The random projection method for hyperbolic conservation laws with stiff reaction terms

Weizhu Bao *and Shi Jin †
School of Mathematics, Georgia Institute of Technology
Atlanta, Georgia, GA 30332, USA

Abstract

In this paper we propose the random projection method for numerical simulations of the hyperbolic conservation laws with stiff source terms arising from chemically reactive flows:

$$U_t + F(U)_x + G(U)_y = \frac{1}{\varepsilon}\Psi(U).$$

In this problem, the chemical time scales may be orders of magnitude faster than the fluid dynamical time scales, making the problem numerically stiff. A classic spurious numerical phenomenon, the incorrect propagation speeds of discontinuities, occurs in underresolved numerical solutions. We introduce a random projection method for the reaction term by replacing the ignition temperature with a uniformly distributed random variable. The statistical average of this method corrects the spurious shock speed, as will be proved with a scalar model problem and demonstrated by a wide range of numerical examples in inviscid denotation waves in both one and two space dimensions.

1 Introduction

Hyperbolic systems with stiff source terms arise, among many other applications, in the modeling of chemically reactive flows. Restricting our attention to inviscid flows, the underlying physical equations are described by the compressible Euler equations of gas dynamics, coupled with source terms representing the chemical reaction. In two space dimensions these equations take the form

$$U_t + F(U)_x + G(U)_y = \frac{1}{\varepsilon}\Psi(U), \tag{1.1}$$

where U is the vector of dependent variables with components mass, momentum, total energy, and densities or concentration for each species in the reacting mixture respectively.

*Email address: wbao@math.gatech.edu. On leave from: Department of Applied Mathematics, Tsinghua University, Beijing 100084, P. R. China

†Email address: jin@math.gatech.edu. Research supported in part by NSF grant No. DMS-9704957

The flux functions $F(U)$ and $G(U)$ describe the fluid dynamical convection while the source term $\Psi(U)$ arises from the chemistry of the reacting species. ε is the reaction time.

The kinetics equations often include reactions with widely varying time scales. Moreover, many of the chemical time scales may be orders of magnitude faster than the fluid dynamical time scales. This can lead to problems of stiffness. Even a stable (for example with an implicit source) scheme could lead to spurious numerical propagation speed when the reaction time scale is not properly resolved numerically. This phenomenon was first observed by Colella, Majda and Roytburd [6] who made a study of the limiting behavior with increasing stiffness for various model systems. In particular, they looked at the Euler equations coupled with a single chemistry variable representing the mass fraction of unburnt gas in a detonation wave. These waves have the structure of a fluid dynamic shock that raises the pressure to some peak value, followed immediately by a reaction zone that brings the pressure back down to a new equilibrium value. On coarse grids it is not possible to resolve this combustion spike and the best one can hope for is a single discontinuity linking the two equilibrium values and moving at the correct speed. They got the correct combustion spike and correct speed of discontinuities only under very fine grids, i.e. resolve the reaction scale (at least 30 grid points in the reaction zone). On the contrary, they observed that on coarse grids, i.e. underresolving the reaction scales, the numerical solutions are qualitatively incorrect. The computed solution consists of a weak detonation wave, in which all the chemical energy is released, followed by a fluid dynamic shock traveling more slowly. The reaction wave always travels at the speed of one grid point per time step, which is totally unphysical. Ben-Artzi [1] also observed the same phenomenon in solving reactive flows using the solution of the generalized Riemann problem. Using scalar models, Leveque and Yee [14] studied this spurious numerical phenomenon. They show that this phenomenon is due to the introduction of nonequilibrium values through numerical dissipation in the advection step, which triggers the reaction too early. See also [16].

Since the numerical viscosity is the cause of the incorrect speed, a natural idea is to avoid any numerical viscosity. In [4] Chorin introduced the random choice method for reacting flows, originated from the classical Glimm's scheme [12] for hyperbolic systems. Since the random choice method does not introduce any numerical viscosity, thus no spurious waves will occur. In [10], Engquist and Sjögreen proposed a temperature extrapolation method, which uses the extrapolated temperature value from outside the shock profile, and obtained the correct detonation speed.

Since numerical viscosity is an essential feature for modern high resolution shock capturing methods, it makes lots of sense to develop methods for reacting flows that, in stead of avoiding the numerical viscosity, make correct use of it. The random projection method proposed here is such a method. Our method consists of the typical two steps: solving the homogeneous hyperbolic conservation laws by a standard modern shock capturing method, and then a projection step for the stiff reaction term. In the underresolved regimes, where the time step is much bigger than the reacting time, all ODE solvers for the reaction term essentially reduce to a projection operator (will be called the **deterministic projection** in the paper), which projects the chemically nonequilibrium data into the equilibrium ones according to the relative value of the temperature versus the ignition temperature. It is this projection that causes the incorrect speed, since the chemically nonequilibrium temperature due to numerical viscosity, once above the ignition temperature, will trigger the chemical reaction too early, forcing the front to move. This is purely a grid effect. Since there is no correlation between the numerical shock location and the gird, the only robust way to handle this numerical difficulty is through a random method. Our idea is, in the projection step, to **replace the ignition temperature by a uniformly distributed random variable**, defined in a suitable domain. Upon a suitable choice of the sampling sequence, at each time step, the front either moves one grid point or does not move, but the statistical average yields the correct speed!

For the problem of model scalar conservation law with stiff source term, we prove that the first order random projection method can indeed capture the correct speed of discontinuity, i.e. obtain the correct jumps in the correct locations, with a first oder accuracy.

The random projection method clearly differs from the random choice method of Chorin [4]. The random choice method is based on the Godunov method using the Riemann solver. The random projection method uses the randomness only for the reaction term, while in the convection step, any modern shock capturing methods, including the efficient methods free of Riemann solvers and local characteristic decompositions, can be used.

The paper is organized as follows. In section 2 we first introduce the random projection method for a scalar model problem. We prove that this method, when the first order upwind or Lax-Friedrichs method is used for the nonlinear convection, yields the correct shock speed with an error of $O(h \log h)$, where h is the grid size. We also present numerical examples for this scalar model. Furthermore, we apply this method to problems with stiff Ricatti source. In section 3 we extend this method to the computation of two-dimensional scalar hyperbolic conservation laws with stiff source terms. Numerical results show the random projection method yield the correct speed of reaction front. In section 4 we extend this method to the computation of one dimensional detonation waves. Numerical examples for a variety of detonation waves are presented. In section 5 we extend this method to the computation of two-dimensional detonation waves. In section 6 we draw some conclusions.

2 One-dimensional scalar model problems

We first introduce the random projection method for a model problem, studied in [14] [11]. Consider the following hyperbolic conservation law with stiff source term

$$u_t + f(u)_x = \frac{1}{\varepsilon}(u-\alpha)(1-u^2), \qquad -1 < \alpha < 1\,, \tag{2.1}$$

with piecewise constant initial data

$$u(x,0) = u_0(x) = \begin{cases} 1, & x \le x_0, \\ -1, & x > x_0. \end{cases} \tag{2.2}$$

Here ε is the reaction time, f is a convex function of u, i.e. $f''(u) > 0$, and x_0 is a given point.

The source term in (2.1) admits three local equilibria, i.e., the unstable one α, and the stable ones ± 1. When the solution is at equilibria, the reaction term has no effect. Thus the exact solution is a shock discontinuity connecting $u = 1$ with $u = -1$ and propagating to the right with the speed determined by the Rankine-Hugoniot jump condition

$$s = \frac{1}{2}[f(1) - f(-1)]\,. \tag{2.3}$$

Namely,

$$u(x,t) = \begin{cases} 1, & \text{if } x \le x_0 + st, \\ -1, & \text{if } x > x_0 + st. \end{cases} \tag{2.4}$$

Note that the speed does not depend on the specific value of α. This is the key for the success of the random projection method, which we introduce next.

Let h be the spatial increment and k be the time step. The numerical solution is evaluated at the points (ih, nk), $i = 0, \pm 1, \pm 2, \cdots$, $n = 0, 1, 2, \cdots$. Let u_i^n approximate $u(ih, nk)$ and u^n be the solution vector of $u(\cdot, nk)$ at time $t = t_n = nk$. When the reaction term is resolved, i.e. $k = O(h) << \varepsilon$, any method which works well for the homogeneous hyperbolic conservation law still works well here. Here we are interested in the underresolved case, where $k = O(h) \ge \varepsilon$.

A standard numerical method, which allows an underresolved discretization, is the fractional step method that solves the homogeneous convection

$$u_t + f(u)_x = 0 \tag{2.5}$$

followed by the reaction step

$$u_t = \frac{1}{\varepsilon}(u - \alpha)(1 - u^2)\,. \tag{2.6}$$

Let $S_c(k)$ denote the discrete operator for the homogeneous system (2.5) over a time step of length k, and $S_r(k)$ be the numerical integrator for the ODE system (2.6). Then the fraction step method takes the form:

$$u^{n+1} = S_r(k)\; S_c(k)\; u^n\,. \tag{2.7}$$

One may use any high resolution shock capturing method for $S_c(k)$. Let $u^* = S_c(k)u^n$. When $k >> \varepsilon$, the solution of the ODE (2.6) approaches the equilibrium states ± 1 exponentially fast. Whether it approaches 1 or -1 depends on whether u is bigger or smaller than α [11]. Thus $S_r(k)$ becomes effectively the **deterministic projection** operator:

$$S_r(k): \quad u_j^{n+1} = \begin{cases} 1, & \text{if } u_j^* > \alpha\,; \\ -1, & \text{if } u_j^* \le \alpha\,. \end{cases} \tag{2.8}$$

This standard method was studied in [14], where it was found that, if $k >> \varepsilon$, the numerical shock moves one grid point per time step. This is the case when the smeared value of u after the convection step is above the critical value α, which will be projected into 1 by the projection step, forcing the shock to advance one grid point. If the smeared value of u is below α, u will always be projected to -1, and then the shock will not move at all.

2.1 The random projection method

The idea of the random projection method is to replace the unstable equilibrium, or the critical value, α, by a uniformly distributed random sequence $\theta_n \in (-1,1)$. Let $u^* = S_c(k)u^n$. We replace S_r by S_θ where

$$S_\theta(k): \quad u_j^{n+1} = \begin{cases} 1, & \text{if } u_j^* > \theta_n; \\ -1, & \text{if } u_j^* \le \theta_n\,. \end{cases} \tag{2.9}$$

In this method, one random value of θ_n will be selected per time step.

This idea is based on the observation that there is no correlation between the center of the shock and the grid. One hopes that the statistical average will automatically correct the wrong speed.

2.2 The choice of the random sequence θ_n

In [8], several sampling schemes were proposed and compared for the random choice method of Chorin [3, 4]. Here and in our practical computations, we always use van der Corput's sampling scheme. The merit of this scheme is that it produces an equidistributed sequence on the interval $[0,1]$ and among all known uniformly distributed sequences the deviation of van der Corput's sequence is minimal [5]. The detailed scheme is the following. Let $1 \le n = \sum_{k=0}^m i_k 2^k$, $i_k = 0,1$, be the binary expansion of the integer n. Then one gets a random number generator on $[0,1]$:

$$\vartheta_n = \sum_{k=0}^m i_k 2^{-(k+1)}, \qquad n = 1,2,\cdots. \tag{2.10}$$

We rescale it in order to get a random number generator θ_n on $[-1,1]$:

$$\theta_n = 2\vartheta_n - 1, \qquad n = 1,2,\cdots. \tag{2.11}$$

In order to get the error estimate of our random projection method, we recall some properties of the van der Corput's sampling sequence $\{\vartheta_n,\ n=1,2,\cdots\}$ [5]. It is rather illuminating to list the first few elements in it:

$$\begin{array}{llll} \vartheta_1 = \frac{1}{2}, & \vartheta_2 = \frac{1}{4}, & \vartheta_3 = \frac{3}{4}, & \vartheta_4 = \frac{1}{8}, \\ \vartheta_5 = \frac{5}{8}, & \vartheta_6 = \frac{3}{8}, & \vartheta_7 = \frac{7}{8}, & \vartheta_8 = \frac{1}{16}. \end{array}$$

One can see that

$$\vartheta_1 = \frac{1}{2}, \qquad \vartheta_n = \begin{cases} < \frac{1}{2}, & \text{if } n \text{ is even,} \\ > \frac{1}{2}, & \text{if } n > 1 \text{ is odd.} \end{cases}$$

In general, if one divides the unit interval into the subintervals $(r2^{-s},(r+1)2^{-s})$, $r = 0,1,\cdots,2^s-1$, then for each r there is exactly one n for which $2^s - 1 < n \le 2^{s+1}-1$ such that $\vartheta_n \in [r2^{-s},(r+1)2^{-s})$. Let

$$N\{j = n_1+1,\cdots,n_2;\ \vartheta_j \in I\} \tag{2.12}$$

denote the number of j, $n_1 < j \le n_2$, such that ϑ_j is contained in an interval $I \subset [0,1]$. Let

$$\delta(\vec{\vartheta}, n_1, n_2, I) \equiv \frac{1}{n_2 - n_1} N\{j = n_1 + 1, \cdots, n_2;\ \vartheta_j \in I\} - |I| \tag{2.13}$$

be the difference between the proportion of times that ϑ_j is contained in $I \subset [0,1]$ and $|I|$, the length of I. The following result from [5] is rather useful in our error estimate in the next subsection.

Lemma 2.1 *For the binary van der Corput's sampling sequence,*

$$\delta(\vec{\vartheta}, n_1, n_2, I) \le \frac{3\ln(2(n_2 - n_1)) + 1}{n_2 - n_1}, \qquad \forall \quad n_1,\ n_2,\ I. \tag{2.14}$$

The above inequality shows that the binary van der Corput's sampling sequence is equidistributed on the interval $[0,1]$ since $\lim_{n_2 \to \infty} \delta(\vec{\vartheta}, n_1, n_2, I) = 0$ for each fixed n_1, I. Thus the sequence $\{\theta_n,\ n = 1, 2, \cdots\}$ is equidistributed on the interval $[-1, 1]$.

2.3 An error estimate on the numerical shock speed

Now we will prove that our random projection method, when combined with the first order upwind method or the Lax-Friedrichs method for the convection term, can capture the correct location of discontinuities for the scalar model problem (2.1) with a first order accuracy. The proof is similar to the error estimate of Glimm's scheme to the homogeneous conservation laws [5].

First we consider the upwind method. Without loss of generality, we assume $f'(u) > 0$. Consider the first order upwind method for the convection, followed by the random projection:

$$S_c(k): \qquad u_j^* = u_j^n - \frac{k}{h}\left(f(u_j^n) - f(u_{j-1}^n)\right), \tag{2.15}$$

$$S_\theta(k): \qquad u_j^{n+1} = \begin{cases} 1, & \text{if } u_j^* > \theta_n, \\ -1, & \text{if } u_j^* \le \theta_n. \end{cases} \tag{2.16}$$

It is easy to see that the above method preserves the monotonicity property of the upwind method. Under the usual CFL condition, i.e. $\sup_{j,n} \left|f'(u_j^n)\right| k/h < 1$, the above algorithm is stable.

First, at any time step n, there is an $l(n) = j_0$, j_0 an integer, such that

$$u_j^n = \begin{cases} 1, & \text{if } j \le l(n), \\ -1, & \text{if } j > l(n). \end{cases} \tag{2.17}$$

Here we assume that x_0, the initial point of discontinuity, is a grid point, i.e. $x_0 = l(0)h$. After the convection step (2.15), one has

$$u_j^* = \begin{cases} 1, & \text{if } j \le l(n), \\ \mu, & \text{if } j = l(n) + 1, \\ -1, & \text{if } j > l(n) + 1; \end{cases} \tag{2.18}$$

where

$$1 > \mu = \frac{k}{h}(f(1) - f(-1)) - 1 = \frac{2k\ s}{h} - 1 \equiv 2\lambda - 1 > -1; \tag{2.19}$$

with $\lambda = ks/h$. In the second step, the random projection (2.16) gives

$$u_j^{n+1} = \begin{cases} 1, & \text{if } j \le l(n), \\ -1, & \text{if } j > l(n)+1; \end{cases} \qquad u_{l(n)+1}^{n+1} = \begin{cases} 1, & \text{if } \mu > \theta_n, \\ -1, & \text{if } \mu \le \theta_n. \end{cases} \tag{2.20}$$

Thus one has

$$l(n+1) = \begin{cases} l(n)+1, & \text{if } \mu > \theta_n, \\ l(n), & \text{if } \mu \le \theta_n. \end{cases} \tag{2.21}$$

Therefore the discontinuities in the approximate solution at each time step either moves by one grid point or does not move. We now examine the accumulative effect over many time steps. In fact, from (2.21), noting (2.19), (2.12) and (2.11), we get

$$\begin{aligned} l(n) &= l(0) + N\{j = 1, \cdots, n;\ \theta_j \in [-1, \mu)\} \\ &= l(0) + N\{j = 1, \cdots, n;\ \vartheta_j \in [0, (\mu+1)/2)\} \\ &= l(0) + N\{j = 1, \cdots, n;\ \vartheta_j \in [0, \lambda)\}. \end{aligned} \tag{2.22}$$

Combining (2.22), (2.17), (2.4), (2.13) and (2.14) with $n_1 = 0$, $n_2 = n$ and $I = [0, \lambda)$, we obtain

$$\begin{aligned} |x_0 + st - l(n)h| &= |st - hN\{j = 1, \cdots, n;\ \vartheta_j \in [0, \lambda)\}| \\ &= \left| st - nh[\lambda + \delta(\vec{\vartheta}, 0, n, [0, \lambda))] \right| \\ &= nh|\delta(\vec{\vartheta}, 0, n, [0, \lambda))| \le h(1 + 3\ln(2t/k)) \\ &= h(1 + 3\ln(2st/(\lambda h)). \end{aligned} \tag{2.23}$$

This shows the first order accuracy for the upwind method.

Next we consider the Lax-Friedrichs method followed by the random projection:

$$S_c(k): \qquad u_j^* = u_j^n - \frac{k}{2h}\left(f(u_{j+1}^n) - f(u_{j-1}^n)\right) + \frac{\sqrt{a}k}{2h}(u_{j+1}^n - 2u_j^n + u_{j-1}^n), \tag{2.24}$$

$$S_\theta(k): \qquad u_j^{n+1} = \begin{cases} 1, & \text{if } u_j^* > \theta_n, \\ -1, & \text{if } u_j^* \le \theta_n. \end{cases} \tag{2.25}$$

In (2.24), the positive constant $\sqrt{a} \ge |f'(u)|$ for all u. After the convection step (2.24), one has

$$u_j^* = \begin{cases} 1, & \text{if } j \le l(n)-1, \\ \mu_1, & \text{if } j = l(n), \\ \mu_2, & \text{if } j = l(n)+1, \\ -1, & \text{if } j > l(n)+1; \end{cases} \tag{2.26}$$

where

$$1 > \mu_1 = \frac{k}{h}(s - \sqrt{a}) + 1 > \mu_2 = \frac{k}{h}(s + \sqrt{a}) - 1 > -1\,. \tag{2.27}$$

In the second step, the random projection (2.25) gives

$$u_j^{n+1} = \begin{cases} 1, & \text{if } j \le l(n) - 1, \\ -1, & \text{if } j > l(n) + 1; \end{cases} \tag{2.28}$$

while

$$u_{l(n)}^{n+1} = \begin{cases} 1, & \text{if } \mu_1 > \theta_n, \\ -1, & \text{if } \mu_1 \le \theta_n. \end{cases} \qquad u_{l(n)+1}^{n+1} = \begin{cases} 1, & \text{if } \mu_2 > \theta_n, \\ -1, & \text{if } \mu_2 \le \theta_n. \end{cases} \tag{2.29}$$

Thus one has

$$l(n+1) = \begin{cases} l(n) - 1, & \text{if } \mu_1 < \theta_n, \\ l(n), & \text{if } \mu_2 < \theta_n \le \mu_1, \\ l(n) + 1, & \text{if } \mu_2 \ge \theta_n. \end{cases} \tag{2.30}$$

Therefore the discontinuities in the approximate solution at each time step could move to the left by one grid point, does not move, or to the right by one grid point. We now examine the accumulative effect over many time steps, which should take into account the possibilities of both moving forward and backward. By the definition of N in (2.12), we have

$$\begin{aligned} l(n) &= l(0) + N\{j = 1, \cdots, n;\ \theta_j \in [-1, \mu_2]\} - N\{j = 1, \cdots, n;\ \theta_j \in (\mu_1, 1]\} \\ &= l(0) + N\{j = 1, \cdots, n;\ \vartheta_j \in [0, \frac{\mu_2 + 1}{2}]\} - N\{j = 1, \cdots, n;\ \vartheta_j \in (\frac{\mu_1 + 1}{2}, 1]\} \\ &= l(0) + N\{j = 1, \cdots, n;\ \vartheta_j \in [0, \lambda_2] - N\{j = 1, \cdots, n;\ \vartheta_j \in (\lambda_1, 1]\}, \end{aligned} \tag{2.31}$$

where $\lambda_i = \frac{1}{2}(\mu_i + 1)$ for $i = 1, 2$. Using the definition of μ_1, μ_2 in (2.27), one obtains

$$\begin{aligned} &|x_0 + st - l(n)h| \\ &= |st - hN\{j = 1, \cdots, n;\ \vartheta_j \in [0, \lambda_2]\} + hN\{j = 1, \cdots, n;\ \vartheta_j \in (\lambda_1, 1]\}| \\ &= \left|st - nh[\lambda_2 + \delta(\vec{\vartheta}, 0, n, [0, \lambda_2])] + nh[1 - \lambda_1 + \delta(\vec{\vartheta}, 0, n, (\lambda_1, 1])]\right| \\ &= \left|snk - nk\frac{1}{2}(s + \sqrt{a}) - nk\frac{1}{2}(s - \sqrt{a}) - nh\delta(\vec{\vartheta}, 0, n, [0, \lambda_2]) + nh\delta(\vec{\vartheta}, 0, n, (\lambda_1, 1])\right| \\ &\le nh|\delta(\vec{\vartheta}, 0, n, [0, \lambda_2])| + nh|\delta(\vec{\vartheta}, 0, n, (\lambda_1, 1])| \le 2h(1 + 3\ln(2t/k)) \\ &= 2h(1 + 3\ln(2st/(\lambda h))). \end{aligned} \tag{2.32}$$

This shows the first order accuracy for the Lax-Friedrichs method.

The above analysis can be summarized by the following theorem:

Theorem 2.1 *The difference between the shock location of the exact solution, $x_0 + st$, and the numerical one, $l(n)h$, as determined by the random projection method*

$$u^{n+1} = S_\theta(k) S_c(k) u^n$$

where $S_c(k)$ is the first order upwind or Lax-Friedrichs method, has the following estimate:

$$|x_0 + st - l(n)h| \le Ch(1 + 3\ln(2st/(\lambda h))), \tag{2.33}$$

where t is a fixed time and $\lambda = sk/h$, $C = 1$ for the upwind method and $C = 2$ for the Lax-Friedrichs method.

From this inequality, one can see the error is first order ($O(h|\ln h|)$ for any fixed time t.

2.4 Numerical examples

Example 2.1 We choose $f(u) = \frac{u^2}{2} + u$, $\alpha = 0$ and $\varepsilon = 10^{-4}$ in (2.1) and $x_0 = 0.2$ in (2.2) and solve the corresponding problem on the interval $[0, 1]$ with 101 grid points by using the random projection method (2.15)-(2.16). The mesh size $h = 0.01$. For this example the speed of discontinuity $s = \frac{f(1)-f(-1)}{2} = 1$. Figure 1 shows the computed results at $t = 0.2$, 0.4, and 0.6 for different time steps $k = 0.004$, 0.001, and 0.0001.

Example 2.2 We choose $f(u) = e^u$, $\alpha = 0$ and $\varepsilon = 10^{-4}$ in (2.1) and $x_0 = 0.2$ in (2.2) and solve the corresponding problem on the interval $[0, 1]$ with 101 grid points. The mesh size $h = 0.01$. For this example the speed of discontinuity $s = (f(1) - f(-1))/2 = (e - e^{-1})/2$ which is an irrational number. Figure 2 shows the numerical solutions at $t = 0.2$, 0.4, and 0.6 for different time steps $k = 0.001$, and 0.0001.

Figures 1-2 show that the random projection method can really capture the correct speeds of the discontinuities for scalar hyperbolic conservation laws with stiff source terms when the reaction term is underresolved. The location of the shock may be off by few grid points at each time step, but such a deviation never grows in time.

A side issue is observed during our numerical experiments. If, instead of the random projection, one uses an explicit method for (2.1) but the spatial increment h does not resolve ε. In this case, even with the resolved time step $k << \varepsilon$, the numerical shock speed is still wrong. This indicates that the incorrect shock speed is induced by the spatial underresolution rather than the temporal one. To illustrate this pont, we solve the problem (2.1) by the following explicit method:

$$u_i^{n+1} = u_i^n - \frac{k(f(u_i^n) - f(u_{i-1}^n))}{h} + \frac{k}{\varepsilon}(u_i^n - \alpha)(1 - (u_i^n)^2). \tag{2.34}$$

For this method, one has to choose $k < \varepsilon$ for numerical stability, when $h >> \varepsilon$.

Example 2.3 we choose $f(u) = \frac{u^2}{2} + u$ and $\alpha = 0$ in (2.1) and $x_0 = 0.2$ in (2.2) and solve the corresponding problem on the interval $[0, 1]$ with 101 grid points by using the explicit method (2.34). The mesh size $h = 0.01$. For this example the speed of discontinuity $s = (f(1) - f(-1))/2 = 1$. Figure 3 shows the computed results for $\varepsilon = 0.1$ and $k = 0.001$, $\varepsilon = 0.01$ and $k = 0.0001$, $\varepsilon = 0.001$ and $k = 0.00001$, and $\varepsilon = 0.0001$ and $k = 0.000001$. We observe that: when the mesh resolves the reaction term, one can get the correct numerical solution. On the contrary, when the mesh underresolves the source terms, one can not get the correct solution even with the time step smaller than ε.

2.5 Hyperbolic equations with Ricatti sources

The random projection method can be applied for numerical simulations of hyperbolic conservation laws with stiff Ricatti sources:

$$u_t + f(u)_x = \frac{1}{\varepsilon} u(1 - u), \tag{2.35}$$

with piecewise constant initial data

$$u(x, 0) = u_0(x) = \begin{cases} 1, & x \le x_0, \\ 0, & x > x_0. \end{cases} \tag{2.36}$$

Here ε is a small parameter, f is a convex function of u, i.e. $f''(u) > 0$, and x_0 is a given point.

The Ricatti source in (2.35) admits two local equilibria, i.e., the unstable one 0, and the stable one 1 [17]. When the solution is at equilibria, the Ricatti source has no effect. Thus the exact solution is a shock discontinuity connecting $u = 1$ with $u = 0$ and propagating to the right with the speed determined by the Rankine-Hugoniot jump condition:

$$\bar{s} = f(1) - f(0)\,. \tag{2.37}$$

Namely,

$$u(x,t) = \begin{cases} 1, & \text{if } x \le x_0 + \bar{s}t, \\ 0, & \text{if } x > x_0 + \bar{s}t. \end{cases} \tag{2.38}$$

The idea of the random projection method is to replace the unstable equilibrium, 0, by the uniformly distributed random sequence $\vartheta_n \in (0,1)$, see (2.10) for detail. Let $u^* = S_c(k)u^n$ with $S_c(k)$ be the discrete operator for the homogeneous equation (2.5) over a time step. The random projection operator S_ϑ is defined as

$$S_\vartheta(k): \quad u_j^{n+1} = \begin{cases} 1, & \text{if } u_j^* > \vartheta_n; \\ 0, & \text{if } u_j^* \le \vartheta_n\,. \end{cases} \tag{2.39}$$

In this method, one random value of ϑ_n will be selected per time step. Combining with the convection step, we get

$$S_{rp}(k): \qquad u^{n+1} = S_\vartheta(k)S_c(k)u^n. \tag{2.40}$$

Similar to those in section 2.3, we can also get an error estimate on the numerical shock speed for problems with Ricatti source.

Theorem 2.2 *The difference between the shock location of the exact solution of the problem (2.35), $x_0 + \bar{s}t$, and the numerical one, $l(n)h$, as determined by the random projection method (2.40) in which $S_c(k)$ is the first order upwind or Lax-Friedrichs method, has the following estimate:*

$$|x_0 + \bar{s}t - l(n)h| \le Ch(1 + 3\ln(2\bar{s}t/(\bar{\lambda}h)), \tag{2.41}$$

where t is a fixed time and $\bar{\lambda} = \bar{s}k/h$, $C = 1$ for the upwind method and $C = 2$ for the Lax-Friedrichs method.

From this inequality, one can see the error is first order ($O(h|\ln h|)$ for any fixed time t.

Remark 2.1: When one considers the problem (2.35) with replacing the Ricatti source $\frac{1}{\varepsilon}u(1-u)$ by $\frac{1}{\varepsilon}u(1+u)$, our random projection method still works with replacing the sampling sequence ϑ_n by another sequence $\tilde{\theta}_n = \vartheta_n - 1$ which is equidistributed on the interval $[-1, 0]$.

3 Two-dimensional scalar model problems

The random projection method can be extended to two dimension in a straightforward way. Consider the following two-dimensional scalar hyperbolic conservation law with stiff reaction term

$$u_t + f(u)_x + g(u)_y = \frac{1}{\varepsilon}\psi(u) \equiv \frac{1}{\varepsilon}(u-\alpha)(1-u^2), \qquad -1 < \alpha < 1, \tag{3.1}$$

with piecewise constant initial data

$$u(x,y,0) = u_0(x,y) = \begin{cases} 1, & (x,y) \in \Omega_0 \subset \mathbb{R}^2, \\ -1, & (x,y) \in (\mathbb{R}^2 \setminus \Omega_0). \end{cases} \tag{3.2}$$

Here ε is the reaction time, f and g are convex functions of u, i.e. $f''(u) > 0$ and $g''(u) > 0$, and Ω_0 is a given domain in $\mathbb{R}^2$.

Let the grid points $(x_j, y_i) = (jh, ih)$, $j, i = \cdots, -1, 0, 1, \cdots$, with equal mesh spacing h. The time levels $t_n = nk$, $k = 0, 1, 2, \cdots$ are also uniformly spaced with time step k. Let $u^n_{j,i}$ be the approximate solution of u at $(x_j, y_i, t_n) = (jh, ih, nk)$. Similar to the one-dimensional scalar model problem, we use the fraction step method consist of the convection step for

$$u_t + f(u)_x + g(u)_y = 0. \tag{3.3}$$

followed by the reaction step.

$$u_t = \frac{1}{\varepsilon}(u-\alpha)(1-u^2)\,. \tag{3.4}$$

Let $S_c(k)$ be any high resolution shock capturing method for (3.3), and $u^* = S_c(k)u^n$. The random projection for the reaction is given by

$$S_\theta(k) : u^{n+1}_{j,i} = \begin{cases} 1, & \text{if } u^*_{j,i} > \theta_n, \\ -1, & \text{if } u^*_{j,i} \le \theta_n; \end{cases} \tag{3.5}$$

where

$$\theta_n = 2\vartheta_n - 1;$$

with ϑ_n (see (2.10) for detail) being the van der Corput's sampling sequence on the interval $[0, 1]$.

Our fractional step method is then

$$u^{n+1} = S_\theta(k)\ S_{f,g}(k)u^n. \tag{3.6}$$

The stability condition for this algorithm is the usual CFL condition determined by the convection part. The initial data is discretized as

$$u^0_{j,i} = \begin{cases} 1, & (x_j, y_i) = (jh, ih) \in \bar{\Omega}_0, \\ -1, & \text{otherwise}. \end{cases} \tag{3.7}$$

Now we will test the above algorithm (3.6) with two numerical examples. In our numerical computations in this section, the operator $S_{f,g}(k)$ is the first order upwind scheme using dimensional splitting.

Example 3.1 We choose $f(u) = g(u) = u$, $\alpha = 0$ and $\varepsilon = 10^{-6}$ in (3.1). The initial condition is

$$u(x, y, 0) = u_0(x, y) = \begin{cases} 1, & x + y \le 1, \\ -1, & x + y > 1. \end{cases} \tag{3.8}$$

This problem is solved on the domain $[0, 1]^2$ with 101×101 grid points using the 2d random projection method (3.6). The mesh size $h = 0.01$. Dirichlet boundary conditions are used at inflow boundary ($x = 0$, and $y = 0$), and outflow boundary conditions are used on the other two sides of the boundary. For this example the speed of discontinuity is 2 in both x and y directions, see [9]. Figure 4 shows the computed results of the discontinuity front (i.e. contour of $u = 0$) at $t = 0.2$, 0.4, and 0.45 for time steps $k = 0.005$ and 0.001, respectively.

Example 3.2 Now consider the nonlinear convection with reaction

$$u_t + (u^2/2)_x + (u^2/2)_y = \frac{1}{\varepsilon} u(1 - u)(u - 1/2), \tag{3.9}$$

with initial condition

$$u(x, y, 0) = u_0(x, y) = \begin{cases} 1, & x^2 + y^2 \le 0.16, \\ 0, & x^2 + y^2 > 0.16. \end{cases} \tag{3.10}$$

This problem is interesting because of the nonlinear flux functions and the curved shock front. The equilibria are different from those of (3.1), so we have to use the van der Coput's sampling sequences $\theta_n = \vartheta_n$. We choose $\varepsilon = 10^{-6}$ and solve this problem on the domain $[0, 1]^2$ with 201×201 grid points using the 2d random projection method (3.6). So the mesh size $h = 0.005$. Homogeneous Neumann boundary conditions are used at the inflow boundary ($x = 0$, and $y = 0$), while outflow boundary conditions are imposed on the other sides of the boundary. For this example the shock front is moving with speed $= 0.5$ along the axises [9]. Figure 5 shows the computed results of the discontinuity front (i.e. contour of $u = 0.5$) at $t = 0.4$, 0.8, and 1.0 for time steps $k = 0.0025$ and 0.001, respectively.

Figures 4-5 show that the random projection method is able to capture the correct front speed in 2d even when the reaction time is not spatially and temporally resolved.

4 One-dimensional detonation waves

In this section, we introduce the random projection method for the computation of one dimensional detonation waves. The key idea is to **make the ignition temperature random**.

Consider the Euler equations that model the time-dependent flow of an inviscid, compressible, reactive gas in one space dimension. Without heat conduction and viscosity, the equations take the form

$$U_t + F(U)_x = \frac{1}{\varepsilon} \Psi(U), \tag{4.1}$$

$$U = \begin{pmatrix} \rho \\ m \\ e \\ \rho z \end{pmatrix}, \quad F(U) = \begin{pmatrix} m \\ m^2/\rho + p \\ m(e+p)/\rho \\ mz \end{pmatrix}, \quad \Psi(U) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\rho z e^{-T_0/T} \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \\ 0 \\ \psi(U) \end{pmatrix}.$$

The dependent variables $\rho(x,t)$, $m(x,t)$, $e(x,t)$ and $z(x,t)$ are the density, momentum, total energy and the fraction of unreacted fluid respectively. The pressure is given by

$$p = (\gamma - 1)\left(e - \frac{m^2}{2\rho} - q_0 \rho z\right)$$

and the temperature is defined as $T = p/\rho$. Let $u = m/\rho$ be the velocity. The parameters q_0, T_0, γ and ε correspond to chemical heat release, ignition temperature, c_p to c_v ratio, and reaction time.

We will focus on the discontinuous solutions of detonation waves. For these waves the viscosity is not as important as for the slower deflagration wave solutions.

We refer to (4.1) as the reactive Euler equations with Arrhenius kinetics. We will also consider (4.1) with

$$-\frac{1}{\varepsilon}\rho z e^{-T_0/T}$$

replaced by the Heaviside kinetics

$$-\frac{1}{\varepsilon}\rho z H(T - T_0),$$

where $H(x) = 1$ for $x > 0$ and $H(x) = 0$ for $x < 0$. Actually the stiffness issue with the Heaviside kinetics is the more severe one [10].

Now we will describe the random projection method to the problem (4.1) with piecewise constant initial data

$$(\rho(x,0), u(x,0), p(x,0), z(x,0)) = \begin{cases} (\rho_l, u_l, p_l, 0), & \text{if } x \le x_0, \\ (\rho_r, u_r, p_r, 1), & \text{if } x > x_0; \end{cases} \tag{4.2}$$

where x_0 is a given point and for simplicity these data are chosen such taht the detonation moves to the right. Initial data for other cases, e.g. detonation moves to the left, can be dealt with straightforwardly. Let $U_j^n = (\rho_j^n, m_j^n, e_j^n, (\rho z)_j^n)$ be the approximate solution of $U = (\rho, m, e, \rho z)$ at $(x_j, t_n) = (jh, nk)$. Similar to the scalar model problem, a fractional step method is used. In the first step, the inviscid compressible Euler equations

$$U_t + F(U)_x = 0, \tag{4.3}$$

are solved using a standard shock capturing method $S_F(k)$. Since any shock capturing method will have a few grid points in the shock profile, and corresponding temperature values, once above the critical temperature T_0, may trigger a too early chemical reaction, causing non-physical waves [6] [1].

4.1 Random projection methods for the reaction term

In the second step of the fractional step method, we solve the chemical reaction term

$$\rho_t = 0, \qquad m_t = 0, \qquad e_t = 0, \qquad (\rho z)_t = \frac{1}{\varepsilon}\psi(U)\,, \tag{4.4}$$

using the random projection method. Here the ignition temperature T_0 is made random. This can be successful since the speed of the front does not depend on the specific value of T_0, as far as it is in the range between the left and right states. Let $U^* = S_c(k)U^n$. Our first choice for the random projection operator S_θ is defined as

$$\begin{aligned} S_\theta(k): \qquad & \rho_j^{n+1} = \rho_j^*, \qquad m_j^{n+1} = m_j^*, \qquad e_j^{n+1} = e_j^*, \\ & z_j^{n+1} = \begin{cases} 0, & \text{if } T_j^* > \theta_n, \\ 1, & \text{if } T_j^n \le \theta_n; \end{cases} \end{aligned} \tag{4.5}$$

where

$$\theta_{n+1} = (T_l - T_r)\vartheta_{n+1} + T_r, \qquad T_l = p_l/\rho_l, \qquad T_r = p_r/\rho_r;$$

with ϑ_n (see (2.10) for detail) being the van der Corput's sampling sequence on the interval $[0, 1]$.

The combination of the two steps gives

$$S_1(k): \qquad U^{n+1} = S_\theta(k)\; S_c(k)\, U^n\,, \tag{4.6}$$

which will be referred to as the **global random projection method**, since the projection operator $S_\theta(k)$ is defined for all grid points.

Since the reaction zone is local, it makes more sense to do the random projection around the reaction zone. In fact, the global projection could produce incorrect solution in certain cases, as will be shown by a numerical example later in this section. Below we describe a **local random projection method**, which performs the random projection near the detonation front. The location of the front can be easily determined from the value of z.

First, notice that, although z may have some intermediate values between 0 and 1 after the convection step, the projection step always makes z either 0 or 1. Therefore, at any time step t_n, there is an $l(n) = j_0$, j_0 an integer, such that

$$z_j^n = \begin{cases} 0, & \text{if } j \le l(n), \\ 1, & \text{if } j > l(n). \end{cases}$$

Here $l(n)$ is the location of the jump for z in the approximate solution at time t_n. Since under this algorithm, the detonation will either move one grid point or does not move, the new location of the detonation front is at most one grid to the left away from that of the previous time step. Thus one only needs to check the value of z at three points, centered at $l(n)$, for the value of $l(n+1)$. Thus (4.5) can be replaced by the following local projection around the front:

$$\tilde{S}_\theta(k): \qquad \rho_j^{n+1} = \rho_j^*, \qquad m_j^{n+1} = m_j^*, \qquad e_j^{n+1} = e_j^*,$$

$$
\begin{aligned}
&\text{Set } l(n+1) := l(n) - 1, && (4.7)\\
&\text{For } \; j = l(n)-1, \cdots, \quad \text{do}: \;\; l(n+1) = j, \quad \text{if } T_j^* > \theta_n;\\
&\qquad z_j^{n+1} = \begin{cases} 0, & \text{if } j \le l(n+1),\\ 1, & \text{if } j > l(n+1). \end{cases} && (4.8)\\
&\text{End if}
\end{aligned}
$$

Then the corresponding algorithm is

$$
S_2(k): \qquad U^{n+1} = \tilde{S}_\theta(k)\; S_c(k) U^n. \tag{4.9}
$$

For numerical comparison, we also mention the **deterministic** method using the given deterministic ignition temperature T_0. In the case of Heaviside kinetics, an implicit method for (4.4) gives

$$
\begin{aligned}
S_r(k): \quad & \rho_j^{n+1} = \rho_j^*, \qquad m_j^{n+1} = m_j^*, \qquad e_j^{n+1} = e_j^*,\\
& (\rho z)_j^{n+1} = \begin{cases} (\rho z)_j^*, & \text{if } T_j^{n+1} \le T_0,\\ (\rho z)_j^*/(1+k/\varepsilon), & \text{if } T_j^{n+1} > T_0. \end{cases}
\end{aligned} \tag{4.10}
$$

Since ρz decreases when T becomes larger than T_0, one gets the same result if the test $T_j^{n+1} > T_0$ is replaced by $T_j^* > T_0$ [10], thereby making it unnecessary to solve any nonlinear equations. Then the corresponding algorithm is

$$
S_3(k): \qquad U^{n+1} = \bar{S}_r(k)\; S_F(k) U^n; \tag{4.11}
$$

where

$$
\begin{aligned}
\bar{S}_r(k): \quad & \rho_j^{n+1} = \rho_j^*, \qquad m_j^{n+1} = m_j^*, \qquad e_j^{n+1} = e_j^*,\\
& (\rho z)_j^{n+1} = \begin{cases} (\rho z)_j^*, & \text{if } T_j^* \le T_0,\\ (\rho z)_j^*/(1+k/\varepsilon), & \text{if } T_j^* > T_0. \end{cases}
\end{aligned} \tag{4.12}
$$

The stability condition for all three algorithms is the usual CFL condition determined by the convection terms.

4.2 Numerical results for detonation wave problems

Now we will test the above three algorithms with a variety of numerical examples, including the Chapman-Jouguet (C-J) detonation and strong detonations. In our computations, we always use the Heaviside kinetics. Among the numerical examples in this and next sections, the convection step is solved using the second order TVD relaxed scheme [13], which is free of Riemann solver and local characteristic decomposition.

Example 4.1 A Chapman-Jouguet (C-J) detonation

As a test case we choose here the case of ozone decomposition C-J detonation discussed and computed in [6] and [1]. We use CGS units and the following parameter values:

$$
\gamma = 1.4, \qquad q_0 = 0.5196 \times 10^{10}, \qquad \frac{1}{\varepsilon} = K = 0.5825 \times 10^{10}, \qquad T_0 = 0.1155 \times 10^{10}.
$$

The initial data are taken as the piecewise constant data defining a C-J detonation as a single wave (recall that in the Chapman-Jouguet model a C-J detonation corresponds to a sonic detonation, or, in other words, a sharp reaction wave that moves at minimal speed relative to the unburnt gas, see [7] for more details). The initial state was given by

$$(\rho, u, p, z) = \begin{cases} (\rho_0, u_0, p_0, 1), & \text{if } x > 0.005, \\ (\rho_{CJ}, u_{CJ}, p_{CJ}, 0), & \text{if } x \leq 0.005; \end{cases}$$

where $p_0 = 8.321 \times 10^5$, $\rho_0 = 1.201 \times 10^{-3}$, $u_0 = 0$, and $p_{CJ} = 6.270 \times 10^6$, $\rho_{CJ} = 1.945 \times 10^{-3}$, $u_{CJ} = 4.162 \times 10^4$. In fact, for any given initial state $(\rho_0, u_0, p_0, 1)$ on the right, one can obtain the C-J initial state on the left by the following formulas [4], [7]:

$$p_{CJ} = -b + (b^2 - c)^{1/2}, \tag{4.13}$$

$$\rho_{CJ} = \frac{\rho_0[p_{CJ}(\gamma+1) - p_0]}{\gamma p_{CJ}}, \tag{4.14}$$

$$D_{CJ} = [\rho_0 u_0 + (\gamma p_{CJ} \rho_{CJ})^{1/2}]/\rho_0, \tag{4.15}$$

$$u_{CJ} = D_{CJ} - (\gamma p_{CJ}/\rho_{CJ})^{1/2}; \tag{4.16}$$

where $b = -p_0 - \rho_0 q_0(\gamma - 1)$, $c = p_0^2 + 2(\gamma - 1)p_0\rho_0 q_0/(\gamma + 1)$, and D_{CJ} is the speed of the sharp front (in a C-J detonation).

Observe that the values p_{CJ}, ρ_{CJ} and u_{CJ} depend only on p_0, u_0, ρ_0 and q_0 (and not on K, i.e. ε or T_0). The speed of the sharp front in above example is $D_{CJ} = 1.088 \times 10^5$. In this example the width of the reaction zone is approximately 5×10^{-5} [1] and [6].

This problem is solved on the interval $[0, 0.05]$. The 'exact' solutions are obtained by using the deterministic method (4.11) and choosing $h = 5 \times 10^{-6}$ (i.e. 10001 grid points on the interval $[0, 0.05]$) and $k = 5 \times 10^{-12}$. These mesh size and time step resolve both the dynamical scale and the chemical scale. Now we compare the results obtained by the random projection methods and the deterministic method when the chemical scale reaction time is underresolved.

Figure 6 shows the numerical solutions by using the global random projection method (4.6) with $h = 5 \times 10^{-4}$ (i.e. 101 grid points for the interval $[0, 0.05]$) and $k = 5 \times 10^{-10}$ at time $t_1 = 5 \times 10^{-8}$, $t_2 = 1 \times 10^{-7}$ and $t_3 = 2 \times 10^{-7}$. Figure 7 shows the numerical solutions by using the deterministic method (4.11) with $h = 5 \times 10^{-4}$ and $k = 5 \times 10^{-10}$ at time $t_1 = 5 \times 10^{-8}$, $t_2 = 1 \times 10^{-7}$ and $t_3 = 2 \times 10^{-7}$. Figure 8 shows the numerical solutions by using the deterministic method (4.11) with $h = 5 \times 10^{-4}$ and $k = 5 \times 10^{-12}$ at time $t = t_1 = 5 \times 10^{-8}$, $t = t_2 = 1 \times 10^{-7}$ and $t = t_3 = 2 \times 10^{-7}$. In this figure, the time step resolves the reaction time but the spatial discretization does not resolve the reaction zone.

In Figure 6, the global random projection method captures the correct speed of the C-J detonation discontinuity with an underresolved discretization. There are some mild statistical fluctuations that contributes to the post front oscillations, which is the natural of a random method. Figures 7-8 show that, when the reaction zone is underresolved spatially, no matter how small the time step is, one cannot obtain the correct speed of discontinuity by using the explicit or implicit deterministic method for the chemical terms. A spurious weak detonation appears ahead of the detonation wave. Thus the stiffness problem is due to the spatial underresolution rather than the temporal one.

Example 4.2 A strong detonation

The initial data in this example is similar to those in the example 4.1 (i.e. γ, q_0, K and T_0 are the same) except that we change the initial state data as

$$(\rho, u, p, z) = \begin{cases} (\rho_0, u_0, p_0, 1), & \text{if } x > 0.005, \\ (\rho_l, u_l, p_l, 0), & \text{if } x \le 0.005; \end{cases}$$

where $u_l = 9.162 \times 10^4 > u_{CJ}$, $\rho_l = \rho_{CJ}$, $p_l = p_{CJ}$ and p_0, u_0, ρ_0, p_{CJ}, u_{CJ} and ρ_{CJ} are the same as those in example 4.1. In this case there is a right moving strong detonation [4], a right moving contact discontinuity, and a stationary shock.

The 'exact' solution is obtained similarly as that in the example 4.1. Figures 9-10 show the numerical solutions for this example by the local random projection method and the deterministic method, respectively. The local random projection method can capture the correct speed of the discontinuity of the strong detonation wave when underresolving the chemical reaction time. However, the deterministic method produces spurious waves.

Example 4.3 Another strong detonation

The initial data is similar to those in the example 4.1 except that we change the initial state data as

$$(\rho, u, p, z) = \begin{cases} (\rho_0, u_0, p_0, 1), & \text{if } x > 0.005, \\ (\rho_l, u_l, p_l, 0), & \text{if } x \le 0.005; \end{cases}$$

where $u_l = 9.162 \times 10^4 > u_{CJ}$, $\rho_l = \rho_{CJ}$, $p_l = 8.27 \times 10^6 > p_{CJ}$ and p_0, u_0, ρ_0, p_{CJ}, u_{CJ} and ρ_{CJ} are the same as those in the example 4.1. The wave pattern in this example is the same as that in example 4.2. The 'exact' solution is obtained similarly as in example 4.1. Figures 11-12 show the numerical solutions for this example by the local random projection method and the deterministic method, respectively.

Example 4.4 The initial data is similar to those in the example 4.1 except that we change the initial data as

$$(\rho, u, p, z) = \begin{cases} (\rho_0, u_0, p_0, 1), & \text{if } x > 0.005, \\ (\rho_l, u_l, p_l, 0), & \text{if } x \le 0.005; \end{cases}$$

where $u_l = u_{CJ}$, $\rho_l = \rho_{CJ}$, $p_l = 8.27 \times 10^6 > p_{CJ}$ and p_0, u_0, ρ_0, p_{CJ}, u_{CJ} and ρ_{CJ} are the same as those in the example 4.1. The solution consists of a right moving strong detonation wave, a right moving contact discontinuity, and a left moving rarefaction wave. The 'exact' solution is obtained similarly to that in example 4.1.

Figure 13 shows the numerical solutions using the local random projection method (4.9) with $h = 5 \times 10^{-4}$ and $k = 5 \times 10^{-10}$ at time $t_1 = 5 \times 10^{-8}$, $t_2 = 1 \times 10^{-7}$, $t_3 = 2 \times 10^{-7}$. Figure 14 shows the numerical solutions by the deterministic method (4.11) with $h = 5 \times 10^{-4}$ and $k = 5 \times 10^{-10}$ at time $t_1 = 5 \times 10^{-8}$, $t_2 = 1 \times 10^{-7}$, $t_3 = 2 \times 10^{-7}$.

We also compute the solution with the global random projection method, and depict the numerical solutions in Figure 15. We take $h = 5 \times 10^{-4}$, and $k = 5 \times 10^{-10}$. The solutions at time $t_1 = 5 \times 10^{-8}$, $t_2 = 1 \times 10^{-7}$, $t_3 = 2 \times 10^{-7}$ are displayed. The solution is essentially wrong, due to the improper projection away from the detonation front.

The local random projection method can capture the correct speed of the discontinuity when the solution is underresolved. On the contrary, the global random projection algorithm (4.6) does not work in this case, although it has produced satisfactory results in our previous examples. In fact, for the previous examples, the two algorithms (4.6) and (4.9) give similar results.

In practice, we recommend to use the local random projection method (4.9), because they are more efficient and work well for all cases we considered.

5 Two-dimensional detonation waves

Now we extend the local random projection method to the computation of two dimensional detonation waves. The equations are

$$U_t + F(U)_x + G(U)_y = \frac{1}{\varepsilon}\Psi(U), \tag{5.1}$$

$$U = \begin{pmatrix} \rho \\ m \\ n \\ e \\ \rho z \end{pmatrix}, \quad F(U) = \begin{pmatrix} m \\ m^2/\rho + p \\ mn/\rho \\ m(e+p)/\rho \\ mz \end{pmatrix}, \quad G(U) = \begin{pmatrix} n \\ mn/\rho \\ n^2/\rho + p \\ n(e+p)/\rho \\ nz \end{pmatrix}, \quad \Psi(U) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \psi(U) \end{pmatrix}.$$

The dependent variables $\rho(x,t)$, $m(x,t)$, $n(x,y,t)$, $e(x,t)$ and $z(x,t)$ are the density, x- and y-momentum, energy and the fraction of unreacted fluid, respectively. The pressure is given by

$$p = (\gamma - 1)\left(e - \frac{m^2+n^2}{2\rho} - q_0\rho z\right)$$

and the temperature is defined as $T = p/\rho$. As in convention, $u = m/\rho$ and $v = n/\rho$ are the velocities in x and y, respectively. The parameters q_0, T_0, γ and $1/\varepsilon$ are the same as those in one-dimensional detonation waves.

For simplicity, here we consider the detonation waves in a two dimensional channel. Now we describe the local random projection method to the problem (5.1) with piecewise constant initial data

$$(\rho(x,y,0), u(x,y,0), v(x,y,0), p(x,y,0), z(x,y,0)) = \begin{cases} (\rho_l, u_l, 0, p_l, 0), & \text{if } x \le \xi(y), \\ (\rho_r, u_r, 0, p_r, 1), & \text{if } x > \xi(y); \end{cases} \tag{5.2}$$

where $\xi(y)$ is a given function of y. Let $U^n_{j,i} = (\rho^n_{j,i}, m^n_{j,i}, n^n_{j,i}, e^n_{j,i}, (\rho z)^n_{j,i})$ be the approximate solution of $U = (\rho, m, n, e, \rho z)$ at $(x_j, y_i, t_n) = (jh, ih, nk)$. Similar to the one-dimensional case, the fractional step method is used. In the first step , the inviscid compressible Euler equations

$$U_t + F(U)_x + G(U)_y = 0, \tag{5.3}$$

are solved by a high resolution shock capturing method $S_c(k)$. In the second step, the chemical reaction term

$$\rho_t = 0, \qquad m_t = 0, \qquad n_t = 0, \qquad e_t = 0, \qquad (\rho z)_t = \frac{1}{\varepsilon}\psi(U). \tag{5.4}$$

is solved using the local random projection method.

At any time step, for each i, there is an $l_i(n) = i_n$, i_n an integer, such that

$$z^n_{j,i} = \begin{cases} 0, & \text{if } j \le l_i(n), \\ 1, & \text{if } j > l_i(n). \end{cases}$$

Here $l_i(n)$ is the location of the jump for z at the grid line $y = y_i$ in the approximate solution at time $t_n = nk$. After the convection step using a shock capturing method, $U^* = S_c(k)U^n$, assume that, for each i, the front at time t_{n+1} is located at $l_i(n+1)$, which can be easily determined from the value of z. Since at each time step, the front moves by at most one grid point, thus $l_i(n+1)$ is at most one grid away from $l_i(n)$, thus one only need to check the three points centered at $l_i(n)$. Then the following local projection around the front is used:

$$\begin{aligned} \hat{S}_\theta(k): \quad & \rho^{n+1}_{j,i} = \rho^*_{j,i}, \qquad m^{n+1}_{j,i} = m^*_{j,i}, \qquad n^{n+1}_{j,i} = n^*_{j,i}, \qquad e^{n+1}_{j,i} = e^*_{j,i}, \\ & \text{For } i \quad \text{do} \\ & \quad \text{Set } l_i(n+1) := l_i(n) - 1, && (5.5) \\ & \quad \text{For } \ m = l_i(n) - 1, \cdots, \ \text{do}: \ l_i(n+1) = m, \quad \text{if } T^*_{m,i} > \theta_n; \\ & \quad z^{n+1}_{j,i} = \begin{cases} 0, & \text{if } j \le l_i(n+1), \\ 1, & \text{if } j > l_i(n+1). \end{cases} && (5.6) \\ & && (5.7) \end{aligned}$$

Then the corresponding algorithm is

$$U^{n+1} = \hat{S}_\theta(k)\ S_c(k)U^n. \tag{5.8}$$

The stability condition for this algorithm is still the usual CFL condition determined from the convection term.

Now we will test the above algorithm (5.8) with a numerical example.

Example 5.1 A two-dimensional detonation wave

We consider the problem (5.1) in a two dimensional channel, the upper and lower boundaries are solid walls. We choose γ, q_0, ε and T_0 the same as those in the example 4.1. The initial data in (5.2) is chosen as $\rho_r = \rho_0$, $u_r = u_0$, $p_r = p_0$, $\rho_l = \rho_{CJ}$, $u_l = 8.162 \times 10^4 > u_{CJ}$ and $p_l = p_{CJ}$ with ρ_l, u_l, p_l, ρ_{CJ}, u_{CJ} and p_{CJ} given in the example 4.1. This problem is solved on $[0, 0.025] \times [0, 0.005]$ with a 501×101 mesh, and

$$\xi(y) = \begin{cases} 0.004 & |y - 0.0025| \ge 0.001, \\ 0.005 - |y - 0.0025| & |y - 0.0025| < 0.001. \end{cases}$$

Thus the mesh size $h = 5 \times 10^{-5}$. The time step is chosen as $k = 5 \times 10^{-11}$.

Figure 16 shows density contours at several different times. One can see that the triple points, which is the important feature of the solution, travel in the transverse direction and bounce back and forth against the upper and lower walls. On the contrary, the triple points cease to move after some time by using the usual deterministic method [10].

6 Conclusions

We propose the random projection method for numerical simulations of the hyperbolic conservation laws with stiff source terms arising from chemically reactive flows. Our method consists of the typical two steps: solving the homogeneous hyperbolic conservation law by a standard modern shock capturing method, and using a project method for the stiff reaction term. In the reaction step, we replace the deterministic ignition temperature by a uniformly distributed random variable in a suitable domain. This random projection method avoids the classical spurious speed of detonation waves when the reaction time is not numerically resolved. For model scalar conservation laws with stiff source term, we prove that the random projection method captures the correct shock speed with a first order accuracy. Extensive numerical examples for one and two dimensional scalar problems, and one and two dimensional detonation waves are presented. The results show that the random projection method can indeed capture the correct speeds of discontinuities with an underresolved numerical discretization.

In the future we will conduct more numerical experiments to validate the applicability of this new method. To conclude, we quote a remark by Chorin in [3]: *a certain randomness is a property of many real flows, and thus a method which exhibits randomness is not necessarily less desirable than a method which yields fully predictable answers.*

References

[1] M. Ben-Artzi, The generalized Riemann problem for reactive flows, J. Comput. Phys. 81, 70-101 (1989).

[2] A. Bourlioux, A. Majda and C. Roytburd, Theoretical and numerical structure for unstable one-dimensional detonations, SIAM J. Appl. Math. 51, 303-343 (1991).

[3] A. J. Chorin, Random choice solution of hyperbolic systems, J. Comput. Phys. 22, 517-533 (1976).

[4] A. J. Chorin, Random choice methods with applications to reacting gas flow, J. Comput. Phys. 25, 253-272 (1977).

[5] P. Colella, Glimm's method for gas dynamics, SIAM J. Sci. Stat. Comput. 3, 76-110 (1982).

[6] P. Colella, A. Majda and V. Roytburd, Theoretical and numerical structure for reacting shock waves, SIAM J. Sci. Stat. Comput. 7, 1059-1080 (1986).

[7] R. Courant and K. O. Friedrichs, Supersonic Flow and Shock Waves, Interscience, New York, 1971.

[8] T. Elperin and O. Igra, About the choice of uniformly distributed sequences to be used in the random choice method, Comput. Methods Appl. Mech. Engrg. 57, 181-189 (1986).

[9] B. Engquist and B. Sjogreen, Numerical approximation of hyperbolic conservation laws with stiff terms, Proc. Third Int'l Conf. on Hyperbolic Problems, 848-860, Studentlitteratur, Lund, 1991.

[10] B. Engquist and B. Sjogreen, Robust difference approximations of stiff inviscid detonation waves, UCLA CAM Report 91-03

[11] H. Fan, S. Jin and Z. H. Teng, Zero reaction limit for hyperbolic conservation laws with source terms, J. Diff. Eqn., to appear.

[12] J. Glimm, Solutions in the large for nonlinear hyperbolic systems of equations, Comm. Pure Appl. Math. 18, 697-715 (1965).

[13] S. Jin and Z. P. Xin, The relaxation schemes for systems of conservation laws in arbitrary space dimensions, Comm. Pure Appl. Math. 48, 235-276 (1995).

[14] R. J. LeVeque and H. C. Yee, A study of numerical methods for hyperbolic conservation laws with stiff source terms, J. Comput. Phys. 86, 187-210 (1990).

[15] A. Majda, A qualitative model for dynamic combustion, SIAM J. Appl. Math. 41, 70-93 (1981).

[16] R.B. Pember, Numerical methods for hyperbolic conservation laws with stiff relaxation. I. spurious solutions, SIAM J. Appl. Math. 53, 1293-1330, 1993.

[17] H.C. Yee and P. K. Sweby, Nonlinear Dynamics & Numerical Uncertainties in CFD, NASA TM 110398.

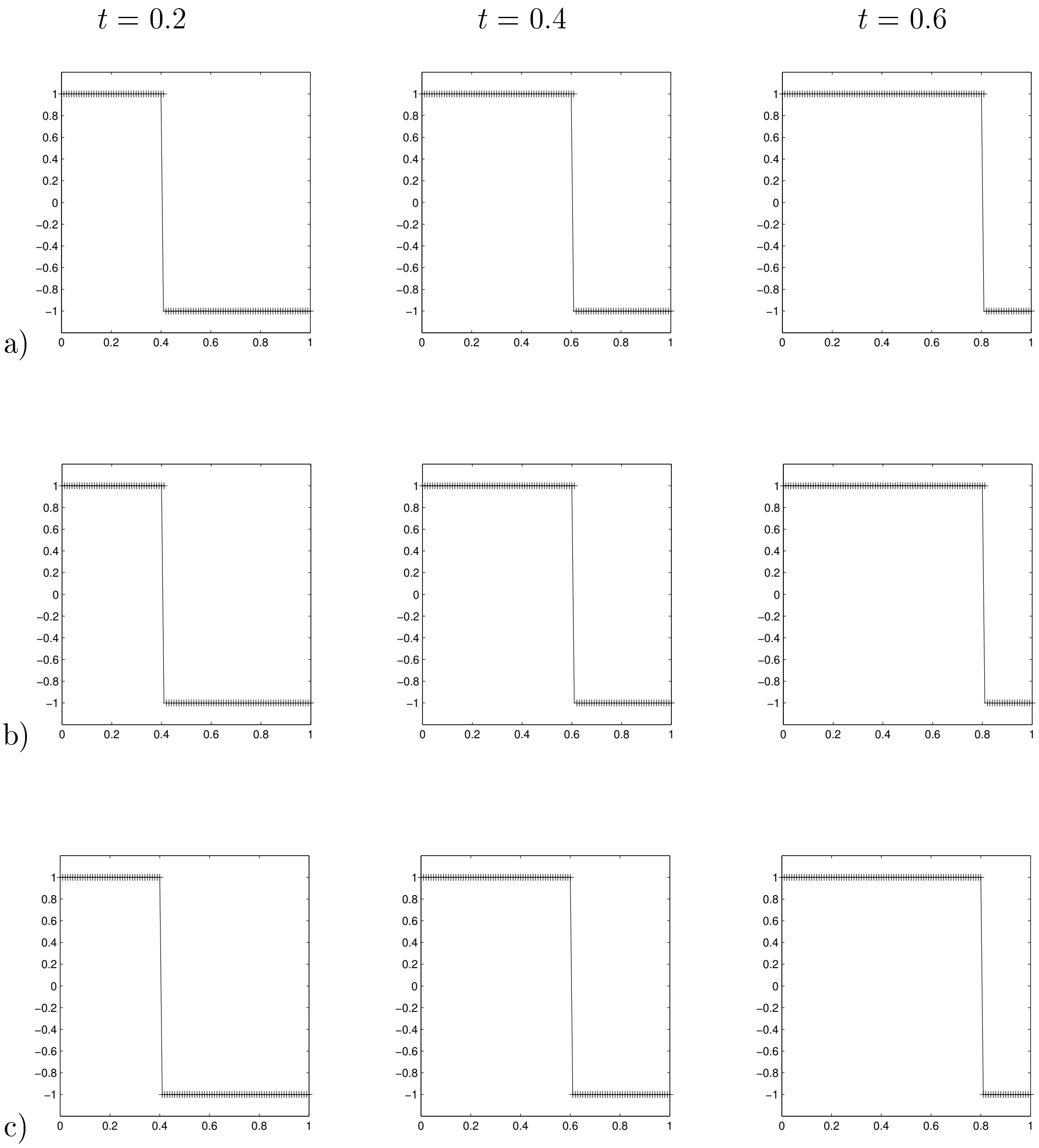

Figure 1. Numerical results by using the random projection method for example 2.1. $\varepsilon = 10^{-4}$, $h = 0.01$. $-$: true solution; $++$: computed solutions. a). $k = 0.004$, b). $k = 0.001$, and c). $k = 0.0001$.

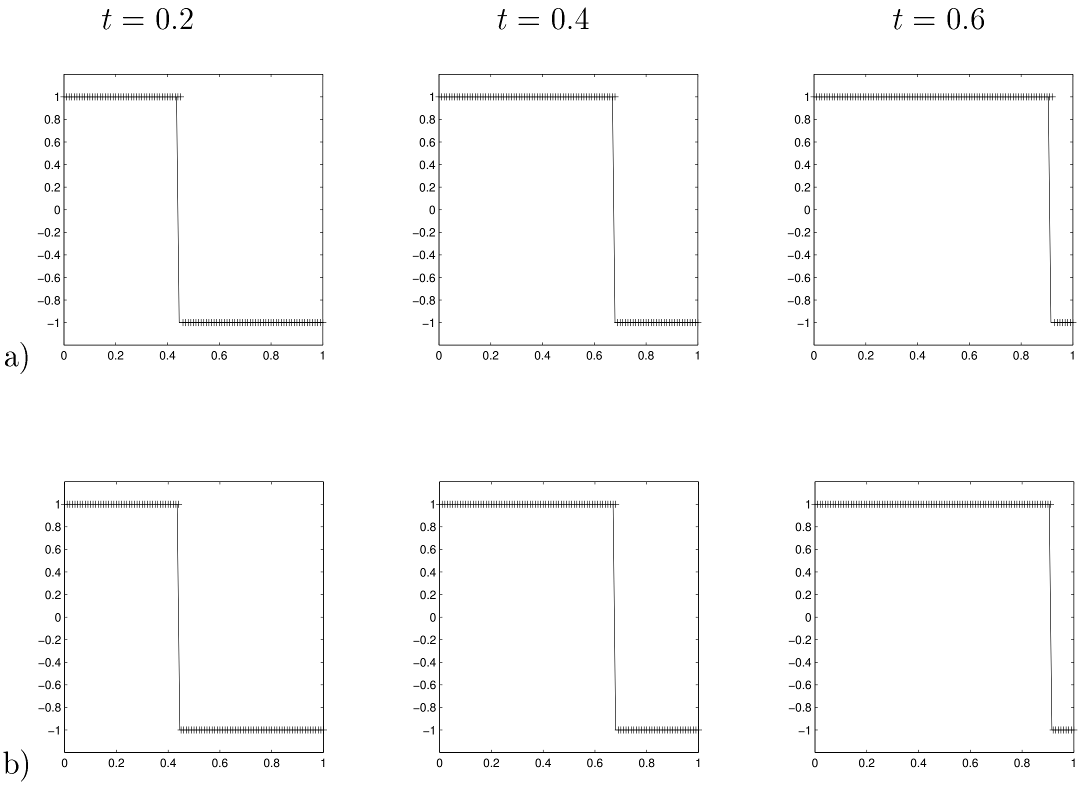

Figure 2. Numerical results by using the random projection method for example 2.2. $\varepsilon = 10^{-4}$, $h = 0.01$. $-$: true solution; $++$: computed solutions. a). $k = 0.004$, b). $k = 0.001$.

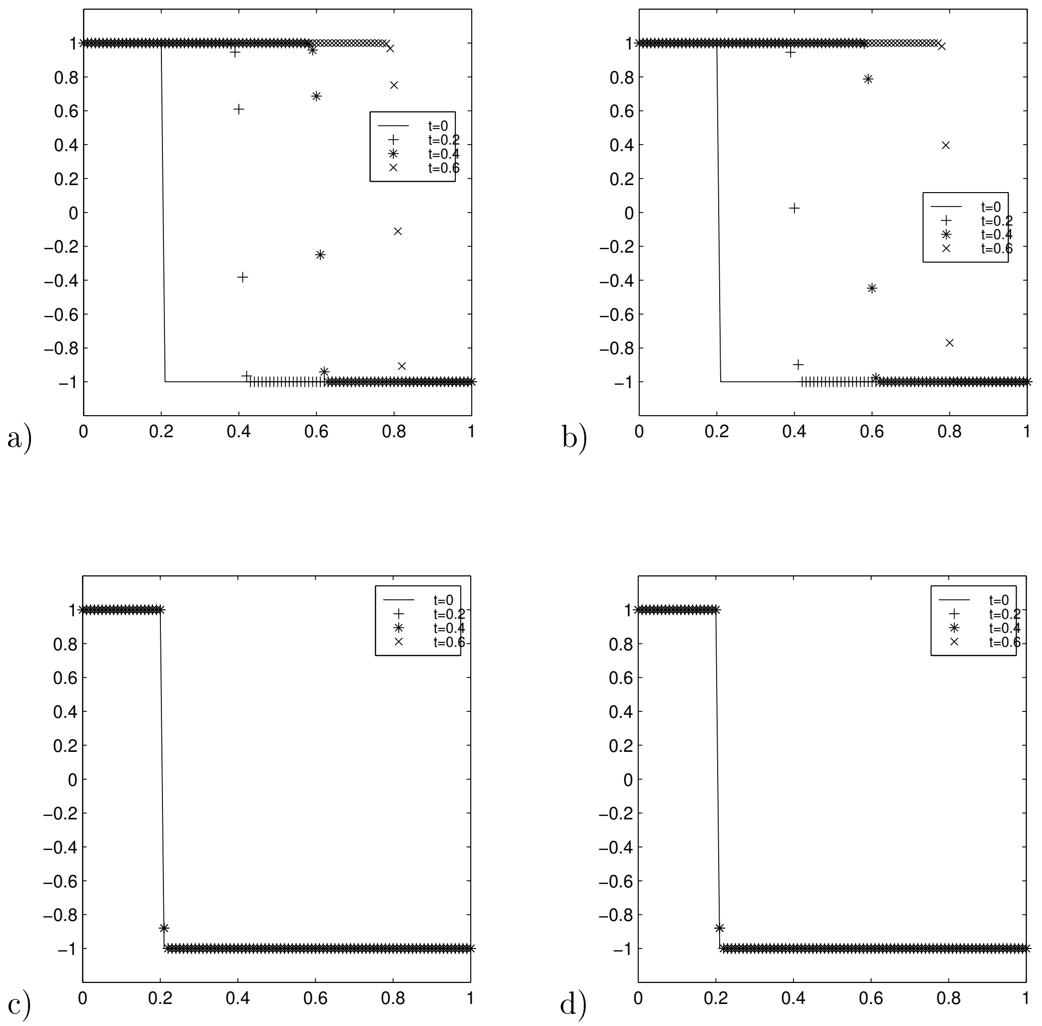

Figure 3. Numerical results by using the explicit method (2.34) for example 2.3. $h = 0.01$. a). $\varepsilon = 0.1$, $k = 0.001$; b). $\varepsilon = 0.01$, $k = 0.0001$; c). $\varepsilon = 0.001$, $k = 0.00001$; d). $\varepsilon = 0.0001$, $k = 0.000001$.

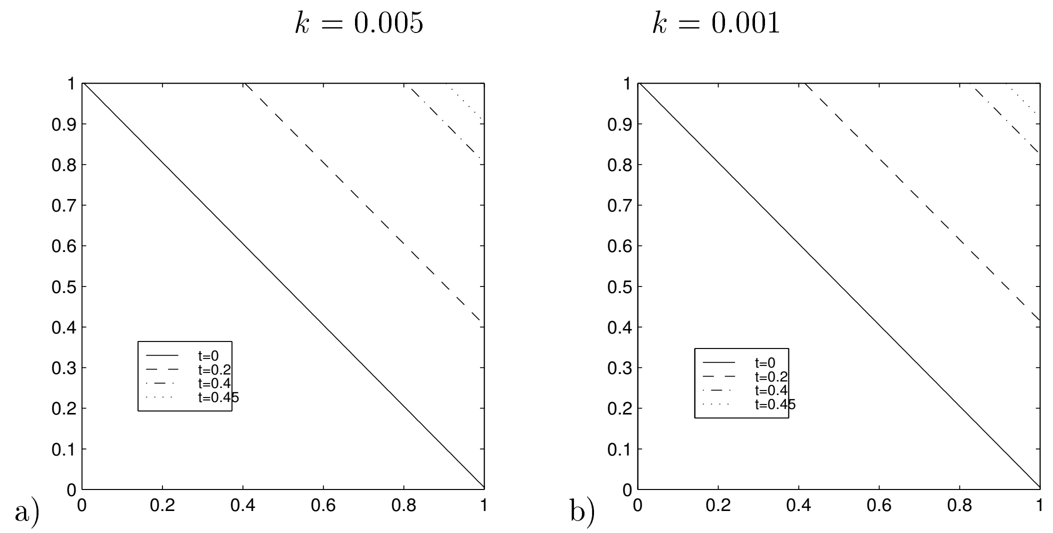

Figure 4. Numerical results for example 3.1 by using the 2D random projection method (3.6). $\varepsilon = 10^{-6}$, $h = 0.01$. a). $k = 0.005$, b). $k = 0.001$.

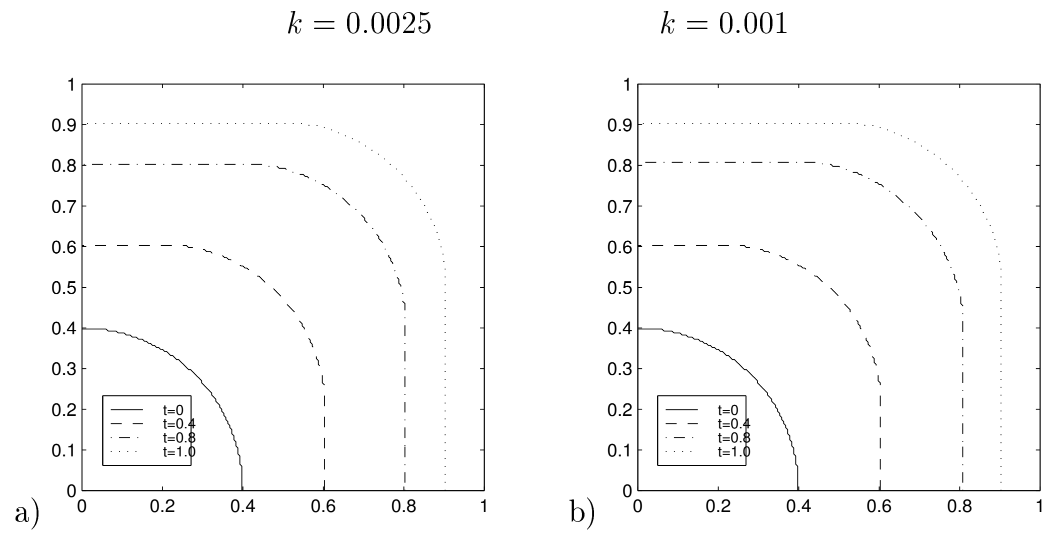

Figure 5. Numerical results for example 3.2 by using the 2D random projection method (3.6). $\varepsilon = 10^{-6}$, $h = 0.005$. a). $k = 0.0025$, b). $k = 0.001$.

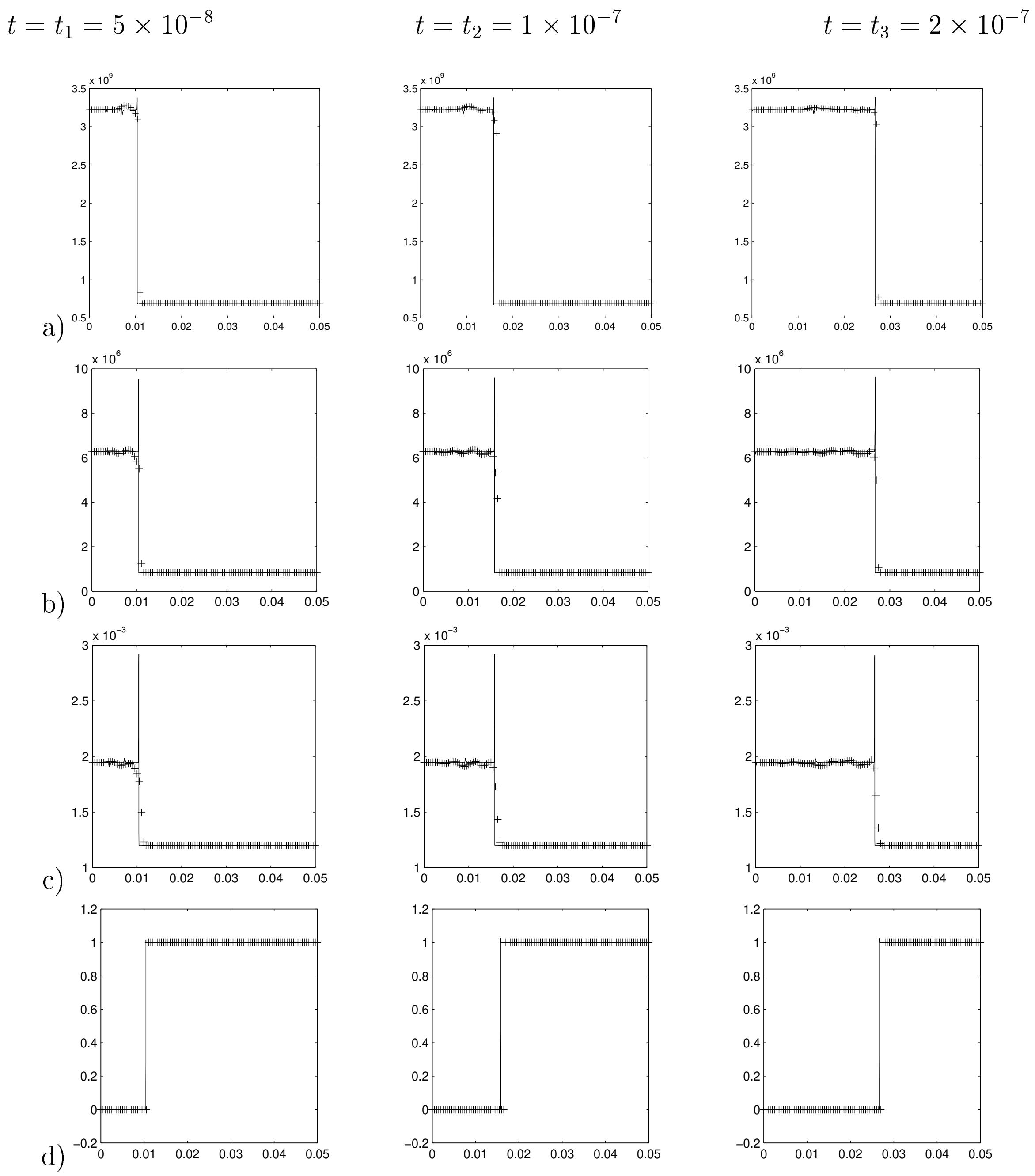

Figure 6. Numerical results for a C-J detonation in example 4.1 by using the global random projection method (4.6). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-10}$. $-$: 'exact' solutions; $++$: computed solutions. a). temperature u, b). pressure p, c). density ρ, and d). z.

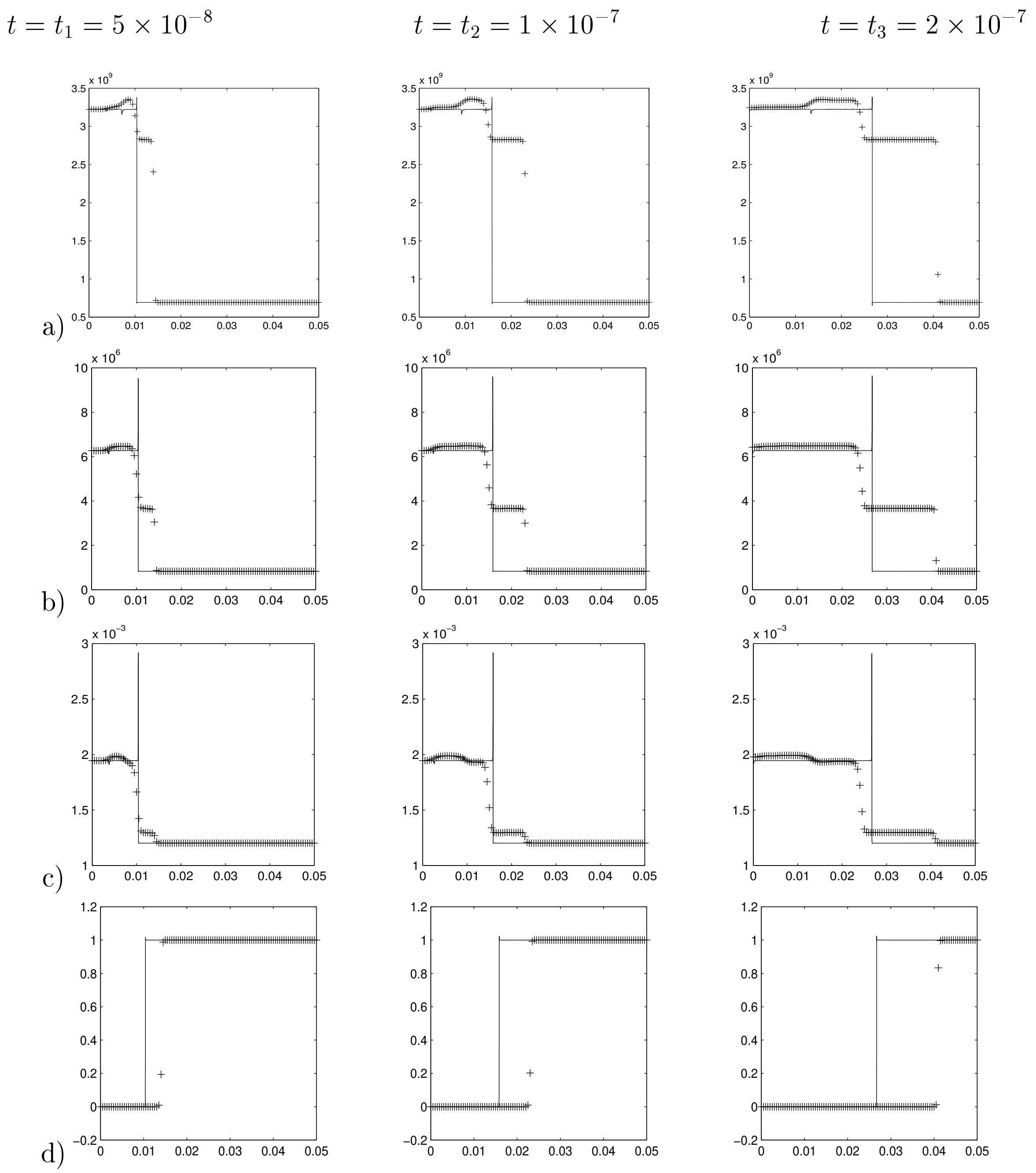

Figure 7. Numerical results for example 4.1 by using the deterministic method (4.11). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-10}$. $-$: 'exact' solutions; $++$: computed solutions. a). temperature u, b). pressure p, c). density ρ, and d). z.

$t = t_1 = 5 \times 10^{-8}$ $\quad$ $t = t_2 = 1 \times 10^{-7}$ $\quad$ $t = t_3 = 2 \times 10^{-7}$

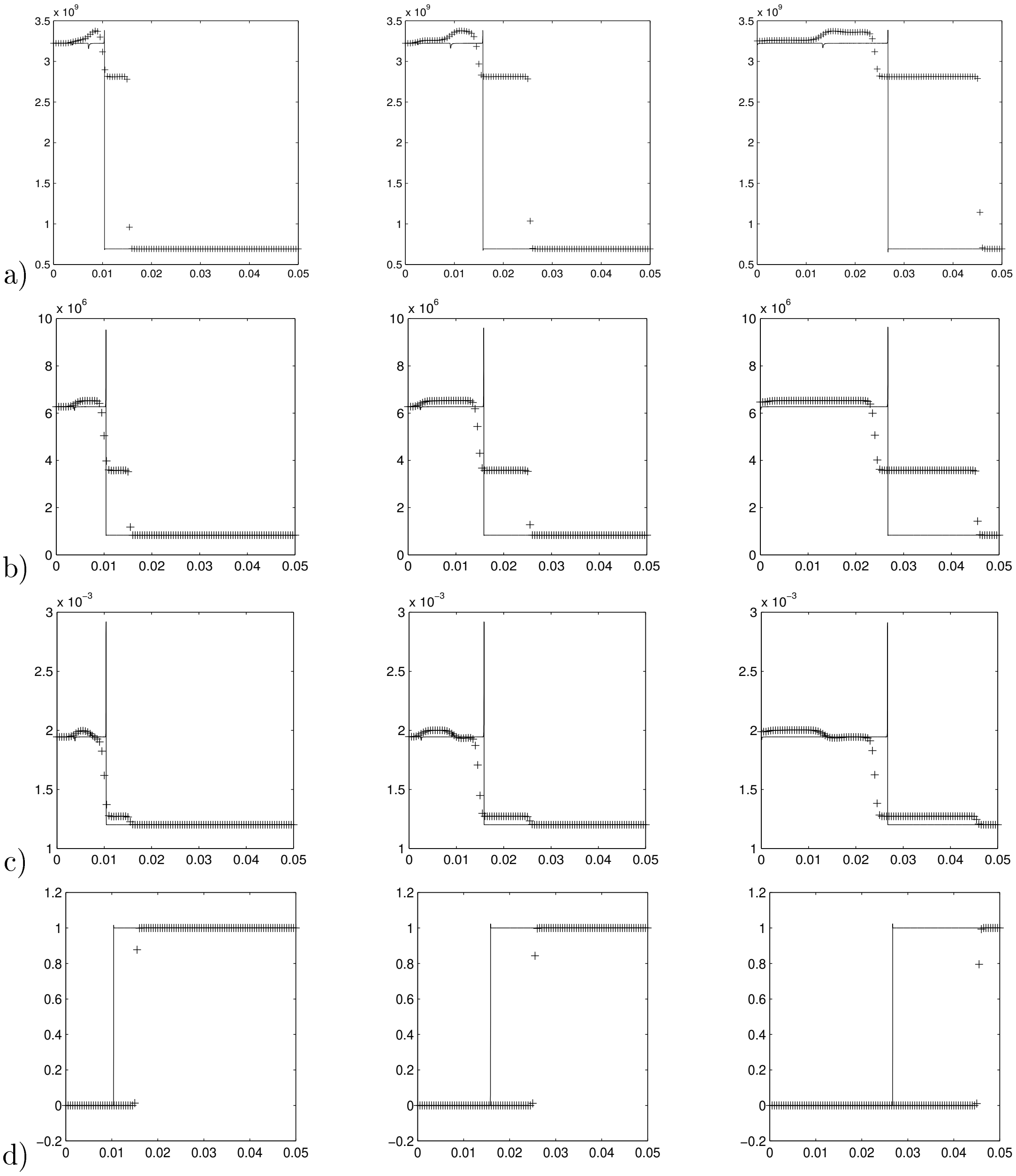

Figure 8. Numerical results for example 4.1 by using the deterministic method (4.11). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-12}$. $-$: 'exact' solutions; $++$: computed solutions. a). temperature u, b). pressure p, c). density ρ, and d). z.

$t = t_1 = 5 \times 10^{-8}$ $\quad$ $t = t_2 = 1 \times 10^{-7}$ $\quad$ $t = t_3 = 2 \times 10^{-7}$

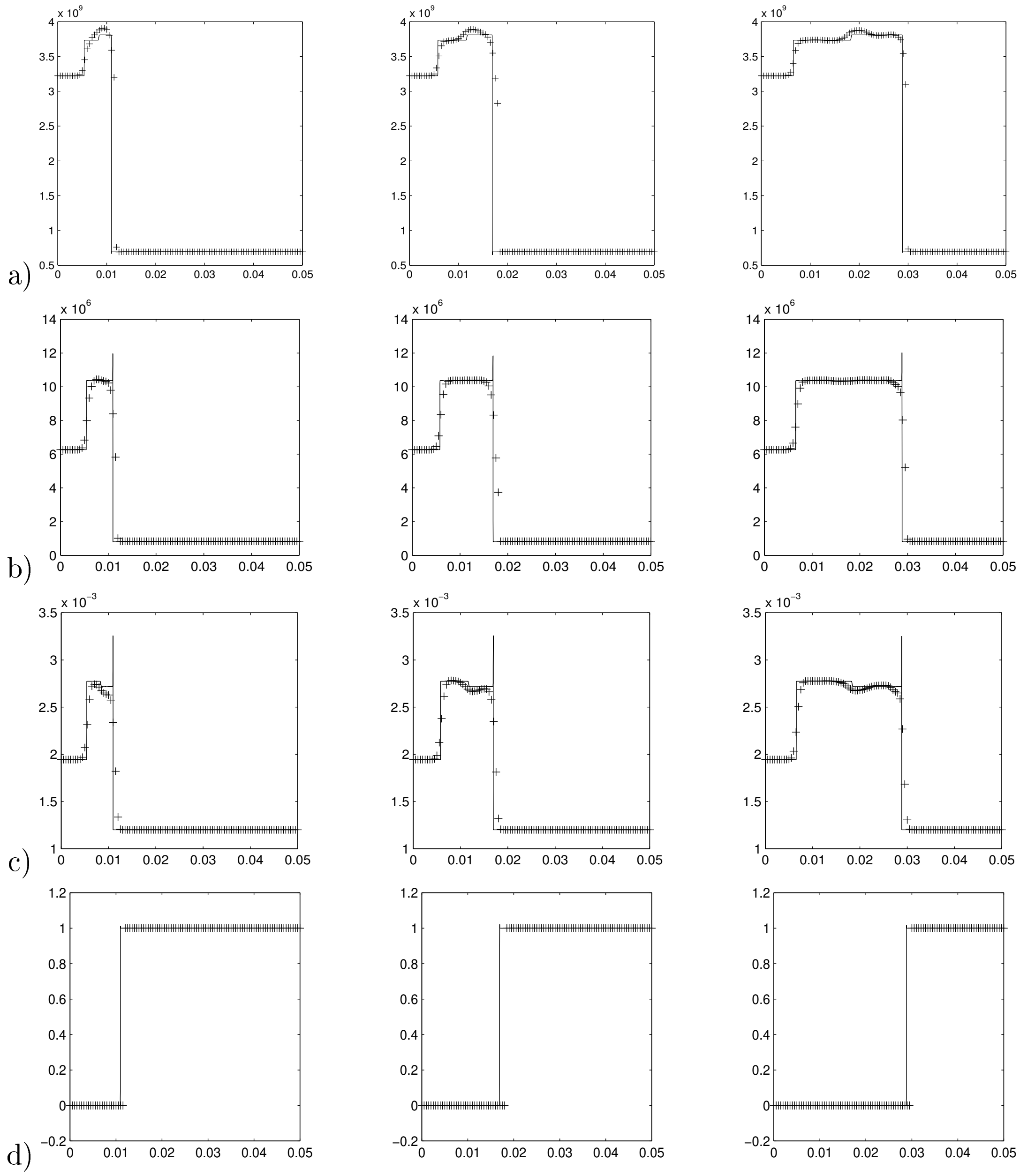

Figure 9. Numerical results for example 4.2 by using the local random projection method (4.9). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-10}$. $\quad -:$ 'exact' solutions; $++:$ computed solutions. a). temperature u, b). pressure p, c). density ρ, and d). z.

$t = t_1 = 5 \times 10^{-8}$ $\quad$ $t = t_2 = 1 \times 10^{-7}$ $\quad$ $t = t_3 = 2 \times 10^{-7}$

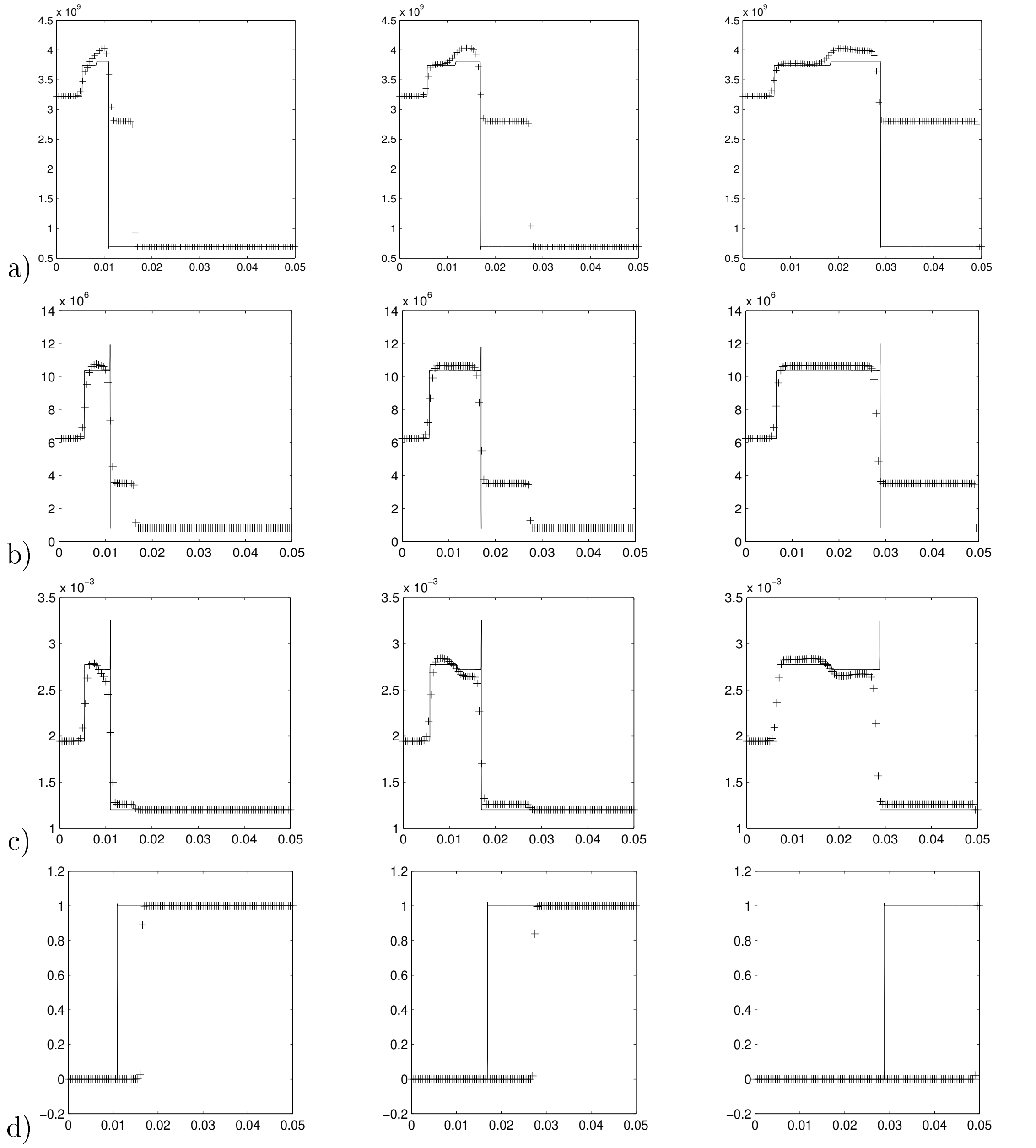

Figure 10. Numerical results for example 4.2 by using the deterministic method (4.11). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-10}$. $\quad$ $-$: 'exact' solutions; $++$: computed solutions. $\quad$ a). temperature u, b). pressure p, c). density ρ, and d). z.

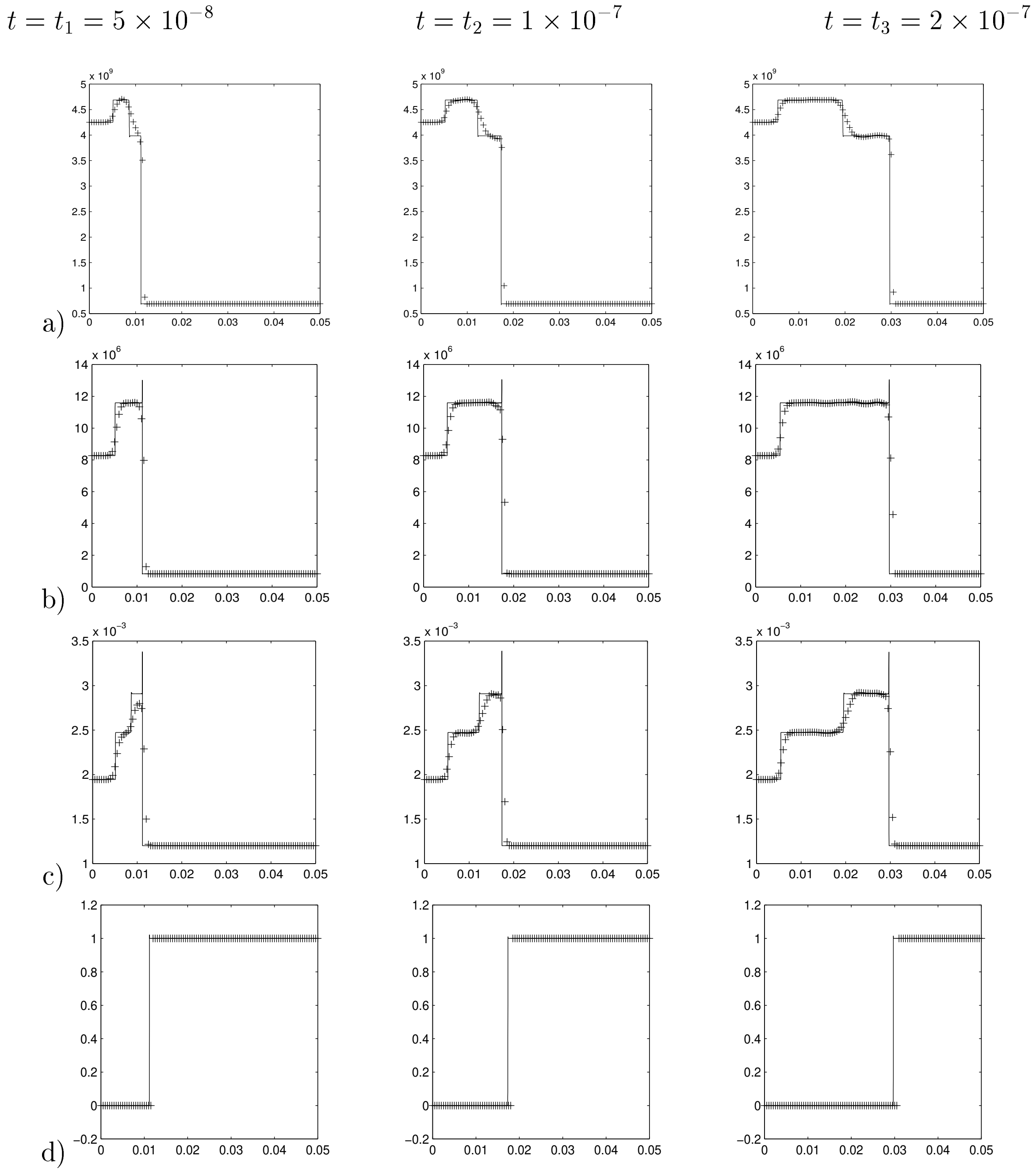

Figure 11. Numerical results for example 4.3 by using the local random projection method (4.9). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-10}$. $-$: 'exact' solutions; $++$: computed solutions. a). temperature u, b). pressure p, c). density ρ, and d). z.

$t = t_1 = 5 \times 10^{-8}$ $\qquad$ $t = t_2 = 1 \times 10^{-7}$ $\qquad$ $t = t_3 = 2 \times 10^{-7}$

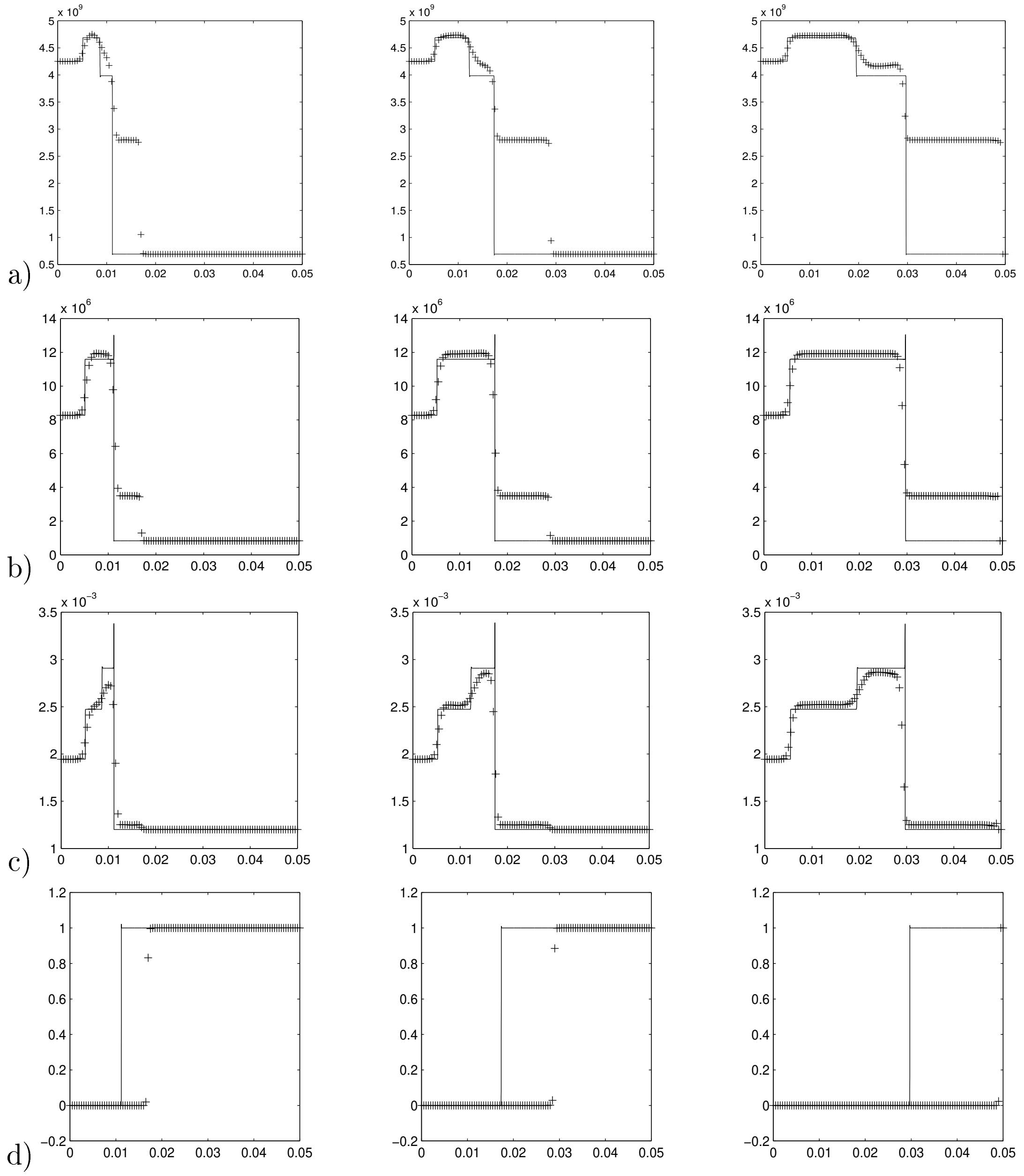

Figure 12. Numerical results for example 4.3 by using the deterministic method (4.11). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-10}$. $-$: 'exact' solutions; $++$: computed solutions. a). temperature u, b). pressure p, c). density ρ, and d). z.

$t = t_1 = 5 \times 10^{-8}$ $t = t_2 = 1 \times 10^{-7}$ $t = t_3 = 2 \times 10^{-7}$

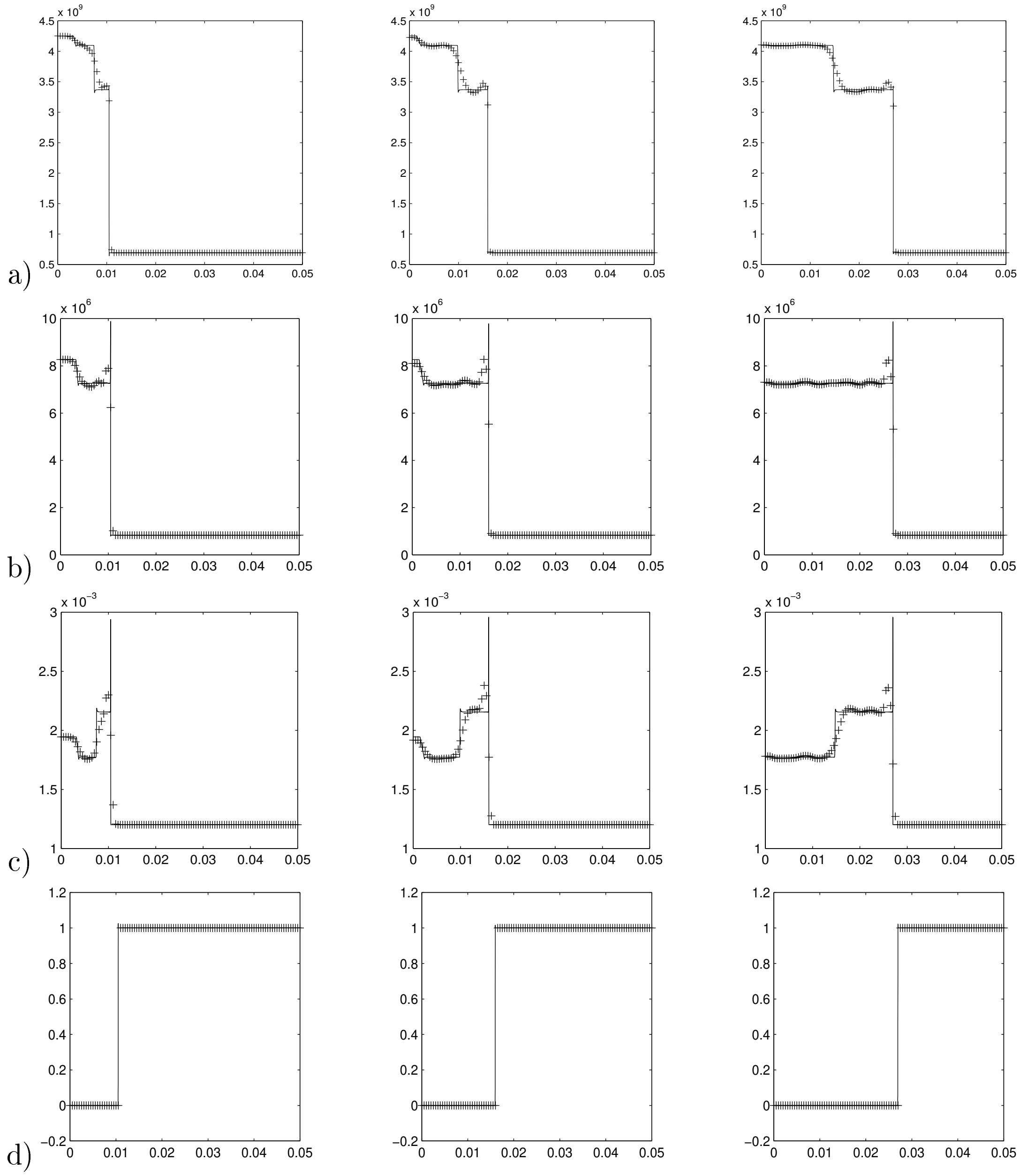

Figure 13. Numerical results for example 4.4 by using the local random projection method (4.9). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-10}$. $-$: 'exact' solutions; $++$: computed solutions. a). temperature u, b). pressure p, c). density ρ, and d). z.

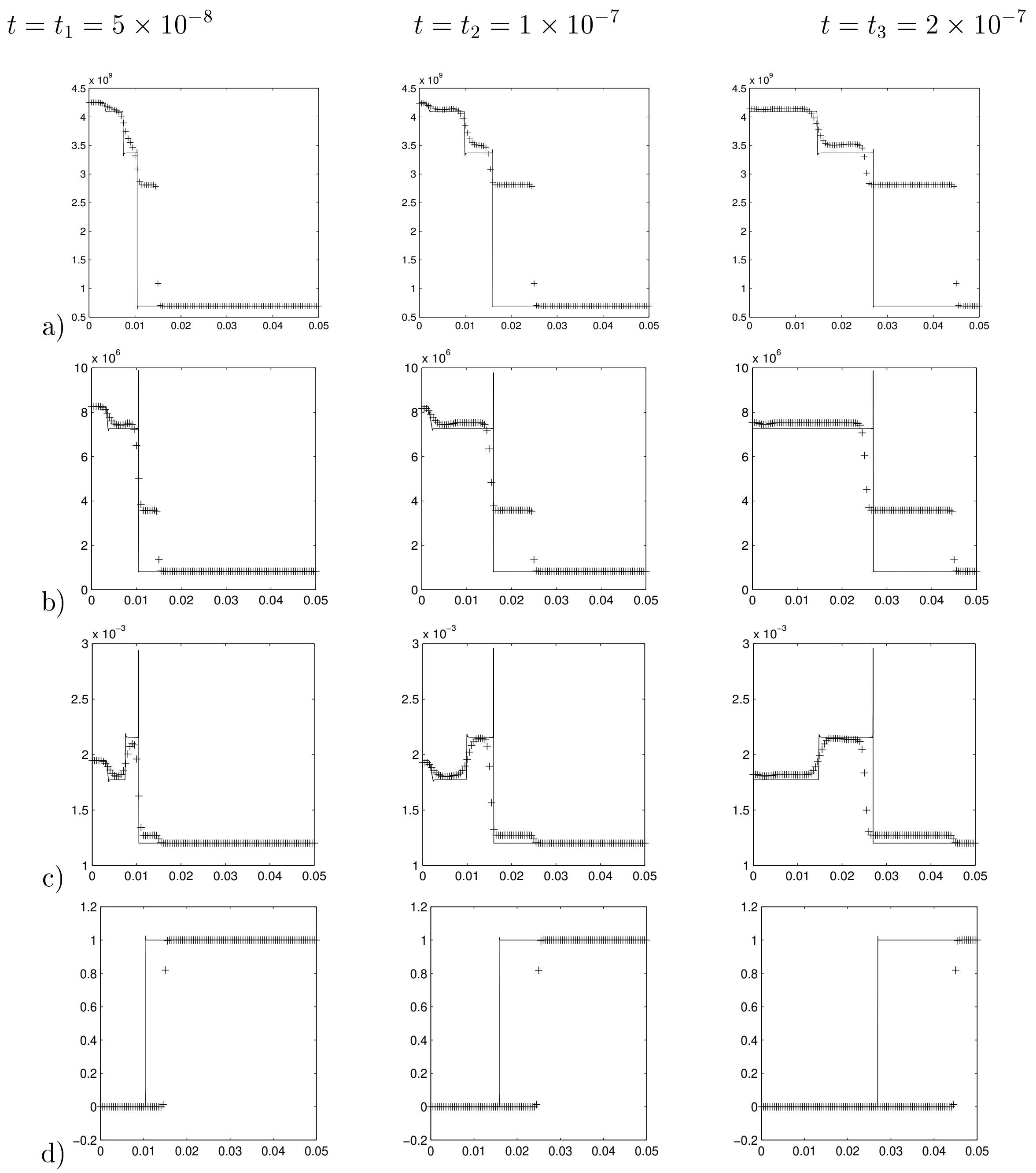

Figure 14. Numerical results for example 4.4 by using the deterministic method (4.11). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-10}$. $-$: 'exact' solutions; $++$: computed solutions. a). temperature u, b). pressure p, c). density ρ, and d). z.

$t = t_1 = 5 \times 10^{-8}$ $\qquad$ $t = t_2 = 1 \times 10^{-7}$ $\qquad$ $t = t_3 = 2 \times 10^{-7}$

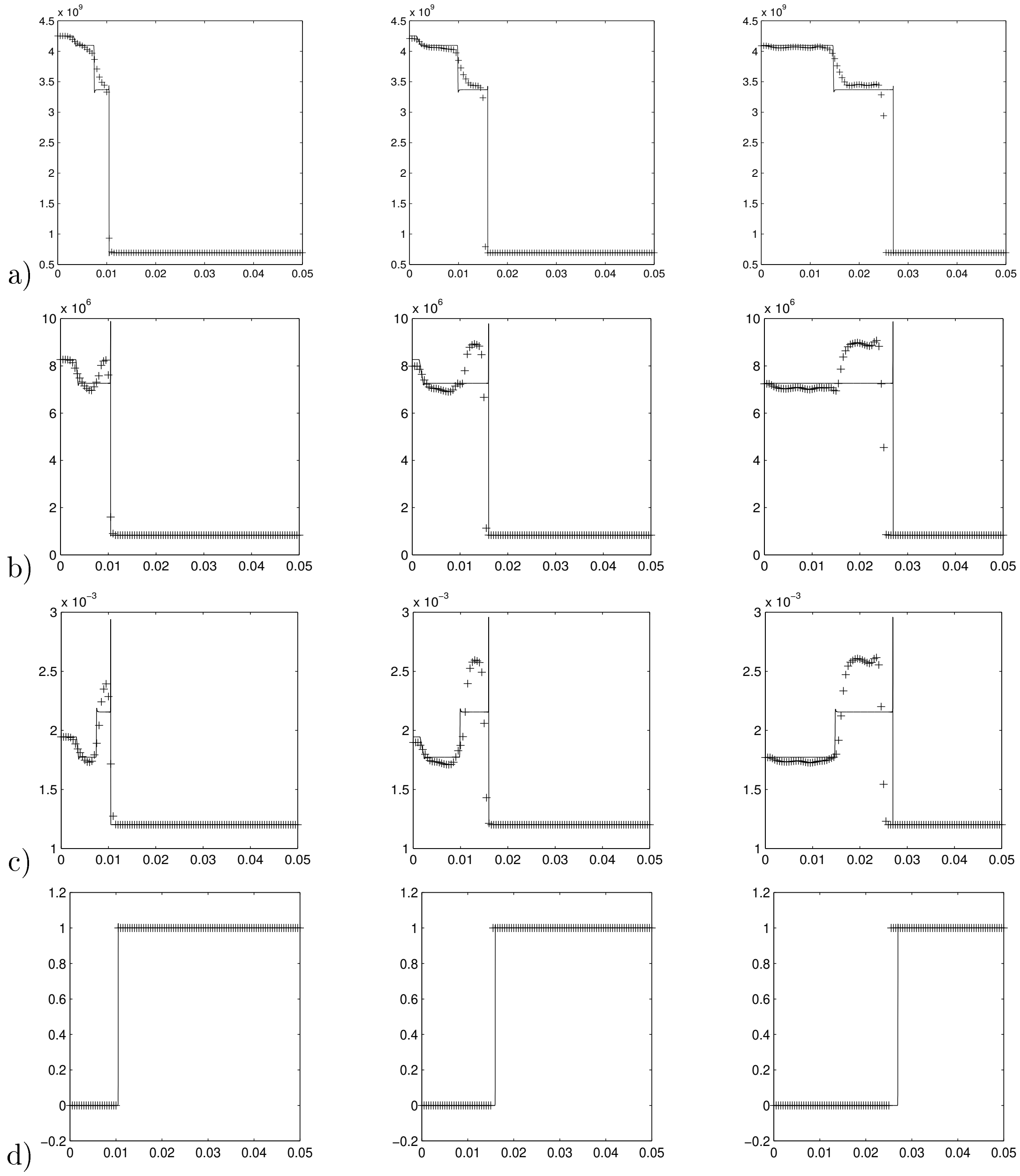

Figure 15. Numerical results for example 4.4 by using the global random projection method (4.6). $h = 5 \times 10^{-4}$, $k = 5 \times 10^{-10}$. $-$: 'exact' solutions; $++$: computed solutions. a). temperature u, b). pressure p, c). density ρ, and d). z.

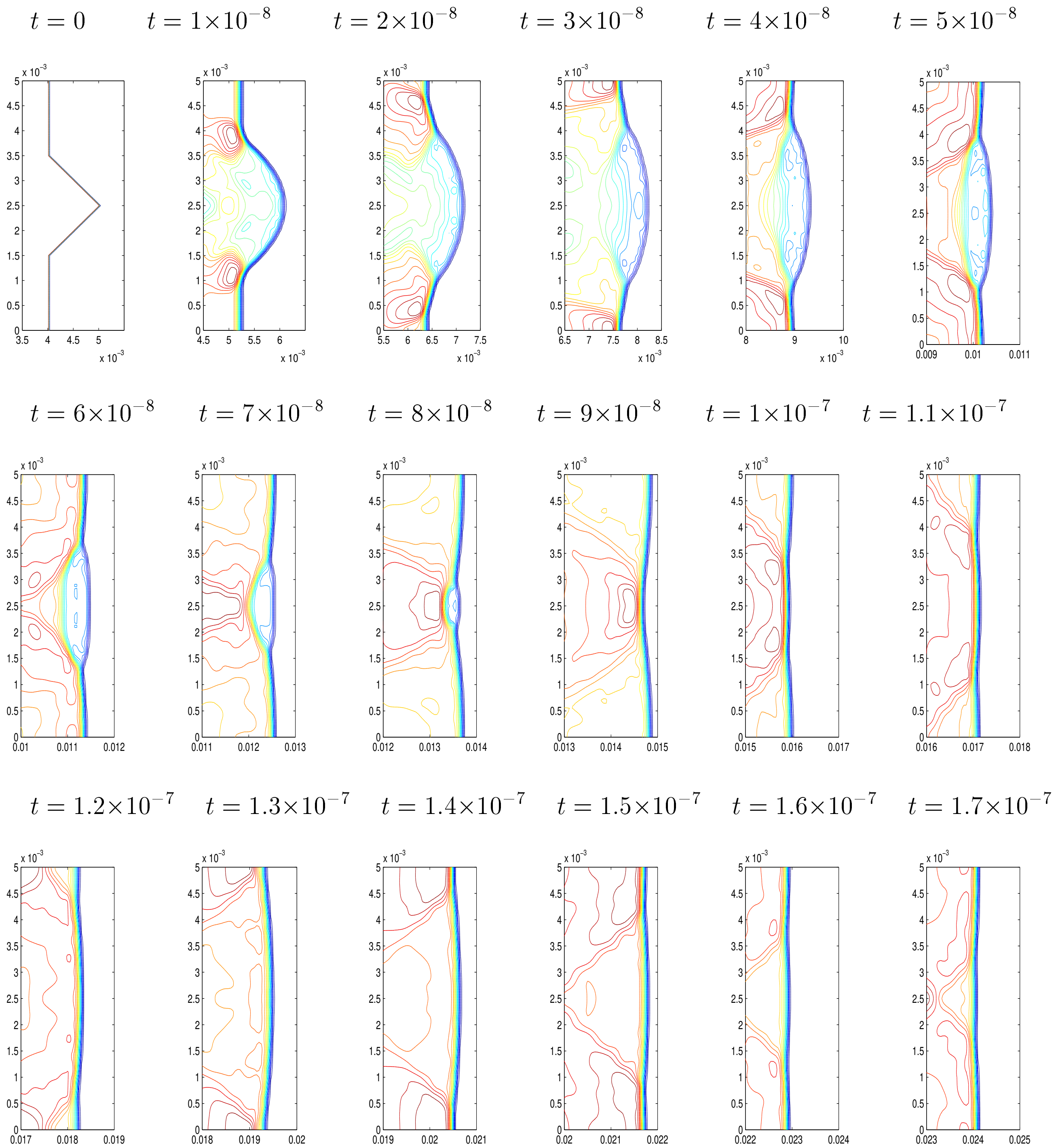

Figure 16. Numerical results of density for example 5.1 by using the 2D local random projection method. $h = 5 \times 10^{-5}$, $k = 5 \times 10^{-11}$.