The Strongly Attracting Character of Large Amplitude Nonlinear Resonant Acoustic Waves Without Shocks. A Numerical Study

by

Dimitri Vaynblat

Magister (B.S. and M.S.), Leningrad Polytechnic Institute (1988)

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 1996

Author .....
Department of Mathematics
August 23, 1996

Certified by .....
Rodolfo R. Rosales
Professor of Mathematics
Thesis Supervisor

Accepted by .....
Hung Cheng
Chairman, Applied Mathematics Committee

Accepted by .....
Richard Melrose
Chair, Department Committee on Graduate Students

The Strongly Attracting Character of Large Amplitude Nonlinear Resonant Acoustic Waves Without Shocks. A Numerical Study

by

Dimitri Vaynblat

Submitted to the Department of Mathematics
on August 23, 1996, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In this thesis, a new class of fully nonlinear solutions for the one-dimensional inviscid Euler equations of Gas Dynamics in a bounded domain with reflecting boundary conditions is obtained numerically. These solutions have the remarkable property that they do not develop shocks at any time (even though they carry large pressure variations) and can be characterized as highly nontrivial, large amplitude, acoustic standing waves. Numerical experiments provide strong evidence that, regardless of the initial conditions, these waves always emerge in the flow after a sufficient time.

The computational simulations reveal a typical scenario for the flow evolution. By the end of an initial (very short) stage of intense interaction, which takes a few acoustical periods, the main features of one of these waves are already developed. During the second stage, which lasts for several hundreds acoustical periods, all shocks decay. By the end of this second stage, the standing wave completely dominates the flow. After that, throughout a last stage, the standing wave oscillates entirely unchanged for a very long time – thousands of computational periods. The motion during this last stage appears to be almost periodic.

Because the study of these new waves requires long time numerical simulations and since these waves seem to be very sensitive to the entropy variations, the computational experiments would not be possible without a new numerical scheme developed in this thesis.

We devised a method which belongs to the Godunov-type group of schemes. Godunov-type schemes are conservative, upstream, fully nonlinear numerical methods. They are widely used for computations involving the Euler equations of Gas Dynamics, because they can handle shocks correctly and can provide detailed information about the flow.

But we could not use already existing Godunov-type schemes. When these methods were tested on various problems, we found out that they all had some very basic common flaws: They all produced too much numerical parasitic entropy in various situations. Moreover, the amplitudes of the parasitic entropy waves could be of the same order of magnitude as that of the acoustic waves in many typical shock – flow disturbance interaction problems.

After we studied and understood the mechanisms which trigger the spurious entropy wave production, we were able to develop a scheme which is capable of controlling and minimizing the production of parasitic entropy. The dramatic reduction of the numerical noise generated by this new scheme is enabled by: (a) Detection of physical discontinuities and their perfect (one-node wide) resolution, (b) Perfect resolution of discontinuity interactions (no smearing is introduced during the interactions).

We tested this code on standard one-dimensional problems with known exact solutions and obtained extremely good results: With very few mesh points, the exact solution is reproduced with very little noise and high accuracy — the scheme is actually second order accurate everywhere, including shocks (L^1-norm second order accurate).

Thesis Supervisor: Rodolfo R. Rosales
Title: Professor of Mathematics

Acknowledgments

I would like to thank my advisor, Professor Ruben Rosales, for his guidance and very generous support during my years at MIT. I have learned a great deal from him, and his scientific views exerted a lasting influence on me.

I am grateful to Professors Fabian Waleffe and Michael Brenner for accepting being part of my thesis committee, and to all the Math department, faculty and staff, for five years of intense yet playful learning. The research in this thesis was partially supported by NSF grant DMS-9311438 and NASA grant NAG1-1519.

My stay at MIT would not have been half as joyful without the presence of magnificent graduate and postdoctoral students, some of whom will remain forever among my best friends. Let me then say thanks to Carlos, Ernest, Esteban, Ethan, Dan, Giuseppe, Jianke, Jihad, Lu, Marcos, Mats, Miklos, Paul, Peter, Radica, Richard, Rodney, Satomi, Seqian, Tony, Wendy, Yuanping, just to mention few. A particular note of appreciation is due to Shinya, Michael, and Guillermo, with whom I shared my learning experience, my office (the former two), and my advisor (the latter two). Their friendship is very dear to me.

There is no way I can thank my parents for all what they have done for me; without their love and support, my stay at MIT, as basically everything in my life, would have been unthinkable. My deepest gratitude goes to Maria, for her patience.

Finally, I would like to thank Aeroflot, Russia State Airlines, now defunct, for their flights being notoriously late. While waiting in Kennedy International Airport for my friend coming from Russia with one of such flights (being 12 hours late) on October 2, 1994, I had plenty of time to come up with all main concepts of the new numerical scheme.

Contents

List of Figures

Chapter 1

Introduction

The problem of the existence of "non-breaking for all times" solutions of the Euler equations is an open question. In fact, the existence of such solutions — with a nontrivial acoustic component — was not even suspected till rather recently. New results have been recently obtained (both analytically and numerically) for the full Gas Dynamics equations [1], which show that the set of non-breaking solutions is large and highly nontrivial. It was shown that there is a variety of traveling and standing waves bifurcating from the equilibrium solutions. Earlier work [20], [23] had shown the existence of standing waves in a small amplitude asymptotic limit. Current work in a similar asymptotic limit [26] shows a much larger variety of waves with nontrivial acoustics and no shocks.

Employing a spectral method numerical technique (with appropriate filtering to deal with shocks), Celentano and Rosales [1] found out that their solutions of the Euler equations appear to be asymptotically stable when subject to rather small but finite disturbances. However, Celentano and Rosales could not explore problems with "arbitrary" initial conditions because of the limitations of spectral methods: They perform amazingly well when applied to nice smooth solutions, but do poorly when shocks and contact discontinuities arise. Their scheme deals well with the weak shocks that result from small initial disturbances, but numerical filters fall short of the correct treatment of strong shocks. *A study of the situation with "arbitrary" initial conditions is very important, as it will show to what extent the "standing" acoustic waves of Celentano and Rosales are "attractors" for the full set of equations.*

It is the purpose of this thesis to find the answer to the following question: Would "arbitrary" initial conditions asymptote these standing waves in the long time limit? For example, consider a close ended tube with quiescent gas inside it. Several membranes placed in the tube enable piecewise constant pressure and density profiles. The membranes are then suddenly removed so shocks and other waves quickly develop and begin to interact with each other and the end walls. After a long time, in the order of several hundreds of acoustical periods, all shocks will die. What will the

flow look like by that time? Will any motion die out with the shocks, with the flow degenerating to quiescent gas with a uniform pressure distribution and, at most, some entropy variations? Or, maybe, standing waves, similar to those reported in [1], will appear? (At least before they are dissipated by the viscosity in the gas).

To answer those questions, we carried out numerical integration of the Euler equations in a bounded domain with reflecting boundaries and "arbitrary" [1] initial conditions [2]. The integration was conducted for a long time (thousands of acoustical periods). In **all** our experiments, regardless of the particular initial conditions being used, the fully nonlinear acoustical standing waves always emerged in the flow after a sufficient time. These waves have the remarkable property that they do not develop shocks at any time (even though they bear large pressure variations).

The computational simulations revealed a typical scenario for the flow evolution. By the end of an initial — very short (only a few acoustical periods) — stage of intense interaction, the main features of one of these waves are already developed. Throughout the second stage, which lasts for several hundreds acoustical periods, all shocks decay. By the end of this second stage, the standing wave completely dominates the flow. After that, during a last stage, the standing wave oscillates entirely unchanged for a very long time – we have tracked them for thousands acoustical periods. The motion during this last stage appears to be almost periodic.

The study of the development of these new waves requires very long time, accurate, numerical simulations. Further, these waves seem to be very sensitive to the entropy variations in the flow [3]. Thus these computational experiments would not be possible without an extremely robust code that minimizes spurious entropy production and other numerical noise [4]. To produce this code we had to develop some new numerical techniques and improve upon some other old ones. This work is also reported in this thesis.

Because of the presence of strong shocks in our problem, and since we needed rather detailed information about the flow, our choice for the "basic numerical engine" naturally felt on conservative, upstream, 2nd order nonlinear schemes (so called Godunov-type schemes) — which appear to be the best schemes known for this type of calculation. We implemented and tested several schemes of this class described in the literature. For example: we used Eulerian and Lagrangian coordinate approaches, various techniques of reconstruction (to avoid spurious oscillations), different kinds of Riemann solvers, etc. We also tried schemes utilizing shock tracking, moving grid, and multi-grids concepts.

All schemes have some basic common flaws, which showed up in our tests of the various codes: They produced too much numerical parasitic entropy in various situations. The first one occurs

[1] piecewise constant pressure and density profiles, zero velocity distribution

[2] a model of the tube problem from the previous paragraph

[3] The entropy background is crucial — according to [1], without it these solutions could not occur. As Celentano and Rosales reported in [1], even very weak variations in the entropy can have substantial effects.

[4] Too much numerical viscosity will just "kill" everything over a long time calculation!

when two shocks of about the same strength collide, or when a shock reflects from a wall. A similar production of spurious oscillations — mostly entropy waves — manifests itself when a shock moves with respect to a grid. The second situation takes place when a strong rarefaction wave is generated.

These problems are well known, though it has not generally being pointed out how big the generated noise can be (as some of our examples show). For our purposes this is a very troublesome flaw that needed to be fixed. A brief review of prior work on this issue is included in subsection 4.2.8 One can now pose a related numerical problem, arising both from the significance of entropy variations in producing strong acoustic couplings (as discussed in [1] and already mentioned above) and the tendency of standard codes to produce spurious entropy: If one carries out a long time numerical simulation using a regular Godunov-type scheme[5], then one can easily get into trouble. Because the spurious entropy waves will interact nonlinearly (and very strongly, in fact — see footnote 3) with the acoustic waves, one can end up with a computed flow which has nothing to do with the true one. In many problems involving the computation of flows involving vorticity, shocks and acoustical waves, this could be a serious issue. Then a code [6] that minimizes noise production can be used to provide "benchmark" tests against which the performance of other codes (and the magnitude of this problem we just mentioned) can be measured.

As pointed out above, since regular Godunov-type schemes could not be trusted with the long time simulation we needed, it was necessary to develop and write a special one-dimensional code, where the production of parasitic numerical entropy and other noise would be minimized and put under control.

To develop the code, it was first necessary to understand in detail the mechanisms governing the generation of parasitic entropy waves and other noise [7]. We concluded that: The production of the spurious entropy at the collision of two shocks of about the same strength (or when a shock travels through a numerical grid) is due to the finite size of the transition layer of a shock captured by a Godunov type scheme. A numerical shock has a width of a few (two or three) mesh points — thus, when "zoomed up", the shock looks like a sequence of smaller jumps rather than one large step. When an interaction of two numerical discontinuities occurs, the step-like structure of the shocks results in a sequence of collisions which take several time steps rather than one (as in the instantaneous collision of two large shocks). Such spread out in space and time interaction can produce lots of parasitic entropy (any smearing done by a code amounts to production of numerical entropy). If the two interacting shocks are of different strength, then the spurious entropy will be washed away by the mean flow. But if the shocks are of about the same strength, the flow velocity

[5]For that matter, all the currently known conservative numerical algorithms that can deal properly with shocks seem to have the same problem of spurious entropy generation.

[6]Even a special purpose one, such as the one we developed, hard to generalize outside the realm of one-dimensional Gas Dynamics

[7]We stress that this problem has also been studied by others. A brief review of prior work on this issue is included in subsection 4.2.8

between the shocks will be very small, and the parasitic entropy will stay and interact with other waves. A similar reasoning explains the spurious entropy waves in unsteady computations of moving shocks. Namely: The reason for this spurious numerical behavior is that the shock transition layer alternates between being thick and thin as it passes through the mesh. Finally, the source of the parasitic entropy produced by a strong rarefaction wave (as it is generated by, say, a shock collision) can be found in the initial averaging of the wave (which must begin being very sharp) which is not sufficiently resolved — again: smearing equals creation of entropy.

Once the mechanisms which trigger the spurious waves are understood, one can begin devising strategies to eliminate the problem. This we were able to do. In our scheme we employ the following concepts: (**i**) On every time level the grid nodes which can produce rarefaction waves or discontinuities with strength exceeding some threshold value (which choice depends on what resolution we expect from a particular numerical simulation) are found. (**ii**) Next, utilizing the underlying fixed mesh/boxes concept [8], a relay type "tracking" of the found strong discontinuities is used during the time step [9]. (**iii**) A strong rarefaction wave is treated as follows: The characteristic fan associated with it is sliced along its — initially — straight characteristics. Then a fine grid is superimposed, which stays until the rarefaction wave is adequately resolved. (**iv**) The time step is always chosen to satisfy the CFL condition *and* to prevent the collision of two neighboring discontinuities in mid-step. In fact, when two neighboring discontinuities are about to collide, the time step is picked precisely so that they reach each other strictly at the end of it. This allows the code to treat collisions of discontinuities exactly [10] (within the error of the method). We would like to emphasize that all these and other modifications hardly slow down the performance of the scheme. Compared with a regular Godunov type scheme, the code runs less then 10% slower.

The code was tested on standard one-dimensional problems with known exact solutions. Its performance is extremely good: With very few mesh points, the exact solution is reproduced with very little noise and high accuracy — the scheme is actually second order accurate everywhere, including shocks (that is, it provides L^1-norm second order accuracy).

The plan of this thesis is as follows: In **Chapter 2** we list some elementary facts about the Euler equations of Gas Dynamics and general hyperbolic systems of conservation laws which we will need throughout the thesis. In **Chapter 3** we review the main results on the non-breaking wave solutions of the Euler equations from past work . It starts with the results in weekly nonlinear acoustics, next it covers resonant waves in the full set of Gas Dynamics equations, and finishes with a review of the question of stability for the resonant waves without shocks (nonlinear standing acoustic waves).

[8] This idea was originally introduced by Harten and Hyman in [12].

[9] Namely, grid nodes are moved so that a discontinuity is always at a grid node. The "relay" name comes because a node is constrained to move within its box, and when approaching the end of the box, "passes" the discontinuity to the node of the next box

[10] unlike the original implementation by Harten and Hyman [12] where tracking was stopped near collisions, which thus were "blurred"

Chapter 4 is dedicated to the study of current numerical schemes and their problems. It begins with a formulation of the requirements which a numerical scheme — to be used for a long time numerical simulations aimed at the observing non-breaking waves — should satisfy. Then the class of Godunov-type schemes are reviewed. Next we present the outcomes of a series of numerical experiments which reveal the deficiencies of these type of schemes — numerical noise generation by moving and interacting discontinuities. An analysis of the spurious waves is carried out and the source of the trouble is pinpointed. Next we review why these behaviors are unacceptable for our purposes. The chapter ends with a review of past work studying the production of this kind of numerical noise by various schemes. In **Chapter 5** we develop our new method, which is free of the shortcomings of regular Godunov-type schemes. The underlying fixed mesh/boxes concept of Harten and Hyman is reviewed. The new Discontinuity Treatment (DT) scheme, which is *uniformly*, including shocks, L^1-norm *first* order accurate and provides for a perfect (one node) resolution of strong discontinuities and their interactions, is developed. Towards the end we extend the DT scheme to the Discontinuity Rarefaction Treatment (DRT) scheme, which is now *uniformly*, including shocks, L^1-norm *second order* accurate. This scheme provides one-node-wide representation of strong discontinuities and their interactions and minimal-noise resolution of rarefaction waves. In **Chapter 6** we report our numerical results on the long time numerical integrations of the Euler equations in a bounded domain with arbitrary piecewise constant initial conditions. The DRT scheme was used for these calculations. In all experiments, non-breaking acoustic nonlinear standing wave eventually appear. This suggests a strongly attractive character for these waves. The properties of these new waves are studied and reported. Observations of the time evolution of the new solutions in the Fourier space reveal a remarkable feature of those waves: their trajectories are confined to a low (usually two, a torus) dimensional manifold of the infinite dimensional Fourier space. We remark that the solutions constructed by Celentano and Rosales [1] were periodic in time, while we observe a slightly more general set of solutions without shocks in our numerical calculations: apparently quasiperiodic with two periods. Similar observations are reported in [26]. In **Chapter 7** the conclusions are drawn and some possible topics for future research arising from this work are outlined.

Chapter 2

Some Preliminary Facts and Notations

In this chapter we introduce some basic facts and definitions we will be using throughout this thesis.

2.1 The Euler Equations of Gas Dynamics

We consider the problem of gas motion inside a close ended rigid pipe with circular cross-section of constant radius. We will assume that the motion is essentially one-dimensional and inviscid everywhere inside the pipe. The influence of the thin boundary layers at the walls will be neglected. These assumptions are valid as long as we are interested in times t which are small compared with a typical dissipation time scale. We will also neglect energy loses due to the pipe not being rigid (radiation). With these assumptions, the problem is described mathematically by the one-dimensional (inviscid) Euler Equations of Gas Dynamics in a bounded domain, with reflecting boundary conditions at the ends.

2.1.1 Equations

The Euler equations governing inviscid one-dimensional compressible Gas Dynamics constitute a nonlinear hyperbolic system of three conservation laws. In Eulerian coordinates they can be written as [4]:

$$\begin{aligned} \rho_t + (\rho v)_x &= 0, \\ (\rho v)_t + (\rho v^2 + p)_x &= 0, \\ (\rho e + \frac{1}{2}\rho v^2)_t + ((\rho e + \frac{1}{2}\rho v^2 + p)v)_x &= 0, \end{aligned} \tag{2.1}$$

where the independent variables are the time t and the spatial coordinate x. The mass density, flow velocity, pressure, and specific internal energy per unit mass are ρ, v, p, and e, respectively.

To close this system of equations, we need the equation of state. We will consider an ideal gas with constant specific heats (polytropic gas). Then the equation of state can be written as

$$e = \frac{p}{(\gamma - 1)\rho}, \tag{2.2}$$

where $1 < \gamma < 2$ denotes the ratio of specific heats.

If we use (2.2) to eliminate e, we end up with a 3×3 system of equations. We will call the set of unknown functions (ρ, v, p) the *primitive* variables.

Throughout this thesis we will be also using another set of variables (ρ, m, E), where m and E denote the momentum and total energy per unit volume, respectively:

$$m = \rho v, \qquad E = \rho e + \frac{1}{2}\rho v^2. \tag{2.3}$$

We will call the latter set of variables the *conservative* variables. The name is obvious since the Euler Equations (2.3) represent the conservation of mass, momentum, and energy, respectively.

We close this section introducing two more functions, which will be frequently used in this thesis: the speed of sound, defined by

$$c = \sqrt{\gamma p/\rho}, \tag{2.4}$$

and the specific entropy, defined, up to an additive constant, by

$$S = c_v \log(p/\rho^\gamma) + constant, \tag{2.5}$$

where c_v is the specific heat at constant volume.

2.1.2 Boundary and Initial Conditions

Most of the time, we will choose the length of the tube L as the basic length scale of the problem. Then the reflecting boundary conditions at the close ends of the tube are

$$v(0,t) = 0, \qquad v(1,t) = 0. \tag{2.6}$$

This close ended problem is *equivalent* to a periodic (in the space variable x) problem with the period twice the tube length. This can be seen by extending the solution satisfying the boundary conditions above to the whole line by "reflecting" recursively on the boundaries: v as an odd function, p and ρ as even functions. Vice versa, if one has a periodic solution with the period twice the tube length, such that v is odd, p and ρ are even, then it reduces to a solution of the problem above when restricted to the tube. Sometimes, it will be convenient to replace the original close ended problem by this *equivalent* periodic problem.

In this thesis we will mostly concentrate on initial conditions with vanishing flow velocity and piecewise constant pressure and density distributions. It corresponds to the physical set-up where, in the initial state, the gas in the tube is at rest and several membranes are placed inside the tube, separating sections of constant pressure and density. These constant values can change from section to section with pressure and density jumping across a membranes.

The locations of the membranes can be arbitrary. The initial constant values of the pressure and density can be any arbitrary non-negative numbers.

2.2 General Set-up

In this section we recall some general facts of the theory of weak solutions for general hyperbolic systems of conservation laws, and how they apply to the Euler Equations we are interested in.

With the vector notations

$$u = \begin{pmatrix} \rho \\ m \\ E \end{pmatrix} \quad \text{and} \quad F = \begin{pmatrix} m \\ \frac{m^2}{\rho} + p \\ \frac{m}{\rho}(E+p) \end{pmatrix}, \tag{2.7}$$

the Euler equations (2.1) can be written as a special case of a general hyperbolic system of conservation laws. So, the problem formulated in section 2.1 can be considered as the initial [1] value problem for a general hyperbolic system of conservation laws:

[1] We can always, when we need it, consider an initial value problem on an infinite domain with periodic initial conditions, because of the equivalence of the reflecting and periodic boundary conditions for our case (see subsection 2.1.2).

$$u_t + f(u)_x = 0 \tag{2.8}$$
$$u(x,0) = u_0(x),$$

where u, called the state or density vector, is an m-column vector function of one space dimension and time, and the flux function f is an m-component vector-valued function of u.

The system above is hyperbolic if the Jacobian

$$A(u) = \frac{\partial f(u)}{\partial u} \tag{2.9}$$

has m real eigenvalues

$$\lambda_1(u) \leq \lambda_2(u) \leq \cdots \leq \lambda_m(u) \tag{2.10}$$

with a complete set of eigenvectors.

Because the system (2.8) is hyperbolic and nonlinear, it can develop discontinuities even from smooth initial data. Solutions with discontinuities are weak solutions that satisfy (2.8) in the sense of distributions. Across each discontinuity the Rankine–Hugoniot relations

$$f(u_R) - f(u_L) = s(u_R - u_L) \tag{2.11}$$

must hold. Here s is the speed of propagation of the discontinuity, and u_L and u_R are the states on the left and right, respectively. These relations guarantee the conservation of u.

However, generally the constraints (2.11) are not adequate since they allow too many discontinuities (making the solution to the problem (2.8) non unique.) To get rid of the ambiguity, we assume that the system of conservation laws has an entropy function [16]. That is, a convex scalar function $U(u)$ and a corresponding entropy flux $F(u)$ exist such that all smooth solutions u of equation (2.8) satisfy

$$U(u)_t + F(u)_x = 0. \tag{2.12}$$

Then, for weak solutions (with discontinuities), the entropy inequality

$$U(u)_t + F(u)_x \leq 0, \tag{2.13}$$

(in the sense of distributions) is required, in addition to the Rankine–Hugoniot relations (2.11) above. This selects the ***physically relevant*** weak solutions of (2.8). Thus, ***admissible*** discontinuities in weak solutions of equation (2.8) satisfy the Rankine-Hugoniot condition (2.11) and the entropy jump inequality:

$$F(u_R) - F(u_L) \leq s(U(u_R) - U(u_L)). \tag{2.14}$$

For the Euler equations (2.1) we have the following expressions for the entropy and its flux:

$$U = \rho S, \qquad F = \rho v S. \tag{2.15}$$

The restrictions above on discontinuities then physically mean that the mass, momentum, and energy of a fluid parcel are conserved (Rankine–Hugoniot conditions) as it crosses a shock front, while its entropy increases.

Chapter 3

Non-breaking Wave Solutions

Consider the initial value problem posed in section 2.1. When all membranes are suddenly removed at time $t = 0$, motion results, with several shocks, contacts, and rarefaction waves emerging from the initial discontinuities. All these waves begin to collide and interact with each other and the end walls, generating new waves, which, in turn, will collide and interact, and so on. Even though the Euler Equations are inviscid, as long as there are shocks in the solution, dissipation occurs. The question then arises: What happens to the flow after a "long time", when all the shocks have "decayed" ? The "traditional" expectation was that any pressure variations in the solution would lead to shock formation and further dissipation. Thus the shocks would completely dissipate the acoustical component, so that the solution would limit to one which is a "pure entropy wave" [5]; that is: a solution where the pressure is constant throughout (then the flow velocity also vanishes and only entropy variations can remain.) However, the results in [1, 20] show that this does not need to be so.

3.1 Weakly Nonlinear Asymptotic Analysis

This question was first studied by Majda, Rosales, and Shonbeck [20], in the weakly nonlinear regime (where the initial data differs little from a constant). In [19] it is shown, through a multiple-scale analysis, that in the case of a weakly nonlinear one-dimensional compressible flow with periodic boundary conditions [1], the time evolution at leading order is governed by a 2×2 system of integro-differential equations of the form

[1] Without loss of generality we can normalize the period to 2π.

$$\alpha_t + (\frac{1}{2}\alpha^2)_x + \int_0^{2\pi} K\left(\frac{x+y}{2}\right)\beta(y,t)dy = 0,$$

$$\beta_t - (\frac{1}{2}\beta^2)_x - \int_0^{2\pi} K\left(\frac{x+y}{2}\right)\alpha(y,t)dy = 0. \quad (3.1)$$

In these equations $\alpha(x,t)$ and $\beta(x,t)$ are the amplitudes of the right and left moving acoustical modes, respectively, while $K(x)$ is the spatial derivative of the entropy, which does not evolve in time in this limit and if the background velocity is zero. The variable x has different meanings for each of the waves; if X and T are the "laboratory" coordinates and c is the sound speed, then: for α, $x = X - cT$, for β, $x = X + cT$, and for K, $x = X$. Finally, t is a "slow time", $t = \epsilon T$, where ϵ is a measure of the small wave amplitude.

The 2×2 system above, which consists of two Burgers' equations coupled by a linear integral term, was studied numerically in [20]. The results reported there show drastically different behavior of this asymptotic system from that of the Burgers' equation. In many cases, after shocks had formed and "decayed", strikingly periodic (in time) wave-trains emerged. This surprising outcome can be explained because the interaction of the acoustic waves coupled through an entropy wave background is dispersive (as shown by the form of the linear coupling term in the equations above). Thus it can balance the nonlinear terms tendency to break the waves. Exact solutions exhibiting this balance for these asymptotic equations are computed in [23] and [20]. A more extensive and detailed study of the behavior of the system above is currently being carried and is reported in [26]. In this thesis we will show that the behaviors observed in the context of the simplified asymptotic equations above, carry over to the full system of the Euler Equations.

3.2 Full Set of Gas Dynamic Equations

Motivated by the *asymptotic* results outlined above for the weakly nonlinear acoustics regime, Celentano and Rosales [1] studied the possibility that the *full* set of Gas Dynamic (the Euler) equations might support non-breaking solutions with nontrivial finite amplitude acoustical components.

Employing bifurcation and other perturbation arguments, Celentano and Rosales showed that solutions with a very strong acoustical component and no shocks ever are possible even with rather small entropy variations. They showed this by producing solutions of the equations periodic in time. This contradicts the intuitive idea — mentioned at the beginning of this chapter — that pressure variations (acoustic component) should produce compression waves that break and form shocks. But — as mentioned in section 3.1) — resonant, nonlocal coupling with entropy variations can, in fact, disable this effect.

Celentano and Rosales found a remarkable new class of solutions to the one-dimensional Euler Equations. These solutions have the interesting property that they do not develop shocks at any time, although they have an important acoustic component. The solutions are given as finite amplitude, continuous, time and space periodic nonlinear acoustic waves propagating on a nonuniform entropy background. Both acoustic components are present and, in fact, the solutions can be characterized as "standing nonlinear acoustic waves".

A varying entropy background is crucial, without it appears that these solutions are not possible. However, it turns out that even an extremely small amount of entropy variation is enough to produce large couplings of the acoustic modes. For example, Celentano and Rosales found that entropy variations as small as 0.1% were enough to allow for non-breaking acoustic components of intensities as large as 160 dB (fully into the nonlinear regime).

3.3 Stability Questions

Utilizing a spectral method numerical technique (with appropriate filtering to deal with shocks) Celentano and Rosales found that their solutions appeared to be asymptotically stable when subject to rather *small* but finite disturbances.

A natural follow up question is then: Are these solutions also stable with respect to large perturbations? How large? A related question is: if one starts with "arbitrary" initial conditions, what will happen to the flow after a "long time", when all shocks have "died out"? Will these new kind of waves show up? A study of the situation with "arbitrary" initial conditions is important, as it will show to what extent the "standing" acoustic waves of Celentano and Rosales are "attractors" for the full set of equations.

Unfortunately, Celentano and Rosales could not explore problems with "arbitrary" initial conditions due to the limitations of spectral methods: they perform amazingly well when applied to nice smooth flows, but do poorly when shocks and contact discontinuities arise. Their scheme deals well with the weak shocks that result from small initial disturbances, but numerical spectral filters are not that good with strong shocks.

Chapter 4

Choosing the Numerical Scheme

To deal with the problem with "arbitrary" initial conditions one needs a code which:

(I) Can handle strong shocks and contacts, and

(II) Produces so little numerical noise, that its output is still reliable after a long time calculation. Especially dangerous is numerical entropy production , because even small amount of parasitic entropy could resonantly couple with acoustic components (see chapter 3) which would lead to non-physical interactions.

In this chapter we will review some of the (rather reliable) numerical techniques in current use for solving the Euler Equations of Gas Dynamics. We will point out their good points and their limitations and thus why we felt the need to develop an improved scheme for the problem of interest to us (introduced in the prior chapters). This new scheme is explained in detail in the next chapter.

4.1 Godunov-type Schemes

Once one needs to carry out a numerical simulation of compressible Gas Dynamics, one will immediately encounter the problem of representing of the shocks and contact discontinuities which will arise in the simulation. There are two main reasons why an adequate representing of the discontinuities of the flow is so important. **Firstly**, the overall accuracy of the simulation is very closely related to the accuracy with which flow discontinuities are represented: If a discontinuity is not resolved properly, it will produce numerical noise which will propagate into the smooth regions, contaminate the solution. It will also interact with the flow, and produce more numerical noise as a result. Therefore, one will end up with the simulation results of very poor accuracy, which may have very little to do with the real physical flow. **Secondly**, the flow discontinuities themselves – their generation, propagation, interaction, and destruction – in their own right, are of great theoretical and practical importance.

We can write out the essential properties required for a numerical method being used for Gas Dynamics flow simulations:

1. Calculates correct positions and speeds for discontinuities.
2. Avoids non-physical discontinuities (e.g., rarefaction shocks).
3. Produces sharp profiles for the shocks and contacts.
4. Avoids producing superfluous oscillations at the shocks and contacts.
5. Achieves high accuracy in the smooth regions of the flow.

In this section we will briefly describe a family of numerical methods, the Godunov-type schemes, which satisfy the Requirements introduced above. Emphasis will be made on the ability of those schemes to treat flow discontinuities. We will start with a collection of the definitions and concepts which will be used throughout this thesis.

4.1.1 Some Definitions

We consider the numerical approximation of initial value problem (2.8).

Let the set $\{\xi_{j-\frac{1}{2}}\}$ be a partition of x-axis (it does not need to be uniform). We will call this partition a computational **grid**. A point $\xi_{j-\frac{1}{2}}$ will be called the j-th computational **node**. The set $\{I_j\}$ denotes the intervals defined by the partition $\{\xi_{j-\frac{1}{2}}\}$

$$I_j = (\xi_{j-\frac{1}{2}}, \xi_{j+\frac{1}{2}}] \tag{4.1}$$

An interval I_j will be called the j-th computational **cell**. We will use the notation

$$\Delta\xi_j = \xi_{j+\frac{1}{2}} - \xi_{j-\frac{1}{2}} \tag{4.2}$$

for the width of the j-th cell.

If the grid is uniform, then all its cells have the same width, $\Delta\xi$. If it is not, we still will be using some typical cell width of the entire grid, $\Delta\xi$ [1], which we will call the **cell size** of the grid. For a given computational domain, $\Delta\xi$ indicates the refinement of the grid.

Consider also a sequence of points on the time axis $t^0, t^1, \ldots, t^n, \ldots$, constructed by the time-marching procedure described below. Each time t^n defines the n-th time level in the calculation. The **time step** at the n-th time level is defined by $\Delta t^n = t^{n+1} - t^n$.

[1] One can think about it, for example, as the average over the cell widths of all the cells in the grid.

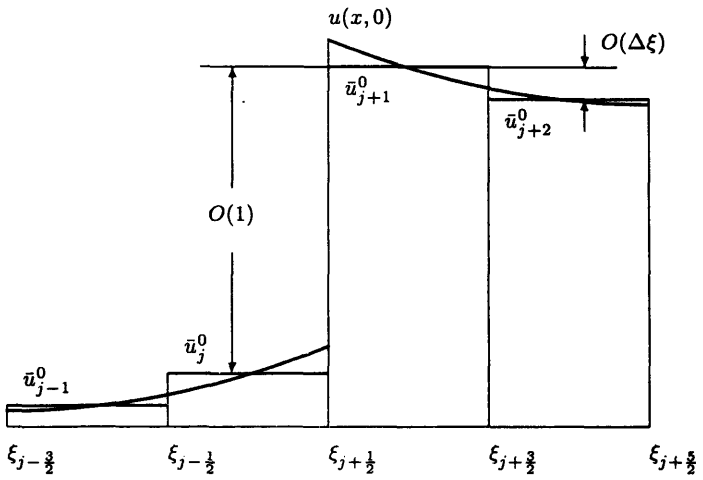

Figure 4.1: Physical vs Nimerical Discontinuities. Jumps at nodes $\xi_{j-\frac{1}{2}}$ and $\xi_{j+\frac{3}{2}}$, being introduced by discretization, are of order of $\Delta\xi$. Jump at node $\xi_{j+\frac{1}{2}}$, corresponding to the jump at $u(x,0)$, is of order of one.

For each time level t^n and each cell I_j we will define a **cell average** of $u(x,t^n)$ by

$$\bar{u}^n_j = \frac{1}{\Delta\xi_j}\int_{\xi_{j-\frac{1}{2}}}^{\xi_{j+\frac{1}{2}}} u(x,t^n)dx, \tag{4.3}$$

The numerical methods we will consider produce approximations u^n_j to $\bar{u}^n_j$ – the cell average [2]. The initial data for the system (2.8) is discretized by computing $\bar{u}^0_j$, the cell averages of $u(x,0)$. $\{\bar{u}^0_j\}$ are used as the initial data for the numerical method: $u^0_j = \bar{u}^n_j$.

Remark 4.1 Now, instead of piecewise smooth function $u(x,0)$ we have a piecewise constant function (with many more jumps). However, in the smooth regions the jumps between two neighboring cells are small (it is easy to show that the jumps are of order of $\Delta\xi$). Jumps of ($O(1)$) occur only where the original piecewise smooth data had discontinuities (see Figure 4.1).

All the methods we will consider are one-level methods. Given the initial piecewise constant

[2] In numerical methods for conservation laws it is preferable to view u^n_j as an approximation to the cell average of $u(x,t^n)$ rather than as an approximation to the pointwise value of $u(x,t^n)$ in the middle of the cell – as it is used in linear finite difference methods. This interpretation is natural since the integral form of the conservation law describes precisely the time evolution of integrals such as that appearing in (4.3).

data $\{u_j^0\}$, an appropriate [3] time step Δt^0 is chosen, and, using a time-marching procedure, the approximation $\{u_j^1\}$ is constructed. The procedure is then repeated recursively. In general, given $\{u_j^n\}$, the scheme generates a time step Δt^n and constructs $\{u_j^{n+1}\}$, which is the approximation to the cell averages of $u(x, t^{n+1})$.

4.1.2 Convergence to Physically Correct Weak Solutions – Requirements 1 and 2

A numerical method for a system of nonlinear conservation laws can converge to a function that is not a weak solution of the original equations (2.8), or that is a wrong weak solution (i.e. does not satisfy the entropy condition (2.13)).

Fortunately, there appears to be a simple and natural requirement which will guarantee that the scheme always converges to a weak solution of (2.8). This is the requirement (Lax-Wendroff theorem [17]) that the method should be in **conservation** form

$$u_j^{n+1} = u_j^n - \frac{\Delta t^n}{\Delta \xi_j}(\bar{f}_{j+\frac{1}{2}} - \bar{f}_{j-\frac{1}{2}}) \tag{4.4}$$

where $\bar{f}_{j+\frac{1}{2}}$ is a **numerical flux** consistent with a flux of system (2.8). Hence, conservative schemes automatically guarantee Requirement 1.

To secure Requirement 2, it suffices to show (Harten-Lax theorem [13]) that a discrete entropy inequality holds

$$U(u_j^{n+1}) = U(u_j^n) - \frac{\Delta t^n}{\Delta \xi_j}(\bar{F}_{j+\frac{1}{2}} - \bar{F}_{j-\frac{1}{2}}), \tag{4.5}$$

where $\bar{F}_{j+\frac{1}{2}}$ is a **numerical entropy flux** consistent with that of system (2.8) (F in (2.13)).

Throughout this thesis we will only consider schemes in conservation form, for which (4.5) holds for a suitable numerical entropy flux function. Thus Requirements 1 and 2 will be automatically satisfied, and we will concentrate on satisfying the other Requirements, 3-5.

4.1.3 Linear Finite Difference Schemes – Conflicting Requirements 3 and 4

Numerous techniques have been invented to cope with the difficulties of representing discontinuities in computer experiments. Initially, in the period 1950 – 1970, the problem of the accurate representation of flows with strong discontinuities was addressed using a brute force approach. Difference schemes were derived using Taylor series expansions for the terms in the differential equations. For

[3] The size of the time step is restricted by stability conditions.

this technique to work, it is fundamental that the solution be *smooth*. At a discontinuity this assumption is completely inappropriate. If nothing special is done about the discontinuity, the resulting difference scheme will generate spurious oscillations at the discontinuity and display other forms of non-physical behavior. Hence, one needs to *force* a well-behaved solution by introducing a non-physically large viscosity (Artificial Viscosity Methods) to smooth out the solution; or a non-physical constraint of monotonicity (Linear Hybridization Methods) [4].

These "ad-hoc" methods provided only a temporal relief from the "disease", suppressing the symptoms but not eliminating the causes of the sickness. Furthermore, they caused bad side effects: Shocks were spread out over many grid points, poorly resolved, and lots of valuable information was lost to the added dissipation.

The following conclusion could be drawn: If one tries to enforce Requirement 4 in the framework of a linear finite difference scheme, then one can not satisfy Requirement 3.

4.1.4 Reconciling Requirements 3 and 4 – Godunov's Method

The first researcher who tried to eradicate the source of the trouble itself, was Godunov. In 1959, having understood that the origin of the difficulties lies in the attempt to solve an intrinsically nonlinear and discontinuous problem using linear smooth methods, he undertook a new approach towards the problem [6]. Godunov introduced *explicit nonlinearity* into the difference method. Instead of building up a full solution to the flow with strong discontinuities by piecing together smooth, small-amplitude solutions, he built up the solution by piecing together nonlinear, discontinuous solutions. These discontinuous solutions closely approximate the smooth ones where those are appropriate, but they have the great additional advantage of approximating the true solution reasonably well *even where that solution is not smooth.*

Godunov made use of a nonlinear flow problem which is simple enough to permit an exact solution — the Riemann Problem. This simple solution describes the nonlinear flow which develops from an initial condition consisting of discontinuous jump separating two constant states [5]. Godunov's approach was to, first, approximate the (discontinuous) flow by a large number of constant states (initial data discretization), and then:

(i) Compute their interaction exactly (to do so, one just needs to solve an array of Riemann problems at the interfaces of the constant states), and

(ii) Average the results in a *conservative* fashion.

[4] To get a brief description of these methods and how they perform see the excellent review article by P. Woodward and P. Colella [35].

[5] In the case of the Euler Equations (2.1), the jump evolves into two nonlinear waves, either shocks or rarefactions, with a contact discontinuity separating them [4].

After the averaging is done, a new set of constant states is obtained, so that the process can be repeated, and so on.

4.1.5 Godunov's Method. Algorithm

Next, we give the details of the scheme outlined above.

The question is: Once we know $\{u_j^n\}$, the cell averages at time $t = t^n$, how can we find the new cell averages at time t^{n+1}? The answer to this question can be easily found, once the following fact is taken into consideration: The new cell average can be computed if the integral form of conservation laws is used. Integrating the system of conservation laws over the rectangle formed by the lines $t = t^n$, $t = t^{n+1}$, and $x = \xi_{j-\frac{1}{2}}$, $x = \xi_{j+\frac{1}{2}}$ (rectangle ABCD in Figure 4.2) gives

$$\int_{\xi_{j-\frac{1}{2}}}^{\xi_{j+\frac{1}{2}}} u(x,t^{n+1})dx = \int_{\xi_{j-\frac{1}{2}}}^{\xi_{j+\frac{1}{2}}} u(x,t^n)dx + \int_{t^n}^{t^{n+1}} f(u(\xi_{j-\frac{1}{2}},t))dt - \int_{t^n}^{t^{n+1}} f(u(\xi_{j+\frac{1}{2}},t))dt. \tag{4.6}$$

Dividing (4.6) by $\Delta\xi_j$ and using the definition of the cell average (4.3), we will obtain

$$u_j^{n+1} = u_j^n - \frac{\Delta t^n}{\Delta\xi_j}(\bar{f}(u_j^n, u_{j+1}^n) - \bar{f}(u_{j-1}^n, u_j^n)), \tag{4.7}$$

where $\bar{f}$, the numerical flux function, is given by

$$\bar{f}(u_j^n, u_{j+1}^n) = \frac{1}{\Delta t^n}\int_{t^n}^{t^{n+1}} f(u(\xi_{j+\frac{1}{2}},t))dt, \tag{4.8}$$

and is the time average of the flux through the cell boundary at $x = \xi_{j+\frac{1}{2}}$ over the time interval (t^n, t^{n+1}).

The integral in (4.8) is trivial because $u(x,t)$ is *constant* at the point $x = \xi_{j+\frac{1}{2}}$ for all times from interval (t^n, t^{n+1}). This follows from the fact that the solution of the Riemann problem at $\xi_{j+\frac{1}{2}}$ is a similarity solution, constant along each ray $(x - \xi_{j+\frac{1}{2}})/t = constant$ [4].

The constant value of $u(x,t)$ along the cell interface $x = \xi_{j+\frac{1}{2}}$ depends only on the data u_j^n and u_{j+1}^n for the Riemann problem. If we denote this value by $u^*(u_j^n, u_{j+1}^n)$, then the numerical flux (4.8) reduces to the flux of the system (2.8) computed at this constant value:

$$\bar{f}(u_j^n, u_{j+1}^n) = f(u^*(u_j^n, u_{j+1}^n)) \tag{4.9}$$

After (4.9) is substituted into (4.7), the main formula for Godunov's method reduces to

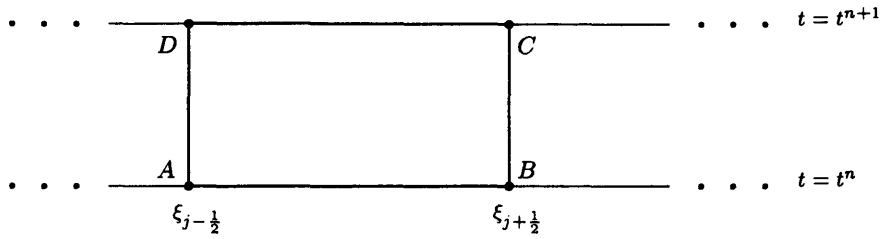

Figure 4.2: Godunov's method.

$$u_j^{n+1} = u_j^n - \frac{\Delta t^n}{\Delta \xi_j}(f(u^*(u_j^n, u_{j+1}^n)) - f(u^*(u_{j-1}^n, u_j^n))) \tag{4.10}$$

In summary, the method is as follows: At any time $t = t^n$ we have a piecewise constant function with values u_j^n defined on the j-th grid-cell. At each cell interface $\xi_{j+\frac{1}{2}}$ we have a Riemann problem with the left constant state u_j^n and the right constant state u_{j+1}^n. So, we have an array of Riemann problems, which we can solve exactly. Once we know the solution of a given Riemann problem, we can compute the flux at the corresponding interface (see (4.9)). Knowing the fluxes at the left and right interfaces of a cell, we add their difference to the old cell average — at time $t = t^n$. This gives us the new cell average — at time $t = t^{n+1}$ (see (4.10)). Thus, starting with the piecewise constant function at time $t = t^n$, we construct a new piecewise constant function at the new time $t = t^{n+1}$, and the process can be repeated to advance the solution in time. Note that the method is explicit in time.

4.1.6 Godunov's Method. Choosing Time Step

To finish with the description of the method, only one more important thing should be added. Nothing has been said yet about the choice of the time step Δt^n. It can not be taken arbitrarily large, since for large Δt^n the solution will not remain constant at $\xi_{j+\frac{1}{2}}$ because of the *interaction* with the waves arising from the neighboring Riemann problems at $\xi_{j-\frac{1}{2}}$ and $\xi_{j+\frac{3}{2}}$. However, since the wave speeds are bounded by the eigenvalues of $A(u)$ (see (2.9)) and the neighboring Riemann problems are distance $\Delta \xi_j$ away, $u(\xi_{j+\frac{1}{2}}, t)$ will remain constant over the interval $[t^n, t^{n+1}]$ provided Δt^n is sufficiently small. To be precise, the time step should satisfy the CFL condition[6]

[6]This condition, an acronym for Courant, Friedrichs, and Lewy, happens to be also a necessary condition for stability of linear finite difference methods (see [24]).

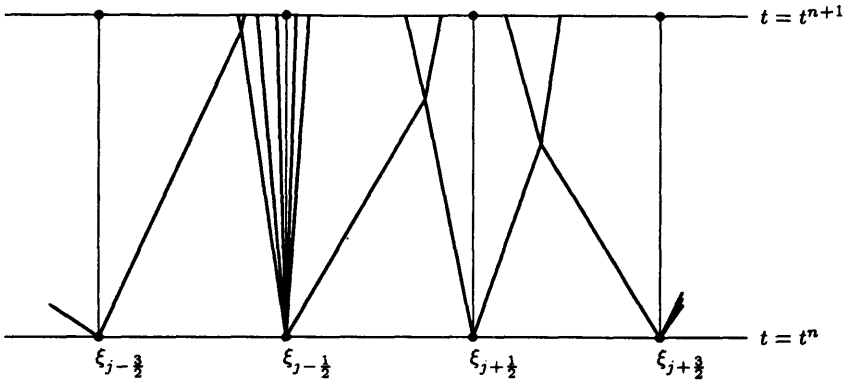

Figure 4.3: CFL condition.

$$|\frac{\Delta t^n}{\Delta \xi_j}\lambda_l(u_j^n)| \leq 1 \tag{4.11}$$

for all eigenvalues λ_l of $A(u_j^n)$ at each u_j^n.

Note that (4.11) allows the interaction of waves from neighboring Riemann problems during the time step, provided the interaction is entirely contained within a grid cell — none of the waves reaches the opposite cell interface during that time (see Figure 4.3). What we need is just to compute fluxes at the interfaces between cells, and for this purpose we do not care about what happens inside a cell as long as the solutions at the cell boundaries are not affected and remain constant.

4.1.7 Godunov's Method. Final Remarks

Godunov's procedure leads to a quite accurate treatment of the flow discontinuities — both Requirements 3 and 4 are satisfied. We should emphasize that a narrow representation of discontinuities is possible by building into the numerical method **knowledge of the propagation and interaction of nonlinear waves**. This knowledge is built into the method in the form of the Riemann solver. The Riemann solver computes the *nonlinear interaction* of two constant states, and tells us what nonlinear waves emerge from this interaction [7]. This is a way the nonlinearity is incorporated into the method.

In conclusion, we see that Godunov's method makes use of nonlinear wave propagation information within the framework of a conservative scheme.

We note that (4.7) shows that Godunov's method can be written in conservation form. It can also be shown that, if a numerical entropy flux is defined by $\bar{F}_{j+\frac{1}{2}} = F(u^*(u_j^n, u_{j+1}^n))$, then the

[7] When the jump between two neighboring states is small ($O(\Delta\xi)$), the scheme reduces to a regular upwind scheme. On the other hand, when the jump is large ($O(1)$) — which means that the flow there is actually discontinuous — the Riemann solver yields a more reliable result than computations based upon a smooth flow model (as in the linear finite difference schemes).

discrete entropy inequality (4.5) holds. Thus Requirements 1 and 2 are met (see subsection 4.1.2).

4.1.8 Higher-Order Generalizations – Meeting Requirement 5

In spite of all the advantages described above, Godunov's scheme is only first order accurate, giving poor accuracy in the smooth regions of the flow — Requirement 5 is not met. Moreover, shocks tend to be heavily smeared and poorly resolved on the grid. In truth, Requirement 3 is not fulfilled in a very satisfactory way. These effects are due to the large amount of numerical dissipation a stable first order scheme will unavoidably have.

Starting from the late seventies, there were numerous attempts to extend Godunov's ideas to the higher order schemes [2, 3, 8, 9, 11, 14, 15, 28, 29]. All these methods consist of three main steps, which are similar to those of Godunov's scheme (see subsection 4.1.4). We will describe these steps for a second order accurate scheme [8].

Suppose that, at time level t^n, we are given the set of cell averages $\{u_j^n\}$ – the flow is approximated by a large number of constant states.

(i) RECONSTRUCTION. Construct a piecewise linear approximation to the solution from the piecewise constant data. At each cell, the densities are thus approximated by linear polynomials, using the constant data at the cell itself, as well as that of the neighboring cells to produce the needed two parameters (mean and slope). This approximation will be second order accurate. Note: it is important (for conservation) that the cell averages of the approximation equal those of the piecewise constant data.

(ii) FLUX CALCULATION. At each cell interface solve the Generalized Riemann Problem — compute the interaction of the neighboring states (which are linear) up to second order of accuracy [9]. Then use the solution of the Generalized Riemann Problem to compute the numerical flux (again, the flux must be computed up to the second order of accuracy).

(iii) UPDATE THE CELL AVERAGES. Average the results in a *conservative* fashion — at each cell add the difference of the numerical fluxes at the cell interfaces to the average density.

After the last step, a new set of constant states is obtained, and the sequence can be repeated to advance in time step by step.

The main difference of this outline from the one for Godunov's scheme (see subsection 4.1.4) is the new step — reconstruction. Unfortunately, this turns out to be a most difficult task if one wants to avoid paying a heavy price for the extra accuracy (generation of oscillations by discontinuities).

[8] Generalizations to r-th order accurate schemes are straightforward (at least in principle, even if rather messy and generally not too useful in practice).

[9] When two neighboring states are constant (i.e., Godunov's scheme), they yield a standard Riemann problem for which an exact solution can be found. When the two neighboring states are linear polynomial functions, we can not, in general, express the result of their interaction in a simple closed form. Nevertheless, we can obtain asymptotic approximations for this solution ($\Delta t^n, \Delta\xi$ being the small parameters) to any desired order of accuracy.

Let us consider the process of approximating a (scalar) function by a piecewise linear function from knowledge of the cell averages (a piecewise constant function) — for a vector valued function we can apply the procedure to each component separately. Then, for each cell one should approximate the function by a line, which implies that the reconstruction of two parameters is required. The parameters could be, for example, the value that the linear function assumes in the middle of the cell and the slope. For the whole approximation to be second order accurate, the mid-cell value must be reconstructed up to second order, while for the slope only first order accuracy is needed. It is clear that the mid-cell value coincides with the given cell average up to second order. The only thing left to do then is to obtain a first order accurate approximation to the slope. For this purpose one can choose either forward or backward divided differences. In either case the value from a neighboring cell (either right or left) will be used. Either will work perfectly well as long as both cells (the cell where the slope is needed and the neighbor providing the data for the divided difference) belong to the same region of smoothness. On the other hand, if the common interface of the two cells is a discontinuity point, the constructed divided difference has nothing to do with the slope we are looking for. The approximation fails; the calculated slope will disagree drastically with the slopes in neighboring cells from the same region of smoothness. The result will be that spurious numerical oscillations will be generated near the discontinuity — thus Requirement 4 will be violated.

When interpolation across a discontinuity occurs, spurious oscillations will be generated. Hence, in attempting to extend Godunov's approach to a higher order method, one needs to take special care with the reconstruction step. A non-oscillatory reconstruction technique is needed [10]. There are two principal ways to address this problem. **Firstly**, use a high order method, but modify the method by increasing the amount of numerical dissipation in a neighborhood of the discontinuity [11]. In these methods interpolation across a discontinuities will occur. However, no oscillations appear for they are killed by the extra amount of numerical viscosity of the vicinity of the discontinuity. Basically: any oscillations that may occur are suppressed by "brute force" (direct intervention via limits on what the interpolating reconstruction can do). **Secondly**, attempt to forbid interpolation across discontinuities [12]. This is done by trying to choose interpolation stencils that do not cross discontinuities [13].

In these ways, high order Godunov-type schemes were devised and successfully used in many simulations of discontinuous flows. The methods do an accurate and well behaved treatment of discontinuities. Numerical shocks in these schemes are very narrow — usually no more than two to

[10] In some cases, small fully controlled oscillation can be allowed [11].

[11] e.g.: flux- and slope- limiter methods (see the review in [18]).

[12] e.g.: ENO schemes [11]

[13] In the example above, when we need to construct a linear interpolant, we can choose between two stencils — backward or forward. If the left interface of the cell is a discontinuity point, then we opt for the forward difference stencil. If, on the other hand, the right interface is a discontinuity point, then we choose the backward difference stencil. In the case of a quadratic interpolation (for a third order scheme), there are three stencils to choose from; in case of a cubic, there are four stencils, and so on.

three points. Furthermore, the smooth regions of the flow are resolved with high accuracy. Thus all five Requirements are satisfied [14].

4.2 Godunov-type Schemes. Shortcomings

We implemented and tested several schemes of the Godunov class described in the literature. For example: we used Eulerian and Lagrangian coordinate approaches, various reconstruction techniques, different kinds of Riemann solvers, etc. We also tried schemes utilizing shock tracking, moving grid, and multi-grids concepts.

While testing these codes on various problems, it became obvious that all the schemes we used had some very basic common flaw [15]: They produced too much numerical noise — mostly parasitic entropy waves — in various situations. For example, when a single shock is moving with respect to the grid, oscillations are generated. A similar production of numerical noise occurs when two shocks of about the same strength collide, or when a shock reflects from a wall. Problems arise also in situations where a strong rarefaction wave is generated (see subsection 5.3.6).

Note: Most of these problems have been pointed out in the literature (see 4.2.8 where a review of prior work is included). But the rather large size the produced errors can take under some circumstances (which, unfortunately, we must face in our problem) is rarely been pointed out. The computations that follow illustrate the problems and our need to device strategies to eliminate this numerical noise [16].

4.2.1 Moving Shock. Initial Boundary Value Problem

Consider the following problem: A shock separating two constant states u_L and u_R moves down a duct at a constant speed s [17]. To be specific, let us suppose that the shock is a "left" shock: $p_L < p_R$.

To be able to carry out a numerical simulation of this problem, we shall reformulate it as an initial boundary value problem. The choice of the initial conditions is obvious:

$$u(x,0) = \begin{cases} u_L & \text{if } x < x_{s,0}, \\ u_R & \text{if } x > x_{s,0}, \end{cases} \tag{4.12}$$

[14] Rigorous analysis assuring this is so (e.g.: total-variation stability, convergence) is mostly only available for the scalar, one-dimensional (nonlinear) case. However, numerical experiments for systems of equations give very good results.

[15] A very troublesome one for the purposes of the type of calculation we needed to do, as described in the prior chapters.

[16] For which we must learn first, precisely, what are its sources.

[17] u_L, u_R, and s must satisfy the Rankine-Hugoniot relations (2.11).

where $x_{s,0}$ is the shock position at time $t_0 = 0$.

The situation with the numerical boundary conditions turns out to be a little more complicated. Because we consider a left shock, the characteristic corresponding to the left sound wave always goes into the shock. Therefore, this characteristic always emanates out of either boundary. However, the direction of the other two characteristics depends on the flow velocity. In some situations, such a characteristic goes into a boundary, and in some, out of it. Thus, at either boundary, we need to prescribe only pieces of data corresponding to the outgoing characteristics, and absorb the data moving on the in-going characteristics without reflecting them back into the computational domain. We accomplish it by imposing absorbing boundary conditions [33] at both boundaries.

4.2.2 Moving Shock. Typical Simulation Outcome

We carried out numerical simulations over a wide range of shock speeds and Mach numbers. As we said earlier, different Godunov-type schemes were tested. The results of a typical calculation is shown here.

A second order Godunov-type scheme with an exact Riemann solver and van Leer type limiters applied to pseudo Riemann invariants was used on a 500-cell grid. A shock with Mach number 1.1 (which corresponds to the pressure ratio across the shock 1.245) is moving to the left at a speed of 0.04. The initial (time $t_0 = 0$) location of the shock is $x_{s,0} = 0.1$. Figures 4.4 and 4.5 show the entropy, pressure, and density profiles at time $t_1 = 0.5$, after the shock moved 0.02 units to the left, to $x_{s,1} = 0.08$.

Here and throughout the thesis we plot the computed solution against the exact solution (when the latter is known). Dashed lines represent the exact solutions, while either points or solid lines are used to depict the computed results. The numerical entropy is measured by the quantity (cf. (2.5))

$$S = \frac{p}{\rho^\gamma}. \tag{4.13}$$

Although there were no oscillations in the flow initially, once the shock begins to move, spurious waves quickly develop. These oscillations are seen primarily in the entropy [18]. The pressure seems unaffected in these figures.

One can estimate from Figure 4.4 that the amplitude of the spurious oscillations in the entropy is about 50% of the entropy jump at the shock. Figure 4.4 reveals two types of parasitic waves in the entropy: A wave train and a leading bump with a magnitude approximately 1.5 times larger than that of the wave train immediately behind it. Both the wave train and the bump are entropy waves, hence they move to the right with the flow velocity

[18] Or the density, where they are very small relative to either the mean or the jump across the shock.

$$\lambda_{2,R} = v_R. \tag{4.14}$$

In this experiment, $\lambda_{2,R} = 0.80$. At time t_1 the position of the bump and the front of the wave train can be computed using the simple formula:

$$x_{2,1} = x_{s,0} + \lambda_{2,R}(t_1 - t_0) \tag{4.15}$$

In this case we get $x_{2,1} = 0.50$, what can be confirmed by the inspection of Figure 4.4.

It seems that there are no spurious waves in the pressure. However, if we increase the resolution in Figure 4.4, then numerical oscillations similar to those in the entropy are observed. Figure 4.6 displays both types of spurious behaviors for the pressure and density: wave trains and a bumps (actually, dents here, which are about 1.5 times larger than the wave train amplitudes).
The pressure can only support acoustic waves, and it is easy to see that both the (two frequency) wave train and bump for the pressure are right acoustic waves, moving to the right with speed

$$\lambda_{3,R} = v_R + c_R, \tag{4.16}$$

which is 1.73. At any time t_1 the position of the acoustic bump and the front of the wave train follow from

$$x_{3,1} = x_{s,0} + \lambda_{3,R}(t_1 - t_0). \tag{4.17}$$

For our data we obtain $x_{3,1} = 0.97$, which agrees with Figure 4.6.
On the other hand, we can see both the acoustic and entropy noise in the density plot. Very small oscillations are seen in the first density plot (Figure 4.5). However, when we zoom in, as in Figure 4.6, all four types of spurious waves found in the the entropy and pressure can be visualized [19]. We can see: (i) the acoustic leading bump at the position $x_{3,1} = 0.97$, (ii) the entropy leading bump (on the density distribution it is flipped and is, actually, a dent) at $x_{2,1} = 0.50$, (iii) the two-frequency acoustic wave train (over the interval $(x_{2,1}, x_{3,1})$), and (iv) a wave resulting from the interaction between the acoustic and the entropy wave trains (over the interval $(x_{s,1}, x_{2,1})$).

[19] No surprise, since the density can support both acoustic and entropy waves.

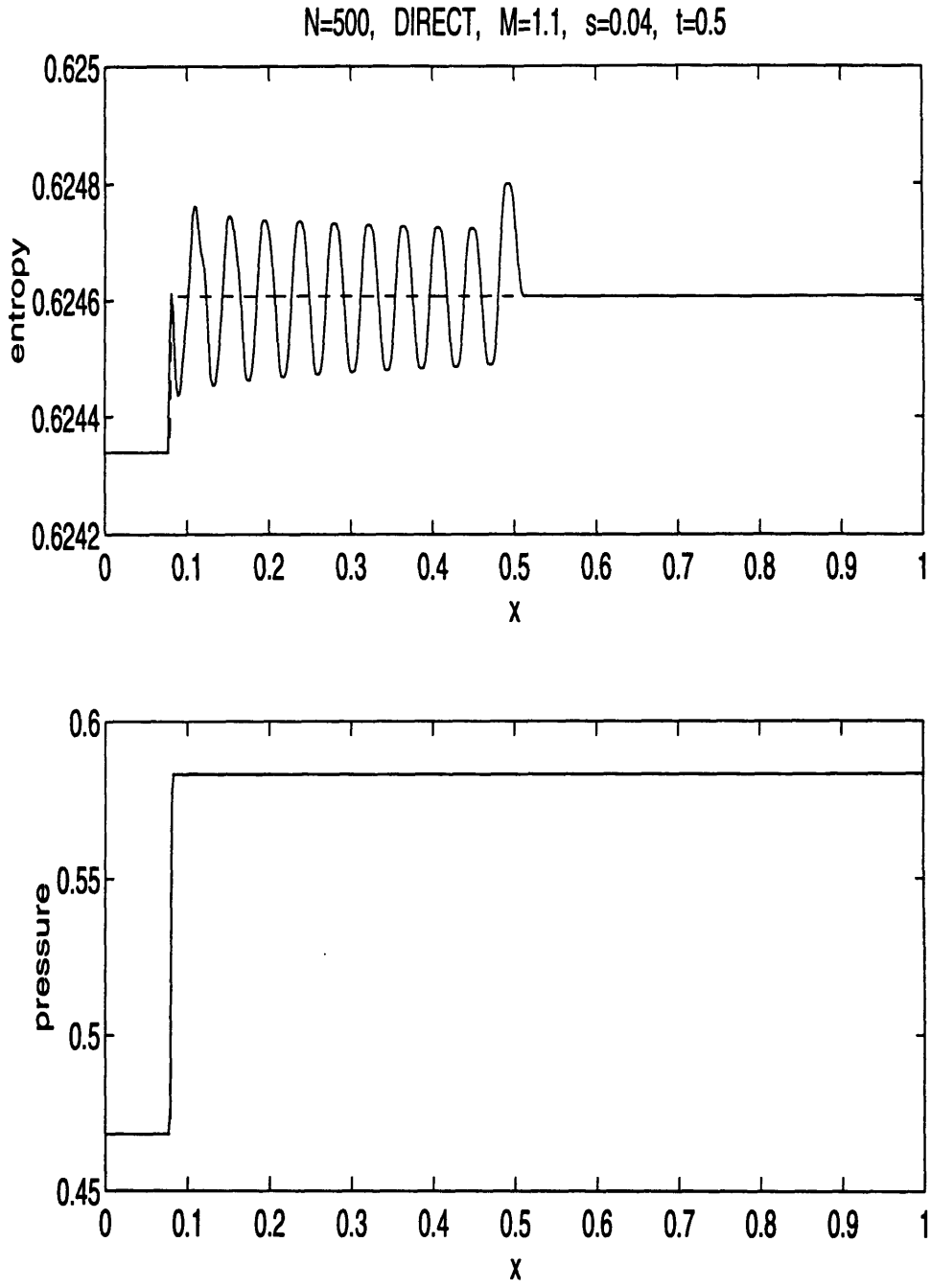

Figure 4.4: Moving Shock. Example 4.1. Entropy and Pressure as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

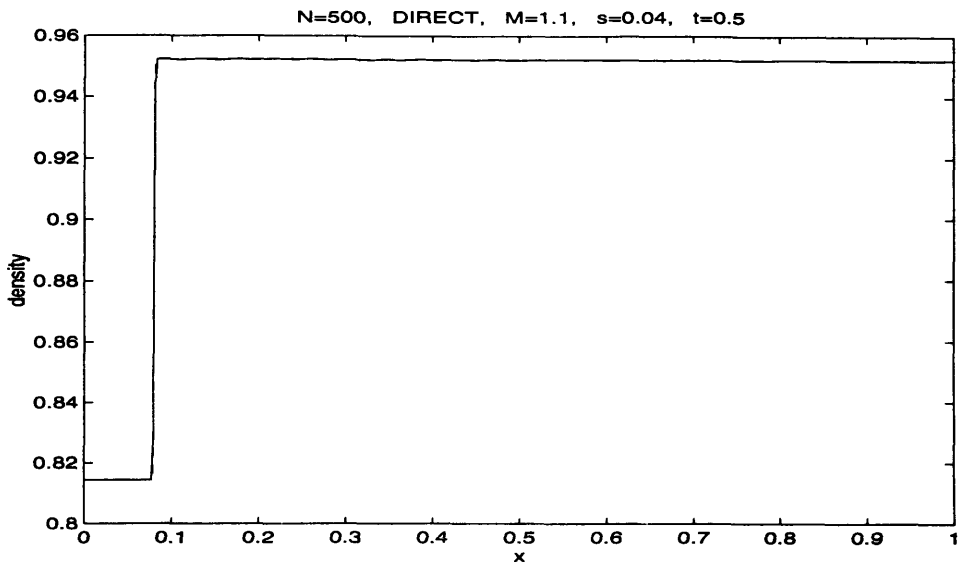

Figure 4.5: Moving Shock. Example 4.1. Density as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

4.2.3 Moving Shock. Effect of Changing Parameters

We ran the experiments using different Godunov-type schemes, various shock parameters, and grids with different cell sizes. All four kinds of waves always can be seen, although their amplitudes and frequencies vary a lot.

Sensitivity to Scheme Type

When we fixed the mesh spacing and the shock parameters, and ran simulations employing various Godunov-type schemes, we were unable to observe any major differences in the parasitic waves sizes and main frequencies. The wave shapes would change, side band harmonics would arise, but all these effects turn out to be of second order. Because variations in the type of scheme seems to be of little importance for this issue, we will use a second order Godunov scheme with an exact Riemann solver and van Leer-type limiters applied to pseudo-invariants throughout this whole chapter (in the calculations that illustrate the problems with numerical noise generation that arise).

Sensitivity to Cell Size

In the next set of the experiments, we fixed the shock parameters and type of the scheme, and investigated the effect of grid resolution on the generation of the spurious waves. For the sake of brevity, only one (typical) set of experiments is shown here.

Figures 4.7 and 4.8 illustrate the effect of the cell size for the set of the experiments corresponding to the set-up described in subsection 4.2.1: the shock is initially at $x_{s,0} = 0.1$ and moves with a

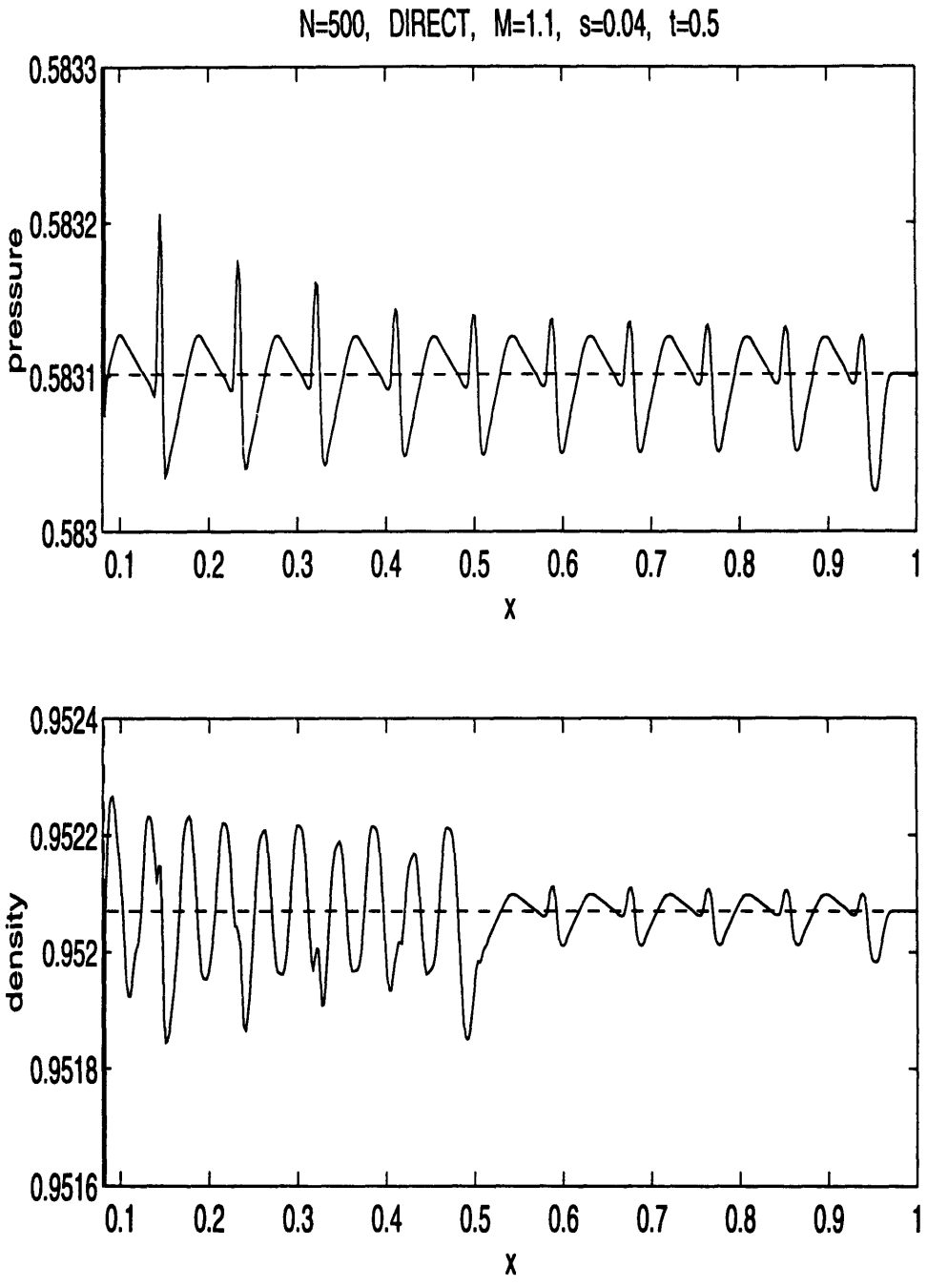

Figure 4.6: Moving Shock. Example 4.1. Pressure and Density as Function of Distance along Tube, x. Enlarged. Solid line represents simulation results. Dashed line represents exact solution.

velocity of $s = 0.04$ to the left; the Mach number is 1.1. The number of the cells in the computations are 250, 500, 1000, and 2000. The entropy distributions are shown at time $t_1 = 0.5$

From Figures 4.7 and 4.8 one concludes that refining the mesh reduces the wavelength of the spurious wave trains: the wavelength appears to be a *linear* function of the cell size $\Delta\xi$. On the other hand, the number of grid points per wavelength does not change with the cell size. It causes each oscillation inside the train to become thinner as the grid is refined, while the number of waves inside the wave train increases (the total length of the wave train does not change with mesh refining). The number of the grid points in the leading bump does not change either with the cell size. Thus is becomes ever more narrow with grid refinement.

Even more **important** is the *disturbing* fact that refining the grid **does not reduce the magnitude of spurious disturbances** right at the shock. The moving shock produces parasitic oscillations of the same amplitude – for all four cell sizes the magnitudes of the entropy waves are the same at the shock ($x = x_{s,1}$). However, the magnitude of the oscillations starts to decrease as they move downstream away from the shock. There is some vague reminiscence with the Gibbs phenomena in these behaviors, but this is just an analogy, no true relationship exists (see Remark 4.3 below).

The drop in amplitude as the noise moves away from the shock should be attributed to the numerical scheme viscosity (see Remark 4.3 below). The numerical dissipation of a Godunov-type scheme depends on the type of stencil and the time step – cell size ratio — as well as on the wavelength of the disturbance. It can be shown [30] that for most Godunov-type schemes the dissipation increases as the wave number decreases, very drastically for near cell size waves: It becomes enormous for wave numbers comparable to the grid size. These schemes do not affect or only slightly damp oscillations with long wavelength, while they ruthlessly suppress short wavelength disturbances. What we observe in our experiments can be explained as follows: The wavelength of the noise generated is tied up to the cell size. For a large cell size, relatively long wavelength oscillations are produced which are almost unaffected by the numerical viscosity. On the other hand, for a small cell size, short wavelength disturbances are generated which are heavily damped by the scheme dissipation.

Sensitivity to Shock Parameters

In this next set of the experiments, we zeroed in on the sensibility of the spurious waves to the shock parameters. We fixed the scheme and grid resolution (we used 500–cell grid), and we performed computations for various shock speeds, s, and Mach numbers, M. Usually a moving shock displays generation of all four types of spurious noise. However, the relative and absolute magnitudes of the generated waves can vary. Some characteristic results are shown below.

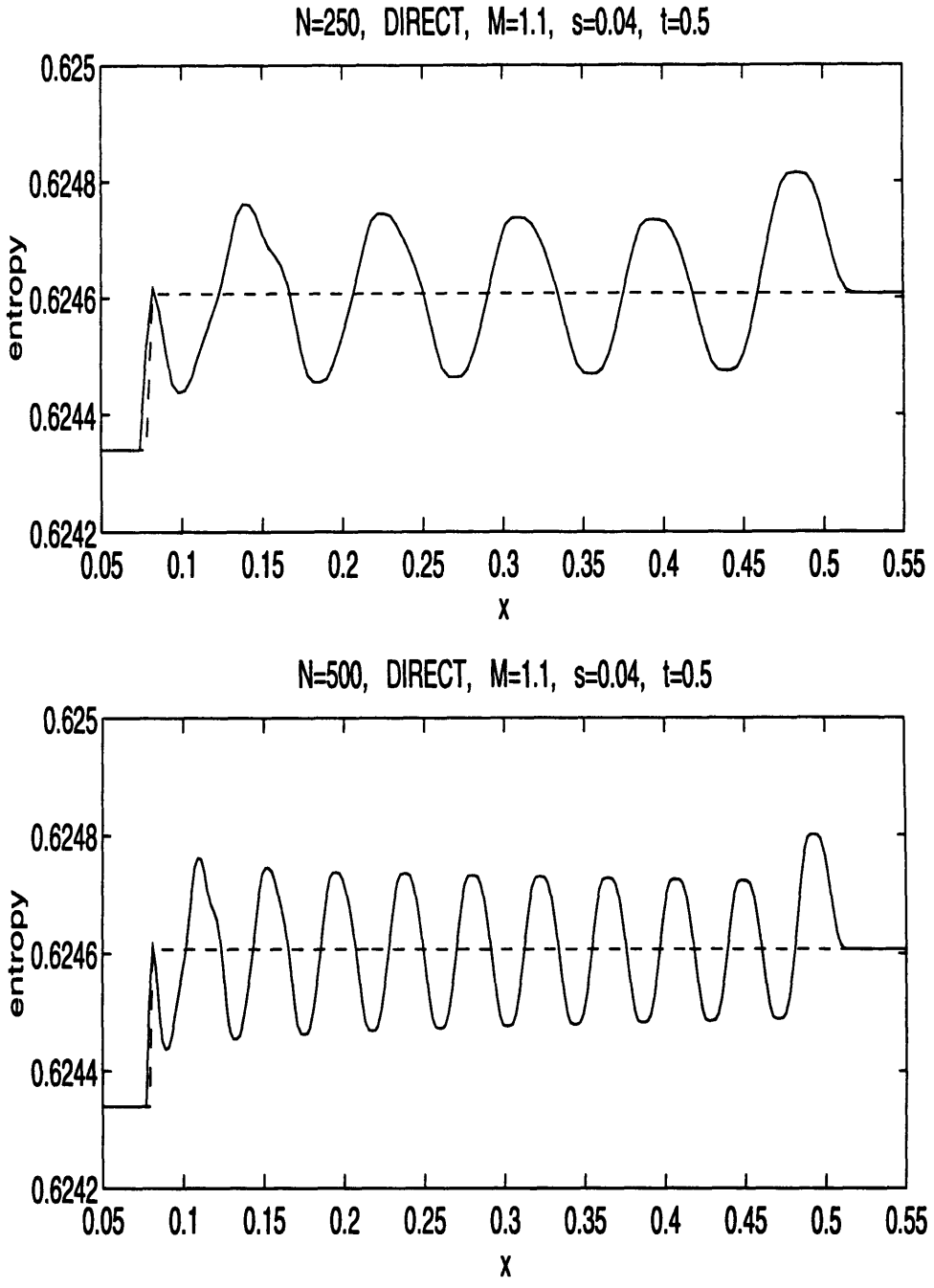

Figure 4.7: Moving Shock. Sensitivity to Cell Size (a). Entropy as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

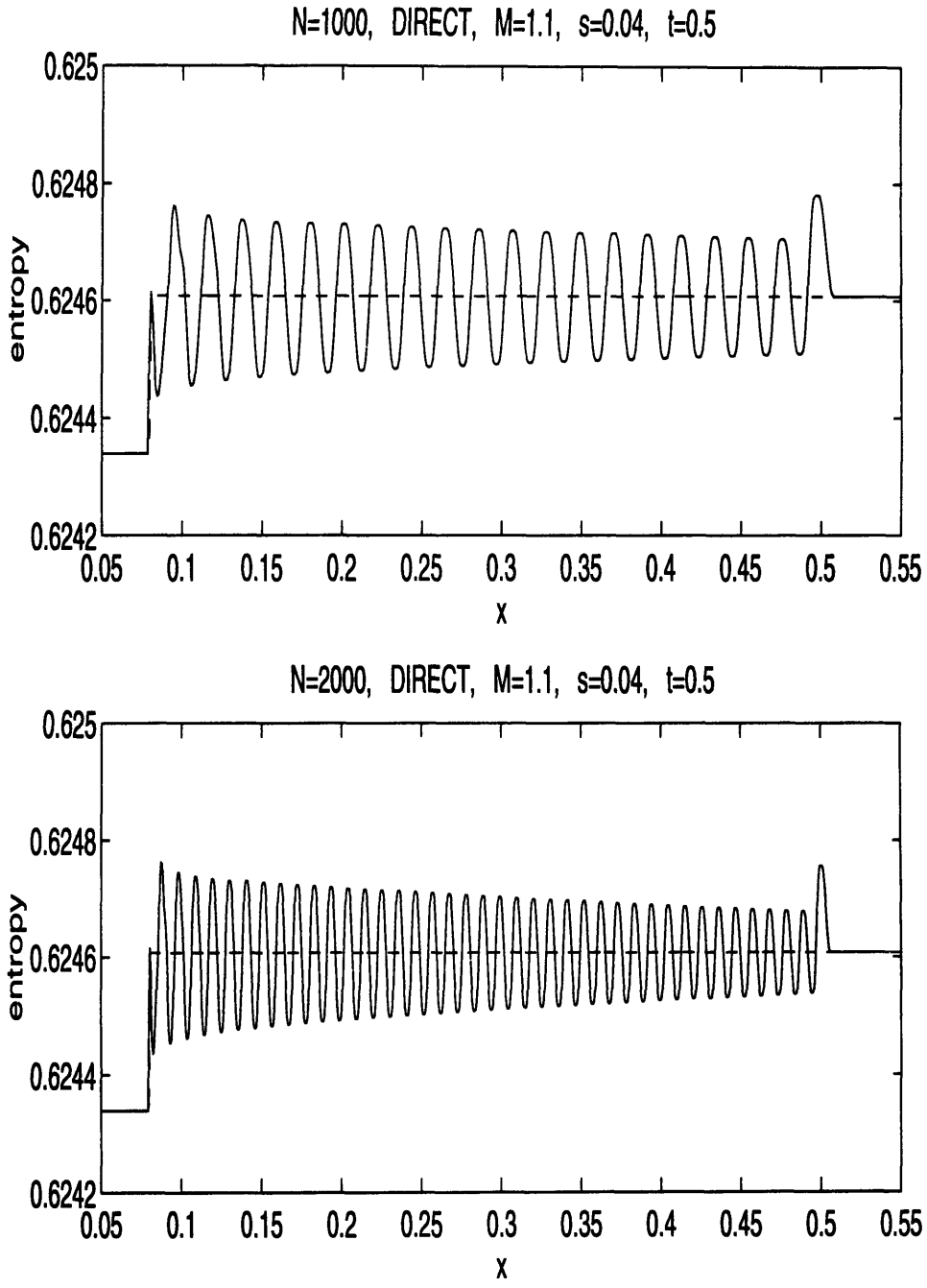

Figure 4.8: Moving Shock. Sensitivity to Cell Size (b). Entropy as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

Example 4.1

$M = 1.1$, $s = 0.04$.

$x_{s,0} = 0.1$, $t_1 = 0.5$.

Left state: $\rho_L = 0.81$, $v_L = 0.95$, $p_L = 0.47$.

Right state: $\rho_R = 0.95$, $v_R = 0.80$, $p_R = 0.58$.

$\lambda_{2,R} = 0.80$, $\lambda_{3,R} = 1.73$.

$x_{2,1} = 0.50$, $x_{3,1} = 0.97$.

Figures 4.4.

Both entropy waves — the bump and the wave train — are huge, more than 50% of the entropy jump across the shock, while both acoustic waves — the bump and the wave train — are indistinguishable small.

Example 4.2

$M = 1.3$, $s = 0.02$.

$x_{s,0} = 0.1$, $t_1 = 0.5$.

Left state: $\rho_L = 0.68$, $v_L = 1.10$, $p_L = 0.36$.

Right state: $\rho_R = 1.02$, $v_R = 0.72$, $p_R = 0.65$.

$\lambda_{2,R} = 0.72$, $\lambda_{3,R} = 1.66$.

$x_{2,1} = 0.46$, $x_{3,1} = 0.93$.

Figures 4.9.

Similar to the previous case. Both entropy waves — the bump and the wave train — are large, around 20% of the entropy jump across the shock, while both acoustic waves — the bump and the wave train — are vanishingly small. The wave train wavelength is twice as long as that of the previous set-up. Therefore, the wave train is almost not affected by the scheme dissipation, and the leading bump has the same magnitude as the oscillations in the wave train. One can see from the juxtaposition of Figures 4.9 and 4.4 that the amplitude of the entropy oscillations in Example 4.1 drops noticeably as they move away from the shock, while the amplitude of the entropy oscillations in the present Example remains almost unchanged.

Example 4.3

$M = 3$, $s = 0.02$.

$x_{s,0} = 0.1$, $t_1 = 0.5$.

Left state: $\rho_L = 0.11$, $v_L = 1.77$, $p_L = 0.03$.

Right state: $\rho_R = 0.41$, $v_R = 0.44$, $p_R = 0.28$.

$\lambda_{2,R} = 0.44$, $\lambda_{3,R} = 1.42$.

$x_{2,1} = 0.32$, $x_{3,1} = 0.81$.

Figures 4.10.

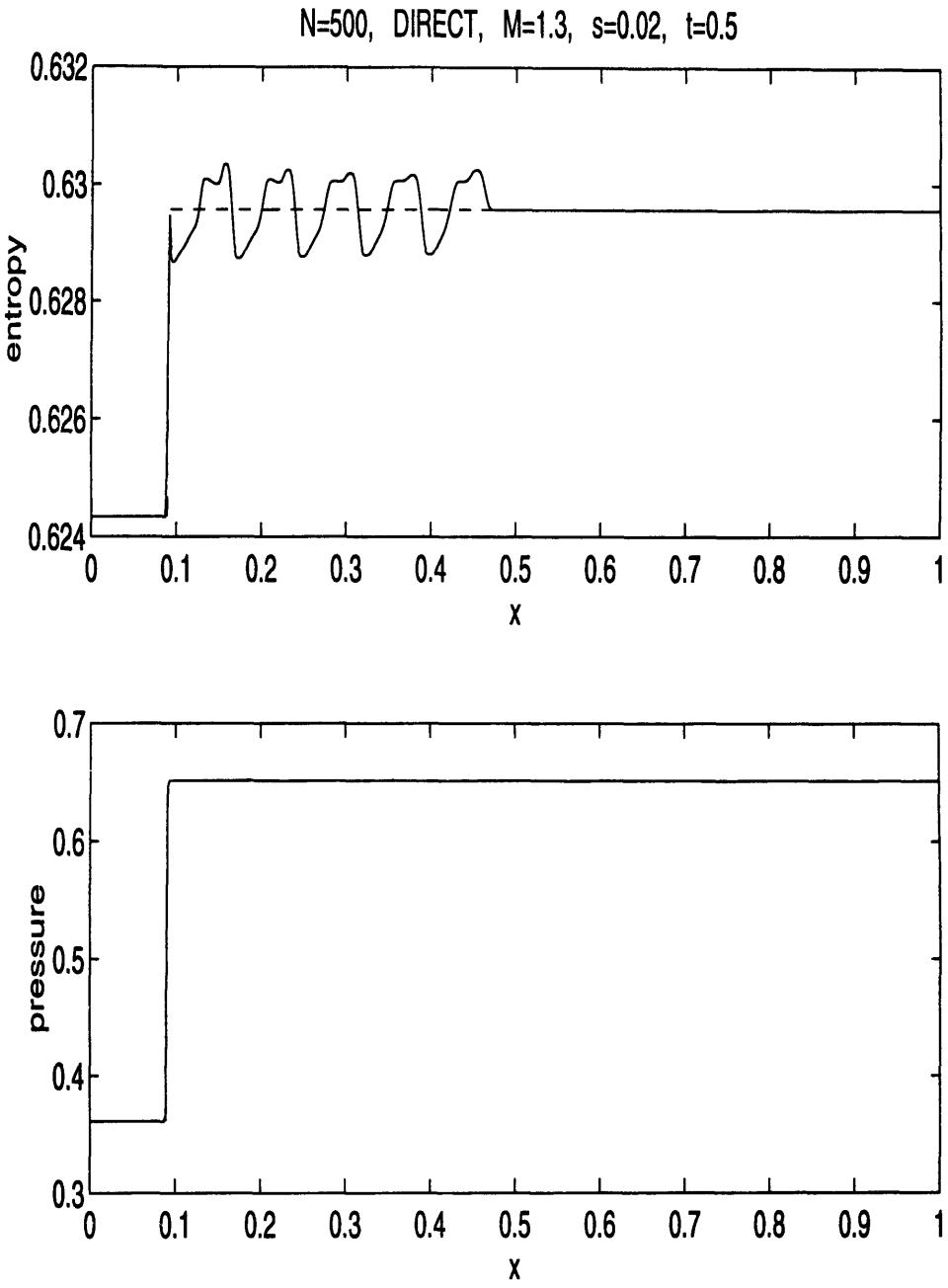

Figure 4.9: Moving Shock. Example 4.2. Entropy as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

Both entropy and acoustic wave trains are noticeable. The amplitude of the entropy wave train is more than 3% of the entropy jump across the shock, and the amplitude of the acoustic wave train is around 2% of the pressure jump across the shock. On the contrary, both bumps are very hard to discern.

Example 4.4

$M = 3$, $s = 0.1$.

$x_{s,0} = 0.1$, $t_1 = 0.5$.

Left state: $\rho_L = 0.11$, $v_L = 1.69$, $p_L = 0.03$.

Right state: $\rho_R = 0.41$, $v_R = 0.36$, $p_R = 0.28$.

$\lambda_{2,R} = 0.36$, $\lambda_{3,R} = 1.34$.

$x_{2,1} = 0.28$, $x_{3,1} = 0.77$.

Figures 4.11.

The wavelength of the entropy wave train is too short — so it is almost immediately killed by the numerical viscosity — but the leading bump survives. It can be found at $x_{2,1}$, its magnitude being around 3% of the entropy jump across the shock. The acoustic waves are still prominent, their amplitudes being up to 2% of the pressure jump across the shock. To distinguish the leading acoustic bump from the following wave train one should enhance the resolution – see Figure 4.12. Figure 4.12 also zooms up the entropy distribution, so the dying of the wave train at the hands of the numerical viscosity can be clearly seen.

Example 4.5

$M = 3$, $s = 1$.

$x_{s,0} = 0.6$, $t_1 = 0.5$.

Left state: $\rho_L = 0.11$, $v_L = 0.79$, $p_L = 0.03$.

Right state: $\rho_R = 0.41$, $v_R = -0.54$, $p_R = 0.28$.

$\lambda_{2,R} = -0.54$, $\lambda_{3,R} = 0.44$.

$x_{2,1} = 0.33$, $x_{3,1} = 0.82$.

Figures 4.13.

Both leading bumps are substantial. The entropy bump, located at $x_{2,1}$, has a magnitude up to 4% of the entropy jump across the shock. The acoustic bump, located at $x_{3,1}$, reaches 2% of the pressure jump across the shock. The entropy wave train is still noticeable (around 1% of the entropy jump), though most of its initial amplitude (see a spike at the shock front), due to the short wavelength of the oscillations, is being quickly killed by the numerical dissipation. The acoustic wave train is vanishingly small because of the same reason: its high frequency oscillations are

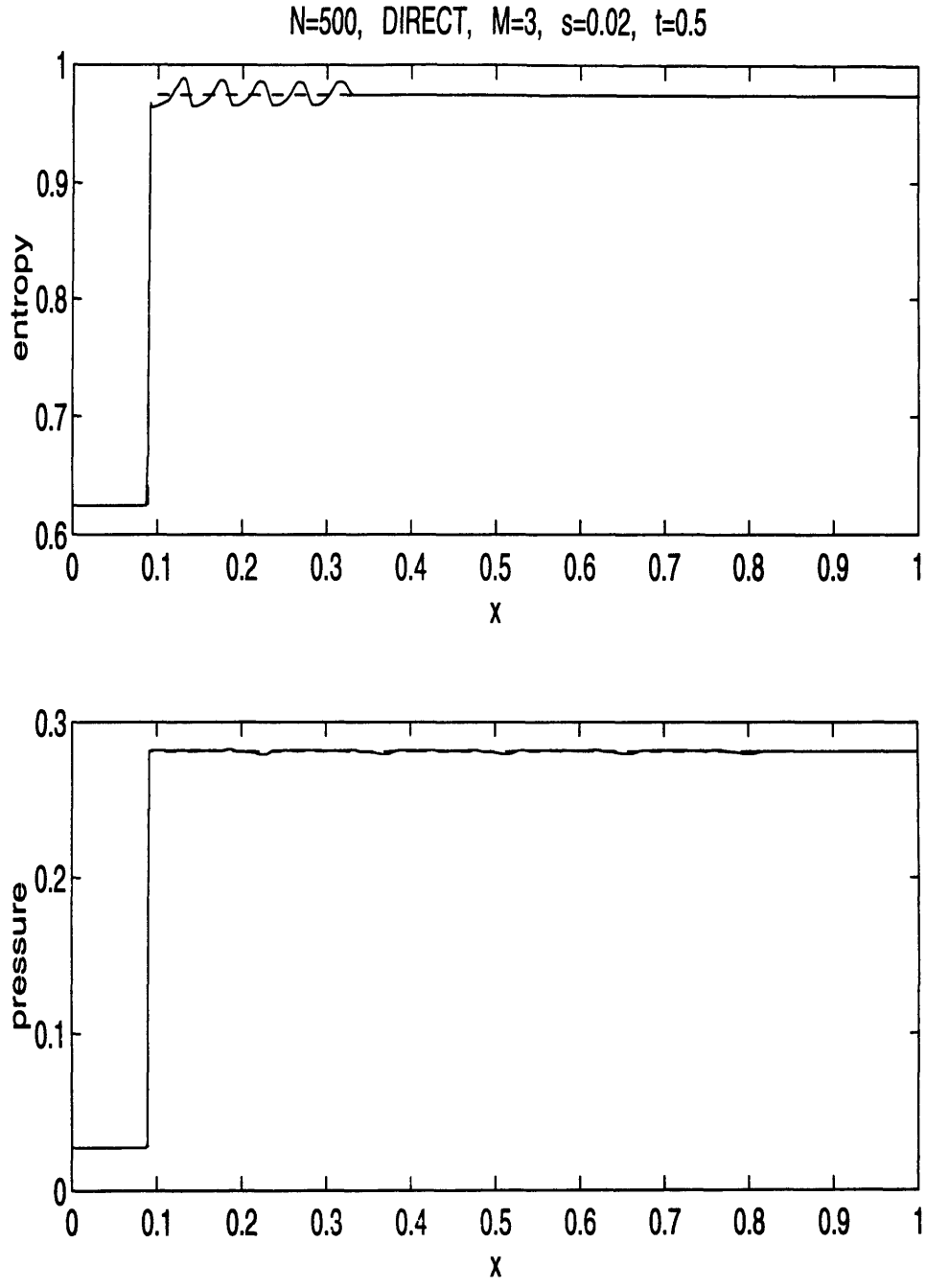

Figure 4.10: Moving Shock. Example 4.3. Entropy as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

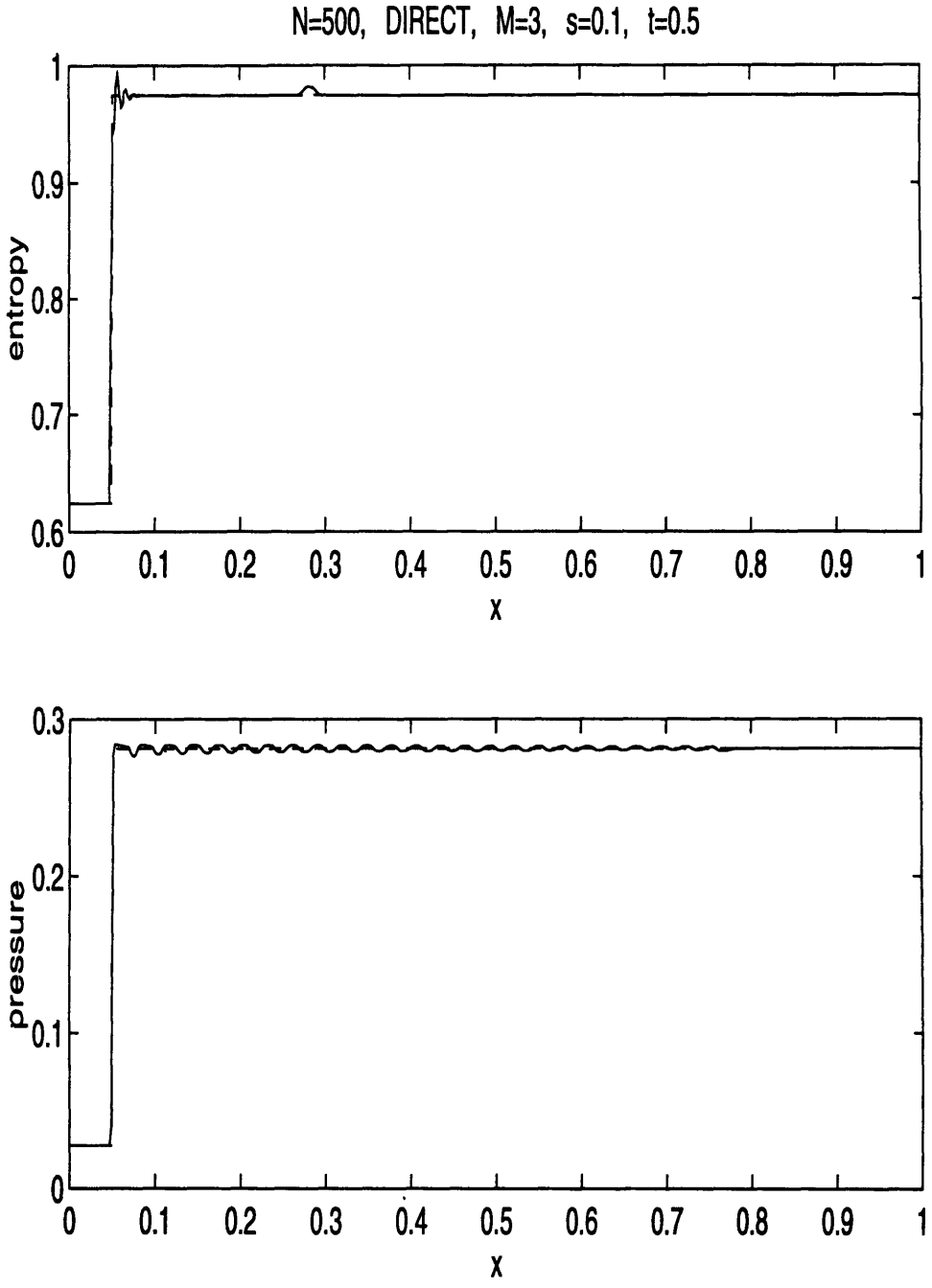

Figure 4.11: Moving Shock. Example 4.4. Entropy as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

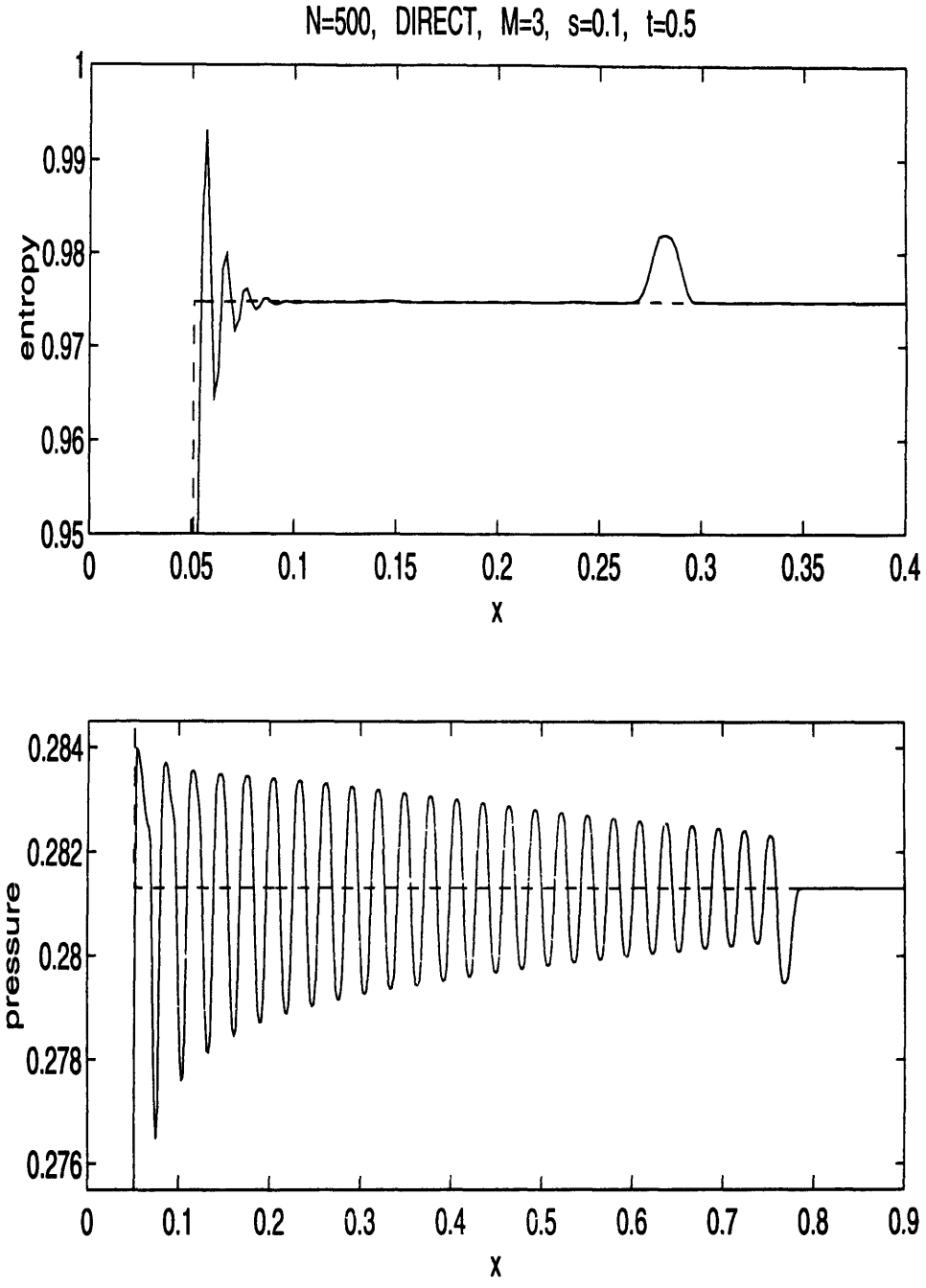

Figure 4.12: Moving Shock. Example 4.4. Entropy as Function of Distance along Tube, x. Enlarged. Solid line represents simulation results. Dashed line represents exact solution.

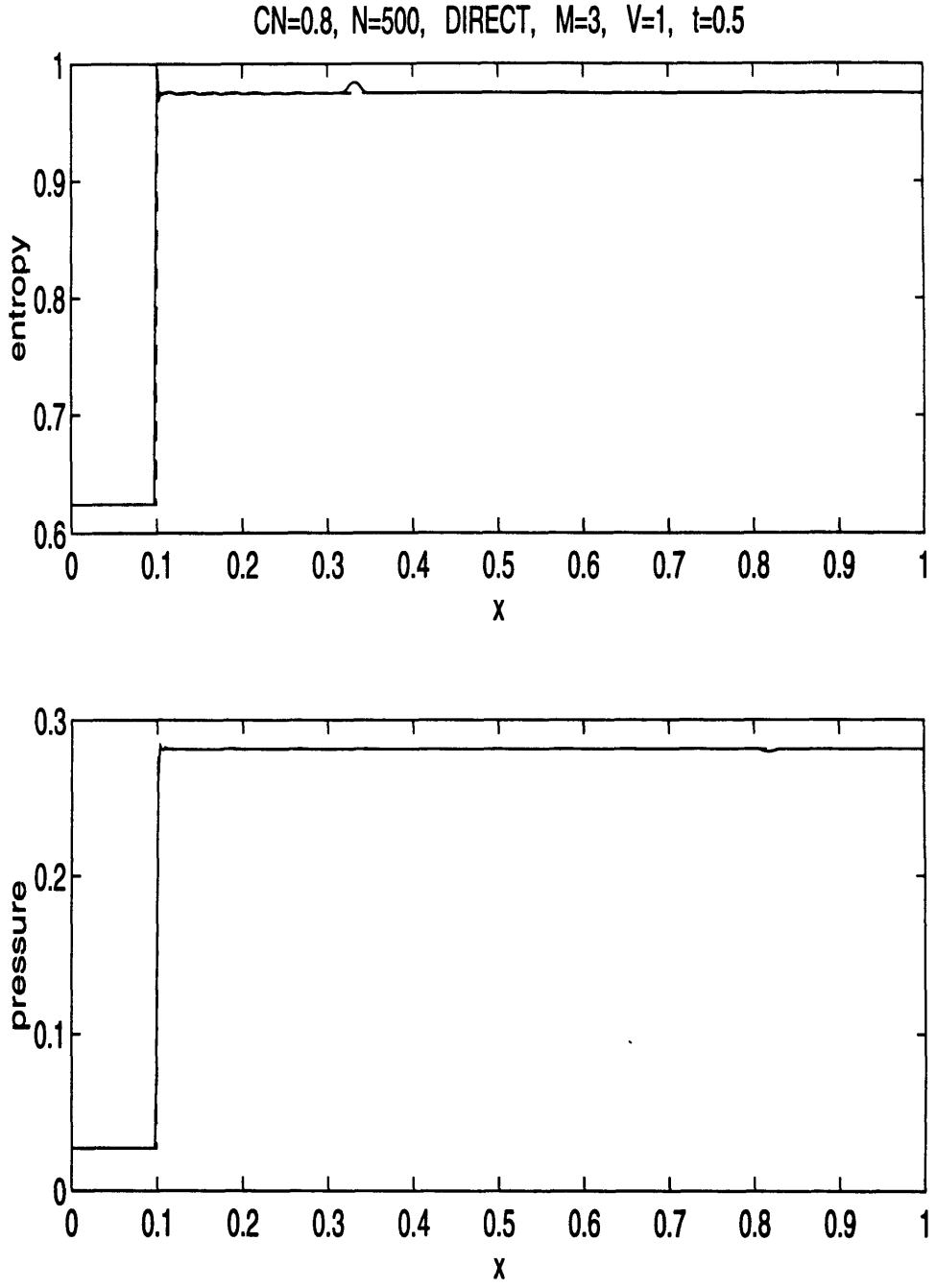

Figure 4.13: Moving Shock. Example 4.5. Entropy as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

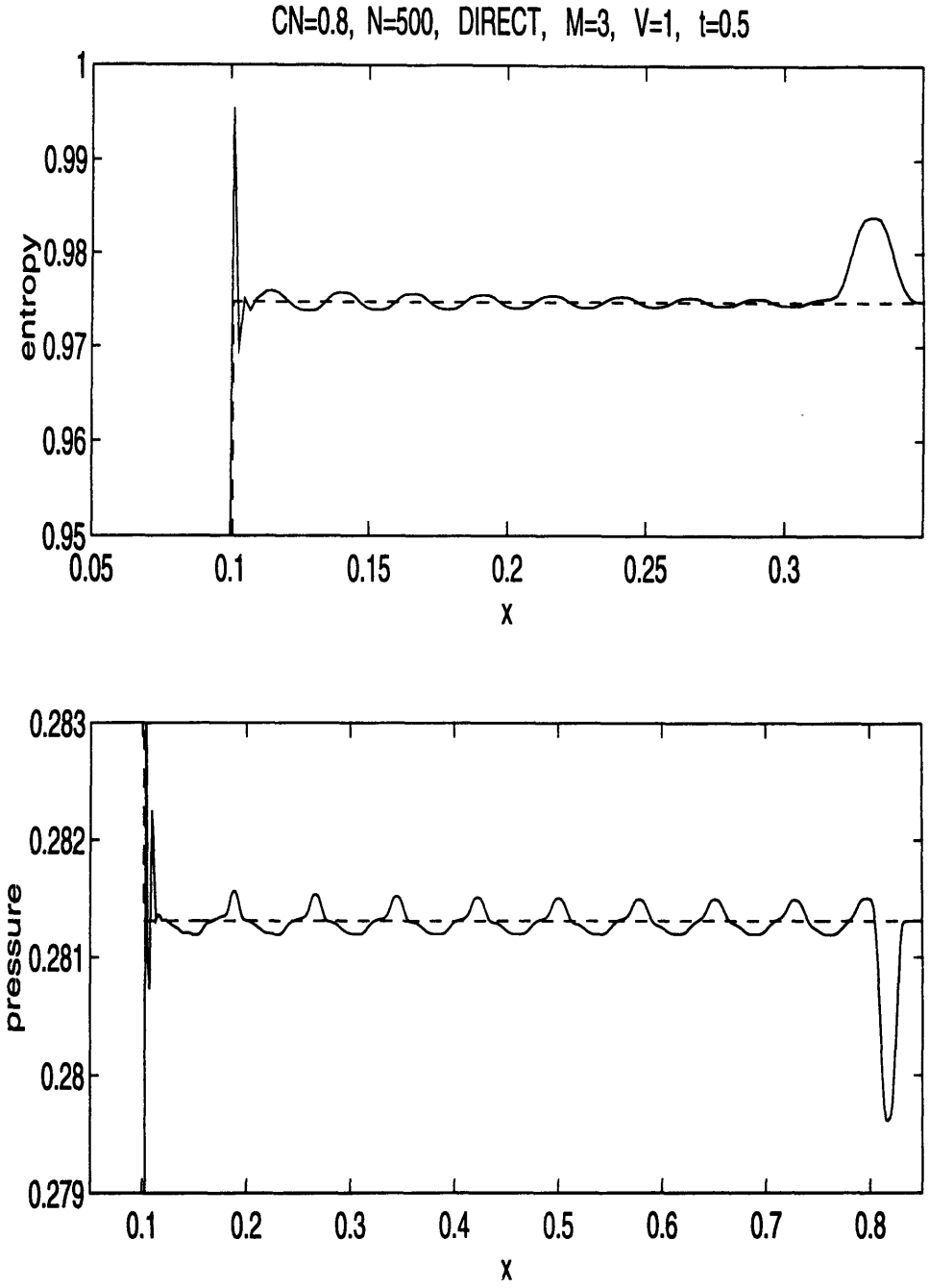

Figure 4.14: Moving Shock. Example 4.5. Entropy as Function of Distance along Tube, x. Enlarged. Solid line represents simulation results. Dashed line represents exact solution.

heavily damped by the viscosity of the scheme. With increased resolution, Figure 4.14, we can see that while the "main frequency" oscillations are completely suppressed by the numerical dissipation, the side band frequency waves — with much longer wavelengths — are only slightly damped.

Example 4.6

$M = 1.3$, $s = 2$.

$x_{s,0} = 0.5$, $t_1 = 0.1$.

Left state: $\rho_L = 0.08$, $v_L = 1.27$, $p_L = 0.36$.

Right state: $\rho_R = 0.12$, $v_R = 0.16$, $p_R = 0.65$.

$\lambda_{2,R} = 0.16$, $\lambda_{3,R} = 2.91$.

$x_{2,1} = 0.52$, $x_{3,1} = 0.79$

Figure 4.15.

The entropy bump, located at $x_{2,1}$, is enormous, around 50% of the entropy jump across the shock. The main frequency entropy wave train is completely damped by the numerical viscosity (only an initial spike at the shock location indicates its presence). However, the low frequency side band oscillations are almost unaffected by the scheme dissipation and can be clearly seen. They are quite big, around 2% of the entropy jump across the shock. The acoustic bump is almost indiscernible – a tiny dent at $x_{3,1}$ on the pressure profile. The acoustic wave train is entirely suppressed by the numerical viscosity.

4.2.4 Shock Reflecting From a Wall

When, in the initial boundary value problem described in subsection 4.2.1, we change the absorbing boundary conditions at the left boundary into the reflecting boundary conditions (2.6), we obtain an initial boundary value problem modeling shock reflection from a rigid wall [20].

Instead, we will consider an equivalent problem — collision of two shocks of equal strength. We prefer two shock collision problem because it can be generalized for the case when two colliding shocks have different strengths.

For the shock collision problem, we use the following initial conditions:

$$u(x,0) = \begin{cases} u_1 & \text{if} \quad x < x_{L,0}, \\ u_0 & \text{if} \quad x_{L,0} < x < x_{R,0}, \\ u_1 & \text{if} \quad x_{R,0} < x, \end{cases} \tag{4.18}$$

[20] Now, besides the requirement that u_L, u_R, and s satisfy the Rankine-Hugoniot relation (2.11), we should prescribe "left" velocity zero value : $v_L = 0$

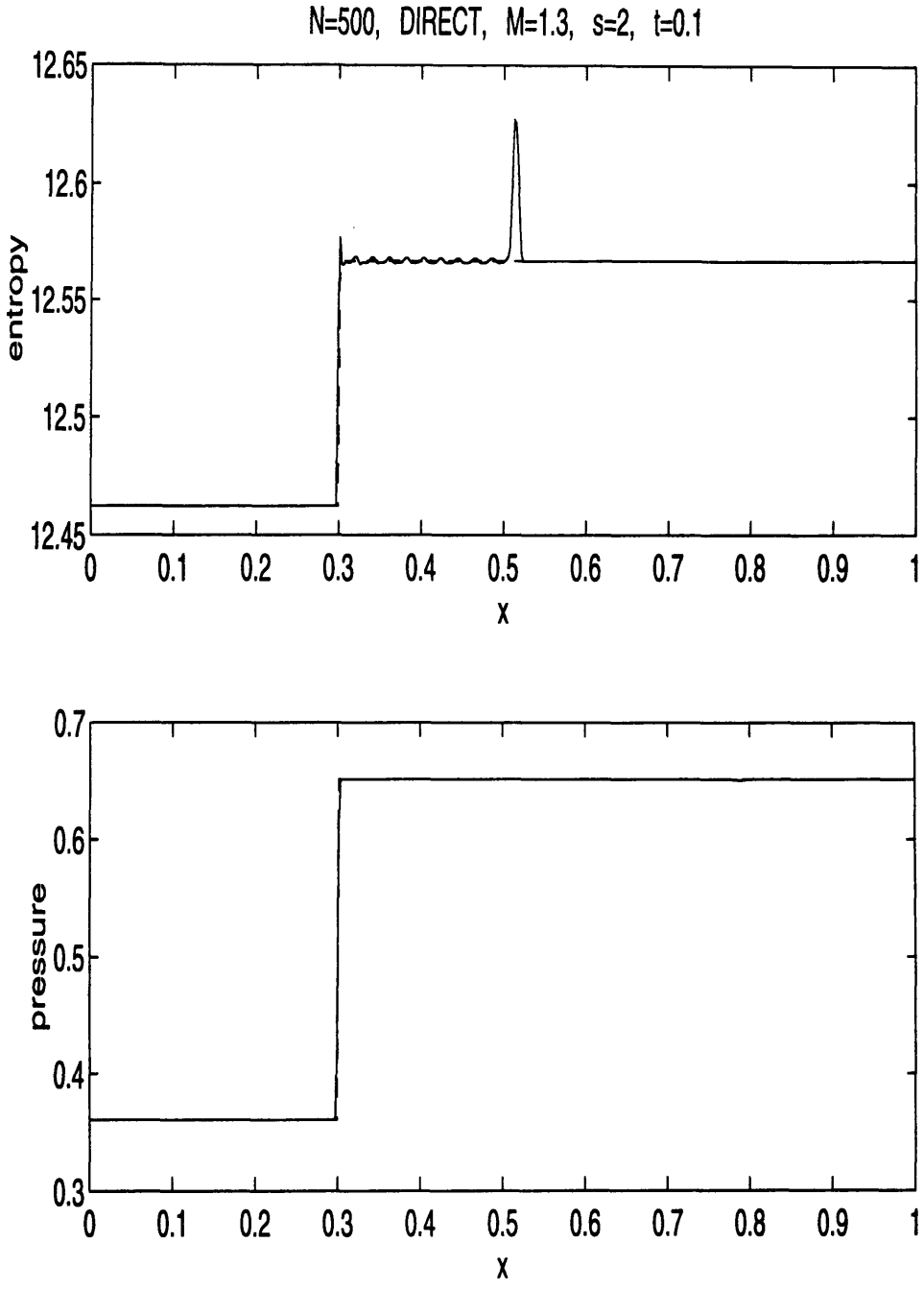

Figure 4.15: Moving Shock. Example 4.6. Entropy as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

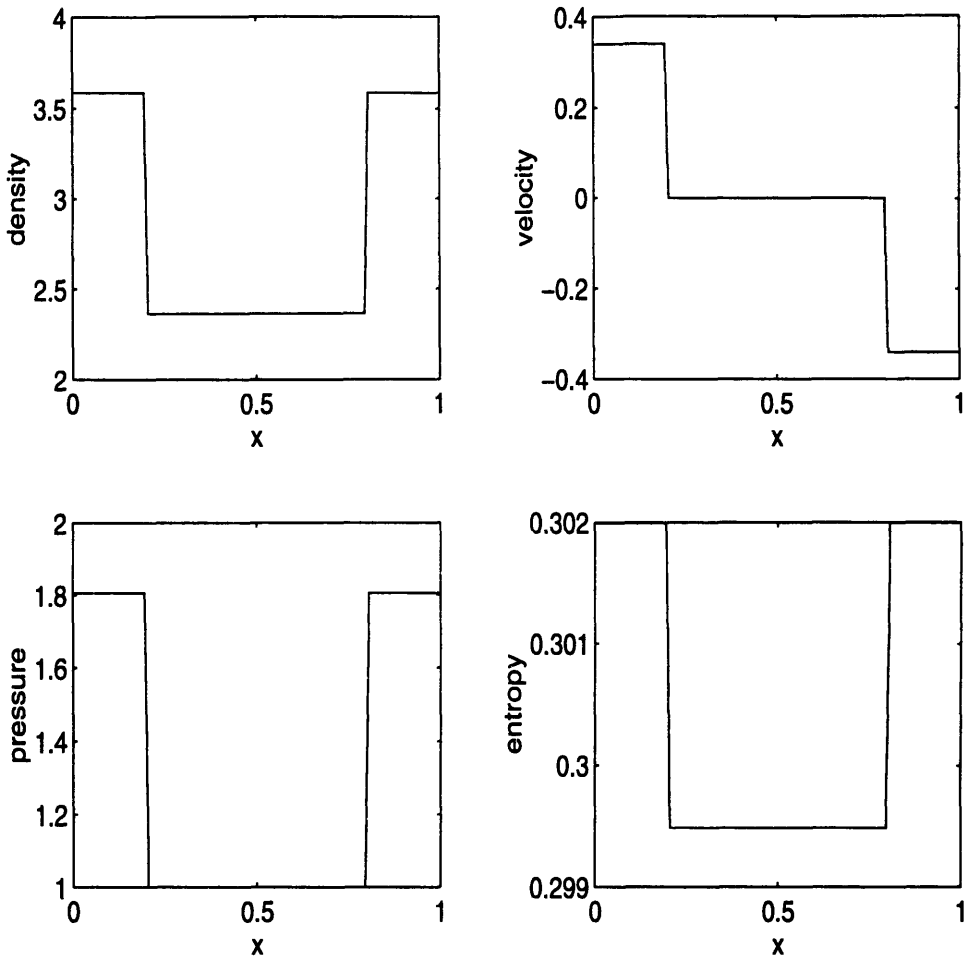

Figure 4.16: Initial conditions for a typical head-on shock collision problem. Density, velocity, pressure, and entropy distributions corresponding to initial conditions (4.18).

where $x_{L,0}$ and $x_{R,0}$ are the shock positions at time $t_0 = 0$. Initially, the shocks are placed symmetrically with respect to the center of the computational domain, some distance apart. The states u_1 and u_0 (with $v_0 = 0$) satisfy the Rankine-Hugoniot condition (2.11) for some shock speed s, such that the shock initially located at $x_{L,0}$ is a right moving shock. Then the shock initially located at $x_{R,0}$ will be a left moving shock. Of course: $p_0 < p_1$. Figure 4.16 illustrates a typical initial condition.

At both boundaries, we impose the absorbing boundary conditions (see subsection 4.2.1).

The flow resulting from the initial conditions (4.18) consists of three constant states u_1, u_0, and u_1, separated by two shocks moving towards each other at constant speed s. After the collision, the solution consists again of three constant states separated by two (equal) shocks, now moving away from each other. We run our simulation for as long as it takes for the shocks to collide (in the middle of the computational domain) and return to the initial positions after "reflecting" from each other.

As in the case of a single moving shock, subsection 4.2.1, we performed computations over a

wide range of shock speeds and Mach numbers, using different Godunov-type schemes and grid resolutions. In the interest of brevity, only one typical calculation is shown here.

Figure 4.17 illustrates the spurious oscillations observed with a second order Godunov-type scheme with an exact Riemann solver and van Leer type limiters applied to pseudo-invariants. In this run a 100-cell grid was used. Two shocks each with Mach number 1.3 (which corresponds to the pressure ratio across the shock 1.805) and velocity 1 were set 0.8 length units apart (see Figure 4.16), raced towards each other, reflected, and returned to the initial position.

First of all, one can observe the same kind of oscillations we encountered in the case of a single moving shock (see subsection 4.2.1): small oscillations in the pressure, larger ones in the density, and the largest in the entropy (about 10% of the entropy jump across the shock). However, besides those oscillations one can see a new type of spurious wave: three big "horns" appear in the entropy plot. The size of these "horns" is about 50% of the entropy jump at the shock. The "horns" appear at the time and place where the shocks collide, and then stay there "for ever" (the entropy moves with the flow speed, which vanishes there). The number of "horns" varied in different experiments (we saw from 2 to 5), depending on the Godunov-type scheme used, but they always emerge at the interaction of two shocks of the same strength. Their size can be close to 100% of the entropy jump at the shock. The same kind of disturbance can be seen in the density profile as well, but in much more "tame" form — three small dents on the density distribution in Figure 4.17.

These "horns" are closely related to the leading/initial entropy bumps which are generated at the moment the shock begins to move with respect to the grid in the experiments of subsection 4.2.2). Both disturbances are entropy waves. Both are "transient" waves is the sense that they occur because of sudden changes — in one case a shock abruptly comes into motion, in the other case a shock drastically stops at the collision point, changes its strength, and starts moving in the opposite direction. Both waves come to life as the result of some drastic changes and adjustments. However, the waves are not transient in the sense of passing quickly into and out of existence. Once generated, they can survive "for ever".

There are some differences between the "horns" and the leading bumps, though. The horns are normally larger. A shock collision generates several "horns", while a shock beginning to move produces only one bump. Because the flow velocity behind a moving shock is usually different from zero, the bumps are "washed" away (entropy waves travel with the flow speed). On the other hand, the gas is always at rest behind the shock running away after a collision/reflection, hence the "horns" stay at the place where the interaction took place.

As in the case of the entropy leading bumps, the main features of the "horns" turn out to be almost insensitive to the type of scheme used and mesh refinement. There are two exceptions, though. Firstly, grid refinement causes the horns to narrow (the number of the grid points per "horn" remains the same as the cell size goes down). Secondly, the number and the shape of the

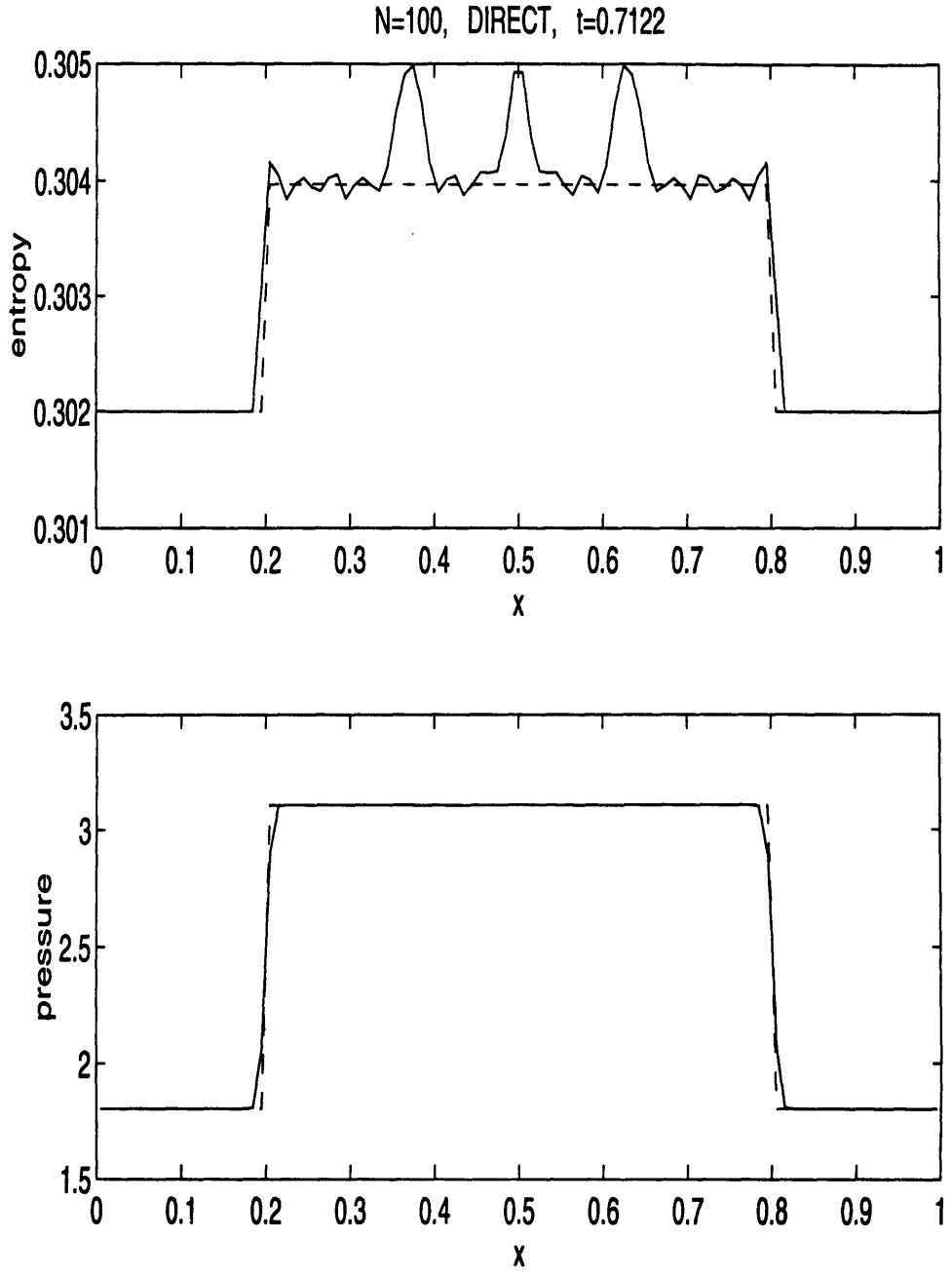

Figure 4.17: Typical head-on shock collision problem. Initial conditions are given in (4.18). Entropy and Pressure as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

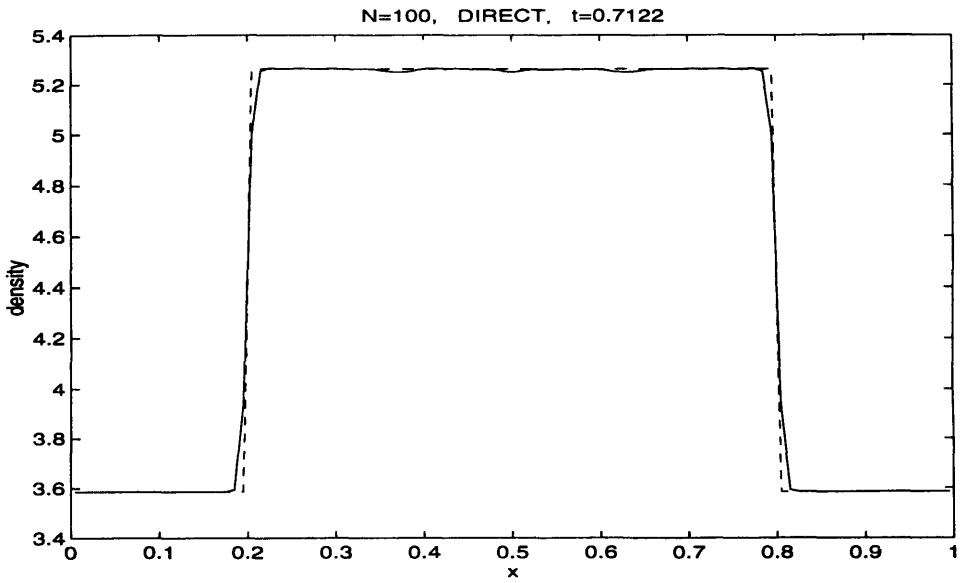

Figure 4.18: Typical head-on shock collision problem. Initial conditions are given in (4.18). Density as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

"horns" varies with the scheme type.

In a shock collision where the shocks have different strengths, but the difference is not too big [21], similar (spurious numerical noise) "horns" still appear.

4.2.5 Main Properties of Spurious Waves

We want to emphasize the following characteristics of the parasitic waves produced by shock movement and interactions.

- We observed four types of parasitic noise waves. First, *adjustment noise*: Bumps/horns (both in the acoustic and entropy field), generated when the state of motion of the shock suddenly changes: either at the beginning of the simulation — at the moment the shock starts to move with respect to the grid, or when the shock collides with a rigid wall or another shock. These disturbances seem to manifest some readjustment processes taking place in the numerical shock. After the readjustments is completed, no more waves of this type are generated. Second, *wave trains*: both acoustic and entropic, generated "continuously", as long as the shock moves with respect to the grid.

- Those spurious waves, especially the entropy ones, can have really large magnitudes, comparable with the jumps across the shock.

[21] When the difference in strengths of the two colliding shocks is large, the structure of the post-collision flow changes: in addition to two shocks moving apart from each other, a strong contact discontinuity emerges.

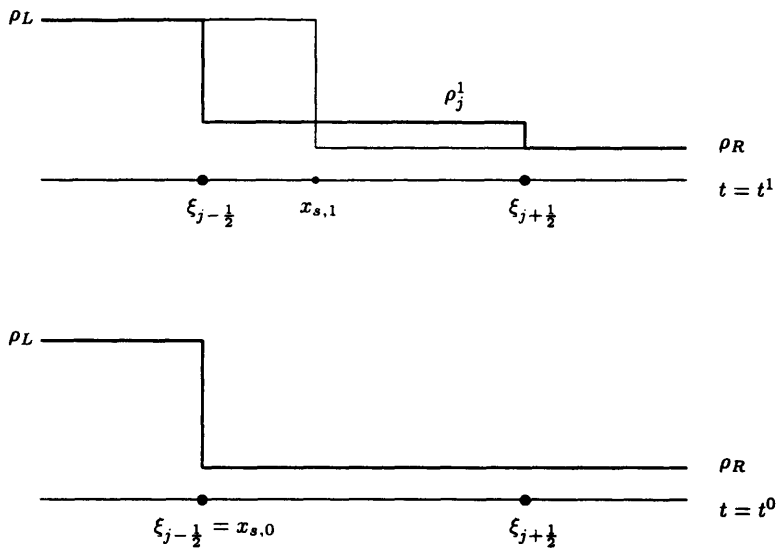

Figure 4.19: Smearing of the initially perfect profile due to the averaging. Thick line represents numerical solution. Thin line represents exact solution.

- The main features of those oscillations seem to be invariant to the type of the scheme being used.
- Increasing the resolution of the scheme (grid refinement) does not help much. The only way the mesh refinement helps is by increasing the frequency of the wave train — which leads to its damping by the numerical dissipation as the waves move away from their generation point. It does not affect the bumps/horns (except for their width).

4.2.6 Source of Problems

After all those observations, we were in a position to understand the mechanisms governing the generation of parasitic waves.

Finite Shock Width

Our conclusion is that the production of the various spurious parasitic noise is due to the finite size of the transition layer of a shock captured by a Godunov-type scheme.

Schemes in conservation form cannot, in general, resolve shocks as perfect — one-node — discontinuities. Conservation and perfect resolution of discontinuities are two conflicting requirements.

Consider, for example, a shock located at the initial moment at the j-th node, $x_{s,0} = \xi_{j-\frac{1}{2}}$ (see Figure 4.19). Let ρ_L and ρ_R denote the left and the right states, respectively. Let the shock be

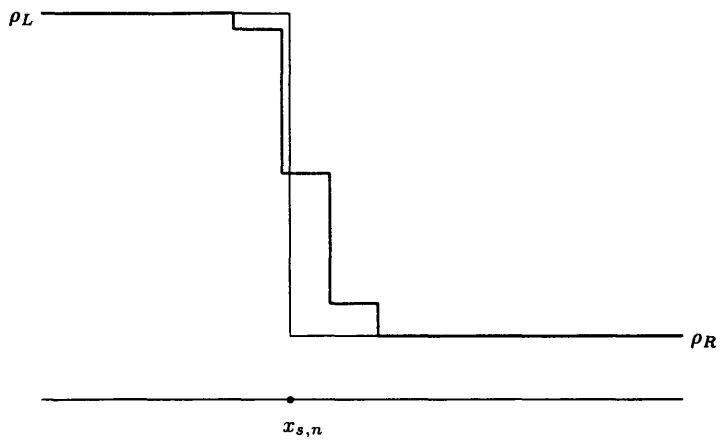

Figure 4.20: "Staircase"-like structure of numerical shock. Thick line represents numerical solution. Thin line represents exact solution.

moving to the right with speed s. After a time step Δt^0 the shock position will be $x_{s,1} = \xi_{j-\frac{1}{2}} + s\Delta t^0$, which, in general, will not coincide with the $(j+1)$-th node, and will be in the interior of the j-th cell I_j. Then conservation (4.4) requires that ρ_j^1 be an average of ρ_L and ρ_R:

$$\rho_j^1 = \frac{\rho_L(x_{s,1} - \xi_{j-\frac{1}{2}}) + \rho_R(\xi_{j+\frac{1}{2}} - x_{s,1})}{\xi_{j+\frac{1}{2}} - \xi_{j-\frac{1}{2}}} \tag{4.19}$$

The exact shock profile is shown with the dashed line in Figure 4.19, and the "computed" shock profile with the solid line. It is clear that the shock width has increased from one node to two — some smearing has been introduced.

After a few time steps, the shock will look like a series of small jumps, rather than one large discontinuity (Figure 4.20). The numerical shock then ends up with a width of a few grid points — when "zoomed up", it resembles a "staircase". The density, instead of jumping from ρ_R to ρ_L across a single point, changes its value less dramatically, through a sequence of smaller jumps across several grid points. Thus the discontinuity is replaced by a (thin) ***transition layer*** from ρ_R to ρ_L. Hence some inevitable smearing is inherent in computations in conservation form with a ***fixed grid.*** This smearing is just another manifestation of the ubiquitous numerical viscosity.

Remark 4.2 If we draw a shock path in x-t plane, we can see that the corresponding characteristic converge into its path. This is a manifestation of the nonlinear wave-breaking phenomena which, in fact, makes shocks necessary. Information flows into the shock continuously via this mechanism, and then it is dissipated there. This means that there is a natural tendency in shocks to "narrow down". Thus, even though numerical viscosity will try to spread a shock (by the mechanism explained

above), the process can go only so far: the converging characteristics "squeeze" it and compensate the process eventually — this is the same process by which in "real" shocks the physical dissipation balances the nonlinear tendency to break.Thus numerical shocks become only slightly broadened — the "staircase" contains only very few steps.

On the other hand, a contact discontinuity path in x-t plane is parallel to the characteristics of its family. A contact does not receive any information, and there is no squeezing mechanism [22]. Therefore nothing prevents the numerical viscosity from smearing contacts without limit — the number of steps in the "staircase" will keep on increasing with each averaging/time step. In fact, it can be argued that a one-dimensional scheme in conservation form will smear a contact discontinuity at a rate of $O(n^{\frac{1}{r+1}})$, where r is the order of accuracy of the scheme, and n is the number of time steps.

Source of Wave Trains

The spurious numerical wave trains are generated because the shock transition layer alternates between being thick and thin as it passes through the grid.

Unlike a real physical shock, which can be viewed as a one-piece *solid* structure — one big step; a "staircase"–like numerical shock is a *loose* collection of several small shocks — numerical shocklets. The only requirement these shocklets should meet is that, at any time, the sum of their jumps be equal to the jump of the physical shock (This requirement follows from numerical conservation) . When such wobbly structure travels through the grid, it changes its width. The shock "breathes". It is more narrow at a cell interface, and wider in the interior of a cell. As the shock changes its width, the strengths of the numerical shocklets change as well: When the transition layer shrinks at a cell interface — the number of steps in the "staircase" decreases — the strength of the remaining shocklets should increase since the combined numerical jump must remain equal to the physical jump (the demand of numerical conservation). When, on the other hand, the transition layer dilates in the interior of a cell, new shocklets are formed, and the strengths of the old ones decline. In this process of strength redistribution and readjustment, perturbations are created which manifest themselves in the only way possible: as acoustic and entropy waves emitted by the shock. It is analogous to driving on a bumpy road. Any time the car/numerical shock hits the bump/node, the car/numerical shock starts to vibrate/emit waves. It should be emphasized, that the car vibrates because it *can*: The car has inner degrees of freedom — inner structure, and the parts of this inner structure can move relative to each other — readjust themselves. On the contrary, a rigid body, a boulder, does not have any inner structure capable of relative motion and readjustment. When such a boulder — a physical, one-point thin, shock — rolls on a bumpy road, it does not vibrate — emit

[22]Technically: Shocks correspond to the acoustic characteristics, which are genuinely nonlinear, while the contacts correspond to the particle paths characteristics, which are linearly degenerate [16].

waves.

The credibility of this explanation can be checked quantitatively. Let us revisit the problem of a single moving shock introduced in subsection 4.2.1. If $\Delta\xi$ is the cell size of the grid, and s is the shock speed, then the frequency at which the shock hits grid nodes is

$$f_{hit} = \frac{s}{\Delta\xi}. \tag{4.20}$$

Because the entropy and acoustic waves travel with velocities $\lambda_{2,R}$ (4.14) and $\lambda_{3,R}$ (4.16), respectively, the wavelengths of the spurious wave trains, l_2 and l_3 must be given by formulas:

$$l_2 = \frac{\lambda_{2,R}}{f_{hit}} = \frac{\lambda_{2,R}\Delta\xi}{s} \tag{4.21}$$

$$l_3 = \frac{\lambda_{3,R}}{f_{hit}} = \frac{\lambda_{3,R}\Delta\xi}{s} \tag{4.22}$$

The wavelengths of the spurious waves observed in the numerical experiments presented in subsections 4.2.2, 4.2.3 agree quite well with those computed by formulas (4.21),(4.22). To illustrate this, we compare the wavelengths of the spurious wave trains found in Examples 4.1–4.6 against those predicted with (4.21),(4.22). In all these Examples $\Delta\xi = \frac{1}{500} = 0.002$.

Example 4.1

$l_2 = 4.02 \times 10^{-2}$.

$l_3 = 8.65 \times 10^{-2}$.

Figure 4.4.

Both wave trains are present, although the amplitude of the acoustic train is very small — one needs a higher resolution, as in Figures 4.6, to observe the acoustic oscillations. Examination of the Figures 4.4 and 4.6 confirms the predicted wavelengths.

Example 4.2

$l_2 = 7.21 \times 10^{-2}$.

$l_3 = 1.67 \times 10^{-1}$.

Figure 4.9.

Only the entropy wave train can be clearly seen in Figure 4.9. The wavelength evaluated from this figure fits the predicted value perfectly.

Example 4.3

$l_2 = 4.45 \times 10^{-2}$.

$l_3 = 1.42 \times 10^{-1}$.

Figure 4.10.

Both wave trains are well resolved. Inspection of Figure 4.10 verifies the "theoretical" wavelengths.

Example 4.4

$l_2 = 7.30 \times 10^{-3}$.

$l_3 = 2.69 \times 10^{-2}$.

Figure 4.11.

As the high frequency acoustic wave train is getting damped by the numerical viscosity, its wavelength shifts towards longer wave numbers — from 0.027, immediately at the shock, to 0.0286, at the head of the wave train. This is probably the result of the scheme dispersion, see Remark 4.3 below. The wavelength of the entropy wave train has the same order of magnitude as the cell size of the grid, 0.002; therefore these oscillations are quickly killed by the scheme dissipation. From Figures 4.12 one can estimate the entropy wave train wave wavelength as 0.009. This turns out to be a little larger than its prediction, 0.007. We attribute the discrepancy to the the numerical dispersion (see Remark 4.3 below).

Example 4.5

$l_2 = 1.07 \times 10^{-3}$.

$l_3 = 8.86 \times 10^{-4}$.

Figure 4.13.

Both wavelengths are smaller than the cell size. Thus the waves are unable to propagate through the grid. On the other hand, side band waves of smaller amplitudes and lower frequencies can be easily observed in Figure 4.14. The wavelength of the acoustic oscillations is $l_3^{100} = 0.09 = 100 l_3$, while that of the entropy wave train is $l_2^{25} = 0.026 = 25 l_2$.

Example 4.6

$l_2 = 1.6 \times 10^{-4}$.

$l_3 = 0.0029$

Figure 4.15.

Both waves, having wavelengths of the order of the cell size or smaller, can not propagate through the grid. However, the entropy side band wave train with the wavelength $l_2^{125} = 0.02 = 125 l_2$, remains alive, being only slightly damped by the numerical viscosity.

Remark 4.3 These wave trains are different from the well known Gibbs phenomenon, although the behavior is quite similar.

The Gibbs phenomenon results from attempts to resolve a discontinuity by a finite number of Fourier

modes, and is the reason pseudo–spectral methods require filtering when dealing with discontinuities — a filter, essentially, replaces the discontinuity by a smoothed out transition, for which the Fourier series converges fast and uniformly. On the other hand, the oscillations we observe result from the width adjustment processes in the shock transition layer created by the averaging procedure of a conservative scheme.

It is also clear that the oscillations that we observe are also not of the type generated by finite difference schemes with insufficient damping: While equation (2.8) itself is inviscid hyperbolic, which means that it has neither dissipation (except at the shocks) nor dispersion, any finite difference approximation to it will posses a certain amount of dissipation and dispersion (see e.g. [29]). Thus, in particular, different wave numbers will travel through the grid with different speeds. A step function has a broad Fourier spectrum, decaying slowly as the wavelength goes to zero. Further, the phase relations between the different components is important, as the convergence of the Fourier series of a discontinuous function is conditional. A dispersive time evolution will quickly alter the phase relations, leading to oscillations — unless the scheme has sufficient damping to smooth out the discontinuity enough so that only a few modes are needed to represent it. These type of oscillation just described is, clearly, very related to the Gibbs phenomenon, as it also has to do with the poor (conditional) convergence of the Fourier series of a discontinuous function.

It should be noted that numerical dissipation and dispersion manifest themselves in our experiments as well. The fact that there is dissipation has been pointed out many times. The presence of dispersion can be seen in some of Examples 4.1 — 4.6 where the wavelength of the oscillations changes slightly as they move away from the shock. This was also pointed out earlier.

Source of Bumps

A finite size transition layer is the culprit in the bump generation process as well. At the initial moment the shock is one-node narrow — the physical shock and the numerical shock, which consists of one shocklet, coincide. When the numerical shock begins to move, it starts to develop the transition layer — the process was described above. While the shock is broadening, new shocklets are formed, and the initial shocklet is getting weaker (due to numerical conservation) at the same time. The readjustment and redistribution of the strength causes the shock to give off extra amount of entropy and pressure, which results in the developing of the entropy and acoustic bumps.

Remark 4.4 We can look at the spurious wave generation process from more "gas dynamical" perspective: To develop a transition layer, the shock must broaden. But this means smearing is occurring to the solution, and this is equivalent to the creation of entropy. This extra entropy, created at the shock location, is the entropy bump seen in the experiments. At the same time, to broaden, the shock must "push" some of the gas to both sides (think of what must happen with the

density profile), and this creates a pressure (acoustic) bumps.

The process of broadening of the shock and creation of the transition layer takes place only once, at the beginning of the simulation. When the layer is fully developed, the bumps are completed as well and grow no more. After that, as the shock moves through the grid, the layer needs only to readjust itself to the local positioning of the shock (relative to the computational nodes.) Those readjustments lead to the wave trains generation — the process was described earlier. But, because those secondary readjustments are usually smaller that the initial adjustment, the wave trains are, as a rule, of a smaller magnitude than the corresponding initial bumps.

A similar reasoning explains the "horns" generated by a shock collision. When an interaction of two numerical shocks occurs, the "staircase" – like structure of the shocks results in a sequence of collisions which take several time steps rather than one (as is the case for the exact solution, where the process is instantaneous). Such spread out in space and time interaction should produce lots of parasitic entropy: The scheme needs to "dismantle" the two "old" transition layers and then create the two "new" ones. Plenty of adjustment work to do, all of which generates entropy, which results in developing "horns". This is very similar to the bump generation by a shock coming into motion. However, the "horns" are usually larger than the "bumps", as should be expected because the former normally require more adjustment work. Furthermore, in the case of the "horns", the adjustment occurs in a several "stages", the collision being a complicated process, which accounts for the fact that more than one "horn" is generated — and why the number should be sensitive to the scheme: Different schemes will have different transition layers structures and will handle the details of the interaction differently.

Remark 4.5 It should be noted, that if two interacting shocks are of different strength, then the spurious entropy will be washed away by the mean flow, even as it is being generated. But if the shocks are of about the same strength, the flow velocity between the shocks will be very small and the parasitic entropy will stay and interact with other waves.

4.2.7 Why Are These Oscillations Bad ?

As the examples show, the spurious waves — in particular entropy waves — can be really big. It turns out that none of the schemes introduced above is able to meet requirement **(II)** posed at the beginning of this chapter. Therefore, we could not use these schemes for our purpose — long time numerical simulation, involving interactions of shocks, contacts, acoustic, and entropy waves.

Furthermore, all the schemes we studied in this chapter experience big difficulties dealing with contact discontinuities — contacts steadily diffuse (see Remark 4.2) as the simulation progresses. Thus, in contacts, we encounter another peril if we are to take on a long time numerical simulation:

They will eventually take over the whole computational domain, unless some (unknown) nonlinear mechanism stops their dilation.

The only way to overcome these problem is to develop a special one-dimensional code, where the production of numerical parasitic entropy (and the spread of the contact discontinuities) are minimized and controlled. Such a code does not quite exist as "over the counter" technology.

Another reason for developing a special code is the following: As can be seen from our experiments, the amplitude of the parasitic entropy waves can be of the same order of magnitude as that of the acoustic waves in many typical shock — flow disturbance interaction problems. These types of problems are very important now because of their relevance to a wide range of noise generation phenomena by fluid flows. Among them are the interaction of turbulence with shocks (which creates a phenomenon known as "broad band shock noise"); the high amplitude, narrow band phenomenon known as "screech" (self-induced supersonic flow oscillations in nozzles and on airfoils which can produce high noise levels); combustion instabilities in engines with supersonic inlets (which lead to the phenomenon known as "buzz"), and so on. Thus the suspicion appears that if one carries out a numerical simulation of a problem of this kind using a regular Godunov-type scheme [23], then one can easily get into trouble. Because the spurious entropy waves will interact nonlinearly (and very strongly, in fact — see chapter 3) with the acoustic waves, one could end up with a computed flow whose acoustic component has nothing to do with the true one.

Thus, the question arises: How badly can those spurious waves alter the results of Gas Dynamics flow simulations? With a code as desired, one could compare its "long time" performance against that of a general purpose code of the type generally used. Thus one would be able to study how serious the spurious entropy production affects the performance of a general purpose code over a long time computation could be.

Of course, one may ask: If you have a code like that, why would you worry about using the current "general purpose" codes, with all their entropy noise generation problems, etc.? The answer is that, in developing the special purpose code, we can use extensively special properties of the one-dimensional equations, while general purpose codes must be able to operate in more than one dimension. Indeed, it is not clear at all that *all* the ideas behind the new code we have developed (which satisfies all the requirements and is described in the next chapter) can be extended to more than one dimension.

4.2.8 Previous Research

In this subsection we review what was known about the problems we have extensively analyzed in this chapter. As we wrote at the beginning of 4.2, most of these problems have been pointed out in

[23] For that matter, all the currently known conservative numerical algorithms that can deal properly with shocks, seem to have the same problems of spurious entropy generation and contact spreading.

the literature, but the rather large size the produced errors can take under some circumstances has not been given sufficient attention, in our judgment.

Bumps and "Horns"

The history of the bumps and "horns" is as old as Godunov's method itself. In the same paper where he first introduced the method [6], Godunov also reported the appearance of bumps and "horns" generated by a shock coming into motion and for a shock reflecting on a rigid wall.

Having conducted computations of a steady plane shock moving with respect to the grid, Godunov reports[24]

> ... if the simulation starts with a step profile, satisfying the Rankine-Hugoniot conditions (with the step located at the point 0), then in several time steps the stationary profile for all computed variables will form. As time passes, this profile will move with speed equal to the shock speed, with the same pressure and velocity jumps. But near the point 0 a dent remains on the graph of τ [25]. This has the following explanation: In the transient process of formation of the stationary profile, the scheme "makes a mistake" in choosing the entropy, which is conserved in the smooth region behind the shock. In the smooth region the scheme is accurate enough to reflect this conservation of entropy. After the transient period is over the pressure behind the shock achieves the right value. However, $p\tau^{\gamma}$ [26] is incorrect, which leads to the appearance of the dent in the graph of τ [27].

Godunov also observed the dents in the problem of the reflection of a shock on a rigid wall.

He suggested, that, because those dents occupy only a few cells, grid refinement would make those dents insignificant. Unfortunately, it seems that Godunov never looked at the entropy profiles. As our experiments reveal – subsections 4.2.2 and 4.2.4, while the density suffers only from relatively small dents, the entropy can be plagued with really huge bumps and "horns", in which case a simple mesh refinement would hardly help. Of course, the fact that even small entropy variations can be very important is relatively new knowledge, as explained in chapter 3.

Wave Trains

Spurious oscillations in computations of strong shocks moving with respect to the grid were reported by Woodward and Colella [35] in 1984. We quote from the authors (p. 141):

---

[24] Our translation from Russian.

[25] The specific volume, defined as $\tau = \frac{1}{\rho}$.

[26] This is, basically, the entropy. See (2.5).

[27] The dent stayed at $x = 0$ because Godunov's scheme, as presented in [6], used the Euler equations in Lagrangian coordinates.

> When strong shocks travel slowly relative to the grid ... the shock structures must be broadened to about 2 zone[28] widths in the worst cases, in order to prevent post-shock oscillations of about 1 to 2% in amplitude from developing. Such post-shock oscillations cannot be effectively damped behind the shocks, because the wavelength of the oscillations are many zone widths in the case of slowly moving shocks. They must therefore be prevented by broadening the shock structure somewhat. This broadening is performed by two methods. First, the interpolated parabolic structures are flattened somewhat. Second, the grid to which the re-mapping is performed is jiggled slightly about its original position as the shock passes, so that the effective speed of the shock relative to the grid cannot vanish.

In other words, this excerpt says that

- Woodward and Colella observed spurious oscillations generated by a shock when it moves slowly with respect to the grid. The wavelength of the oscillations is long — many cell sizes. Thus, once produced, they cannot be damped by the numerical viscosity of the scheme.
- When the shock moves fast, the wavelength of the oscillations produced is short. Thus they are killed by the numerical dissipation.
- Woodward and Colella suggests two ways to cope with the problem. First, to flatten the interpolation polynomials [29], which is equivalent to adding additional numerical dissipation at the shock. Second, to **make the grid itself move** slightly. This will increase the **relative** speed of the shock with respect to the grid, causing a shift in wavelength of the oscillations towards the short wave end. These higher frequency oscillations would then be killed by the *natural* numerical dissipation of the scheme. There is no need to add any extra explicit numerical viscosity. Both methods have the effect of broadening the numerical shock width (equivalent to a local nonlinear viscosity being added at the shock location).

Unfortunately, the first way suggested here has a bad side effect (that makes it totally inadmissible for us.) While it solves the wave train generation problem, it also makes the shock collision problem — "horns" production — much worse by broadening the shock (see subsection 4.2.6). The second way does not resolve all the difficulties of the problem: in some cases, even a slowly moving shock can produce fairy large entropy waves (see Example 4.6). However, this second approach contains a promising way to address the problem — an attempt to **move the grid**

Recently Meadows, Caughey, and Casper [21] (1993), reported spurious wave trains in computations using ENO schemes [11]. These authors studied the effect that different Riemann solvers have

[28] Same as our cells.

[29] The interpolation polynomials are produced during the reconstruction step (see subsection 4.1.8). Woodward and Colella used PPM [3], which employs parabolic (second order) polynomials for reconstruction.

on the spurious noise. Because ENO schemes use an adaptive stencil to reduce spurious oscillations [11], modifications of this adaptive procedure were examined. The effects of shock pressure ratio, Courant number, grid spacing, and shock speed on the amplitude and frequency of the wave trains was also described.

These authors conducted thorough examination of the problem and identified the source of the spurious wave trains. We quote

> The spurious error is due to the discrete motion of the shock moving through the mesh. When the shock is located at a cell interface, it is extremely thin. As it moves through the cell interior, it smears out, weakens in strength, and entropy and pressure waves convect downstream.

The authors conducted thorough examination of the problem. They identified the source of the spurious wave trains.

> The spurious error is due to the discrete motion of the shock moving through the mesh. When the shock is located at a cell interface, it is extremely thin. As it moves through the cell interior, it smears out, weakens in strength, and entropy and pressure waves convect downstream.

Meadows, Caughey, and Casper also presented formulas for the wavenumbers of the wave trains, basically the same as: (4.21), (4.22). However, as Woodward and Colella in [3], Meadows, Caughey, and Casper failed to see considerable side band oscillations when the shock speed through the mesh is comparable to the flow speed, as we observed in (Example 4.6).
They draw the following conclusion:

> Because the amplitude of the spurious entropy wave is a function of the shock speed relative to the grid, the obvious method of eliminating the spurious wave is to **move the computational grid** [30] with the shock during the calculation.

Contacts

The first attempt to curb the diffusion of contact discontinuities was made in 1978 by Harten [7]. In order to prevent the excessive smearing of a linear discontinuity, one can artificially induce convergence of the numerical characteristic field at each monotone strip of the solution. To do this one has to be very careful — requires a lot of checking during the calculation — to not generate non-physical discontinuities by applying too much artificial convergence (for example, where it need not be used at all).

[30] The bold face is ours

Another drawback of this idea is: Shocks should be the only structures in the inviscid Euler equations capable of energy dissipation and entropy production. However, this idea effectively makes the contacts into shock-like structures, potentially allowing them to dissipate and produce entropy! This is directly opposite to the requirement **(II)** on the code we posed at the beginning of this chapter — keep numerical entropy production as low as possible.

In 1989 Harten took a different approach towards this problem — subcell resolution [10]. The main idea here is to **detect** the contact discontinuities. Then, making use of the knowledge of the position and structure of the contact, **propagate** it without smearing. The bottom line is: the scheme makes an attempt to resolve a contact as an almost **perfect**, one–node, discontinuity, by supplying the information of where the contact should be. Every time step the contact is "corrected" using estimates for the location of the "ideal" contact from the information available from the prior step . This "correction" balances the diffusion.

Chapter 5

New Scheme

The previous chapter implies the conclusion: The archenemy is adjustment processes, and they result from averaging across discontinuities. Unfortunately, the averaging is a principal step of Godunov-type schemes (see subsection 4.1.8). In can not be avoided if one wants to use a conservative method. Thus, if we want to stay with Godunov-type schemes and, at the same time, to avoid the spurious waves described in section 4.2, we should keep away from averaging across discontinuities. It can be achieved if and only if, at any time level, there is no cell with a discontinuity inside it — all discontinuities should always stay at the grid nodes. The only way to accomplish this is to allow grid nodes move with discontinuities. Then the speed of a discontinuity relative to a grid is always zero, the discontinuity does not move with respect to the grid, and the discontinuity is always perfectly resolved, it has one node width. No movement with respect to the grid — no adjustment processes — no noise generation.

The same strategy, moving nodes with discontinuities, is certain to solve the problem of the contact diffusion. To stop numerical diffusion, we need something to counteract it (in case of shocks, the balancing mechanism is nonlinearity; contacts, being linear discontinuities, lack such diffusion offsetting mechanism). But when at each time, we compute the exact position of a contact at the next time level and then place the node at this position, we never average across the discontinuity, the contact is never wider than one node, the diffusion never has a chance to start!

And lastly, the same approach should be successful in eradicating the noise resulting from formation processes, such that two shock collision. If *all* discontinuities are located at nodes up to the moment of the interaction, when they coalesce into one node, and then, immediately after the interaction, *all* new discontinuities emerged are placed at grid nodes — the interaction is perfectly resolved; then, till the end and from the beginning, the discontinuities are one-node wide and do not move relative to the grid. It means, there are no readjustments in the interaction process — there is nothing to readjust, the discontinuities are always perfect: It is preposterous to readjust something

which is perfect. And again, no readjustment - no noise.

To be able to fight the spurious oscillations and the numerical diffusion of contacts, the scheme needs to meet the following requirements.

1. When a discontinuity develops, it should be perfectly resolved from the cradle. In any process of formation — a shock starts to move or a new discontinuity is generated as a result of interaction — a new discontinuity must be represented by **one** node from the very beginning.

2. Once a discontinuity was created as a perfect discontinuity, it should remains such for the rest of its life. At any time a discontinuity has to be perfectly resolved — located at one node. The discontinuity does not move **relative** to the grid. The grid node moves with the discontinuity.

3. A perfect discontinuity should remains perfect till the moment it dies. If two discontinuities are to cease in a collision, they should remain one-node resolved up to the moment they coalesce into one node. (From this node new discontinuities should be born, which ought to be perfectly resolved from the very first moment, see 1, and so on).

Therefore, our objective is to blend a Godunov type scheme (nonlinear upwind and numerical conservation notions) with the front tracking concept (moving the grid nodes with a discontinuity).

Remark 5.1 To emphasize the main principles laid behind the new scheme and not to get lost in smaller details, only a first order scheme — generalization of Godunov's method rather than high resolution Godunov type schemes is devised in this chapter. Extensions to high resolution schemes will be briefly outlined in subsection 5.3.6.

The outline of the new scheme seems quite obvious. Given an array of constant states at time $t = t^n$, we solve the Riemann problem at each interface. From the solution of the Riemann problem we know the speeds of the discontinuities, and, hence, can compute the positions of the discontinuity at the advanced time level $t = t^{n+1}$. Then, when a new grid, corresponding to $t = t^{n+1}$, is being constructed, one of its nodes must be placed at the position of each discontinuity, thus securing its perfect resolution. Once the new grid has been formed, the rest of the algorithm becomes similar to the fixed grid case: From the solution of the Riemann problems, the fluxes are computed through either cell interface [1], and the flux difference is added to the old cell average to obtain the new cell average at the time level $t = t^{n+1}$.

[1] In contrast with the fixed grid case, a cell interface can appear slanted when drawn on x-t-plane — the node can move with the discontinuity.

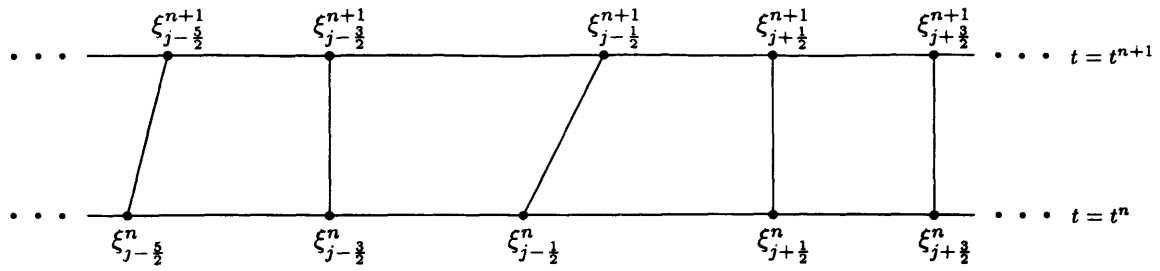

Figure 5.1: Moving numerical grid.

Remark 5.2 When we move the nodes with a discontinuity, we need to know where the discontinuity will be at the next time level. Fortunately, we already have all the information we need once we solve the corresponding Riemann problem.

In Godunov-type schemes most of the structure of the Riemann solver is not used. The exact solution is averaged over each cell. Most of the information of the detailed local structure of the solution obtained by solving the Riemann problem is thrown away. We won't discard this information. We will use it to propagate the discontinuity.

We would like to stress that we are not going to do any extra computations which would slow down the algorithm performance, we will only use the information which was already obtained anyways for different purposes.

Unfortunately, the scheme outlined above, except for applications to a scalar equation — m = 1 in (2.8), is a totally useless theoretical construction. The flaw is flagrant: In case the of a system of conservation laws, the solution to a Riemann problem is a fan of m-waves; so, if we decided to resolve *all* the discontinuities, we, in general, would end up with immense node proliferation. To avoid this, we should, somehow, limit the number of discontinuities to be perfectly resolved. We ought to come up with a criterion which would allow us to select the most significant, in some sense, discontinuities. This is the most challenging part, and we postpone its treatment for a while. Our immediate efforts will be concentrated on two easier tasks. In the next section we modify Godunov's scheme so we would be able to use it on a moving grid. After this, we introduce the algorithm of grid reconstruction for a scalar equation – in which case there is always one wave at each discontinuity, thus there is no node proliferation problem. Only after we tackle those two problems, we will extend the algorithm to a general system of conservation laws — addressing the significant wave criterion problem.

5.1 Godunov's Scheme on Moving Grid

As before, we consider the numerical integration of initial value problem (2.8) However, this time we use a numerical grid which can be different at different time levels $\{\xi^n_{j-\frac{1}{2}}\}$ (Figure 5.1).

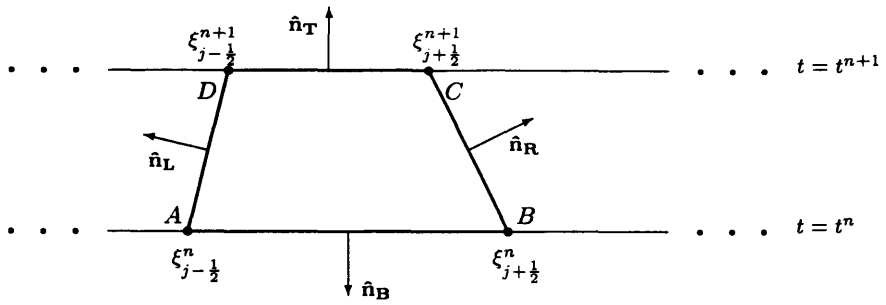

Figure 5.2: Godunov's method on a moving grid.

Remark 5.3 Before, when a fixed grid algorithm was considered, $\xi_{j-\frac{1}{2}}$ denoted the position of the j-th node, which did not change with time. Now, when the node is permitted to move, the notations $\xi_{j-\frac{1}{2}}^n$, $\Delta\xi_j^n$, etc. are used. The superscript n indicates that the node location can change from one time level to another.

Remark 5.4 If node $\xi_{j-\frac{1}{2}}^{n+1}$ from time level t^{n+1} is connected with node $\xi_{j-\frac{1}{2}}^n$ from time level t^n, then we call node $\xi_{j-\frac{1}{2}}^{n+1}$ a **child** of $\xi_{j-\frac{1}{2}}^n$, and $\xi_{j-\frac{1}{2}}^n$ a **parent** of $\xi_{j-\frac{1}{2}}^{n+1}$. When we say "nodes $\xi_{j-\frac{1}{2}}^{n+1}$ and $\xi_{j-\frac{1}{2}}^n$ are in a child-parent relationship" , it means that we are to compute the fluxes through the line connecting $\xi_{j-\frac{1}{2}}^n$ and $\xi_{j-\frac{1}{2}}^{n+1}$.

5.1.1 Main Formula

As in subsection 4.1.5, we should find the equation which gives us cell averages at advanced time level t^{n+1}, once we know cell averages at time level t^n. And, as in the case of a fixed grid method, the integral form of conservation laws should be employed. But this time, instead of integrating the system of conservation laws over a rectangle (Figure 4.2), we ought to integrate it over a trapezoid with vertices A($\xi_{j-\frac{1}{2}}^n, t^n$), B($\xi_{j+\frac{1}{2}}^n, t^n$), C($\xi_{j+\frac{1}{2}}^{n+1}, t^{n+1}$), D($\xi_{j-\frac{1}{2}}^{n+1}, t^{n+1}$), (Figure 5.2).

To do this, we, first, introduce a vector

$$\mathcal{F} = (f(u), u), \tag{5.1}$$

and rewrite system (2.8) as a divergence equation

$$\nabla \cdot \mathcal{F} = 0, \tag{5.2}$$

in which the divergence is to be computed in the (x, t) plane. Then we integrate equation (5.2) over

the trapezoid and apply Gauss' theorem to obtain

$$\oint_{ABCD} \mathcal{F} \cdot \hat{\mathbf{n}}\, d\Gamma = 0 \tag{5.3}$$

where $\hat{\mathbf{n}}$ is the unit normal vector, pointed outside the trapezoid.

Contour integral in (5.3) can be written as a sum of four line integrals over the sides of the trapezoid:

$$\begin{aligned} &\int_{AB} \mathcal{F} \cdot \hat{\mathbf{n}}_{\mathbf{B}}\, d\Gamma \; + \int_{BC} \mathcal{F} \cdot \hat{\mathbf{n}}_{\mathbf{R}}\, d\Gamma \; + \\ &\int_{CD} \mathcal{F} \cdot \hat{\mathbf{n}}_{\mathbf{T}}\, d\Gamma \; + \int_{DA} \mathcal{F} \cdot \hat{\mathbf{n}}_{\mathbf{L}}\, d\Gamma = 0. \end{aligned} \tag{5.4}$$

Here

$$\begin{aligned} \hat{\mathbf{n}}_{\mathbf{B}} &= (0,-1), \qquad \hat{\mathbf{n}}_{\mathbf{R}} = \frac{(1,\; -\dot{\xi}^n_{j+\frac{1}{2}})}{\sqrt{1+(\dot{\xi}^n_{j+\frac{1}{2}})^2}}, \\ \hat{\mathbf{n}}_{\mathbf{T}} &= (0,+1), \qquad \hat{\mathbf{n}}_{\mathbf{L}} = \frac{(-1,\; \dot{\xi}^n_{j-\frac{1}{2}})}{\sqrt{1+(\dot{\xi}^n_{j-\frac{1}{2}})^2}}, \end{aligned} \tag{5.5}$$

are the unit outer normal vectors at the bottom, right, top, and left sides of the trapezoid, respectively. $\dot{\xi}^n_{j-\frac{1}{2}}$ denotes the velocity of the j-th node averaged over time interval Δt^n:

$$\dot{\xi}^n_{j-\frac{1}{2}} = \frac{\xi^{n+1}_{j-\frac{1}{2}} - \xi^n_{j-\frac{1}{2}}}{\Delta t^n}. \tag{5.6}$$

Now, using (5.1) and (5.5) we can treat each line integral of equality (5.4) separately.

On the bottom side x itself plays the role of a parameter

$$x = x, \quad t = t^n. \tag{5.7}$$

Here $x = \xi^n_{j-\frac{1}{2}}$ and $x = \xi^n_{j+\frac{1}{2}}$ correspond to the points A and B, respectively. The differential element is $d\Gamma = dx$. With this parameterization, we obtain:

$$\int_{AB} \mathcal{F} \cdot \hat{\mathbf{n}}_{\mathbf{B}}\, d\Gamma \; = \; - \int_{\xi^n_{j-\frac{1}{2}}}^{\xi^n_{j+\frac{1}{2}}} u(x,t^n)\, dx \tag{5.8}$$

Along the right side, it is convenient to take t as a parameter:

$$x = \xi^n_{j+\frac{1}{2}} + \dot{\xi}^n_{j+\frac{1}{2}}\,(t - t^n), \quad t = t. \tag{5.9}$$

Then $t = t^n$ and $t = t^{n+1}$ correspond to the points B and C, respectively, and $d\Gamma = \sqrt{1 + (\dot{\xi}^n_{j+\frac{1}{2}})^2}\,dt$. Then the line integral can be transformed as:

$$\int_{BC} \mathcal{F}\cdot\hat{\mathbf{n}}_{\mathbf{R}}\,d\Gamma \;=\; \int_{t^n}^{t^{n+1}} (f(u(\xi^n_{j+\frac{1}{2}} + \dot{\xi}^n_{j+\frac{1}{2}}(t-t^n),\ t)) - \dot{\xi}^n_{j+\frac{1}{2}} u(\xi^n_{j+\frac{1}{2}} + \dot{\xi}^n_{j+\frac{1}{2}}(t-t^n),\ t))\,dt \tag{5.10}$$

On the top side we, again, choose x as a parameter.

$$x = x, \quad t = t^{n+1}. \tag{5.11}$$

Here $x = \xi^n_{j+\frac{1}{2}}$ and $x = \xi^n_{j-\frac{1}{2}}$ correspond to the points C and D, respectively, and $d\Gamma = -dx$. The parameterization yields:

$$\int_{CD} \mathcal{F}\cdot\hat{\mathbf{n}}_{\mathbf{T}}\,d\Gamma \;=\; \int_{\xi^{n+1}_{j-\frac{1}{2}}}^{\xi^{n+1}_{j+\frac{1}{2}}} u(x, t^{n+1})\,dx \tag{5.12}$$

Finally, we parameterize the left side as

$$x = \xi^n_{j-\frac{1}{2}} + \dot{\xi}^n_{j-\frac{1}{2}}\,(t - t^n), \quad t = t, \tag{5.13}$$

with $t = t^{n+1}$ and $t = t^n$ corresponding to the points D and A, respectively. On the integration path $d\Gamma = -\sqrt{1 + (\dot{\xi}^n_{j-\frac{1}{2}})^2}\,dt$, and it follows that:

$$\int_{DA} \mathcal{F}\cdot\hat{\mathbf{n}}_{\mathbf{L}}\,d\Gamma \;=\; -\int_{t^n}^{t^{n+1}} (f(u(\xi^n_{j-\frac{1}{2}} + \dot{\xi}^n_{j-\frac{1}{2}}(t-t^n),\ t)) - \dot{\xi}^n_{j-\frac{1}{2}} u(\xi^n_{j-\frac{1}{2}} + \dot{\xi}^n_{j-\frac{1}{2}}(t-t^n),\ t))\,dt \tag{5.14}$$

Equations (5.8) and (5.12) reveal that the bottom and top side integrals are just the cell averages (4.3) multiplied by the cell widths:

$$\int_{AB} \mathcal{F}\cdot\hat{\mathbf{n}}_{\mathbf{B}}\,d\Gamma \;=\; -\Delta\xi^n_j\;\bar{u}^n_j, \tag{5.15}$$

$$\int_{CD} \mathcal{F} \cdot \hat{\mathbf{n}}_{\mathbf{T}} \, d\Gamma \;=\; \Delta\xi_j^{n+1} \; \bar{u}_j^{n+1}, \tag{5.16}$$

On the other hand, equations (5.14) and (5.10) show that the right and left side integrals turn out to be the numerical fluxes multiplied by the time step:

$$\int_{BC} \mathcal{F} \cdot \hat{\mathbf{n}}_{\mathbf{R}} \, d\Gamma \;=\; \Delta t^n \; \bar{f}_G(\dot{\xi}^n_{j+\frac{1}{2}};\; u_j^n, u_{j+1}^n) \tag{5.17}$$

$$\int_{DA} \mathcal{F} \cdot \hat{\mathbf{n}}_{\mathbf{L}} \, d\Gamma \;=\; -\Delta t^n \; \bar{f}_G(\dot{\xi}^n_{j-\frac{1}{2}};\; u_{j-1}^n, u_j^n) \tag{5.18}$$

here, the numerical flux function, $\bar{f}_G$ is given (cf (4.8)) by

$$\begin{aligned} &\bar{f}_G(\dot{\xi}^n_{j-\frac{1}{2}};\; u_{j-1}^n, u_j^n) \\ &\qquad = \frac{1}{\Delta t^n} \int_{t_n}^{t_{n+1}} (f(u(\xi^n_{j-\frac{1}{2}} + \dot{\xi}^n_{j-\frac{1}{2}}(t-t^n),\; t)) \;-\; \dot{\xi}^n_{j-\frac{1}{2}} \; u(\xi^n_{j-\frac{1}{2}} + \dot{\xi}^n_{j-\frac{1}{2}}(t-t^n),\; t)) \; dt, \end{aligned} \tag{5.19}$$

If, for a system of conservation laws (2.8), we introduce a generalized flux at direction [2] $\dot{\xi}$ as

$$f_G(\dot{\xi}; u) = f(u) - \dot{\xi} u \tag{5.20}$$

then the numerical flux (5.19) is the time average of the generalized flux through the cell boundary — which is moving from point $x = \xi^n_{j-\frac{1}{2}}$ at the speed of $\dot{\xi}^n_{j-\frac{1}{2}}$ — over the time interval (t_n, t_{n+1}).

Plugging in Equations (5.15-5.18) into (5.4) and dividing the result by $\Delta\xi_j^{n+1}$ we obtain the main formula of Godunov's method on a moving grid:

$$u_j^{n+1} = \frac{\Delta\xi_j^n}{\Delta\xi_j^{n+1}} \; u_j^n - \frac{\Delta t^n}{\Delta\xi_j^{n+1}} \; (\bar{f}_G(\dot{\xi}^n_{j+\frac{1}{2}};\; u_j^n, u_{j+1}^n) - \bar{f}_G(\dot{\xi}^n_{j-\frac{1}{2}};\; u_{j-1}^n, u_j^n)) \tag{5.21}$$

The equation (5.21) looks almost the same as the one for the fixed grid (4.7). The only differences are that, first, generalized, instead of regular, fluxes are used, and, second, the coefficient in front of u_j^n is not, in general, equal to one. Therefore, as in the fixed grid case, given an array of constant states $\{u_j^n\}$, we solve at each cell interface $\xi^n_{j+\frac{1}{2}}$ the Riemann problem for two neighboring constant states u_j^n and u_{j+1}^n. Once we know the Riemann problem solution, we can easily calculate the flux

[2] we mean a direction on x-t plane.

through the interface, the segment connecting $(\xi^n_{j+\frac{1}{2}}, \xi^{n+1}_{j+\frac{1}{2}})$ in x-t plane. Now, given the fluxes at the interfaces of a cell, we, using (5.21), compute cell averages u^{n+1}_j at the advanced time level t^{n+1}.

In the remaining two subsections we introduce a CFL-type condition and show how to compute integral in the numerical flux expression (5.19).

5.1.2 Choosing Time Step

As in the fixed grid case the stability requirements impose some restrictions on the size of the time step. The CFL-like time step constraint for a Godunov's scheme on a moving grid is similar to the fixed grid case CFL condition presented in subsection 4.1.6. When put into words, it sounds as follows.

The CFL condition. The time step should be chosen small enough, so that no interface belongs to the domains of influence of more than *one* Riemann problem over time interval (t^n, t^{n+1}).

As we pointed out in subsection 4.1.6, the CFL-like time step constraint assures that no interaction occurs at the interface over time interval (t^n, t^{n+1}). Therefore, the density distribution over the interface (which is a segment when drawn on the x-t plane) is defined uniquely by the local Riemann problem. Because the solution of a Riemann problem is easy to compute, the numerical flux through the interface (5.19) can be calculated without difficulties (see subsection 5.1.3 below) [3].

Most of the time in computations with a Godunov's type scheme on a moving grid, one encounters the situation when the interface is contained inside the domain of influence of its own Riemann problem. However, another situation, when the interface is contained inside the domain of influence of a neighboring Riemann problem, can rarely occur as well (see the end of subsection 5.3.2 below). Depending on the case, two different types of constraints on the time step will ensue from the CFL condition written out at the beginning of this subsection. We will separately analyze either case in the following two subsubsections.

Interface Inside Its Own Riemann Problem

If the j-th interface is inside the domain of influence of the Riemann problem centered at the j-th node, $x = \xi^n_{j-\frac{1}{2}}$ (Figure 5.3), then the CFL-like time step constraint demands that the domains of influence of two neighboring problems, $(j-1)$-th and $(j+1)$-th do not contain the child of the j-th node, $x = \xi^{n+1}_{j-\frac{1}{2}}$, at $t = t^{n+1}$ — no wave emanating from the neighboring interfaces, $x = \xi^n_{j-\frac{3}{2}}$ and $x = \xi^n_{j+\frac{1}{2}}$, at $t = t^n$ can reach the j-th interface before $t = t^{n+1}$:

[3] In the constant coefficient case, the CFL-like time step constraint can be proven to be sufficient for stability and has been found numerically to be sufficient for the method to be stable in the non-constant case.

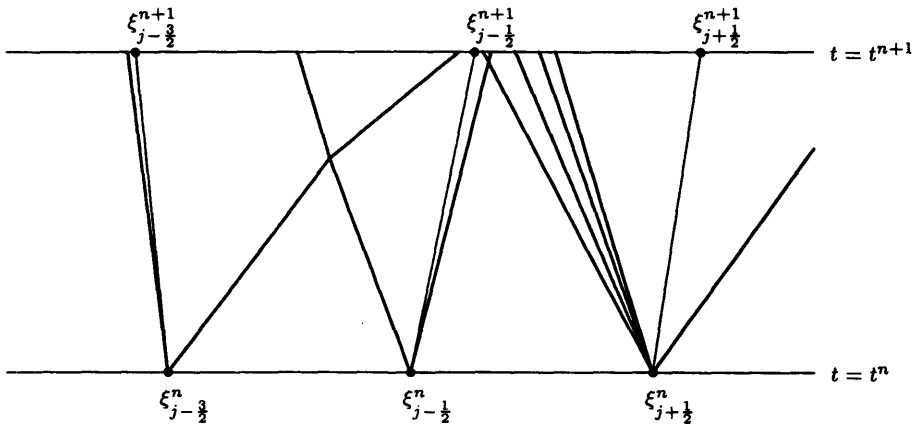

Figure 5.3: The CFL condition on a moving grid. Interface inside its own Riemann problem.

$$\xi^n_{j-\frac{3}{2}} + \lambda^{max}_{j-\frac{3}{2}}\,\Delta t \;<\; \xi^{n+1}_{j-\frac{1}{2}} \;<\; \xi^n_{j+\frac{1}{2}} + \lambda^{min}_{j+\frac{1}{2}}\,\Delta t \tag{5.22}$$

where $\lambda^{min}_{j-\frac{1}{2}} = \min(\lambda_1(u^n_{j-1}), \lambda_1(u^n_j))$, and $\lambda^{max}_{j-\frac{1}{2}} = \max(\lambda_m(u^n_{j-1}), \lambda_m(u^n_j))$ are, respectively, the lower and upper bounds of the velocities of the waves emerging in the j-th Riemann problem.

Interface Inside Neighboring Riemann Problem

A typical example is shown in Figure 5.4, where both the $(j-1)$-th and j-th interfaces are inside the domain of influence of the Riemann problem originated at the $(j-1)$-th node. For this situation to happen, the speed at which the j-th interface moves must be less than the the velocity of the leftmost wave of the j-th Riemann problem:

$$\dot{\xi}^n_{j-\frac{1}{2}} < \lambda^{min}_{j-\frac{1}{2}}. \tag{5.23}$$

As it is clear from Figure 5.4, only the second inequality of (5.22) should be satisfied to meet the CFL-like time step constraint, written out in the beginning of this subsection, for the j-th interface:

$$\xi^{n+1}_{j-\frac{1}{2}} \;<\; \xi^n_{j+\frac{1}{2}} + \lambda^{min}_{j+\frac{1}{2}}\,\Delta t. \tag{5.24}$$

Unlike the previous case, the interface inside the domain of influence of its own Riemann problem, the location of the child of the j-th node, $\xi^{n+1}_{j-\frac{1}{2}}$, does not put any restriction on the position of the

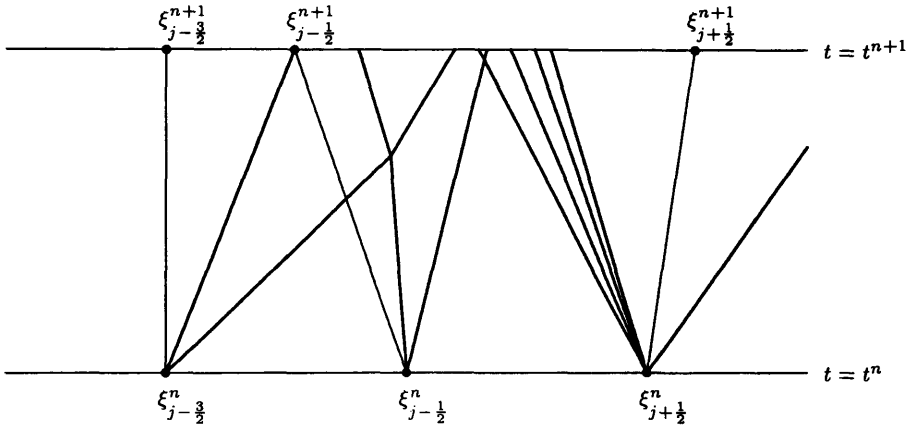

Figure 5.4: The CFL condition on a moving grid. Interface inside neighboring Riemann problem.

rightmost wave of the $(j-1)$-th Riemann problem at time t^{n+1}. If no constraints are imposed on the location of the rightmost wave of the $(j-1)$-th Riemann problem, some stability problems can arise. To avoid it, we should require that this wave never goes beyond some point. It will be shown at the end of subsection 5.3.2 (equation (5.45) and the text above it) that the speed at which the j-th node moves, $\dot{\xi}^n_{j-\frac{1}{2}}$, must be always negative, and the missing constraint is: The rightmost wave of the $(j-1)$-th Riemann problem should never go beyond the position of the j-th node at time t^n:

$$\xi^n_{j-\frac{3}{2}} + \lambda^{max}_{j-\frac{3}{2}} \Delta t \; < \; \xi^n_{j-\frac{1}{2}}. \tag{5.25}$$

The detailed analysis of this situation will be given at the end of subsection 5.3.2.

5.1.3 Flux Computation

Two cases introduced in subsection 5.1.2 are to be treated separately in flux computations.

Interface Inside Its Own Riemann Problem

When the j-th interface is inside the domain of influence of the Riemann problem centered at the j-th node, $x = \xi^n_{j-\frac{1}{2}}$ (Figure 5.3), then the integral in (5.19) becomes trivial since $u(x,t)$ is *constant* along the ray $(x - \xi^n_{j-\frac{1}{2}})/t = \dot{\xi}^n_{j-\frac{1}{2}}$ for all times from interval (t_n, t_{n+1}) (see subsection 4.1.5 for the argument).

The constant value of $u(x,t)$ along the cell interface moving with the velocity of $\dot{\xi}^n_{j-\frac{1}{2}}$ depends only on the data u^n_{j-1} and u^n_j for the Riemann problem and the speed of the interface, $\dot{\xi}^n_{j-\frac{1}{2}}$. If we

denote this value by $u^*(\dot{\xi}^n_{j-\frac{1}{2}};\ u^n_{j-1}, u^n_j)$, then the numerical flux (5.19) reduces to the generalized flux of the system, (2.8), computed at this constant value and at direction $\dot{\xi}^n_{j-\frac{1}{2}}$:

$$\bar{f}_G(\dot{\xi}^n_{j-\frac{1}{2}};\ u^n_{j-1}, u^n_j) = f_G(\dot{\xi}^n_{j-\frac{1}{2}};\ u^*(\dot{\xi}^n_{j-\frac{1}{2}}; u^n_{j-1}, u^n_j)) \tag{5.26}$$

Interface Inside Neighboring Riemann Problem

Once again, the example shown in Figure 5.4 is considered. Both the $(j-1)$-th and j-th interfaces are inside the domain of influence of the Riemann problem originated at the $(j-1)$-th node.

We can not use a simple expression, like (5.26), to compute the numerical flux through the j-th interface. $u(x,t)$ is not constant along the interface anymore because the interface is not one of the rays coming out of the origin of the Riemann problem, $x = \xi^n_{j-\frac{1}{2}}$. On the contrary, the interface traverses such rays.

Therefore, the integral in (5.19) requires a careful treatment. Using the fact that the solution to a Riemann problem is composed of $m+1$ constant states separated by m waves — shocks, contacts, and rarefactions, we can decompose the integral in (5.19) into a sum of integrals, one integral for each constant state or rarefaction fan crossed by the interface.

The integrals across the constant states are trivial (integrand is constant). For a rarefaction fan, although non-constant, $u(x,t)$ has a simple dependence on x and t so the integral is easy to evaluate. Thus, all the integrals across the constant states and the rarefaction fans can be easily computed and summed up to obtain the average numerical flux through the j-th interface.

Remark 5.5 If we compute integral in (5.19) using the Riemann problem in the j-th interface rather than $(j-1)$-th, then we would obtain a simple expression like (5.26) for the numerical flux through the j-th interface. In such computation we neglect the influence of the waves originating from the $(j-1)$-th Riemann problem and reaching the j-th interface in time interval (t^n, t^{n+1}). When those waves are numerically artificial constructs introduced by the discretization process with the strength, $O(\xi)$, beyond the accuracy of the method; the error introduced by the omission of those waves lies within the accuracy of the method. Nevertheless, we usually take some pain to carry on meticulous flux computations and believe that they will reward us with somewhat more accurate results (see also Remark 5.15).

5.2 Self Adjusting Grid Method

Now it is time to specify the procedure, which, given the grid $\{\xi^n_{j+\frac{1}{2}}\}$ at time level $t = t^n$, constructs a new grid $\{\xi^{n+1}_{j+\frac{1}{2}}\}$ at the advanced time level $t = t^{n+1}$. It proves to be an uneasy task.

Suppose, **all** the nodes of the grid are allowed to move. If we simply follow the discontinuity paths originated at the nodes $\{\xi^n_{j+\frac{1}{2}}\}$, we can end up with a very distorted grid. In a few time steps

the grid can become highly non-regular — some nodes would cluster together, and some would be far apart. Those overpopulated and sparse regions can lead to serious problems: slow down of the performance, lost of accuracy, etc. To avoid those troubles, one should impose some restrictions, to keep the grid more or less regular. In 1983, Harten and Hyman [12] suggested an elegant solution to the regularization problem using an **underlying fixed mesh** approach as a part of their self adjusting grid method.

In this section we describe the concept of the underlying fixed mesh and illustrate the performance of the self adjusting grid method on the example of a single shock traveling relative to the underlying fixed mesh — the initial boundary value problem posed in subsection 4.2.1. As we said before, we illustrate the method on a scalar equation ($m = 1$ in (2.8)) rather than a system of conservation laws. The following subsection explains why we have chosen one equation instance.

5.2.1 Node Proliferation Problem and Weighted Average Approach

While the self adjusting grid method shows excellent performance when applied to a problem of a single shock motion, it behaves as bad as any stationary grid Godunov-type scheme (section 4.2) when dealing with discontinuity interaction problems (see subsection 5.2.4 below). The cause of this bad performance lies in the way the method handles the node proliferation problem. When at the j-th cell interface the Riemann problem is solved, and, in the worst case, m-waves emerging from the Riemann problem need to be perfectly resolved, the method compromises and creates only *one* new node at the advanced time level $t = t^{n+1}$ to be placed at the point that minimizes, in a certain sense, the overall diffusion in the Godunov-type scheme.

According to the algorithm, the node should be moved with the speed which is a weighted average of the m speeds of propagation of the waves. The weight assigned to a wave represents the relative significance of the wave in the Riemann problem. The weights used are monotone increasing functions of the strength of the wave that vanish if and only if the strength of the wave is zero.

It follows from this property that if a single shock [4] is located at the j-th node at time $t = t^n$, then the node should be moved with the speed of the shock. Thus, the scheme gives a perfect resolution to a single shock. On the other hand, if we have, for example, two strong discontinuities emerging from one node, which is exactly what happens after two shocks have collided, then the method, following the weighted average rule, is forced to commit the old crime — average across the both shocks, which leads to spurious entropy waves generation. So, the self adjusting grid method, while meeting requirement 2, fails to satisfy requirement 1. Actually, the method can not meet requirement 3 either, as we will see below (Remark 5.7).

Therefore, we conclude that the performance of the method is unsatisfactory for our purposes.

[4] which, particularly, means that all other $m - 1$ discontinuities are of zero strength

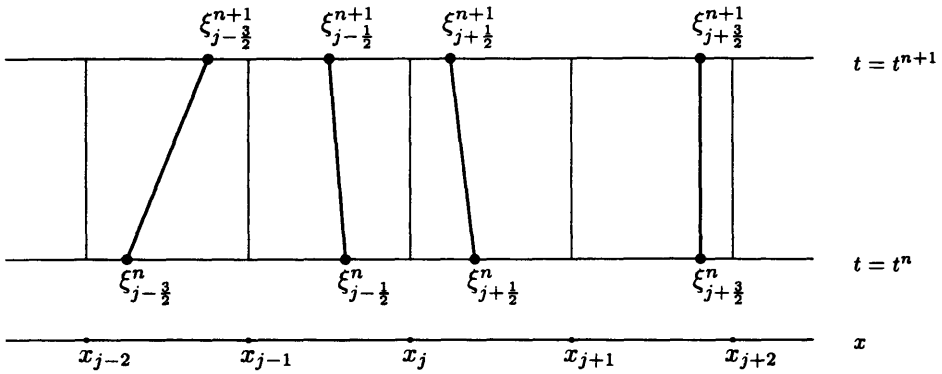

Figure 5.5: Underlying fixed mesh. Thin lines represent underlying fixed mesh and boxes. Filled circles represent nodes. Lines of intermediate thickness represent paths of interfaces.

On the other hand, the method has some attractive features, particularly the underlying fixed mesh concept, which we would like to borrow for our new scheme. Because all these features are already seen in the case of one scalar equation, and since, as we just mentioned, the method performs as bad as any fixed grid Godunov-type method when applied to a system of conservation laws (except for a case of a single moving discontinuity), it seems reasonable to illustrate the scheme on the example of a scalar equation.

5.2.2 Underlying Fixed Mesh

So far, the grid was uniquely defined at any time t^n by specifying the location of its nodes $\xi^n_{j+\frac{1}{2}}$. Now, following [12], we introduce another partition of the x-axis: $\{x_j\}$ (not necessarily uniform), which we will call the **underlying fixed mesh** (see Figure 5.5). A point x_j will be called the j-th **wall**, and an interval $(x_{j-1}, x_j]$ will be referred as the j-th **box**. It should be emphasized, that the walls do not change with time — no n superscript.

The underlying fixed mesh helps impose three principal constraints which play the leading role in preventing the grid from becoming disordered.

1. RELAY CONSTRAINT. The j-th node $\xi^n_{j-\frac{1}{2}}$ is constrained by

$$x_{j-1} < \xi^n_{j-\frac{1}{2}} \leq x_j \tag{5.27}$$

   for all j and n. Equation (5.27) implies (see Remark 5.4), that children live always inside the same box where their parents lived — once node's great-grandparent was born inside a box, the node has no chance to leave this box. A node cannot cross the walls of its box in its movement. A node stays in the same box all the time.

2. NO-EMPTY-BOX CONSTRAINT. At any time level, there must be no box without a node inside. If no discontinuity is located in the j-th box at time $t = t^{n+1}$, then the j-th node should be

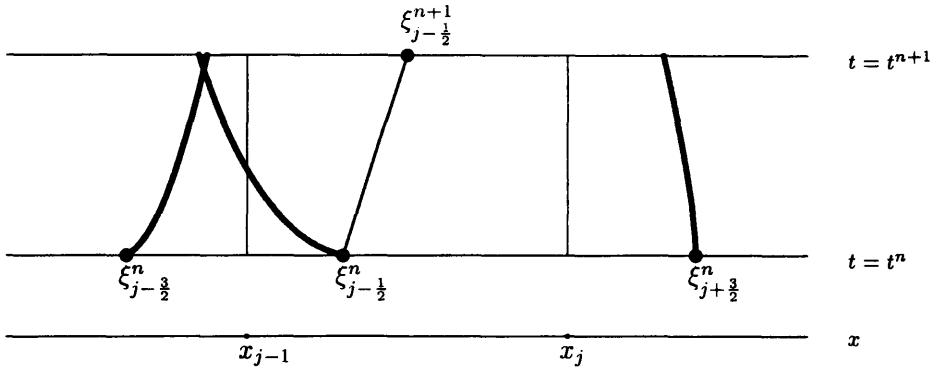

Figure 5.6: No-empty-box constraint. Thin lines represent underlying fixed mesh and boxes. Filled circles represent nodes. The line of intermediate thickness represents the path of the j-th interface. Thick lines represent discontinuity trajectories.

placed at the center of the box:

$$\xi^{n+1}_{j-\frac{1}{2}} = \frac{1}{2}(x_{j-1} + x_j) \tag{5.28}$$

(Figure 5.6). The purpose this constraint serves is transparent: to prevent the grid from getting too sparse.

3. NO-MULTI-NODE-BOX CONSTRAINT At any time, there must be only one node inside a box. If two or more discontinuities land inside the same box, the new single node is placed in the location which is a weighted average of the locations of the discontinuities in the box. The weight, like in subsection 5.2.1, is proportional to the strength of the discontinuity.

Remark 5.6 The last two constraints together are equivalent to the restriction requiring exactly one node in each box.

Remark 5.7 The No-multi-node-box constraint, similar to the weight averaging of the Riemann problem waves (see subsection 5.2.1), works perfectly well in case of a single discontinuity, while it fails to prevent cross-discontinuity averaging and resulting numerical noise production, when two strong discontinuities appear in the same box, which is particularly the case of two shocks moving on the collision course. It indicates, that the self adjusting grid method, in addition to being unable to meet requirement 1 (see subsection 5.2.1), is incapable of fulfilling requirement 3 as well.

Thus, the no-multi-node-box constraint proves to be too restrictive. While the first two constraints will be used in our new scheme, the no-multi-node-box constraint will be cast off. However, this constraint does a good job in a single moving shock case, which will be described in the next subsection.

5.2.3 Single Shock Motion

Suppose that at time $t = t^n$ a single shock is located at the j-th node $x_s^n = \xi_{j-\frac{1}{2}}^n$ (Figure 5.7). Due to the CFL condition (subsection 5.1.2), at the next time level $t = t^{n+1}$, the shock could either stay inside the j-th box or land in one of the immediate neighboring boxes: the $(j-1)$-th or $(j+1)$-th. If at $t = t^{n+1}$ the shock is still inside the j-th box, the j-th node will be placed at its location

$$\xi_{j-\frac{1}{2}}^{n+1} = x_s^{n+1}. \tag{5.29}$$

(Figure 5.7). When the shock eventually left the j-th box, then, because the relay constraint prohibits the j-th node to abandon its box, either $(j-1)$-th or $(j+1)$-th node will be put on the path of the discontinuity, depending on the way the shock moves — in the situation described in Figure 5.7, the shock moves to the right box during time interval (t^{n+2}, t^{n+3}), and at time $t = t^{n+3}$ the $(j+1)$-th node is placed on the shock path.

Actually, at the moment when the shock moved from the j-th box into its neighbor, (the $(j+1)$-th box in our example), there is no discontinuity inside the j-th box and two discontinuity in the $(j+1)$-th. However, the no-empty-box constraint places the j-th node in the center of the j-th box (see Figure 5.7),

$$\xi_{j-\frac{1}{2}}^{n+3} = \frac{1}{2}(x_{j-1} + x_j), \tag{5.30}$$

and the no-multi-node-box constraint, provided that the strength of the discontinuity, originated at the $(j+1)$-th node, is zero (a single shock movement is being considered), places the $(j+1)$-th node at the path of the discontinuity emerging from the j-th node (the single shock):

$$\xi_{j+\frac{1}{2}}^{n+3} = x_s^{n+3}. \tag{5.31}$$

So, as long as a single shock stays inside the j-th box, it is followed by the j-th node: At all these time levels, there is only one node inside the box which is situated at the shock path — the j-th node does its part of the job carrying the shock inside its box. Once the shock leaves the j-th box, say, for $(j+1)$-th box as in the case of Figure 5.7; the j-th node passes the shock to the $(j+1)$-th node and falls back in the center of its box, taking rest and waiting for the next shock to convey.

Remark 5.8 Nodes behave like members of a relay team. When one has gone its part of the distance, it, being relieved by its teammate, passes the shock-baton. And the next node runs his

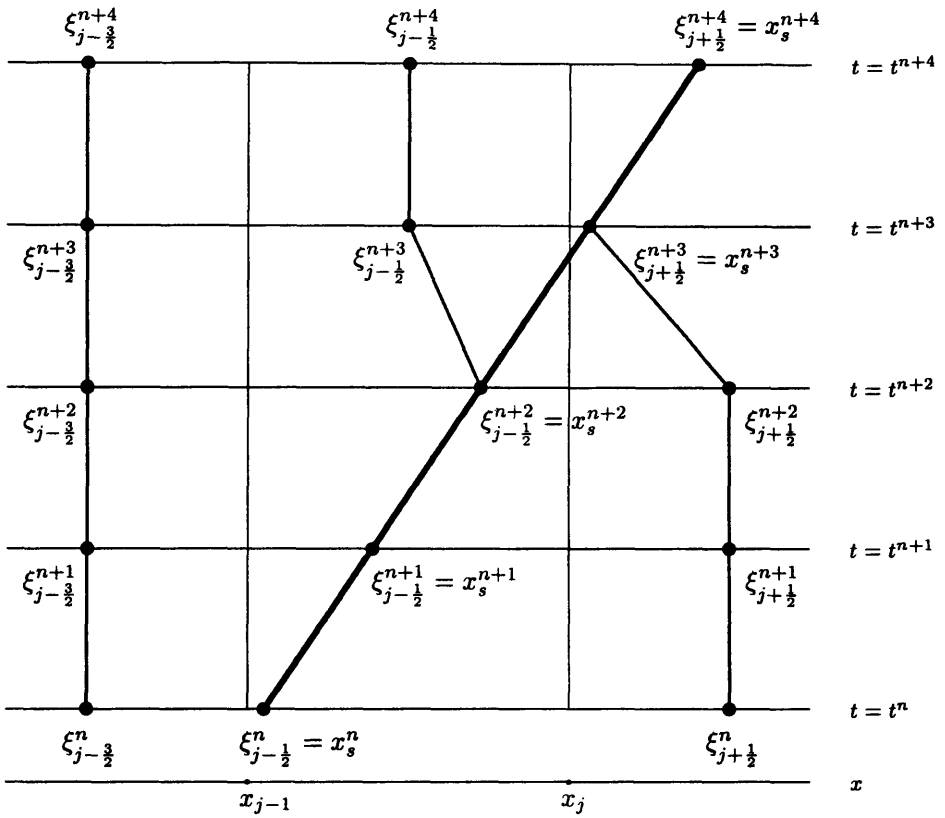

Figure 5.7: Single shock motion. Thin lines represent underlying fixed mesh and boxes. Filled circles represent nodes. Lines of intermediate thickness represent paths of interfaces. The thick line represents the trajectory of the shock.

portion of the distance, carrying the shock, and so on. This analogy makes clear the name of the first constraint.

So, at any time level, location of the shock coincides with one of the nodes — shock is not moving with respect to the grid. It is perfectly resolved all the time, thus no averaging across the shock occurs. Self adjusting grid method treats a single moving shock perfectly well.

5.2.4 Discrepancies of Self Adjusting Grid Method

Following [12] we implemented the method and conducted numerical simulation of the initial boundary value problem described in subsection 4.2.1 for the Euler equation. As we expected, the results indicated no sign of any spurious oscillations at all.

Unfortunately, when we applied the method to the two shock collision problem presented in subsection 4.2.4, the results turned out to be as bad as the outcomes of any of the Godunov-type schemes presented in section 4.2, which is not surprising in light of section 5.2.1 and Remark 5.7. The self adjusting grid method, meeting Requirement 2, falls short of satisfying requirements 1 and 3.The scheme stops to resolve a shock as a perfect discontinuity some time before the interaction and do not even attempt to resume it for some time after the interaction. It, certainly, leads to lots of readjustments and, as a result, spurious oscillations production.

Remark 5.9 Once the perfect resolution has been lost in the interaction, it takes the scheme quite a few steps to recapture the shock. During this readjustment process considerable amount of numerical noise is generated.
A contact, on the contrary, would be never completely recaptured. After the interaction a contact emerges as a few node wide discontinuity. The error, introduced into the contact during its formation, neither grows (which is better than ever spreading contacts on the fixed grid) nor shrinks (which left the contact endowed with a "staircase"-like structure) in time [5].
Therefore, after each interaction a contact would acquire more and more "steps", and if one would use the self adjusting grid method for a long time numerical computations involving shocks and contacts, the contacts would keep on spreading at time due to the interactions. Thus, although the algorithm considerably slows down the diffusion by confining it exclusively into the interaction/formation processes, the diffusion does not stop. The contact diffusion goes on as long as interactions take place.

The conclusions seem to be quite evident: While the constraints of the method are adequate for providing a perfect resolution of a single moving shock, they become too restrictive once two discontinuities appear close to each other. The method slips back to the averaging across discontinuities

[5]The differences in the shock and contact behaviors result from the fact that the former is a genuine nonlinearity while the latter is a linearly degenerate field. See the discussion in Remark 4.2.

— weighted average treatment of several waves emerging from one node (subsection 5.2.1) and two or more discontinuities moving in the same box (No-multi-node-box constraint) [6].

5.3 Discontinuity Treatment Scheme

We take from the self adjusting grid method the brilliant concept of the underlying fixed mesh, with relay and no-empty-box constraints, and discarded the rest. Because the design of the new scheme should be concentrated on the perfect discontinuity resolution, we will call it Discontinuity Treatment (DT) Scheme.

We need a scheme which would allow for the discontinuities to remain perfectly resolved up to the moment of the interaction (requirement 3 at the beginning of this chapter), and which would provide the perfect resolution of the interaction: The shocks and the contacts emerging from the interaction should be perfectly resolved from the beginning (requirement 1 at the beginning of this chapter). To satisfy the former requirement, we need to re-address the problem of two nodes coming too close to each other [7]. To meet the later requirement, we should resolve the node proliferation problem [8]. In the next two subsections we will present our solutions of these two problems.

5.3.1 Node Proliferation Problem and Significant Wave Approach

The node proliferation problem could be efficiently solved by using the **significant discontinuity** and **wave** notions.

Significant Discontinuity

There are two types of discontinuities in a discrete piecewise constant data (see Remark 4.1 and Figure 4.1):

- *Discontinuities introduced by the discretization.* They are fake, numerically artificial, from the physical point of view. The physical flow is continuous at the point of such jump. These

[6] Actually, the method performance in the case of two shocks moving towards each other is even worse than it was described in this subsection. There is one more constraint that we have not mentioned yet. This constraint forbids two nodes to come too close to each other: When the distance between two nodes, each in its own box, is smaller than half a box size, the scheme pushes the nodes back to keep them half a box size apart.
So, when two shocks race towards each other, they, first, stop half a box size apart. This brings to an end shocks perfect resolution and results in a "staircase" structure being developed — the first amount of noise being generated. Then, some of the small steps of the "staircase" start to merge into one node (which location is a weighted average of the steps position) as they begin to move in the same box. This leads to additional noise generation.
This complicated, spread out in time and space, interaction produces two new shocks and a contact, each of them having the finite, a few node, width (plus a significant amount of spurious numerical waves). The shock perfect resolution will be eventually recaptured by the scheme, though at the expense of new parasitic waves due to the width readjustment process, whereas the contact will remain a few nodes wide (see Remark 5.9).

[7] No-multi-node-box constraint, see subsection 5.2.2, and the half a box size constraint, see footnote 6, are used by the self adjusting grid method to handle this problem.

[8] The weighted average approach, see subsection 5.2.1, is used by the self adjusting grid method towards this problem.

discontinuities are insignificantly small, $O(\Delta\xi)$, from the computational perspective.

- *Discontinuities caused by the jumps in the piecewise continuous data.* They are genuine, as far as the physics of the inviscid flow is concerned, and are too large, $O(1)$, to be neglected in computations.

Only the latter discontinuity we will call **significant**.

Remark 5.10 Here we present a first order accurate method: The error of the method is $O(\Delta\xi)$. Therefore, we reconstruct the data up to the first order (which means, we leave it piecewise constant — see subsection 4.1.8). That explains the size of an insignificant discontinuity. If we were considering a second order accurate scheme, then we would reconstruct the data up to the second order (piecewise linear approximation, see subsection 4.1.8). Than the size of an artificial discontinuity would dwindle to $O(\Delta\xi^2)$, and so on. The size of a fake discontinuity reduces as the accuracy of the scheme grows. The insignificant jump shrinks proportionally to the mesh refinement — it is a numerically artificial, grid dependent quantity. However, the size of the significant discontinuity dose not change at all with the reduction of the cell size or switching to the higher accuracy scheme.

Significant Wave

Not all the waves emerging from a significant discontinuity represent the discontinuities of the real flow, shocks and contacts. For instance, at a single shock location only one of the numerical waves corresponds to the physical discontinuity, the shock, the rest being only numerical counterparts of sound and entropy waves.

However, generally, most of these waves appear in the numerical solution of the Riemann problem as weak numerical shocks and rarefaction waves rather than numerical sound and entropy waves. Those shocks and rarefactions, despite being weak, are of a finite strength. The finite strength of those artificially strong waves results from the same discretization error ($O(\Delta\xi)$ for the first order schemes, $O(\Delta\xi^2)$ for the second order schemes, etc.) which is responsible for appearance of the insignificant discontinuities. Actually, those waves are just time evolution of the discretization error initially concentrated at the cell interface and then propagating with all m-waves.

Therefore, similar to an insignificant discontinuity, the strength of such discretization born wave is $O(\Delta\xi)$ for the first order schemes, $O(\Delta\xi^2)$ for the second order schemes, and so on — the strength of these waves drops as either the resolution of the scheme or the refinement of the grid increases. On the other hand, the strength of the numerical wave representing the real flow discontinuity depends neither on the method accuracy nor on the grid cell size.

Thus, for a *significant* discontinuity, all the waves emerging out of it fall into two categories:

- *Insignificant Waves.* They are fake, numerically artificial, waves in the numerical solution of the Riemann problem, resulting from propagation of the discretization error initially concentrated

at the cell interface. They are weak waves. The strength of the wave is of the same order of magnitude as the error of the method and shrinks with increasing the accuracy of the scheme or refinement of the mesh. They correspond to continuous solutions of the physical flow.

- *Significant Waves.* The waves stemming from the numerical Riemann problem, corresponding to the discontinuous solutions of the real flow. They are strong waves , their strength being independent of the error of the scheme and the cell size of the grid.

Significant Waves Rule

In our new algorithm we forbid only the averaging across significant waves, restoring the rights of averaging across any other, insignificant, jumps. Therefore, the rule is: *Only significant waves should be followed by nodes and perfectly resolved.* Although it seems like a very natural and simple requirement, this rule accounts for a great improvement in the algorithm performance.

The rule of significant wave makes the new algorithm move only *few* nodes [9] while the self adjusting grid method moves *all* the nodes of the grid. On the other hand, when two or more genuine discontinuities stem from one node, the self adjusting grid method would not follow any of them but rather place *one* node somewhere in between, falling short of providing a perfect resolution for any discontinuity. In the same situation the new scheme would follow *all* the discontinuities with the equivalent number of nodes which enable the one-node representation of all the discontinuities.

The self adjusting grid method spends equal computational time for *all* the nodes. From the perspective of the method, all the nodes are the same. Which is true if a node is considered as a part of piecewise constant data — *crude* numerical approximation, without any "reconstruction" steps. The self adjusting grid method is satisfied with the results of the discretization — given piecewise constant data, the method "does not want" to do any extra, whatever small, work to recover some more information beyond the piecewise constant structure. The scheme works with the simplest and crudest numerical model, uses totally non-processed data — can not distinguish real *physical* phenomenon behind the numerical appearance of the piecewise constant flow.

In the new code we move one step further from the "crude" discrete data towards the physical flow. We recall that there exists the real flow concealed behind the discrete approximation, and *recover* some of its features from the piecewise constant functions. First, we check the size of each jump and classify nodes as significant and insignificant. Then, for each significant node, we look into the Riemann problem, and do the second classification, dividing the waves into significant and insignificant. By these two steps, we refine the crude piecewise constant model, recovering fine physical structure of the flow, separating the genuine discontinuities of the flow from the numerically artificial ones. By this **recovery** step we obtain a model which reflects more structure of the real

[9] In all real problems piecewise smooth flows are considered. Therefore, in any reasonable computational model the number of discontinuities is much smaller, of different order of magnitude, than the number of the nodes.

flow. This model is the next, after the crude piecewise constant model, level of approximating the real flow. It is still piecewise constant, but now we know locations and the local paths of the flow discontinuities.

Once we separated the significant waves from the rest, we should give an extra treatment only to the significant nodes/waves. The more important is the node — the more significant waves stem from it the better treatment it receives — each wave emerging from the node receives one new node. This strategy is simple — we concentrate on few crucial points which can make big difference in accuracy/noise generation, giving them as much care as we can, and we completely neglect in our treatment the rest of the nodes, do not move them at all. We do not want to move the insignificant nodes since, first, our model of the dynamics of those nodes is unreliable — its error is of the of the same or greater order of magnitude as the error of the method (see discussions in the paragraphs above) and, second, even if we had a trustworthy approximation of the motion of those nodes (for instance, by using a higher order scheme), their careful treatment would result only in the next orders of the accuracy/noise generation improvements [10].

The new scheme spends some time on the research, obtains the full investment information, and than invests only into what has been found to yield a big interest immediately — moves a node with each significant wave at each significant node, and spends nothing on what has become known not to return any profit — does not move an insignificant node at all.

The self adjusting grid method strategy is completely opposite. Because it does not do a recovery step — do not conduct a research, it can not separate the wheat from the chaff, and, with no investment information at all, it invests uniformly everywhere, does not matter whether it is profitable or wasteful: The scheme will generate exactly one node and always move it, regardless of the jump being fake or real and the number of the real discontinuous solutions emerging from the real jump.

So, the new scheme solves the problem of grid movement using two main concepts: The first is *research/data recovery* concept. We obtain the data which help us divide all the nodes into two unequal size classes: a very small one, treatment of which members can make a big difference; and a very large one, treatment of its members will produce almost no response at all. The second concept is *spending most of the effort/time on few influential things, almost neglecting the insignificant ones.*

It seems evident from the discussion above, that the approach of the new scheme gives much better accuracy — actually, it provides the perfect resolution of all significant/physical discontinuities emerging from the node, no adjustment processes, no noise production — spending considerably less computational time on the moving the nodes.

[10] Actually, it would lead us to the next level of approximation/refinement. We could introduce the next level of significance for the nodes/waves corresponding to the weak discontinuities of the real flow — jumps in the first derivatives (the edges of the rarefaction waves) — and give them special treatment as well (place nodes on their paths). It would, certainly, lead to the improvement in the accuracy/noise generation of the scheme, but this improvement would be only of the next order of magnitude, comparing with the effects resulting from the strong discontinuity special treatment.

Remark 5.11 The recovery concept is, in its spirit, similar to the reconstruction of the piecewise linear polynomials from the piecewise constant data (subsection 4.1.8). Both the linear reconstruction and the discontinuity structure recovery use the same, piecewise constant, "crude" model. However, the different physical features are extracted.

The linear reconstruction is concentrated on the enriching the knowledge of the physical flow in the smooth regions. The refined model after the reconstruction process is the piecewise linear approximation. It yields better representation of the smooth regions of the flow — reduces the error in the smooth regions — but still does not include any knowledge about the physical discontinuities structure — the refined model is as bad as the piecewise constant model with respect to the discontinuity resolution.

On the contrary, the discontinuity structure recovery is zeroed in on the collecting the knowledge about the discontinuity structure of the physical flow. The refined model after the recovery procedure is the piecewise constant approximation with the nodes endowed with the corresponding discontinuity structure. It enables perfect discontinuity treatment, diminishing the parasitic noise production. However, in the smooth regions of the flow, this refined model performs as bad the "crude" piecewise constant model.

From this argument, it is evident, that to obtain decent performance in both the smooth regions and at the discontinuities of the flow, we need to carry on both reconstruction processes: the linear reconstruction and the discontinuity structure recovery. It is to be done in subsection 5.3.6.

Remark 5.12 It should be noticed that both the discontinuity structure recovery step and the moving of the nodes procedure take very little computational time. The only noticeable time is to be spend on the first step of the recovery, when the strength of all nodes is checked (see Remark 5.13). After that, the second step of recovery, when the strength of the waves from the significant nodes is checked, and the moving procedure are performed only on a small set of the nodes, so it almost would not affect the computational time, when long time simulations are carried for the real problems (see footnote 9). Besides, all the data needed for the second recovery step and the moving node step — the wave strengths and speeds — are part of the Riemann problem solution, and the Riemann problem is to be solved anyways for other purpose — to obtain numerical fluxes (see 5.2).

Remark 5.13 Nowadays, nobody uses first order schemes (We use the first order scheme in this chapter only as an example — for the sake of clarity. It will be extended to the second order in subsection 5.3.6 to be used for long time simulations in chapter 6). However, all high resolution schemes, during the reconstruction step, do the jump strength check at each interface anyways — to choose the stencil or limiters (see subsection 4.1.8). So, for high resolution schemes, the first step of the recovery does not take any extra time as well. Therefore, for a high resolution method (to be presented later and used in the following chapter), nether recovery no moving node steps slow down

the algorithm noticeably.

Significant Discontinuities/Waves Criterion

So far, the procedure of selecting the significant waves looked rather abstract. In this paragraph we would like to tell a few words concerning the implementation aspects of the problem.

As it follows from the discussion above, quite natural thing would be to compare the jump size/wave strength against some threshold. Several physical variables can be used as an indicator of the strength of the jump/wave (see, e.g., [12]). We, first, tried to use an entropy for this purpose. The good thing about the entropy is that it experience a jump across both the shocks and the contacts. Unfortunately, in real flow problem, those jumps sometimes appear to be too small, hard to be detected, even for fairly large shocks. Therefore, we switched to the following tactics: We used the pressure as the indicator of the shock strength and the density as the indication of the contact strength. This approach worked remarkably well.

One always can come up with some reasonable and natural thresholds, such that when the jump in an interface is larger than the threshold, then the discontinuity will be treated as significant. Let us, for instance, consider the pressure threshold, serving to locate a significant shock wave. Then it seems proper to choose the threshold in the following form:

$$[p] = \theta \, \frac{\bar{\partial} p}{\partial x} \Delta\xi, \tag{5.32}$$

where $\frac{\bar{\partial} p}{\partial x}$ is a typical gradient of the pressure in the smooth regions of the problem, and θ is some safety coefficient, $\theta > 1$. Since the typical size of the numerically artificial jump in a smooth region is

$$\Delta p = \frac{\bar{\partial} p}{\partial x} \Delta\xi, \tag{5.33}$$

(5.32) reduces to

$$[p] = \theta \Delta p. \tag{5.34}$$

(5.34) implies, that if θ is reasonably large, then the threshold (5.32) will successfully separate fake discontinuities from the physical shocks.

Remark 5.14 It should be emphasized, that threshold (5.32) works well as long as the strength of

the shocks in the problem is of different order of magnitude than the product of the typical gradient, $\frac{\bar{\partial p}}{\partial x}$, and the cell size, $\Delta\xi$, of the *particular* grid being used.

$$strength \gg \frac{\bar{\partial p}}{\partial x}\Delta\xi \tag{5.35}$$

So, before we start a computational experiment, given the minimal strength of a shock we are interested in, we should estimate the size of the gradients which could appear in the problem and choose an appropriate mesh spacing such that (5.35) will hold during the entire numerical simulation. If at some point of the experiment condition (5.35) is violated, the gradient becomes too big (our estimate turns out unreliable), we should increase the local refinement of the grid so that condition (5.35) holds again. This local in time and space grid refinement could be achieved through multi-grid approach outlined in subsubsection 5.3.6. The DRT scheme, which would be described in the same subsubsection, automatically locates gradients which are too large and then put an appropriate fine grid around the spot.

The new scheme, due to the recovery/selection mechanism and follow-all-significant-waves/physical-discontinuities procedure ensures that all the shocks and the contacts emerging from the interaction will be perfectly resolved from the beginning. Requirement 1 at the beginning of this chapter is satisfied. To complete the new scheme development, we have yet to address requirement 3. In the next subsection we present the approach which guarantees that the discontinuities remain perfectly resolved up to the moment of the interaction.

5.3.2 Two Nodes Coming Close to Each Other

To secure that the discontinuities remain one node wide all the time up to the moment of the collision, we need to re-address the procedure of choosing the time step (subsection 5.1.2).

After the node selection is done (see subsection 5.3.1), the nodes corresponding to the physical jumps are distinguished from the nodes with fake jumps. Based on this classification, all the cells can be divided into two groups.

- *Significant cells.* Both interfaces of a significant cell are significant nodes.
- *Insignificant cells.* At least one of the interfaces of an insignificant cell is an insignificant nodes.

Insignificant Cell

To an insignificant cell, the CFL-type constraints developed in subsection 5.1.2 are applied.

There are two different cases:

- both nodes of an insignificant cell are insignificant

- one of the nodes of an insignificant cell is significant

When both nodes are insignificant, neither of them moves,

$$\xi_{j-\frac{1}{2}}^{n+1} = \xi_{j-\frac{1}{2}}^{n}, \qquad \xi_{j+\frac{1}{2}}^{n+1} = \xi_{j+\frac{1}{2}}^{n}, \tag{5.36}$$

Both interfaces are contained inside the domains of dependence of their own Riemann problems. Thus, for either node, the CFL-type constraint has form of inequality (5.22). Therefore, the CFL-type condition for such cell reduces to two inequalities:

$$\xi_{j-\frac{1}{2}}^{n} + \lambda_{j-\frac{1}{2}}^{max}\,\Delta t \;<\; \xi_{j+\frac{1}{2}}^{n} \qquad \text{and} \qquad \xi_{j+\frac{1}{2}}^{n} + \lambda_{j+\frac{1}{2}}^{min}\,\Delta t \;>\; \xi_{j-\frac{1}{2}}^{n} \tag{5.37}$$

If one of the nodes is significant, then one or more significant waves arise out of it. All these waves must be perfectly resolved (see subsection 5.3.1), therefore a node would be placed at the location of each significant wave at the next time level.

To be specific, let us consider, for example, the case when the left node is a significant one, and the right node is an insignificant one (the other case can be treated the same way). Then there is one or more significant waves emanating from the left node. Let the rightmost wave stemming out of the left node is a l-wave, $1 \le l \le m$. Depending on the time step size and the speeds of the waves, three different situations can occur.

1. The l-th wave lands inside the $(j-1)$-th box at the advanced time level $t = t^{n+1}$ (Figure 5.8(a)).
2. The l-th wave stays in the j-th box (Figure 5.8(b)).
3. The l-th wave ends up inside the $(j+1)$-th box at the next time level (Figure 5.8(c)).

We should emphasize that at this stage, when the time step size is being chosen, we do not know its final value, so we do not know which of these three situations will really come into existence. If the velocity of the l-th wave is negative, then either the first or the second case can happen. On the other hand, if the velocity of the l-th wave is positive, then either the second or the third situation may occur [11].

Let now pretend, for a moment, that we know which of the situations will finally occur. Then our strategy in choosing the time step would be, depending on the case, as follows:

1. (Figure 5.8(a)) There is no discontinuity inside the j-th box, and according to the no-empty-box constraint (subsection 5.2.2) we should place a new node in the middle of the box:

$$\xi_{j-\frac{1}{2}}^{n+1} = \frac{1}{2}(x_{j-1} + x_j) \tag{5.38}$$

[11] Regardless of the direction of the l-th wave velocity, the second situation takes place if the final time step is small enough.

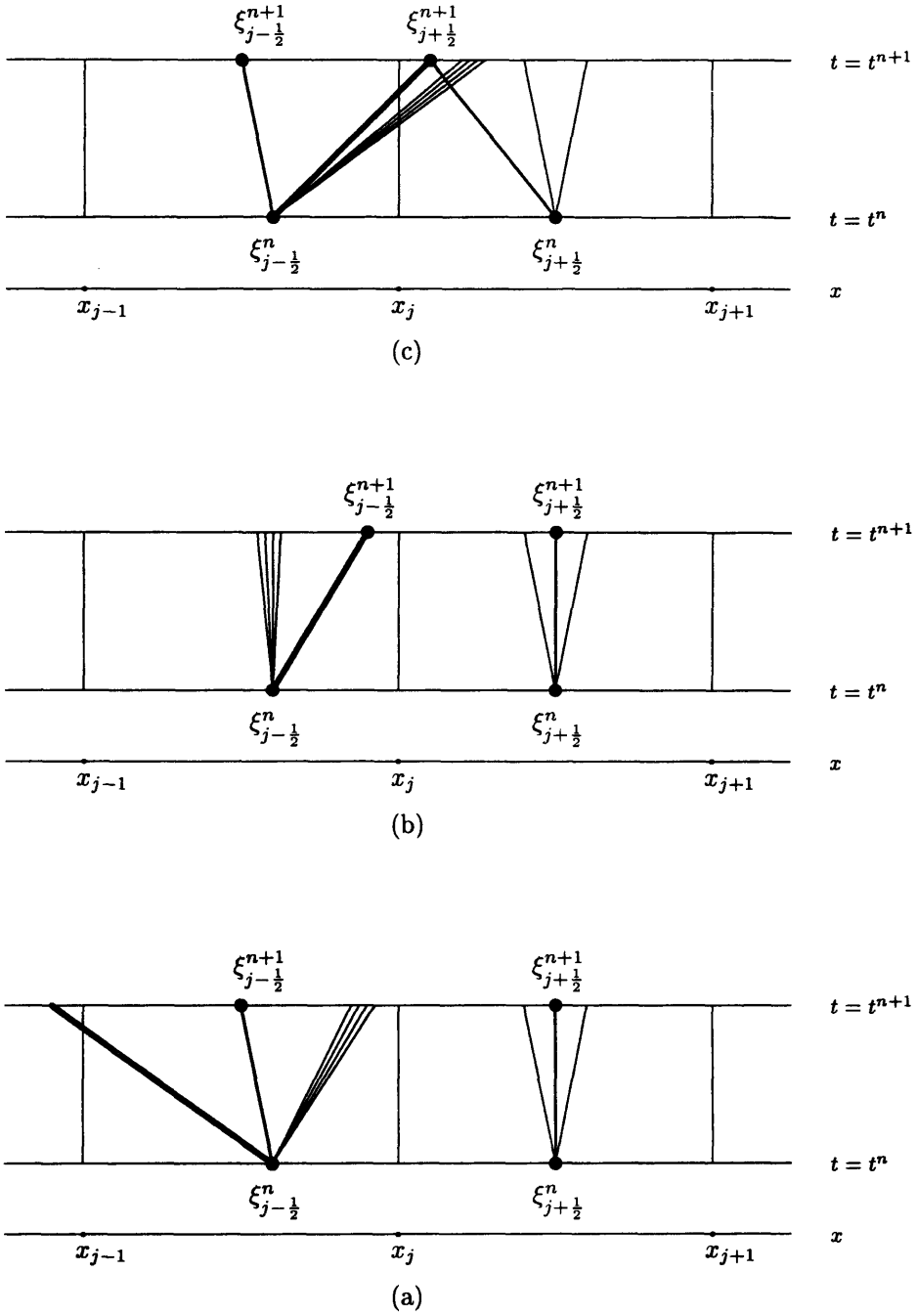

Figure 5.8: Half-significant cell. Thin lines represent underlying fixed mesh, boxes, and insignificant wave paths. Filled circles represent nodes. Lines of intermediate thickness represent paths of interfaces. Thick lines represent trajectories of significant waves.

The $(j+1)$-th node does not move:

$$\xi_{j+\frac{1}{2}}^{n+1} = \xi_{j+\frac{1}{2}}^{n} \tag{5.39}$$

Both interfaces belong to the domains of dependence of their own Riemann problems. Thus, for either node, the CFL-type constraint has form of inequality (5.22). Therefore, for this cell, the CFL-type condition transforms into two inequalities:

$$\xi_{j-\frac{1}{2}}^{n} + \lambda_{j-\frac{1}{2}}^{max}\,\Delta t \;<\; \xi_{j+\frac{1}{2}}^{n} \qquad \text{and} \qquad \xi_{j+\frac{1}{2}}^{n} + \lambda_{j+\frac{1}{2}}^{min}\,\Delta t \;>\; \frac{1}{2}(x_{j-1} + x_j) \tag{5.40}$$

2. (Figure 5.8(b)) At least the rightmost significant wave remains inside the j-th box — the left node moves with the velocity of the l-th wave, $s_{j-\frac{1}{2},\,l}^{n}$:

$$\xi_{j-\frac{1}{2}}^{n+1} = \xi_{j-\frac{1}{2}}^{n} + s_{j-\frac{1}{2},\,l}^{n}\,\Delta t \tag{5.41}$$

The right node does not change its location:

$$\xi_{j+\frac{1}{2}}^{n+1} = \xi_{j+\frac{1}{2}}^{n} \tag{5.42}$$

This case is very similar to the previous one (again inequality (5.22) should be used) and the CFL-type constraint for the cell reduces to:

$$\xi_{j-\frac{1}{2}}^{n} + \lambda_{j-\frac{1}{2}}^{max}\,\Delta t \;<\; \xi_{j+\frac{1}{2}}^{n} \qquad \text{and} \qquad \xi_{j+\frac{1}{2}}^{n} + \lambda_{j+\frac{1}{2}}^{min}\,\Delta t \;>\; \xi_{j-\frac{1}{2}}^{n} + s_{j-\frac{1}{2},\,l}^{n}\,\Delta t \tag{5.43}$$

3. (Figure 5.8(c)) At least the rightmost significant wave moves into the $(j+1)$-th box. Due to the relay constraint (subsection 5.2.2) we should place a child of the $(j+1)$-th node at its location. If all the waves leave the j-th box (Figure 5.8(c)), then children of the $(j+1)$-th node should be placed at their location (again the relay constraint is to blame), and the only child of the j-th node ought to be situated in the center of the j-th box (no-empty-box-constraint). If some of the waves abandon the j-th box and some stay, then children of the $(j+1)$-th node should be placed at the positions of the leaving waves, and children of the j-th node ought to be located at the positions of the staying waves. But this situation does not differ much from the case of Figure 5.8(c) as long as the flux calculation is concerned: In any situation, the fluxes for all the nodes of both boxes, the j-th and the $(j+1)$-th, should be computed using the data of only *one* Riemann problem, the Riemann problem of the j-th interface (This is the case when the interface is contained inside the domain of dependence of a neighboring Riemann problem —

see equation (5.24) and the text around it). The domain of influence of the $(j+1)$-th Riemann problem contains children neither of the j-th node nor of the $(j+1)$-th node. Therefore, the only CFL-type restriction in this case should be imposed by the requirement that no wave from the $(j+1)$-th Riemann problem reaches the rightmost child of the $(j+1)$-th node — the location of the l-wave from the j-th node. It is the same restriction as the second constraint in (5.43) (cf.(5.24)):

$$\xi^n_{j+\frac{1}{2}} + \lambda^{min}_{j+\frac{1}{2}}\,\Delta t \;>\; \xi^n_{j-\frac{1}{2}} + s^n_{j-\frac{1}{2},\,l}\,\Delta t \tag{5.44}$$

Unfortunately, as we already pointed out, in reality, we do not know which of three situations described above will finally occur (because the final time step Δt^n is unknown at that stage). Thus, to be on the safe side, we need to choose the strongest constraints. As it easy to see, all the restrictions above hold if the following constraints are met:

$$\xi^n_{j-\frac{1}{2}} + \lambda^{max}_{j-\frac{1}{2}}\,\Delta t \;<\; \xi^n_{j+\frac{1}{2}} \qquad \text{and}$$
$$\xi^n_{j+\frac{1}{2}} + \lambda^{min}_{j+\frac{1}{2}}\,\Delta t \;>\; max\Big(\;\frac{1}{2}(x_{j-1}+x_j),\;\; \xi^n_{j-\frac{1}{2}} + s^n_{j-\frac{1}{2},\,l}\,\Delta t\;\Big) \tag{5.45}$$

Summarizing, the CFL-type condition requires that the time step should be chosen small enough to assure that inequality (5.37) holds for all both-node-insignificant cells, inequality (5.45) holds for all insignificant cells with the left node being significant, and the analog of inequality (5.45) holds for all insignificant cells with the right significant node. After such time step is found, the last check need to be done — we should examine whether the CFL-type condition is satisfied for the significant cells (if such cells exist at the current time level — those cells occur extremely infrequently in the course of simulation: only for a short time immediately before and after a collision).

Significant Cell

For insignificant cells we used estimates, lower and upper bounds of the velocities of the waves emerging in the Riemann problems, $\lambda^{min}_{j-\frac{1}{2}}$ and $\lambda^{max}_{j-\frac{1}{2}}$, to restrict the step size. As a result, the CFL-type constraints for insignificant cells generate time steps with some safety margins — if we increased such time step a little bit, the CFL condition written out at the beginning of the subsection 5.1.2 would still be satisfied — each interface would be still contained inside the domain of influence of not more than one Riemann problem.

We change this approach when the time step should be chosen from a significant cell consideration. Both interfaces of a significant cell are significant nodes. If the rightmost wave emerging from

the left node and the leftmost wave emanating from the right node move towards each other, then the *exact* [12] time left to the collision is computed. If this time is larger than the time step computed by this moment, then the time step check similar to the insignificant cell time step check should be performed. If it is less than the current time step, the time left to the collision should be chosen as a new current time step.

If the final time step happened to be chosen from a significant cell restriction — the final time step is the time left to collision of the waves stemmed from the left and right interfaces of the significant cell, then the waves remain perfectly resolved up to the collision moment. At the collision moment, time $t = t^{n+1}$, the waves coalesce at one node (as in the real physical flow). At time level $t = t^{n+1}$ at the node where the waves have coalesced, we have a Riemann problem, which, *within the accuracy of the method*, is located at the same position as the real physical discontinuity and has the jumps equal to those of the real physical discontinuity. It should be emphasized, that the whole interaction is concentrated precisely at one point on the space-time plane. Therefore, our objective, to keep the perfect resolution of discontinuities up to the moment of their collision — requirement 3 at the beginning of this chapter — is achieved [13].

Now we look at some details of the time step selection based upon the significant cell CFL-type constraints. Let the j-th cell be a significant cell. It means that its both interfaces are significant nodes. Suppose that the rightmost significant wave stemming from the left node is the l-th wave, moving with the velocity of $s^n_{j-\frac{1}{2},\,l}$, and the leftmost significant wave arising from the right node is the r-th wave, moving at the speed of $s^n_{j+\frac{1}{2},\,r}$. If $s^n_{j-\frac{1}{2},\,l} > s^n_{j+\frac{1}{2},\,r}$ then the discontinuities will eventually collide. The time left to the collision can be easily computed:

$$\xi^n_{j-\frac{1}{2}} + s^n_{j-\frac{1}{2},\,l}\,\Delta t \;=\; \xi^n_{j+\frac{1}{2}} + s^n_{j+\frac{1}{2},\,r}\,\Delta t \tag{5.46}$$

and, therefore,

$$\Delta t \;=\; \frac{\xi^n_{j+\frac{1}{2}} - \xi^n_{j-\frac{1}{2}}}{s^n_{j-\frac{1}{2},\,l} - s^n_{j+\frac{1}{2},\,r}} \tag{5.47}$$

If the time step computed from (5.47) is less than the current time step, then the newly computed

[12] Exact within the accuracy of the method. There are two types of errors in the calculation of the path of a discontinuity: (1) A first order accurate scheme calculates the speed of the discontinuity with an error of $O(\Delta_\xi)$. Therefore, the path can not be computed with an accuracy exceeding $O((\Delta_\xi)^2)$ (see subsection 5.3.4 and formula (5.53)). (2) If, for example, a right shock emerging from the left node hits a contact stemming from the right node, the shock, in general, first interacts with the left numerical shock/rarefaction wave generated by the right Riemann problem. This left numerical wave is a ghost of discretization, and while the shock strength is $O(1)$, the strength of the left wave is $O(\Delta_\xi)$. As a result of the interaction, the shock position, and, therefore, the time of the collision will be computed with $O((\Delta_\xi)^2)$ error, which, again, is within the error of the scheme.

[13] After the discontinuities have coalesced into one discontinuity, which exactly, within the accuracy of the method, equivalent to the discontinuity in the physical flow, at the next time level t^{n+1} this discontinuity will be selected as significant and resolved according to the follow-all-significant waves procedure (see subsection 5.3.1) — it will generate new discontinuities which will be one-node wide from the beginning. The whole interaction is perfectly resolved.

time step should be chosen as a new current time step. If the final time step is the time step computed from (5.47), then for the j-th cell both waves emerging from the j-th and $(j+1)$-th nodes coalesce into one point at time $t = t^{n+1}$. To give a perfect resolution to the interaction, we should place a new node at this point. This new node can not be connected with both, the left and the right, old nodes. If a node has two parents, it would violate the numerical conservation — the worst crime one can commit devising a numerical scheme for a system of conservation laws. Therefore, we have two options. Either to connect the new node with the left or with the right old node.

In case of the Euler equation, we use the choice to enable the fulfillment of the CFL condition written out at the beginning of this chapter. There are three generic types of collision for the Euler equation [4]. Depending on the type of interaction, different choice of the moving interface should be made.

1. *A head-on collision of two shocks.* Shocks are of different families: a right shock collides with a left shock. Either connection does not violate the CFL condition, because both waves are the outer waves of the Riemann problem. If, for example, we choose to connect the right node with the new node, the right shock emerging from the left node is a rightmost wave of the left Riemann problem, and this wave touches the segment connecting the new node and the right node only at its end point at time t^{n+1}. Therefore no wave arising from the left node reaches the segment over time interval (t^n, t^{n+1}). If we decide to link the new node with the left node, then, since the left shock is the leftmost wave of the right Riemann problem, the similar to the above argument proves that the CFL condition holds.

2. *The overtaking of shock waves.* Both shocks belong to the same family. If, for instance, a right shock overtakes another right shock, then the right node should be joined with the new node. Because the right shock of the left node is the rightmost wave of the left Riemann problem, and since it touches the segment linking the right node and the new node only at its endpoint at time $t = t^{n+1}$, no wave emerging from the left Riemann problem can reach the segment over time interval (t^n, t^{n+1}). The CFL condition is met. On the other hand, if we connected the left node with the new node, then, because the right shock is the rightmost wave of the right Riemann problem, and it would touch the segment linking the left and the new nodes at $t = t^{n+1}$, the other two waves of the right Riemann problem, which move faster to the left than the right shock, would hit the segment at some earlier times from interval (t^n, t^{n+1}). It would violate the CFL condition. The same argument shows that when a left shock overtakes another left shock, the left node should be joined with the new node at the coalescence point.

3. *Reflection and refraction of shock waves on contact discontinuities.* Because a contact is an inner wave, and a shock is an outer wave of a Riemann problem, when they collide, no other wave from the Riemann problem which produced a shock may reach the contact before the

moment of the collision. However, the wave of the opposite to the shock family from the Riemann problem associated with the contact will definitely hit the shock before it collides with the contact. Hence, we should connect the new node with the old node associated with the contact to satisfy the CFL condition.

Remark 5.15 Actually, if in any of the last two cases we chose a connection which violates the CFL-type condition, we introduce an error of the order of the error of the scheme. Suppose, for instance, in the last case we chose to connect the new node with the old node associated with the shock rather than with the contact. This new interface belongs to the domain of influence of both, the left and the right, Riemann problems. However, while the shock represents a significant, $O(1)$, jump; the wave, originated from the Riemann problem of the contact which hits the interface before the contact does, is just a small ghost of discretization, $O(\Delta\xi)$. Hence, if we do not consider its influence, using only the data of the Riemann problem from which the shock arises in flux computations, we will introduce an error of $O(\Delta\xi)$ which is exactly the error with which the fluxes are being computed. As we just saw, the choice of better node connection is kind of excessive step. If it is not done, the scheme won't lose its accuracy or stability. However, although the improvement goes beyond the scheme accuracy; we believe that it helps enable somewhat better interaction representation (see also Remark 5.5).

Remark 5.16 It should be noticed that if one wants to use the choice of better node connection, the relay constraint could become too stringent, so one should not use it with a significant cell. If the final time step is the time step computed from (5.47), then for the j-th cell either of two generic situations depicted in Figure 5.9 can happen. Either the cell end nodes belong to the same box (a) or to the different neighboring boxes (b). In both situations, the waves emerging from the j-th and $(j+1)$-th nodes coalesce into one point at time $t = t^{n+1}$. As it easy to see in the second case (Figure 5.9(b)), the relay constraint leaves no choice but to connect the new node with the left discontinuity, regardless of the type of the interaction and the type of the left wave.

Remark 5.17 In the case of a general system of conservation laws with $m > 3$, the choice of better node connection usually does not work, and one should content oneself with the possible higher order disturbances from the neighboring Riemann problems being omitted in the flux computations. As it was argued above (Remark 5.15), this neglect should not cause the scheme to lose either accuracy or stability.

If the time left to the collision is greater than the current time step, then the waves won't collide at time interval (t^n, t^{n+1}). They won't coalesce into one node, and to be perfectly resolved either should be followed by a node — a new node should be placed at the location of either wave at time t^{n+1}. Therefore, we will need to compute fluxes through both interfaces of the j-th cell. To be

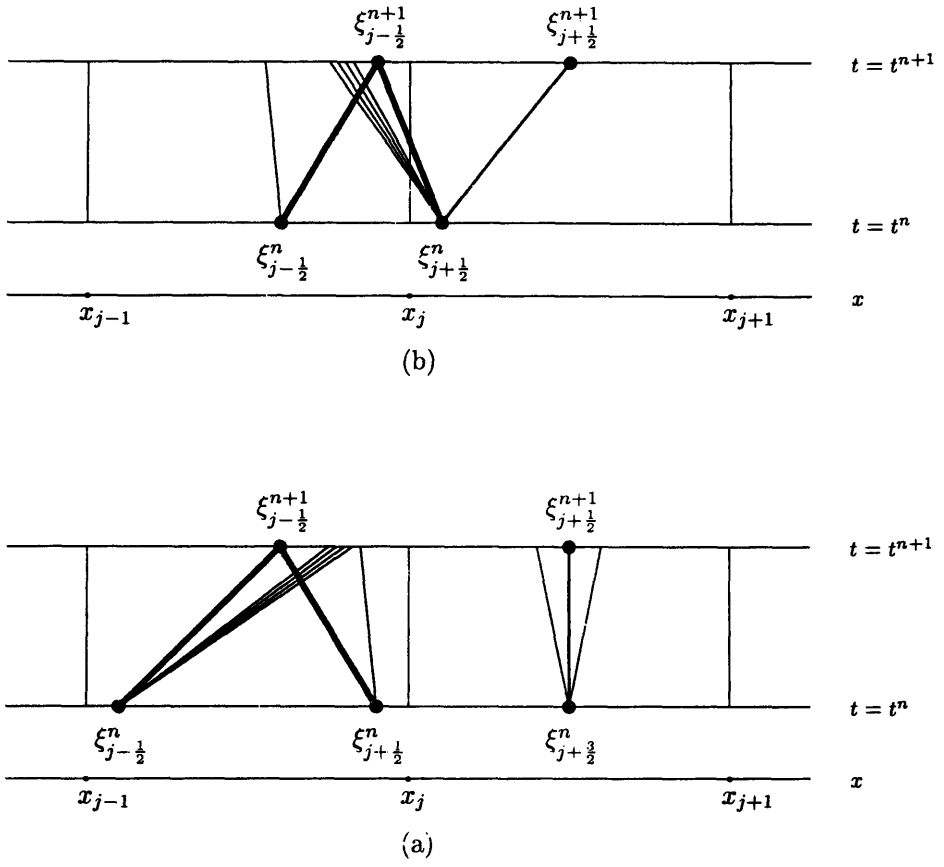

Figure 5.9: Significant cell. Thin lines represent underlying fixed mesh, boxes, and insignificant wave paths. Filled circles represent nodes. Lines of intermediate thickness represent paths of interfaces. Thick lines represent trajectories of significant waves.

able to do it, we ought to guarantee that either interface is not contained inside of the domains of dependence of more than one Riemann problems — the CFL condition written out at the beginning of subsection 5.1.2. Again, we have two alternatives. Either keep on applying relay constraint for a significant cell or drop it. If we stick with the relay constraint, we will introduce some error in flux computations, however it will be within the bounds of the error of the scheme (see Remark 5.5). If we drop the constraint, we could obtain somewhat more accurate results. In the latter case, either interface is contained inside the domain of dependence of its own Riemann problem. The constraint (5.22) should be used. Because

$$\xi^{n+1}_{j-\frac{1}{2}} = \xi^{n}_{j-\frac{1}{2}} + s^{n}_{j-\frac{1}{2},\,l}\,\Delta t \qquad \text{and} \qquad \xi^{n+1}_{j+\frac{1}{2}} = \xi^{n}_{j+\frac{1}{2}} + s^{n}_{j+\frac{1}{2},\,r}\,\Delta t, \tag{5.48}$$

The CFL-type condition for insignificant cell transforms into two inequalities:

$$\xi^{n}_{j-\frac{1}{2}} + \lambda^{max}_{j-\frac{1}{2}}\,\Delta t \;<\; \xi^{n}_{j+\frac{1}{2}} + s^{n}_{j+\frac{1}{2},\,r}\,\Delta t \tag{5.49}$$

and

$$\xi^{n}_{j+\frac{1}{2}} + \lambda^{min}_{j+\frac{1}{2}}\,\Delta t \;>\; \xi^{n}_{j-\frac{1}{2}} + s^{n}_{j-\frac{1}{2},\,l}\,\Delta t. \tag{5.50}$$

Hence, if the time left to the collision is greater than the current time step, the current time step should be checked against the time step restriction of the CFL-type condition (5.49).

Remark 5.18 It should be pointed out that the same approach of the profitable (computational) time investment into few "things" as we used to solve the node proliferation problem (see subsection 5.3.1), is employed again. At this time, it helps resolve the difficulties arising when two nodes are coming close to each other.

(1) Overwhelming majority of the cells are both-node-insignificant cells. Those cells do not play any crucial role, and a simple time step estimate is used for them (5.37).

(2) Insignificant cells with one significant node correspond to a single discontinuity propagation problem. In each time level the number of those cells is much smaller than the number of both-node-insignificant cells. However, they are more important then the both-node-insignificant cells so we use more complicated time step estimate for them (5.45).

(3) The most uncommon cells to meet are significant cells. They appear only at few places at few time levels. However, they are the most important cells as far the accuracy is concerned, because they control discontinuity interaction process. Therefore the biggest time investment is made into those cells — a series of "if" checks as described in this paragraph.

Notwithstanding all those checks, the performance of the algorithm does not undergo any noticeable slow down, because all the complicated checks occur only on few cells, the rest of the cells being regular non-moving cells as the cells in a fixed mesh Godunov's scheme.

Remark 5.19 At first sight, there seems to be a disadvantage in allowing two nodes to come close to each other. When it happens, the CFL-type conditions force the scheme to choose a small time step Δt^n. It seems to slow down the performance of the algorithm.
However, if we look carefully at this situation, we could see that the reduction of the time step size actually happens only for a very short period of time — immediately before and after the interaction. Since in any reasonable real problem interactions occur infrequently, at very few time levels, most of the time the scheme marches in time with a regular size time step.

5.3.3 New Scheme. Overview

The new algorithm can be presented as a sequence of steps:

(i) RECONSTRUCTION.

- a RECONSTRUCTION PROPER. Since the scheme presented here is first order method, the piecewise constant data representation is adequate for the accuracy of the method.
- b SIGNIFICANT NODES RECOVERY. The nodes with the jumps exceeding the threshold are marked as significant.

(ii) FLUX CALCULATION.

- a RIEMANN PROBLEM SOLUTION. At each cell interface solve the Riemann Problem exactly. It gives us a local solution at the interface up to the first order of accuracy.
- b SIGNIFICANT WAVES RECOVERY. Each wave of each Riemann problem with the strength surpassing the threshold is marked as significant.
- c TIME STEP GENERATION. Start with some initial time step. For each cell check it against the CFL-type constraint of the cell. Depending on the type of the cell — significant, half-significant, insignificant — different CFL-type constraints are used. If the constraint is violated by the current time step, reduce the time step to meet the constraint.
- d NEW GRID CONSTRUCTION. Using data from the solution of the corresponding Riemann problem, compute the position of each significant wave at the next time level t^{n+1}. The path of a significant wave is computed up to the second order of accuracy — see (5.53). At the location of each significant wave at the advanced time level t^{n+1} place a node of a new grid. Check all the boxes. If a box is without a node, place a node of a new grid at the center of the box.
- e FLUX CALCULATION PROPER. Use the solution of the Riemann Problem and the newly contracted grid to compute the numerical flux (again, the flux must be computed up to the first order of accuracy).

(iii) AVERAGES UPDATE. Average the results in a *conservative* fashion – at each cell add the difference of the numerical fluxes at the cell interfaces to the cell averages.

Remark 5.20 As in the case of Godunov's methods on fixed grid with variable time step, the scheme can be implemented with only two loops over cells for each step in time. The first loop should be done to determine the time step size — it includes all steps up to step **(ii)**–c. On the second loop the new grid is generated, the fluxes are computed, and the new cell averages are obtained — steps **(ii)**–d – **(iii)** of the scheme.

5.3.4 New Scheme. Accuracy

We consider numerical schemes in conservation form (4.4) that are derived from approximating system of conservation laws (2.8). As we noticed earlier (see footnote 2 on page 24), in numerical methods for conservation laws it is preferable to view u_j^n as an approximation to a cell average of $u(x,t^n)$, $\bar{u}_j^n$ – see (4.3), rather than as an approximation to a pointwise value of $u(x,t^n)$ in the center of the cell, $u(\frac{1}{2}(\xi_{j-\frac{1}{2}}^n + \xi_{j+\frac{1}{2}}^n),\ t^n)$, as it is used in linear finite difference methods. Hence, it is only natural to consider truncation error in the sense of cell averages as well — error in L^1-norm. That is, to find a truncation error of a scheme, we should replace the approximate solution u_j^n in the difference equation by the *average* of the true solution, $\bar{u}_j^n$, rather than by the mid-cell pointwise value of the true solution. How well exact averages satisfy the difference equation gives an indication of how well the exact solution of the difference equation satisfies the differential equation (2.8).

For a both-nodes-insignificant cell — neither node moves, the scheme reduces to a fixed grid Godunov's scheme. Therefore, when the accuracy analysis is applied to the new scheme for such a cell, a first order truncation error will be found.

On the other hand, if at least one of the nodes of the cell moves, we will get a different result. Consider, for example, the scheme on the cell shown in Figure 5.10. The left node of the cell has moved. It means that the left node is placed at the locations of some discontinuity. However, in the numerical method a discontinuity location is computed with some numerical error. Therefore, the left end node coincides with the path of the discontinuity at time $t = t^{n+1}$ not exactly, but up to this error. (In Figure 5.10, computed discontinuity location, the left node at time $t = t^{n+1}$, is indicated by D, while the exact discontinuity position is marked by S).

We can approximate path of a significant discontinuity emanating from a node $\xi_{j-\frac{1}{2}}^n$ at time t^n, with any order of accuracy, by a Taylor series expansion about a point $(\xi_{j-\frac{1}{2}}^n, t^n)$:

$$x_s(t) = \xi_{j-\frac{1}{2}}^n + \dot{x}_s(t^n)(t-t^n) + \frac{1}{2}\ddot{x}_s(t^n)(t-t^n)^2 + \dots \tag{5.51}$$

where $\dot{x}_s(t^n)$ is the discontinuity velocity at point $(\xi_{j-\frac{1}{2}}^n, t^n)$, $\ddot{x}_s(t^n)$ is the discontinuity acceleration

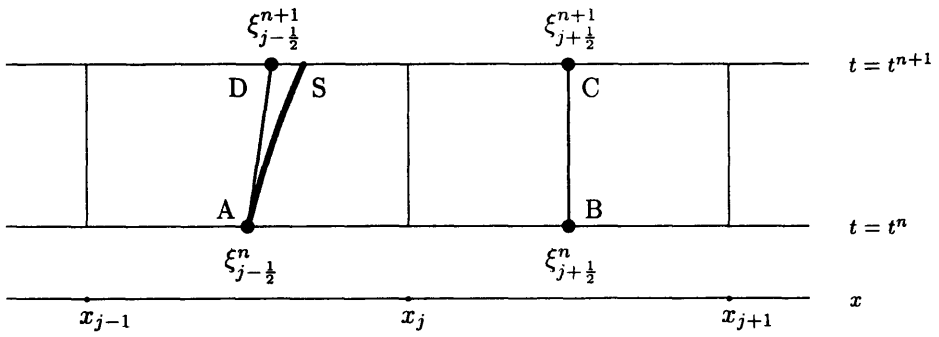

Figure 5.10: Accuracy in the cell with a shock. Thin lines represent underlying fixed mesh and boxes. Filled circles represent nodes. Lines of intermediate thickness represent paths of interfaces. The thick line represents the *real* trajectory of the shock.

at point $(\xi^n_{j-\frac{1}{2}}, t^n)$, and so on. Therefore, at time $t = t^{n+1}$ the location of the discontinuity can be computed as

$$x_s(t^{n+1}) = \xi^n_{j-\frac{1}{2}} + \dot{x}_s(t^n)\Delta t^n + \frac{1}{2}\ddot{x}_s(t^n)(\Delta t^n)^2 + \dots \tag{5.52}$$

The velocity of the discontinuity $\dot{x}_s(t^n)$ is know as long as the j-th Riemann problem is solved. To find the acceleration, we need to solve the Generalized Riemann problem up to the second order — to enable this, linear reconstruction of the data is required (see subsection 4.1.8), and we can go on and on.

In this chapter we consider a first order accurate scheme. It uses piecewise constant data, and solving Riemann problems provides the velocities of all significant waves, $\dot{x}_s(t^n)$. Therefore, within the accuracy of this method, the significant wave position can be computed with the error of $O((\Delta t^n)^2)$ — from (5.52):

$$x_s(t^{n+1}) = \xi^n_{j-\frac{1}{2}} + \dot{x}^n_s \Delta t^n + O((\Delta t^n)^2) \tag{5.53}$$

Thus, when we say we place a node at the discontinuity position at time t^{n+1}, it means that the discontinuity location will be within $O((\Delta t^n)^2)$ distance from the node position, distance DS in the Figure.

Because AS is a significant discontinuity, the jump at the density across it is $O(1)$. Therefore, if we start with the averages of the exact solution $\bar{u}^n_j$ at time $t = t^n$, at the next time level the scheme outcome will be the approximation u^{n+1}_j of the averages of the true solution $\bar{u}^{n+1}_j$ at time $t = t^{n+1}$,

which has a first order error:

$$u_j^{n+1} = \bar{u}_j^{n+1} + O((\Delta t^n)) \tag{5.54}$$

Thus, the truncation error for a cell with at least one significant interface is $O(1)$. The scheme is losing one order of accuracy on such cells.

The similar situation would happen for the second order scheme — it would be second order locally accurate (in the sense of L^1-norm) for each cell except for the cell with at least one significant node. For such cells the local accuracy will drop one order — the scheme will be first order accurate for such cells. The same pattern holds for higher order extensions: For the r-th order accurate scheme, the accuracy drops one order in the significant and half-significant cells.

Comparing with Godunov-type schemes on a fixed grid, this is a real advantage of the method. All such schemes are *zero* order accurate throughout *several* cells in the vicinity of a discontinuity.

We can upgrade the performance of the scheme in a (half)significant cell if, in this cell and its neighbor(s), we carry on data reconstruction one order higher than in a regular cell and solve the Generalized Riemann problem with one extra order of accuracy. Then we reduce the error of the shock location one order down, which elevates the formal truncation error for such a cell one order up, making the formal accuracy of the scheme uniform for all cells. For instance, for the first order accurate method we should use linear rather than constant data reconstruction next to a significant node and then solve the Generalized Riemann problem up to the first order to approximate the discontinuity path by parabola rather than by a straight line.

This approach is close in its spirit to the PPM method of Colella and Woodward [3] and subcell resolution scheme of Harten [10]. Colella and Woodward used third order reconstruction (parabolas) for the second order accurate method. Harten employed third order reconstruction only for the second field of the Euler equations — the field responsible for contacts, while the other two waves were reconstructed only up to the second order, which limited the accuracy of the method to the second order. Our upgrading the accuracy of the discontinuity path calculation is kind of localized, at the significant node, version of Colella and Woodward/Harten approach.

5.3.5 New Scheme. Performance

We would like to lay the stress on the fact that at every time level we have a Godunov's method on the essentially *uniform* grid — only couple of nodes are displaced from box centers. Therefore, in its computational speed, the new method does not change much from a regular Godunov's scheme. On the other hand, the new method surpasses any fixed grid Godunov's scheme in accuracy. This is achieved because we were able to devise a method which, first, selects few crucial points which can strongly influence the accuracy and, then, concentrates most of its efforts on the chosen points,

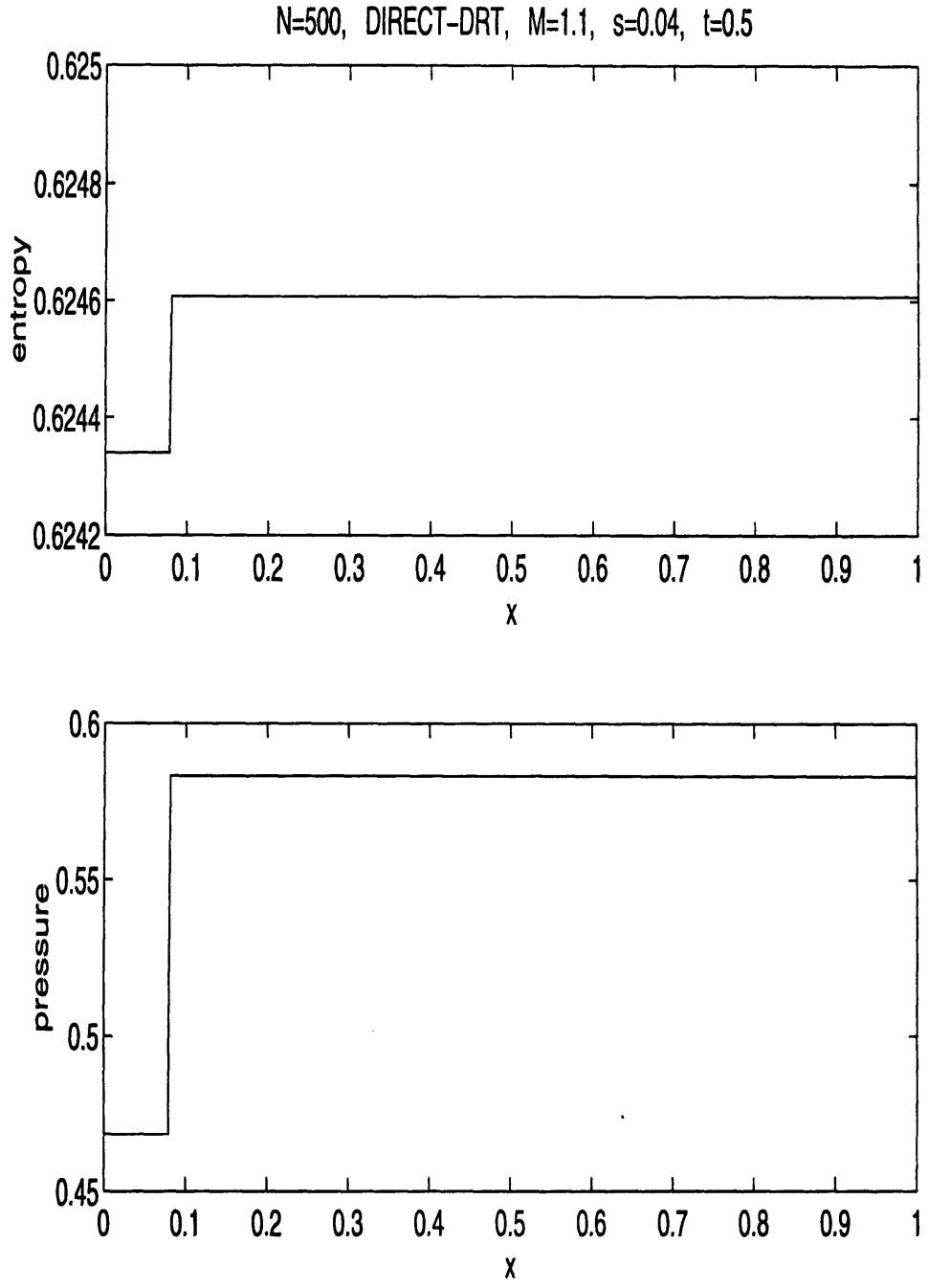

Figure 5.11: Moving Shock. Example 4.1. Entropy and Pressure as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

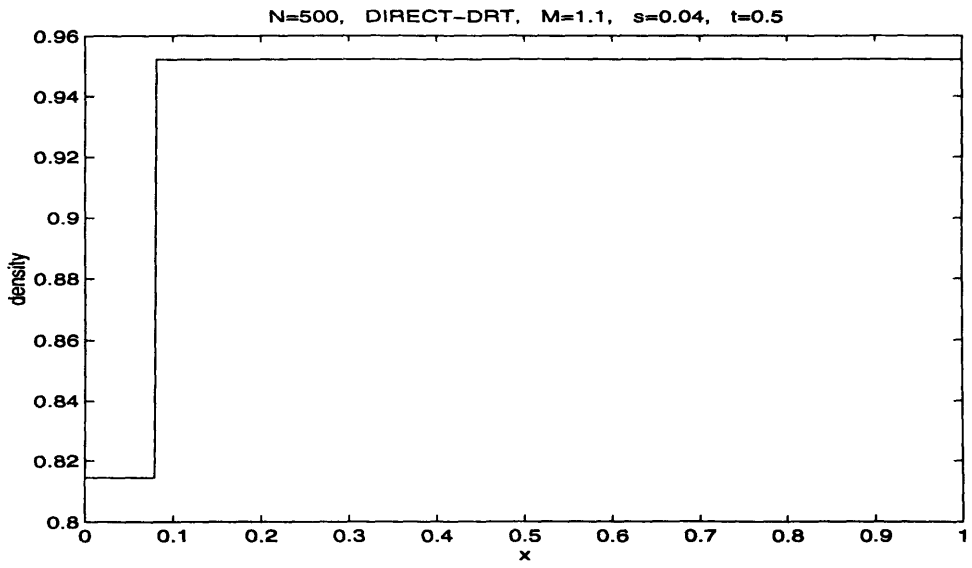

Figure 5.12: Moving Shock. Example 4.1. Density as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

providing them with exceptional treatment. With this approach, we pay the price for the improved accuracy in complicated programming, which is done once and for all, rather than in the performance slow down.

The second accent we would like to place on the "recycling" tactics we used in our algorithm. Once the data were obtained, the computational time was spent, we try to use and reuse them as much as possible. The data gained by solving a Riemann problem are used, besides their direct purpose to participate in flux computations, in significant waves selection, the time step generation, and the new grid construction. It leads to the accuracy improvement with almost no extra computational time being spent.

All in all, when we measured the performance of the method on some typical test problems, we never observed any slow down beyond 10%.

The scheme does not use much of the computer memory either. It does not require to store much beyond the needs of a fixed grid Godunov's method. Because the grid is essentially uniform all the time, the scheme does not need to store the grids and the data for two consecutive time levels (unlike the self adjusting grid scheme). Besides, the method does not track the discontinuity in the traditional sense, all the time storing their locations and other essential parameters in memory. On the contrary, the set of discontinuities to be followed is constructed anew at the beginning of each step and is forgotten at the end.

The scheme gives us two bonuses.

- All nonlinear interaction is concentrated around the strong discontinuities of the flow. At these places the exact Riemann solver must be used. However, the exact Riemann solver is usually

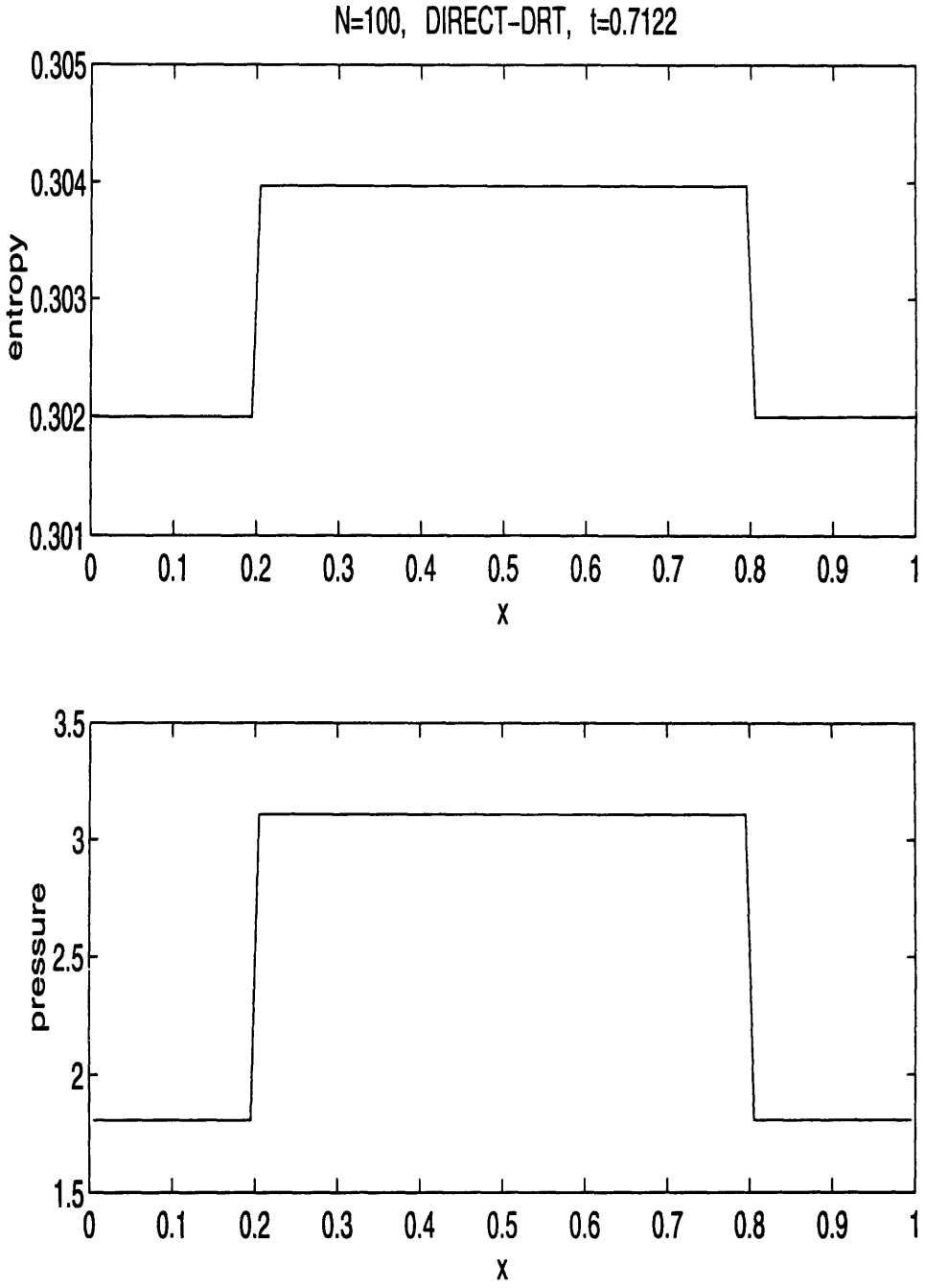

Figure 5.13: Typical head-on shock collision problem. Initial conditions are given in (4.18). Entropy and Pressure as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

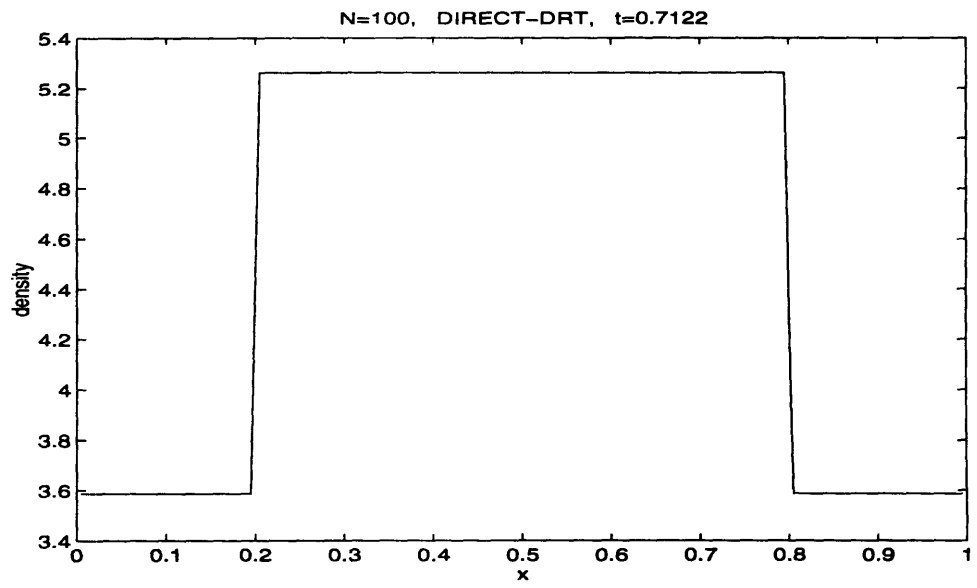

Figure 5.14: Typical head-on shock collision problem. Initial conditions are given in (4.18). Density as Function of Distance along Tube, x. Solid line represents simulation results. Dashed line represents exact solution.

rather expensive. For instance, in the case of the Euler equation it solves the system of two nonlinear algebraic equations using Newton's method, which employs many iteration calls. While the use of the exact Riemann solver is absolutely mandatory at the discontinuities, it turns out to be unnecessary in the smooth regions of the flow. Since our new scheme detects the flow discontinuities anyways, we can use expensive direct Riemann solver only at the significant nodes, and employ cheap approximate solver, like Roe's [25] or Osher's [22] (or even linear finite difference approximation), in the rest of the nodes, which constitute the overwhelming majority. The use of fast approximate Riemann solvers for the most of the nodes should considerably augment the speed of the algorithm.

- As we pointed out earlier (subsection 4.1.8), the reconstruction step proves to be the most crucial one for high resolution methods. The main obstacle is the spurious oscillations generated as a result of interpolation across a discontinuity. However, the new scheme keeps the strong discontinuities one point sharp and knows their precise locations. Therefore, it takes no effort at all to avoid the interpolation across a discontinuity. Thus, the importance of the particular reconstruction technique being used is played down. In the following section we extend our scheme to the second order of accuracy. We implemented this second order accurate method using different reconstruction procedures. As our computational experience with the scheme indicates, in practice, the choice of a particular reconstruction techniques makes little difference indeed [14].

[14]It is still important not to interpolate across the weak discontinuities, edges of the rarefaction waves. For this

The performance of the new scheme on the problems of a moving shock, section 4.2.1, and two shock collision, section 4.2.4, is illustrated in Figures 5.11, 5.12 and Figures 5.13, 5.14 . What we can see in these figures is completely different from the previous results, Figures 4.4, 4.5 and Figures 4.17, 4.18. While Figures 4.4, 4.5 and Figures 4.17, 4.18 display great amount of different kinds of parasitic waves, one can not find any oscillations in Figures 5.11, 5.12 and Figures 5.13, 5.14. Actually, one can not tell the exact solution from the computed one from these figures. The scheme provides a perfect treatment of a single discontinuity motion as well as discontinuity interactions.

5.3.6 Beyond Discontinuity Treatment Scheme

To be able to use the DT scheme for long time numerical computations, we need to furnish it with some additional features.

First of all, we should upgrade the accuracy of the scheme. For the numerical experiments of the next chapter, we use second order scheme. In this scheme, after the non-oscillatory reconstruction has been done and piecewise linear data approximation has been obtained, the generalized Riemann problem is solved for each interface, and the fluxes are computed up to the second order of accuracy [34].

We improved interaction treatment by providing initial resolution not only to the strong discontinuities, the shocks and the contacts, but also to the rarefaction waves. When the strong rarefaction wave is generated we split the fan into the several wedges. This prevents the scheme from the averaging across the rarefaction wave which would lead to the entropy production [32]. Depending on the level of the parasitic entropy which we permit in the simulation and on the strength of the rarefaction, the number of the wedges is chosen.

The splitting provides the initial resolution of a strong rarefaction wave. Usually, we need to keep the wave resolved for some period of time, until it weakens enough. We achieve it with a conservative multi-grid approach. We superimpose a fine grid on the region currently occupied by the rarefaction. The procedure is developed in such a way that the method remains conservative [31]. Besides, Compliant Interface approach [31] prevents the method from noise generation at the grid boundaries.

purpose, the choice of the reconstruction technique retains its significance. However, the magnitude of the spurious waves produced by the accidental interpolation across an edge of a rarefaction wave is of a secondary order comparing with an amplitude of the oscillations generated due to an interpolation across a shock. Therefore, the requirements to the choice of the reconstruction technique to be applied in the vicinity of weak discontinuities are much milder than the requirements to the reconstruction techniques dealing with strong discontinuities.

Chapter 6

Long Time Numerical Simulation

In this chapter we present our numerical results on the long time behavior of the solutions of the Euler equations with "arbitrary" piecewise constant initial data. In all numerical integrations we carried out, regardless of the piecewise constant initial conditions being used, the fully nonlinear non-breaking standing acoustic waves of the type described in section 3.2 always eventually emerge in the flow. Therefore, these experiments provide substantial evidence of the strongly attracting character of the resonant waves.

6.1 Overview of Experiments

We carried out numerous long time numerical simulations of the inviscid gas motion inside a close ended tube. Various piecewise constant initial conditions of the type described in subsubsection 2.1.2 were used: In all our experiments the tube is initially divided by membranes into several sections, the quiescent gas has different constant pressures and densities in each section of the tube. In all our experiments we modeled the tube of unit length (as it was mentioned in subsubsection 2.1.2) filled with air at moderate temperature: $\gamma = 1.4$.

Always uniformly second order DRT scheme described in the previous chapter was used. We ran the simulations on the grids with different cell sizes: Number of computational cells ranged from 100 to 2000. For any initial data we used, we always were able to find a cell size (which depended on the particular initial data), such that all experiments performed on grids with smaller cell sizes showed very similar behavior which did not depend anymore on the scheme resolution: We believed that the numerical solution converged to the physical solution. We will call the grid with the critical cell size a principal grid. (See also Remark 6.2 below). For some initial data 100-cell grids were sufficient for the convergence. However, some initial data required higher resolution, up to 400 cell.

Out of the various initial value problem simulations we conducted, only two are reported here due to the brevity motivations of . These particular initial conditions have been chosen for the

illustration mostly because of the "historical" reasons: They have been widely used as test problems for new schemes for quite a while. These initial conditions have been around for a long time, so we think they can serve as rather convincing arguments.

The first problem, known as the **blast waves problem**, was first introduced in [36]. In has been extensively used afterwards as a test problem (see for, example, [10, 11, 35]). The initial condition consists of three constant states of a gas which is at rest. The density is everywhere unity, while in the leftmost tenth of the volume the pressure is 1000, in the rightmost tenth it is 100, and in between it is 0.01.

$$u(x,0) = \begin{cases} u_L & \text{if } 0.0 \le x < 0.1, \\ u_M & \text{if } 0.1 \le x < 0.9, \\ u_R & \text{if } 0.9 \le x < 1.0. \end{cases} \tag{6.1}$$

where

$$\begin{aligned} (\rho_L, v_L, p_L) &= (1, 0, 10^3), \\ (\rho_M, v_M, p_M) &= (1, 0, 10^{-2}), \\ (\rho_R, v_R, p_R) &= (1, 0, 10^2). \end{aligned} \tag{6.2}$$

Immediately after the membranes has been removed, two *strong* blast waves develop and collide, producing a new contact discontinuity. This evolution is quite complex.

The second problem, known as the **Sod example**, was first introduced in [27]. Since then, it showed up in a multitude of papers as a test problem [8, 11, 12, 14, 25, 29]. The initial condition consists of two constant states of a quiescent gas. At the middle of the tube the density and the pressure jump down a factor 8 and a factor 10, respectively.

$$u(x,0) = \begin{cases} u_L & \text{if } 0.0 \le x < 0.5, \\ u_R & \text{if } 0.5 \le x < 1.0. \end{cases} \tag{6.3}$$

where

$$\begin{aligned} (\rho_L, v_L, p_L) &= (1.000, 0, 1.0), \\ (\rho_R, v_R, p_R) &= (0.125, 0, 0.1). \end{aligned} \tag{6.4}$$

The initial discontinuity in this problem breaks into a *weak* shock wave followed by a contact discontinuity and a rarefaction wave.

Remark 6.1 Two most important differences between the blast waves problem and the Sod example are: (a) The shocks developed in the blast waves problem are ***strong*** while the shock generated in the Sod example is ***weak***; (b) Three main contacts are formed at the beginning of the blast waves problem simulation, whereas only one principal contact arises in the flow in the Sod example. These differences result in different properties of the resonant waves that will emerge in the simulations: the wave of the blast waves problem is much stronger and exhibits more complicated behavior than that of the Sod example.

6.2 Data Presentation. Physical and Fourier Spaces

There are two complementary ways to view the solutions of the system of hyperbolic partial differential equations (2.8), $u(x,t)$.

- x-SPACE REPRESENTATION. The solution $u(x,t)$ is considered as a time evolution of *one* vector-function $u(x,\cdot)$ of a *continuous* variable x. The spatial variable x is continuous, it can assume any value from the interval $(0,1)$.

- k-SPACE REPRESENTATION. After we replace the original close ended problem with the equivalent periodic problem with the period 2, twice the tube length (see subsection 2.1.2), we can express the solution $u(x,t)$ as a Fourier series with respect to x,

$$u(x,t) = \sum_{k=-\infty}^{\infty} u_k(t) e^{\mathrm{i}k\pi x}, \qquad -1 < x < 1 \tag{6.5}$$

where $\mathrm{i} = \sqrt{-1}$, and

$$u_k(t) = \frac{1}{2} \int_{-1}^{1} u(x,t) e^{-\mathrm{i}k\pi x}\, dx, \qquad k = 1, 2, \ldots \tag{6.6}$$

Therefore the solution $u(x,t)$ can be considered as a time evolution of an ***infinite countable*** set of complex vectors $\{u_k(\cdot)\}$. The spatial variable k is discrete, it can take only integer values.

This switch from the continuum, x-space, to the infinite dimensional k-space is, in a sense, equivalent to thinking of the system of hyperbolic partial differential equation as an infinite dimensional system of ordinary differential equations. This will allow us to employ some techniques which are commonly used for finite-dimensional dynamical systems. We look at the behavior of the solutions in k-space in section 6.4.

The advantage of the solution representation in the Fourier space is that it enables one to observe and analyze the dynamics on different scales of the flow separately: Low wave-number modes, absolute values of k close to zero, represent the large scale motion; while the small scale evolution manifests itself in the dynamics of high wave-numbers.

In the case of the Euler equation, due to the symmetry (see subsection 2.1.2), the pressure, being an even function of x, is expressed as a cosine series

$$p(x,t) = \frac{1}{2}p_0(t) + \sum_{k=1}^{\infty} p_k(t)\cos(k\pi x), \qquad 0 < x < 1, \tag{6.7}$$

while the velocity, being an odd function of x, is expressed as a sine series:

$$v(x,t) = \sum_{k=1}^{\infty} v_k(t)\sin(k\pi x), \qquad 0 < x < 1, \tag{6.8}$$

where

$$p_0(t) = 2\int_0^1 p(x,t)\,dx, \tag{6.9}$$

$$p_k(t) = 2\int_0^1 p(x,t)\cos(k\pi x)\,dx, \qquad k = 1,2,\ldots \tag{6.10}$$

and

$$v_k(t) = 2\int_0^1 v(x,t)\sin(k\pi x)\,dx, \qquad k = 1,2,\ldots \tag{6.11}$$

It should be noticed that the zeroth Fourier coefficient of the pressure (6.9) is twice the pressure average defined as

$$\langle p \rangle = \int_0^1 p(x,t)\,dx. \tag{6.12}$$

6.3 Three Stages of Flow Evolution

For any piecewise constant initial set-up, we always observed similar patterns in the resulting flow. In all the experiments three main stages could be clearly distinguished:

1. INITIAL STAGE. The stage of *formation.* The main features, which the flow will possess throughout the whole experiment, are, for the most part, already developed by the end of this

stage. In this sense, this stage *determines* the following two stages. It is a *fast transition* stage. The stage is characterized by the presence of strong shocks, vehement interaction, and swift changes. It is a *transient* stage, lasts for a very short time. The resonant standing wave can not be seen in the pressure and velocity profiles yet, but it is mostly developed with all its main features — as long as the dynamics of the low wave-number Fourier coefficients of the velocity and the pressure is concerned — by the end of this stage.

2. INTERMEDIATE STAGE. The main characteristic of this stage is that all *shocks are dying* throughout it. It is a *slow transition* stage. Slow and small scale changes occur during this stage. This stage takes much more time than the previous one. The resonant standing wave, while hardly distinguished in the pressure and velocity profiles at the beginning of the stage, is clearly visible in the pressure and velocity by the end of the stage.

3. FINAL STAGE. This is a *stationary* stage. The motion is quasiperiodic. Everything is killed in the previous stage but the resonant standing wave. At this stage, the wave is the flow, the flow is the wave. The wave is fully nonlinear and develops no shocks. This stage lasts "forever" (before the numerical viscosity will be able to kill it).

In the this section we describe each stage in detail.

6.3.1 Initial, Fast Transition, Stage

After the membranes have been suddenly removed at time $t = 0$, shocks, contact discontinuities, and rarefaction waves immediately develop and begin to interact with each other and reflect from the end walls. Due to the interactions, the shock strength drops drastically, the average pressure and the total entropy of the system rapidly increases, and the principal/large contacts — which will mainly determine the final entropy distribution (which is to regulate the long time behavior of the flow as it follows from chapter 3) — are formed. At the end of this stage only few (usually one or couple) strong shocks can be observed, which begin to travel from one end of the pipe to the other, reflecting from the tube end walls and reflecting/refracting on the principal/large contacts. This back and forth movement of the shocks sets off the splashing motion of the bulk of gas in the pipe, which will eventually, at the later stages, result in the appearance of quasiperiodic non-breaking fully nonlinear standing waves.

Even though this stage lasts for only a short period of time — few acoustical periods [1], it is characterized by very intense interaction processes and is extremely important for the whole experiment: At this stage all main features of the flow to ensue are formed — the principal entropy

[1] We call the acoustical period a time which is required for a wave traveling at the average sound speed to move from one end of the tube to the other, reflect, and come back.

background (which will control the long time behavior), the main frequencies and the amplitudes of the standing waves which will dominate the flow at the following stages.

Although, at this stage, the standing waves themselves can not be distinguished, if one looks at the density, pressure, or velocity distribution, the embryo of the future standing wave is already there. One can observe it from the time history of the zeroth Fourier coefficient of the pressure or the first Fourier coefficient of the velocity (see Figures 6.1 and 6.2): Already at this stage they exhibit time oscillations with the main frequency close to the main frequency of those oscillations from the later stages. However, the amplitudes of the oscillations are larger than the amplitudes of those oscillations from the following stages. They dwindle rapidly during this stage and, to the end of this stage, become practically equal to the amplitudes of the stationary oscillations of the last stage, corresponding to the stationary/final standing waves. The level around which the average pressure oscillates also dramatically changes during this stage: Initially it is lower than the level around which the average pressure oscillations occur at the last, stationary, stage; but it quickly rises up to the level of the final stage.

6.3.2 Intermediate, Slow Transition, Stage

By the beginning of this stage, the principal features of the final, stationary, flow have already developed; and throughout this stage the fine details of the ultimate flow are taking their final shape.

At the beginning of this stage, we still have a couple of strong shocks which survived the first stage. They travel back and forth, bumping into the contacts and the end walls. Due to this interactions, the shocks are gradually getting weaker. In contrast to the first stage, the shocks of this stage are of moderate strength, the interaction is less vigorous, and the rate at witch the shocks decay diminishes. The weaker a shock grows, the less its decay rate becomes. The stage ends when all the shocks eventually die out. For this to happen, many interactions should occur, therefore this stage is much longer than the previous one. This stage usually lasts for several hundred acoustical periods.

Remark 6.2 We should clarify the meaning of the phrase "the shocks eventually die out". Obviously, from the mathematical point of view, a shock never dies, it just becomes weaker and weaker. Therefore, if we would like to talk about the shock free flow, we can only speak of the flow which is a *limit*, in some sense, of a sequence of the flows with the decreasing shock strengths. However, for all practical purposes we can employ the following procedure. We run the experiments with the same initial conditions on a sequence of grids with shrinking cell size, for instance, grids with 100, 200, 400, 800, 1600 points. For a given cell size, $\Delta\xi$, we, evidently, consider the shock dead if we can not mark it as a significant wave — the shock strength is less that the threshold. When we choose the

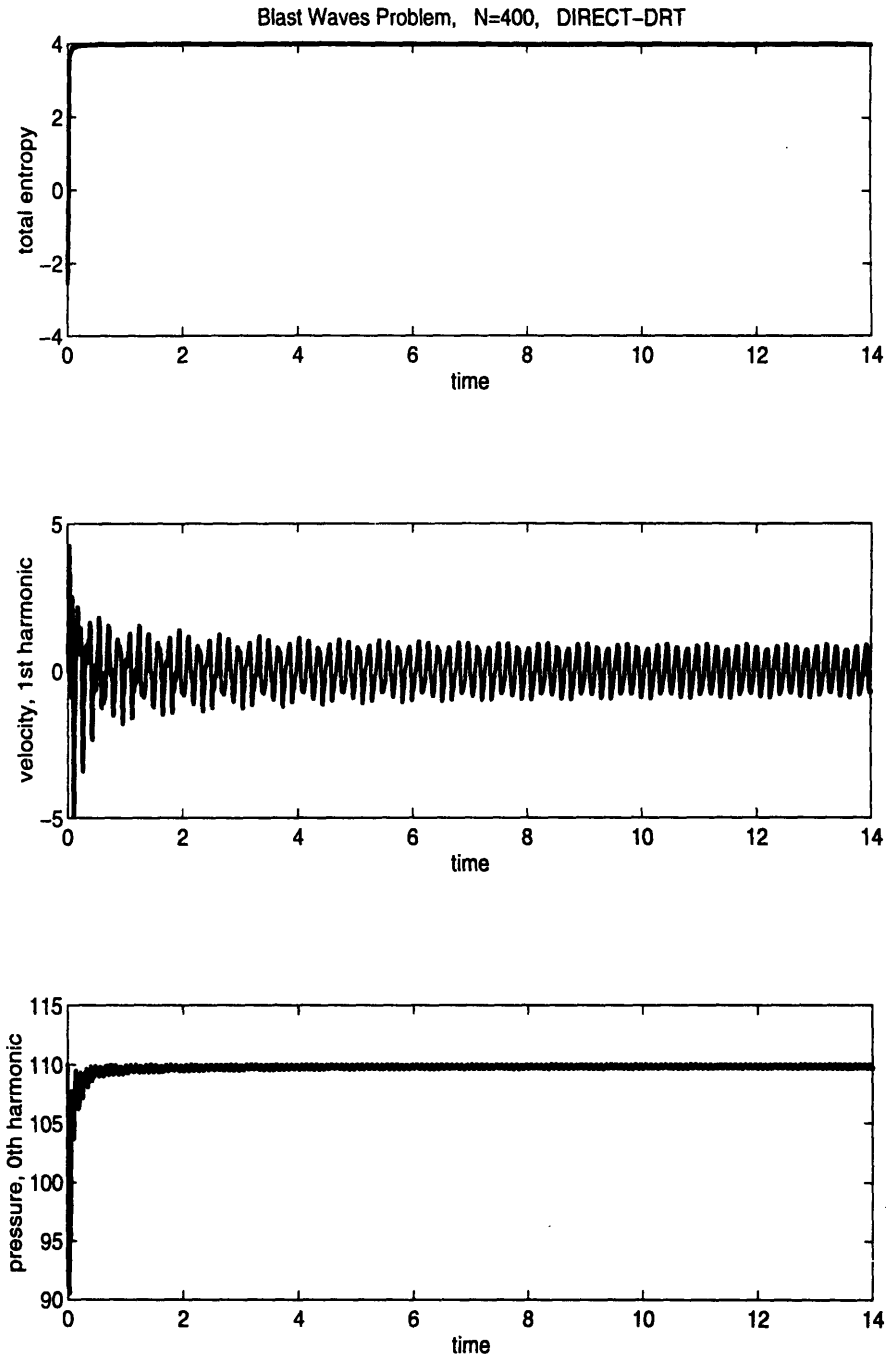

Figure 6.1: The blast waves problem. The first two stages of flow evolution.

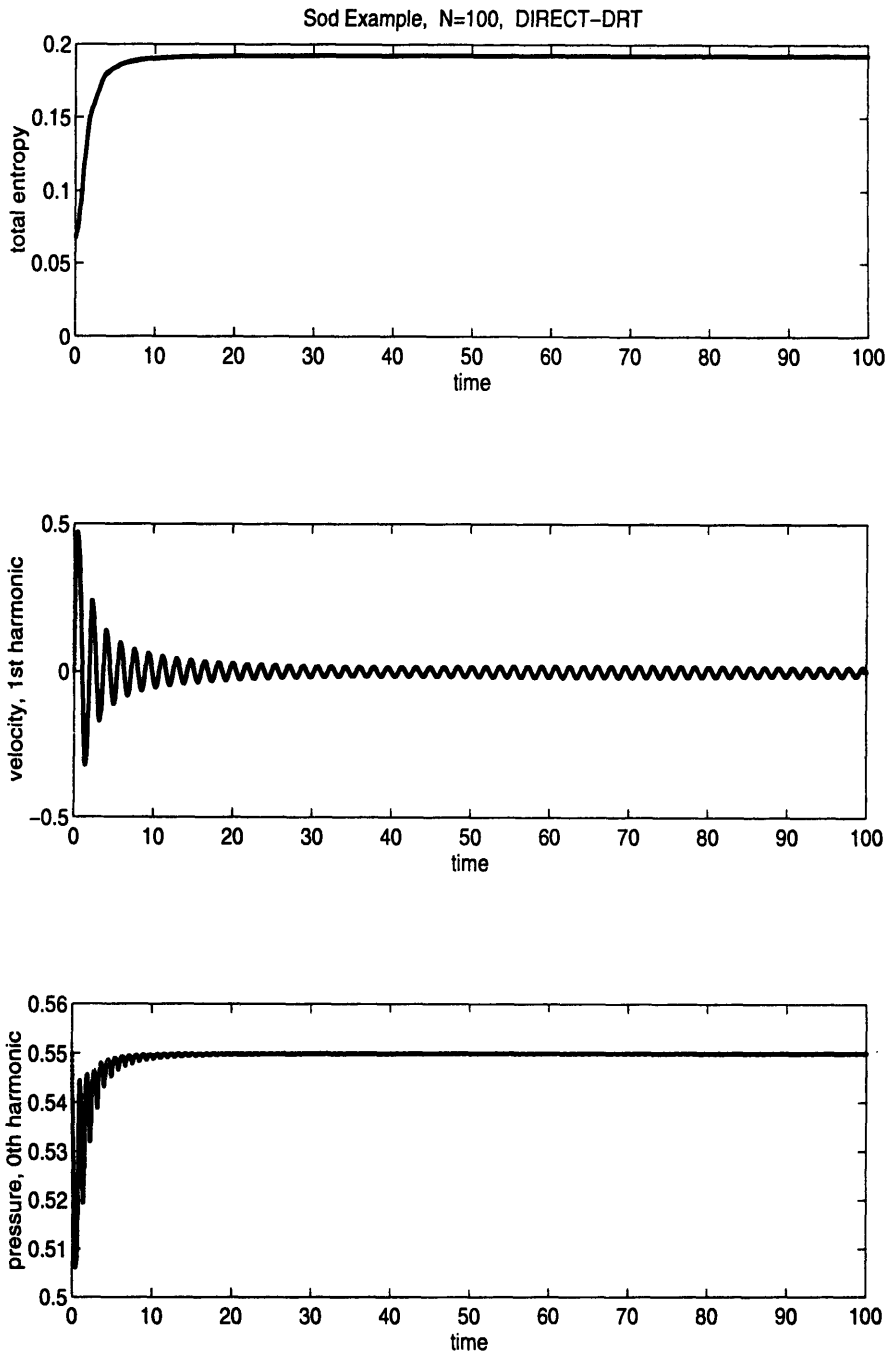

Figure 6.2: The Sod example. The first two stages of flow evolution.

threshold as it was suggested in subsubsection 5.3.1, formula (5.32), the threshold is proportional to the cell size, $\Delta\xi$. Therefore, as we reduce the mesh spacing, we lower the threshold at the same time, which makes us to "track" weaker shocks. For coarse grids, the final result, the properties of the final standing waves — mainly the phase and less evidently the amplitude — turn out to depend on the mesh spacing and the size of the threshold. However, beginning from some grid resolution (which depends on the particular problem), this influence of the mesh spacing becomes negligible (from what we conclude that the numerical results converge to the true solution). We call the grid with the cell size at which the effect of the grid resolution becomes unimportant the principal grid (see section 6.1 above). The motion is considered shock free, and the second stage ends, when shock "tracking" is relinquished on the principal grid.

During the second stage the total entropy still rises but at the rate which is much lower than that at the first stage (see Figures 6.1 and 6.2). After the second stage is over, there are no shocks left, the only increase in the total entropy is due to the scheme viscosity. The whole entropy spatial structure, developed at the first stage, endure the second stage without any major changes. It is dominated by the principal contacts formed at the first stage. While all the shocks vanish during the second stage, the principal contacts, their strength and their motion, remain chiefly unchanged. As we mentioned earlier, these contacts are the main structures of the entropy background which determine the properties of the resonant non-breaking standing waves.

The shapes of the pressure and velocity profiles go through the visible changes during the second stage. At the beginning of the stage, they appear to be very jagged and irregular due to the presence of the shocks and rarefactions. Although the motion of the gas as a whole, after the first stage is over, already displays all main features of a splashing wave, the shape of the standing wave is still badly contaminated with the multitude of secondary waves and shocks still existing in the duct. Throughout the second stage, the secondary waves and shocks die out, and the fully developed nonlinear resonant standing wave comes to dominate the flow. Its periodic motion now can be easily observed at the pressure and velocity profiles, which by the end of the second stage lose their ragged appearance.

The time history of the zeroth Fourier coefficient of the pressure and the first Fourier coefficient of the velocity exhibit oscillatory behavior already at the beginning of the second stage (Figures 6.1 and 6.2). The main frequencies, the amplitudes, and the levels around which the oscillations occur are very close to those of the oscillations corresponding to stationary standing waves of the third stage. In this sense the standing waves are present at the second stage. However, the fine details of those oscillations undergo some changes. While at the beginning of the second stage there are some small amplitude disturbances presented at the oscillations, they are all gone by the end of the stage. The resulting oscillations are not contaminated by any residual noise. It should be noticed that the remaining oscillations usually have several frequencies, although the one is clearly dominant.

6.3.3 Final, Stationary, Stage

At the third, and final, stage the flow motion is just a standing fully nonlinear non-breaking resonant wave. There are no shocks left in the flow; no dissipation, except for the numerical dissipation of the scheme.

The whole motion consists of the self repeating patterns (Figures 6.3–6.6): At some moment the velocity of the gas inside the tube is zero almost everywhere, but the pressure increases from one end of the tube to the other. This gradient of the pressure causes the gas to begin to move in one direction. The gas is accelerated by the pressure gradient. At the same time as the speed of the gas increases, the pressure gradient weakens. At the moment when the gradient becomes zero almost everywhere, the gas moves with its largest velocity. Then the pressure gradient changes its sign which leads to the gas deceleration. While the gas slows down, the magnitude of the pressure gradient increases. At the moment the gas speed becomes zero almost everywhere in the tube (half a cycle is completed), the pressure gradient assumes its largest value which makes the gas to start moving in the opposite direction. Then, while the gas speeds up, the gradient drops, crosses zero, the gas slows down, the pressure difference rises, the gas almost entirely comes to rest, the pressure gradient assumes its largest magnitude. We came back to the initial pattern. Then the cycle repeats itself, and so on.

So, we have a splashing motion of the gas. It moves back and forth inside the tube, with the most of the action taking place close to the middle of the tube (away from the pipe ends): that is where the largest gradients occur and the highest velocities appear.

The behavior described above is very similar to that of a small amplitude acoustic standing wave. It is the same type of the pressure gradient driven oscillations. They have the same frequency as a sound standing wave in this tube would have:

$$\mathrm{f} = \frac{\omega}{2\pi} = \frac{1}{2}\langle c \rangle \tag{6.13}$$

where $\langle c \rangle$ is an average sound speed. However, those waves are fully in the nonlinear regime. (The amplitude of the pressure oscillations amounts to 25% of the average pressure in the blast waves problem and to 10% in the Sod example.The amplitude of the velocity oscillations reaches 10% of the average sound speed in the blast waves problem and 4% in the Sod example. The wave amplitudes in the Sod example are two and a half times smaller than the wave amplitudes in the blast waves problem — Remark 6.1).

Although these waves are fully nonlinear, they never show any evidence of shock formation. After the second stage is completed, there are no shocks left in the flow, and no new shocks ever appear. Since there are no shocks at the third stage, there is no dissipation in the flow, except for the numerical dissipation of the scheme. The entropy remains constant up to the vanishingly

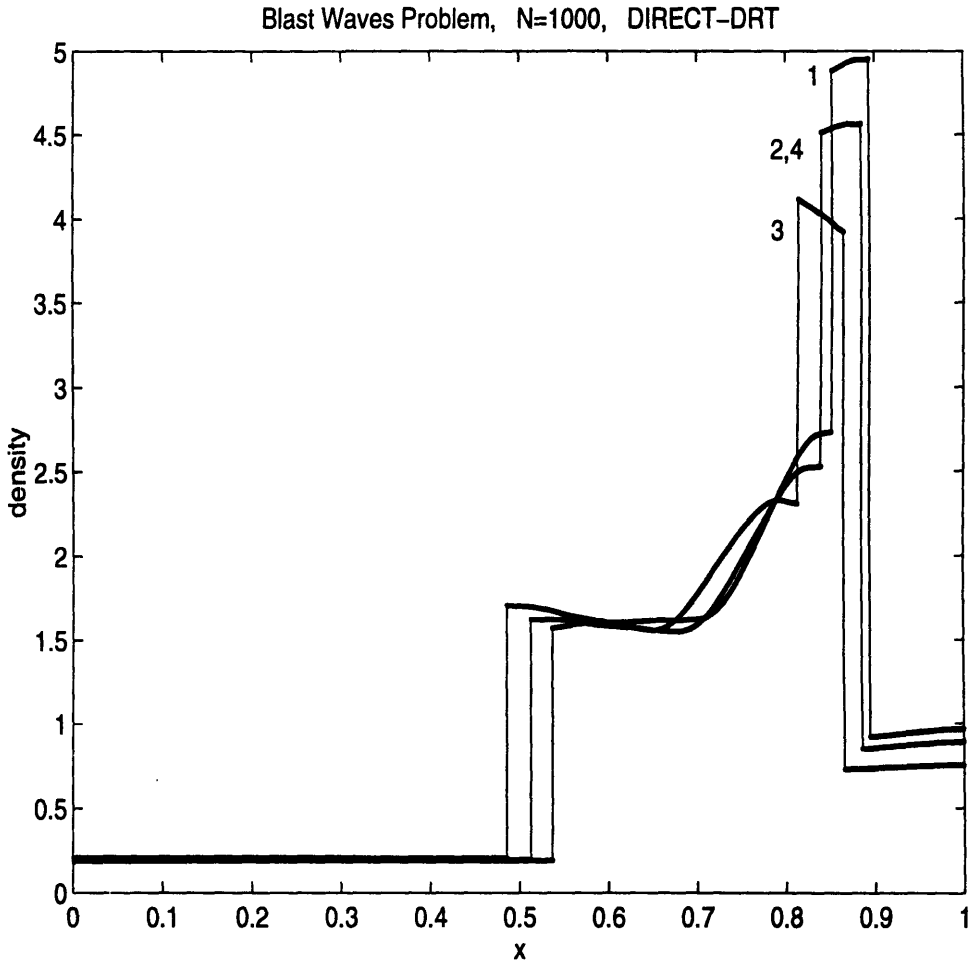

Figure 6.3: The blast waves problem. Density profiles of the resonant standing wave at four critical points of the period of the first Fourier coefficient of the velocity, $v_1(t)$: time 1 — $v_1(t)$ vanishes, time 2 — $v_1(t)$ assumes its maximum value, time 3 — $v_1(t)$ vanishes again, time 4 — $v_1(t)$ assumes its minimum value.

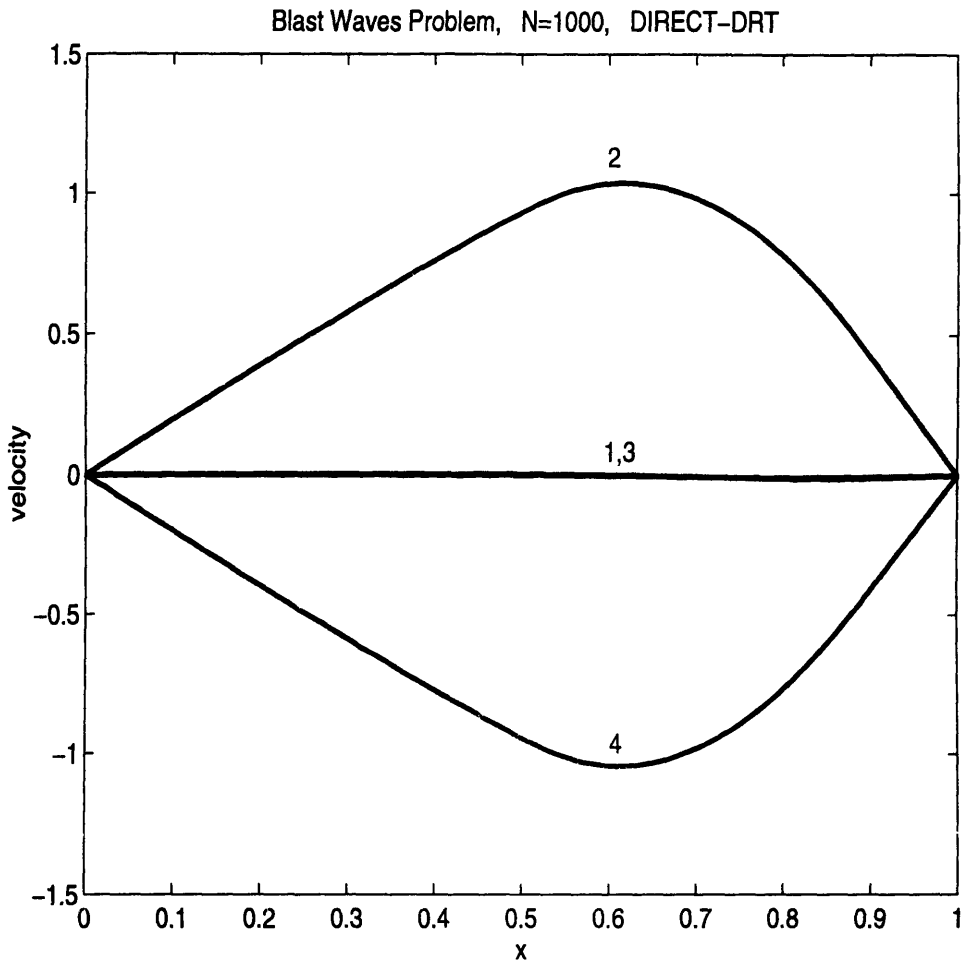

Figure 6.4: The blast waves problem. Velocity profiles of the resonant standing wave at four critical points of the period of the first Fourier coefficient of the velocity, $v_1(t)$: time 1 — $v_1(t)$ vanishes, time 2 — $v_1(t)$ assumes its maximum value, time 3 — $v_1(t)$ vanishes again, time 4 — $v_1(t)$ assumes its minimum value.

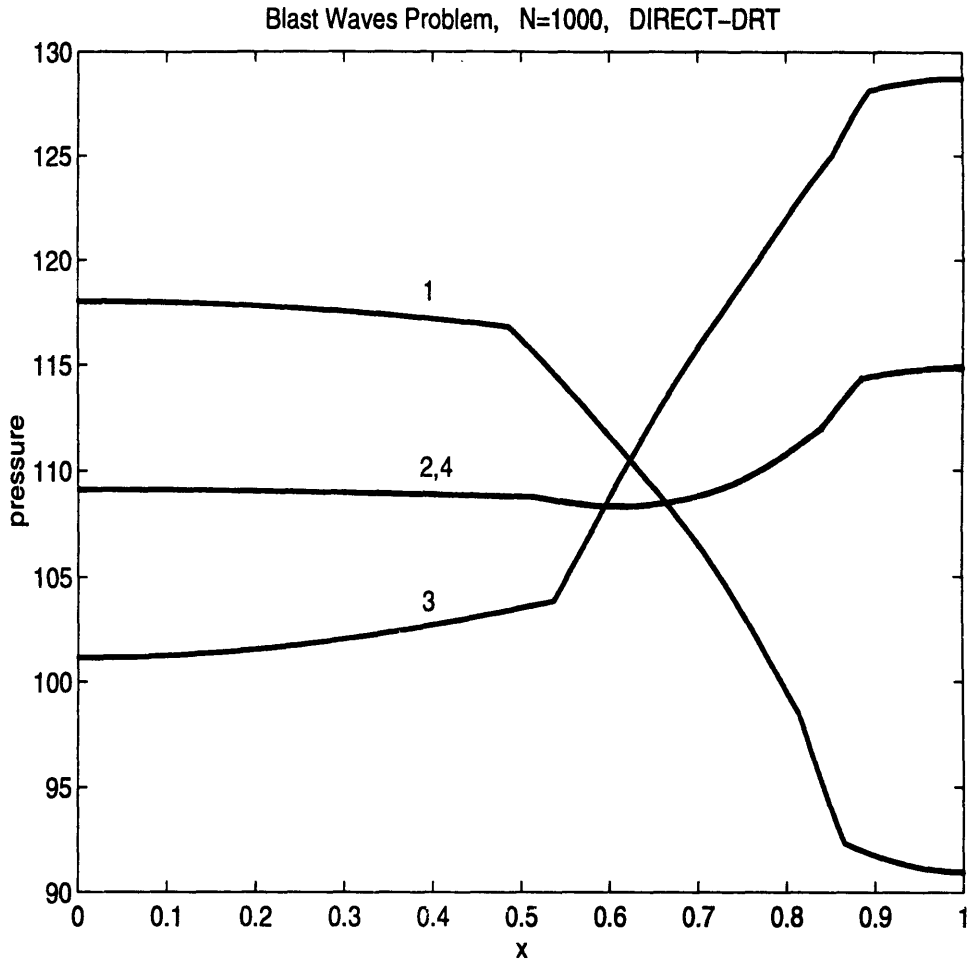

Figure 6.5: The blast waves problem. Pressure profiles of the resonant standing wave at four critical points of the period of the first Fourier coefficient of the velocity, $v_1(t)$: time 1 — $v_1(t)$ vanishes, time 2 — $v_1(t)$ assumes its maximum value, time 3 — $v_1(t)$ vanishes again, time 4 — $v_1(t)$ assumes its minimum value.

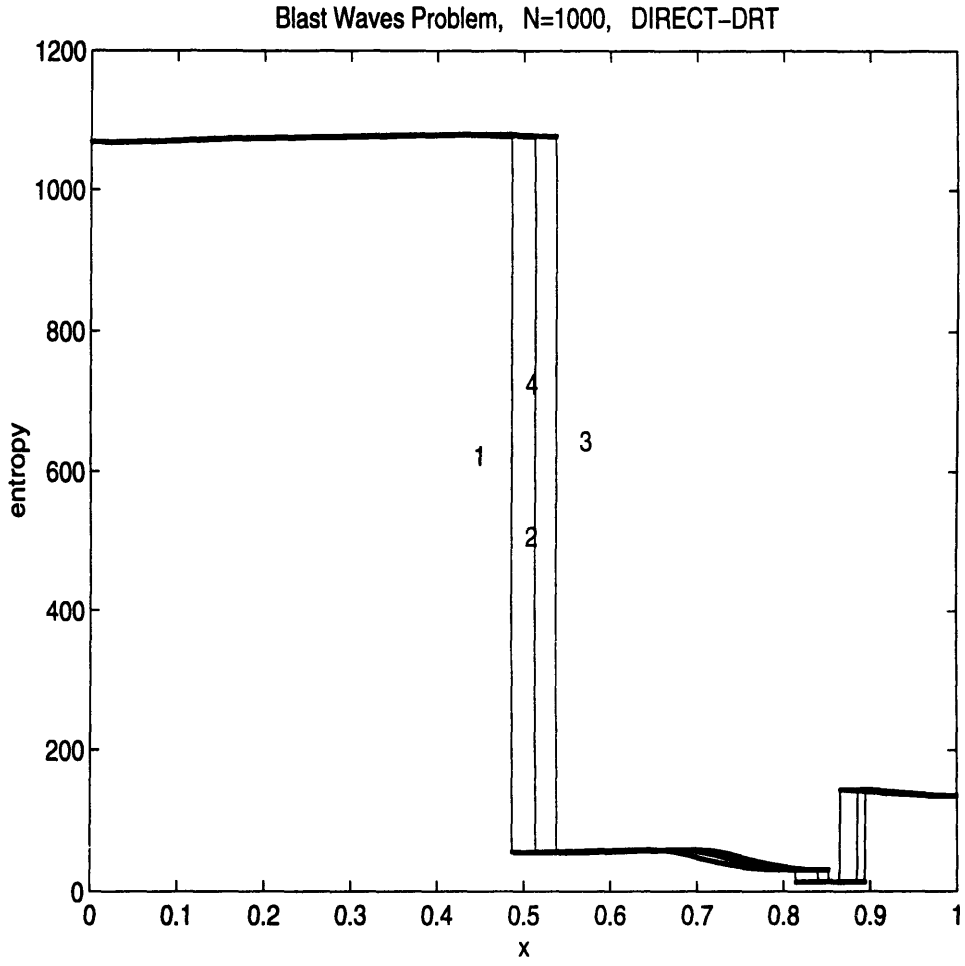

Figure 6.6: The blast waves problem. Entropy profiles of the resonant standing wave at four critical points of the period of the first Fourier coefficient of the velocity, $v_1(t)$: time 1 — $v_1(t)$ vanishes, time 2 — $v_1(t)$ assumes its maximum value, time 3 — $v_1(t)$ vanishes again, time 4 — $v_1(t)$ assumes its minimum value.

insignificant increase due to the scheme viscosity, which is far beyond the error of the method.

Because there is virtually no dissipation at the third stage, those standing wave motion can go on without any visible changes "forever", until the numerical viscosity eventually damp and kill it. Since the numerical viscosity of the second order accurate DRT scheme can be estimated as $O(\Delta\xi^3)$ [34], the numerical dissipation effects will show up only at the time scale $t = O(\frac{1}{\Delta\xi^3})$, which is translated into times orders of $10^6 - 10^7$ for the range of the cell sizes, $\Delta\xi$, used in our experiments, from $\frac{1}{100}$ to $\frac{1}{2000}$. In all our experiments, the average speed of sound ranged from order of ten (the blast waves problem) to order of unity (the Sod problem). Therefore, a typical acoustical period lay in the interval between order of one tenth and order of unity. That leads us to the estimate that the time scale of the viscous processes is about $10^6 - 10^8$ acoustical periods.

We ran some of our experiments up to several thousand acoustical periods. We observed that once all shocks died out and the pure standing resonant wave emerges at the end of the second stage, this self-repeating motion remains unchanged till the end of the experiment. This highly regular motion persists from the beginning of the third stage, which happens in a few ***hundreds*** acoustical periods after the start, until we stop the experiment, in a few ***thousands*** acoustical periods after the start. This is not surprising in light of the discussion of the previous paragraph.

Therefore, there exits a rather large time "window" for the resonant waves to exist without being killed by the *numerical* viscosity. What about the *physical* viscosity of the flow, which is neglected in the Euler equation (see section 2.1)?

There are, basically, two dissipation scales. One is provided by the viscous dissipation inside the body of gas, and is given by formula

$$t \ll L^2/\nu, \tag{6.14}$$

where ν is the kinematic viscosity of the gas, and L is a characteristic length of the problem. A quick calculation shows that this is a very long time: For $\nu = 1.5 \times 10^{-5} \frac{m^2}{sec}$ (dry air at 20^0 C and 1 atmosphere) this time scale can be estimated from (6.14) as $10^5 sec$ for a one meter pipe. However, the dissipation in the boundary layer is *much more efficient* as the "effective" L is a lot shorter. Actually, the dissipation in the boundary layer responds to a trickier equation than (6.14). The total amount of dissipation there goes like the area of the tube; and this must be compared with the volume (to get dissipation per unit volume). Thus one ends up with a formula where the ratio (pipe radius)/(boundary layer thickness) appears multiplying the estimate above. So, for a fairly "thick" pipe the boundary layer dissipation time will also be long. Finally, one can increase t by increasing the density of the gas — this makes ν smaller. Therefore, we can still have a fairly large time "window"to observe the resonant waves.

At first sight, the gas motion at the third stage looks exactly as a periodic large amplitude

acoustic standing wave. The waves, similar to such splashing mode waves, were reported in [1]. However, when we look more carefully at the standing waves appearing in our experiments, we can notice that in some experiments, besides the main frequency splashing mode, some other modes (although of much smaller amplitude) also have outlived the second stage and persist throughout the third stage. This can be most clearly seen from the temporal history of the zeroth Fourier coefficient of the pressure and the first Fourier coefficient of the velocity (see Figures 6.7 – 6.10). After the second stage is over, the first Fourier coefficient of the velocity oscillates with the frequency defined by (6.13), and the zeroth Fourier coefficient of the pressure oscillates with the frequency which is twice the frequency of (6.13). In some experiments, the amplitude of these oscillations remains fixed throughout the third stage, and the oscillations seem to have only one frequency. However, in most of the experiments, we observed that the amplitude of these waves are modulated by some lower frequencies. The low frequency amplitude modulations turns out to be quite large: Their magnitudes are usually of around several percents of the amplitudes of the oscillations. Hence, in some cases, the zeroth spatial harmonic of the pressure and the first spatial harmonic of the velocity can exhibit multi-frequency oscillations. This means that in these situations, after all shocks have died, not one but rather several modes of the resonant waves (see [1]) survived and abide at the third stage. Nevertheless, even in this case, the main splashing mode clearly dominates the rest.

Remark 6.3 The presence of several modes at the third stage explains why the flow velocity never becomes exactly zero throughout the whole pipe, as it was predicted for large amplitude resonant waves in [1]. In [1] only the pure modes were considered, which must become exact zeroes at some moments of time (modulo the temporal period of the mode). However, those moments of time, as well as the temporal periods, are, in general, different for different modes. In our experiments with arbitrary initial conditions, the system itself, rather than we, chooses which modes to kill and which, to spare. Therefore, several different modes can endure the first two stages. Thus, when we have a few modes at the third stage, at the time when the main, splashing, mode becomes an exact zero, the other modes need not turn to zero at the same time; and the flow never have a chance to come to complete rest.

6.4 Dynamics in Fourier Space

In the previous section we described the time evolution of the lowest pressure and velocity modes, $p_0(t)$ and $v_1(t)$: They quickly start to oscillate with the "limiting" main frequency. By the end of the first stage, the oscillations pick up the "limiting" amplitude and the average level. During the second stage, they maintain their basic parameters. At the same time, many secondary temporal frequency oscillations die out throughout the second stage. The oscillations survived the second

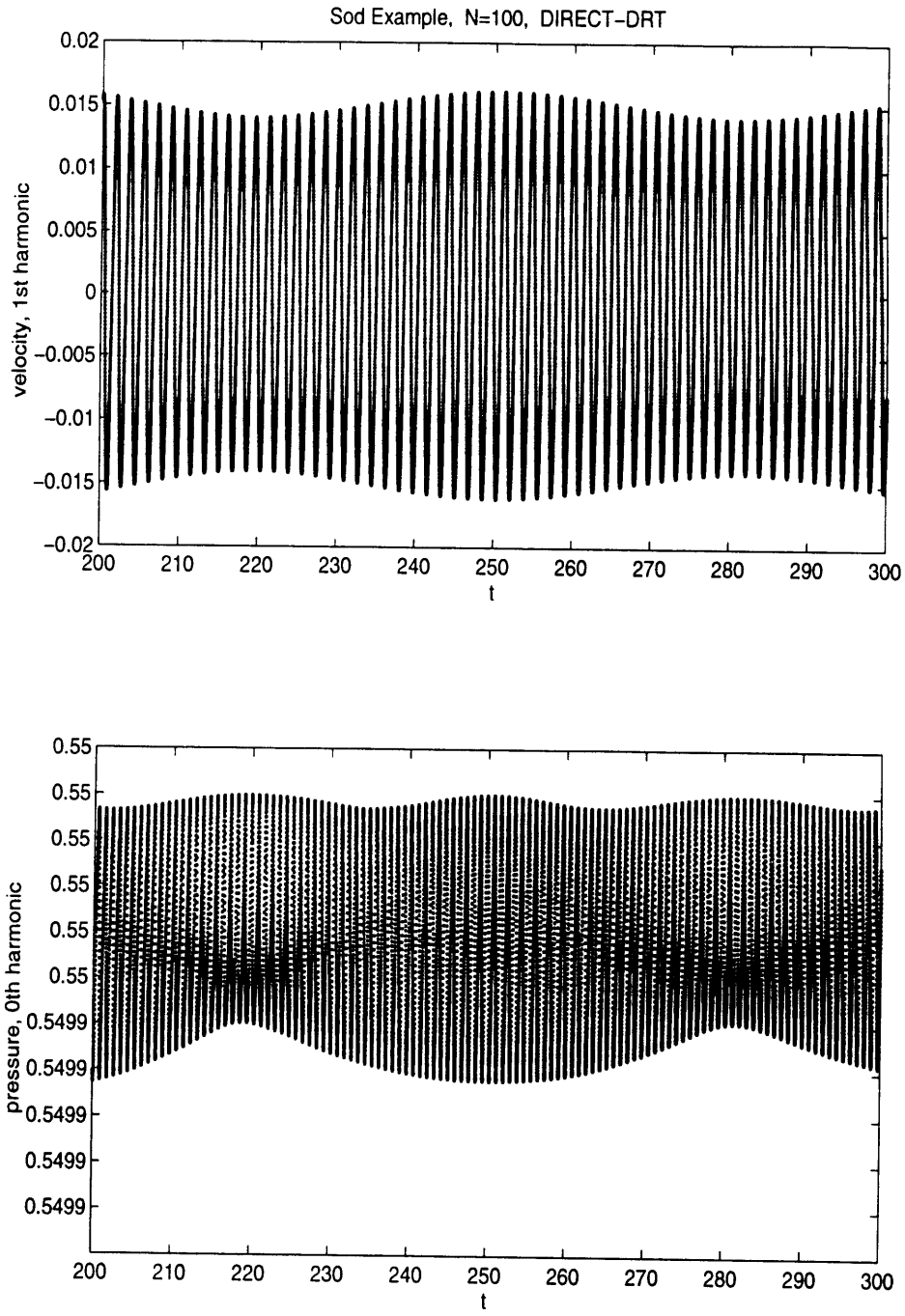

Figure 6.7: The Sod example. The lowest velocity and pressure harmonics.

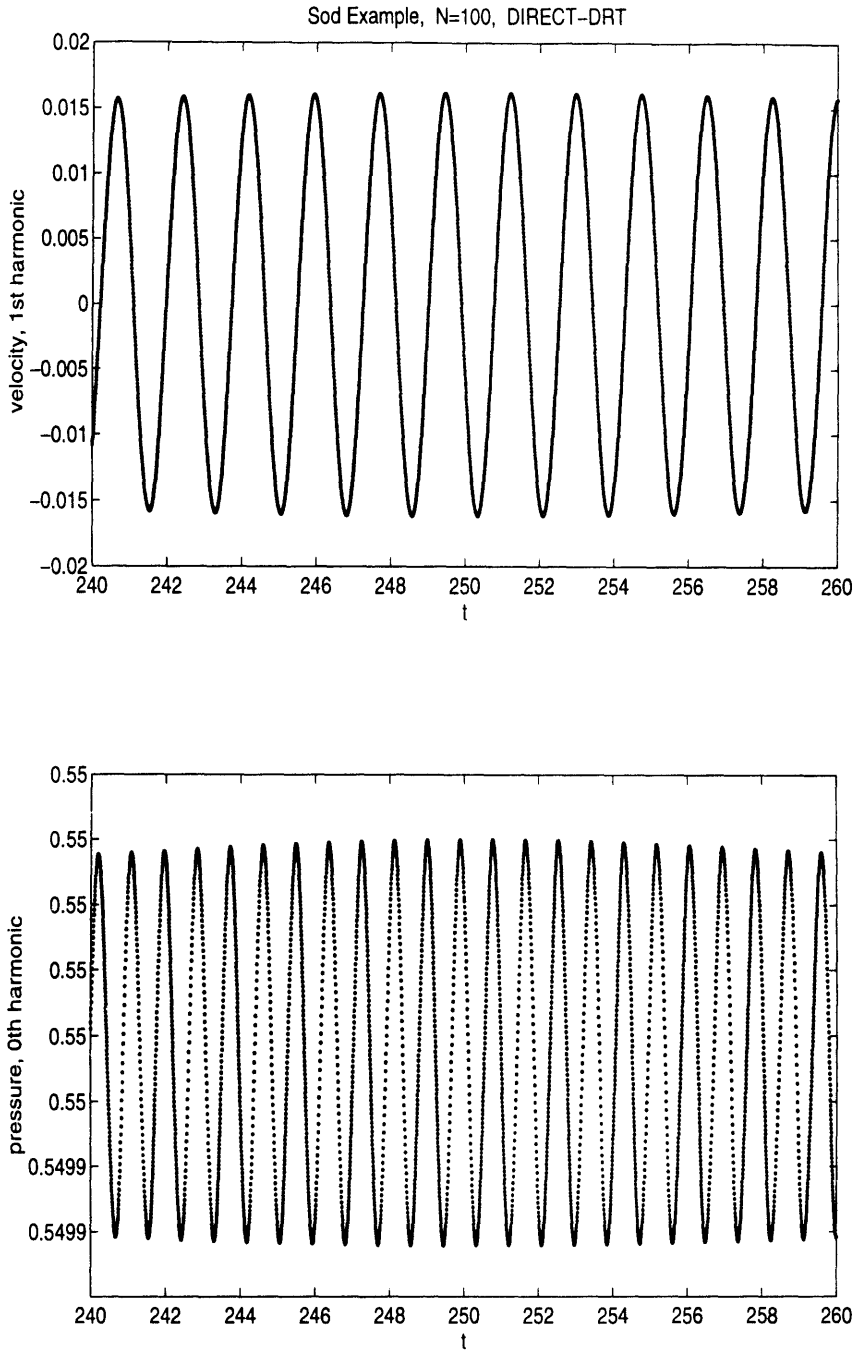

Figure 6.8: The Sod example. The lowest velocity and pressure harmonics. Enlarged.

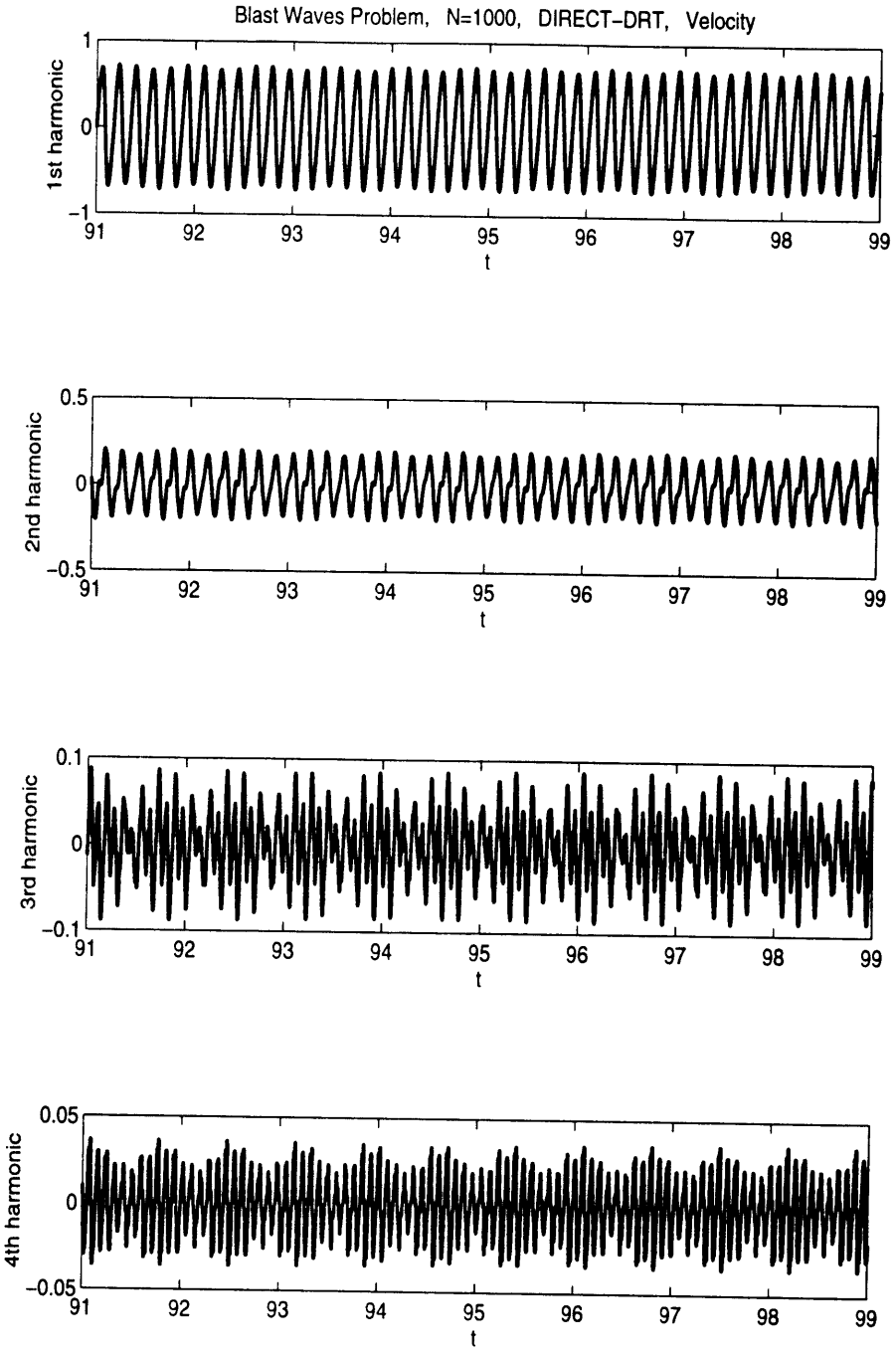

Figure 6.9: The blast waves problem. The first four velocity harmonics.

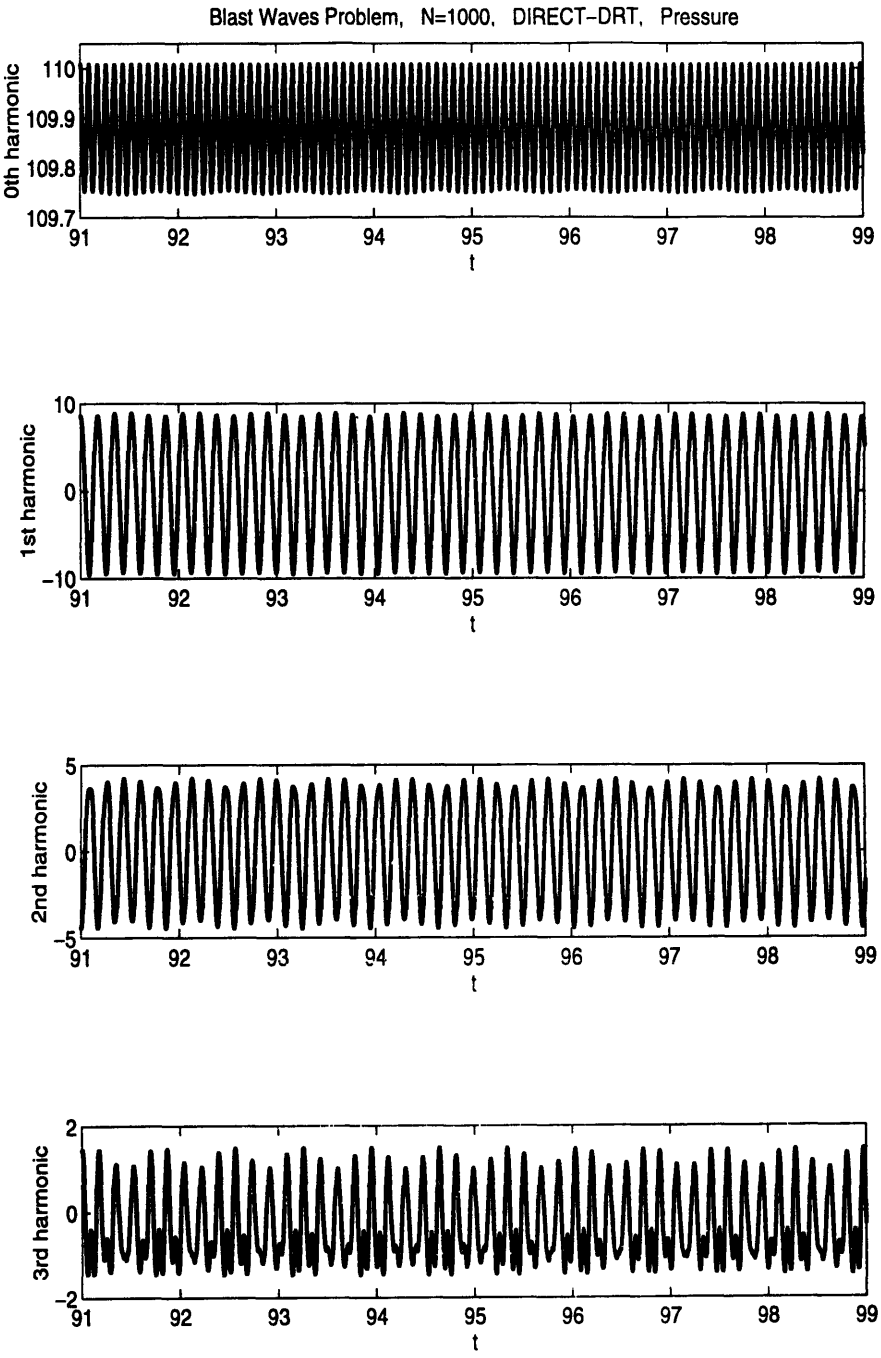

Figure 6.10: The blast waves problem. The first four pressure harmonics.

stage — usually the main frequency plus a couple of secondary frequencies — remain unchanged throughout the third stage.

When we look at the higher modes: $p_1(t), p_2(t), \ldots$ and $v_2(t), v_3(t), \ldots$, we observe pretty much the same picture with the only difference that the higher a mode the more time it takes to reach "limiting" parameters and, after that, to cast off the secondary frequency oscillations. Besides, the "limiting" oscillations of higher modes are usually comprised of more temporal frequencies and display more complicated motion than the lowest mode (see Figures 6.9 and 6.10).

The most remarkable property reveals itself when we look at the simultaneous evolution of several Fourier modes. As we noticed in section 6.2, we can think of the system of hyperbolic partial differential equation as an infinite dimensional system of ordinary differential equations for the Fourier coefficients. Then we can think of the k-space as a *phase space* for our infinite dimensional system of ordinary differential equations. An infinite set of Fourier coefficients of the solution of the system of partial differential equations at a particular time t will be represented by a point at this phase space: $(r_0(t), r_1(t), \ldots, v_1(t), \ldots, p_0(t), p_1(t), \ldots)$ As time evolves, this point will move in the phase space. The curve drawn by this point as it travels through the phase space, called a *trajectory*, represents the solution of the system of partial differential equations in the phase space.

Because the dimension of the phase space is a little bit too large, we decided to consider the projections of the trajectory on the low dimensional subspaces of the phase space. We examined the projections of the trajectory on various three dimensional subspaces: (p_0, p_1, p_2), (p_1, p_2, p_3), $\ldots$, (v_1, v_2, v_3), $\ldots$, (p_2, v_2, v_3), $\ldots$. Regardless of the choice of the subspace, at the third stage we always observed the same results: The projection of the trajectory on a three dimensional subspace was a *two dimensional surface*! The surface had always a shape of a flattened torus. All the motion in a three dimensional subspace was entirely confined to the band: The projection of the trajectory never left the flattened torus (Figures 6.11 and 6.12).

These observations provide strong evidence that the trajectory itself, not only its projections, is confined within a two dimensional manifold. It is hard to believe that a high dimensional object would always have two dimensional projections. Of course, there is always a possibility that we have not chosen the right projection yet — the projection that will reveal the third dimension of the trajectory. However, taking into the consideration a great number of the different subspaces we tried, this possibility seems highly implausible.

Therefore, our conjecture is that the trajectory of the solution corresponding to the resonant standing waves is confined to the two dimensional manifold. These waves appeared to be rather low dimensional objects when considered from the "dynamical system" point of view.

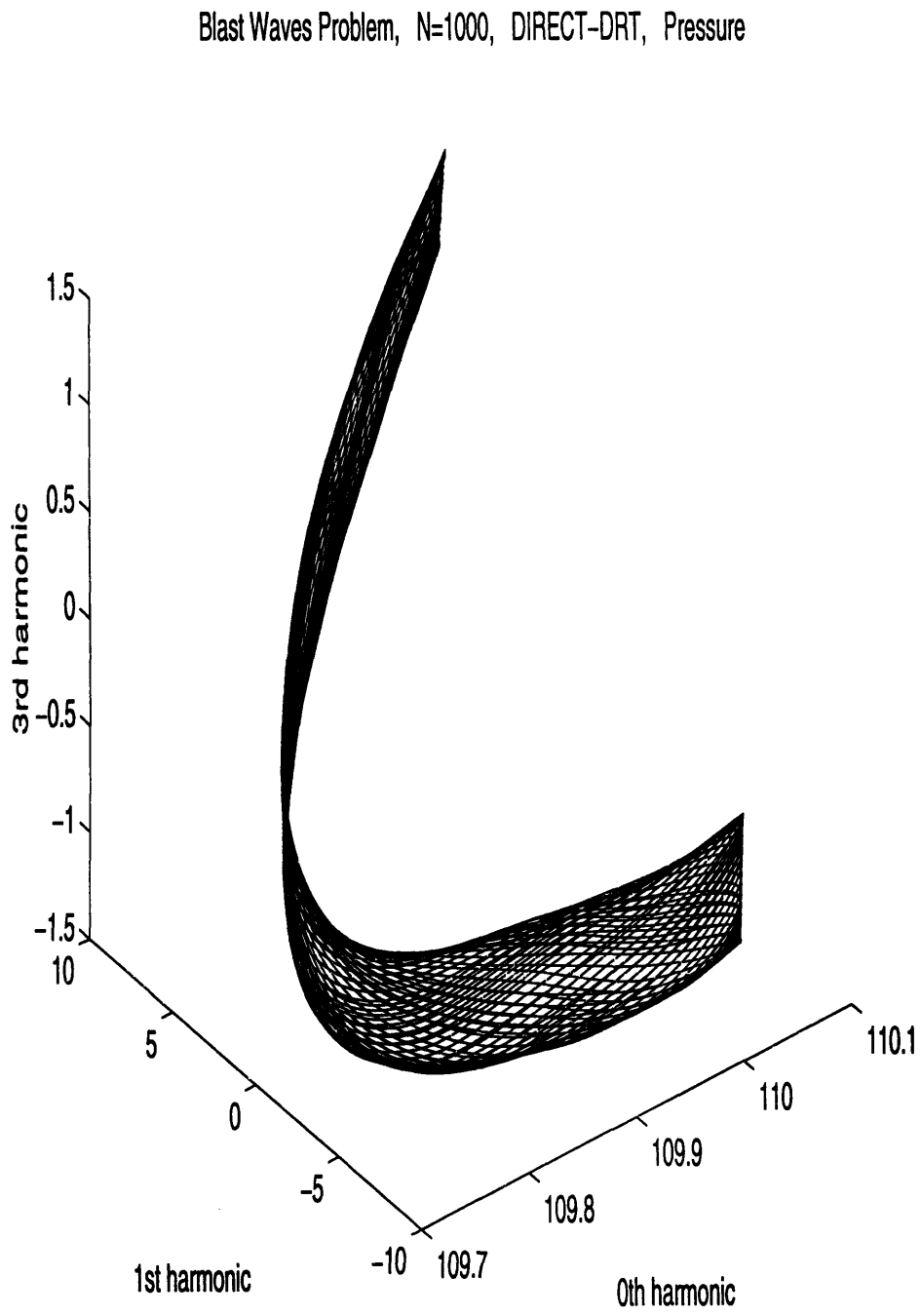

Figure 6.11: The blast waves problem. A three-dimensional projection of the trajectory in Fourier space. Example 1

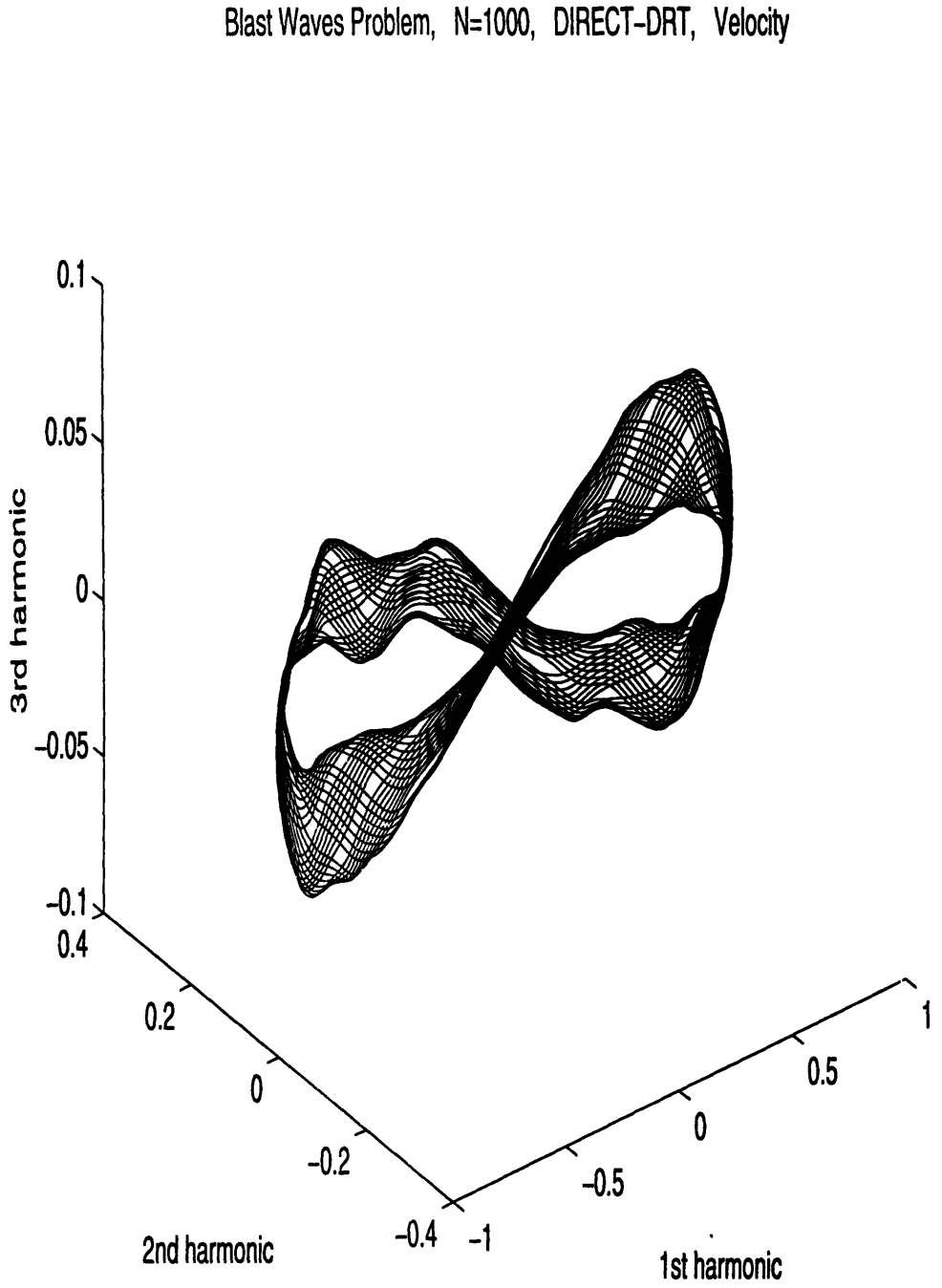

Figure 6.12: The blast waves problem. A three-dimensional projection of the trajectory in Fourier space. Example 2

Chapter 7

Conclusions

A new class of solutions in one-dimensional gas dynamics is found. Remarkably, these solutions do not develop shocks at any time, even though the genuinely nonlinear modes (acoustic) are excited. The solutions are given as finite amplitude, continuous, and (quasi)periodic acoustic standing waves.

These new waves seem to be closely related to the solutions of the Euler equations discovered in [1].

Numerical experiments give substantial evidence that, regardless of the piecewise constant initial conditions, these standing waves will always appear in the flow after all shocks have decayed. It points out at the strongly attractive character of the new solutions — independently of the initial conditions, the flow eventually converges to a non-breaking nonlinear standing wave.

This was just a first look at the phenomenon. We wanted to see whether these waves ever appear. Beyond our expectations, they seem to emerge every time when piecewise initial conditions were used. The mere existence of these ways poses a great number of questions to be answered, not to mention their peculiar properties like low-dimensional behavior in the Fourier space.

A new numerical scheme was developed. The scheme provides a perfect, one-node wide, resolution of strong discontinuities and their interactions. Rarefaction waves are adequately resolved as well. The scheme generates very little numerical noise, and is second order everywhere, including strong discontinuities (L^1-norm second order accurate). The scheme can be easily extended to higher accuracy.

The method can be generalized in two directions. (1) The same way as a strong discontinuity is resolved by, first, detecting it and, then, placing a node on its path, a weak discontinuity can be treated. This will solve the problem of noise production by the edges of rarefaction waves [1]. (2) Extension of the method to two dimensional problems.

[1] Although it is of the next order comparing with the noise generated by a strong discontinuity, some situations are possible when one would need to get rid of it.

Bibliography

[1] C. Celentano. *Finite Amplitude Resonant Acoustic Waves Without Shocks.* PhD dissertation, MIT, Department of Mathematics, January 1995.

[2] P. Colella. A direct Eulerian MUSCL scheme for gas dynamics. *SIAM J. Sci. Stat. Comput.*, 6:104–177, 1985.

[3] P. Colella and P. R. Woodward. The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comput. Phys.*, 54:174–201, 1984.

[4] R. Courant and K. O. Friedrichs. *Supersonic Flow and Shock Waves.* Springer, New York, 1984.

[5] W. E and H. Yang. Numerical study of oscillatory solutions of the gas-dynamic equations. *Stud. Appl. Math.*, 85:29–52, 1991.

[6] S. K. Godunov. A difference scheme for numerical computation of discontinuous solutions of equations of fluid dynamics. *Mat. Sb.*, 47:271–306, 1959.

[7] A. Harten. The artificial compression method for computation of shocks and contact discontinuities: III. Self-adjusting hybrid schemes. *Math. Comput.*, 32:363–389, 1978.

[8] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comp.Phys.*, 49:357–393, 1983.

[9] A. Harten. On a class of high resolution total-variation-stable finite-difference schemes. *SIAM J. Numer. Anal.*, 21:1–23, 1984.

[10] A. Harten. ENO schemes with subcell resolution. *J. Comp.Phys.*, 83:148–184, 1989.

[11] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes,III. *J. Comput. Phys.*, 71:231–303, 1987.

[12] A. Harten and J. M. Hyman. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comp.Phys.*, 50:235–269, 1983.

[13] A. Harten and P. D. Lax. A random choice finite-difference scheme for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 18:289–315, 1981.

[14] A. Harten and S. Osher. Uniformly high-order accurate nonoscillatory schemes. I. *SIAM J. Numer. Anal.*, 24:279–309, 1987.

[15] A. Harten, S. Osher, B. Engquist, and S. Chakravarthy. Some results on uniformly high order accurate essentially non-oscillatory schemes. *Appl. Numer. Math.*, 2:347–377, 1986.

[16] P. D. Lax. *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves.* SIAM, Philadelphia, Pa., 1972.

[17] P. D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237, 1960.

[18] R. LeVeque. *Numerical methods for hyperbolic conservation laws.* Birkhäuser, Basel;Boston;Berlin, 1992.

[19] A. Majda and R. Rosales. Resonantly interacting weakly nonlinear hyperbolic waves. I. A single space variable. *Stud. Appl. Math.*, 71:149–179, 1984.

[20] A. Majda, R. Rosales, and M. Schonbeck. A canonical system of integrodifferential equations arising in resonant nonlinear acoustics. *Stud. Appl. Math.*, 79:205–262, 1988.

[21] K. R. Meadows, D. A. Caughey, and J. Casper. Computing unsteady shock waves for aeroacoustic applications. In *15th Aeroacoustics Conference, AIAA-93-4329*, pages 1–11, Long Beach, CA, October 1993. The American Institute of Aeronautics and Astronautics.

[22] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Comput.*, 38:339–374, 1982.

[23] R. L. Pego. Some explicit resonating waves in weakly nonlinear gas dynamics. *Stud. Appl. Math.*, 79:263–270, 1988.

[24] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial-value Problems.* Wiley-Interscience, 1967.

[25] P. L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *J. Comp.Phys.*, 43:357–372, 1981.

[26] M. Shefter and R. Rosales. Quasi periodic solutions of weakly nonlinear gas dynamics equations. In preparation, 1996.

[27] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.*, 27:1–31, 1978.

[28] B. van Leer. Towards the ultimate conservative difference scheme IV. A new approach to numerical convection. *J. Comput. Phys.*, 23:276–299, 1977.

[29] B. van Leer. Towards the ultimate conservative difference scheme V. A second order sequel to godunov's method. *J. Comput. Phys.*, 32:101–136, 1979.

[30] D. Vaynblat and R. Rosales. Chosing the time step of a godunov-type scheme based on the linear stability analysis. In preparation, 1996.

[31] D. Vaynblat and R. Rosales. Conservative multigrids with compliant interfaces. In preparation, 1996.

[32] D. Vaynblat and R. Rosales. Discontinuity rarefaction treatment scheme. In preparation, 1996.

[33] D. Vaynblat and R. Rosales. Roe flux splitting and absorbing boundary conditions, private communication. In preparation, 1996.

[34] D. Vaynblat and R. Rosales. Second order accurate drt scheme. In preparation, 1996.

[35] P. Woodward and P. Colella. The numerical simulation of two dimensional fluid flow with strong shocks. *J. Comp.Phys.*, 54:115–173, 1984.

[36] P. R. Woodward. Trade-offs in designing explicit hydrodynamics schemes for vector computers. In G. Rodrigue, editor, *Parallel Computation.* Academic Press, New York, 1982.