{d}L$, or technically over surface elements $\mathrm{d}S = r\ \mathrm{d}L$,

$$m\frac{\mathrm{d}}{\mathrm{d}t}\left\langle\mathbf{U}\right\rangle + \int_L \left(P - \overline{P}\right)\mathbf{N}\ r\ \mathrm{d}L = \mathbf{0}. \tag{1.22}$$

1.2.2 Area-weighted

The area-weighted formulation of the Euler equations (1.1) is reached with the use of the expression for the volume integral of the gradient operator (1.17). The only actual difference between the control volume and area-weighted formulations thus appears in the momentum conservation equation. Under the area-weighted formulation, the momentum conservation equation (1.1c) assumes the form

$$m\frac{\mathrm{d}}{\mathrm{d}t}\left\langle\mathbf{U}\right\rangle + \overline{r}\int_L P\mathbf{N}\ \mathrm{d}L = \mathbf{0}. \tag{1.23}$$

If we divide (1.23) by the average radius $\overline{r}$, we arrive at

$$\frac{m}{\overline{r}}\frac{\mathrm{d}}{\mathrm{d}t}\left\langle\mathbf{U}\right\rangle + \int_L P\mathbf{N}\ \mathrm{d}L = \mathbf{0}. \tag{1.24}$$

Upon closer inspection of (1.24), a perceptive reader can realize that this equation is similar to the momentum conservation equation in Cartesian coordinates, the only difference being the division of mass by the averaged radius $m/\overline{r}$. The term $\mu = \frac{m}{\overline{r}}$

is the so-called Cartesian inertia. It is however not constant in time, in other words, it does not have the required Lagrangian conservative property, therefore the use of the momentum equation in this form can be spurious. This is the source of many deficiencies of the area-weighted formulation, notably the violation of conservative properties, which the control volume formulation inherently retains.

With a small amount of foreshadowing, it is possible to summarize at this point that whereas the control volume formulation prioritizes the inherent Lagrangian conservative properties, at the cost of symmetry preservation, which is violated by the source term in the momentum conservation equation (1.19c), the area-weighted formulation prioritizes symmetry preservation by almost artificially removing the source term in the momentum conservation equation, at the cost of certain conservative properties. All of these statements will be demonstrated more intuitively, once we will present the numerical results of discretization schemes derived from both formulations on select test problems.

2 Cylindrical first-order EUCCLHYD numerical scheme

One of the main tasks formulated in this thesis is to extend the cell-centred Lagrangian code developed in [12] and [13], which was up until this point developed in 2D Cartesian coordinates, into 2D cylindrical coordinates. The first numerical scheme implemented in the code in 2D Cartesian coordinates was the first-order EUCCLHYD scheme [9]. Therefore it seemed only natural to select its first-order cylindrical equivalent, introduced in [7], as the first scheme-of-choice for the extension of this code into cylindrical coordinates. The reason is quite apparent, since the first-order cylindrical EUCCLHYD is formulated both in control volume and area-weighted forms. Therefore, both formulations can be tested and compared, especially when simulating laser absorption. Additionally, the scheme and its properties are well-documented which is extremely beneficial as far as result assessment is concerned, not to mention the fact that the treatment of boundary conditions in this framework is rather simple, which is extremely important from the technical perspective. Such details regarding implementation will be discussed in a separate section.

In this chapter, we will introduce the first-order cylindrical EUCCLHYD numerical scheme for the discretization of the Euler equations in unstructured formulation.

From this moment on we are operating in the realm of the discrete numerical formulation of the physical system described by the Euler equations (1.1). For this purpose, we assume a certain domain in 2D cylindrical coordinates, which consists of a number of generally polygonal computational cells, which we denote $\Omega_c(t)$. The cells are not overlapping, nor can there be gaps between them, so that they constitute a mesh covering the entire domain. Since we are operating in an unstructured formulation, each cell is assigned a unique index c. Similarly, vertices of this mesh are denoted by their unique index p. Since every cell $\Omega_c(t)$ is defined by the position of its vertices, we denote the anti-clockwise ordered list of vertices of a cell by $\mathcal{P}(c)$.

The list of cell vertices is the bare minimum in terms of mesh connectivity, which we require for a description of this numerical scheme. However, practically speaking, we typically store many more lists containing mesh connectivity information, such as lists containing cell-cell connectivity etc., mostly to accelerate the implementation. Again, such practical details will be discussed at an appropriate time.

2.1 Face flux discretization

Now we shall conduct the discretization of (1.1) itself. It is desired to express the system (1.1) in the framework of cell, nodal and subzonal variables, analogously to the Cartesian discretization in [9] or [12]. However, we shall initially formulate the discretization using the so-called *numerical face flux* form. The reason for this decision is motivated by the desire to provide some substance to the claims related to symmetry preservation of control volume and area-weighted formulations.

2.1.1 Control volume formulation

To conduct the numerical face flux discretization in the control volume formulation, we have to apply (1.19) to the discrete computational cells $\Omega_c(t)$. Also, we introduce an anti-clockwise ordered list of cell faces $\mathcal{F}(c)$. The faces themselves can be recovered by their vertices, which are p and p^+ in the anti-clockwise sense, see Fig. 4 for a visual representation of used notation. Thus we arrive at the discrete form of the system (1.19)

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \sum_{f\in\mathcal{F}(c)} r_f^c L_f^c \mathbf{U}_f^c \cdot \mathbf{N}_f^c = 0, \tag{2.1a}$$

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_c + \sum_{f\in\mathcal{F}(c)} r_f^c L_f^c \Pi_f^c \mathbf{N}_f^c - A_c P_c \mathbf{e}_r = \mathbf{0}, \tag{2.1b}$$

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}E_c + \sum_{f\in\mathcal{F}(c)} r_f^c L_f^c \left(\Pi\mathbf{U}\right)_f^c \cdot \mathbf{N}_f^c = 0, \tag{2.1c}$$

where $\mathbf{N}_f^c$ is the face unit outward normal, L_f^c is the face length and the face radius r_f^c is given by the average of its endpoint radii

$$r_f^c = \frac{1}{2}\left(r_p + r_{p+}\right).$$

Similarly to (1.22), the discrete momentum conservation equation (2.1b) can be rewritten using the geometric identity (its discrete form) (1.20) to obtain

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_c + \sum_{f\in\mathcal{F}(c)} r_f^c L_f^c \left(\Pi_f^c - P_c\right)\mathbf{N}_f^c = \mathbf{0}. \tag{2.2}$$

The numerical face fluxes present in the sums of (2.1) are given by

$$\mathbf{U}_f^c = \frac{1}{r_f^c L_f^c}\int_f \mathbf{U} r \,\mathrm{d}L, \tag{2.3a}$$

$$\Pi_f^c = \frac{1}{r_f^c L_f^c}\int_f P r \,\mathrm{d}L, \tag{2.3b}$$

$$(\Pi\mathbf{U})_f^c = \frac{1}{r_f^c L_f^c}\int_f P\mathbf{U} r \,\mathrm{d}L. \tag{2.3c}$$

Finally, the kinematic equation (1.2), after discretization, assumes the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{X}_p = \mathbf{U}_p, \qquad \mathbf{X}_p|_{t=0} = \mathbf{x}_p. \tag{2.4}$$

The equation (2.4) serves as the prescription for the mesh movement, since $\mathbf{U}_p$ denotes nodal velocity and $\mathbf{X}_p$ nodal positions.

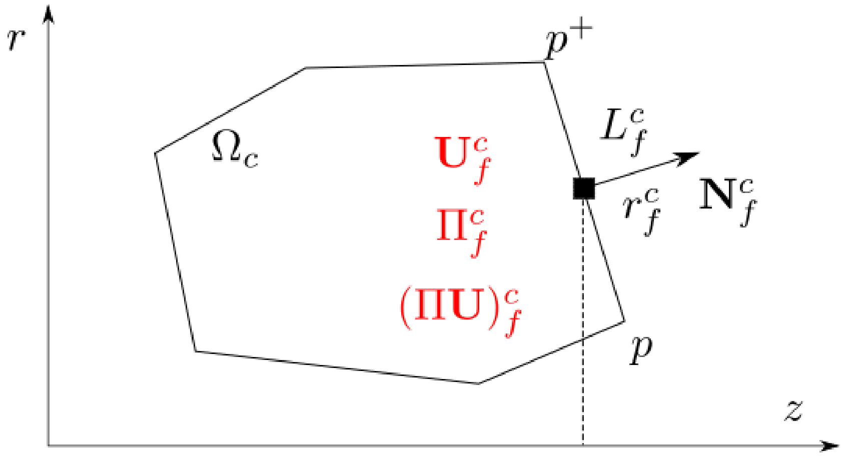


Figure 4: Visualization of notation in the numerical face flux discretization.

2.1.2 Area-weighted formulation

Again, to reach the area-weighted formulation, we will use the identity (1.17), this time in its discrete form. We remark once again that the only difference between the area-weighted and control volume formulation lies in the form of the momentum conservation equation (1.23). We immediately arrive at

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_c + \overline{r_c} \sum_{f \in \mathcal{F}(c)} L_f^c \widehat{\Pi_f^c} \mathbf{N}_f^c = \mathbf{0}, \tag{2.5}$$

where $\overline{r_c} = V_c/A_c$ is the cell-averaged radius and $\widehat{\Pi_f^c}$ is the area-weighted face pressure flux, defined by

$$\widehat{\Pi_f^c} = \frac{1}{L_f^c} \int_f P \; \mathrm{d}L. \tag{2.6}$$

If we divide (2.5) by the cell-averaged radius $\overline{r_c}$, we reach

$$\frac{m_c}{\overline{r_c}} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_c + \sum_{f \in \mathcal{F}(c)} L_f^c \widehat{\Pi_f^c} \mathbf{N}_f^c = \mu_c \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_c + \sum_{f \in \mathcal{F}(c)} L_f^c \widehat{\Pi_f^c} \mathbf{N}_f^c = \mathbf{0}, \tag{2.7}$$

which is phenomenologically the Cartesian form of the discrete momentum conservation equation, where $\mu_c = m_c/\overline{r_c} = \rho_c A_c$ is the cell Cartesian inertia.

The main reason why we decided to include the numerical face flux discretization in area-weighted and control volume formulations was due to the claims made about symmetry preservation, as we remarked beforehand. It is not the purpose of this thesis to analyze or verify these properties, but using the derived expressions (2.1) for the control volume numerical face flux discretization or (2.1a), (2.7), (2.1c) for the area-weighted numerical face flux discretization, it can be proven, that the area-weighted formulation preserves symmetry on equi-angular polar meshes, contrary to the control volume formulation. We shall not do so within the scope of this thesis, for the full proof see [7].

At this moment, we only lack a solver providing the nodal velocities for the mesh movement. We shall rectify this momentarily, since we will conduct the full first-order discretization in the following section.

2.2 Node-based discretization

One of the primary issues concerning cell-centred Lagrangian numerical schemes is the fact that, unlike staggered schemes, the velocities are defined in cell centres and not in nodes. Therefore there is no inherent means to move the mesh and it is necessary to construct a solver to obtain the nodal velocities $\mathbf{U}_p$, in order to realize said mesh movement. Since the mesh movement is equivalent to cellular volume variation, the nodal solver has to be compatible with the volume variation equation (GCL) (2.1a). Let us rewrite (2.1a) into a more suitable form, so the use of the label *volume variation equation* is justified. Since $\frac{m_c}{\rho_c} = V_c$, (2.1a) becomes

$$\frac{\mathrm{d}V_c}{\mathrm{d}t} - \sum_{f \in \mathcal{F}(c)} r_f^c L_f^c \mathbf{U}_f^c \cdot \mathbf{N}_f^c = 0. \tag{2.8}$$

As we can see in (2.8) however, the nodal velocities $\mathbf{U}_p$ are still hidden behind the face velocities $\mathbf{U}_f$. For this reason, we will perform a transition from the numerical face flux discretization to a node-based discretization.

For a graphic visualization of the situation at hand see Fig. 5.

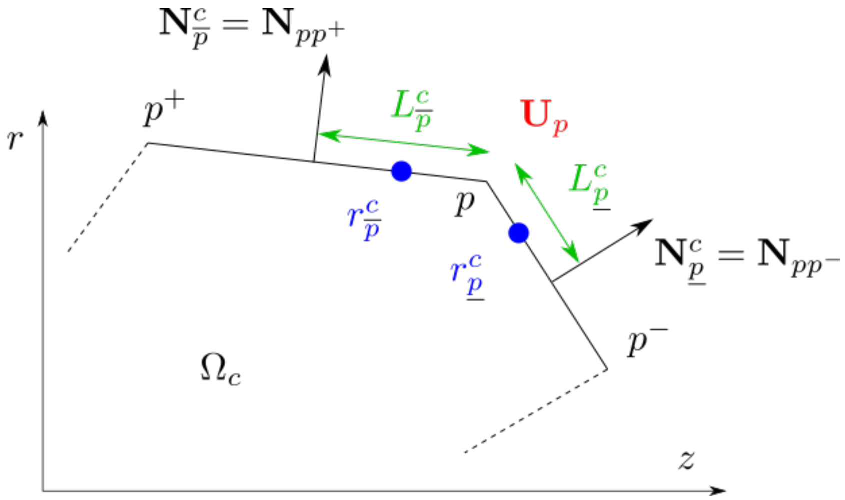


Figure 5: Node-based discretization in cell Ω_c.

Firstly, we will perform a linear interpolation of the velocity field $\mathbf{U}_c$ along the cell faces to be able to express the face velocities $\mathbf{U}_f$ using nodal velocities $\mathbf{U}_p$. It is in fact an interpolation of a product of two linear functions $\mathbf{U}$ and r, thus leading to the recognizable Simpson's rule. This process yields the conversion expression between the numerical face flux and node-based discretizations

$$r^c_f \mathbf{U}^c_f = \frac{1}{2}\left(\frac{r_{p^+} + 2r_p}{3}\mathbf{U}_p + \frac{r_p + 2r_{p^+}}{3}\mathbf{U}_{p^+}\right) = 0. \tag{2.9}$$

The previous equation essentially expresses the velocity of a face f using the velocities of its endpoints p and p^+. The idea is to express the nodal velocity $\mathbf{U}_p$ using the knowledge of geometry of the cell faces surrounding it. To that effect, we will define new geometric objects corresponding to the announced shift in perception of the situation (i.e. from expressing face using nodes to expressing a node using connected faces in a cell).

Assuming we are located at a node p, we define the half-lengths $L^c_{\underline{p}}$, $L^c_{\overline{p}}$, outward-facing normals $\mathbf{N}^c_{\underline{p}}$, $\mathbf{N}^c_{\overline{p}}$ and radii $r^c_{\underline{p}}$, $r^c_{\overline{p}}$ of each connected face (in the anti-clockwise sense) as follows

$$\begin{aligned} L^c_{\underline{p}} &= \frac{1}{2}L_{pp^-}, & \mathbf{N}^c_{\overline{p}} &= \mathbf{N}_{pp^-}, \\ L^c_{\underline{p}} &= \frac{1}{2}L_{pp^-}, & \mathbf{N}^c_{\overline{p}} &= \mathbf{N}_{pp^-}, \\ r^c_{\underline{p}} &= \frac{r_{p^-} + 2r_p}{3}, & r^c_{\overline{p}} &= \frac{r_{p^+} + 2r_p}{3}. \end{aligned} \tag{2.10}$$

Additionally, we define nodal pressures $\Pi^c_{\underline{p}}$ and $\Pi^c_{\overline{p}}$ for later use in the discretization process, see Fig. 6 for a graphic visualization of their localization in the cell corner.

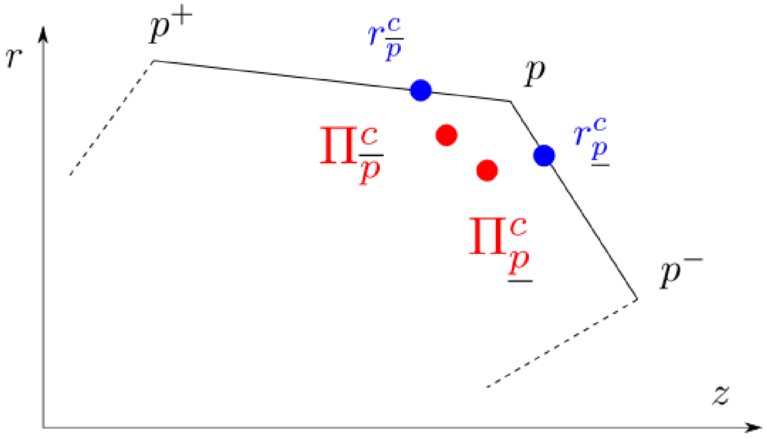


Figure 6: Localization of nodal pressures at node p

Similarly to the nodal velocities $\mathbf{U}_p$, the nodal pressures $\Pi^c_{\underline{p}}$ and $\Pi^c_{\overline{p}}$ are obtained by linear interpolation along the cell faces. The face momentum flux $\widehat{\Pi^c_f}$ is then expressed using nodal pressures

$$\widehat{\Pi^c_f} = \frac{1}{2}\left(\Pi^c_{\underline{p^+}} + \Pi^c_{\overline{p}}\right).$$

Although it seems that the nodal pressures $\Pi^c_{\underline{p}}$ and $\Pi^c_{\overline{p}}$ introduce extra unknowns into the system, they can fortunately be expressed using nodal velocity, using the approximate half Riemann problems [3]. Since we will employ the acoustic approximation, the nodal pressures can be calculated using

$$\begin{aligned}\Pi^c_{\underline{p}} &= P_c - Z_c\left(\mathbf{U}_p - \mathbf{U}_c\right)\cdot\mathbf{N}^c_{\underline{p}},\\ \Pi^c_{\underline{p}} &= P_c - Z_c\left(\mathbf{U}_p - \mathbf{U}_c\right)\cdot\mathbf{N}^c_{\underline{p}},\\ Z_c &= \rho_c a_c,\end{aligned} \tag{2.11}$$

where Z_c is the acoustic impedance of the cell and a_c is the cell sound speed.

If we substitute (2.9) and (2.10) into the equation (2.8) and shift the nodal indices $p^{\pm}$ appropriately, we arrive at

$$\frac{\mathrm{d}V_c}{\mathrm{d}t} - \sum_{p\in\mathcal{P}(c)}\left(r^c_{\underline{p}}L^c_{\underline{p}}\mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}}L^c_{\overline{p}}\mathbf{N}^c_{\overline{p}}\right)\cdot\mathbf{U}_p = 0. \tag{2.12}$$

The equation (2.12) represents the GCL in node-based discretization, but it is also useful in a couple more ways. Firstly, if the term with the sum is shifted to the right-hand-side and subsequently divided by cell volume, i.e.

$$\frac{1}{V_c}\frac{\mathrm{d}V_c}{\mathrm{d}t} = \left\langle\nabla\cdot\mathbf{U}\right\rangle_c = \frac{1}{V_c}\sum_{p\in\mathcal{P}(c)}\left(r^c_{\underline{p}}L^c_{\underline{p}}\mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}}L^c_{\overline{p}}\mathbf{N}^c_{\overline{p}}\right)\cdot\mathbf{U}_p, \tag{2.13}$$

we have recovered an expression for the divergence operator in the node-based discretization. Also, if we remind ourselves of the identity (1.20), the sum in (2.12) represents the integral on the left-hand side. Therefore we can write the form of the identity (1.20) in our node-based discretization as

$$\sum_{p\in\mathcal{P}(c)}\left(r^c_{\underline{p}}L^c_{\underline{p}}\mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}}L^c_{\overline{p}}\mathbf{N}^c_{\overline{p}}\right) = A_c\mathbf{e}_r. \tag{2.14}$$

The newly obtained identity (2.14) also provides us with a direct means to calculate the cell area A_c. At this point in text, since we have stumbled upon the formula for cell area, we will also include the formula for calculating the cell volume.

The cell volume V_c of a generally polygonal cell Ω_c is calculated using its triangular decomposition and subsequent rotation around the z-axis, see Fig. 7 for a graphic visualization.

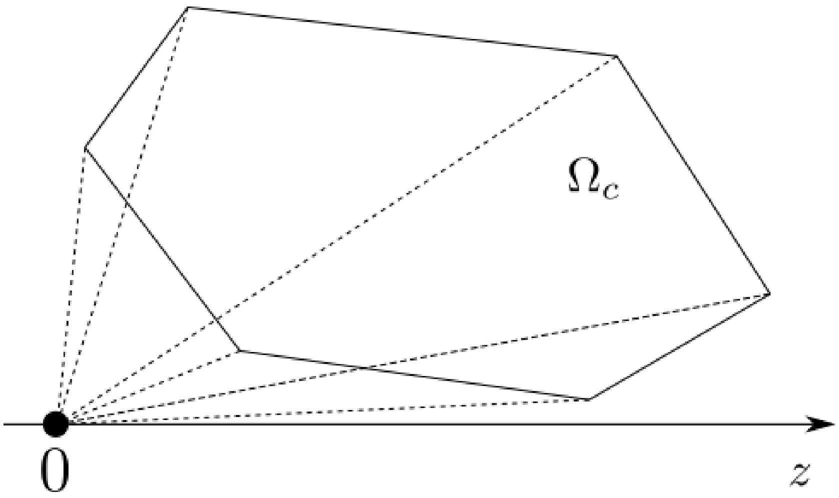


Figure 7: Calculation of cell volume V_c of a cell Ω_c.

The formula for cell volume then writes

$$V_c = \frac{1}{6} \sum_{p \in \mathcal{P}(c)} \left(r_p + r_{p^+}\right) \left(z_p r_{p^+} - r_p z_{p^+}\right). \tag{2.15}$$

The numerical face flux and node-based discretizations can also be linked by the geometric identity (2.14), since

$$\begin{aligned} \sum_{p \in \mathcal{P}(c)} \left(r_{\underline{p}}^c L_{\underline{p}}^c \mathbf{N}_{\underline{p}}^c + r_{\overline{p}}^c L_{\overline{p}}^c \mathbf{N}_{\overline{p}}^c \right) &= A_c \mathbf{e}_r, \\ \sum_{f \in \mathcal{F}(c)} r_f^c L_f^c \mathbf{N}_f^c &= A_c \mathbf{e}_r, \\ \Longrightarrow \sum_{p \in \mathcal{P}(c)} \left(r_{\underline{p}}^c L_{\underline{p}}^c \mathbf{N}_{\underline{p}}^c + r_{\overline{p}}^c L_{\overline{p}}^c \mathbf{N}_{\overline{p}}^c \right) &= \sum_{f \in \mathcal{F}(c)} r_f^c L_f^c \mathbf{N}_f^c. \end{aligned} \tag{2.16}$$

At this point, we can discretize the momentum conservation equation using the introduced node-based discretization. We will do so both for the control volume and the area-weighted variants, using their respective derived expressions for the differential operators.

2.2.1 Control volume discretization of the momentum equation

For the control volume variant, we shall make use of the discrete form of (1.14) and its compatible discrete divergence operator from (2.13). The discrete cell pressure gradient in control volume formulation then becomes

$$\langle \nabla P \rangle_c^{CV} = \frac{1}{V_c} \left[\sum_{p \in \mathcal{P}(c)} \left(r_{\underline{p}}^c L_{\underline{p}}^c \Pi_{\underline{p}}^c \mathbf{N}_{\underline{p}}^c + r_{\overline{p}}^c L_{\overline{p}}^c \Pi_{\overline{p}}^c \mathbf{N}_{\overline{p}}^c \right) - A_c P_c \mathbf{e}_r \right]. \tag{2.17}$$

Using the discrete pressure gradient operator (2.17), we can express the momentum conservation equation in the form

$$m_c \frac{\mathrm{d}\mathbf{U}_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \Pi^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \Pi^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) - A_c P_c \mathbf{e}_r = \mathbf{0}. \tag{2.18}$$

We have thus derived a node-based control volume discretization of the momentum conservation equation, which is equivalent to the numerical face flux discretization (2.1b), on the condition that the numerical face flux satisfies

$$r^c_f \Pi^c_f = \frac{1}{2} \left(\frac{2r_p + r_{p+}}{3} \Pi^c_{\overline{p}} + \frac{r_p + 2r_{p+}}{3} \Pi^c_{\underline{p+}} \right), \tag{2.19}$$

in other words, that the nodal values were obtained by linear interpolation along the cell faces.

2.2.2 Area-weighted discretization of the momentum equation

The area-weighted variant of the momentum conservation equation is obtained using the corresponding definition of the gradient operator (1.17) in its discrete form

$$\langle \nabla P \rangle^{AW}_c = \frac{\overline{r_c}}{V_c} \sum_{p\in\mathcal{P}(c)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \Pi^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \Pi^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right). \tag{2.20}$$

Under these conditions, the area-weighted variant of the momentum conservation equation assumes the form

$$m_c \frac{\mathrm{d}\mathbf{U}_c}{\mathrm{d}t} + \overline{r_c} \sum_{p\in\mathcal{P}(c)} \left(L^c_{\underline{p}} \Pi^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + L^c_{\overline{p}} \Pi^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) = \mathbf{0}. \tag{2.21}$$

The area-weighted variant of the interpolaton condition is as follows

$$\widehat{\Pi^c_f} = \frac{1}{2} \left(\Pi^c_{\underline{p}} + \Pi^c_{\overline{p}} \right). \tag{2.22}$$

2.2.3 Node-based discretization of the energy conservation equation

Finally, once possessing a discretization of the divergence operator and using the node-based discretization introduced in the beginning of section 2.2, we can discretize the final equation of the system (1.1). The node-based discretization of the energy conservation equation is expressed as

$$m_c \frac{\mathrm{d}E_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \Pi^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \Pi^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) \cdot \mathbf{U}_p = 0. \tag{2.23}$$

Once more, this formulation is valid only if the following condition, expressing the compatibility of the node-based discretization with the numerical face flux discretization, is satisfied

$$r^c_f \left(\Pi\mathbf{U}\right)^c_f = \frac{1}{2}\left(\frac{2r_p + r_{p+}}{3}\Pi^c_{\overline{p}}\mathbf{U}^c_p + \frac{r_p + 2r_{p+}}{3}\Pi^c_{\underline{p+}}\mathbf{U}^c_{p+}\right). \tag{2.24}$$

2.2.4 Nodal solver compatible with GCL

The current endeavour culminates in the search for the nodal velocities $\mathbf{U}_p$. To that end we have to construct a solver, under the condition that the mesh movement realized be the computed velocities $\mathbf{U}_p$ is compatible with the GCL (1.1b). The construction of the solver has already been thoroughly described in [9], in the Cartesian variant or in [7], in the cylindrical variant, therefore we shall be rather brief and mention only the most important elements regarding its construction. The construction itself is derived, quite predictably, from conservation of total energy and conservation of momentum. One of the ubiquitous concepts appearing in the cell-centred numerical schemes in Cartesian coordinates implemented in [12] and [13], as far as the quest for nodal velocities $\mathbf{U}_p$ is concerned, is the solution of 1D Riemann problems in the direction of the face. This cell-centred cylindrical numerical scheme is not any different in this regard.

As we were saying, the first step is to assume total energy conservation around the node p. Let $\mathcal{C}(p)$ be a list of cells connected to node p. In that case, total energy conservation is represented by the sufficient condition

$$\sum_{c\in\mathcal{C}(p)} \underbrace{r^c_{\underline{p}} L^c_{\underline{p}} \Pi^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \Pi^c_{\overline{p}} \mathbf{N}^c_{\overline{p}}}_{\mathbf{F}_{pc}} = \mathbf{0}. \tag{2.25}$$

The argument of the sum in (2.25) can be interpreted mechanically as a subzonal[3] force $\mathbf{F}_{pc}$ impinging at node p. Therefore the condition (2.25) is merely expressing the balance of subzonal forces around node p. Substituting the defined nodal pressures (2.11) into the expression for the subzonal force in (2.25), the subzonal force can be written as

$$\mathbf{F}_{pc} = \left(r^c_{\underline{p}} L^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \mathbf{N}^c_{\overline{p}}\right) P_c - \mathsf{M}_{pc}\cdot\left(\mathbf{U}_p - \mathbf{U}_c\right), \tag{2.26}$$

where M_{pc} is the subzonal matrix defined as

$$\mathsf{M}_{pc} = Z_c\left[r^c_{\underline{p}} L^c_{\underline{p}}\left(\mathbf{N}^c_{\underline{p}} \otimes \mathbf{N}^c_{\underline{p}}\right) + r^c_{\overline{p}} L^c_{\overline{p}}\left(\mathbf{N}^c_{\overline{p}} \otimes \mathbf{N}^c_{\overline{p}}\right)\right]. \tag{2.27}$$

Since nonzero nodal velocity $\mathbf{U}_p$ stems from an imbalance of forces at node p, the solver then, using (2.25), (2.26) and (2.27), finally assumes the form

$$\mathsf{M}_p\mathbf{U}_p = \sum_{c\in\mathcal{C}(p)} \mathbf{F}_{pc}, \tag{2.28}$$

where nodal matrix M_p is defined as

$$\mathsf{M}_p = \sum_{c\in\mathcal{C}(p)} \mathsf{M}_{pc}. \tag{2.29}$$

[3] Also known as *subcell*.

It is important to note at this point, that the subzonal matrix M_{pc} defined by (2.27) is symmetrical positive-definite by design, thus even the nodal matrix M_p is symmetrical positive-definite, and therefore always invertible. In other words, the nodal solver (2.28) is well-defined and always provides a valid solution.

It may be useful to note here that we have yet to take the boundary conditions into account, which, in the framework of cell-centred numerical schemes can sometimes prove rather tricky. Although this particular scheme is indeed provided with a proper boundary condition treatment from the theoretical standpoint, from experience, the practical implementation may or may not require certain ingenuity beyond a theoretically sound formula. We shall elaborate on the topic of practical implementation of boundary conditions at an appropriate time.

2.2.5 Boundary condition treatment

As we remarked, now it is necessary to apply the boundary conditions. As far as this numerical scheme is concerned, the boundary conditions are applied in the nodal solver, in other words, we have to properly calculate the balance of subzonal pressures at boundary nodes. We are assuming two situations in this regard. Either we are operating with prescribed pressure at the boundary edge or we shall have prescribed velocity at an edge. In any case, the boundary condition manifests itself formally as a missing subcell force contribution in the subcell force balance around node p (2.25), where we re-designate $p^{\pm}$ as next and previous nodes situated on the boundary. We shall designate the boundary force term as $\mathbf{F}_p^{\star}$ which is defined phenomenologically as

$$\sum_{c \in \mathcal{C}(p)} \mathbf{F}_{pc} = \mathbf{F}_p^{\star}. \tag{2.30}$$

Thus the nodal solver (2.28) for nodes p on the boundary of the domain becomes

$$\mathsf{M}_p \mathbf{U}_p = \sum_{c \in \mathcal{C}(p)} \mathbf{F}_{pc} - \mathbf{F}_p^{\star}. \tag{2.31}$$

Our remaining task is now to specify the nodal boundary force term $\mathbf{F}_p^{\star}$ for both assumed types of boundary conditions.

Prescribed pressure

The case of prescribed pressure is fairly simple and easy to comprehend. Assuming we have a certain pressure prescribed on the boundary cell faces surrounding the node p, the boundary force term is quite naturally

$$\mathbf{F}_p^{\star} = r_{\underline{p}}^{\star} L_{\underline{p}}^{\star} \Pi_{\underline{p}}^{\star} \mathbf{N}_{\underline{p}}^{\star} + r_{\overline{p}}^{\star} L_{\overline{p}}^{\star} \Pi_{\overline{p}}^{\star} \mathbf{N}_{\overline{p}}^{\star}, \tag{2.32}$$

see Fig. 8 for visualization of notation on the domain boundary.

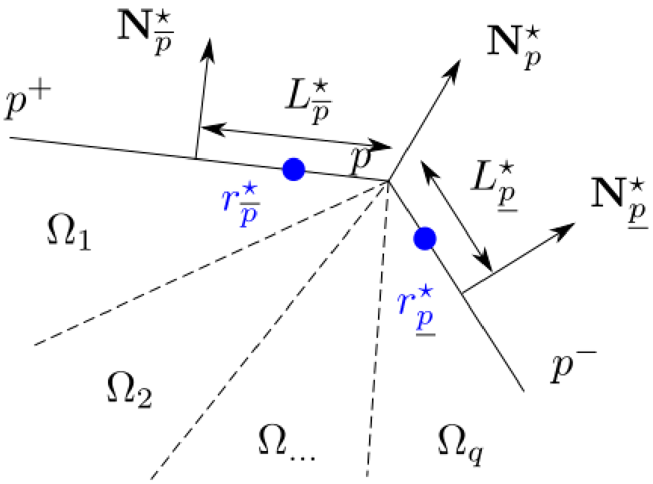


Figure 8: Boundary nodes.

Prescribed velocity

In the case of velocity prescribed on the boundary faces, the situation is a bit more complicated. First, we have to project the general prescribed velocity $\boldsymbol{\mathcal{V}}^\star$ onto the faces, to obtain the normal velocity applied on the boundary faces

$$\begin{aligned} \mathcal{V}^\star_{\underline{p}} &= \mathbf{N}^\star_{\underline{p}} \cdot \boldsymbol{\mathcal{V}}^\star, \\ \mathcal{V}^\star_{\overline{p}} &= \mathbf{N}^\star_{\overline{p}} \cdot \boldsymbol{\mathcal{V}}^\star. \end{aligned} \tag{2.33}$$

Then we define the nodal normal $\mathbf{N}_p$, using the expression

$$\mathbf{N}_p = \frac{\mathbf{N}^\star_{\underline{p}} + \mathbf{N}^\star_{\overline{p}}}{\left\| \mathbf{N}^\star_{\underline{p}} + \mathbf{N}^\star_{\overline{p}} \right\|}. \tag{2.34}$$

Now we have to translate the normal velocities prescribed on the edges into an average pressure acting at the node p. Using the results already obtained in [12], we arrive at the formula for the average boundary nodal pressure $\Pi^\star_p$.

$$\Pi^\star_p = \frac{\left[\mathsf{M}_p^{-1} \left(\sum_{c \in \mathcal{C}(p)} \mathbf{F}_{cp} \right) \cdot \mathbf{N}_p \right] - \left(r^\star_{\underline{p}} L^\star_{\underline{p}} \mathcal{V}^\star_{\underline{p}} + r^\star_{\overline{p}} L^\star_{\overline{p}} \mathcal{V}^\star_{\overline{p}} \right)}{\mathsf{M}_p^{-1} \mathbf{N}_p \cdot \mathbf{N}_p}. \tag{2.35}$$

The boundary force for the velocity boundary condition is then

$$\mathbf{F}^\star_p = \left(r^\star_{\underline{p}} L^\star_{\underline{p}} \mathbf{N}^\star_{\underline{p}} + r^\star_{\overline{p}} L^\star_{\overline{p}} \mathbf{N}^\star_{\overline{p}} \right) \Pi^\star_p. \tag{2.36}$$

2.2.6 Time step control

The time step is controlled using three requirements. The first one is the standard CFL condition, the second condition is based on limiting cell volume variation over a time step and the third serves as a condition, which allows the time step to increase during the simulation. We shall introduce all of them in this section.

CFL condition

The CFL condition concerns itself with the numerical scheme's stability. The CFL condition employed in this case, in the physical point of view, aims to preserve positive entropy generation in cells Ω_c across the time step. A time step calculated using such a condition can be expressed as

$$\Delta t_{\mathrm{cfl}} = C_{\mathrm{cfl}} \min_c \left(\frac{L_c^n}{a_c^n} \right), \tag{2.37}$$

where C_{cfl} is the Courant-Friedrichs-Lewy number, a coefficient satisfying

$$C_{\mathrm{cfl}} \ \in \ (0,1],$$

L_c^n is the distance between two closest nodes of a cell Ω_c and a_c^n is the cell sound speed. The coefficient C_{cfl} is selected based on a stability analysis conducted for the particular scheme. According to [7], the choice of $C_{\mathrm{cfl}} = 0.25$ provides stable numerical results and is compatible with entropy monotonicity, therefore we regard this value as a benchmark.

Volume variation condition

The second condition aims to limit the volume variation of a cell during a time step. This particular condition is conceptually very natural and extremely useful from the practical standpoint. Firstly, assuming we are located in cell Ω_c at the time level t^n with volume V_c^n, we will estimate the new cell volume V_c^{n+1} at the next time level using the Taylor expansion

$$V_c^{n+1} = V_c^n + \frac{\mathrm{d}}{\mathrm{d}t} V_c(t^n)\Delta t,$$

where the volume derivative is calculated using the discrete GCL (volume variation equation) (2.12). The limit for volume variation is expressed using a strictly positive coefficient C_V, therefore we aim to calculate the new volume V_c^{n+1} so that

$$\frac{|V_c^{n+1} - V_c^n|}{V_c^n} \leq C_V.$$

To achieve that, the time step must be

$$\Delta t_V = C_V \min_c \left[\frac{V_c^n}{\left| \frac{\mathrm{d}}{\mathrm{d}t} V_c(t^n) \right|} \right]. \tag{2.38}$$

In practice, we usually require that the new cell volume V_c^{n+1} does not vary from the initial volume V_c^n by more than 10%, in other words $C_V = 0.1$.

Full formula for the time step

The timestep is then evaluated by selecting the minimum of three proposed time steps. The third condition we mentioned simply prevents the time step to increase too rapidly, should time steps calculated from (2.37) or (2.38) be much greater than the previous one. Such a condition is expressed using a strictly positive coefficient C_t. To be precise, the time step selection is performed by evaluating

$$\Delta t^{n+1} = \min\left(\Delta t_{\text{cfl}}, \Delta t_V, C_t \Delta t^n\right). \tag{2.39}$$

We typically require that the maximum time step increase is 1%, in other words $C_t = 0.01$.

2.2.7 Equation of state

The scheme achieves thermodynamic closure with the use of an equation of state (1.3). In this case we use the equation of state for the ideal gas in the form

$$P = \rho \mathcal{E} \left(\gamma - 1\right), \tag{2.40}$$

where γ is the Poisson constant.

2.2.8 Discretization summary for the first-order cylindrical EUCCLHYD numerical scheme

Finally, we present the results of the discretization for both control volume and area-weighted formulations. Firstly, we will mention the variant-specific equations and then summarize the common features. We include all necessary formulas used for the construction of the numerical scheme below.

Control volume

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \sum_{p \in \mathcal{P}(c)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) \cdot \mathbf{U}_p = 0, \tag{2.41a}$$

$$m_c \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{U}_c + \sum_{p \in \mathcal{P}(c)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \Pi^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \Pi^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) - A_c P_c \mathbf{e}_r = \mathbf{0}, \tag{2.41b}$$

$$m_c \frac{\mathrm{d}}{\mathrm{d}t} E_c + \sum_{p \in \mathcal{P}(c)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \Pi^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \Pi^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) \cdot \mathbf{U}_p = 0, \tag{2.41c}$$

Area-weighted

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \sum_{p \in \mathcal{P}(c)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) \cdot \mathbf{U}_p = 0, \tag{2.42a}$$

$$m_c \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{U}_c + \overline{r_c} \sum_{p \in \mathcal{P}(c)} \left(L^c_{\underline{p}} \Pi^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + L^c_{\overline{p}} \Pi^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) - A_c P_c \mathbf{e}_r = \mathbf{0}, \tag{2.42b}$$

$$m_c \frac{\mathrm{d}}{\mathrm{d}t} E_c + \sum_{p \in \mathcal{P}(c)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \Pi^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \Pi^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) \cdot \mathbf{U}_p = 0, \tag{2.42c}$$

Common

The discrete kinematic equation (2.4) and the derived nodal solver (2.28) are the same for both variants

$$\begin{aligned}
\mathbf{U}_p &= \mathsf{M}_p^{-1} \sum_{c \in \mathcal{C}(p)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right) P_c - \mathsf{M}_{pc} \cdot \left(\mathbf{U}_p - \mathbf{U}_c \right), \\
\mathsf{M}_{pc} &= Z_c \left[r^c_{\underline{p}} L^c_{\underline{p}} \left(\mathbf{N}^c_{\underline{p}} \otimes \mathbf{N}^c_{\underline{p}} \right) + r^c_{\overline{p}} L^c_{\overline{p}} \left(\mathbf{N}^c_{\overline{p}} \otimes \mathbf{N}^c_{\overline{p}} \right) \right], \quad \mathsf{M}_p = \sum_{c \in \mathcal{C}(p)} \mathsf{M}_{pc}, \\
\Pi^c_{\underline{p}} &= P_c - Z_c \left(\mathbf{U}_p - \mathbf{U}_c \right) \cdot \mathbf{N}^c_{\underline{p}}, \\
\Pi^c_{\underline{p}} &= P_c - Z_c \left(\mathbf{U}_p - \mathbf{U}_c \right) \cdot \mathbf{N}^c_{\underline{p}}, \\
Z_c &= \rho_c a_c.
\end{aligned}$$

Geometric quantities

Cell area A_c and volume V_c are calculated using the formulae (2.14) and (2.15)

$$\begin{aligned} A_c \mathbf{e}_r &= \sum_{p \in \mathcal{P}(c)} \left(r^c_{\underline{p}} L^c_{\underline{p}} \mathbf{N}^c_{\underline{p}} + r^c_{\overline{p}} L^c_{\overline{p}} \mathbf{N}^c_{\overline{p}} \right), \\ V_c &= \frac{1}{6} \sum_{p \in \mathcal{P}(c)} \left(r_p + r_{p^+} \right) \left(z_p r_{p^+} - r_p z_{p^+} \right). \end{aligned}$$

Time step

The time step is evaluated using three requirements

$$\begin{aligned} \Delta t^{n+1} &= \min \left(\Delta t_{\text{cfl}}, \Delta t_V, C_t \Delta t^n \right), \\ \Delta t_{\text{cfl}} &= C_{\text{cfl}} \min_c \left(\frac{L^n_c}{a^n_c} \right), \\ \Delta t_V &= C_V \min_c \left[\frac{V^n_c}{\left| \frac{\mathrm{d}}{\mathrm{d}t} V_c(t^n) \right|} \right], \end{aligned}$$

with typical values

$$\begin{aligned} C_t &= 0.01, \\ C_{\text{cfl}} &= 0.25, \\ C_V &= 0.1. \end{aligned}$$

This concludes the description of the cylindrical first-order cell-centred EUCCLHYD numerical scheme.

3 Modeling of laser absorption in plasma

In this section, we shall introduce the mechanisms of propagation and absorption of laser radiation in plasma and provide a comprehensive overview of practically applied approaches to laser plasma simulations. We do not aim to provide an exhausting theoretical decomposition of the subject, since our main concern is the implementation and assessment of a numerical model. For a more extensive description of the underlying physical processes refer to [1].

3.1 Propagation of radiation in plasma

Since our main task will eventually be modeling plasma absorption on critical surface, as discussed for example in [14], we will first mention the underlying physical argument for this model. The simplest approach for description of propagation of radiation in plasma is the following.

Let us assume the radiation in question is an electromagnetic field with angular frequency ω, satisfying the dispersion relation

$$\omega^2 = \omega_p^2 + \|\mathbf{k}\|^2 c^2, \tag{3.1}$$

where $\mathbf{k}$ is the wave vector, c phase speed of light and ω_p is the plasma frequency defined by the expression [2]

$$\omega_p = \sqrt{\frac{4\pi e^2 n_e}{m_e}}, \tag{3.2}$$

where n_e is the electron density, e is the electron charge and m_e electron mass. Using (3.1) and (3.2), it is possible to show that there exists a frequency ω, for which the electromagnetic field can no longer propagate. We shall rearrange the dispersion relation (3.1) to demonstrate this claim more clearly. We will isolate the norm of the wave and obtain

$$\|\mathbf{k}\|^2 = \frac{\omega^2 - \omega_p^2}{c^2}.$$

Thus, for a particular frequency $\omega_{\text{crit}} = \omega_p$, the norm of the wave vector can become equal to zero, which means no propagation.

Since we are ultimately going to simulate interaction of laser radiation with a target, it is beneficial to express the radiation using its wavelength λ and not angular frequency ω. Since

$$\omega = 2\pi f = \frac{2\pi c}{\lambda},$$

then, using (3.2) and taking into account, that $\omega_{\text{crit}} = \omega_p$, we arrive at the equation

$$\left(\frac{2\pi c}{\lambda}\right)^2 = \frac{4\pi e^2 n_e^{\text{crit}}}{m_e},$$

where n_e^{crit} is the critical electron density. Now we can finally write the defining expression for critical electron density

$$n_e^{\text{crit}} = \frac{m_e \pi c^2}{e^2 \lambda^2}. \tag{3.3}$$

We have therefore shown that the electromagnetic field propagates only when the electron density of the environment is lower than the critical electron density n_e^{crit} defined by (3.3). Note, that we can also bind the environment's permittivity and frequency or electron density by the means of

$$\varepsilon = 1 - \frac{\omega_p^2}{\omega^2} = 1 - \frac{n_e}{n_e^{\text{crit}}}. \tag{3.4}$$

It is also necessary to point out that the aforementioned process merely reveals the existence of an upper limit of electron density within the material, in which the radiation can propagate, but there is no mention of any absorption processes. For the absorption model we will eventually employ, the exact mechanism of absorption of radiation in the material is completely irrelevant, therefore it is also unnecessary to delve deeper into the physical details of any underlying mechanisms. For an increased degree of understanding concerning the physical processes involved, see [1].

3.2 Design of absorption model

Since we are discussing 2D simulations, the issue of propagation of radiation within the plasma is extremely important and selecting an appropriate propagation model and meticulously establishing its frame of applicability greatly influences the level of realism offered by the numerical simulation. For simplicity, we shall assume that the laser field is represented by an array of mutually independently propagating rays. This assumption is valid for such an environment which does not undergo too rapid changes over time.

There are essentially two approaches to simulating the propagation of radiation in plasma, under the assumptions made above. The first and rather natural way is to solve the ray equation for every ray propagating through the environment. Let us assume that the material which we intend to irradiate is covered by a computational mesh consisting of generally polygonal cells Ω_c. The ray is entering the domain at a certain boundary cell. At every cell edge, we solve the Snell law to determine the next direction of propagation. Such models can also take the permitivity of the material into account, adjusting the speed and direction of propagation even further for increased realism. The absorption of energy carried by the individual rays is then carried out, once the ray reflects from a particular cell. This type of model is quite aptly called *ray tracing*.

A simpler version of a model using rays for the representation of laser radiation assumes that the ray does not change its propagation direction, therefore is a straight line across the domain. It is possible to make such a simplification in such constellation of environment and radiation, where the direction of propagation is colinear with the permitivity gradient of the environment [14]. We assume that this ray is propagating through the material, until it reaches a cell with density, that is higher than critical density, i.e. until it reaches the critical surface. The energy carried by the ray is then deposited in the cell that is hit by the ray. Since it would not be realistic to deposit the entire amount of energy carried by the ray, for the reason that a certain ratio of energy is reflected from the material, this model is accompanied by an empirical coefficient representing the fraction of energy of the radiation, which we allow to absorb.

Our task is to implement the model of absorption on critical surface. We shall do so in the following text.

3.3 Absorption on critical surface

As we remarked before, in this case, the laser radiation is represented by an array of colinear rays, assuming that they propagate in the same direction across the domain. The rays are localized at the beginning of the simulation and their position remains the same throughout the whole simulation. The irradiated material is represented by a disc in 2D cylindrical coordinates, while the laser radiation represented by the rays is assumed to propagate in the negative direction of the z-axis, see Fig. 9 for details regarding the geometry of the situation.

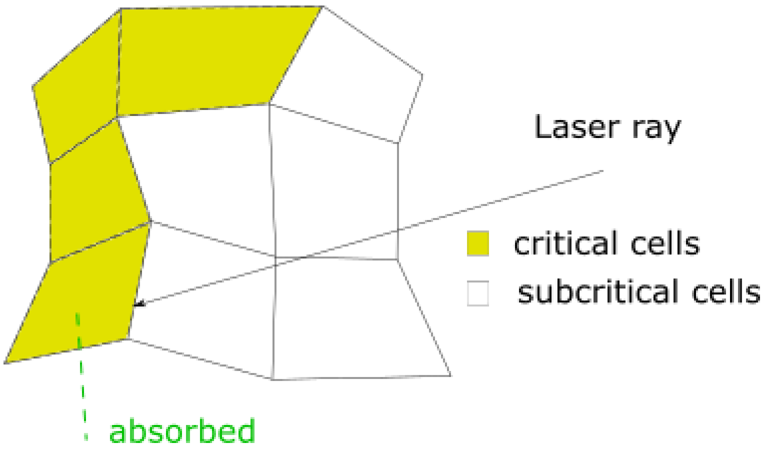


Figure 9: Schematic for absorption on critical surface.

The energy carried by the individual rays is determined by the intensity profile. The main laser profile which we will use, will be the Gaussian profile in time and space

$$I(r,t) = I_{\max} \exp\left[-8\frac{(r-r_s)^2}{d_0^2}\right] \exp\left[-4\ln 2\frac{(t-t_s)^2}{\tau^2}\right], \tag{3.5}$$

where $I_{\max}$ is the maximum intensity of the laser, which can be calculated from the pulse total energy, r_s is the shift in r-coordinate, t_s is the shift in time, d_0 is beam diameter[4] and τ is beam FWHM[5] in time domain.

The ray itself is then characterized by its position on the r-axis r_r, its width d_r, given by its distance from the surrounding rays and its intensity I_r calculated using

[4] defined by the width at $1/e^2$ of peak intensity

[5] Full width at half maximum

the formula (3.5). When the ray hits a cell with density higher than critical density, in other words $\rho_c \geq \rho_{\text{crit}}$, the energy of the ray is deposited in the cell, effectively acting as an energy source term in the form of divergence of intensity, i.e.

$$\nabla \cdot \mathbf{I} \approx \frac{\mathcal{A}}{V_c} \sum_{k=1}^{N} I_k d_k r_k, \tag{3.6}$$

where k denotes the ray index, N is the number of rays hitting cell Ω_c and $\mathcal{A}$ is an additional absorption coefficient, which we employ as a macroscopic parameter, $\mathcal{A} \in (0{,}1)$, roughly simulating the absorption mechanism. The cell internal energy update then, taking into account that the laser absorption takes place after the Lagrangian step, and therefore using the internal energy values at time level t^{n+1}, using (3.6), yields

$$\tilde{\mathcal{E}}_c^{n+1} = \mathcal{E}_c^{n+1} + \mathcal{A} \frac{\Delta t}{V_c} \sum_{k=1}^{N} I_k d_k r_k. \tag{3.7}$$

The equation (3.7) represents the laser absorption in our particular model.

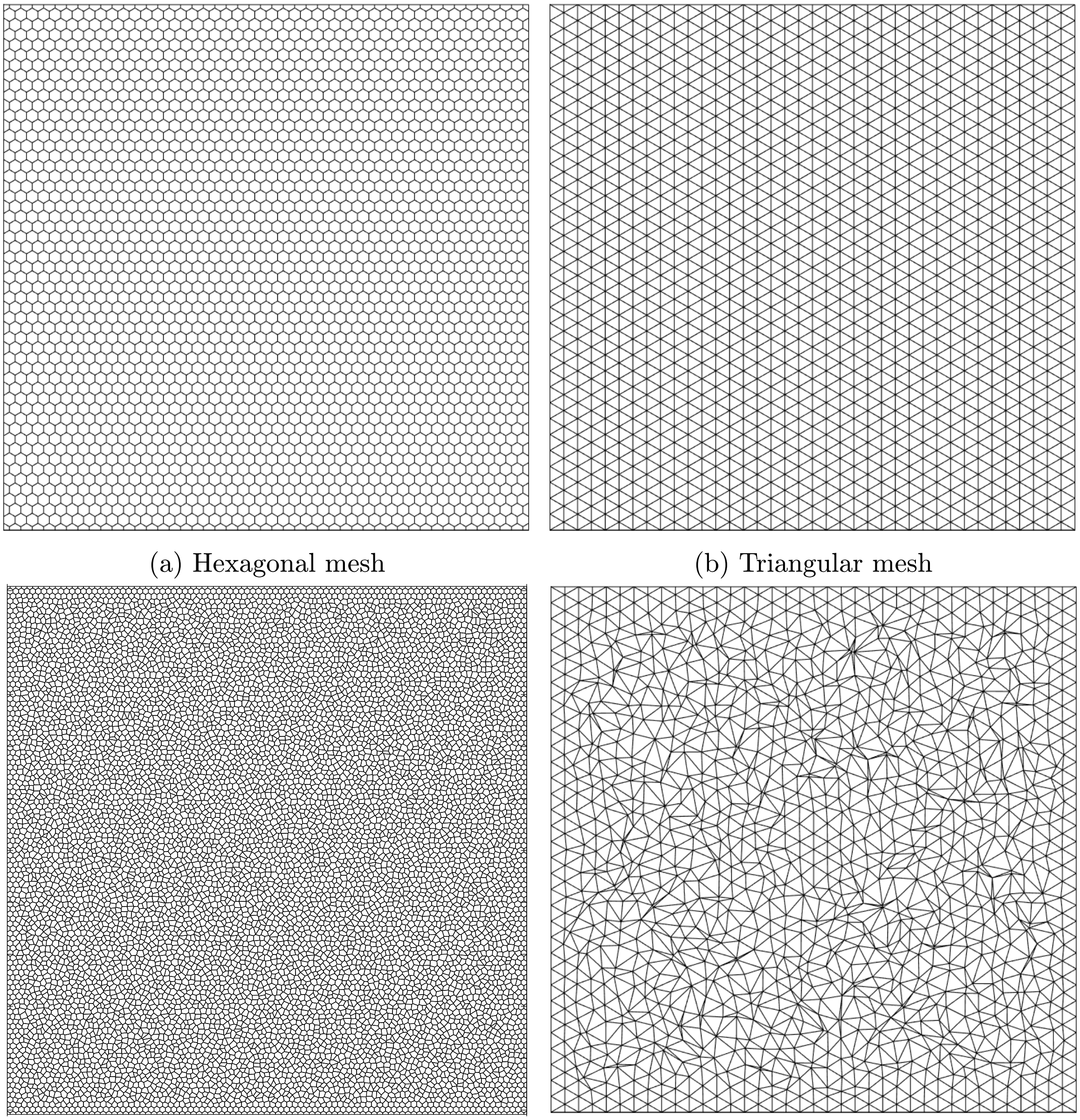
4 CCLUS Lagrangian cell-centred code

Now we shall describe the developed code itself. We refer to the code by the acronym CCLUS (*Cell-Centred Lagrangian Unstructured Suite*). It was conceived as a 2D unstructured code for Lagrangian hydrodynamics and its development began first in [12], with a continuation in [13]. The core feature of the code is its modularity, and its primary use is for research purposes. The sense of the mentioned modularity is mostly to allow interchangeability of different components related to various numerical schemes which are implemented within, to allow testing of various scheme settings, be it multitudes of time step computation methods, interpolation methods, which was extensively discussed in [13], usage of various mesh types and analysis of scheme behaviour related to said mesh types, which was one of the instrumental tools in our article [4], where we analysed the performance of the FLW Cartesian numerical scheme on various generally polygonal and perturbed meshes, and the ability to conduct tests on various industry-standard numerical problems, such as Noh implosion or Sedov explosion tests.

At the current state of implementation, the code includes both Cartesian and cylindrical numerical schemes. The Cartesian schemes include first- and second-order EUCCLHYD numerical schemes, introduced in [9] and a Lax-Wendroff type predictor-corrector type FLW (*Fridrich-Liska-Wendroff*) numerical scheme, introduced in [5]. The second-order Cartesian EUCCLHYD numerical scheme was also implemented in the scope of this thesis. The FLW numerical scheme was additionally extended to include new interpolation methods in the predictor [13]. The cylindrical extension of this code was initialized with the first-order EUCCLHYD numerical scheme [7] in both control volume and area-weighted formulations, see section 2 for its description.

The aforementioned numerical schemes present are implemented in unstructured formulations, therefore it is necessary to provide computational meshes in corresponding unstructured format. The code allows for an import of computational meshes from file, however, the code is also equipped with an unstructured mesh generator, so there is no dependence on an external mesh file. The integrated mesh generator allows for construction of unstructured[6] meshes based on logically rectangular meshes, polar meshes etc., including certain special types of meshes, such as a Cartesian mesh with varying spacing based on geometrical series, which are used in laser plasma absorption problems. We present examples of two meshes which can be used in simulations in Fig. 10, namely a regular mesh consisting of hexagons Fig. 10a, a regular mesh consisting of triangles Fig. 10b, an irregular mesh consisting of general polygons Fig. 10c and an irregular mesh consisting of general triangles Fig. 10d.

[6] In terms of representation.



(a) Hexagonal mesh

(b) Triangular mesh

(c) Polygonal mesh

(d) Irregular triangular mesh

Figure 10: Examples of meshes in unstructured format usable in simulations.

The modularity also manifests itself in aspects such as ease of extension in various ways. For example, although the code in its current state uses an equation of state for ideal gas for thermodynamic closure of the system, it is evaluated by calling separate routines, rather than hard coded. Therefore it is possible to merely change the function for equation of state to incorporate a new one. This theme is maintained throughout the whole code, so there is an opportunity for quick reformulation, should the simulated physical model require it.

4.1 Practical remarks

We shall now discuss some practical aspects regarding the implementation, which we were already foreshadowing before. During the whole process, we have established that while certain things seem theoretically obvious and easy, it is usually extremely helpful to properly document any difficulties encountered, so it may be at immediate disposal for future reference.

4.1.1 Code structure

For a general overview of the basic functionality of the code, see Fig. 11 below.

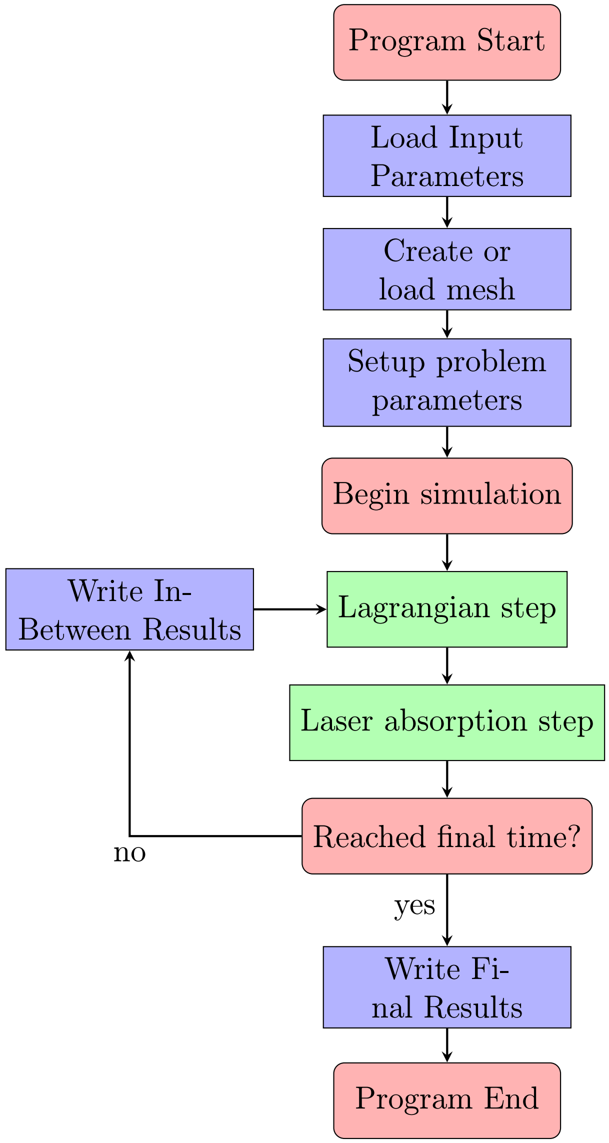


Figure 11: Flowchart of the CCLUS Lagrangian unstructured code.

4.1.2 Symmetrical boundary conditions

One of the crucial aspects of cell-centered schemes are the boundary conditions. While it may seem rather trivial at first glance, their implementation can become difficult in certain situations. In case of the first-order cylindrical EUCCLHYD scheme, the boundary conditions manifest themselves during the computation of the nodal velocities as part of the nodal solver (2.28). While we have treated any imaginable boundary condition theoretically using the expressions (2.31), (2.32) and (2.36), their compuation in practice is not always so straightforward. Since we are now operating in the 2D cylindrical coordinates, the bottom boundary is located on the z-axis, where we typically prescribe the zero normal velocity boundary condition. Similar situation takes place on the left boundary, which is located on the r-axis. These two boundary conditions, the bottom and left zero velocity boundary condition, can generate certain problems in the implementation.

In certain types of problems, mesh nodes situated on both mentioned boundaries should only move in the direction of the boundary axis, and they should never deviate from the axes themselves. For example, if we blindly attempt to solve the linear system (2.28) for the bottom boundary, because the radius of the boundary nodes is $r_p = 0$, the matrix of the system (2.28) becomes singular. Therefore the bottom boundary demands separate treatment. Since we remarked that the nodes on the bottom boundary are not allowed to deviate from the z-axis, we can set the r-velocity component identically to zero, in other words

$$u_p = 0.$$

Thus, the nodal solver for the bottom boundary nodes assumes the form of an algebraic equation for the z-velocity component

$$v_p = \frac{(\mathbf{F}_p)_z}{(\mathsf{M}_p)_{zz}}, \tag{4.1}$$

where $(\mathbf{F}_p)_z$ is the z-component of the subzonal force (2.26) and $(\mathsf{M}_p)_{zz} = (\mathsf{M}_p)_{1,1}$ is the element from the first row and first column of the nodal matrix (2.29). We stress the fact that this replacement is a necessity in terms of successful implementation.

For the left boundary, the linear system (2.28) is solvable, unlike for the bottom boundary. However, it is still sometimes beneficial to fix the nodes on the r-axis as well, since it is in line with the assumptions made when simulating certain types of problems. The treatment for this boundary is similar to (4.1), we merely switch the components

$$v_p = 0, \qquad u_p = \frac{(\mathbf{F}_p)_r}{(\mathsf{M}_p)_{rr}}. \tag{4.2}$$

For the other boundaries, right and top, the generic formula for the nodal solver (2.28) is perfectly applicable under all circumstances.

As a side note, the treatment of boundary conditions by the means of formulating a boundary force contribution in the form of (2.30) is not the only possible approach. There also is the possibility of constructing an array of cells located outside the boundary, named *ghost cells*, outfitting them with desired parameters constituting the boundary conditions and then applying the formula for internal nodes (2.28) for all nodes including the boundary nodes. This method for boundary condition treatment is theoretically equivalent to the boundary force approach, however, once again, the complexity in terms of implementation varies. We do not use ghost cells for boundary condition treatment in this code.

5 Numerical results

In this section, we will display the results obtained from simulations by our code in the cylindrical extension. First we conduct tests on three well-known test problems, the Sod test problem, Noh implosion problem and Sedov blast wave problem. Ultimately we shall present the results of laser absorption on critical surface simulation. For the aforementioned widely used test problems, we will adhere to the convention of displaying dimensions and field variables as dimensionless quantities. For laser absorption problems, we will use the CGS system of units.

5.1 Spherical Sod test problem

For the testing of the first-order cylindrical schemes, we will employ the extension of the well-known Sod shock tube test [11] into spherical geometry. This extension consists of a spherical shock tube of unit radius, where the interface between the two states is located at $r = 0.5$. The initial conditions are two states, we shall remain calling them *left* for the state closer to the origin and *right* for the outer state. The left state is physically a high pressure fluid defined by

$$\begin{pmatrix} \rho_L \\ P_L \\ \mathbf{U}_L \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \mathbf{0} \end{pmatrix},$$

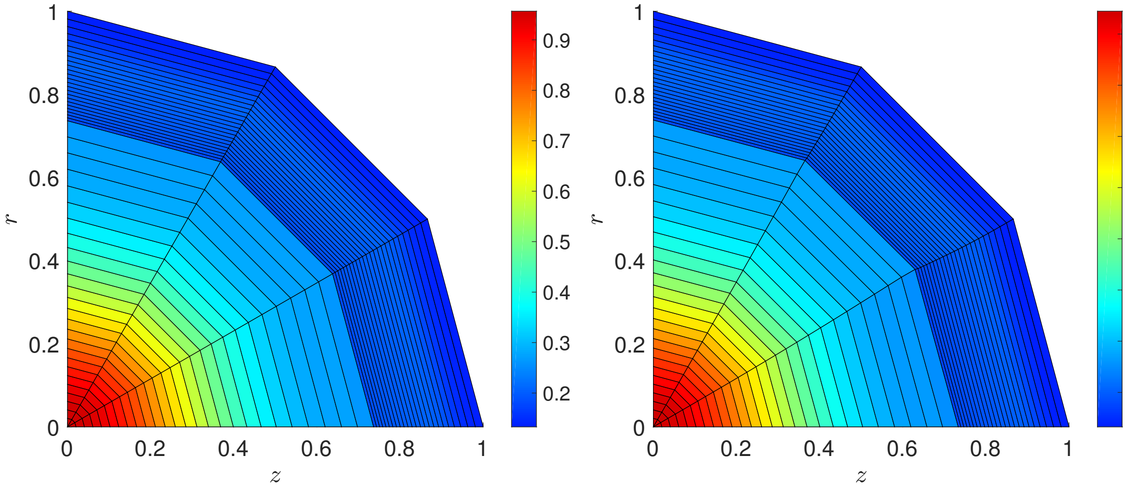
while the right state is a low pressure fluid with the parameters

$$\begin{pmatrix} \rho_R \\ P_R \\ \mathbf{U}_R \end{pmatrix} = \begin{pmatrix} 0.125 \\ 0.1 \\ \mathbf{0} \end{pmatrix}.$$

The Poisson constant in the equation of state (2.40) for this problem is $\gamma = 7/5$. The initial computational domain is defined in polar coordinates by $(r, \varphi) = [0,1] \times \left[0, \frac{\pi}{2}\right]$ since it is run by default on an unstructured polar mesh. The primary goal when attempting to simulate this test is to assess symmetry preservation of the used numerical scheme, be it control volume or area-weighted variant. The final time of the simulation is set at $t_{\text{fin}} = 0.2$. The solution is a point-symmetric shock-wave emerging from the interface between the two states.

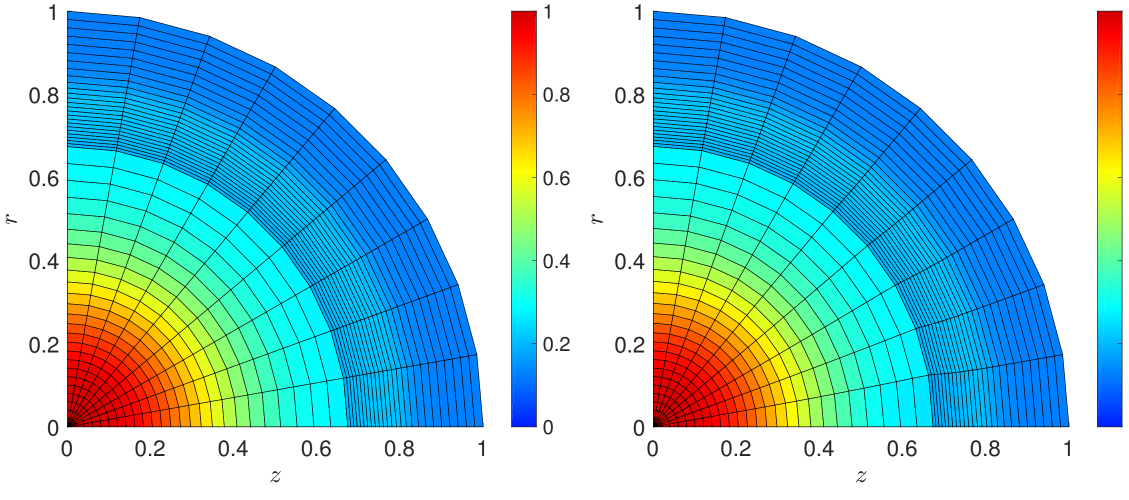
We present the solution for density in Figs. 12 - 16, on polar meshes of various dimensions. As we stated above, our primary concern regarding the solution of this problem is the assessment of the level of symmetry preservation by the numerical scheme.

It is fairly well visible that the number of cells in the angular direction influences symmetry preservation, while the number of cells in the radial direction is the primary factor in terms of accuracy and convergence towards the reference solution. The reference solution used for validation in Fig. 16 was obtained by a calculation using a 1D spherical Lagrangian code, using a mesh constising of 10 000 cells. As far as symmetry is concerned, it is evident, based on the solution displayed in the scatter plot in Fig. 16b, that the control volume scheme violates symmetry, whereas the area-weighted scheme in Fig. 16a preserves it.



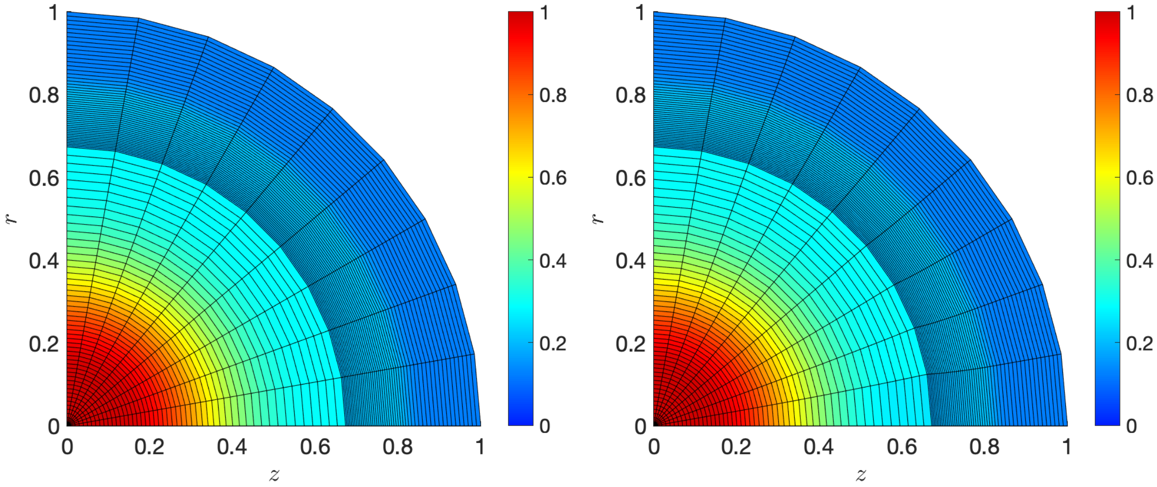
(a) Area-weighted (b) Control volume

Figure 12: Density colour plot for the Spherical Sod test on polar mesh with 3x50 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted (b) Control volume

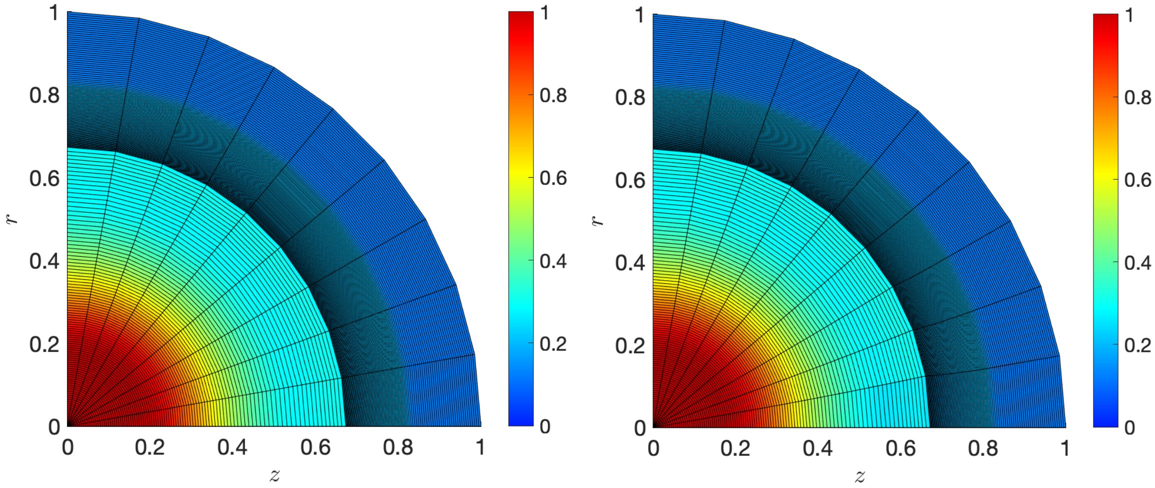
Figure 13: Density colour plot for the Spherical Sod test on polar mesh with 9x50 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

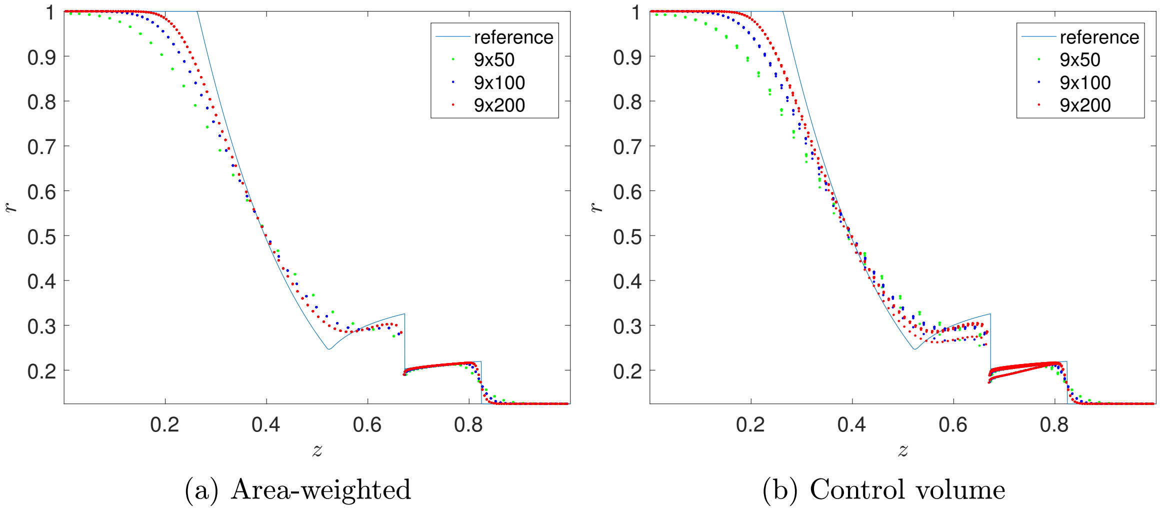
Figure 14: Density colour plot for the Spherical Sod test on polar mesh with 9x100 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

Figure 15: Density colour plot for the Spherical Sod test on polar mesh with 9x200 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

Figure 16: Density scatter plots for the Spherical Sod test on polar meshes with increasing number of cells in the radial direction. Computed using area-weighted and control volume schemes.

5.2 Noh test

The Noh test problem [10] is a classical hydrodynamic test developed to verify proper conversion of kinetic energy into internal energy, essentially validating the used Lagrangian scheme in the presence of strong shock waves. It is physically an implosion at the origin, where the initial conditions are defined by the expression

$$\begin{pmatrix} \rho_0 \\ P_0 \\ \mathbf{U}_0 \\ \mathcal{E}_0 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 10^{-6} \\ -\frac{\mathbf{X}_c}{\|\mathbf{X}_c\|} \\ \frac{P_0}{\rho_0(\gamma-1)} \end{pmatrix}, \tag{5.1}$$

where we set the Poisson constant to $\gamma = \frac{5}{3}$, which corresponds to monoatomic gas. The initial computational domain is $[0,1] \times [0,1]$ and the final time of simulation is $t_{\text{fin}} = 0.6$. The solution of the Noh problem is an outward propagating shock wave with constant velocity $U = 1/3$, where the density region inside the shock is constant with the value $\rho_{\text{in}} = 64$, while density in the region outside the shock has the profile $\rho_{\text{out}} = \left(1 - \frac{t}{r}\right)^2$.

We are displaying the solution for density in Figs. 17 - 27. When we examine the solution computed using Cartesian meshes in Figs. 17 - 21, it is possible to notice the apparent influence of whether the control volume or area-weighted variant of the scheme is used. The trend in Fig. 19 and Fig. 21 shows that the solution obtained by the area-weighted variant preserves symmetry much better than the conrol volume variant, however, the plateau in the region behind the shock wave is slightly lower than in case of the control volume scheme. The distortion in the direction of the diagonal, most noticeable in Fig. 20, is consistent with the results obtained with the Cartesian variant of the first-order EUCCLHYD scheme [12].

The issue of symmetry preservation is much better shown on the results obtained using polar meshed in Figs. 22 - 27, where the violation of symmetry by the control volume variant is evident at first glance.

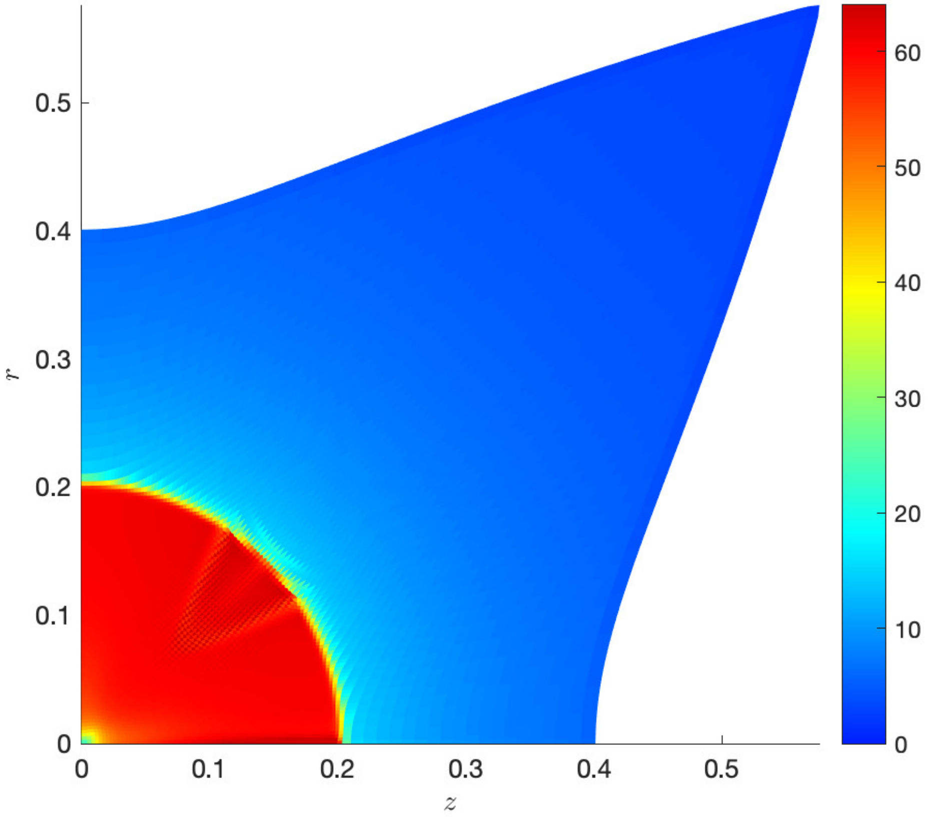
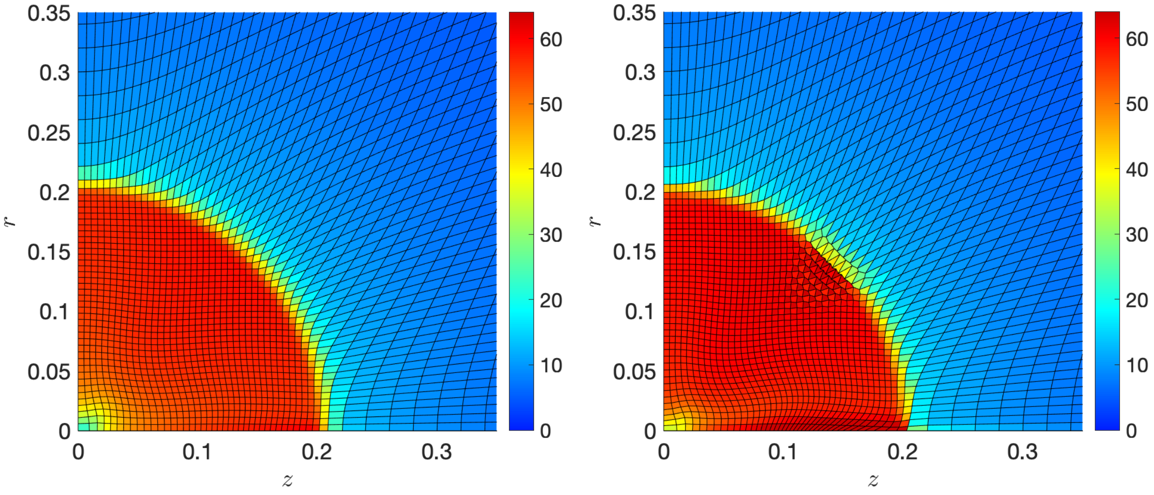


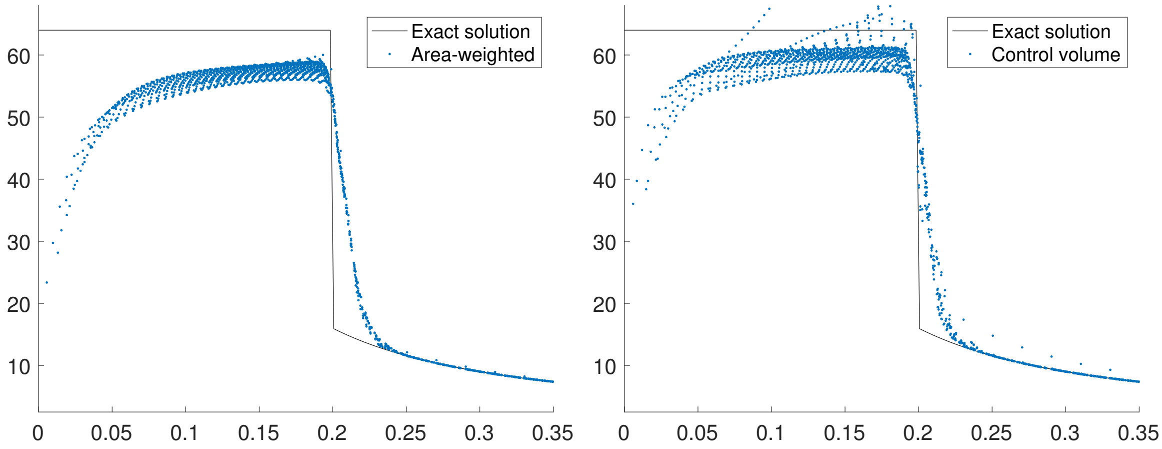
Figure 17: Density colour plot for the Noh test on Cartesian mesh with 100x100 cells. Computed using area-weighted scheme. Whole computational mesh displayed.



(a) Area-weighted

(b) Control volume

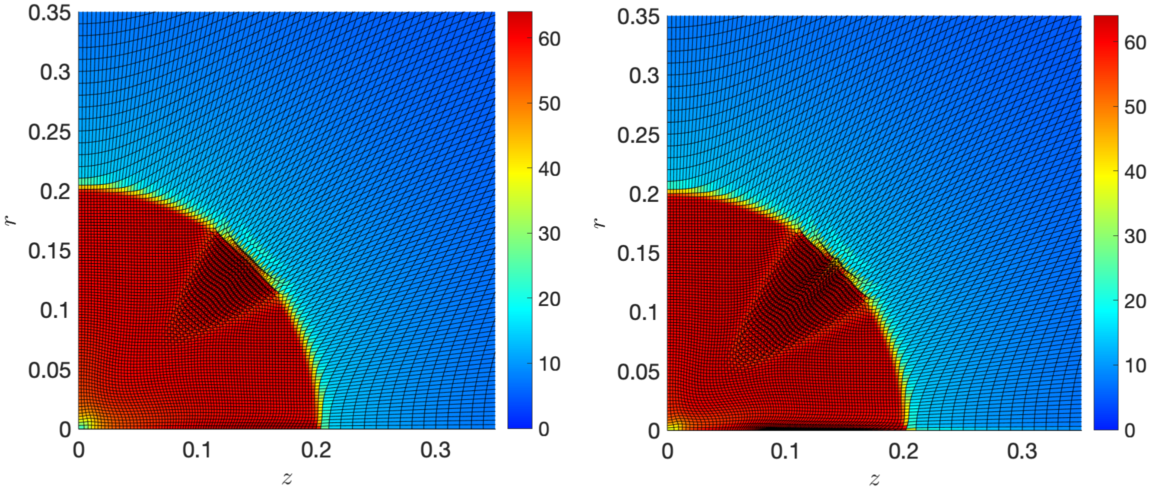
Figure 18: Density colour plot for the Noh test on Cartesian mesh with 50x50 cells. Detail of the region where the shock wave is located. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

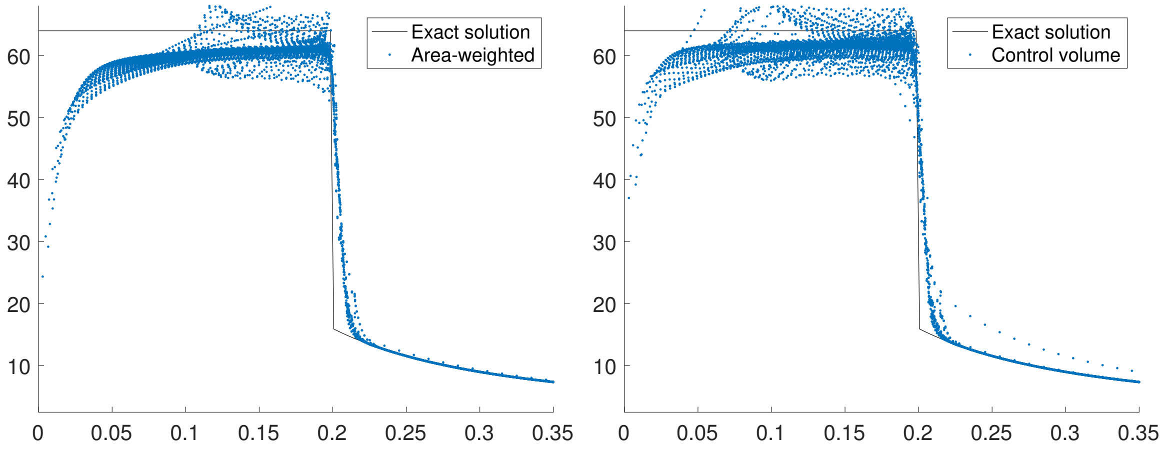
Figure 19: Density scatter plot for Noh test on Cartesian mesh with 50x50 cells. Detail of the region where the shock wave is located. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

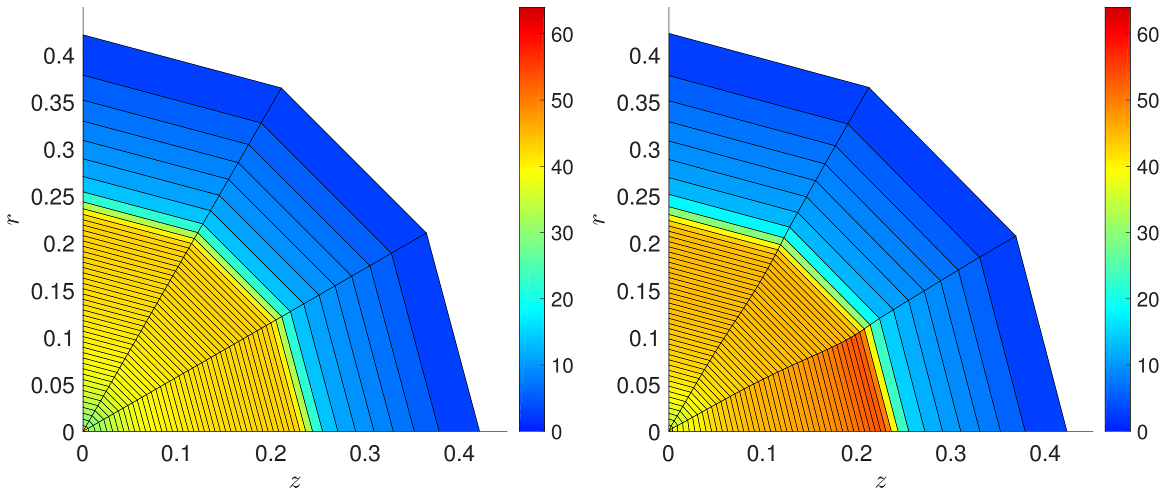
Figure 20: Density colour plot for the Noh test on Cartesian mesh with 100x100 cells. Detail of the region where the shock wave is located. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

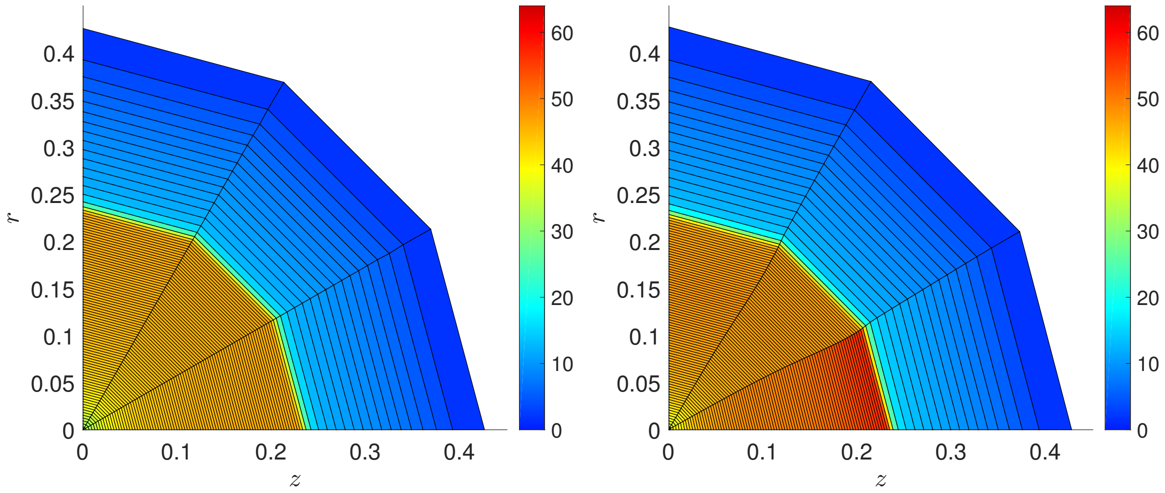
Figure 21: Density scatter plot for Noh test on Cartesian mesh with 50x50 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

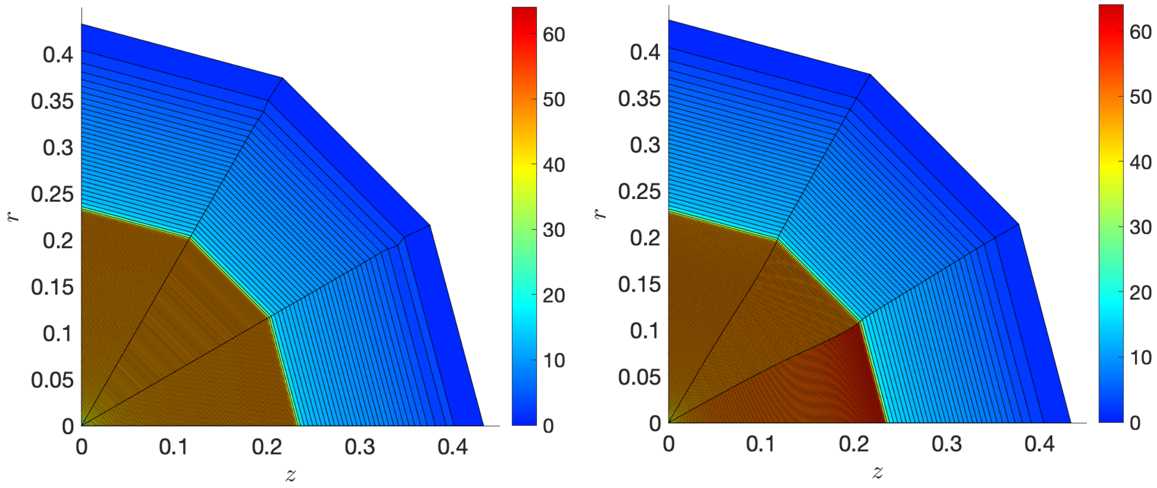
Figure 22: Density colour plot for the Noh test on polar mesh with 3x50 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

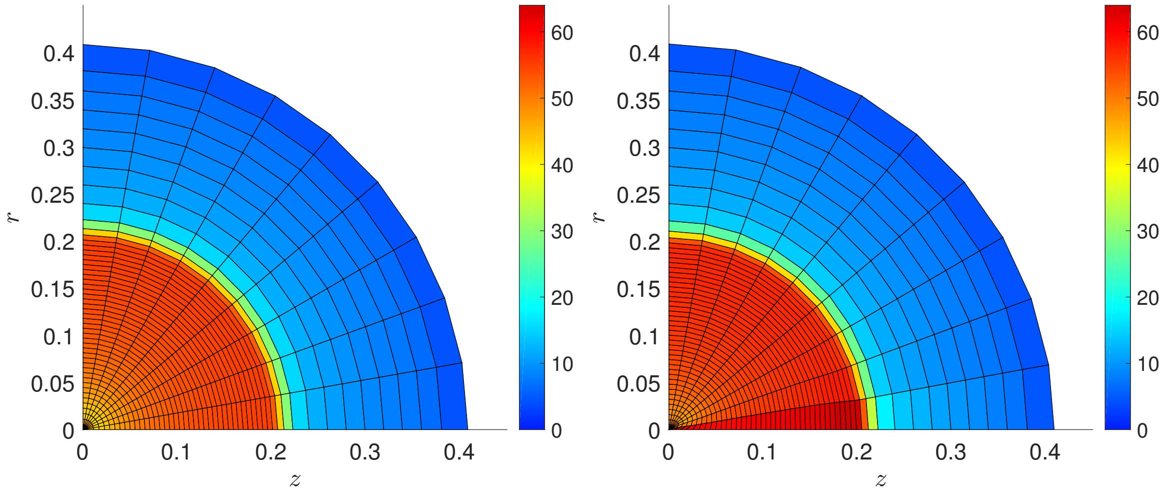
Figure 23: Density colour plot for the Noh test on polar mesh with 3x100 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

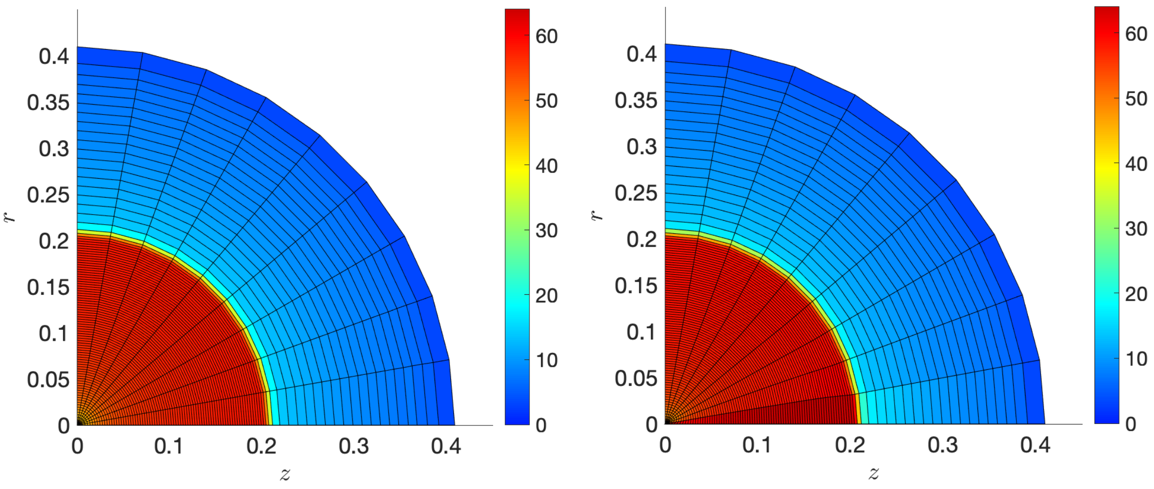
Figure 24: Density colour plot for the Noh test on polar mesh with 3x200 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

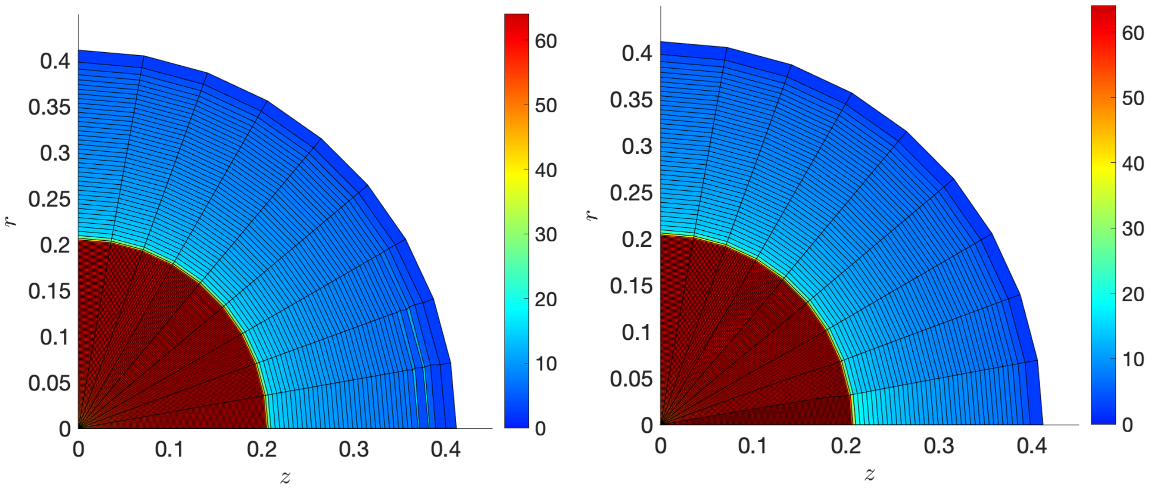
Figure 25: Density colour plot for the Noh test on polar mesh with 9x50 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

Figure 26: Density colour plot for the Noh test on polar mesh with 9x100 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

Figure 27: Density colour plot for the Noh test on polar mesh with 9x200 cells. Computed using area-weighted and control volume schemes.

5.3 Sedov test

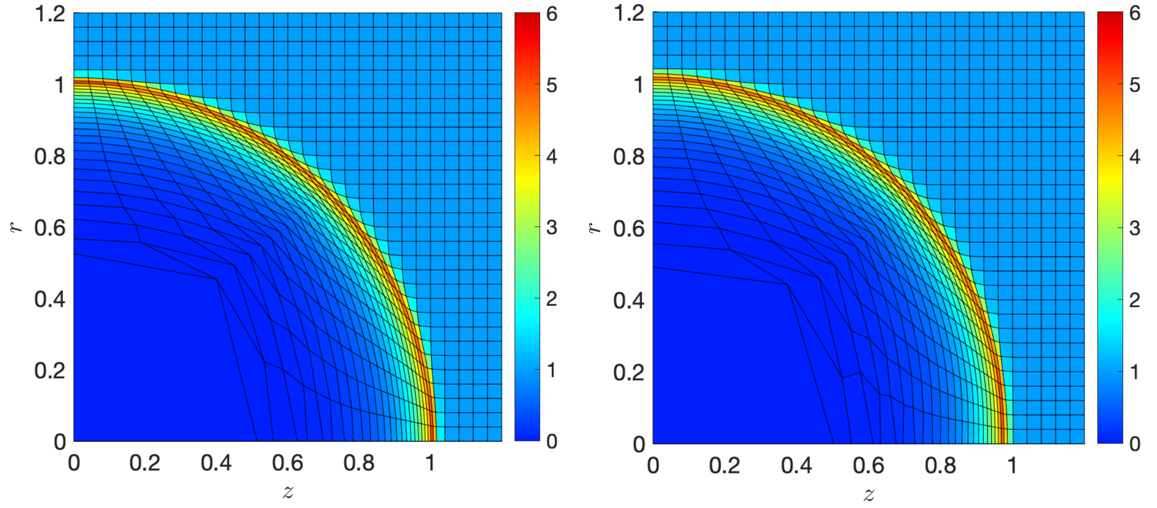
The Sedov test is also a ubiquitously used hydrodynamic test, which represents an explosion at the origin followed by a spherical shock wave propagating in the outward direction. The initial conditions are determined by inserting the total system energy into the cells in the epicentre. The epicentre is defined as a neighbourhood of the origin point $\mathcal{U}(\mathbf{0})$. Therefore, the initial conditions can be expressed mathematically as

$$\begin{aligned}
\begin{pmatrix}\rho_0\\ P_0\\ \mathbf{U}_0\end{pmatrix} &= \begin{pmatrix}1.0\\ 10^{-6}\\ \mathbf{0}\end{pmatrix}, && \mathbf{X}_c \notin \mathcal{U}(\mathbf{0}),\\
\begin{pmatrix}\rho_0\\ P_0\\ \mathbf{U}_0\end{pmatrix} &= \begin{pmatrix}1.0\\ \rho_0(\gamma-1)\frac{E_{\text{tot}}}{\sum_{c\in\mathcal{U}} V_c}\\ \mathbf{0}\end{pmatrix}, && \mathbf{X}_c \in \mathcal{U}(\mathbf{0}),
\end{aligned} \tag{5.2}$$

where again we set $\gamma = \frac{7}{5}$ and $E_{tot} = 0.425536$. To avoid confusion, this energy value represents total energy inserted into the 3D physical domain. However, we must not forget, that all variables are defined per unit radian, and therefore the actual numerical value we store in the epicentre cells is $0.425536/2\pi$. We use the initial computational domain of size $[0.0, 1.2] \times [0.0, 1.2]$.

The solution is an outward propagating blast wave, reaching the peak of $\rho = 6$ situated at $R = \sqrt{z^2 + r^2} = 1$ at time $t = 1$.

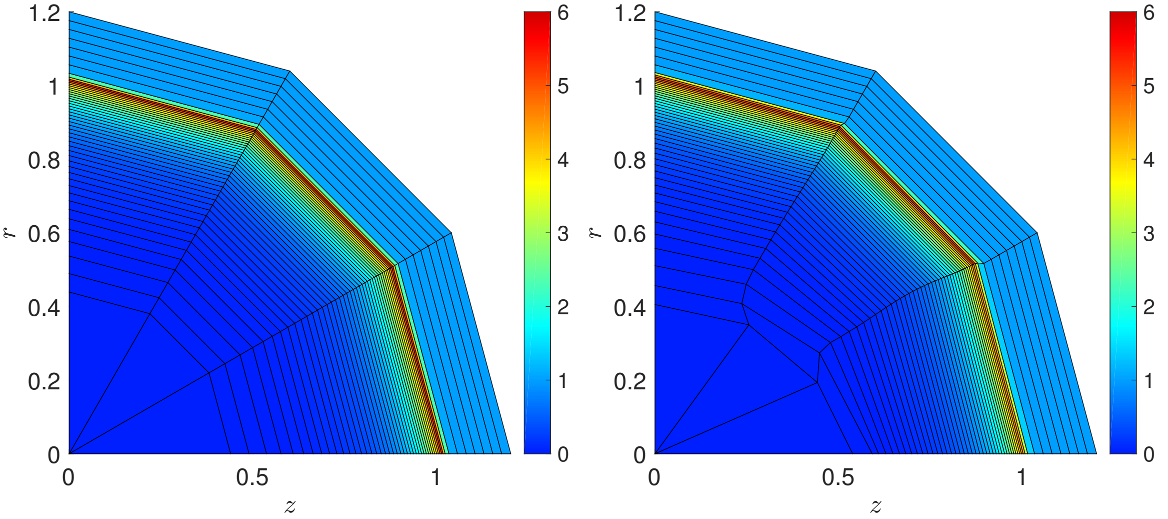
The solution for the Sedov blast wave problem is displayed in Figs. 28 - 34. It is immediately noticeable, for example in Fig. 28 or Fig. 32, that the solution obtained by the area-weighted variant is symmetrical, whereas the solution procured by the control volume variant is not, if we focus on the shock position in the vicinity of the z-axis. The loss of symmetry is also quite apparent in Fig. 34a, where we display the result of computation on 30x30 Cartesian mesh using both schemes. As we can see, the solution obtained by the control volume scheme is significantly more dispersed than the area-weighted solution. Furthermore, the mesh during the computation using the control volume scheme is subject to some intense deformation. This is well distinguishable in Fig. 29. In addition, if we examine the solution by the area-weighted scheme in Figs. 34b - 34c, it is also possible to observe the improving shock resolution with increasing mesh fineness.



(a) Area-weighted

(b) Control volume

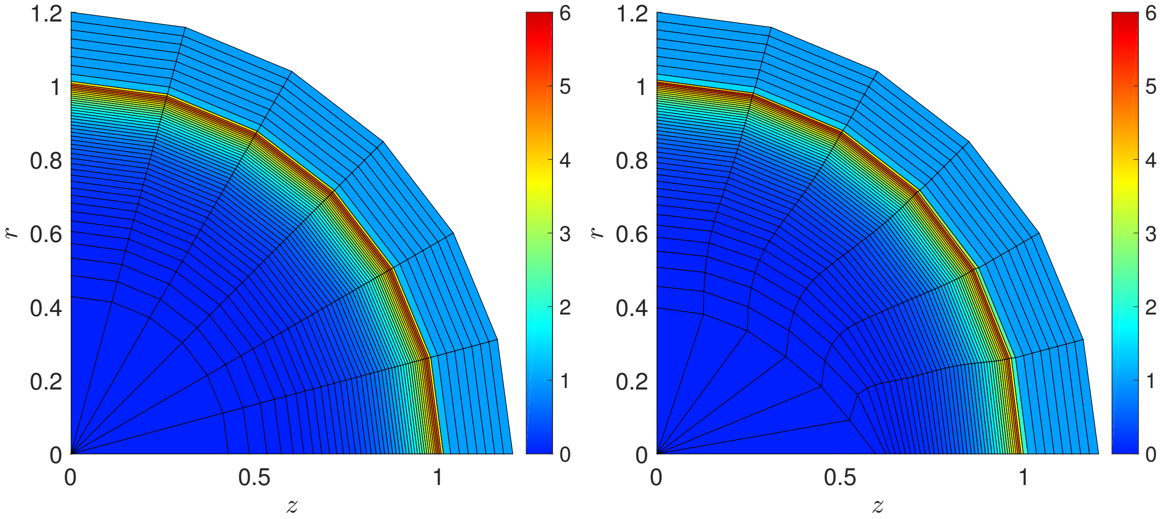
Figure 28: Density colour plot for the Sedov test on Cartesian mesh with 30x30 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

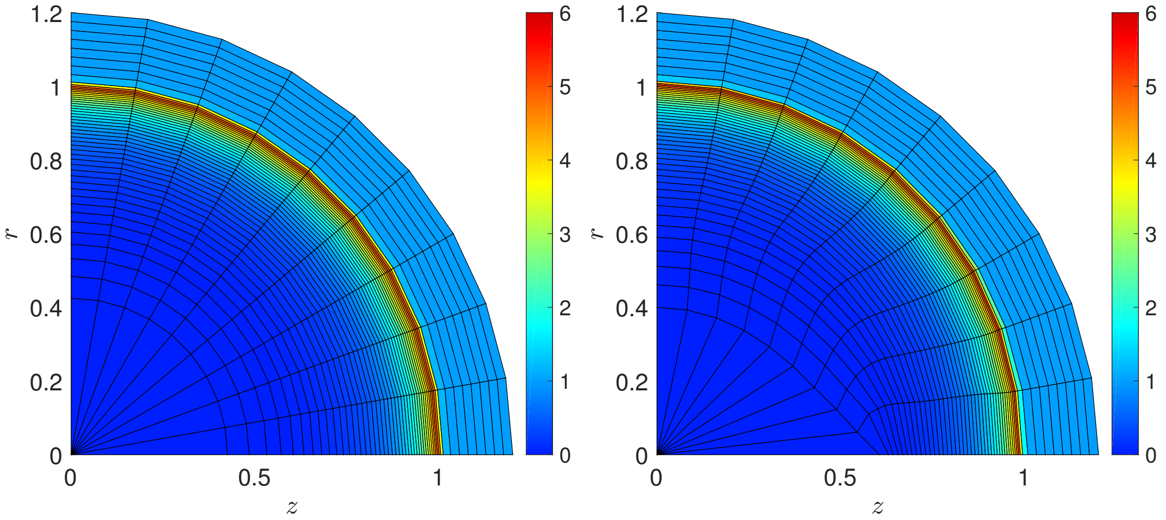
Figure 29: Density colour plot for the Sedov test on polar mesh with 3x50 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

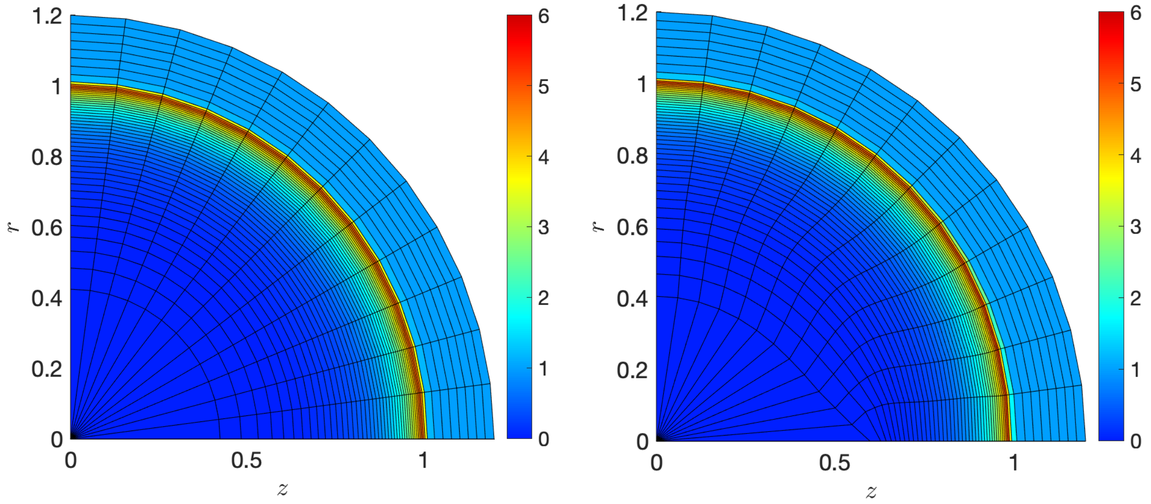
Figure 30: Density colour plot for the Sedov test on polar mesh with 6x50 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

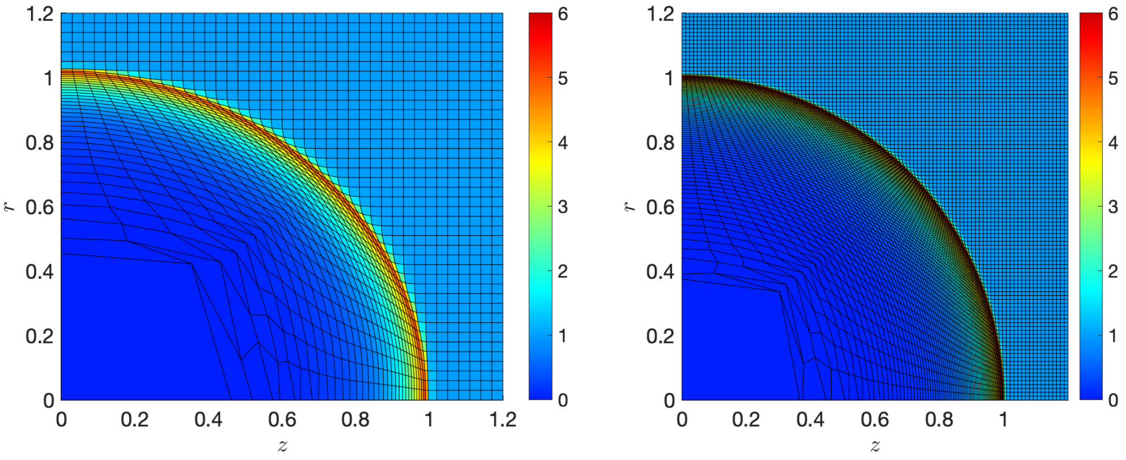
Figure 31: Density colour plot for the Sedov test on polar mesh with 9x50 cells. Computed using area-weighted and control volume schemes.



(a) Area-weighted

(b) Control volume

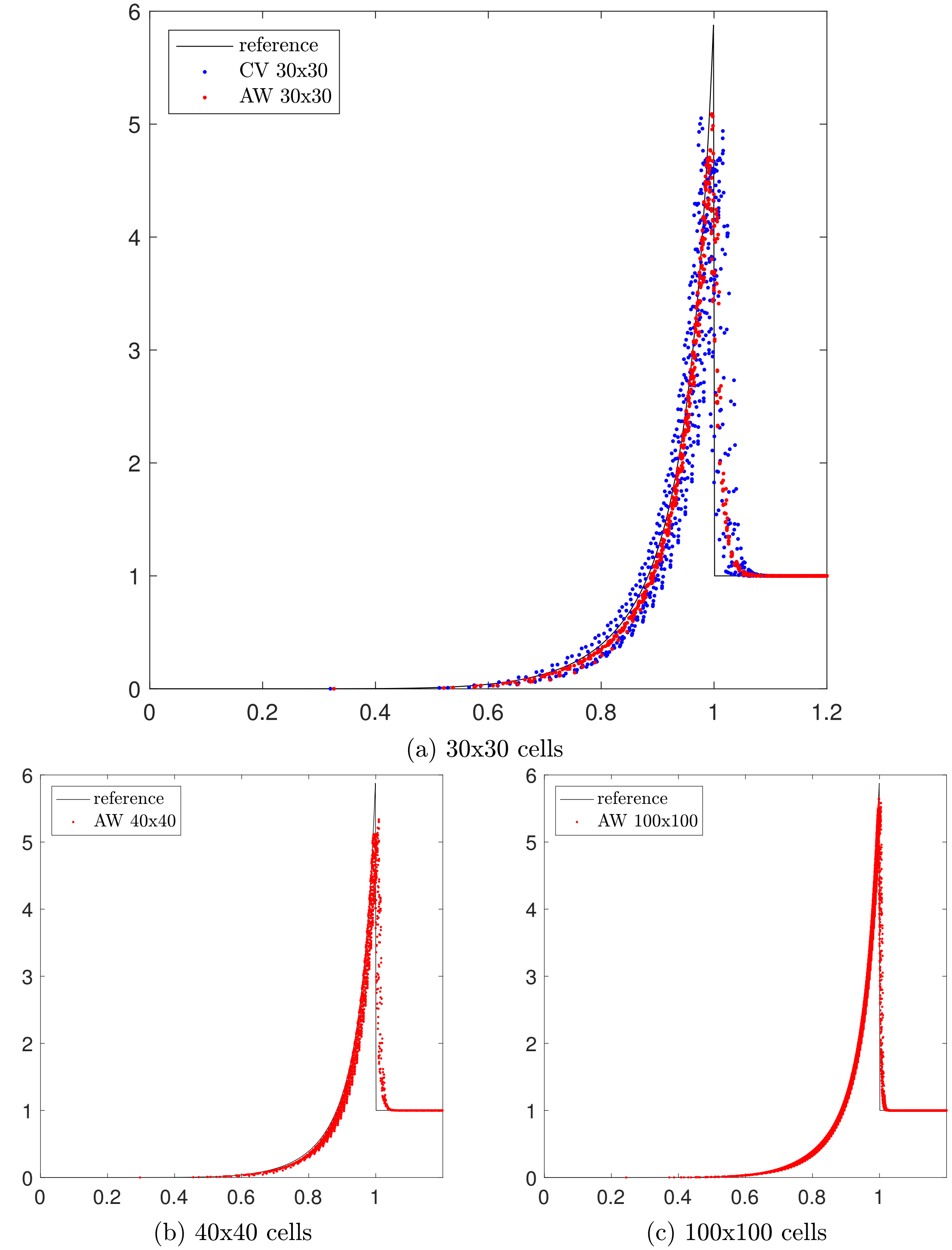
Figure 32: Density colour plot for the Sedov test on polar mesh with 12x50 cells. Computed using area-weighted and control volume schemes.



(a) 40x40 cells

(b) 100x100 cells

Figure 33: Density colour plot for the Sedov test on Cartesian meshes computed using the area-weighted scheme.



(a) 30x30 cells

(b) 40x40 cells

(c) 100x100 cells

Figure 34: Density scatter plots for the Sedov test on Cartesian meshes computed using the area-weighted and control volume schemes.

5.4 Laser absorption

We simulate a very simple laser absorption problem, with parameters inspired from the interaction of laser radiation from the third harmonic frequency of the PALS laser, $\lambda = 438\,\mathrm{nm}$, with an Aluminium target, used as one of the basic absorption problems in the PALE code [14]. All values concerning the laser absorption simulation will be displayed in the CGS (*centimeter-gram-second*) system of units. The initial mesh constitutes the Aluminium target with parameters in Tab. 1.

ρ [g/cm^2]	P [Ba]	$\mathbf{U}$ [cm/s]
2.7	$4 \cdot 10^{10}$	$\mathbf{0}$

Table 1: Table with initial parameters for every cell in the computational domain.

The pressure value $P = 4 \cdot 10^{10}$ Ba corresponds to room temperature. Assuming a Gaussian profile (3.5), the maximum laser intensity is

$$I_{\max} = 2 \cdot 10^{22}\ \mathrm{erg} \cdot \mathrm{s}^{-1} \cdot \mathrm{cm}^{-2},$$

with focal spot of the radius

$$r_0 = 100\,\mu\mathrm{m},$$

and FWHM of

$$\tau = 800\,\mathrm{ps}.$$

The critical density corresponding to these parameters has the value

$$\rho_{\mathrm{crit}} = 0.02\,\mathrm{g/cm}^2,$$

and we set the absorption coefficient from (3.7) as $\mathcal{A} = 0.75$. Because the simulation is extremely time-consuming, coupled with the fact that our code is working with generally unstructured meshes, thus theoretically slower than structured codes such as PALE [14] from the perspective of computational speed and efectiveness, we have decided to select the final time

$$t_{\mathrm{fin}} = 400\,\mathrm{ps},$$

in other words, we terminate the simulation when the laser radiation reaches peak intensity. We are using an Aluminium foil with the dimensions $20\,\mu\mathrm{m} \times 500\,\mu\mathrm{m}$ as the target, which is also the initial computational domain. The laser radiation is represented in the form of rays colinear with the z-axis, while we distribute the rays in such manner that there are 6 rays hitting each cell on the right boundary of the target.

The computational mesh used during the simulation is a rectangular mesh of the size 40x100 with geometric factors $q_z = 0.85$ and $q_r = 1.01$ and is generated using the formula for cell dimensions

$$dz = \frac{z_{\max} - z_{\min}}{q_z^{n_z} - 1}, \tag{5.3}$$

$$dr = \frac{r_{\max} - r_{\min}}{q_r^{n_r} - 1}, \tag{5.4}$$

where $r_{\max}$,$z_{\max}$ and $r_{\min}$,$z_{\min}$ are the maximal and minimal coordinate values respectively and n_z, n_r represent the number of cells in respective axes. The geometric factors specify how the cell size decreases ($q < 1$) or increases ($q > 1$) in the positive direction of an axis. Out of many simulations, we shall display a representative set of results below.

The results of the simulation are plotted in Figs. 35-38. In Fig. 36 we can see the progression of the density profile in time until the simulation is terminated at 400 ps. In Fig. 35, we can clearly see how the radiation is evaporating the material, drilling a hole in the process. This is visualized in greater detail in Fig. 38, where we are zooming on the area where the critical surface is localized. We can say that the result we obtained is similar to the simulation results obtained from the PALE code [14], however, we have to keep in mind that we terminated the simulation 400 ps earlier.

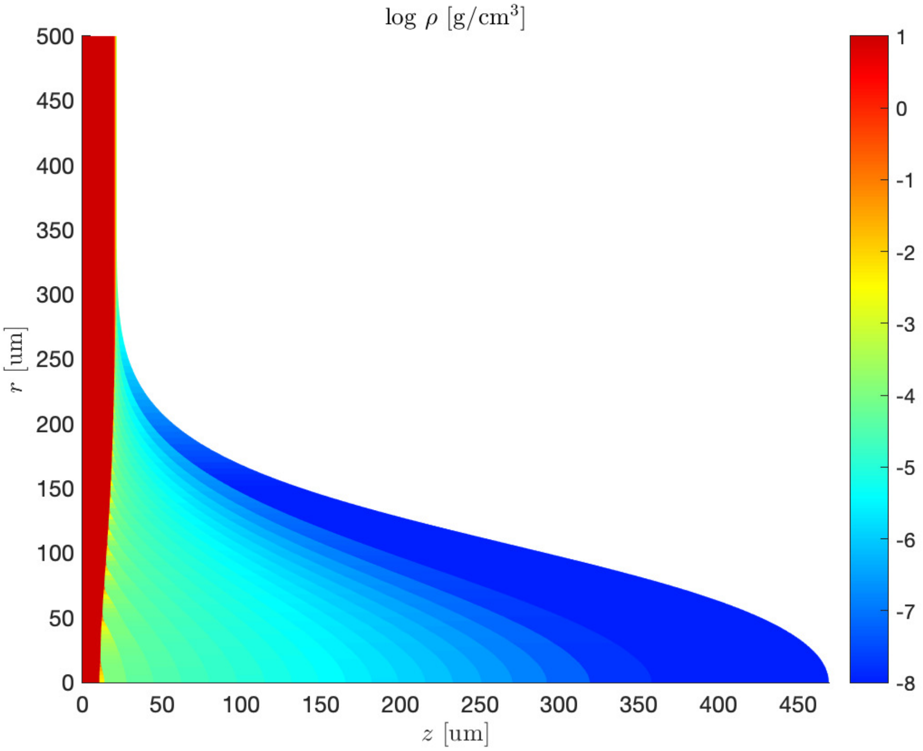
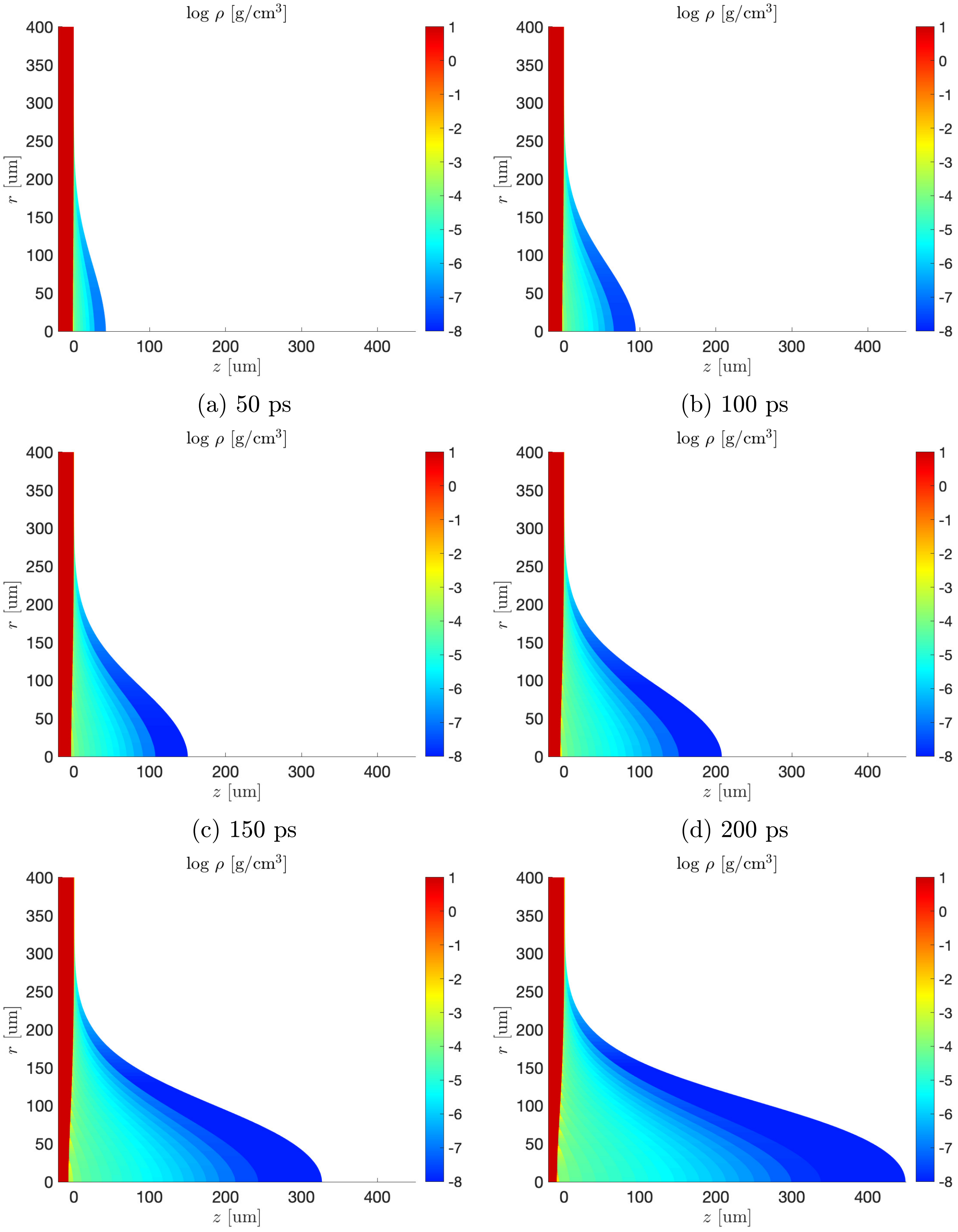


Figure 35: Logarithmic density plot for laser absorption on critical surface at stopping time. Mesh consisting of 40x100 cells, $q_z = 0.85$, $q_r = 1.01$.



(a) 50 ps

(b) 100 ps

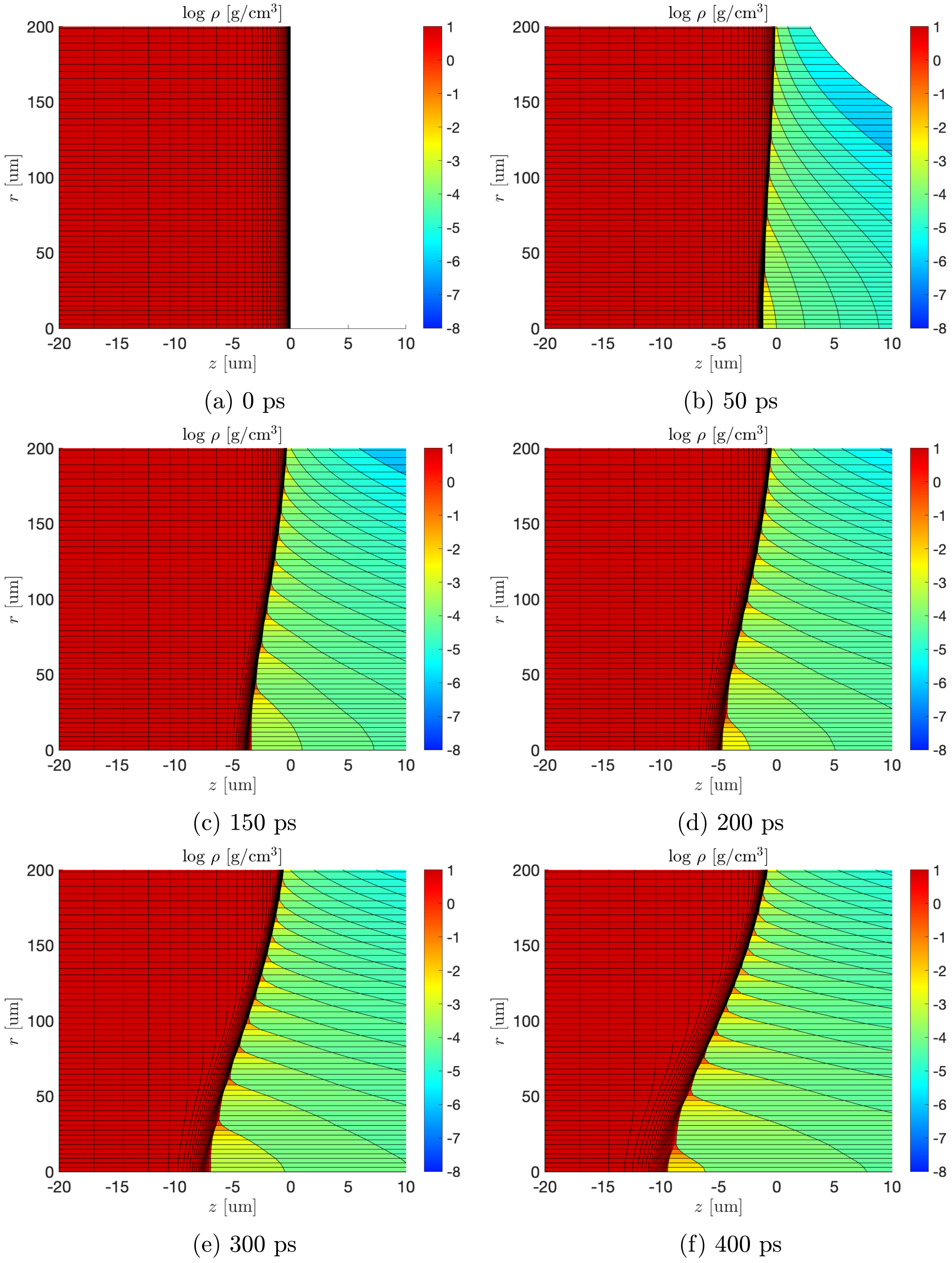
(c) 150 ps

(d) 200 ps

(e) 300 ps

(f) 400 ps

Figure 36: Time evolution of the plasma corona during the laser absorption on critical surface simulation from starting time 0 ps until stopping time 400 ps. Displaying logarithm of density.



(a) 0 ps

(b) 50 ps

(c) 150 ps

(d) 200 ps

(e) 300 ps

(f) 400 ps

Figure 37: Time evolution of the crater during the laser absorption on critical surface simulation from starting time 0 ps until stopping time 400 ps. Displaying logarithm of density.

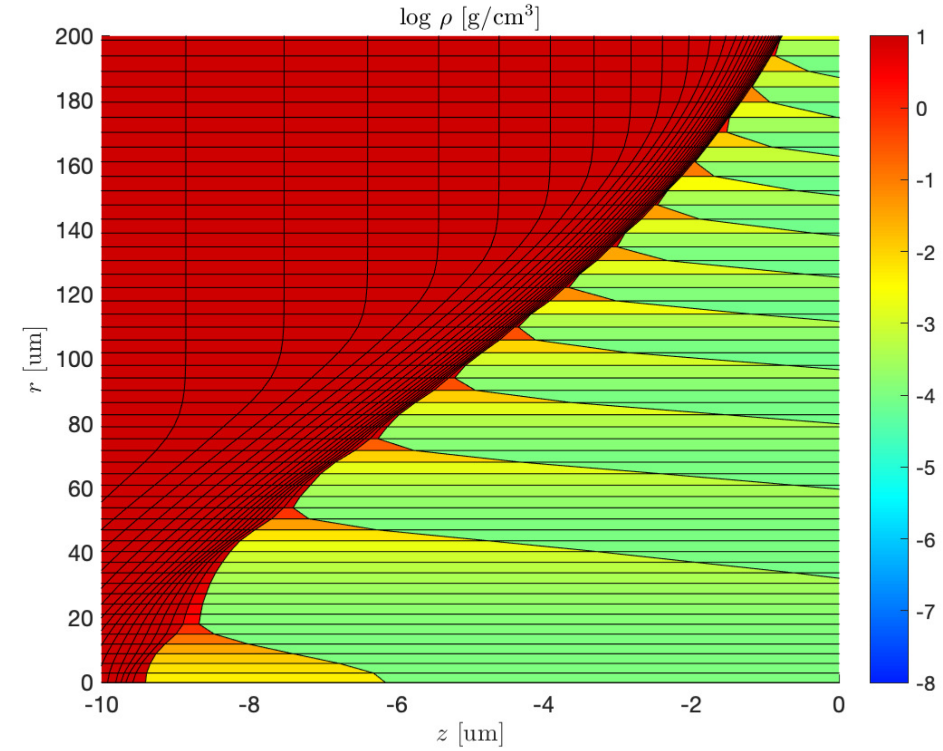


Figure 38: Density plot for laser absorption on critical surface at stopping time. Detail of the region around the critical surface. Displaying logarithm of density.

6 Conclusion and perspectives

The main objectives of this thesis were the extension of the 2D cell-centred unstructured Lagrangian hydrodynamical code into cylindrical coordinates, as a natural continuation of the research and development documented in [12] and [13] and to subsequently devise and implement a simple model for laser absorption on critical surface. We have selected the first-order cylindrical EUCCLHYD numerical scheme for the discretization of the Euler equations for the implementation, in both area-weighted and control volume variants. The scheme and its construction was described in a detailed manner, to elucidate some of the complexities regarding its practical implementation. We have also included a variety of standard numerical tests to assess the attributes of the aforementioned numerical scheme such as symmetry preservation and robustness. The assessment of the first-order control volume and area-weighted variants yields, that the first-order control volume scheme displays significant loss of symmetry preservation, whereas the area-weighted scheme preseves symmetry quite well. This is consistent with the results in [7].

We have subsequently designed and implemented a model for laser absorption on critical surface, intended primarily as a proof-of-concept, that the hydrodynamic code is applicable for laser plasma simulations and can provide results similar to equivalent settings of the PALE code [6], [14], which we are using for reference. Thus, we have fulfilled all requirements of the assignment.

In the current state of affairs, the CCLUS code, as we have decided to call it, is quite an extensive tool for research of Lagrangian cell-centred schemes in unstructured formulation. It contains a variety of implemented schemes, namely the first-order and second-order EUCCLHYD scheme in Cartesian geometry [9], a FLW predictor-corrector [5] scheme in Cartesian geometry and most recently the aforementioned first-order area-weighted and control volume EUCCLHYD scheme in cylindrical geometry. As we have already mentioned in publications [4] or [13], the FLW scheme is also outfitted with several weighting methods for use in its predictor phase for the interpolation of nodal variables. In addition, the code is designed to operate with arbitrary unstructured meshes, provided in the form of node and cell connectivity lists and node positions.

There are countless possibilities regarding its further development. Since the code in its current state is purely Lagrangian, probably the most natural continuation would be its extension into the framework of ALE, incorporating rezoning and remapping into the code. The extension into ALE would be extremely valuable, because this methodology safeguards the simulations against significant mesh deformation, which happens rather often in laser plasma simulations, see the results in [14] for reference.

To increase the degree of sophistication of the physical model, it would also be beneficial to incorporate heat conductivity and additional equations of state, such as QEOS[7], into the model. However, the issue of heat conduction poses certain issues. The code operates with meshes in unstructured format and heat conduction is manifesting itself as an additional parabolic term in the energy equation (1.1d). The system for heat conduction is then solved implicitly. The use of an implicit scheme for variables

[7] quotidian equation of state

defined on generally unstructured meshes is problematic in practical terms, because it results in solution of a system with a sparse matrix, which is extremely ineffective without an appropriate robust solver.

To conclude, the CCLUS code has a lot of potential and can become a useful tool in research of cell-centred Lagrangian schemes and simulations of physical problems where the discretized governing equations are applicable.

References

[1] Stefano Atzeni and Jürgen Meyer-ter Vehn. *The physics of inertial fusion: beam plasma interaction, hydrodynamics, hot dense matter*. OUP Oxford, 2004.

[2] N. N. Demchenko and V. B. Rozanov. A hydrodynamic model of the interaction of picosecond laser pulses with condensed targets. *Journal of Russian Laser Research*, 22(3):228–242, 2001.

[3] J. K. Dukowicz, M. C. Cline, and F. L. Addessio. A general topology Godunov method. *Journal of Computational Physics*, 82(1):29–63, 1989.

[4] D. Fridrich, R. Liska, I. Tarant, P. Váchal, and B. Wendroff. Cell-centered Lagrangian Lax-Wendroff HLL hybrid scheme on unstructured meshes. *Acta Polytechnica*, 61(SI):68–76, 2021.

[5] D. Fridrich, R. Liska, and B. Wendroff. Some cell-centered Lagrangian Lax–Wendroff HLL hybrid schemes. *Journal of Computational Physics*, 326:878–892, 2016.

[6] R. Liska, M. Kuchařík, J. Limpouch, O. Renner, P. Vachal, L. Bednarik, and J. Velechovský. Ale method for simulations of laser-produced plasmas. In *Finite Volumes for Complex Applications VI Problems & Perspectives*, pages 857–873. Springer, 2011.

[7] P.-H. Maire. A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry. *Journal of Computational Physics*, 228(18):6882–6915, 2009.

[8] P.-H. Maire, R. Loubère, and P. Váchal. Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme. *Communications in Computational Physics*, 10(4):940–978, 2011.

[9] Pierre-Henri Maire. *Contribution to the numerical modeling of inertial confinement fusion.* Habilitation thesis, Université Bordeaux I, 2011.

[10] W. F. Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Computational Physics*, 72(1):78–120, 1987.

[11] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational physics*, 27(1):1–31, 1978.

[12] I. Tarant. Multidimensional numerical cell-centered Lagrangian methods. Bachelor's thesis, Czech Technical University in Prague, 2020.

[13] I. Tarant. Lagrangian Lax-Wendroff methods for hydrodynamics. Research project, Czech Technical University in Prague, 2021.

[14] J. Velechovský. Modelování absorpce laserového záření v plazmatu. Research project, Czech Technical University in Prague, 2010.

[15] J. von Neumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21(3):232–237, 1950.