KU LEUVEN

The 1D Riemann Problem: Isothermal Hydrodynamics

Individual Assignment

Bert Depoorter
MSc. Master of Physics
Option Theoretical Physics
Academic Year 2024-2025

Faculty of Science
Department of Physics & Astronomy
Institute for Astronomy and Astrophysics
G0B30a: Computational Methods for Astrophysical Applications
Instructor: prof. Rony Keppens & prof. Jon Sundqvist

The assignment reads: Riemann problem for 1D isothermal hydro: make python scripts for getting the exact solution to the Riemann problem in state space views, and show the various possible out- comes. This requires you to go through, and understand, the more technical/advanced section 4.2 from chapter 4, with exercises 4.2 until 4.5 (chapter 4).

Riemann problem for 1D isothermal hydro: make python scripts for getting the exact solution to the Riemann problem in state space views, and show the various possible outcomes. This requires you to go through, and understand, the more technical/advanced section 4.2 from chapter 4, with exercises 4.2 until 4.5 (chapter 4).

The isothermal hydrodynamics are governed by the following set of PDE's in 1D:

$$\begin{aligned} \partial_t \rho + \partial_x(\rho v) &= 0 \\ \partial_m + \partial_x(\rho v^2 + p) &= 0 \end{aligned} \tag{1.1}$$

The closure relation for the isothermal case is given by $p(\rho) = c_i^2 \rho$, where c_i is the isothermal sound speed. Note that this system only contains the conservation equations for matter and momentum, not energy: by requiring isothermality we assume an input of energy to keep the temperature constant. This is typically the case in astrophysical scenarios where incoming radiation heats the matter on a timescale much faster than the timescale of change of momentum/matter.

The Riemann problem implies a specific choice of initial conditions.

$$\rho(x,0) = \begin{cases} \rho_L & x < 0 \\ \rho_R & x > 0 \end{cases} \tag{1.2}$$

$$v(x,0) = \begin{cases} v_L & x < 0 \\ v_R & x > 0 \end{cases} \tag{1.3}$$

Physically, this choice corresponds to 2 gases of different density and volume making contact at $x = 0, t = 0$.

1.1 Alternative writings of 1D isothermal hydrodynamics

We rewrite the isothermal hydrodynamics system in 1D to the conservation form:

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{0} \tag{1.4}$$

where $\boldsymbol{U} = (\rho, m)^T$ and $\boldsymbol{F}(\boldsymbol{U}) = (m, \frac{m^2}{\rho} + c_i^2 \rho)^T$. We can rewrite this system in various ways, as described in the course notes. One way is to work with primitive instead of conservative

variables. This gives the form

$$\begin{pmatrix}\rho\\ v\end{pmatrix}_t + \begin{pmatrix} v & \rho \\ \frac{c_i^2}{\rho} & v\end{pmatrix}\begin{pmatrix}\rho\\ v\end{pmatrix}_x = \begin{pmatrix}0\\0\end{pmatrix} \tag{1.5}$$

Using some algebraic manipulation, this can be written in the form

$$\begin{cases}(v - c_i \ln\rho)_t + (v-c_i)(v - c_i\ln\rho)_x = 0\\ (v + c_i \ln\rho)_t + (v+c_i)(v + c_i\ln\rho)_x = 0\end{cases} \tag{1.6}$$

1.1.1 Hugoniot Locus

The Rankine-Hugoniot relations describe the family of states connected via a shock to a given initial state. Via this relation

$$\boldsymbol{F}(\tilde{\boldsymbol{U}}) - \boldsymbol{F}(\hat{\boldsymbol{U}}) = s(\tilde{\boldsymbol{U}} - \hat{\boldsymbol{U}}) \tag{1.7}$$

This represents a system of 2 equations for three unknowns, the shock speed s and the state variable $\boldsymbol{U} = (\rho, m)^T$. The solution is then parametrized by one variable, which we take $\theta \in [-1, +\infty)$ such that $\rho = \hat{\rho}(1+\theta)$. As explained in the notes, all states that can be connected to $\hat{\boldsymbol{U}}$ via a shock are described by the 1-parameter solution

$$\begin{cases} m &= \frac{\rho}{\hat{\rho}}\hat{m} \pm \sqrt{\frac{\rho}{\hat{\rho}}}c_i(\rho - \hat{\rho}) \\ s &= \frac{\hat{m}}{\hat{\rho}} \pm \sqrt{\frac{\rho}{\hat{\rho}}}c_i \end{cases} \tag{1.8}$$

The curves for the Hugoniot locus are plotted on figure 1.1 for two different choices of initial state $\hat{\boldsymbol{U}}$. On the plot, also the derivative in this initial state is shown.

The derivative is calculated via the direction of both right eigenvectors of a given system:

$$\vec{r}_1 = \begin{pmatrix}1\\ v - c_i\end{pmatrix}, \quad \vec{r}_2 = \begin{pmatrix}1\\ v + c_i\end{pmatrix} \tag{1.9}$$

In that case, the derivative of the Hugoniot locus in $\hat{\boldsymbol{U}}$ is given as

$$\left.\frac{dm}{d\rho}\right|_{\hat{\rho}} = \hat{v} \pm c_i. \tag{1.10}$$

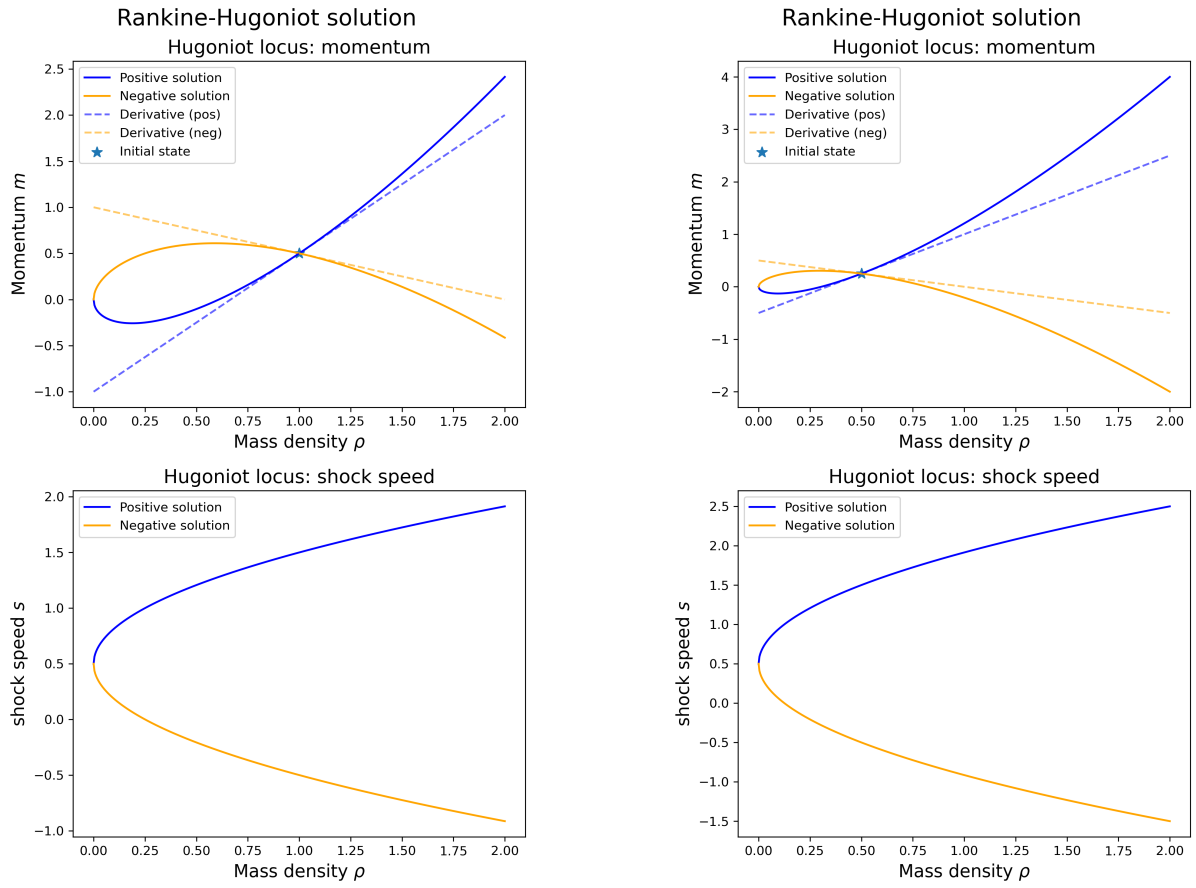
On the plot, we can visually see that these curves very much look like the derivatives at this particular point.

The Hugoniot locus actually consists of 4 curves coming together in the state $\hat{\boldsymbol{U}}$, of which 2 connect to lower density and 2 connect to higher density. The notes describe how the Lax entropy condition can be used to determine which parts of the curves are physically relevant.

1.1.2 Integral curves

However, there is more to the Riemann problem than just this pure discontinuity-based solution. Here we will treat the continuous self-similar part of a possible solution. For this, we exploit the invariance of the isothermal HD equations under the coordinate transformation $x \to ax, t \to at$. In the notes, it is explained how we arrive at the integral curves describing all possible states that can be connected to an initial state $\hat{\boldsymbol{U}}$ via either a 1-rarefaction wave or a 2-rarefaction.

$$\begin{cases} m(\rho) = \hat{v}\rho - c_i\rho\ln\frac{\rho}{\hat{\rho}} & \text{1-rarefaction} \\ m(\rho) = \hat{v}\rho + c_i\rho\ln\frac{\rho}{\hat{\rho}} & \text{2-rarefaction}\end{cases} \tag{1.11}$$



(a) Initial state $\hat{\boldsymbol{U}} = (1, 1)^T$.

(b) Initial state $\hat{\boldsymbol{U}} = (1, 0.5)^T$.

Figuur 1.1: The Hugoniot locus for two different initial states $\hat{\boldsymbol{U}} = (\rho, m)^T$. The upper plot shows the momentum as a function of ρ, the lower plot shows the shock speed s as a function of ρ.

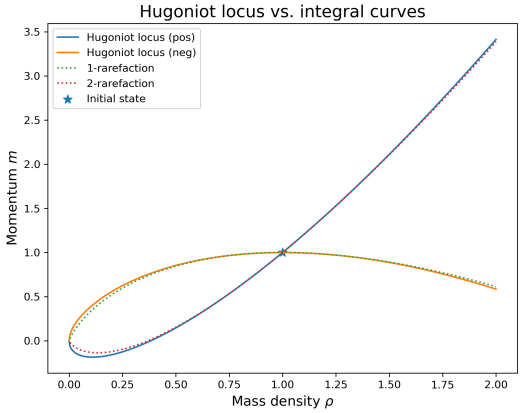
Note that here the derivative in the initial state $\hat{\boldsymbol{U}}$ is still given by equation (1.10). This means locally both curves are the same. On Figure 1.2 the integral curves and Hugoniot locus are plotted for a certain state $\hat{\boldsymbol{U}} = (1, 1)^T$. We see that locally both curves are identical (i.e. the derivative (1.10) is the same) but at the ends both curves start to separate.

As can be seen on Figure 1.2, there are *a priori* 8 curves coming together at one point in state space, which corresponds to the initial state $\hat{\boldsymbol{U}}$. However, based on additional considerations, four of these can be discarded as unphysical. For a left state, the Lax entropy condition only allows 1-shocks to higher density and 2-shocks to lower density. For rarefactions, there is an opposite requirement, so we have that a left state can connect to a right state with lower density, higher velocity via a 1-rarefaction.

1.2 Solving the Riemann problem in general

The goal of the assignment is to implement an algorithm in python that determines, from only the left initial state $\boldsymbol{U}_L = (\rho_L, m_L)^T$ and the right initial state $\boldsymbol{U}_R = (\rho_R, m_R)^T$, how the system will evolve under the isothermal hydrodynamic equations.

As explained in the lecture notes, there are 9 possibilities depending on the orientation of the relative location of $\boldsymbol{U}_L$ and $\boldsymbol{U}_R$ to each other. I have constructed the following decision tree



Figuur 1.2: Integral curves representing rarefaction waves compared to the Hugoniot loci for shocks. Both are calculated for the same state. the same state. We see that locally both curves have the same derivative in $\hat{\boldsymbol{U}} = (1, 1)^T$

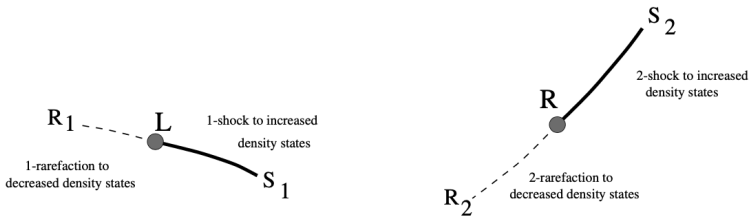
based on the information provided in the lecture notes. Depending on the initial locations, there will form an intermediate state $\boldsymbol{U}_m = (\rho_m, m_m)^T$. This state is connected to the left state $\boldsymbol{U}_L$ via a 1-wave and to the right state $\boldsymbol{U}_R$ via a 2-wave.

For both the 1-wave and 2-wave there are 3 possibilities: a shock wave, a rarefaction wave or no wave (absent). This last case is only possible when either the left or right state lies on the Hugoniot locus or the integral curve of the other state. *A priori*, there are nine possible scenarios. In case the left and right states are identical; there will be no wave at all, since both mass density ρ and momentum m are equal for the left and right state. There are four cases that lead to an absent 1- or 2-wave. That also means there is no intermediate state. Here we indicate $\boldsymbol{U}_L$ with L and $\boldsymbol{U}_R$ with R.

- R on Hugoniot locus of L
  - 1-wave: shock
  - 2-wave: absent
- R on integral curve of L
  - 1-wave: rarefaction
  - 2-wave: absent
- L on Hugoniot locus of R
  - 1-wave: absent
  - 2-wave: shock
- L on integral curve of R
  - 1-wave: absent
  - 2-wave: rarefaction

These degenerate cases are checked first. If none of the conditions is fulfilled, we proceed to the four generic cases: here an intermediate state will form, connected to the left state via the 1-wave and to the right state via the 2-wave.

To determine the wave type, we need to consider the relative orientation of $\boldsymbol{U}_L$ and $\boldsymbol{U}_R$.



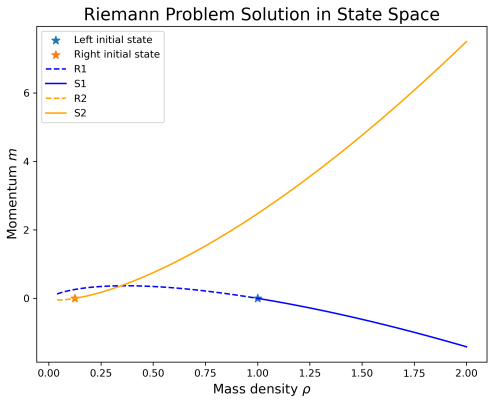
Figuur 1.3: The elementary graphs for a left state and a right state, respectively. Figures taken from the lecture notes.

We only need to consider the intersection points of the two elementary graphs shown in Figure 1.3. Based on the location of the state $\boldsymbol{U}_L$, we use the formulas for the Hugoniot loci and integral curves to determine which parts of the curve will intersect. This provides us with a qualitative picture of which type of shock/rarefaction will develop.

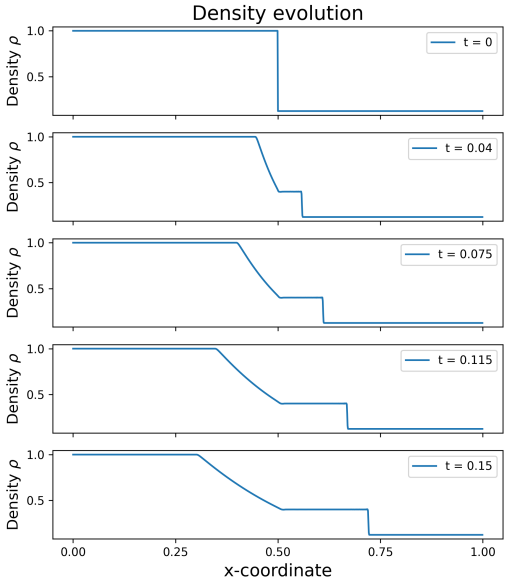
In the special cases where both the 1-wave and 2-wave are shock-type or rarefaction-type waves, the intermediate state can be calculated exactly. The formulae are given in the notes.

The python script implements this decision tree and afterwards plots the appropriate parts of the curve. When both waves are shocks or rarefactions, the intermediate state is plotted as well. This function is part of the *plotting* class and is called full_solution.

On Figure 1.4, 1.5 and 1.6 the three tests described in the lecture notes are repeated. Note that I find the same results for the 1-wave connecting the left state to the intermediate state and for the 2-wave connecting the intermediate state to the right state. On the left, the qualitative conclusion is shown. On the right, I simulated the same Riemann initial conditions with MPI-AMRVAC. We conclude that the results

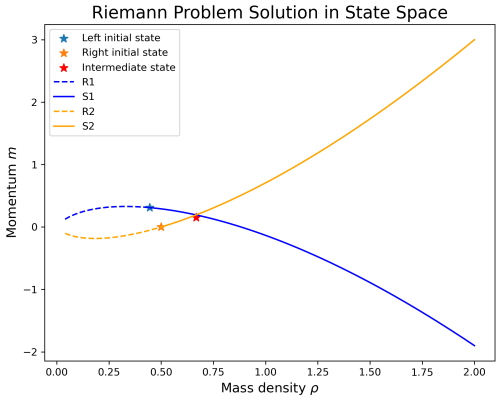


(a) Qualitative state space solution of test 1. Based on this figure, we see that the intersection point lies on the 1-shock and 2-rarefaction curves. We conclude that 1-wave is a shock, and the 2-wave a rarefaction wave.

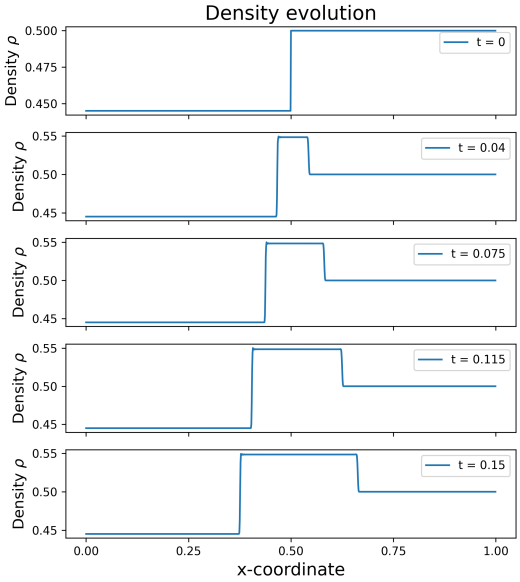


(b) Density evolution for Riemann problem using MPI-AMRVAC ([1]).

Figuur 1.4: Test 1: $\boldsymbol{U}_L = (1, 0)^T$, $\boldsymbol{U}_R = (0.125, 0)^T$. According to the function full_solution() the 1-wave is a shock while the 2-wave is a rarefaction.

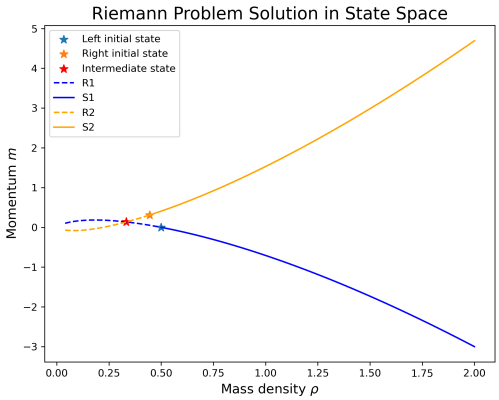


(a) Qualitative state space solution of test 2. Based on this figure, we see that the intersection point lies on both shock curves. We conclude that both the 1- and 2-wave will be shocks.

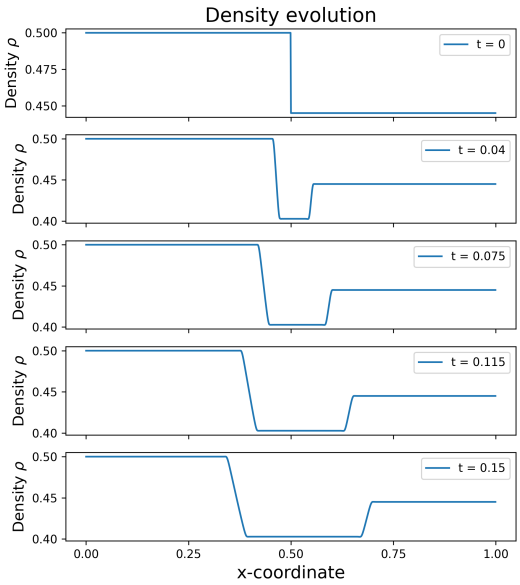


(b) Density evolution for Riemann problem using MPI-AMRVAC ([1]).

Figuur 1.5: Test 3: $\boldsymbol{U}_L = (0.445, 0.31061)^T$, $\boldsymbol{U}_R = (0.5, 0)^T$. According to the function full_solution() both the 1-wave and the 2-wave are shocks. The intermediate state is calculated and plotted as well.



(a) Qualitative state space solution of test 3. Based on this figure, we see that the intersection point lies on both rarefaction curves. We conclude that both the 1- and 2-wave will be rarefactions.



(b) Density evolution for Riemann problem using MPI-AMRVAC ([1]).

Figuur 1.6: Test 3: $\boldsymbol{U}_L = (0.5, 0)^T$, $\boldsymbol{U}_R = (0.445, 0.31061)^T$. According to the function full_solution() both the 1-wave and the 2-wave are rarefactions. The intermediate state is calculated and plotted as well.

```
[7]: # Plot only solutions of decision tree

     # test 1
     print('Result test 1:')
     solve(1, 0.125, 0, 0, figure=False)
     print('\n')

     # test 2
     print('Result test 2:')
     solve(0.445, 0.5, 0.31061/0.445, 0, figure=False)
     print('\n')

     # Test 3
     print('Result test 3:')
     solve(0.5, 0.445, 0, 0.31061/0.445, figure=False)

     Result test 1:
     {'1-wave': 'rarefaction', '2-wave': 'shock'}


     Result test 2:
     {'1-wave': 'shock', '2-wave': 'shock'}


     Result test 3:
     {'1-wave': 'rarefaction', '2-wave': 'rarefaction'}
```

Figuur 1.7: Output of the three tests as described in the notes for the implemented decision tree. The results are the same.

The output of the implemented decision tree is shown in Figure 1.7. These are identical to the results as described in the notes. Also, when plotting the left and right state, we can deduce the same results from the created plots.

On this page I briefly outline the code I have written. I have added an example notebook with my submission on Toledo which provides the figures I created and included in my report.

2.1 Code structure

The code consists of two classes: one class, the Riemann class, serves mostly under the hood; it only needs to be initialized with the Riemann initial conditions and (optional) general parameters like temperature and molecular mass μ of the molecules making up the fluid. All user interaction is intended to go through the plotting class, which contains various functions for plotting different aspects of the problem at hand.

- The integral curves of a given state. These are all states which can be connected to a given state by means of a rarefaction wave.
- The Hugoniot locus of a given state. Starting from an initial state, we calculate the Hugoniot loci curves that describe the possible states connectable to this initial state by means of a shock wave.
- The numerical solution of the Riemann problem. The implemented options to evolve the initial discontinuity include:
  - Limiter: minmod, vanleer, woodward
  - flux scheme method: TVDLF, upwind, MacCormack
  - Time integration: onestep, twostep

  Note that the numerical scheme is self-implemented, i.e. the solution will likely not be the fastest / most accurate / most stable. I have not included figures created using this solver; the results were not very trustworthy.

I have added different functions which create plots with various elements. Each function contains a docstring with a short description as well as the necessary input and what the output will be.

The code is structured around the central file *riemann.py*. This file contains a class equipped with functions for calculating the Hugoniot locus, the integral curves, and for numerically evolving the Riemann initial conditions. Also, the decision tree to solve the Riemann problem based on the relative location of the left and right state is part of this class. The user interaction is supposed to go via the plotting class, which takes the arguments to initialize the riemann class and produces the desired plots.

I have gathered all code I used, including the mod_usr.f and parameter file for amrvac, in a Github repository: `https://github.com/BertDepoorter/Individual_CMfAA`. At the end of the semester, I will make this repository private.

REFERENCES

[1] Rony Keppens e.a. „MPI-AMRVAC: A parallel, grid-adaptive PDE toolkit". In: *Computers & Mathematics with Applications*. Development and Application of Open-source Software for Problems with Numerical PDEs 81 (1 jan 2021). Citation for use of MPI-AMRVAC., p. 316–333. ISSN: 0898-1221. DOI: 10.1016/j.camwa.2020.03.023. URL: https://www.sciencedirect.com/science/article/pii/S0898122120301279 (bezocht op 15-01-2025).

[2] Jon Sundqvist Rony Keppens. „Computational Methods for Astrophysical Applications". Unpublished lecture notes, accessed via Toledo.