1-D Shock Wave Propagation

ASTRON 255 PROJECT I

Ruoyi Yin

February 21, 2020

1 Introduction

The project is about simulating shock wave propagation using finite difference methods, which is a computational way for representing partial differential equations and simulating dynamic physical systems achieved by programming. For our project specifically, we evaluated the dynamics of 1-D shock wave by setting up boundary and initial conditions and the inner finite difference equations. The second part of the report detailedly describes the theory and methods used, with all of the parameters specified. The third part discusses our results, linear and quadratic artificial viscosity, result comparisons by choosing different constants for artificial viscosity, and exploration of differences between our results and analytical solutions.

2 Theory and Methods

2.1 Finite Difference Method

We use Python to run shock wave propagation. Since the wave evolves with time, we construct two loops, one of which loops through time, and in each time-step we also loop through each spatial point.

Our independent variable is time, dependent variables are coordinate (x), velocity (u), energy (E), pressure (P), density (ρ) and artificial viscosity (q). Specifically, x is defined at integer time and integer space points; u is defined at half-integer time and integer space; E, P, and ρ are defined at integer time and half-integer space; q is defined at half-integer time and half-integer space.

For boundary & initial conditions: x and u are specified from time-step 1 - 51 (51 points). E, P, q and ρ are specified at 1.5 - 50.5 (50 points). Additionally, P and q are also defined at 0.5 time-step, since we need a constant pressure to drive the shock wave. Pressure at 0.5 point always has value 5, while artificial viscosity at 0.5 point always has value 0.

In the simulation, we apply equal mass zoning, so $\delta m = 6$, $\delta t = 0.1$.

At each time-step, we utilize 6 equations to update 6 variables respectively. First, we update velocity, then coordinate, density, sound speed (we don't need to save this variable at each loop since it's always just based on the

previous time-step), artificial viscosity, energy, and at last pressure. The detailed equations and steps are shown in Lecture Note P.96-97.

2.2 Artificial Viscosity

We include both linear and quadratic terms in artificial viscosity. C_0^2 is the coefficient for quadratic q term, and C_1 is the coefficient for linear q term. We employ 4 different models by setting up different C_0^2 and C_1 terms. However, not all of the models work. For the first model, the simulation works fine and we get values close to the results calculated from analytical solutions. For the last three models, we could not get correct solutions if setting C_1 to be 0. Therefore, we want to discuss the reasons we think about that are related to the situation and also provide the least C_1 needed to get stable propagation results.

2.3 Analytical Solutions

$$\rho_1 = \frac{\gamma + 1}{\gamma - 1}\rho_0 \tag{1}$$

$$\rho_1(D - u) = \rho_0 D \tag{2}$$

$$\rho_0 D u = (P_1 - P_0) \tag{3}$$

Here, ρ_1 is the post-shock density, ρ_0 is the pre-shock density. P_1 is the post-shock pressure, and P_0 is the pre-shock pressure. D is the velocity of the shock front, and u is the post-shock velocity. Therefore, by plugging in the initial and boundary conditions ($\rho_0 = 6$, D = 1, $\gamma = 1.4$), we can get analytical solutions as provided in the lecture note. The Hugoniot relations also involve another equation about internal energy before and post shocks. Since we don't explore internal energy in the project, that equation is not included [2].

In the next section, we will compare our simulation results with analytical results, which are:

$$\rho_1 = 36 \tag{4}$$

$$u_1 = \frac{5}{6} \tag{5}$$

$$P_1 = 5 \tag{6}$$

(Post shock density, velocity and pressure)

2.4 Modifications

We make some small modifications to the code, slightly different from equations in lecture notes. For the sound speed, we use pressure and density both from previous time step. Second, for equation 3 that we use to derive density for new time step, we use x coordinate value from previous time step (n) for the next spatial point (l + 1) , since the current time step (n + 1) for the next spatial point (l + 1) has not been updated yet.

3 Results and Discussion

3.1 Direct Comparison

For the first case, the post-shock pressure we get is pretty close to 5, as the analytical solution. The post-shock density is between 34 - 40 and get stable at around 39, which is higher than the analytical solution 36. The velocity is around 0.8, which is also close to and a little bit lower than the analytical solution 0.833. For artificial viscosity, we can see that the value is high at around the shock front and almost equal to 0 at other spatial points, which is what we hope to get. We only hope to get artificial viscosity value around the shock front and not to affect other spatial points too much. However, the highest value of artificial viscosity is at around x = 48, instead of 45, showing that the shock front moves a little bit faster than what we supposed (we supposed it to be 0).

For the second case, we cannot get a stable simulation result by setting C_1 to 0 and only including the quadratic artificial viscosity term. We run the code and include an IF statement to print out the values that violate stable criteria, and do get values printed out. We try to shorten δt and try values 0.2, and 0.02, and could get it run for the first several hundreds of time step but none of them can reach t = 45. Therefore, we choose the smallest C_1 value (0.2) with 1 decimal point precision that could return stable result (Figure 2). The post-shock pressure we get is pretty close to 5, as the analytical solution. The post-shock density is roughly between 32 - 40 and get stable

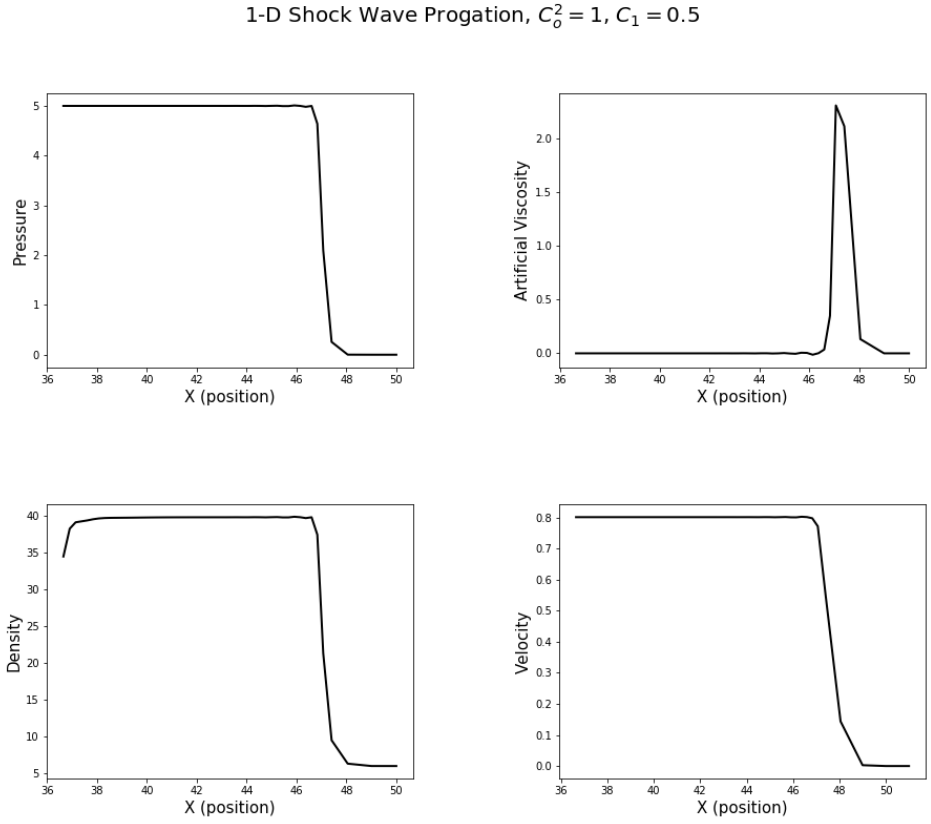

Figure 1: Shock wave propagation for $C_0^2 = 1$ and $C_1 = 0.5$. Four plots are pressure, artificial viscosity, density, and velocity versus coordinate position X respectively.

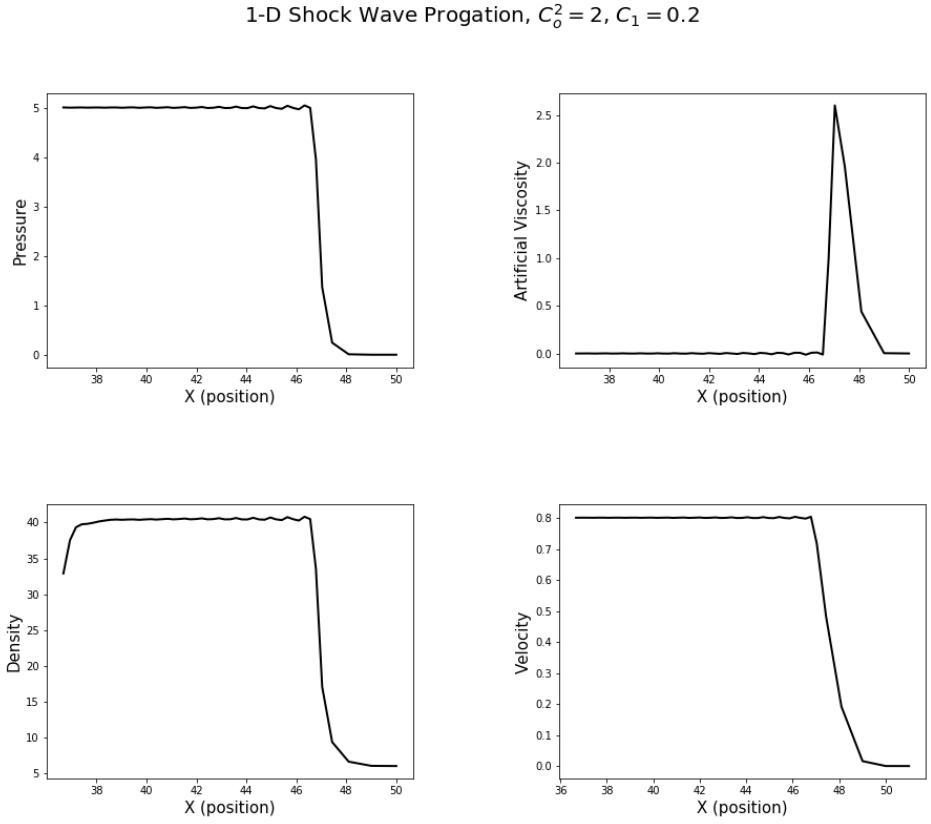

Figure 2: Shock wave propagation for $C_0^2 = 2$ and $C_1 = 0.2$. Four plots are pressure, artificial viscosity, density, and velocity versus coordinate position X respectively.

at around 39, which is higher than the analytical solution 36. The velocity is around 0.8, which is also close to and a little bit lower than the analytical solution 0.833. For artificial viscosity, we can see that the value is high at around the shock front and almost equal to 0 at other spatial points, which is what we hope to get. And the wave front moves faster than we suppose.

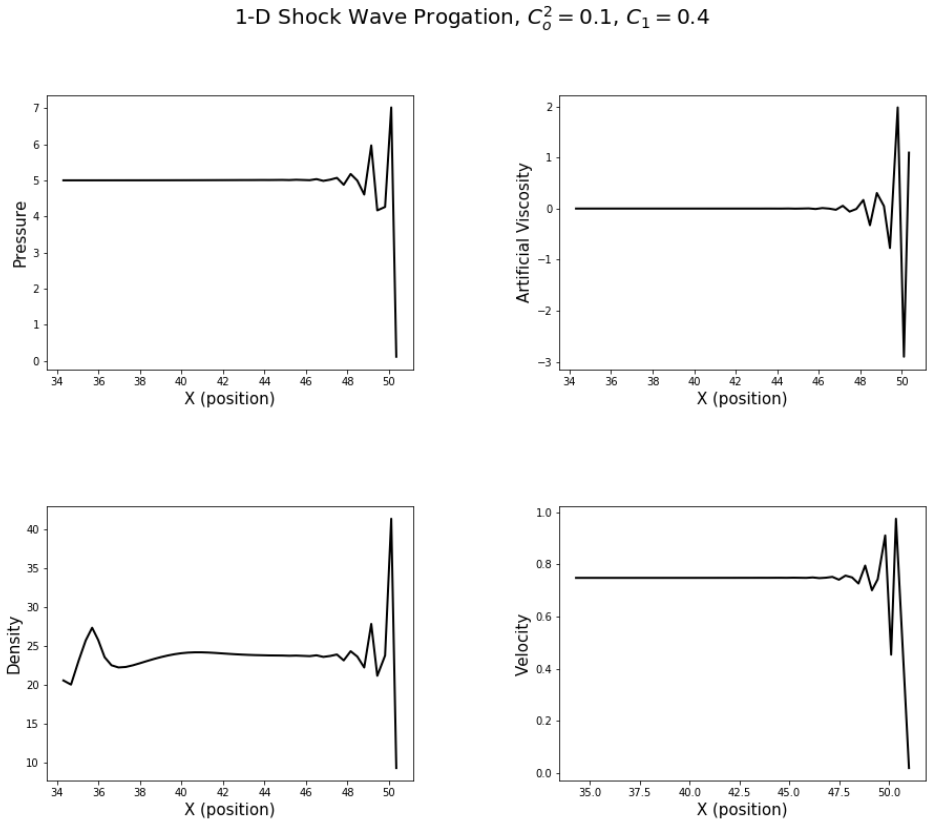

Figure 3: Shock wave propagation for $C_0^2 = 0.1$ and $C_1 = 0.4$. Four plots are pressure, artificial viscosity, density, and velocity versus coordinate position X respectively.

For the third case, we cannot get a stable simulation result by setting C_1 to 0 and only including the quadratic artificial viscosity term. We run the code and include an IF statement to print out the values that violate stable criteria, and do get values printed out, almost the same situation as the second case. Therefore, we choose the smallest C_1 value (0.4) with 1 decimal point precision that could return stable result (Figure 3). The post-shock pressure we get is pretty close to 5, as the analytical solution. The post-shock density, however, varies a lot and get stabilized at around 24, which is way smaller than the real analytical result 36. The velocity is around 0.75,

which is lower than the analytical solution 0.833. For artificial viscosity, we can see that the value is not always positive, and we get a pretty large value oscillation near shock front. At points far from shock front, the value is close to 0. And the wave front moves faster than we suppose.

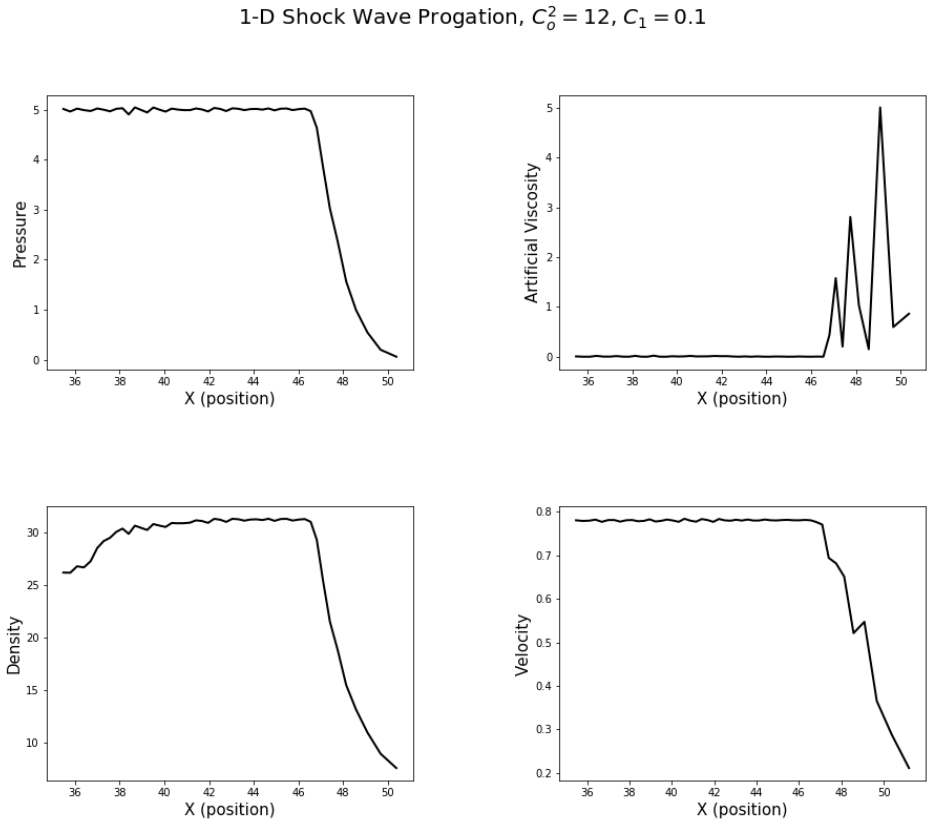

Figure 4: Shock wave propagation for $C_0^2 = 12$ and $C_1 = 0.1$. Four plots are pressure, artificial viscosity, density, and velocity versus coordinate position X respectively.

For the fourth case, we cannot get a stable simulation result by setting C_1 to 0 and only including the quadratic artificial viscosity term. We run the code and include an IF statement to print out the values that violate stable criteria, and do get values printed out. We try to shorten δt and could get it run for the first several hundreds of times but none of them can reach t = 45. Therefore, we choose the smallest C_1 value (0.1) with 1 decimal point precision that will return stable result (Figure 4). The post-shock pressure we get is pretty close to 5 with only small amount of oscillations, as the analytical solution. The post-shock density is roughly between 26 - 32 and get stable at around 30, which is lower than the analytical solution 36. The

velocity is around 0.79, which is also close to and a little bit lower than the analytical solution 0.833. For artificial viscosity, we can see that the value is high at around the shock front and almost equal to 0 at other spatial points, but it's not one peak but 3 peaks combined, showing that there are large oscillations at q values near shock front. And the wave front moves faster than we suppose.

3.2 Linear and Quadratic Artificial Viscosity

Physically, the shock wave has a width of few mean free paths, so it's not a discontinuity in real world. However, since the numerical simulation cannot resolve the wave front due to the limit set by mesh grid size. If we want to better simulate the movement of shock wave, we need to include a term, artificial viscosity, in our Lagrangian Hydrodynamic equations. The formula is:

$$q = -\rho(c_0\Delta x)^2 \frac{\partial u}{\partial x} \mid \frac{\partial u}{\partial x} \mid \tag{7}$$

for quadratic term, and:

$$q = -\rho(c_1\Delta x) C_s \mid \frac{\partial u}{\partial x} \mid \tag{8}$$

for linear term. Since we hope the artificial viscosity term to just effect in shock front position, it's better to have an q term that has small or no effect on places far from shock front. Therefore, we hope the q term only functions when $\frac{\partial u}{\partial x} < 0$. Since our reference frame moves with the shock front, and the pre-shock velocity is zero in normal reference frame, the pre-shock gas has negative velocity with respect to Lagrangian frame.

We hope the artificial viscosity is strong enough to damp out the oscillations and overshoots in the post-shock region, while diminishing the width of the shock front to make it more similar to real-world cases. Therefore, we have a trade-off here, since stronger damping term usually means that it would effect in a relatively wide range and increase the shock front width [1].

The quadratic term contains $\mid \frac{\partial u}{\partial x} \mid^2$, since the velocity pre- and post-shock stays relatively constant, this quadratic term will only effect in a small width

around the shock front. However, due to its quadratic term, its effect reduces quickly away from the shock front, and sometimes cannot damp out all the oscillations into an acceptable range. If we increase C_0^2, we would be able to damp the oscillation, but will also result into wider shock front, which is not ideal [1].

The linear term contains only $|\frac{\partial u}{\partial x}|$ term, so it would not become small very quickly, but since it's a linear term, it would effect on larger spatial area, increasing shock front width. Therefore, even though we introduce the linear term in our expression, the coefficient for it would generally be smaller than quadratic term coefficient [1].

Therefore, we combine the quadratic and linear terms together in our artificial viscosity expression in order to maximize the damping effect while minimizing shock front width [1].

3.3 Further Discussion on Shock Front Width and Glitches

3.3.1 Shock Wave Spread

For the first case, from pressure, density and velocity plots, it's a little bit hard to evaluate the width of shock front. From artificial viscosity term, if we treat the width of the peak as our shock front width, it's roughly 1.6 spatial steps.

For the second case, when looking at the width of artificial viscosity peak, it's roughly 2.7 spatial steps.

For the third case, the shock front width is roughly 3.3 spatial steps and since it does not reach stability as t $>$ 51, we cannot know if the shock width should include more spatial steps after t $>$ 51. Since our spatial steps are within range from 1 - 51, we won't consider steps larger than 51 here.

For the fourth case, The shock front width is roughly 4.4 spatial steps.

3.3.2 Glitches

Comparing the results we get from the 4 plots, We are able to find out that for the first two cases, the shock wave propagation returns pretty smooth

values. Even though in the second case, there are some small oscillations in all variables in the post-shock region, the general shapes are smooth.

However, in the third and fourth cases, there are pretty significant signatures of oscillations, which we suppose are the glitches.

Specifically, in the third case, there are glitches or large value jumps reported in every variables plotted. Since artificial viscosity is used for smoothing the shock front and eliminating discontinuity, we think the problem here is that the artificial viscosity term is not large enough to quickly damp out oscillations and make the wave front smooth.

So here, we introduce another plot that we made for $C_0^2 = 0.1$ (the same as third case), while only increase C_1 and set it to be 0.8 (Figure 5).

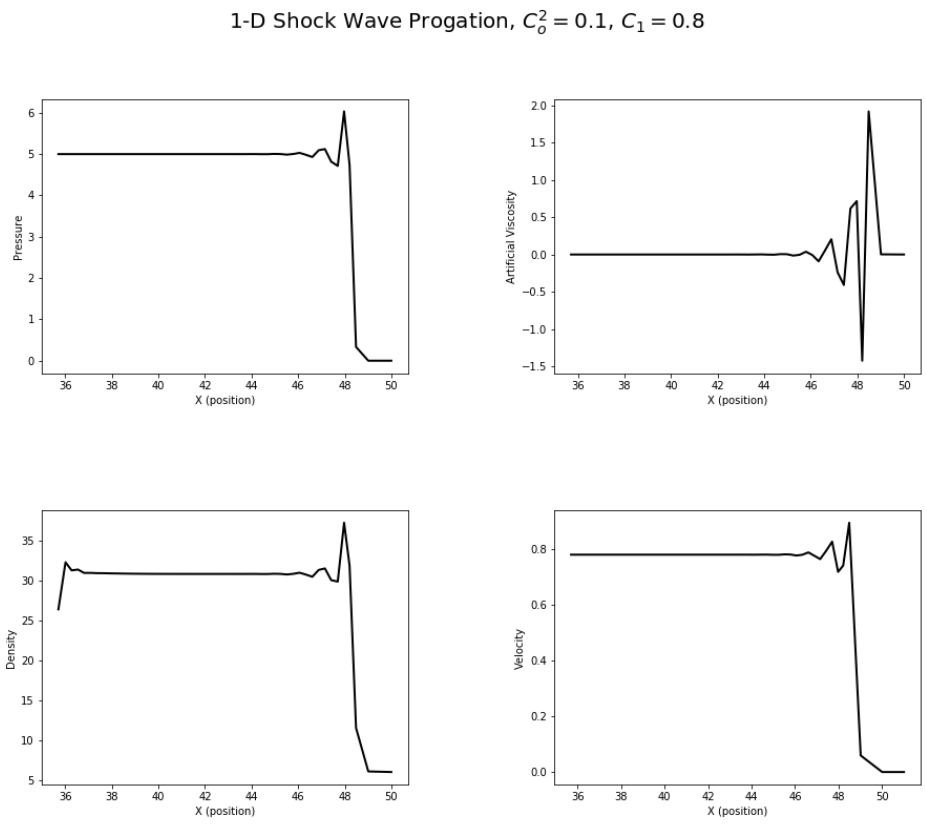

Figure 5: Shock wave propagation for $C_0^2 = 0.1$ and $C_1 = 0.8$. Four plots are pressure, artificial viscosity, density, and velocity versus coordinate position X respectively.

Comparing to Figure 3, Figure 5 shows more stable shock wave propagation simulation. The shock front width is smaller and the values for all variables

remain their pre-shock values for t $>$ 49. From the comparison, we see that the glitch depends partially on numerical method and the value we choose for both the quadratic and linear terms.

For the fourth case, all of the curves except for artificial viscosity stays roughly stable and smooth. However, the shock wave width is large. Since as explained in the previous part, a large quadratic term coefficient would cause the artificial viscosity to effect on a wider range around the shock front, thus increasing shock width.

Therefore, the glitches here all partially depend on choice of artificial viscosity, a numerical reason.

Besides, partial differential equations limit the independent variable intervals to be infinitely small. However, as we do simulation using finite difference method to approximate PDE, our intervals cannot be infinitely small, since we also need to consider computing costs. Therefore, by choosing interval length, we get simulation values that oscillate around real values since our method is not a continuous one but a quantized one. This is also a numerical reason, not related to artificial viscosity, but existed in finite difference method in general.

Furthermore, physically in real world, the gas dynamics in and around shock front would not be smooth, since there are a lot of physics going on in small scale. Our simulation only predicts the shock wave propagation in a macro scale. But in micro scale, the particles move randomly. In addition, our simulation only includes pressure term in momentum conservation equation, but in real world, there would also be viscosity term, gravitational, electric and magnetic force that would cause oscillations.

References

[1] James Campbell and Rade Vignjevic. "Artificial Viscosity Methods forModelling Shock Wave Propagation". In: *Predictive Modeling of Dynamic Processes: A Tribute to Professor Klaus Thoma* (June 2009), pp. 349–365. DOI: 10.1007/978-1-4419-0727-1_19.

[2] Y.B. Zel'dovich and Y.P. Raizer. *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena*. Dover Books on Physics. Dover Publications, 2002. ISBN: 9780486420028. URL: https://books.google.com/books?id=zVf27TMNdToC.