On Wall Overheating and Other Computational Difficulties of Shock-Capturing Methods

W.H. Hui and S. Kudriakov

Department of Mathematics and Center for Scientific Computation
The Hong Kong University of Science & Technology
Clear Water Bay, Hong Kong
email: whhui@ust.hk

Key words: Lagrangian description; Eulerian description; wall overheating; shock-adaptive Godunov scheme; inviscid compressible flow.

Running head: On Wall Overheating

Subject classifications: 76A02, 76N10, 76N15, 65M99, 65C20.

Proofs should be sent to:

Professor W. H. Hui
Department of Mathematics
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong
email: whhui@ust.hk
fax: 852 2358 1643

Abstract.

Sudden compression of a gas flow due to shock reflection from a solid wall or sudden expansion due to an abrupt withdrawal of a piston from the gas are often associated with a phenomenon known generically as wall overheating. Conventional shock-capturing numerical methods accurately predict pressure and velocity, but over-predict temperature at the wall, hence the name.

We propose a new shock-capturing method that completely eliminates the wall overheating, while correctly predicting the whole flow field. The method uses: **(a)** the Lagrangian coordinates, **(b)** the shock-adaptive Godunov scheme and, **(c)** the entropy conservation equation in the smooth flow region. Furthermore, the new method also overcomes other computational difficulties commonly encountered by conventional shock-capturing methods, namely, the start-up errors, slow-moving shocks, low density flows, and the sonic point glitch. Test examples including one-dimensional unsteady flow and two-dimensional steady flow are given.

1 Introduction.

Sudden compression of a gas flow due to shock reflection from a solid wall or sudden expansion due to an abrupt withdrawal of a piston from the gas are often associated with a phenomenon known generically as wall overheating. Most conventional shock-capturing numerical methods accurately predict pressure and velocity, while under-predicting density and thus over-predicting temperature near the wall, hence the name. More generally, overheating occurs on a contact line (also called a slip line) when the latter changes its direction abruptedly (see Figs.1 and 2) or when it is generated suddenly, as in some Riemann problems or as due to shock-shock interaction. Such a contact line does not need to be a wall but, in general, can be in the interior of the flow field. The phenomenon of overheating is directly related to the properties of a contact line, and should perhaps be more appropriately named contact overheating. Wall overheating occurs because the wall is also a contact line: a boundary contact line. We shall use both names interchangeably in this paper. In essence, the problem of overheating is the problem of correctly capturing the contact line. Overheating is more easily seen and better known in Lagrangian computation which resolves contact sharply; but it can also occur in Eulerian computation (see Figs.4, 13 and 25).

The fundamental properties of a contact line are that the pressure p and velocity u are continuous across it, while the density ρ (and hence also the temperature T and the entropy $S = p/\rho^\gamma$ of a perfect gas obeying the gamma-law) jumps across it. This is to say that pressure and velocity behave regularly near the contact line but all other flow variables behave singularly in that their spatial derivatives are infinite there. These infinite derivatives are difficult for shock-capturing methods to simulate.

Computational results for a sudden compression (the Noh's problem) and for a sudden expansion (due to a piston withdrawing from the gas) are shown in Figs.4,5 and 7 (detailed descriptions of these flows are given in section 5 later). These are computed using Godunov shock-capturing method with the MUSCL update to higher resolution (using a minmod limiter). It is indeed to the credit of Godunov method (and many other shock-capturing

methods) that it is capable of correctly predicting the regular flow variables: the pressure and velocity at the wall. At the same time, in attempting to mimic the infinite derivative of density (hence also of temperature and entropy), Godunov method yields very large gradients near the contact in the form of a density layer (and a temperature layer and entropy layer). These very large gradients lead to uncertainties and hence large errors of density, temperature and entropy at the wall. As the entropy error is always positive (see section 4.1 and Appendix A) while the wall pressure is correctly predicted, the wall density is under-predicted and wall temperature over-predicted, resulting in wall overheating.

The elimination of wall overheating, which can be quite serious, is of central importance for shock-capturing methods. The problem has been studied since von Neumann's pioneering work on simulating shock waves [1] and has been revisited ever since [2]-[11], culminating in the two most recent papers [12],[13]. Many suggestions have been put forward as to the cause of such numerical errors, ranging from the smeared numerical shock profile [8], the entropy layer near the wall and the lack of heat conduction there to dissipate it [5], to the phase error and grid nonuniformity [13], etc. Several special treatments have also been proposed to alleviate it: the method of artificial heat conduction at the wall [5], the flux-splitting method [9], and the "isobaric fix" near the wall as a form of boundary condition [12], etc. While these special treatments can reduce the amount of overheating, they do not eliminate it.

In this paper, we propose a new shock-capturing method that completely eliminates the contact overheating, while correctly predicting all flow variables. The method consists of three components: (a) the use of Lagrangian coordinates, (b) the use of the shock-adaptive Godunov scheme, i.e. using the Riemann solution to fit the shock wave in the overall Godunov shock-capturing method and, (c) the use of the entropy conservation equation in the smooth flow region. The reasons why this method can eliminate contact overheating are as follows: the use of Lagrangian coordinates enables the shock fitting of component (b) to be carried out, while determining the exact shock location and entropy jump across it. This, in turn, frees us from using the Euler equations in conservation form, and the use of the entropy conservation equation in component (c) in the smooth

flow region then yields exact entropy everywhere in the flow field. Finally, this exact entropy, together with the correctly predicted pressure (by any shock-capturing method), correctly predicts the density and hence the temperature, including that at the contact. Test examples including both one-dimensional unsteady flow and two-dimensional steady flow will be shown.

The paper is arranged as follows. In section 2, we first point out the singular nature of the contact overheating problem and then identify the source of overheating — numerical errors of the shock-capturing method — to be the spurious entropy generated near the singular point. In section 3, we discuss the Lagrangian coordinates and the shock-adaptive Godunov scheme. In section 4, we discuss the way to eliminate the spurious entropy generated near the singular point, leading to our proposed numerical method. Test examples for 1-D unsteady flow are given in section 5, whereas those for 2-D steady supersonic flow in section 6. In section 7, we show the new method also overcomes other computational difficulties commonly encountered by conventional shock-capturing methods, namely, the start-up errors, slow-moving shocks, low density flows, and the sonic point glitch. Finally, conclusions are given in section 8.

2 Source of Wall Overheating

2.1 Nature of Overheating

Sudden compression of gas flow may arise when a shock reflects from a solid wall, or when two shocks intersect to produce another two shocks and a contact in between. Sudden expansion of gas flow may, on the other hand, result from an abrupt withdrawal of a piston at finite speed from the gas. All of these are just special examples of flows resulting from the Riemann problem of a shock tube; for instance, the much-studied Noh's problem [5],[13] is a boundary Riemann problem. They are **singular problems** in that the contact line (the wall) changes its direction abruptedly at some point at the boundary or in the

interior of the flow field (Figs.1 and 2), resulting in a geometric singular point 0 there. Since the Riemann solution is the building block for many shock-capturing methods, it is of central importance that all aspects of the Riemann solution, including the density and temperature along the contact line, be captured correctly.

2.2 Source of Overheating

It is, of course, well known that the Riemann problem of a shock tube admits an exact solution. Therefore, to study the contact overheating problem we need first find out why a shock-capturing method cannot numerically re-produce all aspects of the Riemann solution correctly but always under-predicts the density and over-predicts the temperature on the contact line. That is to say, we need first find out the source of the overheating errors.

The Riemann solution consists of three elementary waves: the shock wave, the contact wave and the rarefaction (expansion) wave. They are all centered waves, and the Riemann solution is multi-valued at the geometric singular point and varies rapidly around that point. This causes difficulties in any shock-capturing method where a computational cell is necessarily of finite size, however small it may be. Take the Godunov method as an example, in the cell that contains the singular point there is large variation of flow variables across it, and using cell averages (of the conserved flow variables) to replace the actual flow variables in the cell – which is an essential ingredient of Godunov method – generates large errors, the largest being attained at the wall.

Our method is a modification of the Godunov method, of which the only error arises from cell-averaging of the conserved variables. To eliminate contact overheating, we must avoid the entropy error (spurious entropy) generated due to cell-averaging, especially near the singular point. The motivations for singling out the entropy error as the cause of contact overheating are: (a) Once it is generated near the singular point, it cannot decrease with time while the fluid particle moves along the contact (except it may be smeared in Eulerian computation); otherwise the validity of the numerical method would be questionable. (b) For the special case of a gas with $\gamma = 1$, entropy is directly proportional to

temperature and, hence, avoidance of spurious entropy generation will immediately lead to the elimination of excessive temperature, whence no overheating. (c) An entropy layer near the wall has been observed in all shock-capturing computations that exhibit wall overheating [14]. Indeed, the spurious entropy generated at the singular point and being carried un-diminished by the fluid along the wall, together with the correctly predicted pressure will then under-predict the wall density and hence over-predict the wall temperature. This is the source of overheating that we want to eliminate in this paper; once it is eliminated, the entropy and hence the density and all flow variables will be correctly predicted.

3 Lagrangian Coordinates and the Shock-Adaptive Scheme

3.1 Lagrangian Coordinates versus Eulerian Coordinates

It was pointed out in [13] that the results of computing a shock reflection strongly depend on the chosen coordinate system. In [15], we found that for 1-D unsteady flow Lagrangian coordinates plus the shock-adaptive Godunov scheme gives the best results (for 2-D unsteady flow the situation is quite different, see [16]). They will be discussed here and applied to solve the contact overheating problem.

The Euler equations in Eulerian coordinates for a perfect gas with a gamma law in 1-D flow are

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho e\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ u(\rho e+p)\end{pmatrix}=0 \tag{1}$$

where t is time and x is a space variable. In (1), the specific total energy e is

$$e=\frac{1}{2}u^2+\frac{1}{\gamma-1}\frac{p}{\rho} \tag{2}$$

The Euler equations in Lagrangian coordinates are

$$\frac{\partial}{\partial \lambda}\begin{pmatrix} \tau \\ u \\ e \end{pmatrix} + \frac{\partial}{\partial \xi}\begin{pmatrix} -u \\ p \\ up \end{pmatrix} = 0 \tag{3}$$

where λ is Lagrangian time, ξ is a particle-identifying coordinate and $\tau = 1/\rho$ is the specific volume.

Eqs (1) and (3) are related by the transformation

$$\begin{cases} dt &= d\lambda \\ dx &= ud\lambda + \tau d\xi \end{cases} \tag{4}$$

and they are special cases of the unified coordinates given in [15].

As mentioned earlier, the Riemann solution consists of a shock, a contact and an expansion wave. In the Godunov method using Eulerian coordinates, typically all three waves co-exist within a cell (Fig.3a), whereas at most 2 waves co-exist in a cell if Lagrangian coordinates are used, because the contact always coincides with a cell interface (Fig.3b).

3.2 Shock-Adaptive Godunov Scheme

With the use of Lagrangian coordinates, contact discontinuity can be captured sharply and the only discontinuity we need to resolve is the shock, for which we shall apply the shock-adaptive Godunov scheme introduced in [15],[17],[18]. The Riemann solution provides us with the exact location of the shock wave. The basic idea of the shock-adaptive Godunov scheme consists of splitting a shock-cell, i.e., a computational cell containing a shock wave, along the trajectory of the shock. The splitted shock-cell becomes two sub-cells: one entirely upstream of the shock and the other entirely downstream. In this way, the cell-averaging procedure across the shock discontinuity, and the error associated with it, are avoided, resulting in infinite shock resolution. The fititious cell boundary separating the two sub-cells and moving through the regular grid at the local shock speed shall be

called a partition. With this abstraction, the two sub-cells and the other ordinary cells are treated on an equal footing in the Godunov scheme.

For more details of the shock-adaptive Godunov scheme and its numerical implementation in 1-D flow, see [15] and those in 2-D steady flow, see [17],[18].

We remark that the shock-adaptive scheme is, in effect, doing a shock-fitting in an overall Godunov shock-capturing method. This costs us no extra effort, since the information needed for fitting a shock is already available in the exact Riemann solution. It should also be pointed out that, contrary to popular belief, it takes little time to compute the exact Riemann solution for a perfect gas. Fitting a shock, however, has the important consequence that the Euler equations, (3), need no longer be written in conservation form, allowing us to use equations in other form (see section 4.2). Furthermore, the shock location and the entropy jump across it are exactly determined by the Riemann solution.

We note in passing that an idea similar to that of shock-adaptive scheme was proposed in [19] in conjunction with Eulerian computation, but it was not a complete success. It can handle an isolated shock or an isolated contact, but would have difficulty when both shock and contact are present (see Fig.3a). By contrast, shock-adaptive Godunov scheme is most effective and easy to use in Lagrangian coordinates which take care of the contact resolution already, leaving only the shock to be fitted.

4 Elimination of Wall Overheating

4.1 Spurious Entropy Generation due to Cell-Averaging

In the Godunov scheme, the update of flow variables from one time level to the next consists of two steps: (i) solving Riemann problems for every pair of adjacent cells and (ii) cell-averaging of the conserved variables. The Riemann problems are solved exactly and there is no spurious entropy generation at this step. However, spurious entropy is

generated due to cell-averaging of the conserved variables and is always positive (see Appendix A). This excessive entropy is then carried along the contact un-diminished, resulting in contact overheating as explained in section 2.2.

We note that in Lagrangian computation the contact overheating is always clear and pronounced (Figs.5,7,14,20,22 & 37). In contrast, in Eulerian computation contact overheating may be clear and pronounced (Figs.4,13,16 & 25) but sometimes can be very smeared (Fig.36).

4.2 The Use of Entropy Conservation Equation

We remark that the spurious entropy generated due to cell-averaging can be large for strong expansion or compression, which is the case near the singular point in the sudden expansion (see Figs.1b and 2b) or sudden compression (see Figs.1a and 2a). Furthermore, once generated near the singular point it cannot diminish with time while the fluid particle moves along the contact (it may be smeared in Eulerian computation); otherwise the validity of the numerical method would be in doubt. This, as pointed out in section 2, is the source of contact overheating. Therefore, the only way to eliminate contact overheating is to avoid the generation of spurious entropy due to cell-averaging of the conserved variables near the singular point. This can be done by replacing the energy conservation equation (the last equation in (3)) by the entropy conservation equation

$$\frac{\partial S}{\partial \lambda} = 0 \tag{5}$$

which is valid, and will be used, in the smooth flow region. In Eq.(5), $S = p\tau^{\gamma}$ is a measure of the entropy of the gas. We note in passing that the Euler equations of gas dynamics for 1-D unsteady flow admit four and only four linearly independent conservation laws: conservation of mass, momentum, energy, and entropy. With the use of Eq.(5) the Euler equations in Lagrangian coordinates become

$$\frac{\partial}{\partial \lambda}\begin{pmatrix} \tau \\ u \\ p\tau^{\gamma} \end{pmatrix} + \frac{\partial}{\partial \xi}\begin{pmatrix} -u \\ p \\ 0 \end{pmatrix} = \mathbf{0} \tag{6}$$

which is to be used in the smooth flow region.

To summarize, in our method if a shock is present, it is fitted using information of the exact Riemann solution obtained from solving Eq.(3). The entropy jump across the shock is determined by the Riemann solution, whereas the entropy in the smooth flow region is exactly conserved by solving Eq.(6). Our method therefore gives entropy exactly everywhere in the flow field, including the singular point region. In other words, our method eliminates the spurious entropy. This exact entropy, together with the correctly captured pressure, then correctly predicts the density and hence the temperature, eliminating the contact overheating, while predicting all flow variables correctly. We remark that in the special case of a perfect gas with $\gamma = 1$, a correct entropy value immediately ensures a correct temperature value, whether or not the pressure is correctly predicted. We further note that accuracy in predicting smooth flow can, of course, always be further improved by using high resolution TVD schemes e.g., the MUSCL scheme, as is done in sec. 5 to 7.

5 Test Examples in One-Dimensional Unsteady Flow

In the test examples given in sections 5 and 7, we apply the conventional Godunov-MUSCL shock-capturing method to Eq.(1) based on Eulerian coordinates and to Eq.(3) based on Lagrangian coordinates. We also apply our new method to Eq.(6), as explained at the end of last section, and compare these results with exact solutions.

In all computations, $\gamma = 1.4$ was used except in the Noh's problem, for which $\gamma = 5/3$ was used. Also, in all computations, Godunov method was used with MUSCL update using the minmod limiter to higher resolution, unless otherwise stated. The results are computed with $\triangle x = 0.01$ (Eulerian coordinates) or $\triangle \xi = 0.01$ (Lagrangian coordinates) and the CFL number is 0.3. The numerical solutions in this section are presented in physical space.

Test example #1 is the well-known Noh's problem: the initial condition is a uniform flow with velocity $u = -1$, density $\rho = 1$ and pressure $p = 0$ in a unit domain $0 < x < 1$. This flow is reflected at time $t = 0$ from a solid wall situated at $x = 0$ and, we want to compute the flow after the reflection. Eulerian computations using Eq.(1) are shown in Fig.4, whereas Lagrangian computations using Eq.(3) are shown in Fig.5, both results are compared with the exact solutions. We see from these figures that the conventional Godunov-MUSCL method in Eulerian or Lagrangian coordinates predicts wall pressure and velocity correctly, but it fails to predict the density (and hence also temperature and entropy) correctly near the wall. The failure is worse in the Lagrangian computation (but also see Test example # 7 where the opposite is true).

Fig.6 presents computational results for the same problem using our new method, i.e., Eq.(6) with the shock-adaptive Godunov scheme. The wall overheating is seen completely eliminated; indeed, the exact solution is reproduced by our new method.

Test example #2 is the flow resulting from an abrupt withdrawal, at time $t = 0$, of a piston with a velocity $u = -2$ from the gas at rest (Fig.1b) having pressure $p = 1$ and density $\rho = 1$. If a Eulerian grid is used, this is a moving boundary problem in that the piston position is not fixed with a grid line, and some special treatment must be applied in order to correctly satisfy the boundary condition there (see, for instance, [12]). Such treatment would produce additional errors. In contrast, there is no difficulty in Lagrangian grid, and we shall present only Lagrangian computations.

Fig.7 shows the computed results using conventional Godunov-MUSCL method based on the conservation form equations, (3). The method is seen capable of predicting the wall pressure and velocity (not shown here) correctly, but fails to predict the wall density, temperature and entropy correctly. We note that this failure is due to the inability of conventional shock-capturing methods to correctly simulate the flow near the start of the motion. This flow near the start of the motion is singular in nature, as pointed out in subsection 2.1, and consequently, the error **cannot be reduced** by grid refinement. This is demonstrated in Fig.8, which plots the computed wall entropy layers using four different grid sizes: 1/100, 1/200, 1/400, and 1/800. We see the spurious entropy generated near

the wall remains unchanged with grid refinement. This is expected, because the flow near the start of the motion is a centered (rarefaction) waves which has no length scale.

Using our new method, i.e. shock-adaptive Godunov method applied to Eq.(6), the wall overheating is completely eliminated while the whole flow field is correctly predicted, as seen in Fig.9

The crucial role spurious entropy plays in wall overheating was already pointed out in section 2.1. We now compute the time histories of the flow variables of the wall cell, i.e., the cell adjacent to the piston (Fig.10) for this example. Using the conventional Godunov-MUSCL method, i.e. Eq. (3), the results are shown in Fig.11. There are initial errors in all flow variables, but after some time, pressure approaches its correct value, while density, temperature and entropy all asymptote to wrong values, giving rise to wall overheating. We note, in particular, spurious entropy once generated never decreases following the cell, which is literally a fluid particle. Using our new method of computation, i.e. Eq.(6) with shock-adaptive Godunov scheme, the results are shown in Fig.12 It is seen that although initially there are errors in all flow variables except entropy, after some time they all approach their correct values and there is no wall overheating. The computed entropy is exact for all time, as expected.

Test example #3 is the so-called 1-2-3 problem which consists of two strong rarefaction waves and a stationary contact in the interior of the flow. Although the contact is not a solid wall in this case, overheating still occurs at the contact when computed using conventional shock-capturing methods. Specifically, the initial data are

$$(\rho, u, p) = \begin{cases} (1, 2.5, 0.4) & for \;\; x < 1 \\ (1, -2.5, 0.4) & for \;\; x > 1 \end{cases} \tag{7}$$

Eulerian computations using Eq.(1) are plotted in Fig.13, showing wall overheating. Lagrangian computations using Eq.(3) are shown in Fig.14. It is seen from these figures that serious overheating near the contact occurs with conventional shock-capturing methods, although the absolute error in density is very small.

Using our new method, i.e., adaptive Godunov method applied to Eq.(6), overheating is completely eliminated, as seen from Fig.15. We note that the spurious entropies in

Eulerian and Lagrangian computations (Figs.13 and 14) are negative in the neighbourhood of the contact, which is the result of using MUSCL upgrading. In the first order Godunov method, which is physically correct, the spurious entropy is always positive, as seen in Fig.16. Negative spurious entropy also occurs in other problems when MUSCL upgrading is applied (see Figs.17,18,19), but their magnitudes are quite small.

In **Test example #4**, we consider gradual compression (expansion) of gas flow due to a piston moving into (away from) the gas at rest with pressure $p = 1$ and density $\rho = 1$. The piston path is: $x = t^2/2$ for compression and $x = -t^2/2$ for expansion. We applied Godunov-MUSCL shock-capturing method to Eq.(3), and the results are shown in Figs.17 and 18. In this example, there is no sudden expansion or sudden compression and, hence, no wall overheating. However, there is a small but observable spurious entropy generated at the wall. This is due to the fact that in the numerical computations, the smooth piston path is replaced by line segments, each corner of which causes a sudden (albeit mildly) compression or sudden expansion of the flow. It is this suddenness that generates the spurious entropy. But, unlike examples #1 and #2, the spurious entropy **can be reduced** by grid refinement, as seen in Fig.19 The effect of grid refinement is to reduce the corner angles, leading to the reduction in the spurious entropies generated at these corners. In contrast, if the corner angle is finite, as in examples #1 and #2, the flow problem is a singular one and grid refinement is powerless in reducing spurious entropy, as was demonstrated in Fig.8.

6 Two-Dimensional Steady Supersonic Flow

6.1 Euler Equations in Generalized Lagrangian Coordinates

The Euler equations of gas dynamics for 2-D steady flow using cartesian coordinates (x, y) are

$$\frac{\partial \mathbf{E}_e}{\partial x} + \frac{\partial \mathbf{F}_e}{\partial y} = \mathbf{0} \tag{8}$$

where

$$\begin{aligned} \mathbf{E}_e &= (\rho u, \rho u^2 + p, \rho uv, \rho uH)^T \\ \mathbf{F}_e &= (\rho v, \rho uv, \rho v^2 + p, \rho vH)^T \end{aligned}$$

In (8), u and v are the x- and y-component of fluid velocity, respectively, and H is the specific enthalpy

$$H = \frac{1}{2}q^2 + \frac{\gamma}{\gamma - 1}\frac{p}{\rho}, \quad q = (u^2 + v^2)^{1/2} \tag{9}$$

Under the transformation [20],[21],[16]

$$\left\{ \begin{array}{rcl} dx &=& hud\lambda + Ud\xi \\ dy &=& hvd\lambda + Vd\xi \end{array} \right. \tag{10}$$

where $h(\lambda, \xi)$ is an arbitrary function, the Euler equations become

$$\frac{\partial \mathbf{E}}{\partial \lambda} + \frac{\partial \mathbf{F}}{\partial \xi} = \mathbf{0} \tag{11}$$

where

$$\begin{aligned} \mathbf{E} &= (K, H, Ku + pV, Kv - pU, U, V)^T \\ \mathbf{F} &= h(0, 0, -pv, pu, -u, -v)^T \end{aligned}$$

with $K = \rho(uV - vU)$. The coordinate ξ is clearly a stream function, because along $\xi = const$, Eq.(10) yields

$$\frac{dx}{dy} = \frac{u}{v} \tag{12}$$

which is the equation of a streamline. Thus, the new coordinates (λ, ξ) are streamline coordinates, also called generalized Lagrangian coordinates [21], which possess the advantages of the Lagrangian ones. The coordinate λ is a Lagrangian time if $h = 1$ and a Lagrangian distance if $h = 1/q$. We shall choose

$$h = \frac{1}{q} exp \int^{\xi} \frac{1}{\rho q^2} \frac{\partial p}{\partial \xi} d\xi \tag{13}$$

so that the resulting coordinates (λ, ξ) are orthogonal [20],[21], meaning that $\nabla\lambda \perp \nabla\xi$, which is equivalent to

$$uU + vV = 0 \tag{14}$$

In the orthogonal coordinates the Euler equations (8) simplify to

$$\frac{\partial \mathbf{E}_1}{\partial \lambda} + \frac{\partial \mathbf{F}_1}{\partial \xi} = \mathbf{0} \tag{15}$$

where

$$\begin{aligned} \mathbf{E}_1 &= (K, H, Ku + pV, Kv + pVv/u, V)^T \\ \mathbf{F}_1 &= h(0, 0, -pv, pu, -v)^T \end{aligned}$$

For supersonic flow, both Eqs.(8) and (15) are of hyperbolic type and a space marching approach is appropriate and preferred, with x as a marching variable for Eq.(8) and λ for Eq.(15). It has been shown in [20],[22] that the orthogonal (λ, ξ) are the optimal coordinates for 2-D steady supersonic flow.

Like the 1-D unsteady flow case, the conventional Godunov-MUSCL shock-capturing method will be applied to solve Eq.(8) in Eulerian coordinates and Eq.(15) in Lagrangian coordinates. Furthermore, as in the 1-D unsteady flow case, in conjunction with shock-adaptive Godunov scheme the entropy conservation equation

$$\frac{\partial S}{\partial \lambda} = 0 \tag{16}$$

will be used in the smooth flow region to replace one of the equations of (15), resulting in

$$\frac{\partial \mathbf{E}_2}{\partial \lambda} + \frac{\partial \mathbf{F}_2}{\partial \xi} = \mathbf{0} \tag{17}$$

where

$$\begin{aligned} \mathbf{E}_2 &= (p/\rho^\gamma, K, H, V(\rho q^2 + p), V(\rho q^2 + p)v/u)^T \\ \mathbf{F}_2 &= h(0, 0, 0, -pv, pu)^T \end{aligned}$$

To summarize, in our new method if a shock is present, it is fitted using the exact Riemann solution obtained by solving Eq.(15). The entropy jump across the shock is determined by the Riemann solution, whereas the entropy is exactly conserved by solving Eq.(17) in the smooth flow region. In this way the entropy is computed exactly everywhere in the flow field, including the singular point region. It therefore completely eliminates the contact overheating, while all flow variables are correctly computed.

In this regard, we note with interest that it is actually easier to numerically solve Eq.(17) in the generalized Lagrangian coordinates than Eq.(8) in Eulerian coordinates, because in Eq.(17) the laws of conservation of mass K, enthalpy H and entropy S are satisfied exactly, leaving only two momentum equations to be solved numerically, whereas all four conservation equations in (8) need be solved numerically.

Test examples will now be given to show the superiority of our new method. Here we apply Godunov-MUSCL shock-capturing method to Eq.(8) based on Eulerian coordinates and to Eq.(15) based on Lagrangian coordinates. We also apply our new method to Eq.(17), as explained in the paragraph before the last one, and compare results of these three computations with exact solutions. The results are computed with $\triangle y = 0.01$ (Eulerian coordinates) or $\triangle \xi = 0.01$ (Lagrangian coordinates) and the CFL number is 0.3. The numerical solutions in this section are presented using the self-similar variable y/x.

Test example #5. The wedge flow: a supersonic stream with Mach number $M = 10$ past a wedge with wedge angle of 30^0 (Fig.2a). Computational results using conventional

Godunov-MUSCL method based on Eq.(15) are shown in Fig.20, where it is seen that the pressure p and the normal to the wall component of velocity $\mathbf{v} \cdot \mathbf{n}$ are well captured. These are regular variables as they are continuous across a 2-D slip line (the wall), but the tangential to the wall velocity component $\mathbf{v} \cdot \tau$ and the density (and hence also the temperature T and the entropy S) are singular variables as they jump discontinuously across the 2-D slip line. It is seen from Fig.20 that all these singular variables are incorrectly captured at the wall (a slip line), in particular there is serious wall overheating.

Using our new method based on Eq.(17), all flow variables are accurately captured in the whole flow field as shown in Fig.21. Indeed, our computation reproduces the exact solution in this case.

Test example #6: Prandtl-Meyer expansion flow. Here a supersonic stream of Mach $M = 10$ is turned by an angle of 10^0 (Fig.2b). Computational results using conventional Godunov-MUSCL method based on Eq.(15) are shown in Fig.22, where it is seen that the pressure p and the normal to the wall component of velocity $\mathbf{v} \cdot \mathbf{n}$ are well captured, but the tangential to the wall velocity component $\mathbf{v} \cdot \tau$ and density, temperature and entropy are all incorrectly captured at the wall for the same reasons as given for test example #5 above. Again, the use of our new method accurately captures all flow variables in the whole flow field, as shown in Fig.23.

Test example #7: A 2-D Riemann problem. There are two parallel steady uniform streams, both in the x-direction (Fig.24), with data

$$(p, \rho, M) = \begin{cases} (0.25, 0.5, 12) & for \;\; y > 0 \\ (1, 1, 2.4) & for \;\; y < 0 \end{cases} \tag{18}$$

where M is Mach number. They interact to produce a shock, an expansion wave and a slip line (contact line) in between. Eulerian computational results using conventional Godunov-MUSCL method based on Eq.(8) are shown in Fig.25, where it is seen that the computed pressure p and the normal to the slip line component of velocity $\mathbf{v} \cdot \mathbf{n}$ are acceptable at the slip line, but the tangential (to the slip line) velocity component $\mathbf{v} \cdot \tau$ and density, temperature and entropy are all incorrectly captured, in particular, there

is serious overheating at the slip line, for the same reasons as given for test example #5 above. It should be pointed out that Eulerian computation is seriously flawed: in particular, the large errors in density near the slip line have caused inaccuracies of all flow variables even in the smooth flow region. This re-enforces the importance of having accurate computation of flow discontinuities (as pointed out in [14]), especially when Eulerian coordinates are used.

Lagrangian computational results using conventional Godunov-MUSCL method based on Eq.(15) are shown in Fig.26, where it is also seen that the pressure p and the normal component of velocity $\mathbf{v} \cdot \mathbf{n}$ are well captured, while tangential velocity component $\mathbf{v} \cdot \tau$ and density, temperature and entropy are more accurately captured than Eulerian computation, although mild overheating at the slip line still exists. Comparing Figs.25 and 26, we see that Eulerian computation in every aspect is much worse than Lagrangian computation, in spite of the fact that the Eulerian grid is more regular than the Lagrangian one (Fig.27). This is in contradiction to [13], where grid nonuniformity was regarded as a possible source of wall overheating.

Again, the use of our new method based on Eq.(17) accurately captures all flow variables in the whole flow field, as shown in Fig.28.

In our opinion, this 2-D Riemann problem should be used as a test problem for every 2-D algorithm.

7 Other Computational Difficulties

Most shock-capturing methods experience some special computational difficulties, especially when Eulerian coordinates are used. A list of such difficulties has been compiled in [24],[25] which includes the wall overheating, the start-up errors, slow-moving shocks, low density flows, and the sonic point glitch, etc. We shall now show that our new method not only eliminates wall overheating, but also overcomes all these other difficulties.

7.1 Start-Up Errors

Numerical errors are often observed in using conventional shock-capturing methods to compute the propagation of an isolated shock, especially when the initial data are chosen as a sharp discontinuity. These methods require averaging across a shock, leading to a smeared shock profile and, in the process, generating acoustic and entropy waves. While acoustic waves dissipates quickly the entropy waves remain observable.

Test example #8 is a flow problem, taken from [24], of an isolated shock, initially located at $x = 0.5$, moving at a speed equal to 0.92. Eulerian computations based on Eq.(1) are shown in Fig.29, whereas Lagrangian computations based on Eq.(3) are shown in Fig.30, both using the conventional Godunov-MUSCL shock-capturing method. In both cases, the start-up errors occur in the form of entropy waves in the density profile and in the entropy profile.

In our method, the shock is fitted using exact Riemann solution and is taken to be a cell interface, avoiding cell-averaging across it. Consequently, the start-up errors are avoided, as shown in Fig.31.

7.2 Slow-Moving Shocks

In addition to the start-up errors in computing an isolated shock, if the shock is slow-moving the disturbances due to the shock is of low frequencies (long waves). The numerical dissipation near the shock is small for slow-moving shocks and may not be sufficient to damp the long waves down. In these cases, the strong acoustic and entropy waves generated will give rise to some oscillations in the pressure as well as density. **Test example #9**, from [24], is shown in Fig.32 where the shock speed is 0.11.

In the Lagrangian coordinates the speed of a shock wave is always greater (in magnitude) than the characteristic speed, $a\rho$, and hence the shock cannot be slow-moving. Therefore, the errors due to slow-moving shocks that occur in Eulerian computation will never happen in Lagrangian computation, although the start-up errors may still exist. This is clearly seen by comparing the Lagrangian results in Fig.33 with the Eulerian ones

in Fig.32 for the same flow problem. With our new method, both errors are eliminated, as shown in Fig.34.

7.3 Low Density Flows

A difficulty occurs in some shock-capturing methods with a linearized Riemann solution that use conservation formulation [26]. In this formulation the total energy, density and momentum are the conserved variables, but the pressure p is computed from the total energy after subtracting the kinetic energy determined from the momentum and density. In regions of high speed and low density flow, the kinetic energy dominates the internal energy and inaccuracies in the conserved quantities can easily lead to the kinetic energy exceeding the total energy, causing computational breakdown in the form of pressure being negative. It was shown in [26] that this will not happen with the (first order) Godunov method for the Euler equations but can happen if a linearized Riemann solution is used.

The difficulty of negative pressure appearing will never happen in our method in which, for the smooth flow regions, pressure is determined from the constancy of the entropy (see Eq.(6)) and the conserved quantity τ, and can never become negative. On crossing a shock, pressure is increased as determined by the exact Riemann solution and can not become negative either. Fig.35 shows the expansion flow due to a piston withdrawing at speed equal to 5 from a gas at rest with a unit pressure and density. It was computed using our new method, i.e shock-adaptive Godunov-MUSCL method on Eq.(6). It is seen that even when the flow near the piston is almost vacuum, e.g. the pressure is $p = 1.6 \times 10^{-7}$, the flow is computed without any difficulty.

7.4 The Sonic Point Glitch

In Eulerian computation, many shock-capturing methods experience difficulties in the resolution of rarefaction waves. We consider the problem from [24] with initial data

$$(\rho, u, p) = \begin{cases} (1, 0.75, 1.0) & for \;\; x < 1 \\ (0.125, 0.0, 0.1) & for \;\; x > 1 \end{cases} \tag{19}$$

Fig.36, which uses the first order Godunov method, shows a "glitch" in the rarefaction waves region. This phenomenon is associated with the fact that at the point where the "glitch" situates the flow is sonic, hence the characteristic velocity for the genuinely nonlinear wave field is zero (see [27] for an explanation). In Lagrangian computation, the corresponding characteristic velocities are $a\rho$ and $-a\rho$, and will never become zero. Consequently, the sonic point glitch can never occur in Lagrangian computation, as demonstrated in Fig.37 which was computed using the same Godunov method but based on the Lagrangian equations, Eq.(3). We note that contact overheating is greatly smeared in Eulerian computation (Fig.36), but is more pronounced in Lagrangian computation (Fig.37). The overheating is completely eliminated using our new method based on Eq.(6), as shown in Fig.38.

8 Conclusions

The notorious wall overheating is completely eliminated while all flow variables correctly predicted by the new shock-capturing method which uses: (a) the Lagrangian coordinates, (b) the shock-adaptive Godunov scheme and, (c) the entropy conservation equation in the smooth flow region. In addition, the new method also overcomes other computational difficulties commonly encountered by conventional shock-capturing methods, namely, the start-up errors, slow-moving shocks, low density flows, and the sonic point glitch.

ACKNOWLEDGMENTS

This research was funded by the Research Grants Council of Hong Kong to the senior author. We thank Dr. K. Xu for useful discussions.

References

[1] R.E. Peierls, Theory on von Neumann's Method of Treating Shocks, *Technical Report LA-332(Los Alamos Scientific Laboratory)*, (1945).

[2] R. Landshoff, A Numerical Method for Treating Fluid Flow in the Presence of Shocks, *Technical Report LA-1930(Los Alamos Scientific Laboratory)*, (1955).

[3] I.G. Camerson, An Analysis of the Errors Caused by Using Artificial Viscosity Terms to Represent Steady State Shock Waves, *J. Comput. Phys.*, **1**, 1 (1966).

[4] L.G. Margolin, H.M. Ruppel and R.B. Demuth, Gradient Scaling for a Nonuniform Meshes, in *Numerical Methods for Laminar and Turbulent Flow*, (Pineridge Press, Swansea, UK) (1985).

[5] W.F. Noh, Errors for Calculations of Strong Shocks using an Artificial Viscosity and an Artificial Heat Flux, *J. Comput. Phys.*, **72**, 78 (1987).

[6] P. Glaister, An Approximate Linearized Riemann Solver for the Euler Equations for Real Gases, *J. Comput. Phys.*, **74**, 382 (1988).

[7] R. Sanders and A. Weiser, Higher Resolution Steggered Mesh Approach for Nonlinear Hyperbolic Systems of Conservation Laws, *J. Comput. Phys.*, **101**, 314 (1992).

[8] R. Menikoff, Erros when Shock Waves interact due to Numerical Shock Width, *SIAM J. Sci. Comput.*, **15**, 1227 (1994).

[9] R. Donat and A. Marquina, Capturing Shock Reflections: An Improved Flux Formula, *J. Comput. Phys.*, **125**, 42 (1996).

[10] M. Gehmeyr, B. Cheng and D. Mihalas, Noh's Constant-Velocity Shock Problem Revisited, *Shock Waves*, **7**, 255 (1997).

[11] E.J. Caramana, M.J. Shashkov and P.P. Whalen, Formulations of Artificial Voscosity for Multi-Dimensional Shock Wave Computations, *J. Comput. Phys.*, **144**, 70 (1998).

[12] R.P. Fedkiw, A. Marquina and R. Merriman, An Isobaric Fix for the Overheating Problem in Multimaterial Compressible Flows, *J. Comput. Phys.*, **142**, 545 (1999).

[13] W.J. Rider, Revisiting Wall Heating, *J. Comput. Phys.*, **162**, 395 (2000).

[14] P. Woodward and P. Colella, The Numerical Simulation of Two- Dimensional Fluid Flow with Strong Shocks, *J. Comput. Phys.*, **54**, 115 (1984).

[15] W.H. Hui and S. Koudriakov, The role of Coordinates in the Computation of Discontinuities in One-Dimensional Flow, *Comput. Fluid Dyn. J.*, **8**, 495 (2000).

[16] W.H. Hui, P.Y. Li and Z.W. Li, A Unified Coordinate System for Solving the Two-Dimensional Euler Equations, *J. Comput. Phys.*, **153**, 596 (1999).

[17] W.H. Hui and C.Y. Loh, A New Lagrangian Method for Steady Supersonic Flow Computation, Part III: Strong Shocks, *J. Comput. Phys.*, **103**, 465 (1992).

[18] C.Y. Lapage and W.H. Hui, A Shock-Adaptive Godunov Scheme Based on the Generalized Lagrangian Formulation, *J. Comput. Phys.*, **122**, 291 (1995).

[19] A. Harten and J.M. Hyman, Self Adjusting Grid Methods for One-Dimensional Hyperbolic Conservation Laws, *J. Comput. Phys.*, **50**, 235 (1983).

[20] W.H. Hui and D.L. Chu, Optimal Grid for the Steady Euler Equations, *Comput Fluid Dyn. J.*, **4**, 403 (1996).

[21] W.H. Hui, Generalized Lagrangian Formulation of Computational Fluid Dynamcs, in *Comput. Fluid Dyn. Review–1995*, Wiley, 382 (1995).

[22] W.H. Hui and Y. He, Hyperbolicity and Optimal Coordinates for Three-Dimensional Steady Supersonic Flow, *SIAM J. Appl. Math.*, **57**, 893 (1997).

[23] H.M. Glaz and A.B. Wardlaw, A High-Order Godunov Scheme for Steady Supersonic Gas Dynamics, *J. Comput. Phys.*, **58**, 157 (1985).

[24] R.J. LeVeque, Nonlinear Conservation Laws and Finite Volume Methods for Astrophysical Fluid Flow, in *Computational Methods for Astrophysical Fluid Flow*, edited by O. Steiner and A. Gautschy, Springer-Verlag, 1998).

[25] J.J. Quirk, A Contribution to the Great Riemann Solver Debate, *Int'l J. Num. Mehtods in Fluids*, **18**, 555 (1994).

[26] B. Einfeldt, C.D. Munz, P.L. Roe and B. Sjogreen, On Godunov-Type Methods near Low Densities, *J. Comput. Phys.*, **92**, 273 (1991).

[27] K. Xu, An Explanation for the Sonic Point Glitch, to be published.

A Appendix A. Spurious Entropy Generation due to Cell-Averaging of Conserved Variables.

To be specific we consider Lagrangian coordinates where the Euler equations in conservation form for a perfect gas with a γ-law are

$$\frac{\partial}{\partial\lambda}\begin{pmatrix}\tau\\ u\\ e\end{pmatrix}+\frac{\partial}{\partial\xi}\begin{pmatrix}-u\\ p\\ up\end{pmatrix}=0 \tag{A.1}$$

where λ is Lagrangian time, ξ is a particle-identifying coordinate. In Eq.(A.1), u is gas velocity, $\tau=1/\rho$ is the specific volume and e is the specific total energy

$$e=\frac{1}{2}u^2+\frac{1}{\gamma-1}p\tau \tag{A.2}$$

In this formulation, the conserved variables are (τ,u,e), but the entropy S per unit mass

$$S=p\tau^{\gamma} \tag{A.3}$$

is not a conserved variable.

In the Godunov scheme, the cell-averaging is applied to the conserved variables (using the exact Riemann solution) in order to ensure the laws of conservation of mass, momentum and energy are satisfied. But it is not applied to non-conserved variables, e.g. pressure p, entropy S and temperature T, etc. The averages of the latter variables, (e.g. pressure) are, however, needed in order to solve the Riemann problems at the next time level, and they are obtained from those of the conserved variables by a decoding process. (If they were obtained by cell-averaging using the exact Riemann solution, as was done for the conserved variables, that would imply the physical conservation laws are not satisfied).

The cell-averages of the conserved variables, denoted with an over bar, i.e. $(\overline{\tau},\overline{u},\overline{e})$, are obtained exactly from the finite volume (integral form) method applied to (A.1). On the other hand, the so-called "cell-average" of the entropy, $\overline{S}$, is then given by

$$\overline{S}=\overline{p}\cdot\overline{\tau}^{\gamma} \tag{A.4}$$

where $\overline{p}$, the so-called "cell-average" of pressure, is in turn obtained from the averages of the conserved variables, $(\overline{\tau}, \overline{u}, \overline{e})$, by decoding as follows:

$$\overline{e} = \overline{\frac{1}{2}u^2 + \frac{1}{\gamma - 1}p\tau} = \frac{1}{2}\overline{u^2} + \frac{1}{\gamma - 1}\overline{p\tau} \tag{A.5}$$

Since only $(\overline{\tau}, \overline{u}, \overline{e})$ are known, Eq.(A.5) is replaced, incorrectly, by

$$\overline{e} = \frac{1}{2}\overline{u}^2 + \frac{1}{\gamma - 1}\overline{p} \cdot \overline{\tau} \tag{A.6}$$

so that

$$\overline{p} = \frac{\gamma - 1}{\overline{\tau}}\left(\overline{e} - \frac{1}{2}\overline{u}^2\right) \tag{A.7}$$

In this numerical process of the Godunov scheme, spurious entropy is generated. For instance, in the special case when the cell contains only a rarefaction wave as given by the Riemann solution the entropy is uniform, S_0, across the cell and hence its averaged value, $\overline{S}_{before}$, is also equal to S_0. But after cell-averaging of the conserved variables, the so-called "cell-averaged" entropy, $\overline{S}_{after}$

$$\begin{cases} \overline{S}_{after} &= \overline{p} \cdot \overline{\tau}^{\gamma} \\ \overline{p} &= \frac{\gamma - 1}{\overline{\tau}}\left(\overline{e} - \frac{1}{2}\overline{u}^2\right) \end{cases} \tag{A.8}$$

is no longer the same as $\overline{S}_{before}$. Their difference, $(\overline{S}_{after} - \overline{S}_{before})$, is the spurious entropy generated due to cell-averaging of the conserved variables, and we want to find out if this difference is always positive or not.

As the Riemann solution consists of three elementary waves: a rarefaction, a shock and a contact (also called a slip line), these waves also appear in a computational cell. In general, all three waves appear within one cell in Eulerian coordinates (Fig.3a), but only the rarefaction and the shock can appear within one cell in Lagrangian coordinates, (Fig.3b), because the contact always coincides with a cell interface of a Lagrangian cell. These elementary waves will not intersect with each other within the cell because of the requirement of the CFL condition. We therefore can consider spurious entropy generation due to cell-averaging across these waves separately.

A.1 Spurious Entropy Generation due to Cell-Averaging Across a Rarefaction Wave (Fig.39)

The flow inside a rarefaction wave is given by (see e.g., [15])

$$\begin{cases} p \cdot \tau^{\gamma} = const. = S_0 \\ u - \frac{2c}{\gamma-1} = const. = U_0 \quad (ah \le \xi \le bh) \\ \frac{\xi}{\lambda} = \frac{c}{\tau} \end{cases} \tag{A.9}$$

where $c = (\gamma p \tau)^{\frac{1}{2}}$. From these, we get

$$\tau = \left(\frac{\xi}{f}\right)^{-m}, \quad m = \frac{2}{\gamma+1}, \quad f = k(\gamma S_0)^{\frac{1}{2}} \tag{A.10}$$

$$u = U_0 + \frac{2}{\gamma-1}(\gamma S_0)^{\frac{1}{2}} \left(\frac{\xi}{f}\right)^{\kappa}, \quad \kappa = \frac{\gamma-1}{\gamma+1} = 1 - m \tag{A.11}$$

We want to find the cell-averaged values of the conserved variables (τ, u, e) across a cell of width h at time $\lambda = k$ with the rarefaction wave situating in $ah \le \xi \le bh$, $(0 \le a < b \le 1)$. The cell-average of τ is

$$\overline{\tau} = \frac{1}{h}\int_0^h \tau d\xi = \frac{1}{h}\left[\int_0^{ah} \tau d\xi + \int_{ah}^{bh} \tau d\xi + \int_{bh}^{h} \tau d\xi\right] = a\tau_a + (1-b)\tau_b + \frac{1}{h}\int_{ah}^{bh} \tau d\xi \tag{A.12}$$

where $(.)_a$ denotes $(.)$ evaluated at $\xi = ah$, and similarly for $(.)_b$, thus

$$\tau_a = g a^{-m}, \quad \tau_b = g b^{-m}, \quad g = \left(\frac{h}{k(\gamma S_0)^{1/2}}\right)^{-m} \tag{A.13}$$

Using (A.10) and (A.13) in (A.12), we get

$$\overline{\tau} = g\left[b^{\kappa-1} + \frac{2}{\gamma-1}(b^{\kappa} - a^{\kappa})\right] \tag{A.14}$$

Similarly, we get

$$\overline{u} = U_0 + \frac{2}{\gamma-1}(\gamma S_0)^{\frac{1}{2}} g^{-\kappa/m}\left[b^{\kappa} - \frac{\gamma-1}{2\gamma}(b^{\kappa+1} - a^{\kappa+1})\right] \tag{A.15}$$

and

$$\overline{e} = ae_a + (1-b)e_b + \frac{1}{h}\int_{ah}^{bh} e d\xi \tag{A.16}$$

In the last integral,

$$\begin{aligned} e &= \frac{1}{2}u^2 + \frac{1}{\gamma-1}p\tau = \frac{1}{2}u^2 + \frac{S_0}{\gamma-1}\frac{1}{\tau^{\gamma-1}} \\ &= \frac{1}{2}U_0^2 + \frac{2U_0}{\gamma-1}(\gamma S_0)^{\frac{1}{2}}\left(\frac{\xi}{f}\right)^{\kappa} + \frac{3\gamma-1}{(\gamma-1)^2}S_0\left(\frac{\xi}{f}\right)^{2\kappa} \end{aligned} \tag{A.17}$$

Substituting (A.17) into (A.16) and integrating, we get

$$\begin{aligned} \overline{e} &= \frac{1}{2}U_0^2 + \frac{2U_0}{\gamma-1}(\gamma S_0)^{\frac{1}{2}}g^{-\kappa/m}\left[b^{\kappa} - \frac{\gamma-1}{2\gamma}(b^{\kappa+1} - a^{\kappa+1})\right] \\ &+ \frac{(3\gamma-1)U_0}{(\gamma-1)^2}g^{-2\kappa/m}\left[b^{2\kappa} - \frac{2(\gamma-1)}{3\gamma-1}(b^{2\kappa+1} - a^{2\kappa+1})\right] \end{aligned} \tag{A.18}$$

Formulae (A.14), (A.15) and (A.18) give the required cell-averaged values of the conserved variables. Using them in (A.8) we obtain the so-called "cell-averaged" value of entropy.

$$\begin{aligned} \overline{S}_{after} &= S_0\left[b^{\kappa-1} + \frac{2}{\gamma-1}(b^{\kappa} - a^{\kappa})\right]^{\gamma-1} \times \\ &\left[\frac{3\gamma-1}{\gamma-1}b^{2\kappa} - 2(b^{2\kappa+1} - a^{2\kappa+1}) - \frac{2\gamma}{\gamma-1}\{b^{\kappa} - \frac{\gamma-1}{2\gamma}(b^{\kappa+1} - a^{\kappa+1})\}^2\right] \end{aligned} \tag{A.19}$$

Finally, we get the percentage spurious entropy generation, S_r, due to cell-averaging of the conserved variables across a rarefaction wave in a Lagrangian cell

$$\begin{aligned} S_r &\equiv \frac{\overline{S}_{after} - \overline{S}_{before}}{\overline{S}_{before}} = -1 + \left[b^{\kappa-1} + \frac{2}{\gamma-1}(b^{\kappa} - a^{\kappa})\right]^{\gamma-1} \times \\ &\left[\frac{3\gamma-1}{\gamma-1}b^{2\kappa} - 2(b^{2\kappa+1} - a^{2\kappa+1}) - \frac{2\gamma}{\gamma-1}\left\{b^{\kappa} - \frac{\gamma-1}{2\gamma}(b^{\kappa+1} - a^{\kappa+1})\right\}^2\right] \end{aligned} \tag{A.20}$$

(The subscript r denotes rarefaction wave). We note that for weak rarefaction waves, $b \to a$, we get $S_r = 0$ from (A.20), as expected. We further note with interest that for a gas with $\gamma = 1$, we also get $S_r = 0$ from (A.20).

A.2 Spurious Entropy Generation due to Cell-Averaging Across a Shock Wave (Fig.40)

From [15], the flow across the shock wave is given by

$$\begin{cases} \tau_2 &= \tau_1 \left(\dfrac{1+\kappa\alpha}{\kappa+\alpha} \right) \\ u_2 &= u_1 - (\alpha - 1) \left[\dfrac{(1-\kappa)p_1\tau_1}{\kappa+\alpha} \right]^{\frac{1}{2}} \end{cases} \tag{A.21}$$

where $\alpha = \dfrac{p_2}{p_1}$ represents the strength of the shock. In this case

$$\begin{cases} \overline{S}_{before} &= (1-a)S_1 + aS_2 \\ S_1 &= p_1\tau_1^{\gamma} \ and \ S_2 = p_2\tau_2^{\gamma} \end{cases} \tag{A.22}$$

Hence

$$\overline{S}_{before} = S_1 \left[1 - a + a\alpha \left(\frac{1+\kappa\alpha}{\kappa+\alpha} \right)^{\gamma} \right] \tag{A.23}$$

We want to find the cell-averaged values of the conserved variables, $(\overline{\tau}, \overline{u}, \overline{e})$, across a cell of width h at time $\lambda = k$ with the shock wave located at $\xi = ah$, $(0 < a \leq 1)$. The averaged value of τ across the cell is

$$\overline{\tau} = \frac{1}{h}\int_0^h \tau d\xi = \frac{1}{h}\left[\int_0^{ah} \tau d\xi + \int_{ah}^h \tau d\xi \right] = (1-a)\tau_1 + a\tau_2 \tag{A.24}$$

Similarly,

$$\overline{u} = (1-a)u_1 + au_2 \tag{A.25}$$

$$\overline{e} = (1-a)e_1 + ae_2 = (1-a)\left[\frac{1}{2}u_1^2 + \frac{1}{\gamma-1}p_1\tau_1 \right] + a\left[\frac{1}{2}u_2^2 + \frac{1}{\gamma-1}p_2\tau_2 \right] \tag{A.26}$$

Formulae (A.24) to (A.26) give the required cell-averaged values of the conserved variables. Using them in (A.8), we obtain the so-called "cell-averaged" value of entropy

$$\begin{aligned}\overline{S}_{after} &= \frac{p_1\tau_1^{\gamma}}{(\kappa+\alpha)^{\gamma}}\left[(1-a)(\kappa+\alpha)+a(1+\kappa\alpha)\right]^{\gamma-1}\times \\ &\quad \left[(1-a)\left\{\kappa+\alpha+a\kappa(\alpha-1)^2\right\}+a\alpha(1+\kappa\alpha)\right] \qquad \text{(A.27)}\end{aligned}$$

Finally, we get the percentage spurious entropy generation, S_s, due to cell-averaging of the conserved variables across a shock wave in a Lagrangian cell

$$\begin{aligned}S_s &\equiv \frac{\overline{S}_{after}-\overline{S}_{before}}{\overline{S}_{before}} = -1+\left[(1-a)\{\kappa+\alpha+a\kappa(\alpha-1)^2\}+a\alpha(1+\kappa\alpha)\right]\times \\ &\quad \left[(1-a)(\kappa+\alpha)+a(1+\kappa\alpha)\right]^{\gamma-1}/\left[(1-a)(\kappa+\alpha)^{\gamma}+a\alpha(1+\kappa\alpha)^{\gamma}\right] \qquad \text{(A.28)}\end{aligned}$$

We note that for weak shock waves, $\alpha \to 1$, we get $S_s = 0$ from (A.28), as expected. We further note with interest that for a gas with $\gamma = 1$, we also get $S_s = 0$ from (A.28).

A.3 Discussions

In the general case, where both rarefaction wave and shock wave are present in one cell, (the contact wave always coincides with a cell interface in Lagrangian computation) the time step k must be small enough (CFL condition) that they do not intersect with each other. Consequently, the total spurious entropy generation due to cell-averaging of the conserved variables across a Lagrangian cell is the sum of those generated due to averaging across the rarefaction wave and those across the shock wave.

For $\gamma = 5/3$, 1.4 and 1.2, S_r is plotted in Fig.41 against the strength of the rarefaction wave, $(b-a)$, for various wave front positions b. It is seen to be positive in all cases for $0 < a < b < 1$. Likewise, for the same γ values, S_s is plotted in Fig.42 against the shock strength, $log_{10}\frac{p_2}{p_1}$, for various shock positions a. It is also seen to be positive in all cases for $0 < a < 1$ and $log_{10}\frac{p_2}{p_1} > 0$. We therefore conclude that **the total spurious entropy generation due to cell-averaging of the conserved variables across a Lagrangian**

cell is always positive. Although a formula for the spurious entropy generated due to cell-averaging of the conserved variables across a contact can also be easily given and proved, mathematically as well as numerically, to be negative for all cases, it does not alter this conclusion. This is because Lagrangian computation avoids averaging across a contact altogether, so the formula is not used (nor is it given here).

Detailed analysis can also be given, in Eulerian coordinates, for the spurious entropy generated across each of three elementary waves, namely, the rarefaction, the shock and the contact, and they are all positive. We therefore also conclude that **the total spurious entropy generated due to cell-averaging of the conserved variables across an Eulerian cell is always positive**.

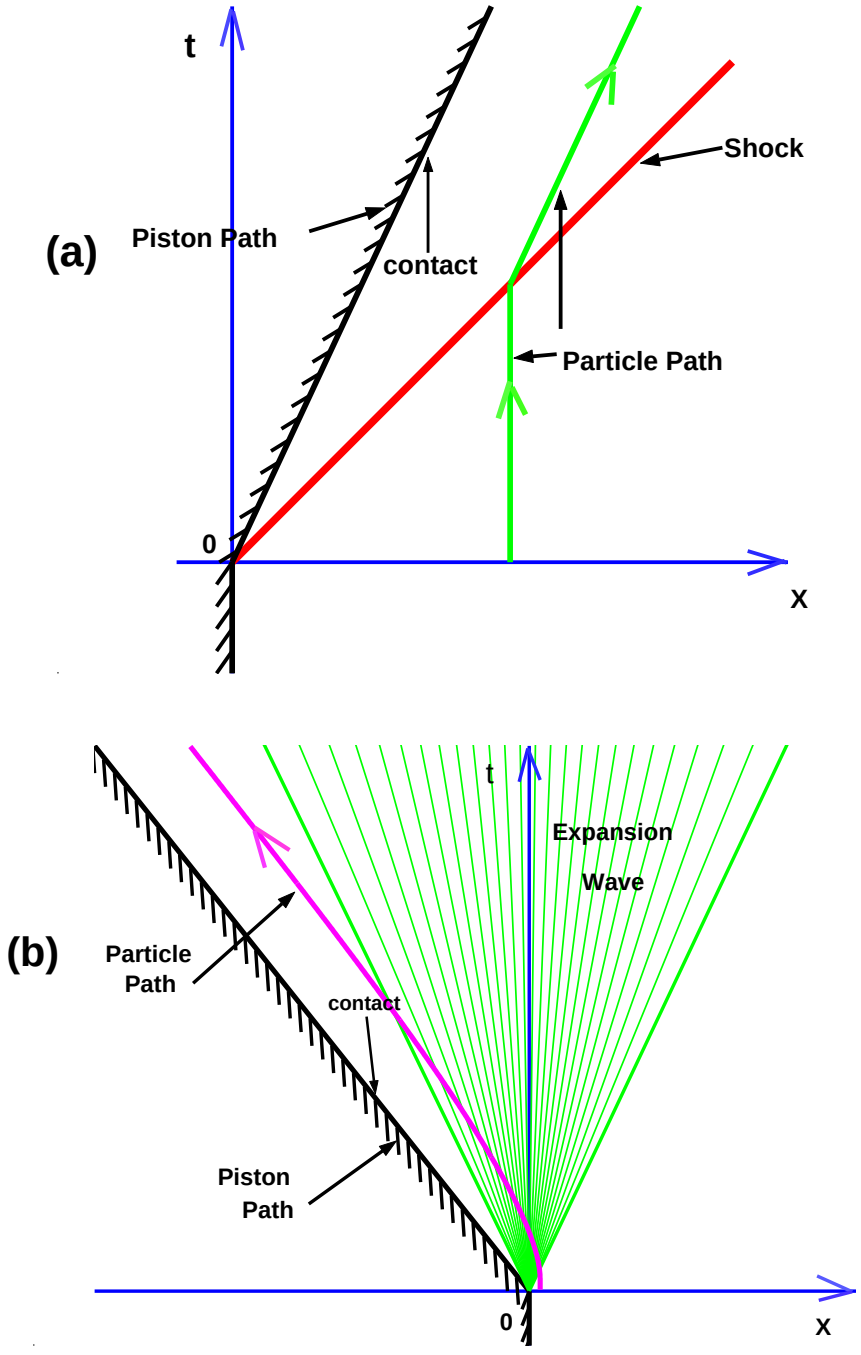

Figure 1: Abrupt change of direction of a contact in 1-D unsteady flow: **(a)** Sudden compression, **(b)** Sudden expansion.

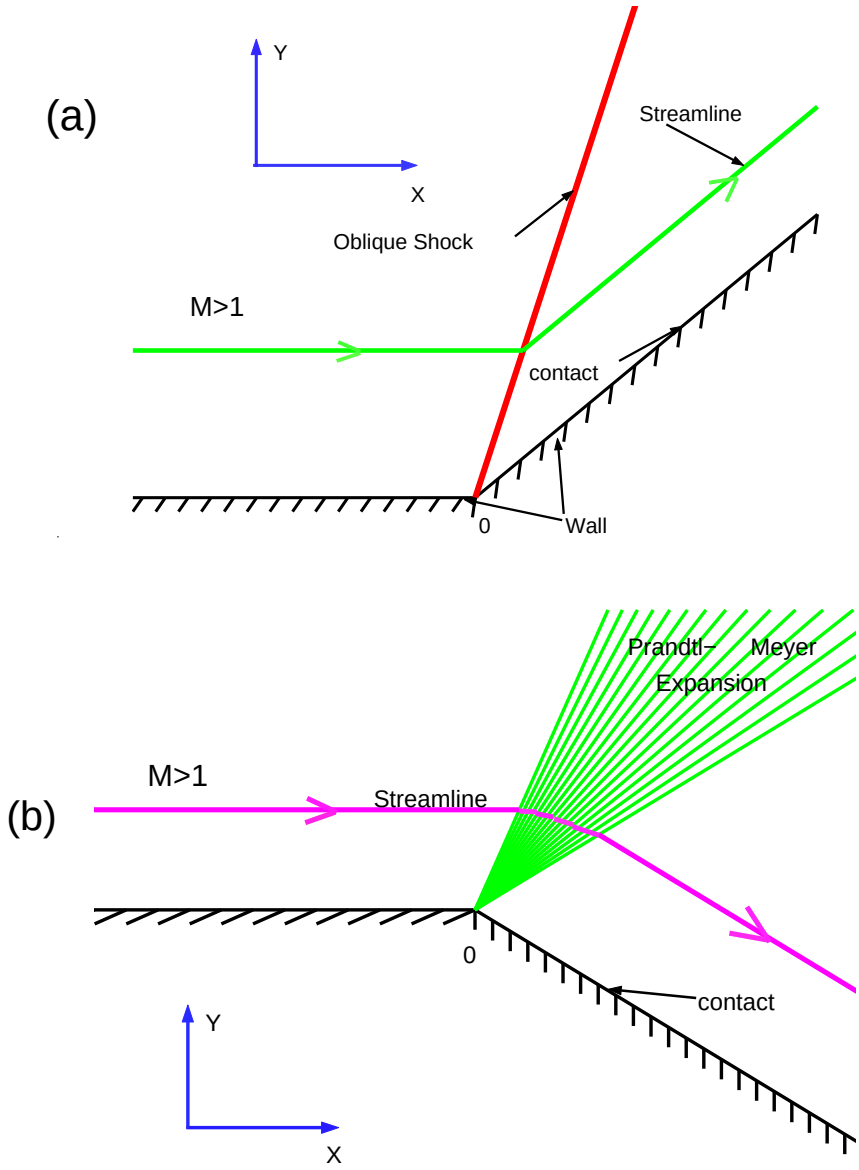

Figure 2: Abrupt change of direction of a contact in 2-D steady supersonic flow: (**a**) Sudden compression, (**b**) Sudden expansion.

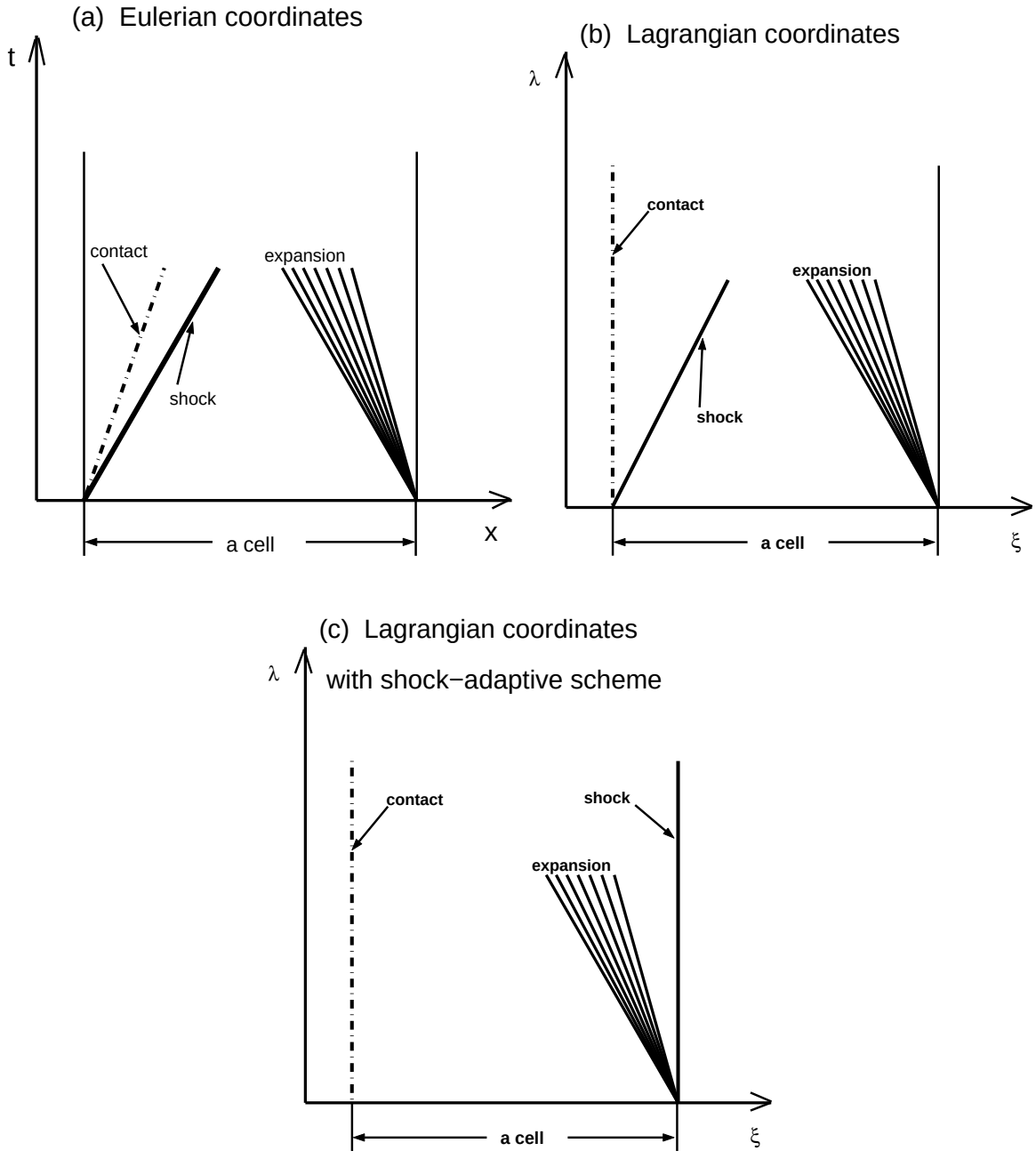

Figure 3: Elementary waves in a cell.

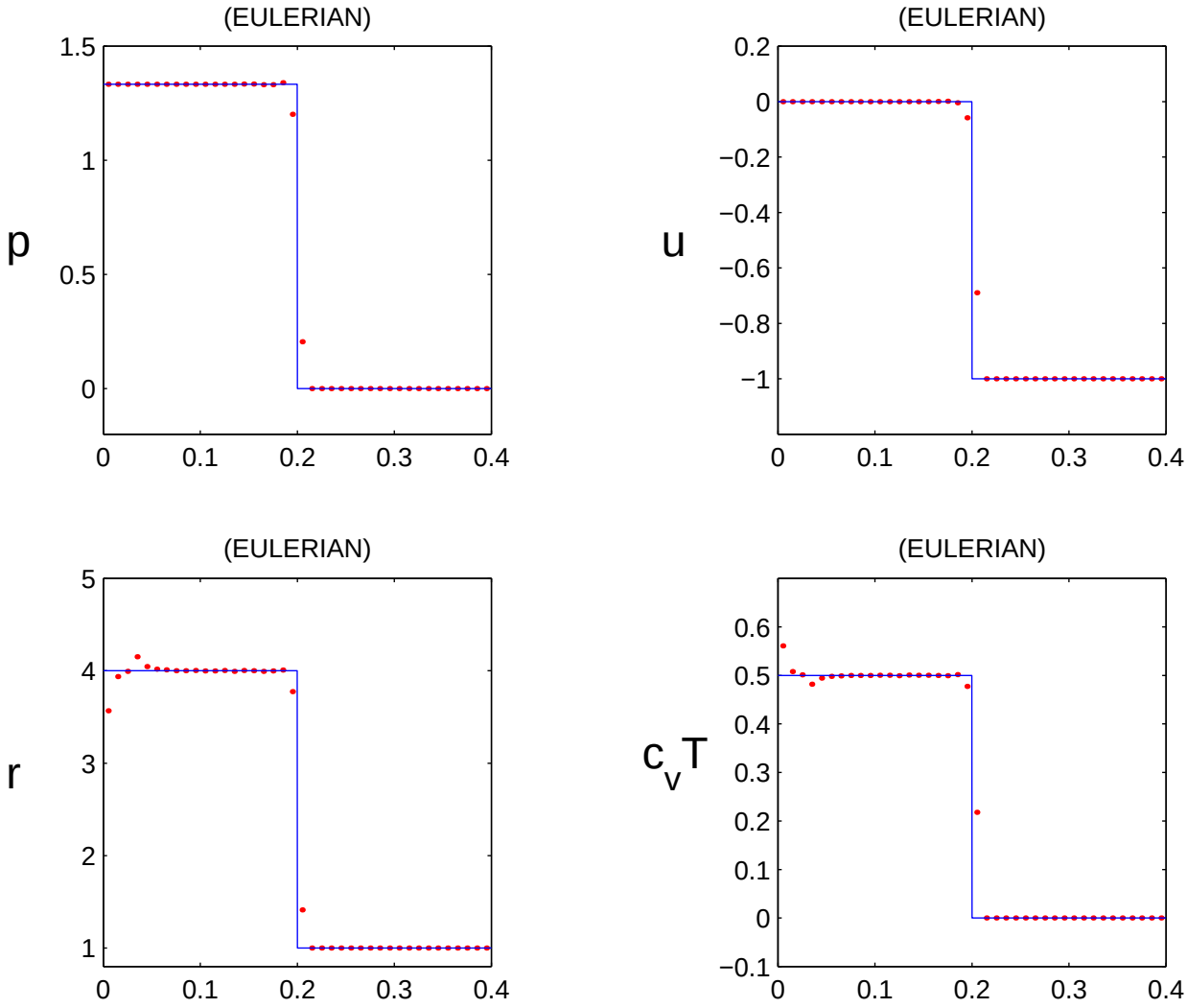

Figure 4: Test #1: Noh's problem. Eulerian computation (dots) employing Godunov-MUSCL scheme based on Eq.(1), compared with exact solution (solid line). $t = 0.6$.

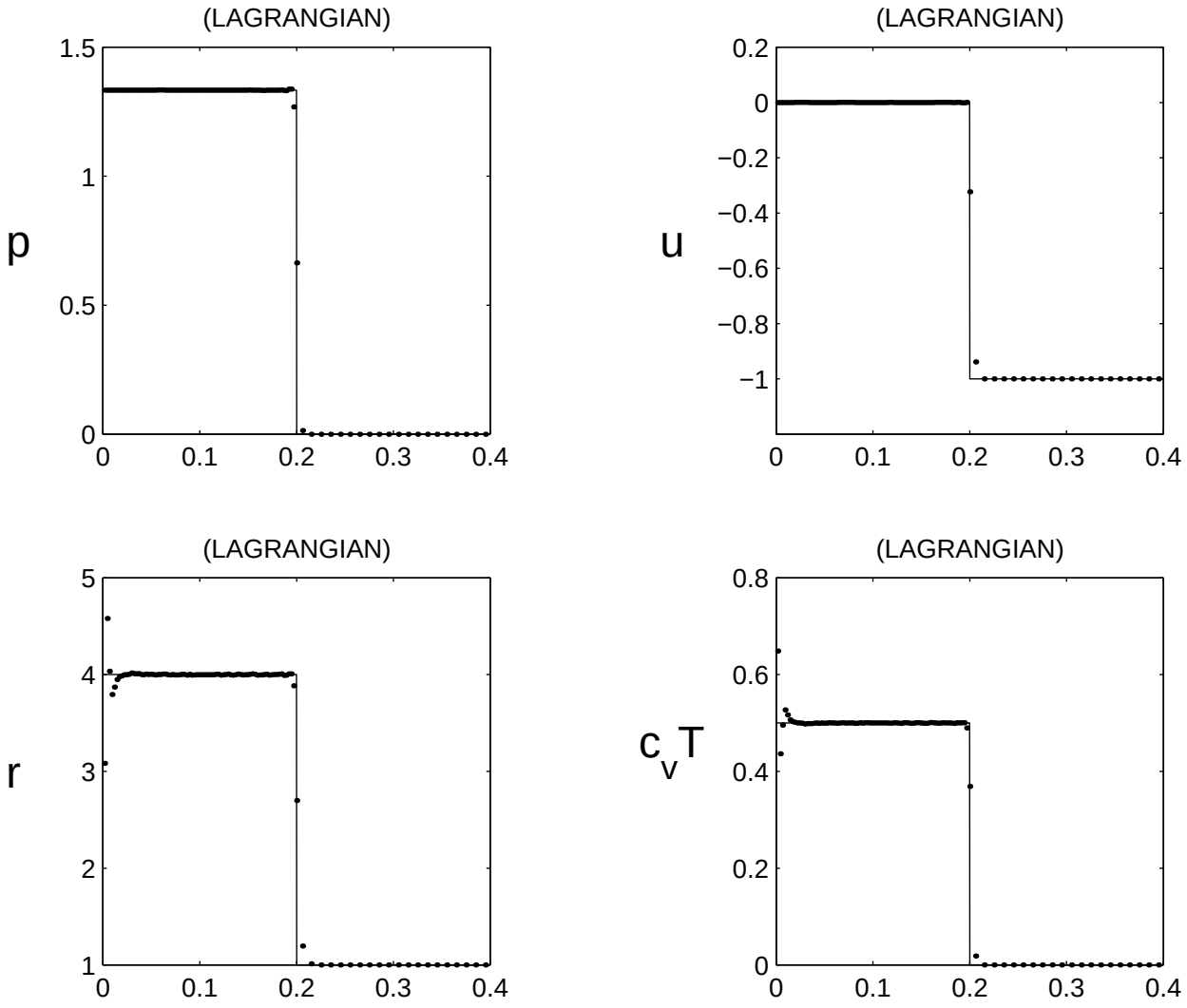

Figure 5: Test #1: Noh's problem. Lagrangian computation (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line). $t = 0.6$.

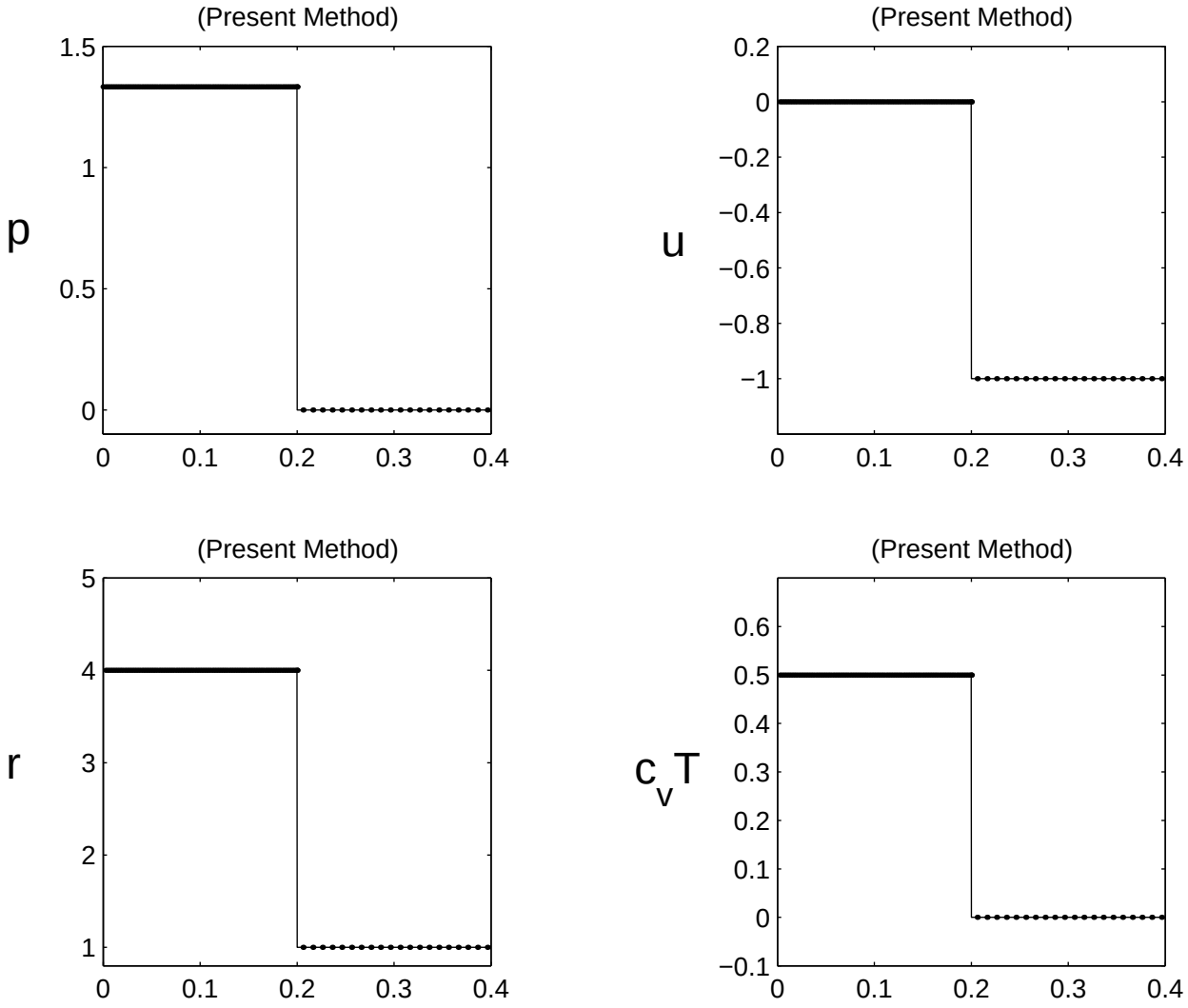

Figure 6: Test #1: Noh's problem. Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL scheme based on Eq.(6), compared with exact solution (solid line). $t = 0.6$.

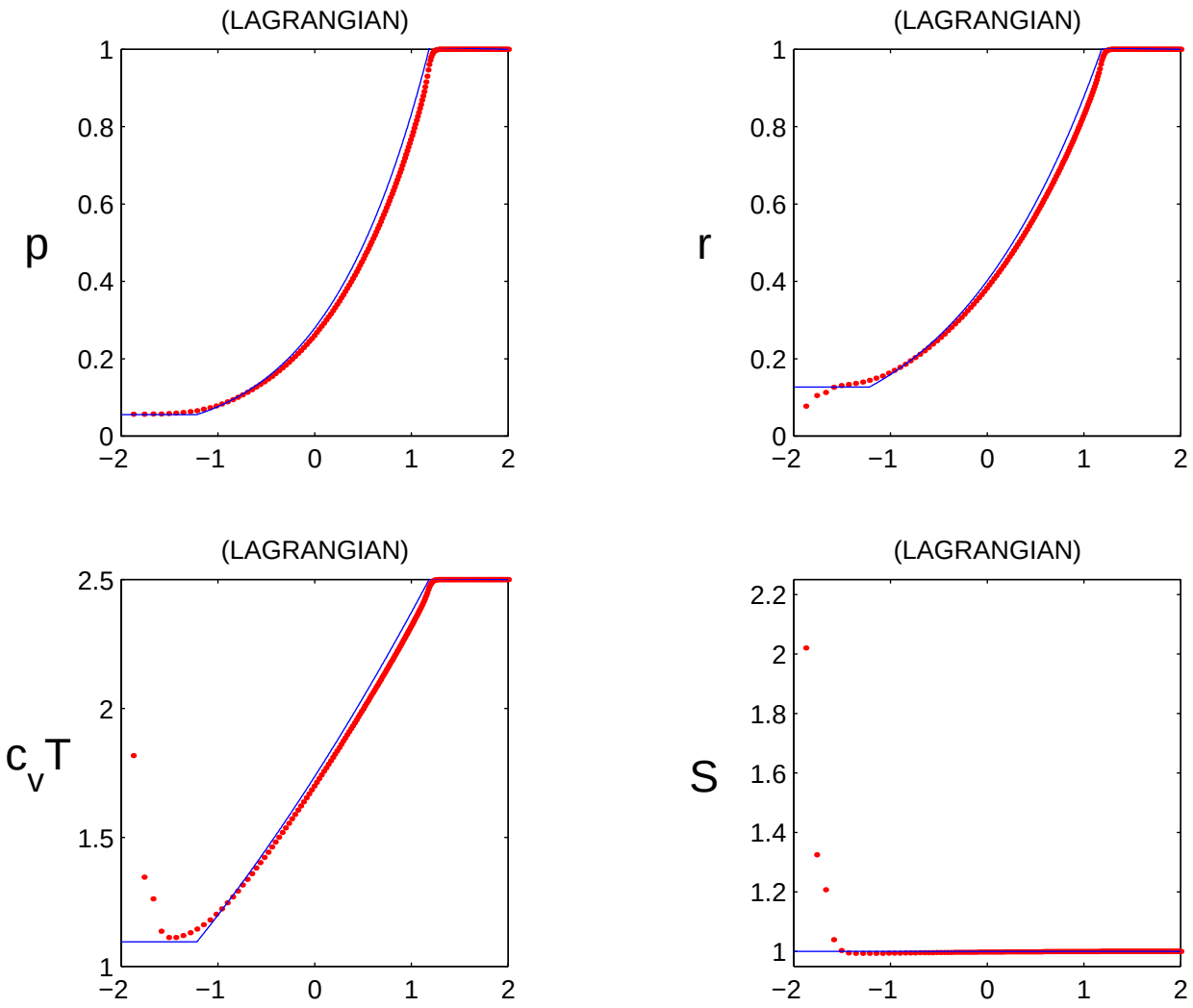

Figure 7: Test #2: Sudden expansion of a gas. Lagrangian computation (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line). $t = 1.0$.

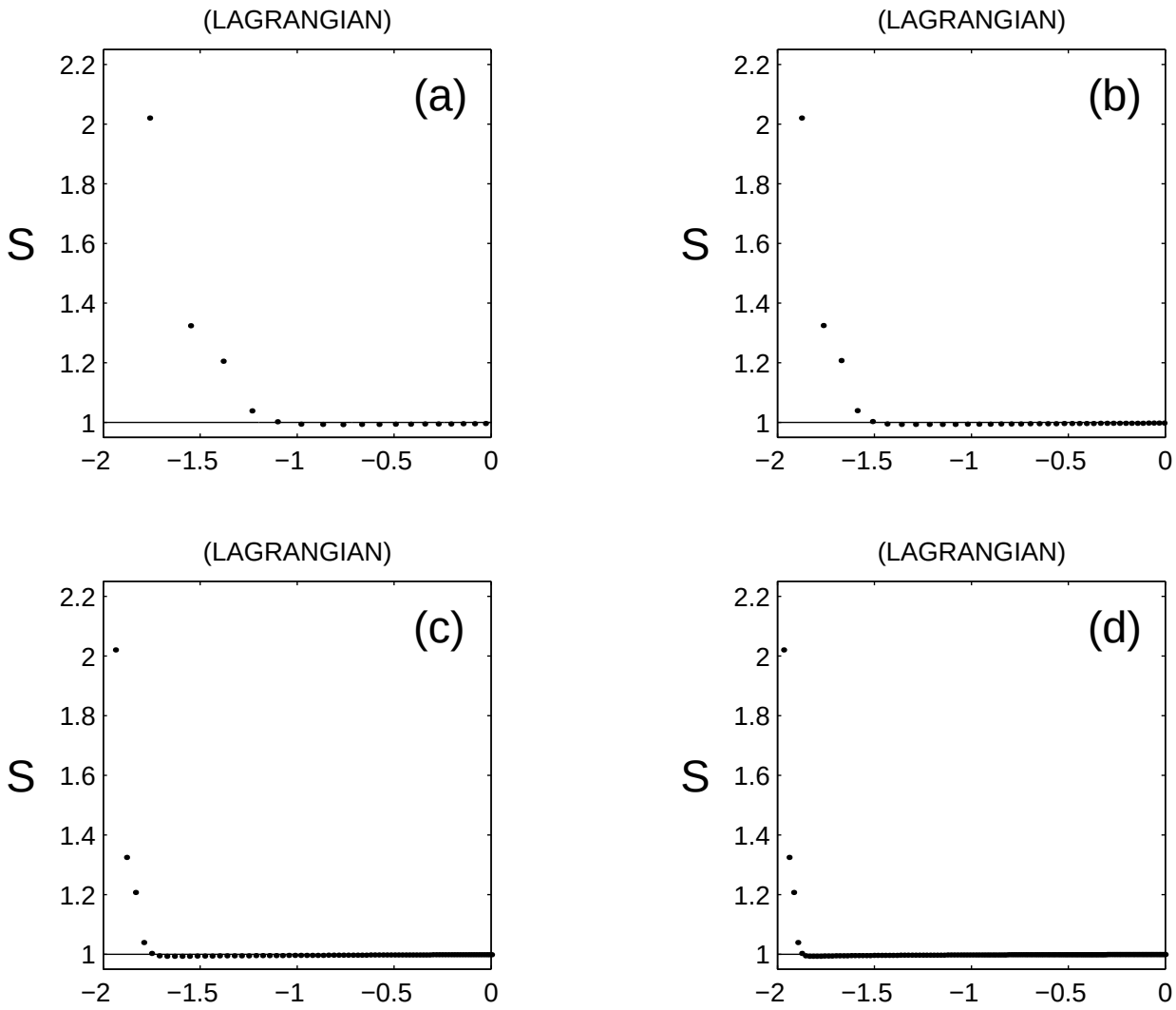

Figure 8: Test #2: Sudden expansion of a gas: effects of grid refinement. Computed entropy (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line) for different gird sizes: **(a)** $\triangle\xi = 1/100$, **(b)** $\triangle\xi = 1/200$, **(c)** $\triangle\xi = 1/400$, **(d)** $\triangle\xi = 1/800$; $t = 1.0$.

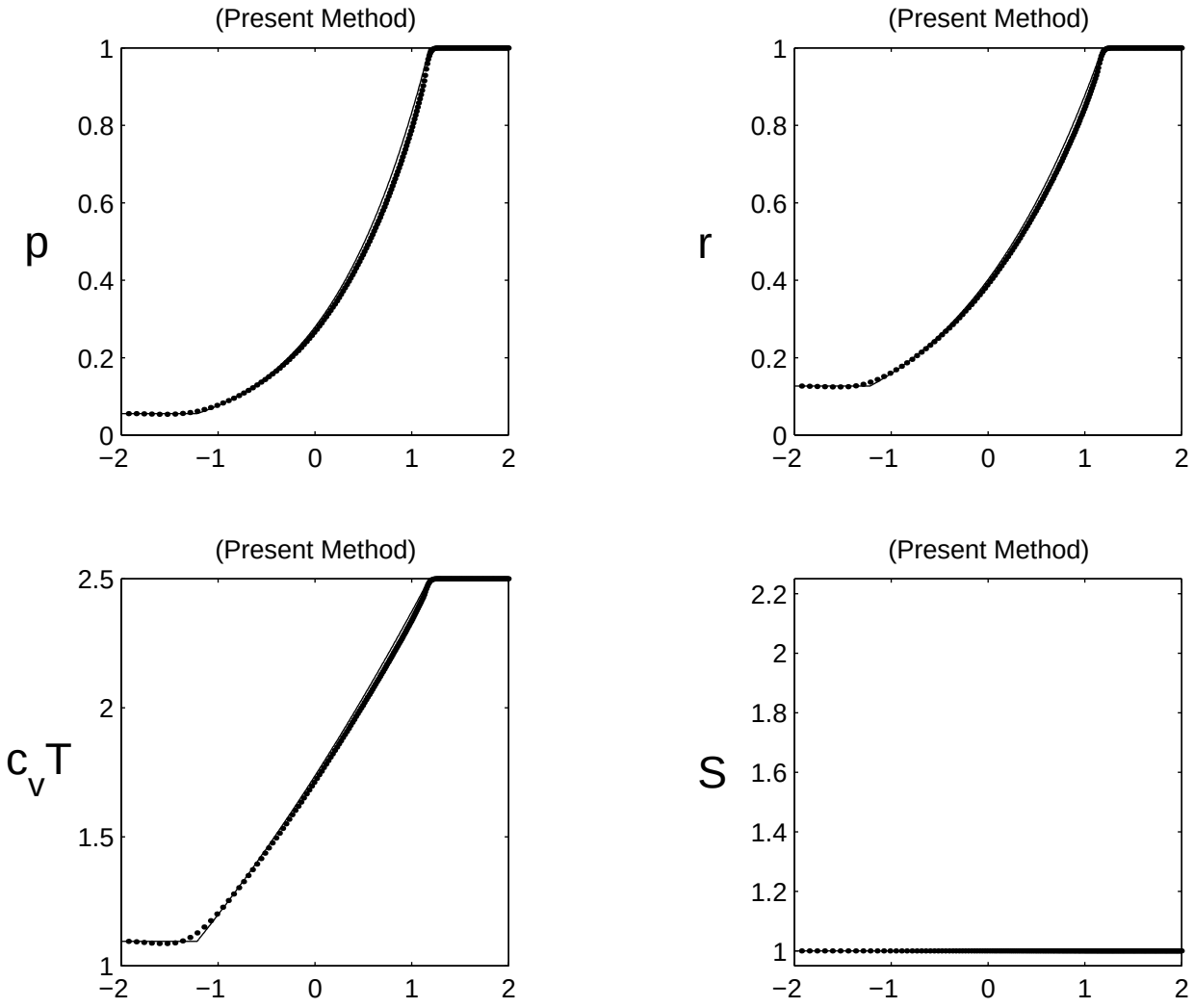

Figure 9: Test #2: Sudden expansion of a gas. Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL scheme based on Eq.(6), compared with exact solution (solid line). $t = 1.0$.

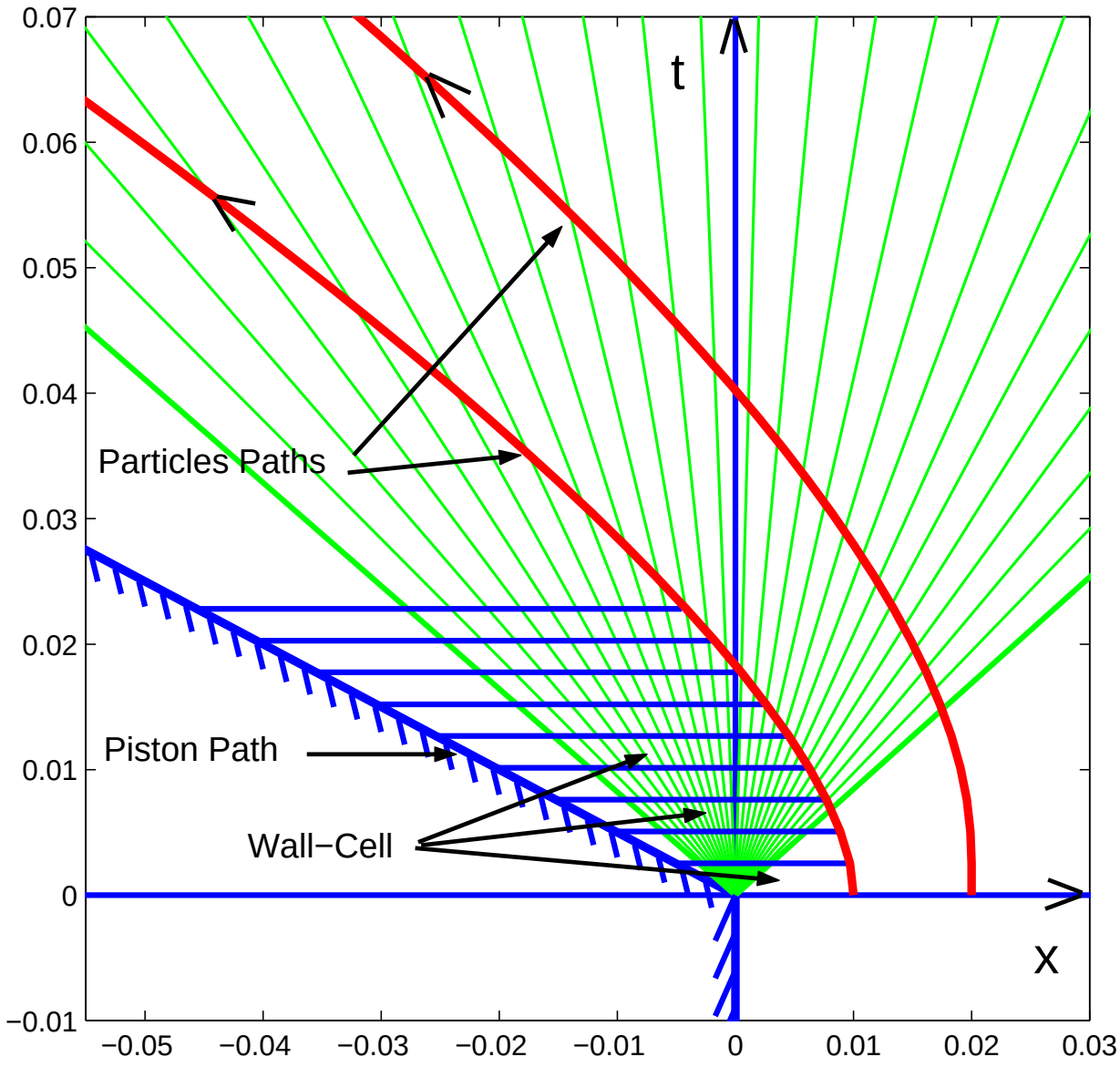

Figure 10: Test #2: Sudden expansion of a gas. Time evolution of the wall-cell.

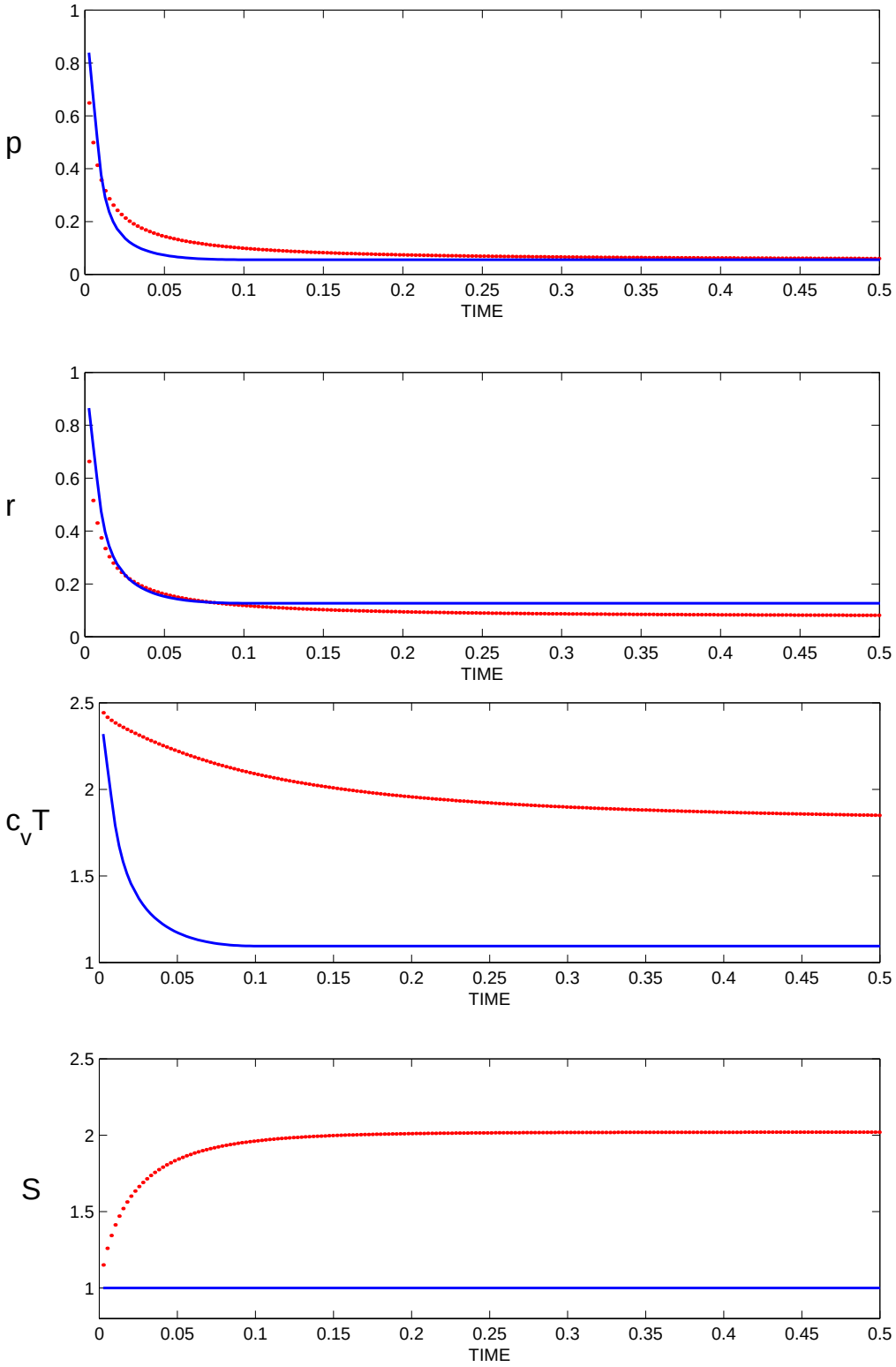

Figure 11: Test #2: Sudden expansion of a gas. Flow history of the wall-cell; Lagrangian computation (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line).

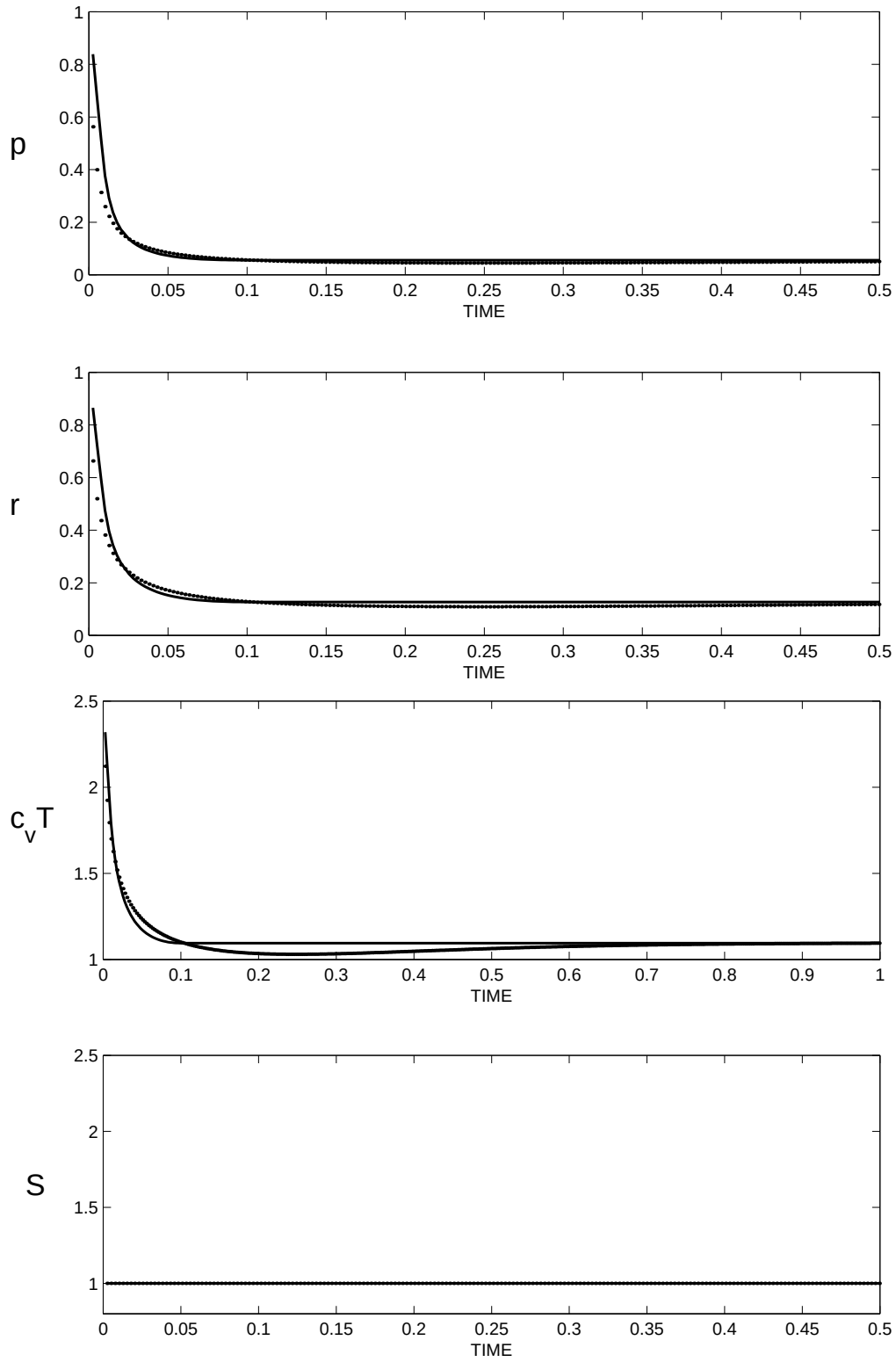

Figure 12: Test #2: Sudden expansion of a gas. Flow history of the wall-cell; Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL scheme based on Eq.(6), compared with exact solution (solid line).

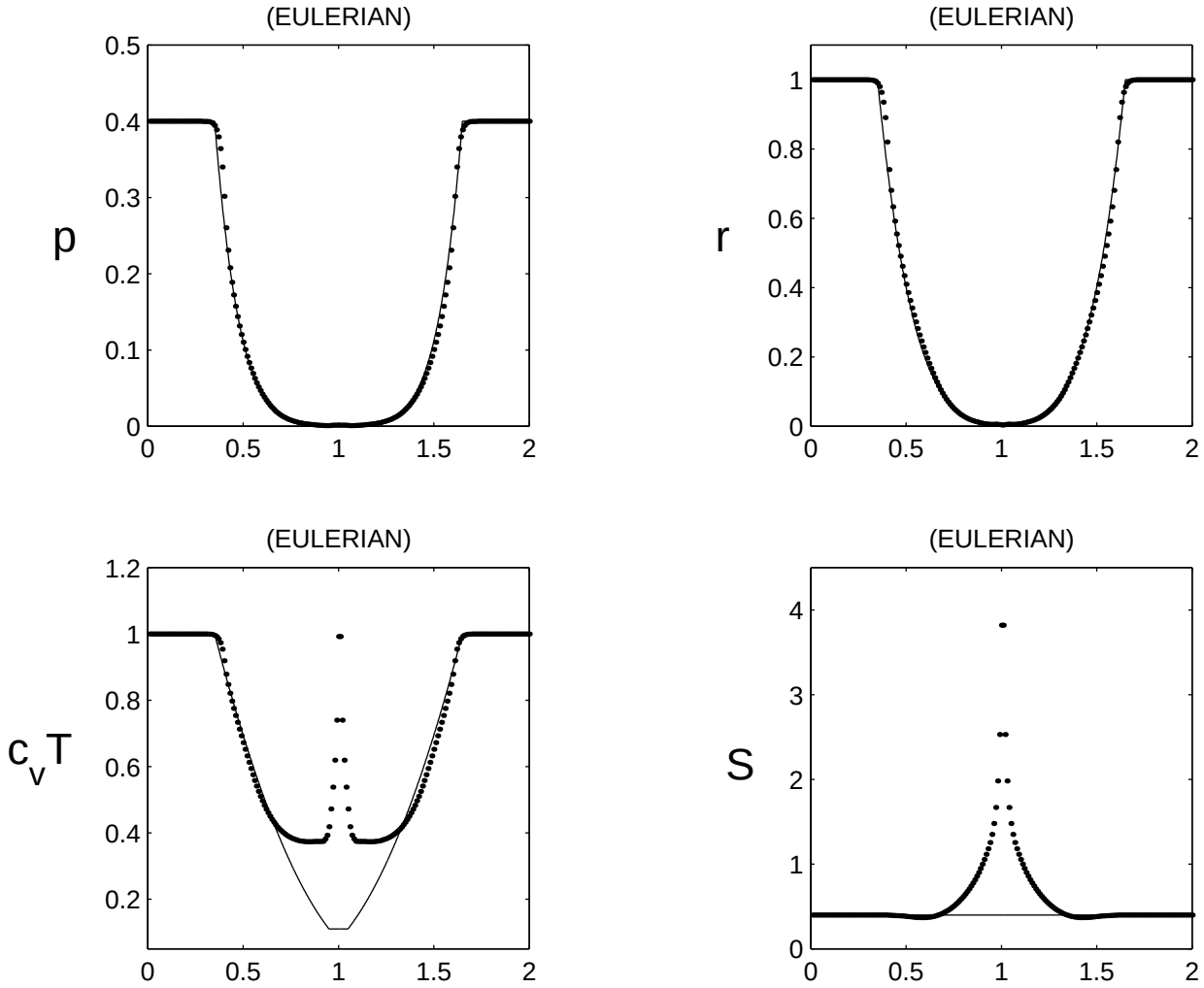

Figure 13: Test #3: $1-2-3$ problem. Eulerian computation (dots) for the employing Godunov-MUSCL scheme based on Eq.(1), compared with exact solution (solid line). $t = 0.2$.

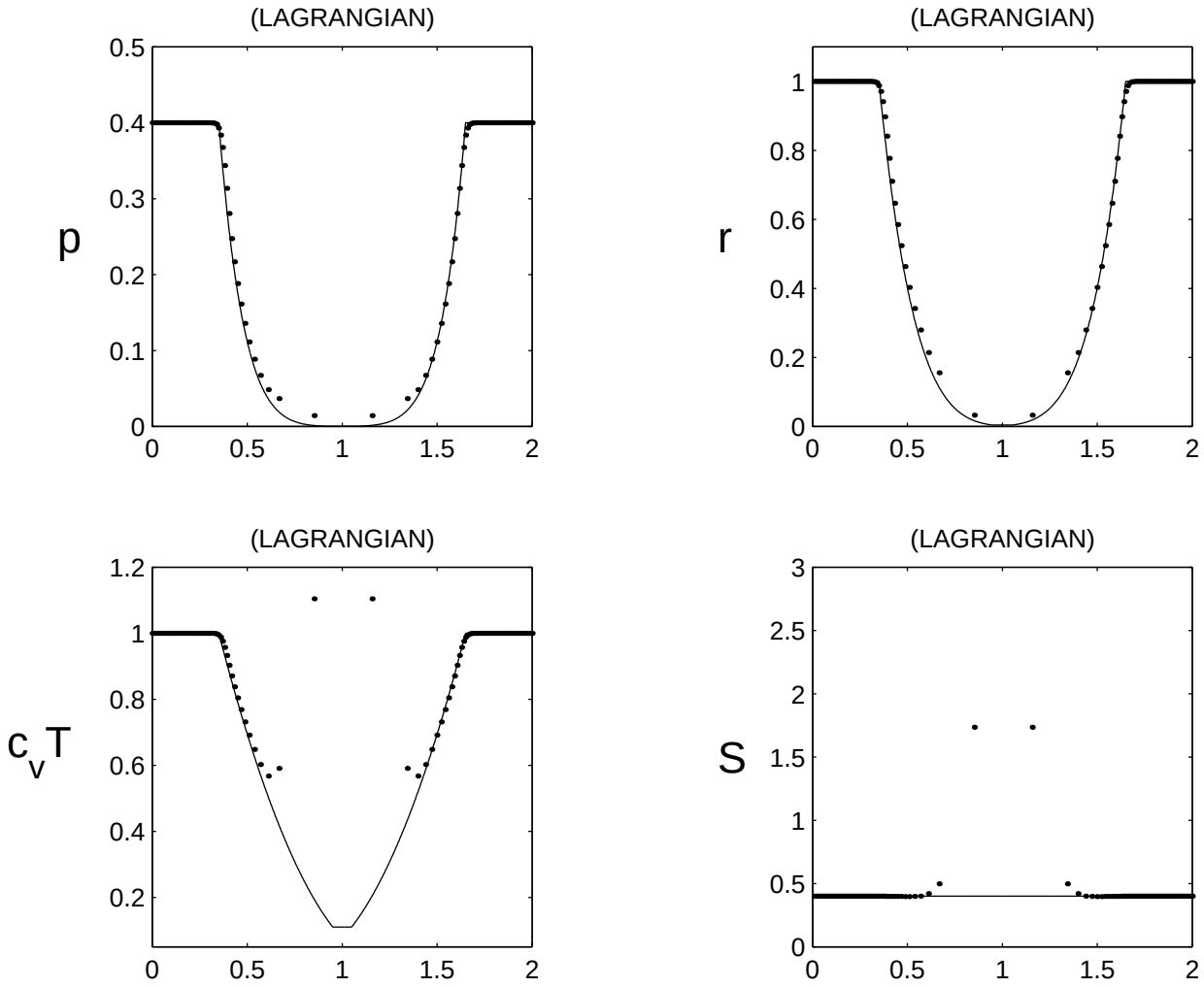

Figure 14: Test #3: $1-2-3$ problem. Lagrangian computation (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line). $t = 0.2$.

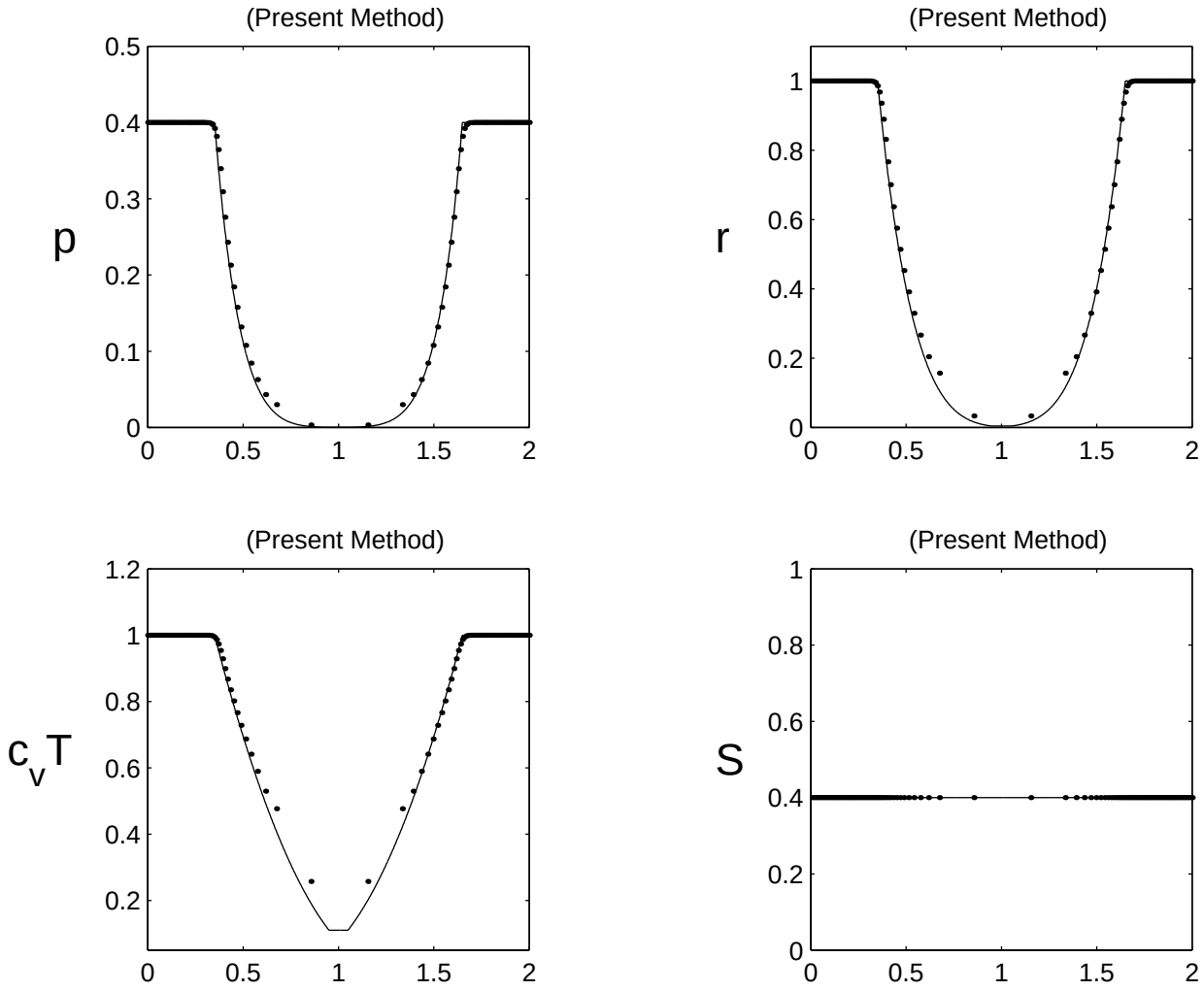

Figure 15: Test #3: $1-2-3$ problem. Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL scheme based on Eq.(6), compared with exact solution (solid line). $t = 0.2$.

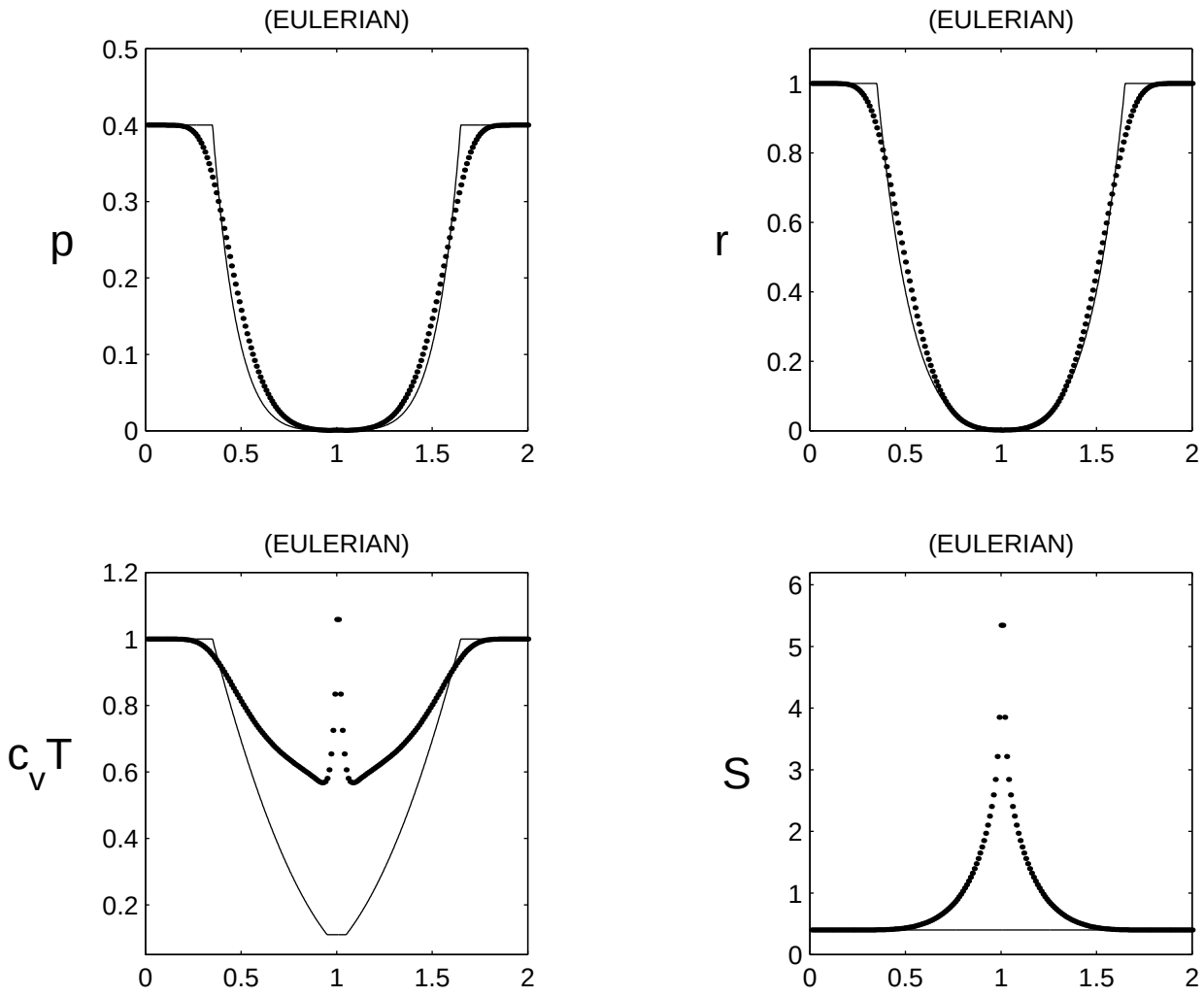

Figure 16: Test #3: $1-2-3$ problem. Eulerian computation (dots) employing Godunov scheme based on Eq.(1), compared with exact solution (solid line). $t = 0.2$.

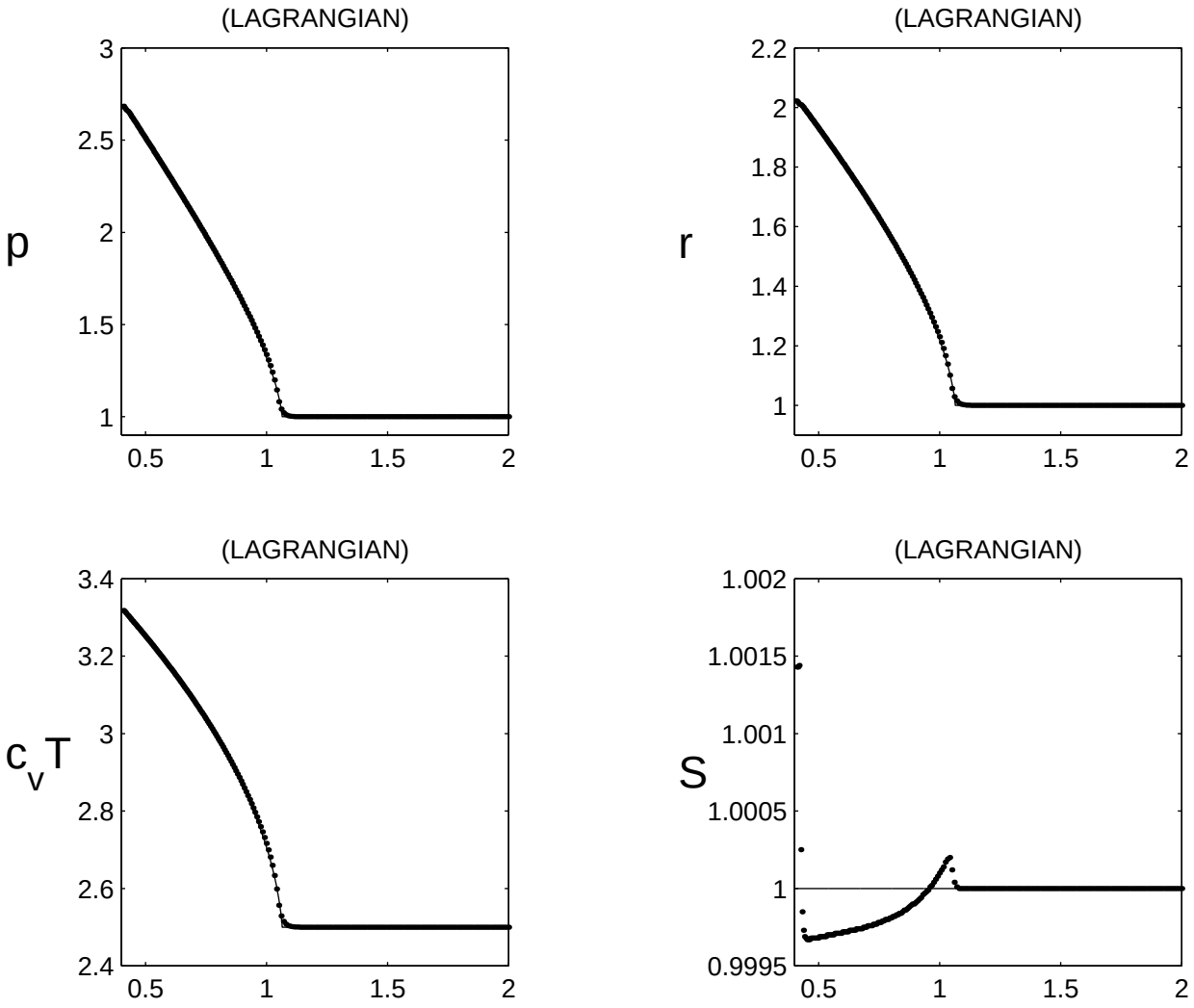

Figure 17: Test #4: Gradual compresion of a gas. Lagrangian computation (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line). $t = 0.9$.

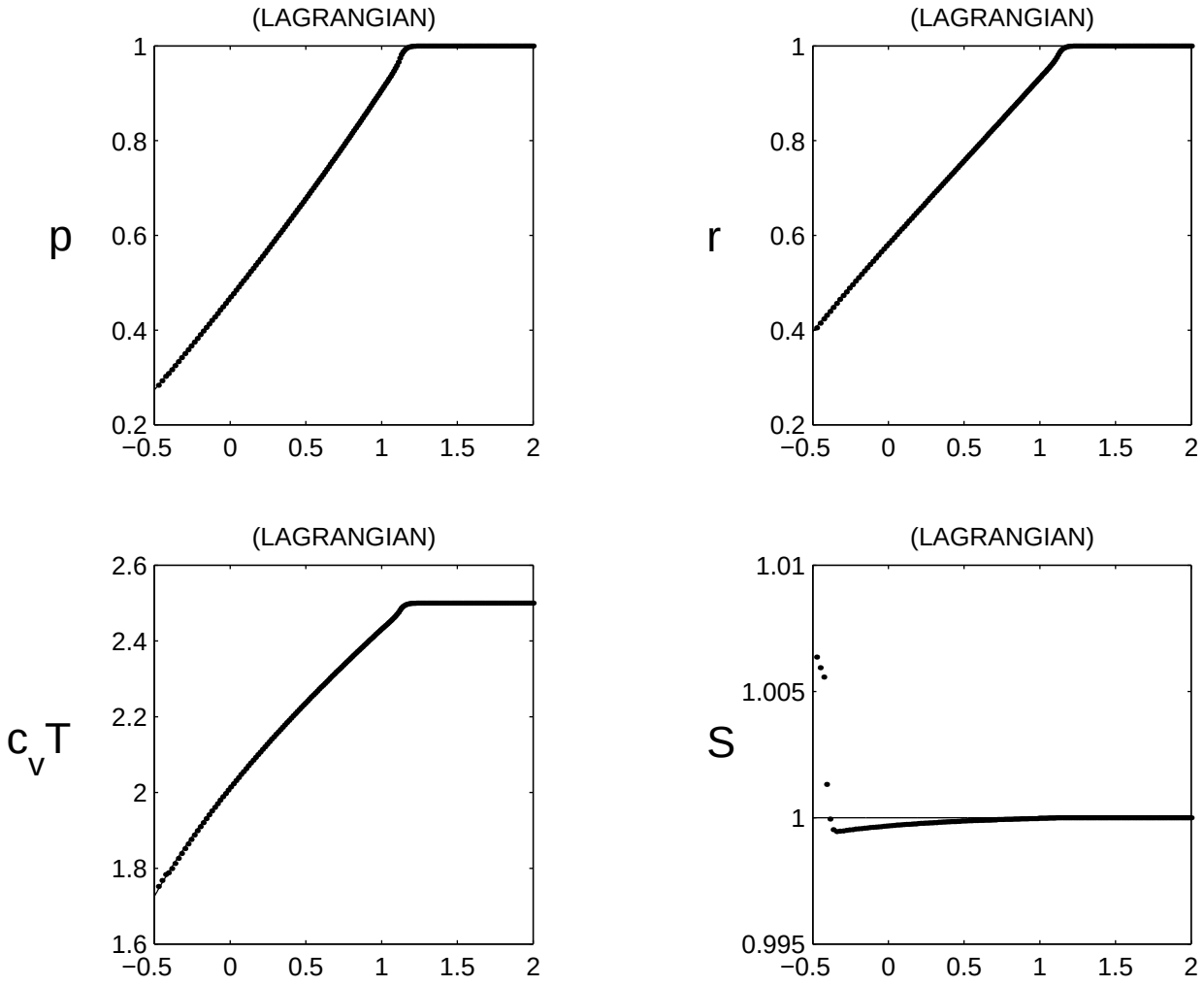

Figure 18: Test #4: Gradual expansion of a gas. Lagrangian computation (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line). $t = 1.0$.

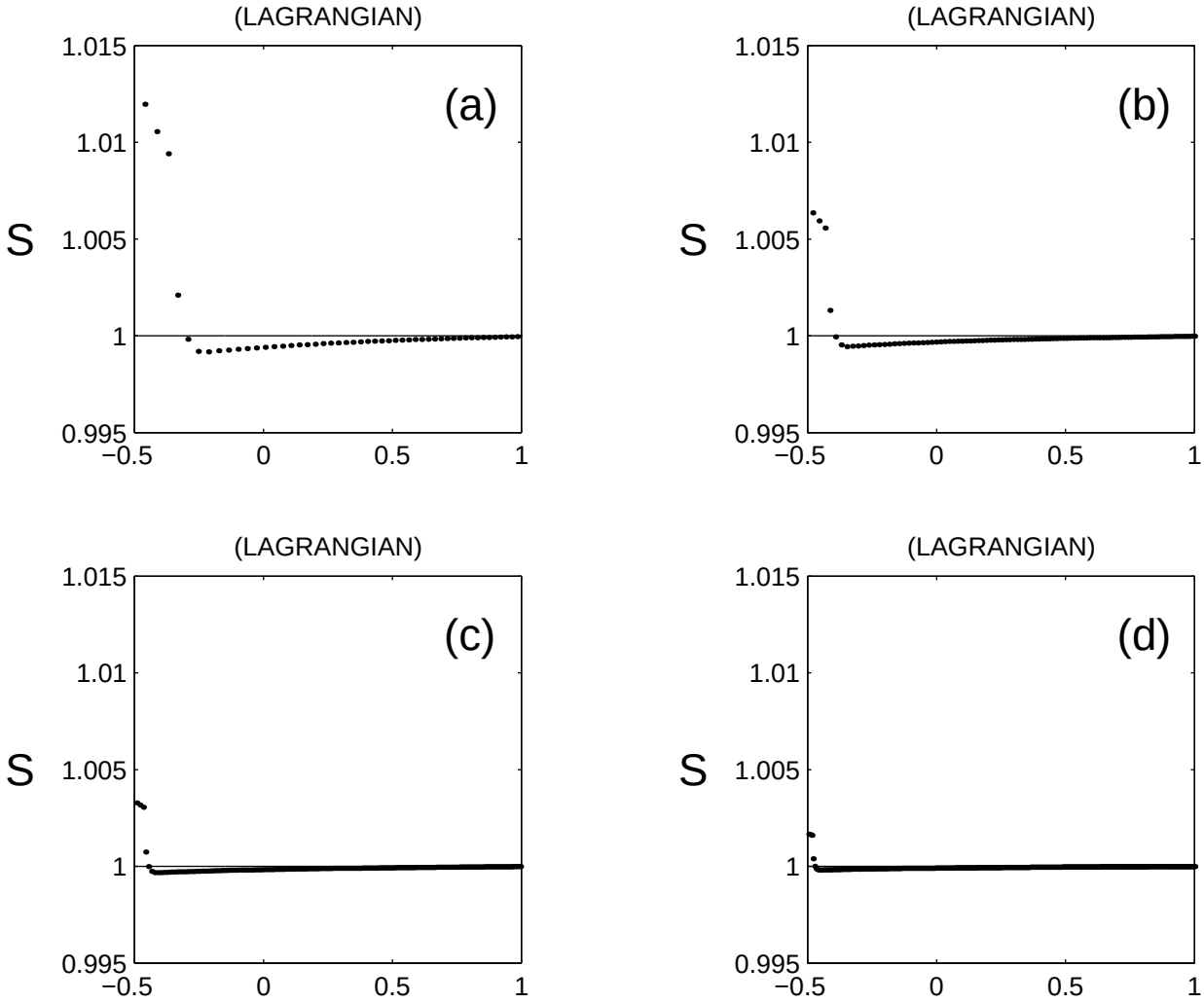

Figure 19: Test #4: Gradual expansion of a gas: effects of grid refinement. Computed entropy (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line); **(a)** $\triangle\xi = 1/100$, **(b)** $\triangle\xi = 1/200$, **(c)** $\triangle\xi = 1/400$, **(d)** $\triangle\xi = 1/800$; $t = 1.0$.

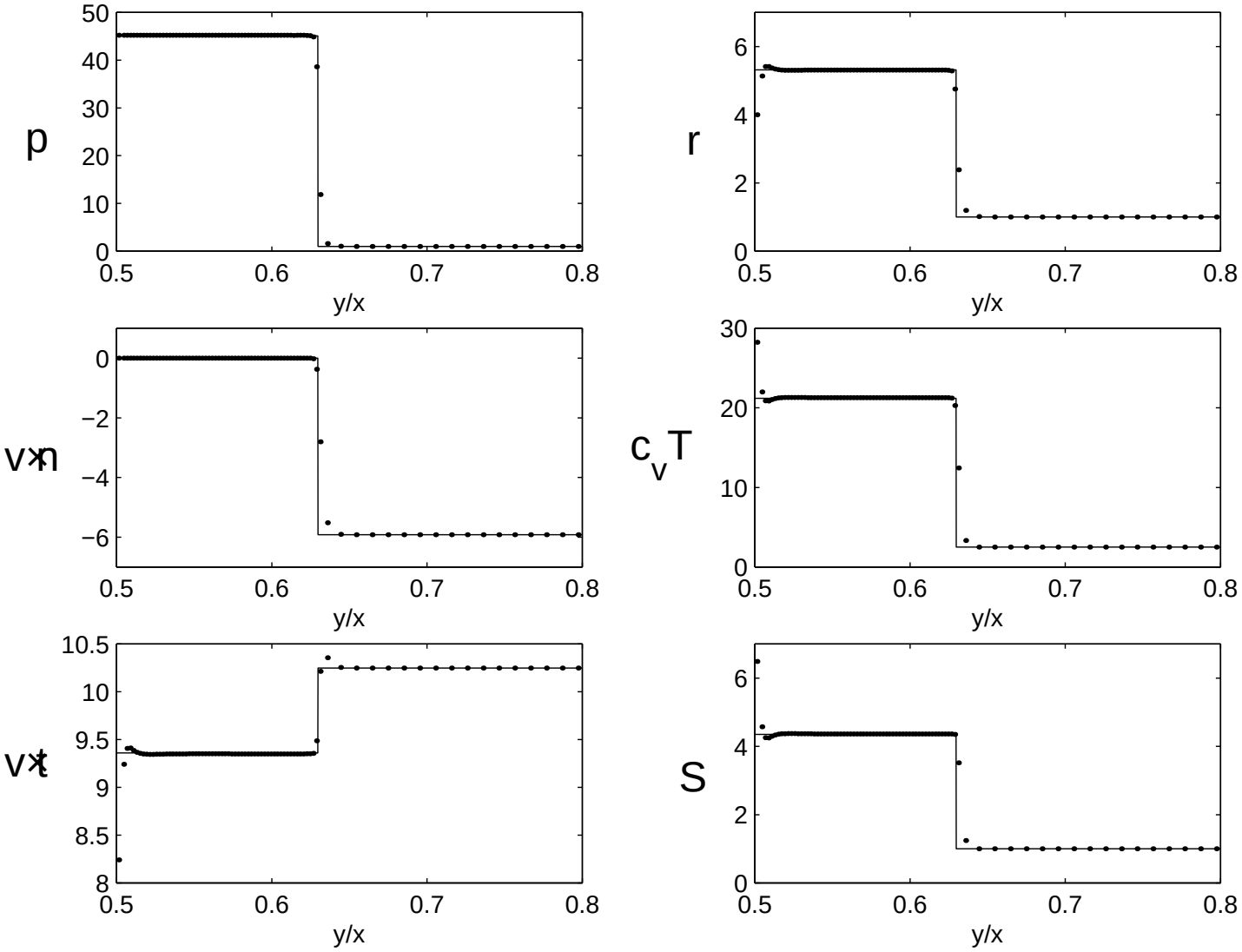

Figure 20: Test #5: Supersonic wedge flow. Lagrangian computation (dots) employing Godunov-MUSCL method based on Eq.(15) compared with exact solution (solid line)

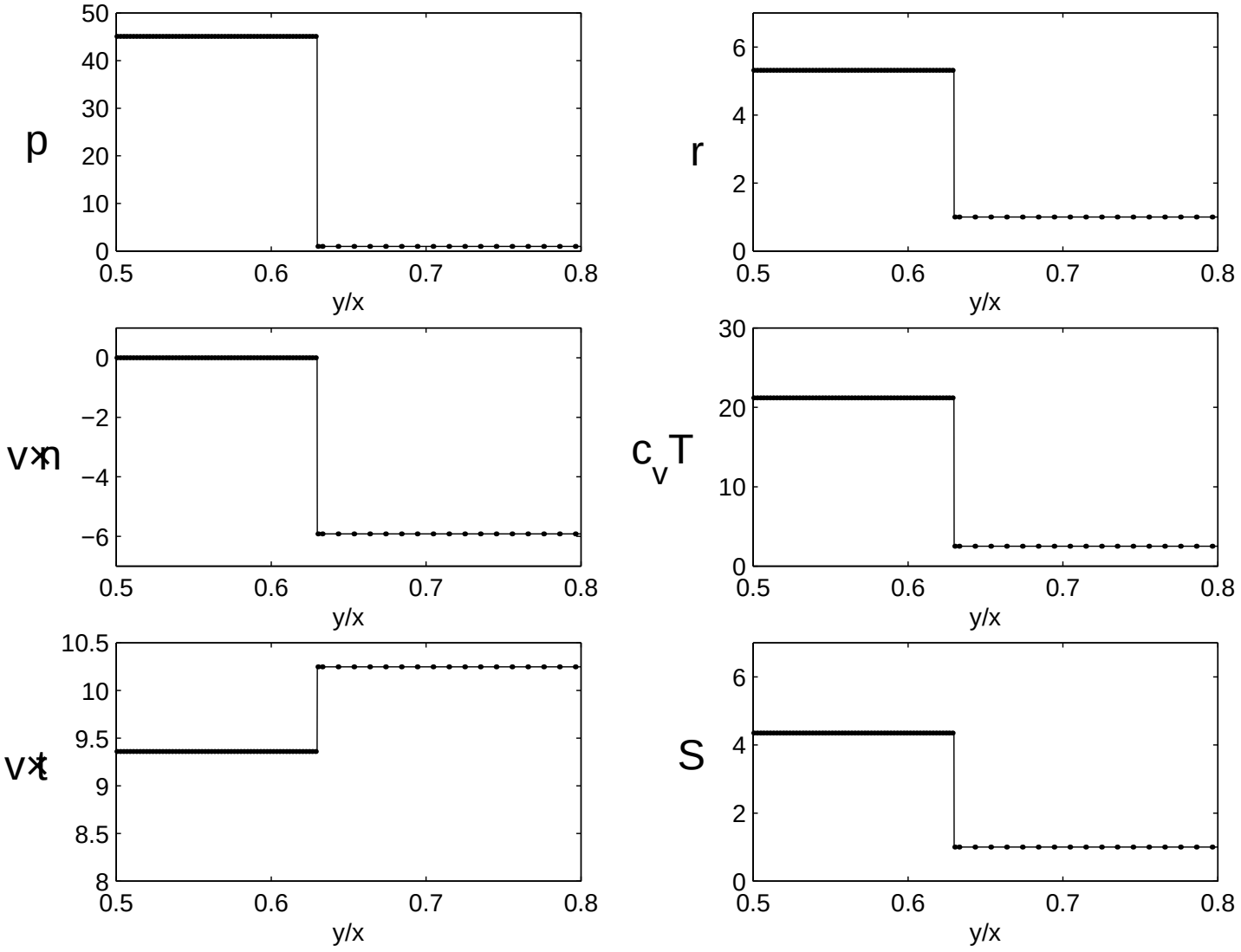

Figure 21: Test #5: Supersonic wedge flow. Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL method based on Eq.(17) compared with exact solution (solid line)

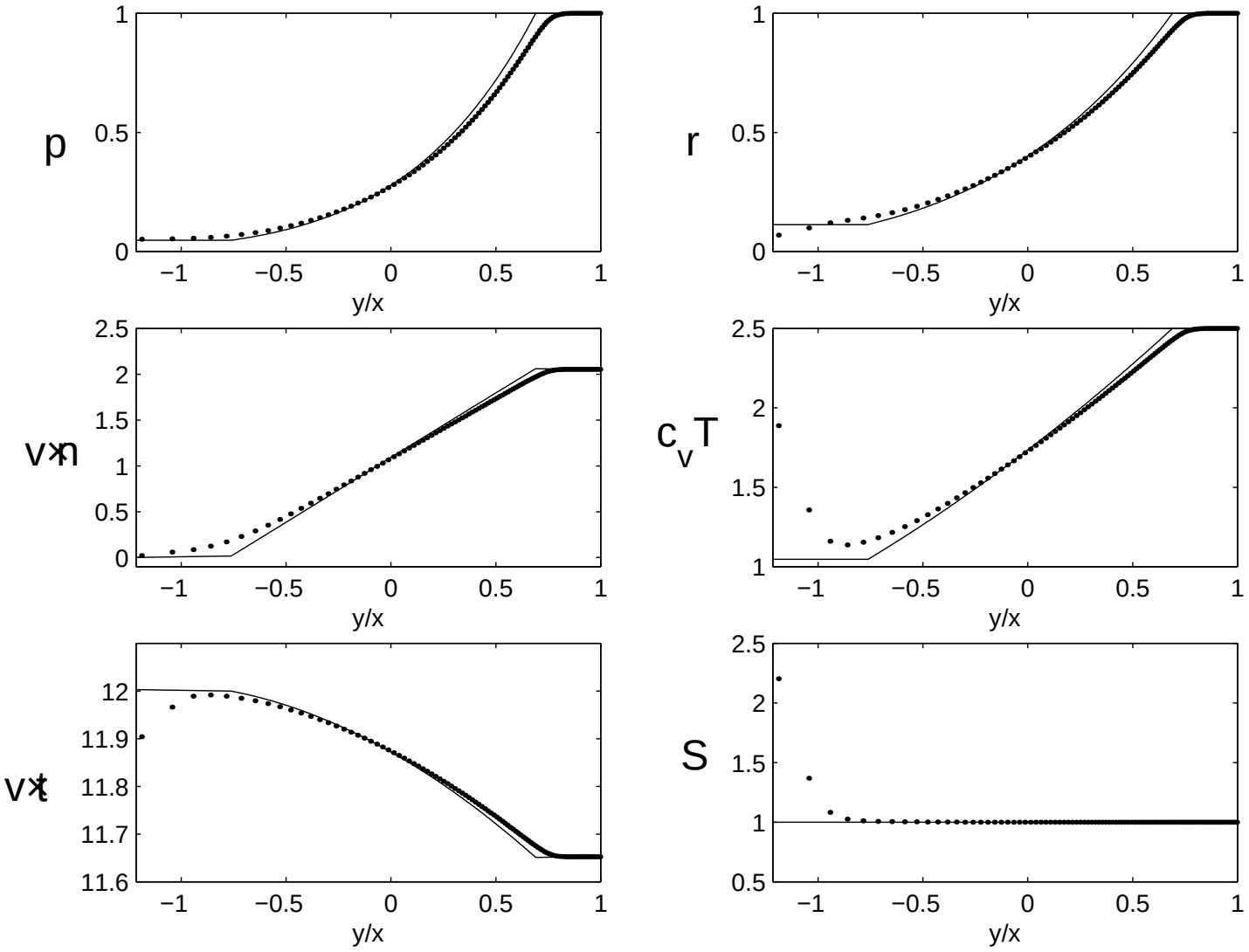

Figure 22: Test #6: Prandtl-Meyer expansion flow. Lagrangian computation (dots) employing Godunov-MUSCL method based on Eq.(15) compared with exact solution (solid line)

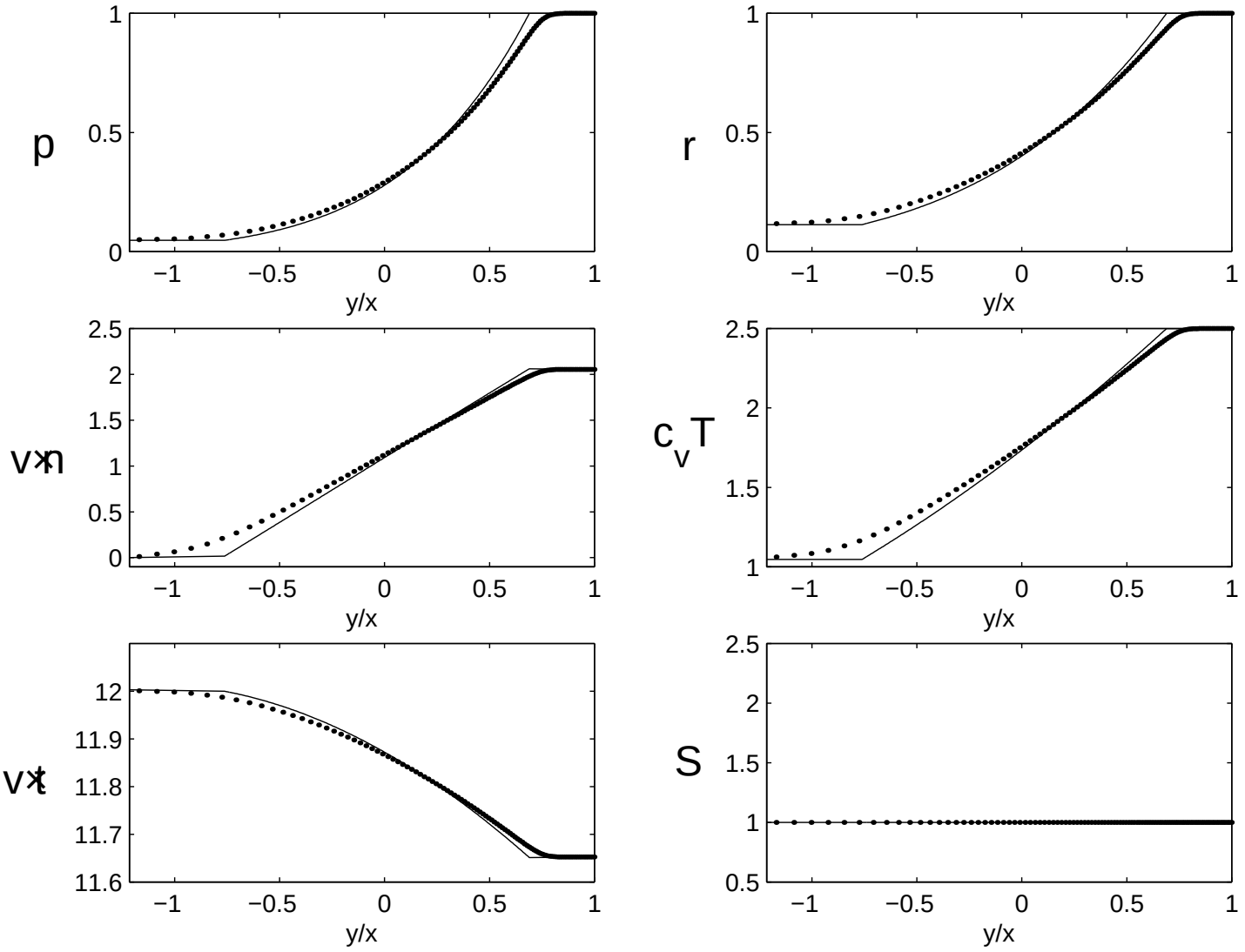

Figure 23: Test #6: Prandtl-Meyer expansion flow. Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL method based on Eq.(17) compared with exact solution (solid line)

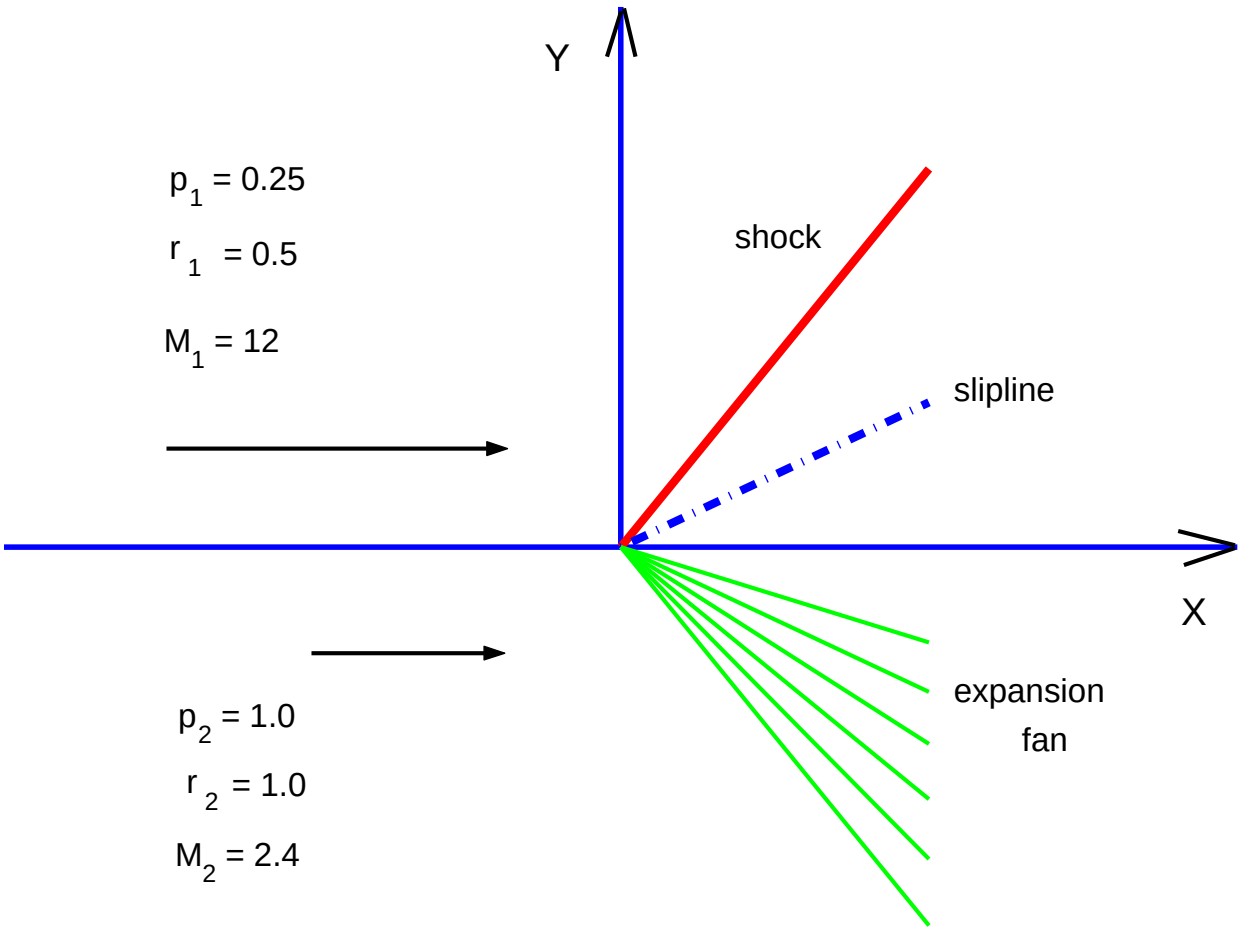

Figure 24: Test # 7: Sketch of a 2-D Riemann problem.

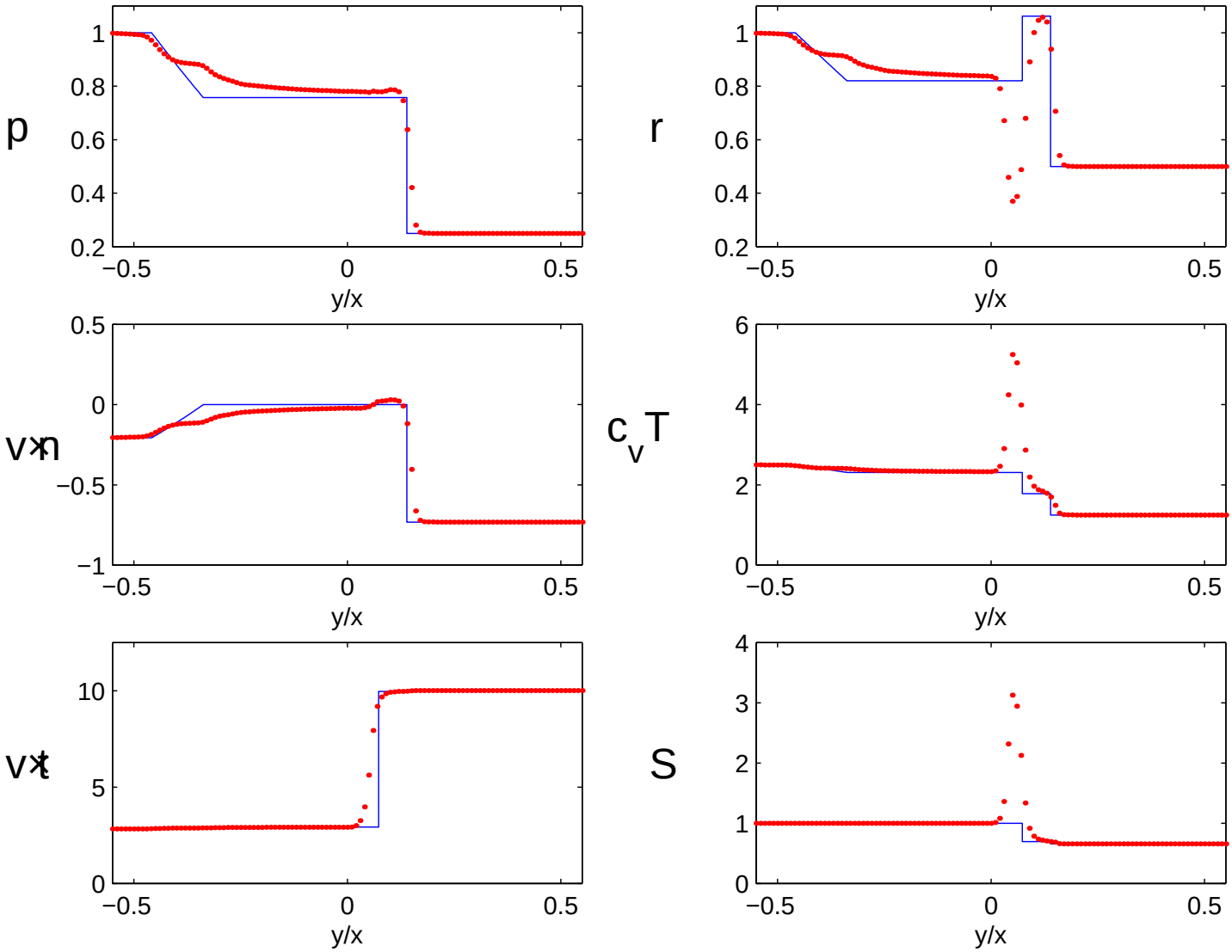

Figure 25: Test # 7: 2-D Riemann problem. Eulerian computation (dots) employing Godunov-MUSCL scheme based on Eq.(8), compared with exact solution (solid line).

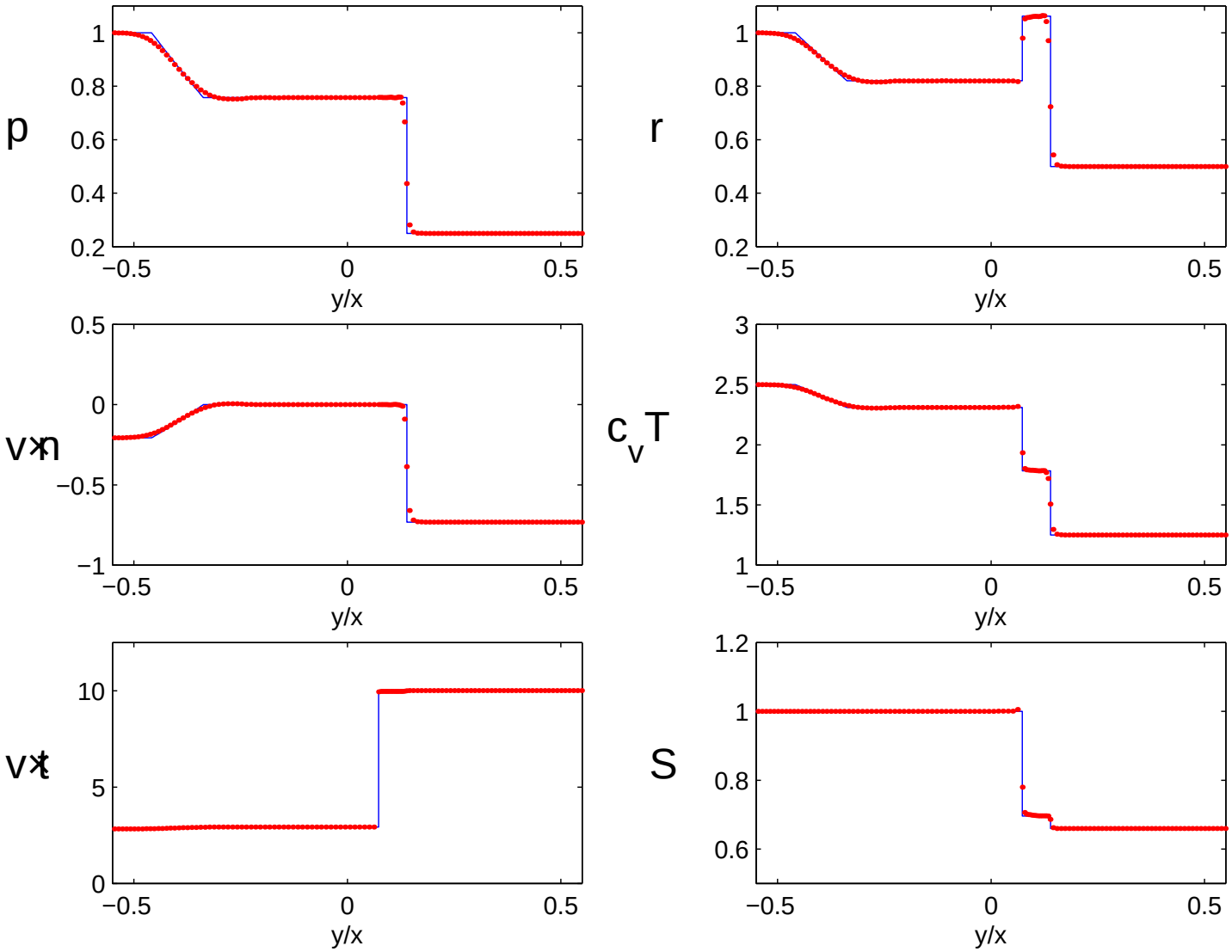

Figure 26: Test # 7: 2-D Riemann problem. Lagrangian computation (dots) employing Godunov-MUSCL scheme based on Eq.(15), compared with exact solution (solid line).

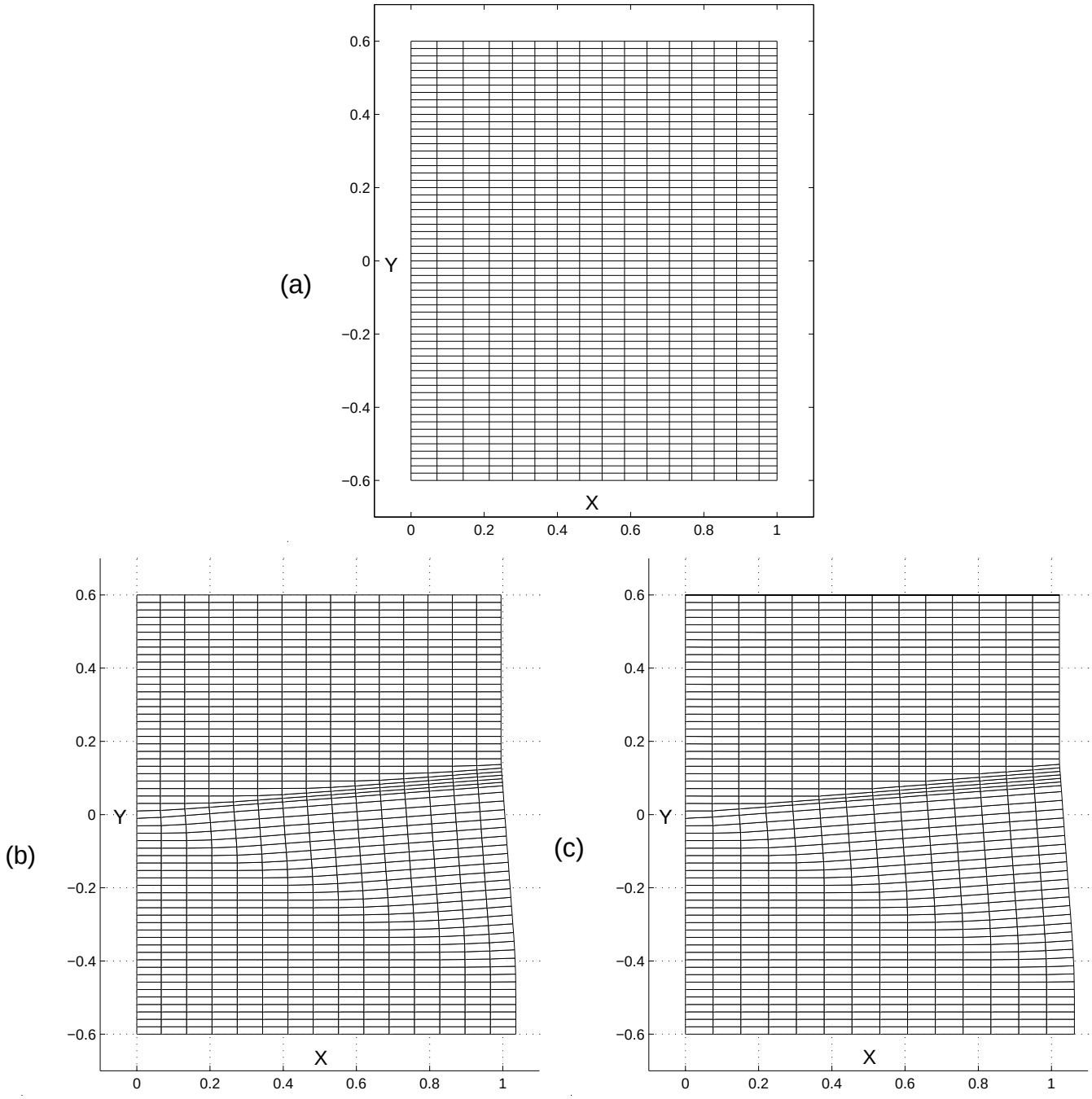

Figure 27: Test # 7: 2-D Riemann problem. Computed grids: **(a)** in Eulerian coordinates, Eq.(8) employing Godunov-MUSCL scheme, **(b)** in Lagrangian coordinates, Eq.(15) employing Godunov-MUSCL scheme, **(c)** in Lagrangian coordinates, Eq.(17) employing **shock-adaptive** Godunov-MUSCL scheme. (In Y-direction each second line is plotted, while in X-direction - each 33rd.)

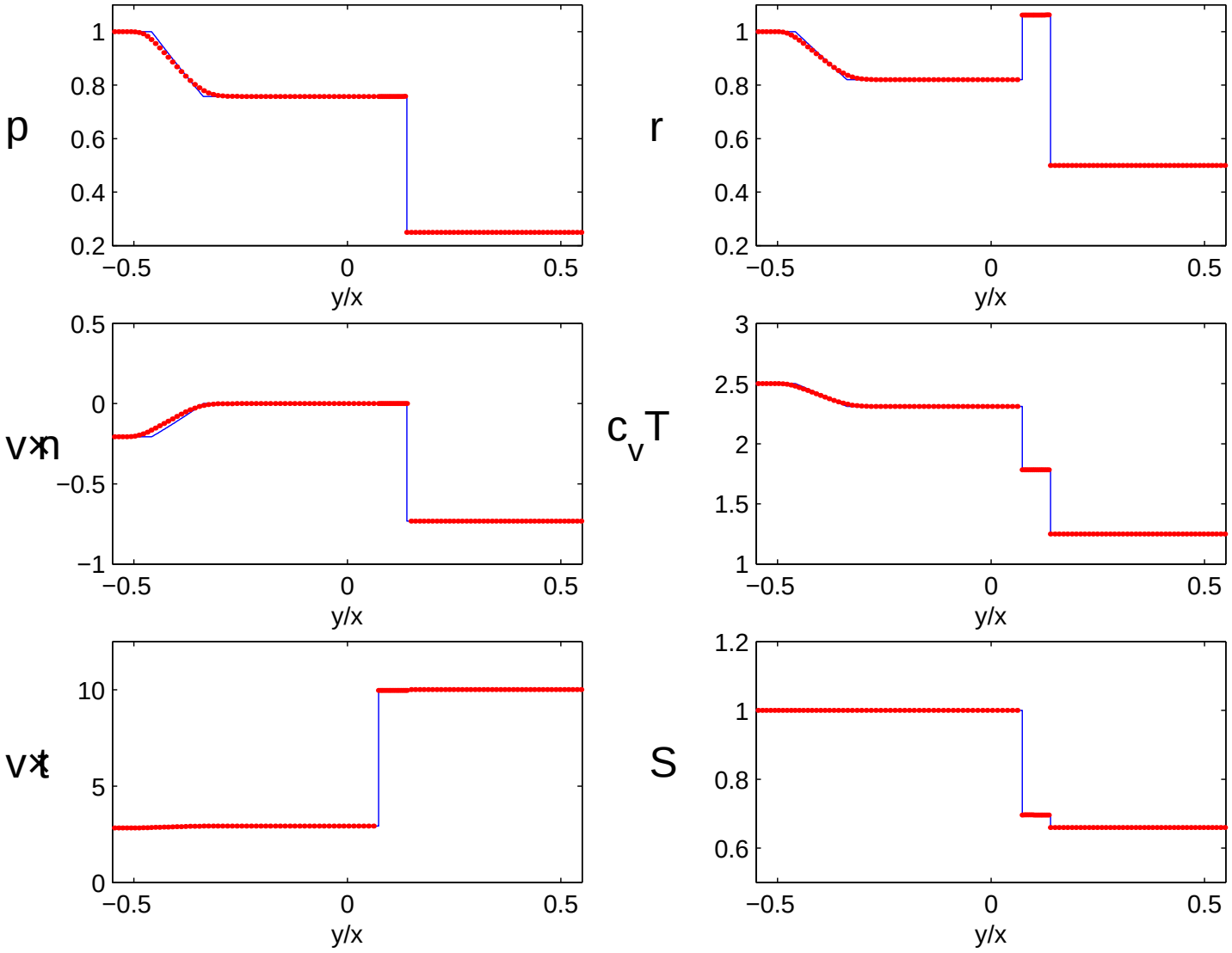

Figure 28: Test # 7: 2-D Riemann problem. Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL scheme based on Eq.(17), compared with exact solution (solid line).

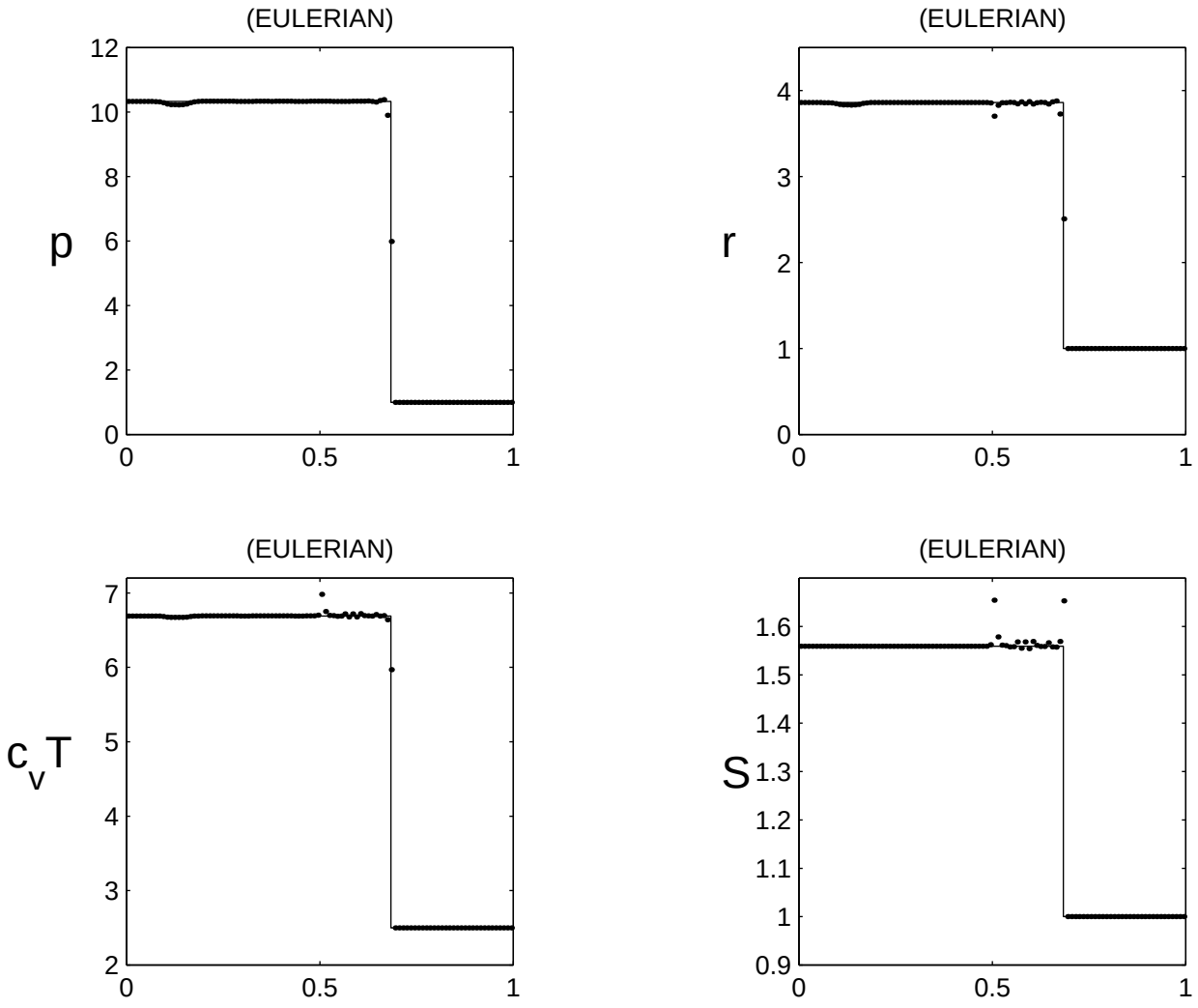

Figure 29: Test # 8: Start-up errors. Eulerian computation (dots) employing Godunov-MUSCL scheme based on Eq.(1), compared with exact solution (solid line). $t = 0.2$.

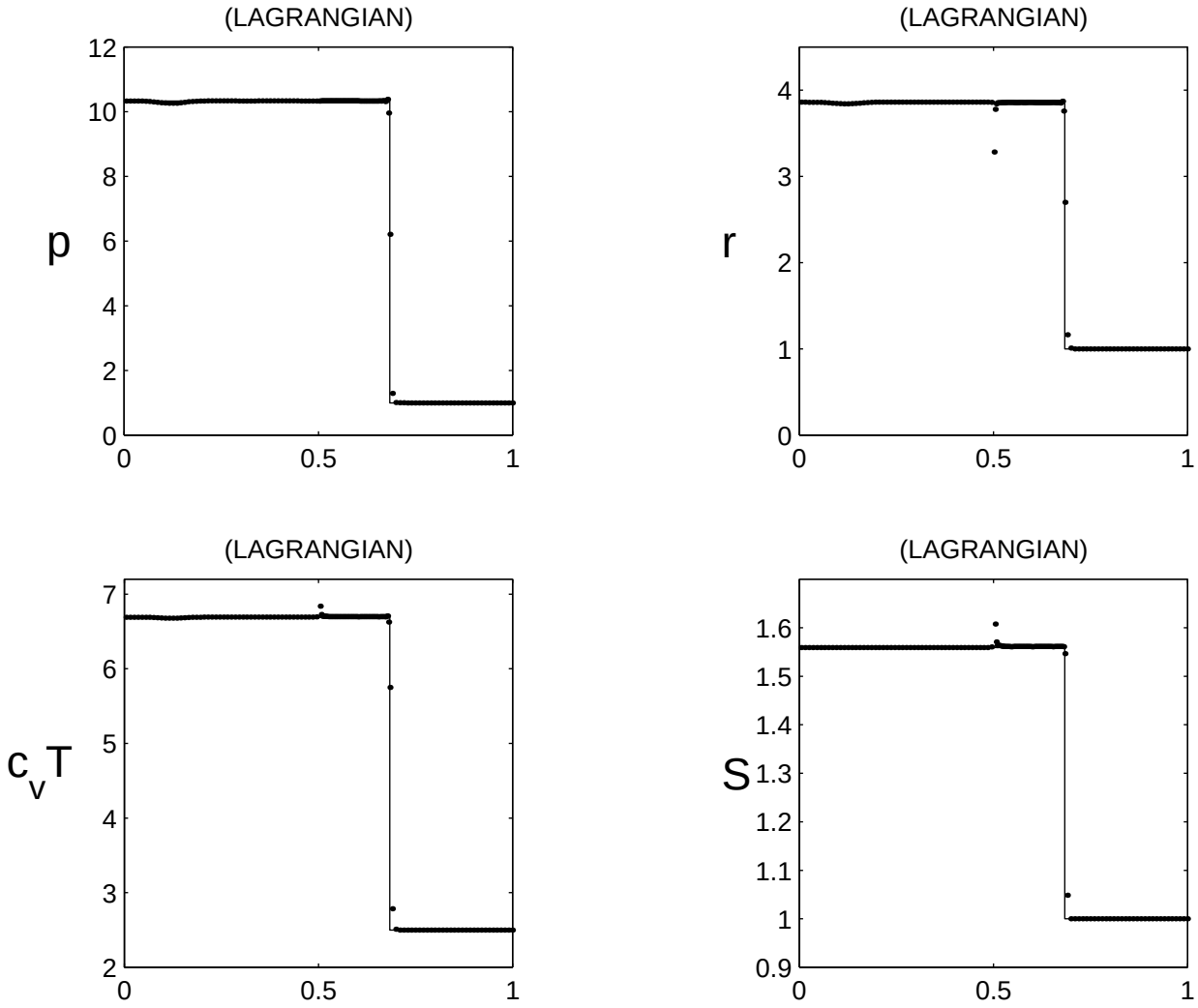

Figure 30: Test # 8: Start-up errors. Lagrangian computation (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line). $t = 0.2$.

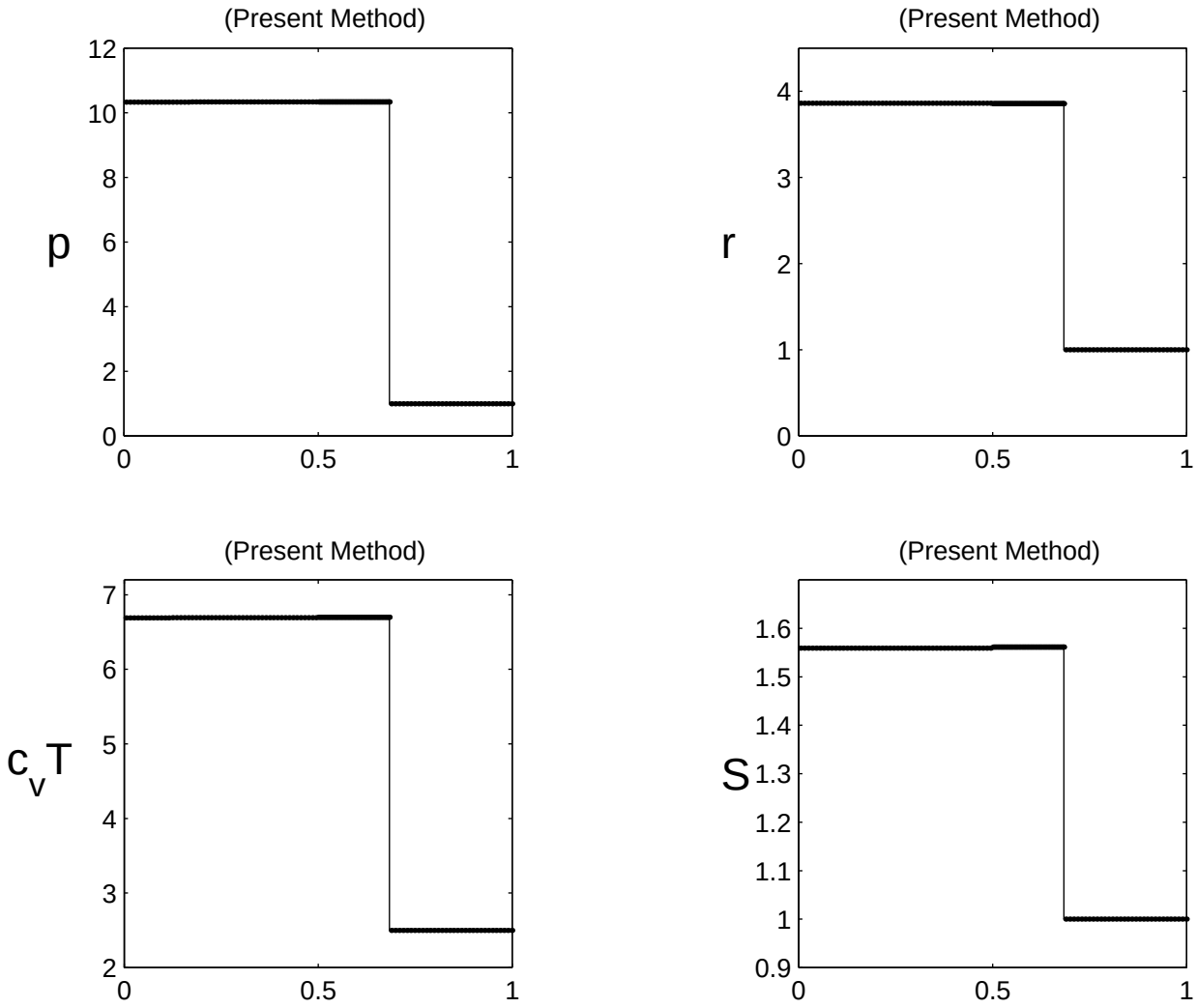

Figure 31: Test # 8: Start-up errors. Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL scheme based on Eq.(6), compared with exact solution (solid line). $t = 0.2$.

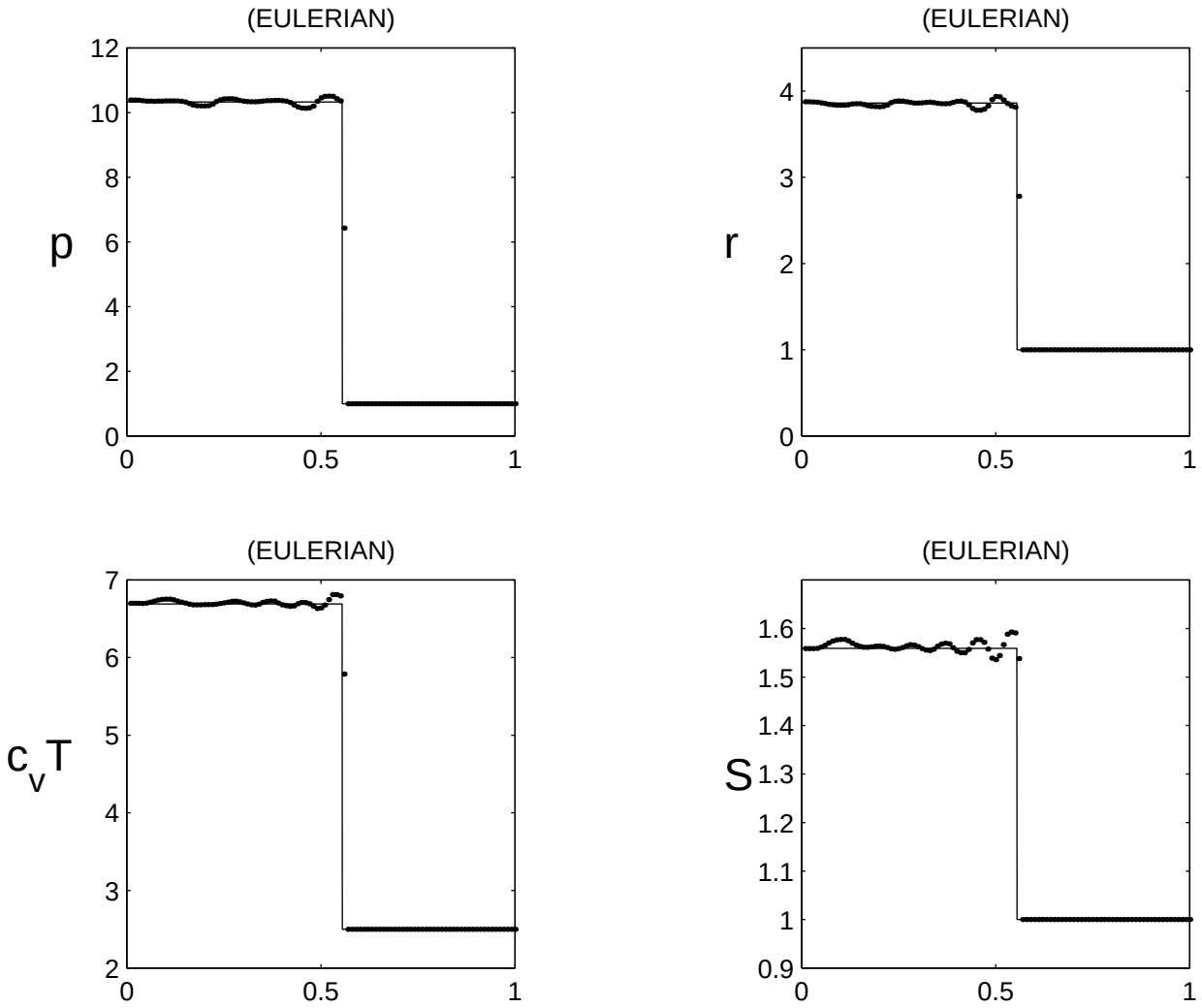

Figure 32: Test # 9: Slow-Moving shocks. Eulerian computation (dots) employing Godunov-MUSCL scheme based on Eq.(1), compared with exact solution (solid line). $t = 0.5$.

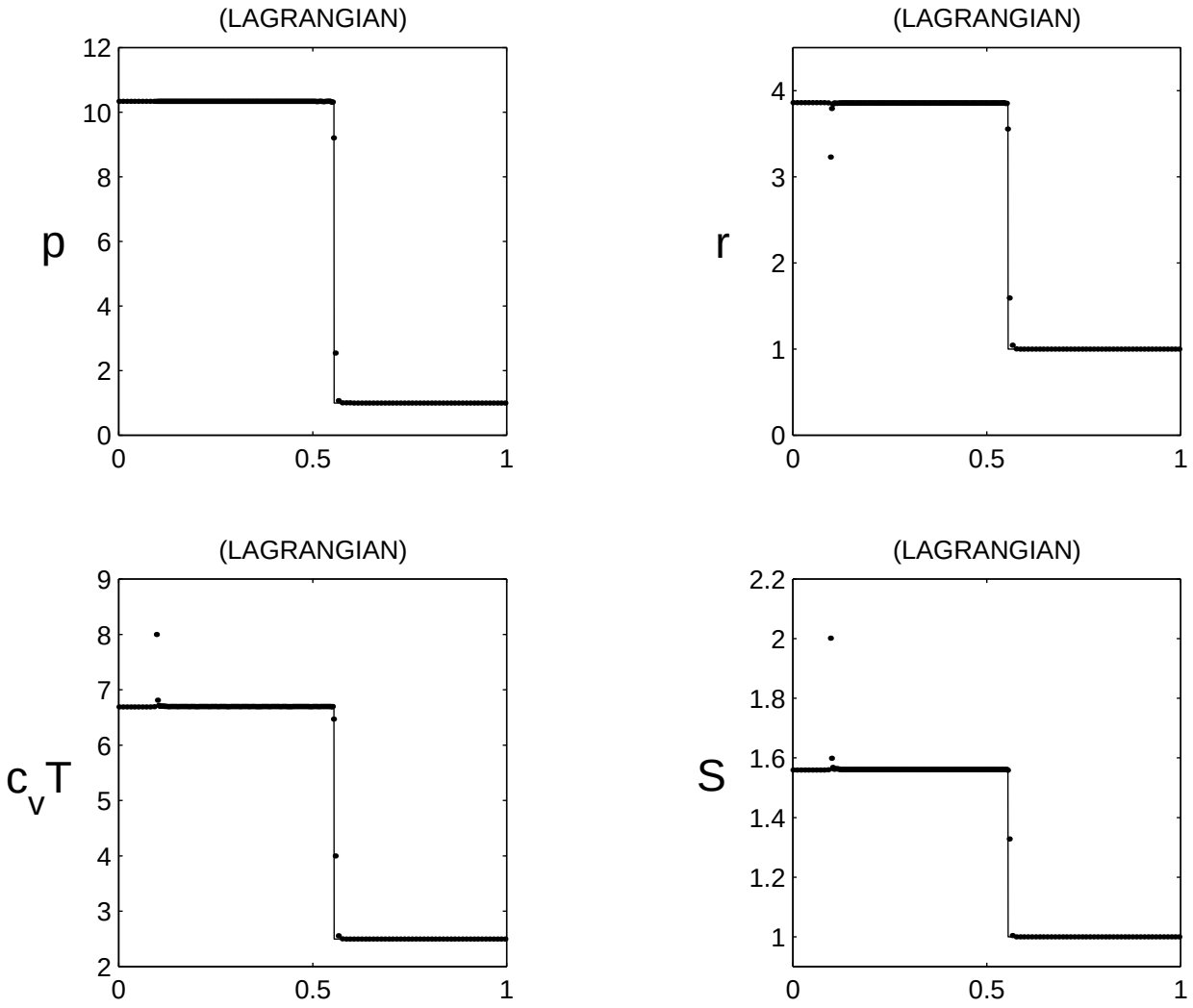

Figure 33: Test # 9: Slow-Moving shocks. Lagrangian computation (dots) employing Godunov-MUSCL scheme based on Eq.(3), compared with exact solution (solid line). $t = 0.5$.

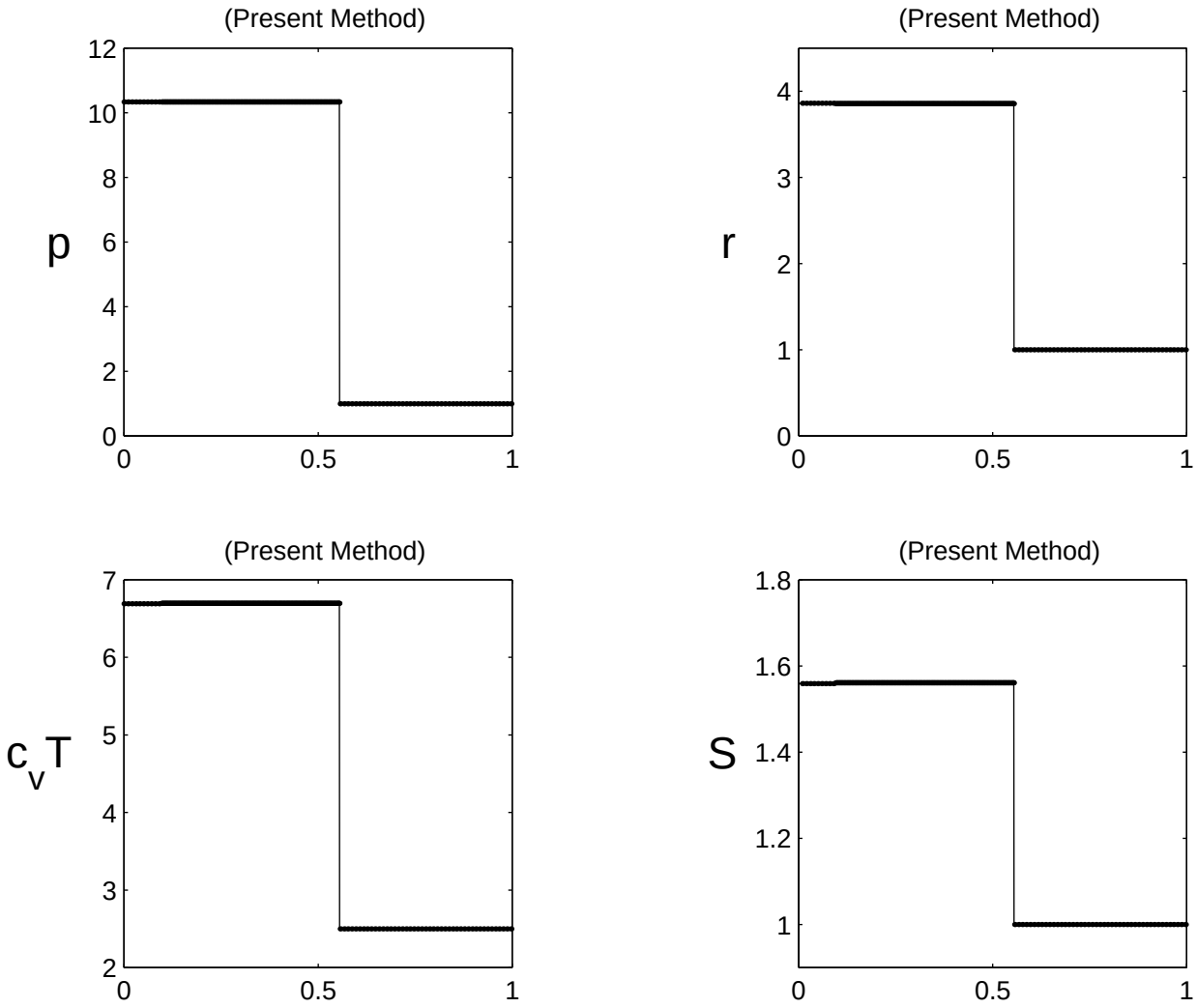

Figure 34: Test # 9: Slow-Moving shocks. Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL scheme based on Eq.(6), compared with exact solution (solid line). $t = 0.5$.

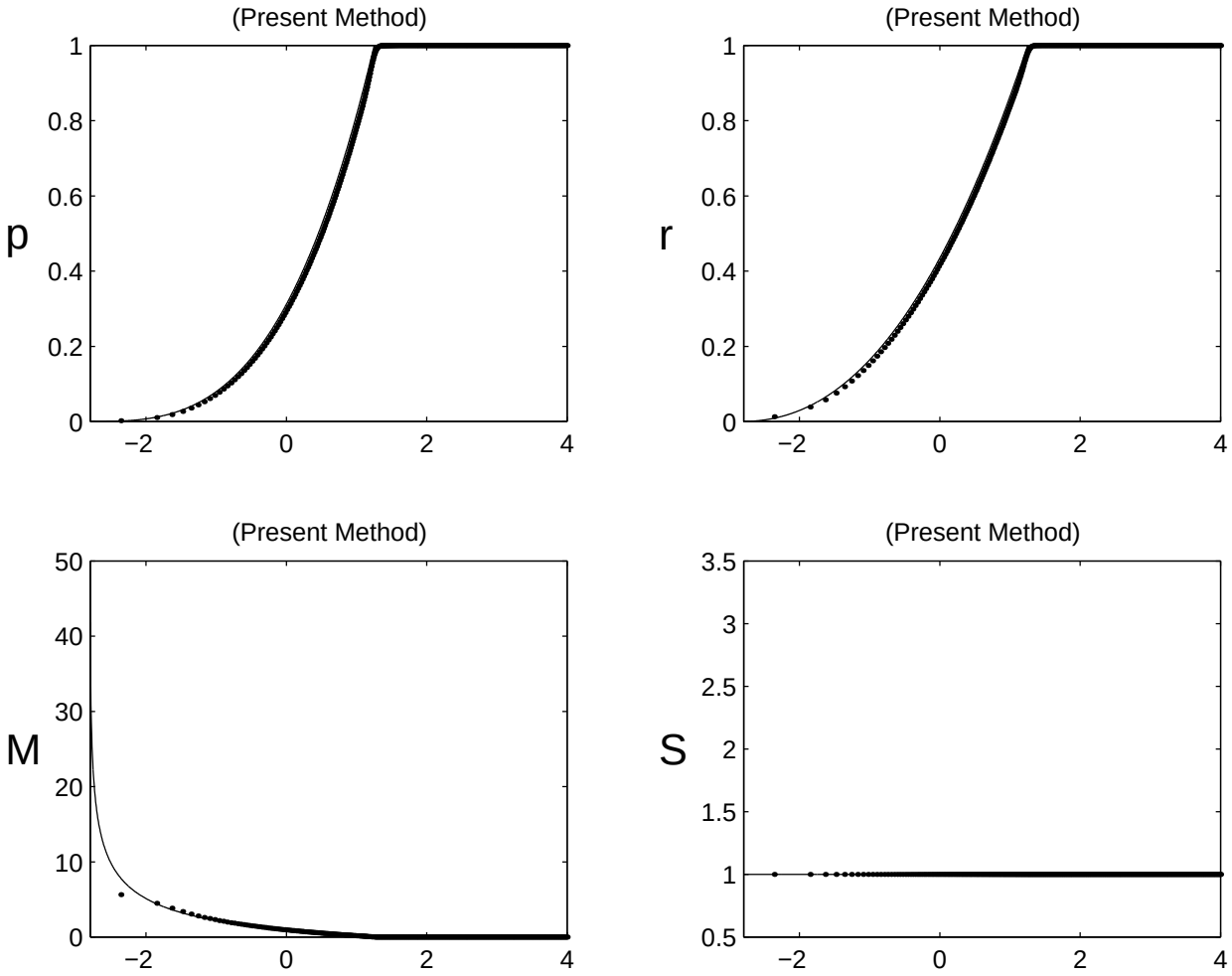

Figure 35: Test # 10: Low density flows. Lagrangian computation (dots) employing **shock-adaptive** Godunov-MUSCL scheme based on Eq.(6), compared with exact solution (solid line). $t = 1.057$.

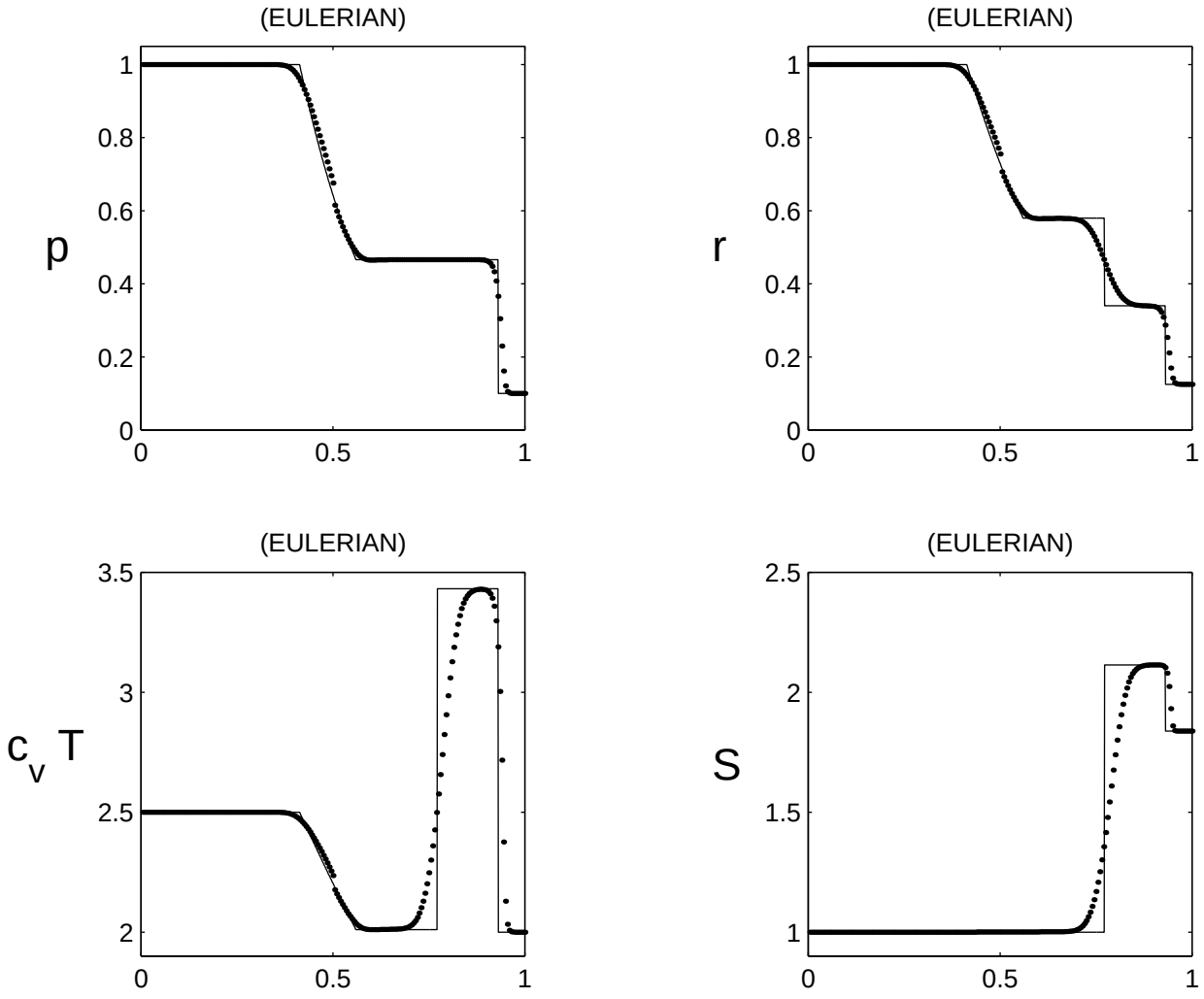

Figure 36: Test # 11: A Riemann problem showing the sonic point glitch. Eulerian computation (dots) employing Godunov scheme based on Eq.(1), compared with exact solution (solid line).

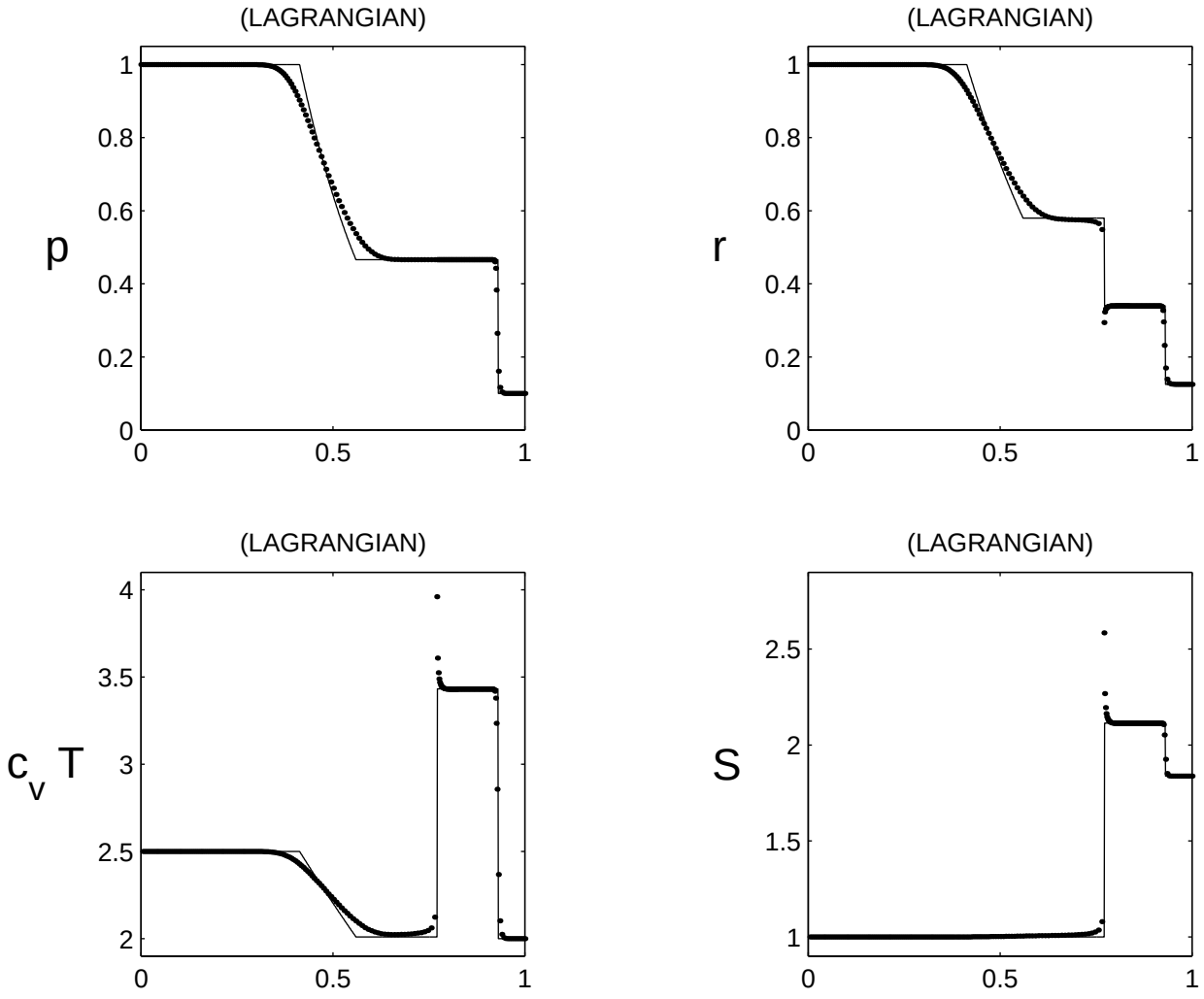

Figure 37: Test # 11: A Riemann problem showing the sonic point glitch. Lagrangian computation (dots) employing Godunov scheme based on Eq.(3), compared with exact solution (solid line).

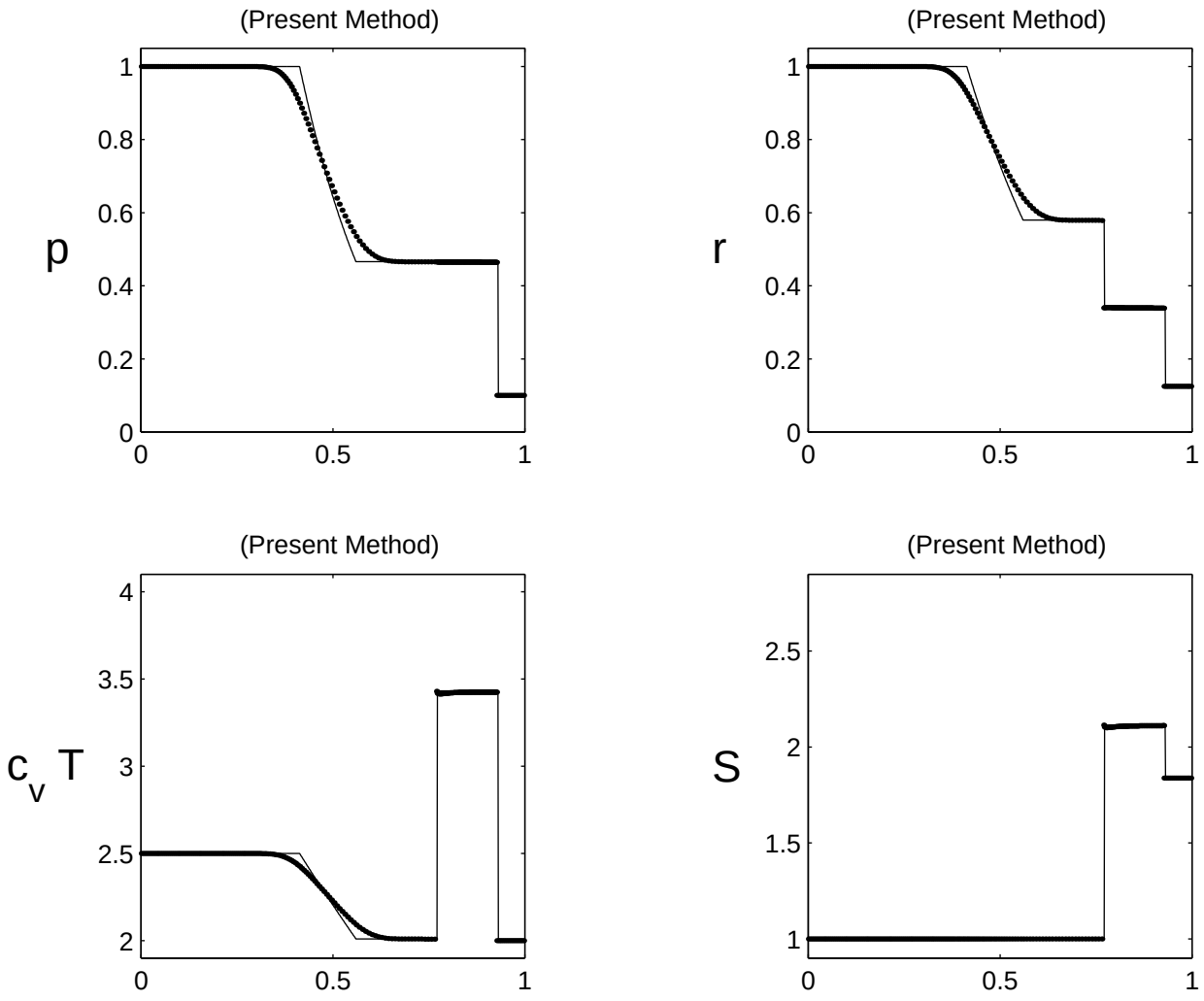

Figure 38: Test # 11: A Riemann problem showing the sonic point glitch. Eulerian computation (dots) employing **shock-adaptive** Godunov scheme based on Eq.(6), compared with exact solution (solid line).

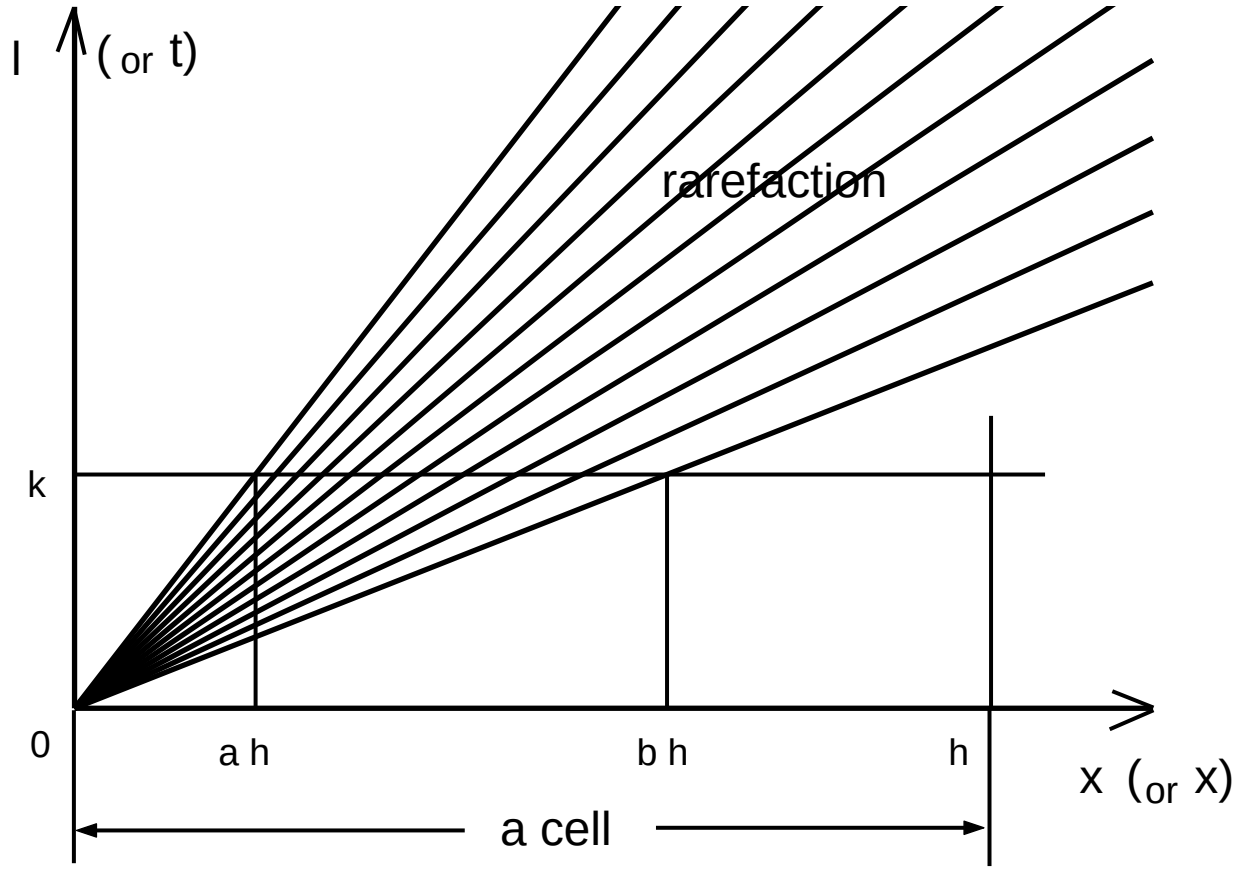

Figure 39: Sketch of a rarefaction wave showing notation.

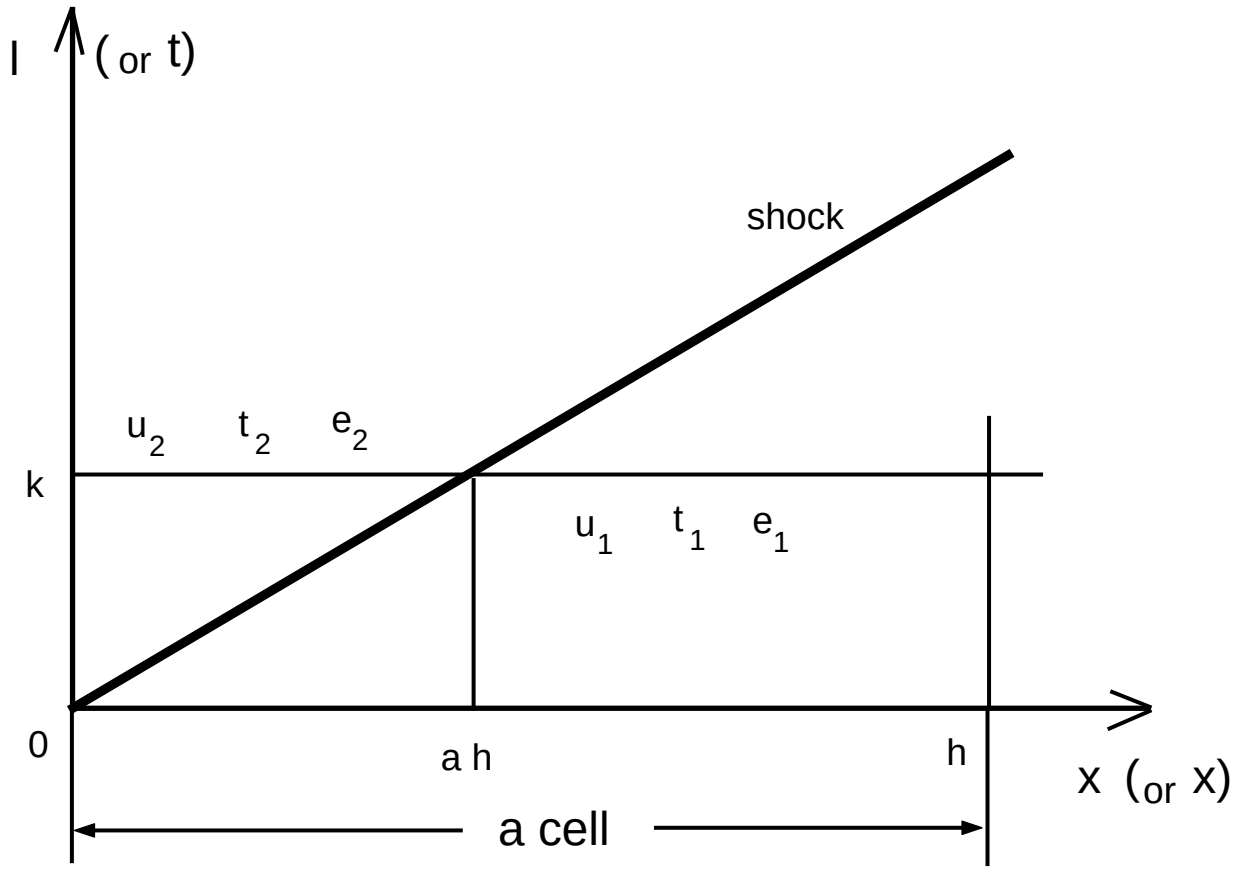

Figure 40: Sketch of a shock wave showing notation.

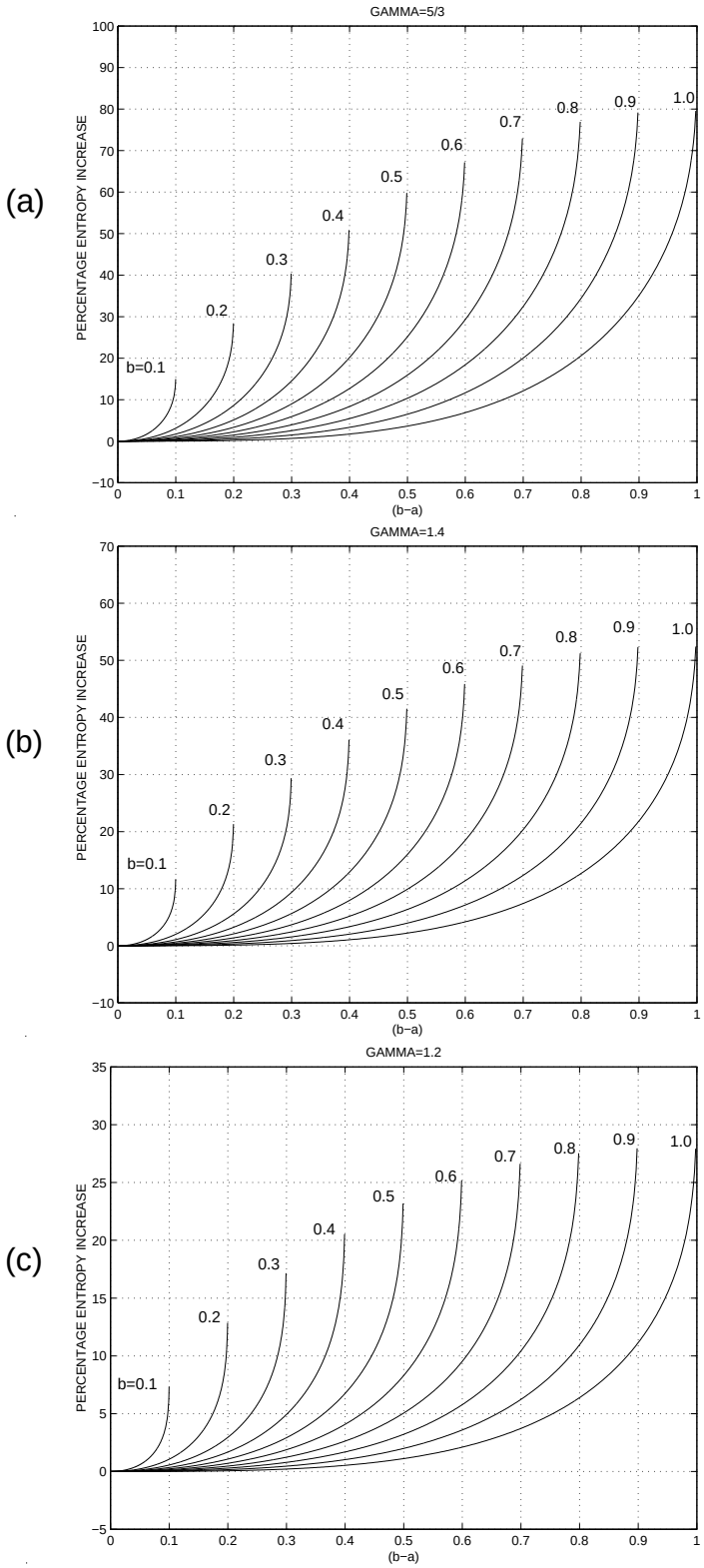

Figure 41: Percentage spurious entropy increase, S_r, given by (A.20) due to cell-averaging of the conserved variables across a rarefaction wave in a Lagrangian cell for **(a)** $\gamma = 5/3$, **(b)** $\gamma = 1.4$, **(c)** $\gamma = 1.2$

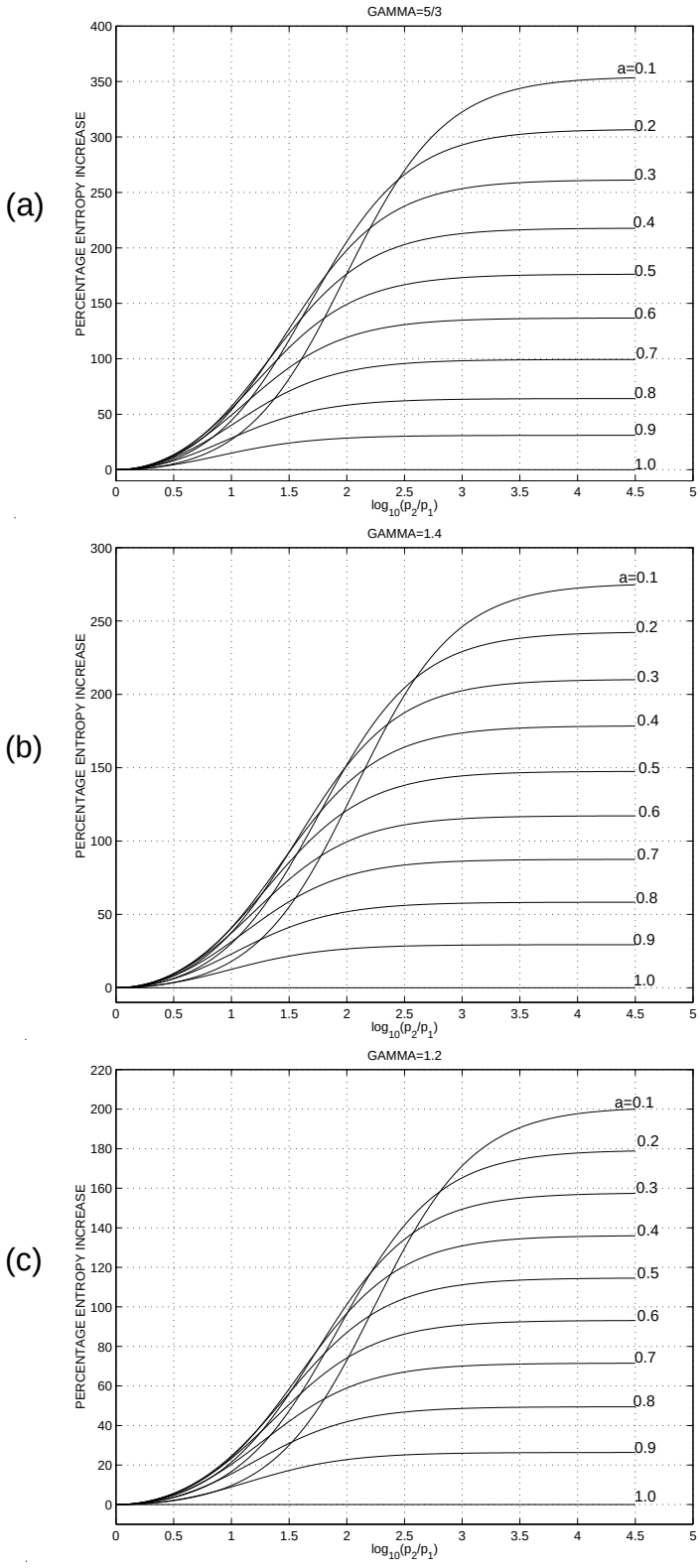

Figure 42: Percentage spurious entropy increase, S_s, given by (A.28) due to cell-averaging of the conserved variables across a shock wave in a Lagrangian cell for **(a)** $\gamma = 5/3$, **(b)** $\gamma = 1.4$, **(c)** $\gamma = 1.2$