Error estimates on the random projection methods 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 give error estimates on the random projection methods, recently introduced by the authors, for numerical simulations of the hyperbolic conservation laws with stiff reaction terms:

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

In this problem, the reaction time ε is small, making the problem numerically stiff. A classic spurious numerical phenomenon – the incorrect shock speed – occurs when the reaction time scale is not properly resolved numerically. The random projection method, a fractional step method that solves the homogeneous convection by any shock capturing method, followed by a random projection for the reaction term, was introduced in [1] to handle this numerical difficulty. In this paper, we prove that the random projection methods capture the correct shock speed with a first order accuracy, if a monotonicity-preserving method is used in the convection step. We also extend the random projection method for more general source term $-\frac{1}{\varepsilon}g(u)$, which has finitely many simple zeroes and satisfying $ug(u) > 0$ for large $|u|$.

1 Introduction

In this paper we conduct the error estimates on the random projection methods proposed recently by the authors [1] for numerical simulations of the hyperbolic conservation laws with stiff reaction terms:

$$u_t + f(u)_x = -\frac{1}{\varepsilon}(u-\alpha)(u^2-1), \qquad -1 < \alpha < 1, \tag{1.1}$$

where ε is the reaction time, and f is a convex function of u. This is the simplest model for the reacting flows, where the source term, being the derivative of the typical double well

*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

potential, accounts for chemical reaction, while α plays the role of ignition temperature. In a reacting flow, the reacting time ε is much smaller than the characteristic time scale of the fluid, making the source term in (1.1) stiff.

Numerical methods for the stiff reaction term has attracted lots of attention in the last decade. It was first observed by Colella, Majda and Roytburd [7] that an underresolved numerical method, where ε is not resolved by suitable small time steps and grid sizes, leads to spurious shock that travels one grid per time step. Since then, lots of attention have been paid to study this peculiar numerical phenomenon (see [2], [15], [16], [3] and references therein) and or find a robust way to fix it (see [9, 10]). It is known that numerical shock profile, an essential mechanism in all shock capturing method, leads to too early chemical reaction once the smeared value in the numerical shock layer is above the ignition temperature (α in model (1.1)).

Recently we proposed the random projection method for this problem, and more generally, to inviscid detonation wave calculation [1]. Unlike the classical random choice method for reacting flow [5], originated from Glimm's scheme [12, 4], which requires solving a generalized Riemann problem for hyperbolic systems with source terms, our method is a fraction step method, which combines a standard –no Riemann solver is needed –shock capturing method for the homogeneous convection with a strikingly simple random projection step for the reaction term. In the random projection step, the ignition temperature, or α in (1.1), is chosen to be uniformly distributed random variable between the two stable equilibria ± 1. Although at each time step, this random projection will move the shock with incorrect speed, the statistical average yields the correct speed, even though the small time scale ε is not numerically resolved. In particular, when the random number is chosen to be the equidistributed van der Corput sampling sequence [13], we can prove, for model (1.1), a first order accuracy for both the first order upwind and Lax-Friedrichs methods. We also extended this method straightforwardly to the calculation of detonation waves, where in the projection step for the reaction terms, the ignition temperature is made random. A large amount of numerical experiments for one and two dimensional detonation waves demonstrate the robustness of this novel approach.

In this paper, using (1.1) as a model problem, we first prove that a first order accuracy ($O(h \log h)$) is achieved for any monotonicity-preserving method–thus including all TVD methods of course–applied to the homogeneous convection. We carry out this error estimate for both the global and local random projection methods, which are to be defined in Section 2. Furthermore we extend the local random projection method for more general source term $-\frac{1}{\varepsilon}g(u)$, having finitely many simple zeroes and satisfying $ug(u) > 0$ for large $|u|$ and use various numerical examples to show its ability to obtain the correct shock speed.

The paper is organized as follows: In section 2 we recall the global and the local random projection methods introduced in [1] for (1.1). In section 3 we prove an error estimate for the global random projection method when any monotonicity-preserving method is used for the convection. In section 4 the same analysis is performed for the local random projection method. In section 5 we extend the local random projection method for the more general source term $-\frac{1}{\varepsilon}g(u)$, having finitely many simple zeroes and satisfying $ug(u) > 0$ for large $|u|$. We also present some numerical results for this more general problem.

2 The random projection method

In this section we review the random projection methods introduced in [1] for the following hyperbolic conservation law with stiff source term,

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

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 [15] [11]:

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

Namely,

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

Let h be the spatial increment and k be the time step, such that x_0 is a grid point, i.e. $x_0 = l(0)h$ with $l(0)$ an integer. 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

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

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 (2.6). Then the fraction step method takes the form:

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

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

2.1 The global 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 the deterministic projection operator $S_r(k)$ by the following **global random projection** operator $S_\theta(k)$ 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. The combination of the two steps gives the numerical scheme

$$S_{gp}(k): \qquad u^{n+1} = S_\theta(k) S_c(k) u^n. \tag{2.10}$$

The stability condition for this method, as well as the local random projection method (2.13), are the usual CFL condition determined by the convection term.

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.

While effective most of the time, the global random projection method fails when there are waves that are at most m points away from the detonation fronts, where m is the number of smeared points after the convection $S_c(k)$ [1]. The reason for this is that, during each step, the random projection operator could move the detonation front, forward or backward, by m grid points, forcing it to interact, unphysically, with the neighboring wave. This motivates the introduction of the **local random projection method**, which allows the front to move at most one grid point per time step. Another advantage of the local method is that it can be easily applied to more general reaction terms, as will be demonstrated later.

2.2 The local random projection method

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

$$u_j^n = \begin{cases} 1, & \text{if } j \le l(n), \\ -1, & \text{if } j > l(n). \end{cases} \tag{2.11}$$

Here we assume that x_0, the initial point of discontinuity, is a grid point, i.e. $x_0 = l(0)h$. Let $u^+ = S_c(k)u^n$. The local random projection operator $\tilde{S}_\theta(k)$ is defined as

$$\begin{aligned} \tilde{S}_\theta(k): \quad & \text{Set } l(n+1) := l(n) - 1, \\ & \text{For } \ j = l(n)-1, \cdots, l(n)+1 \quad \text{do}: \ \ l(n+1) = j, \quad \text{if } \ u_j^+ > \theta_n; \\ u_j^{n+1} = & \begin{cases} 1, & \text{if } \ j \le l(n+1), \\ -1, & \text{if } \ j > l(n+1). \end{cases} \end{aligned} \tag{2.12}$$

Again in this method, one random value of θ_n will be selected per time step. The combination of the two steps gives the numerical method:

$$S_{lp}(k): \qquad u^{n+1} = \tilde{S}_\theta(k) S_c(k) u^n. \tag{2.13}$$

The difference between the local random projection method and its global counterpart is that the local one allows, after each time step, the front to move–forward or backward–at most one grid point. When the number of smeared values of u by S_c is at most 2 after the convection step, for example, when one uses the upwind scheme or Lax-Friedrichs scheme for the convection part, the local random projection operator (2.12) is identical to the global random projection operator (2.9). However, when there are more than 2 smeared values of u between -1 and 1 after the convection, these two methods are different. For problems where waves are well-separated, these two methods generate essentially the same numerical results. However, the local method is more adequate for problems involving complex wave interactions.

2.3 The choice of the random sequence θ_n

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 [13, 6]. 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 $[0,1]$:

$$\vartheta_n = \sum_{k=0}^{m} i_k 2^{-(k+1)}, \qquad n = 1, 2, \cdots. \tag{2.14}$$

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

$$\theta_n = 2\vartheta_n - 1, \qquad n = 1, 2, \cdots. \tag{2.15}$$

In order to get error estimates on the global and local random projection methods, we recall some properties of the van der Corput's sampling sequence $\{\vartheta_n, \ n = 1, 2, \cdots\}$. One can see detail in [13], [6] and [1]. Let

$$N\{j = n_1 + 1, \cdots, n_2; \ \vartheta_j \in I\} \tag{2.16}$$

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

$$\delta(\vec{\vartheta}, n_1, n_2, I) \equiv \frac{1}{n_2 - n_1} N\{j = n_1 + 1, \cdots, n_2;\ \vartheta_j \in I\} - |I| \tag{2.17}$$

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 [13] and [6] is rather useful in our error estimate in the next two sections.

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

$$\delta(\vec{\vartheta}, n_1, n_2, I) \le \frac{3\ln(2(n_2 - n_1)) + 1}{n_2 - n_1}, \qquad \forall \quad n_1,\ n_2,\ I. \tag{2.18}$$

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 $[-1,1]$.

3 An error estimate on the global random projection method

Now we will prove that the global random projection method, when a monotonicity-preserving method is used 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 that of Glimm's scheme [6], and the error estimate made in [1] for first order schemes in the convection.

Theorem 3.1 *The difference between the shock location of the exact solution, $x_0 + st_n$, and the numerical one, $l(n)h$, as determined by the global random projection method (2.10), in which any conservative, monotonicity-preserving scheme is used for the convection part, has the following estimate:*

$$|x_0 + st_n - l(n)h| \le h\left[C_1 + C_2 \ln(2st_n/\lambda h)\right], \tag{3.1}$$

where $t_n = nk$ is a fixed time, $\lambda = sk/h$ and C_1 and C_2 are two constants independent of t_n, n, h and k.

Proof: Let

$$u^+ = S_c(k)u^n, \qquad\qquad u^* = S_c(k)u^0, \tag{3.2}$$

where $S_c(k)$ is any conservative, monotonicity-preserving scheme for the convection part:

$$u_j^+ = u_j^n - \frac{k}{h}\left[F\left(u_{j-p}^n, \cdots, u_{j+1}^n, \cdots, u_{j+q}^n\right) - F\left(u_{j-p-1}^n, \cdots, u_j^n, \cdots, u_{j+q-1}^n\right)\right], \tag{3.3}$$

F is the numerical flux function that satisfies $F(u, \cdots, u, \cdots, u) = f(u)$, $p \ge 0$ and $q \ge 0$ are integers.

Due to the projection, at any time step n, there is an $l(n) = j_0$, j_0 an integer, such that

$$u_j^n = \begin{cases} 1, & \text{if } j \le l(n), \\ -1, & \text{if } j > l(n). \end{cases} \tag{3.4}$$

After the convection step (3.3), (3.4), and using (3.2), one has

$$\begin{aligned} u_j^+ &= \begin{cases} 1, & \text{if } j \le l(n) - q, \\ 1 - \frac{k}{h}\left[F(1,\cdots,1,\cdots,-1) - F(1,\cdots,1,\cdots,1)\right], & \text{if } j = l(n) - q + 1, \\ \vdots, & \vdots, \\ 1 - \frac{k}{h}\left[F(1,\cdots,-1,\cdots,-1) - F(1,\cdots,1,\cdots,-1)\right], & \text{if } j = l(n), \\ -1 - \frac{k}{h}\left[F(1,\cdots,-1,\cdots,-1) - F(1,\cdots,-1,\cdots,-1)\right], & \text{if } j = l(n) + 1, \\ \vdots, & \vdots, \\ -1 - \frac{k}{h}\left[F(-1,\cdots,-1,\cdots,-1) - F(1,\cdots,-1,\cdots,-1)\right], & \text{if } j = l(n) + p + 1, \\ -1, & \text{if } j > l(n) + p + 1; \end{cases} \\ &= \begin{cases} 1, & \text{if } j \le l(n) - q, \\ u^*_{l(0)-q+1}, & \text{if } j = l(n) - q + 1, \\ \vdots, & \vdots, \\ u^*_{l(0)}, & \text{if } j = l(n), \\ u^*_{l(0)+1}, & \text{if } j = l(n) + 1, \\ \vdots, & \vdots, \\ u^*_{l(0)+p+1}, & \text{if } j = l(n) + p + 1, \\ -1, & \text{if } j > l(n) + p + 1. \end{cases} \end{aligned} \tag{3.5}$$

Since the scheme (3.3) is monotonicity-preserving, one has

$$\cdots = u^+_{l(n)-q} = 1 \ge u^+_{l(n)-q+1} \ge \cdots \ge u^+_{l(n)} \ge \cdots \ge u^+_{l(n)+p+1} \ge -1 = u^+_{l(n)+p+2} = \cdots \tag{3.6}$$

In the second step, the global random projection (2.9) gives

$$u_j^{n+1} = \begin{cases} 1, & \text{if } j \le l(n) - q, \\ -1, & \text{if } j > l(n) + p + 1; \end{cases} \quad u^{n+1}_{l(n)+r} = \begin{cases} 1, & \text{if } u^+_{l(n)+r} > \theta_n, \\ -1, & \text{if } u^+_{l(n)+r} \le \theta_n; \end{cases} \quad -q < r \le p + 1. \tag{3.7}$$

Combining (3.7), (3.4), noting (3.5), one obtains

$$l(n+1) = \begin{cases} l(n) - q, & \text{if } u^+_{l(n)-q+1} < \theta_n \le u^+_{l(n)-q} = 1, \\ \vdots, & \vdots, \\ l(n), & \text{if } u^+_{l(n)+1} < \theta_n \le u^+_{l(n)}, \\ l(n) + 1, & \text{if } u^+_{l(n)+2} < \theta_n \le u^+_{l(n)+1}, \\ \vdots, & \vdots, \\ l(n) + p + 1, & \text{if } -1 = u^+_{l(n)+p+2} \le \theta_n \le u^+_{l(n)+p+1}; \end{cases}$$

$$
= \begin{cases} l(n)-q, & \text{if } u^*_{l(0)-q+1} < \theta_n \le u^*_{l(0)-q} = 1, \\ \vdots, & \qquad \vdots, \\ l(n), & \text{if } u^*_{l(0)+1} < \theta_n \le u^*_{l(0)}, \\ l(n)+1, & \text{if } u^*_{l(0)+2} < \theta_n \le u^*_{l(0)+1}, \\ \vdots, & \qquad \vdots, \\ l(n)+p+1, & \text{if } -1 = u^*_{l(0)+p+2} \le \theta_n \le u^*_{l(0)+p+1}. \end{cases} \tag{3.8}
$$

Therefore the discontinuities in the approximate solution at each time step moves r ($-q \le r \le p+1$) grid points away from that of the previous time step, where a negative r means moving to the left. We now examine the accumulative effect over many time steps. In fact, from (3.8), noting (2.15) and (2.16), one gets

$$
\begin{aligned}
l(n) &= l(0) + (p+1)N\{j=1,\cdots,n;\ \theta_j \in [-1, u^*_{l(0)+p+1}]\} \\
&\quad + \cdots + N\{j=1,\cdots,n;\ \theta_j \in (u^*_{l(0)+2}, u^*_{l(0)+1}]\} \\
&\quad - N\{j=1,\cdots,n;\ \theta_j \in (u^*_{l(0)}, u^*_{l(0)-1}]\} \\
&\quad - \cdots - qN\{j=1,\cdots,n;\ \theta_j \in (u^*_{l(0)-q+1}, 1]\} \\
&= l(0) + (p+1)N\{j=1,\cdots,n;\ \vartheta_j \in [0, (1+u^*_{l(0)+p+1})/2]\} \\
&\quad + \cdots + N\{j=1,\cdots,n;\ \vartheta_j \in ((1+u^*_{l(0)+2})/2, (1+u^*_{l(0)+1})/2]\} \\
&\quad - N\{j=1,\cdots,n;\ \theta_j \in ((1+u^*_{l(0)})/2, (1+u^*_{l(0)-1})/2]\} \\
&\quad - \cdots - qN\{j=1,\cdots,n;\ \theta_j \in ((1+u^*_{l(0)-q+1})/2, 1]\}.
\end{aligned} \tag{3.9}
$$

Using definitions (2.17) and (2.18) with $n_1 = 0$, $n_2 = n$ and $I \subset [0,1]$, and combining (3.9), (2.11), (2.4), (2.3), (3.5), we obtain

$$
\begin{aligned}
&|x_0 + st_n - l(n)h| \\
&= \Big| st_n - h(p+1)N\{j=1,\cdots,n;\ \vartheta_j \in [0,(1+u^*_{l(0)+p+1})/2]\} \\
&\quad - \cdots - hN\{j=1,\cdots,n;\ \vartheta_j \in ((1+u^*_{l(0)+2})/2, (1+u^*_{l(0)+1})/2]\} \\
&\quad + hN\{j=1,\cdots,n;\ \theta_j \in ((1+u^*_{l(0)})/2, (1+u^*_{l(0)-1})/2]\} \\
&\quad + \cdots + hqN\{j=1,\cdots,n;\ \theta_j \in ((1+u^*_{l(0)-q+1})/2, 1]\} \Big| \\
&= \Big| st_n - nh \Big[(p+1)(1+u^*_{l(0)+p+1})/2 + (p+1)\delta\left(\vec{\vartheta}, 0, n, [0, (1+u^*_{l(0)+p+1})/2]\right) \\
&\quad + \cdots + (u^*_{l(0)+1} - u^*_{l(0)+2})/2 + \delta\left(\vec{\vartheta}, 0, n, ((1+u^*_{l(0)})/2, (1+u^*_{l(0)-1})/2]\right) \\
&\quad - (u^*_{l(0)-1} - u^*_{l(0)})/2 - \delta\left(\vec{\vartheta}, 0, n, ((1+u^*_{l(0)})/2, (1+u^*_{l(0)-1})/2]\right) \\
&\quad - \cdots - q(1-u^*_{l(0)-q+1})/2 - q\delta\left(\vec{\vartheta}, 0, n, ((1+u^*_{l(0)-q+1})/2, 1]\right) \Big] \Big| \\
&= \Big| st_n - \frac{nh}{2} \Big[(p+1) - q + u^*_{l(0)+p+1} + \cdots + u^*_{l(0)+1} + u^*_{l(0)} + u^*_{l(0)-1} + \cdots + u^*_{l(0)-q+1} \Big] \\
&\quad - nh \Big[(p+1)\delta\left(\vec{\vartheta}, 0, n, [0, (1+u^*_{l(0)+p+1})/2]\right) + \cdots \\
&\quad + \delta\left(\vec{\vartheta}, 0, n, ((1+u^*_{l(0)})/2, (1+u^*_{l(0)-1})/2]\right) - \delta\left(\vec{\vartheta}, 0, n, ((1+u^*_{l(0)})/2, (1+u^*_{l(0)-1})/2]\right) \\
&\quad - \cdots - q\delta\left(\vec{\vartheta}, 0, n, ((1+u^*_{l(0)-q+1})/2, 1]\right) \Big] \Big|
\end{aligned}
$$

$$
\begin{aligned}
&\leq \left| st_n - \frac{nk}{2}\left(F(1,\cdots,1,\cdots,1) - F(-1,\cdots,-1,\cdots,-1)\right)\right| + h(C_1 + C_2\ln(2t_n/k)) \\
&= \left| st_n - \frac{nk}{2}(f(1) - f(-1))\right| + h(C_1 + C_2\ln(2t_n/k)) \\
&= |st_n - snk| + h(C_1 + C_2\ln(2st_n/\lambda h)) \\
&= h[C_1 + C_2\ln(2st_n/\lambda h)], \qquad (3.10)
\end{aligned}
$$

where C_1 and C_2 depend on p and q only. Thus the proof of (3.1) is completed.

From this theorem, we know that the global random projection method yields the correct shock speed with an error of $O(h\ln h)$ when a monotonicity-preserving method is used. Since the solution is a step function, this error estimate holds for the l^1-norm between the numerical solution of (1.1) and the exact solution.

4 Error estimates on the local random projection method

Now we will prove that the local random projection method, when using a monotonicity-preserving method for the convection term, captures the correct location of discontinuities with a first order accuracy for the scalar model problem (2.1).

Theorem 4.1 *Suppose $S_c(k)$ is a monotonicity-preserving scheme for the homogeneous hyperbolic conservation law (2.5). Let*

$$
u^* = S_c(k)u^0, \qquad s^* = \frac{h(u^*_{l(0)} + u^*_{l(0)+1})}{2k}. \tag{4.1}
$$

Then the difference between the shock location of the exact solution, $x_0 + st_n$, and the numerical one, $l(n)h$, as determined by the local random projection method (2.13) has the following estimate:

$$
|x_0 + st_n - l(n)h| \leq |s - s^*|t_n + h\left[C_1 + C_2\ln(2st_n/\lambda h)\right], \tag{4.2}
$$

where $t_n = nk$ is a fixed time, C_1 and C_2 are constants independent of t_n, h, n and k.

Proof: First, due to the projection, at any time step n, there is an $l(n) = j_0$, j_0 an integer, such that

$$
u_j^n = \begin{cases} 1, & \text{if } j \leq l(n), \\ -1, & \text{if } j > l(n). \end{cases} \tag{4.3}
$$

Let $u^* = S_c(k)u^0$. Since $S_c(k)$ monotonicity-preserving, after the convection step (3.2), one has

$$
u^+ = S_c(k)u^n, \qquad 1 \geq u^+_{l(n)} = u^*_{l(0)} \geq u^+_{l(n)+1} = u^*_{l(0)+1} \geq -1. \tag{4.4}
$$

In the second step, the local random projection (2.12) gives

$$
u_j^{n+1} = \begin{cases} 1, & \text{if } j \leq l(n) - 1, \\ -1, & \text{if } j > l(n) + 1; \end{cases} \tag{4.5}
$$

$$
u_{l(n)}^{n+1} = \begin{cases} 1, & \text{if } u^+_{l(n)} > \theta_n \\ -1, & \text{if } u^+_{l(n)} \leq \theta_n \end{cases} = \begin{cases} 1, & \text{if } u^*_{l(0)} > \theta_n, \\ -1, & \text{if } u^*_{l(0)} \leq \theta_n; \end{cases} \tag{4.6}
$$

$$
u_{l(n)+1}^{n+1} = \begin{cases} 1, & \text{if } u^+_{l(n)+1} > \theta_n \\ -1, & \text{if } u^+_{l(n)+1} \leq \theta_n \end{cases} = \begin{cases} 1, & \text{if } u^*_{l(0)+1} > \theta_n, \\ -1, & \text{if } u^*_{l(0)+1} \leq \theta_n. \end{cases} \tag{4.7}
$$

Thus one has

$$
\begin{aligned}
l(n+1) &= \begin{cases} l(n)-1, & \text{if } u^+_{l(n)} < \theta_n \le 1, \\ l(n), & \text{if } u^+_{l(n)+1} < \theta_n \le u^+_{l(n)}, \\ l(n)+1, & \text{if } -1 \le \theta_n \le u^+_{l(n)+1}; \end{cases} \\
&= \begin{cases} l(n)-1, & \text{if } u^*_{l(0)} < \theta_n \le 1, \\ l(n), & \text{if } u^*_{l(0)+1} < \theta_n \le u^*_{l(0)}, \\ l(n)+1, & \text{if } -1 \le \theta_n \le u^*_{l(0)+1}. \end{cases} \qquad (4.8)
\end{aligned}
$$

Therefore the discontinuities in the approximate solution at each time step moves at most one grid away from that of the previous time step. We now examine the accumulative effect over many time steps. In fact, from (4.8), noting (2.15) and (2.16), one gets

$$
\begin{aligned}
l(n) &= l(0) + N\{j=1,\cdots,n;\ \theta_j \in [-1, u^*_{l(0)+1}]\} - N\{j=1,\cdots,n;\ \theta_j \in (u^*_{l(0)}, 1]\} \\
&= l(0) + N\{j=1,\cdots,n;\ \vartheta_j \in [0, (1+u^*_{l(0)+1})/2]\} \\
&\quad -N\{j=1,\cdots,n;\ \theta_j \in ((1+u^*_{l(0)})/2, 1]\}. \qquad (4.9)
\end{aligned}
$$

Combining (4.9), (4.1), (2.4), (2.3), (2.17) and (2.18) with $n_1 = 0$, $n_2 = n$ and $I \subset [0,1]$, we obtain

$$
\begin{aligned}
&|x_0 + st_n - l(n)h| \\
&= \Big| st_n - hN\{j=1,\cdots,n;\ \vartheta_j \in [0, (1+u^*_{l(0)+1})/2]\} \\
&\quad + hN\{j=1,\cdots,n;\ \theta_j \in ((1+u^*_{l(0)})/2, 1]\} \Big| \\
&= \Big| st_n - nh\Big[(1+u^*_{l(0)+1})/2 - (1-(1+u^*_{l(0)})/2) + \delta\left(\vec{\vartheta}, 0, n, [0, (1+u^*_{l(0)+1})/2]\right) \\
&\quad -\delta\left(\vec{\vartheta}, 0, n, ((1+u^*_{l(0)})/2, 1]\right)\Big]\Big| \\
&\le |st_n - nks^*| + h(C_1 + C_2 \ln(2t_n/k)) \\
&\le t_n|s - s^*| + h\left[C_1 + C_2 \ln(2st_n/\lambda h)\right]. \qquad (4.10)
\end{aligned}
$$

Thus (4.2) is proved.

In this proof, s^* is the averaged numerical shock speed for the local random projection method. If $s^* = s$, then the corresponding method yields the correct shock speed with a first order accuracy. On the other hand, if $s^* \neq s$, a classic spurious numerical phenomenon will happen. Now we will study the averaged numerical shock speeds for some specific monotonicity-preserving schemes.

For the usual first order upwind scheme, it can be shown that

Lemma 4.1 *If $S_c(k)$ is the upwind scheme, namely,*
If $f(1) \ge f(-1)$ then

$$u_j^{n+1} = u_j^n - \frac{k}{h}\left(f(u_j^n) - f(u_{j-1}^n)\right), \qquad (4.11)$$

if $f(1) < f(-1)$ then

$$u_j^{n+1} = u_j^n - \frac{k}{h}\left(f(u_{j+1}^n) - f(u_j^n)\right); \qquad (4.12)$$

then the corresponding averaged numerical shock speed $s^* = s$.

Proof: If $f(1) \geq f(-1)$, from (4.11) and (2.2), a computation shows

$$u^*_{l(0)} = 1, \qquad u^*_{l(0)+1} = -1 - \frac{k}{h}(f(-1) - f(1)); \tag{4.13}$$

Combining (4.13) and (4.1) gives

$$s^* = \frac{h(u^*_{l(0)} + u^*_{l(0)+1})}{2k} = \frac{h}{2k}\left[1 + \left(-1 - \frac{k}{h}(f(-1) - f(1))\right)\right] = \frac{f(1) - f(-1)}{2} = s. \tag{4.14}$$

If $f(1) < f(-1)$, from (4.12) and (2.2), one can show

$$u^*_{l(0)} = 1 - \frac{k}{h}(f(-1) - f(1)), \qquad u^*_{l(0)+1} = -1; \tag{4.15}$$

Combining (4.15) and (4.1) yields

$$s^* = \frac{h(u^*_{l(0)} + u^*_{l(0)+1})}{2k} = \frac{h}{2k}\left[\left(1 - \frac{k}{h}(f(-1) - f(1))\right) - 1\right] = \frac{f(1) - f(-1)}{2} = s. \tag{4.16}$$

Therefore the lemma is proved.

For the Lax-Friedrichs type scheme, one has

Lemma 4.2 *If $S_c(k)$ is the first-order Lax-Friedrichs type scheme*

$$u_j^{n+1} = u_j^n - \frac{k}{2h}\left(f(u^n_{j+1}) - f(u^n_{j-1})\right) + \frac{\sqrt{a}k}{2h}\left(u^n_{j+1} - 2u^n_j + u^n_{j-1}\right), \tag{4.17}$$

where a is a positive constant satisfying

$$-\sqrt{a} \leq f'(u) \leq \sqrt{a} \qquad \text{for all} \quad u, \tag{4.18}$$

then the corresponding averaged numerical shock speed $s^* = s$.

Proof: From (4.17) and (2.2), a computation shows

$$u^*_{l(0)} = 1 + \frac{k}{2h}(f(1) - f(-1)) - \frac{\sqrt{a}k}{h}, \qquad u^*_{l(0)+1} = -1 + \frac{k}{2h}(f(1) - f(-1)) + \frac{\sqrt{a}k}{h}. \tag{4.19}$$

Then combining (4.19) and (4.1) leads to

$$s^* = \frac{h(u^*_{l(0)} + u^*_{l(0)+1})}{2k} = \frac{h}{2k}\frac{k}{h}(f(1) - f(-1)) = \frac{f(1) - f(-1)}{2} = s. \tag{4.20}$$

The lemma is proved.

Clearly, the above theorem and lemmas imply that, if the first order upwind or Lax-Friedrichs type scheme is used for the convection, the local random projection method yields a first order convergence in the shock location, or the solution in l^1. This is the result proved in [1]. Actually, such a method coincides with the global random projection method since after S_c these first order methods introduce intermediate points at most one point away from $x_{l(n)}$, thus there is no difference between the local and the global random projection. They become different only if there are intermediate points at least two points away from $x_{l(n)}$.

Since a second order TVD method reduces to a first order scheme, and in particular, if the solution at t_n is a step function with the discontinuity at $x_{l(n)}$, it is easy to check that after the convection, the second order TVD scheme gives exactly the same numerical results as its first order counterpart, thus the above error estimates hold for any TVD scheme in which no intermediate time step between t_n and t_{n+1} is taken. We summarize this by the following theorem.

Theorem 4.2 *If we choose $S_c(k)$ as any TVD method without intermediate time step between t_n and t_{n+1}, then*

$$|x_0 + st_n - l(n)h| \leq h\left[C_1 + C_2 \ln(2st_n/\lambda h)\right]. \tag{4.21}$$

For the scalar model (1.1), when the local random projection method is used, there is no difference whether a first order method or a formally second order –such as a TVD method, with no intermediate time step between t_n and t_{n+1}, is used for the convection. Thus, to solve a scalar equation such as (1.1), it is not necessary to use a formally second order method for the convection. A first order method suffices. Nevertheless, for nonlinear systems, such as the detonation wave computation, our numerical experiments indicate that a method with second order convection gives much better resolution than its first order counterpart.

Finally, we point out that, a method of line approach is adopted for S_c, namely, a TVD spatial discretization combined with a Runge-Kutta ODE solver for time discretization, could lead to incorrect speed. We illustrate this point using a Lax-Friedrichs type spatial discretization combined with the second order Runge-Kutta method for time, giving S_c as

$$u_j^{(1)} = u_j^n - kD_+ f(u_j^n), \quad u_j^{(2)} = u_j^{(1)} - kD_+ f(u_j^{(1)}), \tag{4.22}$$

$$u_j^{n+1} = \frac{1}{2}(u_j^n + u_j^{(2)}); \tag{4.23}$$

where

$$D_+ f(u_j) = \frac{1}{2h}(f(u_{j+1}) - f(u_{j-1})) - \frac{\sqrt{a}}{2h}(u_{j+1} - 2u_j + u_{j-1}),$$

with a a positive constant satisfying (4.18). Then, in general, the corresponding averaged numerical shock speed $s^* \neq s$. Particularly, if we choose $a = \frac{h^2}{k^2}$, then $s^* = \frac{s}{2}$.

To see this, using (4.22) and (2.2), one sees that

$$u_j^{(1)} = \begin{cases} 1, & j < l(0), \\ 1 + \lambda^-, & j = l(0), \\ -1 + \lambda^+, & j = l(0) + 1, \\ -1, & j > l(0) + 1; \end{cases} \tag{4.24}$$

where

$$\lambda = \frac{k}{2h}\left(f(1) - f(-1)\right) = \frac{sk}{h}, \qquad \lambda^{\pm} = \frac{k}{2h}\left(f(1) - f(-1)\right) \pm \frac{\sqrt{a}k}{h} = \lambda \pm \frac{\sqrt{a}k}{h}.$$

Similarly, from (4.22) and (4.24), one obtains

$$u_{l(0)}^{(2)} = 1 + \lambda^- - \frac{k}{2h}\Big[f(-1+\lambda^+) - f(1)\Big] + \frac{\sqrt{a}k}{2h}\left(-2 - \lambda + \frac{3\sqrt{a}k}{h}\right), \tag{4.25}$$

$$u_{l(0)+1}^{(2)} = -1 + \lambda^+ - \frac{k}{2h}\Big[f(-1) - f(1+\lambda^-)\Big] + \frac{\sqrt{a}k}{2h}\left(2 - \lambda - \frac{3\sqrt{a}k}{h}\right). \tag{4.26}$$

Then combining (4.23), (4.25), (4.26), (4.1) and (3.4) with $n = 0$, we have that

$$\begin{aligned}
s^* &= \frac{h(u_{l(0)}^* + u_{l(0)+1}^*)}{2k} = \frac{h(u_{l(0)}^0 + u_{l(0)}^{(2)} + u_{l(0)+1}^0 + u_{l(0)+1}^{(2)})}{4k} \\
&= \frac{h(u_{l(0)}^{(2)} + u_{l(0)+1}^{(2)})}{4k} = \frac{h(3\lambda - \lambda\sqrt{a}k/h + k(f(1+\lambda^-) - f(-1+\lambda^+))/2h)}{4k} \\
&= \frac{3s}{4} + \frac{f(1+\lambda^-) - f(-1+\lambda^+)}{8} - \frac{\lambda\sqrt{a}}{4} \\
&= s + \frac{f(1+\lambda^-) - f(1) - f(-1+\lambda^+) + f(-1)}{8} - \frac{\lambda\sqrt{a}}{4} \\
&\equiv s + \tilde{s}.
\end{aligned} \tag{4.27}$$

In general $\tilde{s} \neq 0$, thus $s^* \neq s$. Particularly, when we choose $a = \frac{h^2}{k^2}$, then $\lambda^\pm = \lambda \pm 1$. Thus from (4.27), we obtain

$$s^* = s + \frac{f(\lambda) - f(1) - f(\lambda) + f(-1)}{8} - \frac{\lambda}{4}\frac{h}{k} = s - \frac{s}{4} - \frac{s}{4} = \frac{s}{2}. \tag{4.28}$$

.

It is important to emphasize that the negative result in the above theorem has no generalization to the detonation wave calculations, as demonstrated in [1]. The reason is that in the detonation calculation only the concentration variable is projected into a step function, while all the physical conservative quantities, such as density, momentum and total energy, are not projected.

5 Extensions to problems with more general source terms

Consider the following hyperbolic conservation law with general source term

$$u_t + f(u)_x = -\frac{1}{\varepsilon}g(u), \tag{5.1}$$

where ε is the reaction time, f is a convex function of u, i.e. $f''(u) > 0$ and the source term $\frac{1}{\varepsilon}g(u)$ have finitely many simple zeros satisfying $ug(u) > 0$ for large $|u|$.

Without loss of generality, here we assume

$$\frac{1}{\varepsilon}g(u) = \frac{1}{\varepsilon}(u-a)(u-\alpha)(u-b)(u-\beta)(u-c), \tag{5.2}$$

with $a > \alpha > b > \beta > c$. The initial data is piecewise constant, i.e.

$$u(x,0) = u_0(x) = \begin{cases} a & x \leq x_0, \\ b & x_0 < x \leq x_1, \\ c & x > x_1; \end{cases} \tag{5.3}$$

where x_0 and x_1 are given points and satisfying $x_0 < x_1$.

The source term (5.2) admits five local equilibria, i.e., the unstable ones α and β, and the stable ones a, b and c. When the solution is at equilibria, the reaction term has no effect. Thus the exact solution is a two shock discontinuities connecting $u = a$ with $u = b$ and $u = b$ with $u = c$ and propagating with the speeds determined by the Rankine-Hugoniot jump conditions. In finite time, the shocks may collide.

Similar to the scalar model problem, a fractional step method is used. In the first step, the homogeneous conservation law

$$u_t + f(u)_x = 0 \tag{5.4}$$

is solved using any shock capturing scheme $S_c(k)$.

5.1 The local random projection method for the source term

In the second step, we solve for the source term

$$u_t = -\frac{1}{\varepsilon}g(u) \equiv -\frac{1}{\varepsilon}(u-a)(u-\alpha)(u-b)(u-\beta)(u-c) \tag{5.5}$$

by a local random projection method. First, notice that, although u may have some intermediate values between a and b or b and c after the convection step, the projection step always projects u to a, b or c. Therefore, at any time step $t_n = nk$, there are $l_1(n) = j_0$ and $l_2(n) = j_1$ with $j_0 \leq j_1$ integers, such that

$$u_j^n = \begin{cases} a, & \text{if } j \leq l_1(n), \\ b, & \text{if } l_1(n) < j \leq l_2(n), \\ c, & \text{if } j > l_2(n). \end{cases} \tag{5.6}$$

Suppose that $x_0 = l_1(0)h$, and $x_1 = l_2(0)h$. Since $x_0 \leq x_1$, then $l_1(0) \leq l_2(0)$. Let $u^* = S_c(k)u^n$. Then one can use the following algorithm recursively to find the positions of the shocks at time $t_{n+1} = (n+1)k$, i.e. $l_1(n+1)$ and $l_2(n+1)$, if the positions of the shocks at any time $t_n = nk$, i.e. $l_1(n)$ and $l_2(n)$, are known. Thus we can project the profile of u according to the shocks positions. The detail algorithm is

$\bar{S}_\theta(k):$ If $l_1(n) = l_2(n)$,

Set $l_1(n+1) := l_1(n) - 1$,

For $j = l_1(n)-1, \cdots, l_1(n)+1$ do : $l_1(n+1) = j$, if $u_j^* > \theta_n^1$;

$l_2(n+1) = l_1(n+1)$

$$u_j^{n+1} = \begin{cases} a, & \text{if } j \leq l_1(n+1), \\ c, & \text{if } j > l_1(n+1); \end{cases} \tag{5.7}$$

Else

Set $l_1(n+1) := l_1(n) - 1$,

For $j = l_1(n) - 1, \cdots, l_1(n) + 1 \quad$ do : $\; l_1(n+1) = j, \quad$ if $u_j^* > \theta_n^2$;

Set $l_2(n+1) := l_2(n) - 1$,

For $j = l_2(n) - 1, \cdots, l_2(n) + 1 \quad$ do : $\; l_2(n+1) = j, \quad$ if $u_j^* > \theta_n^3$;

If $l_1(n+1) > l_2(n+1)$ then

$l_1(n+1) = (l_1(n+1) + l_2(n+1))/2, \qquad l_2(n+1) = l_1(n+1)$;

End if

$$u_j^{n+1} = \begin{cases} a, & \text{if } j \le l_1(n+1), \\ b, & \text{if } l_1(n+1) < j \le l_2(n+1), \\ c, & \text{if } j > l_2(n+1); \end{cases} \tag{5.8}$$

End if

In this algorithm, where

$$\theta_n^1 = (a-c)\vartheta_n + c, \qquad \theta_n^2 = (a-b)\vartheta_n + b, \qquad \theta_n^3 = (b-c)\vartheta_n + c \tag{5.9}$$

with ϑ_n (see (2.14) for detail) being the van der Corput's sampling sequence on the interval $[0,1]$. Since the binary van der Corput's sampling sequence is equidistibuted on the interval $[0,1]$, the sampling sequences θ_n^1, θ_n^2 and θ_n^3 are equidistibuted on the intervals $[a,c]$, $[a,b]$ and $[b,c]$, respectively. The combination of the two steps gives the scheme:

$$S_{gs}(k): \qquad u^{n+1} = \bar{S}_\theta(k) S_c(k) u^n. \tag{5.10}$$

The stability condition for this algorithm is the usual CFL condition determined by the convection terms.

5.2 Numerical examples

Now we will test the above algorithm with a variety of numerical examples, including different kind of shocks collision. Among the numerical examples, the convection step is solved by using the Lax-Friedrichs scheme.

Example 5.1 We choose $f(u) = \frac{u^2}{2} + u$ and $\varepsilon = 10^{-6}$ in (5.1), $a = 1$, $\alpha = 0$, $b = -1$, $\beta = -2.5$, $c = -4$, $x_0 = 0.3$ and $x_1 = 1.8$ in (5.2) and (5.3) and solve the corresponding problem on the interval $[0,2]$ with 201 grid points by using the local random projection method (5.10). The mesh size $h = 0.01$ and we choose time step $k = 0.0005$. For this example the speeds of discontinuities $s_1 = \frac{f(1)-f(-1)}{2} = 1$ and $s_2 = \frac{f(-1)-f(-4)}{3} = -\frac{3}{2}$ corresponding to the two shocks initially at $x_0 = 0.3$ and $x_1 = 1.8$, respectively. The two shocks will collide at $t = 0.6$. After collision, the new shock moves at the speed $s_3 = \frac{f(1)-f(-4)}{5} = -\frac{1}{2}$. Figure 1 shows the computed results at $t = 0$, $t = 0.3$, $t = 0.6$ and $t = 1.0$.

Example 5.2 We choose $f(u) = \frac{u^2}{2} + u$ and $\varepsilon = 10^{-6}$ in (5.1), $a = 1$, $\alpha = 0$, $b = -1$, $\beta = -1.5$, $c = -2$, $x_0 = 0.2$ and $x_1 = 1.4$ in (5.2) and (5.3) and solve the corresponding problem on the interval $[0,2]$ with 201 grid points by using the local random projection method (5.10). The mesh size $h = 0.01$ and we choose time step $k = 0.001$. For this example the speeds of discontinuities $s_1 = \frac{f(1)-f(-1)}{2} = 1$ and $s_2 = \frac{f(-1)-f(-2)}{1} = -\frac{1}{2}$ corresponding to the two shocks initially at $x_0 = 0.2$ and $x_1 = 1.4$, respectively. The two shocks will collide at $t = 0.8$. After collision, the new shock moves at the speed $s_3 = \frac{f(1)-f(-2)}{3} = \frac{1}{2}$. Figure 2 shows the computed results at $t = 0$, $t = 0.4$, $t = 0.8$ and $t = 1.6$.

Example 5.3 We choose $f(u) = e^u$ and $\varepsilon = 10^{-6}$ in (5.1), $a = 2$, $\alpha = 1.5$, $b = 1$, $\beta = 0$, $c = -1$, $x_0 = 0.2$ and $x_1 = 1.0$ in (5.2) and (5.3) and solve the corresponding problem on the interval $[0, 2]$ with 201 grid points by using the local random projection method (5.10). The mesh size $h = 0.01$ and we choose time step $k = 0.0002$. For this example the speeds of discontinuities $s_1 = \frac{f(2)-f(1)}{1} = e^2 - e$ and $s_2 = \frac{f(1)-f(-1)}{2} = \frac{e-e^{-1}}{2}$ corresponding to the two shocks initially at $x_0 = 0.2$ and $x_1 = 1.0$, respectively. The two shocks will collide at $t = \frac{1.6}{2e^2-3e+e^{-1}} = 0.22886$. After collision, the new shock moves at the speed $s_3 = \frac{f(2)-f(-1)}{3} = \frac{e^2-e^{-1}}{3}$. Figure 3 shows the computed results at $t = 0$, $t = 0.1$, $t = 0.2$ and $t = 0.4$.

Figures 1-3 show that the local random projection method can capture the correct speeds of the discontinuities for the hyperbolic conservation laws with general stiff source terms when the source terms are underresolved. Although the location of the shock may be off by few grid points at each time step, but such a deviation never grows in time.

References

[1] W. Bao and S. Jin, The random projection method for hyperbolic conservation laws with stiff reaction terms, preprint.

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

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

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

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

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

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

[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] J. M. Hammersley and D. C. Handscomb, Monte Carlo Methods, Methuen, London, 1965.

[14] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49, 357-393, (1983).

[15] 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).

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

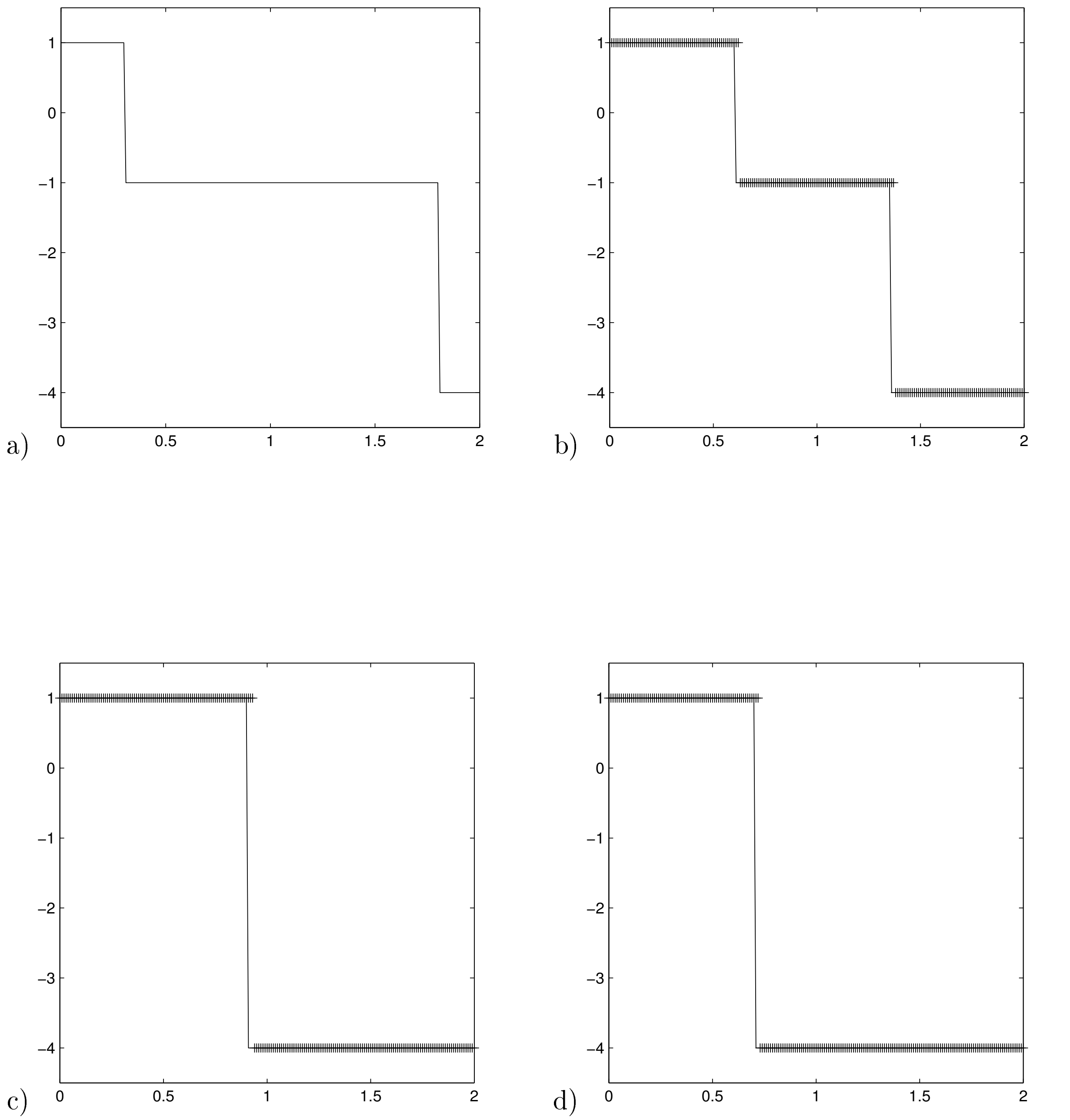

Figure 1. Numerical results by using the local random projection method (5.10) for example 5.1. $-$: true solution; $++$: computed solutions. a). $t = 0.0$, b). $t = 0.3$, c). $t = 0.6$ and d). $t = 1.0$.

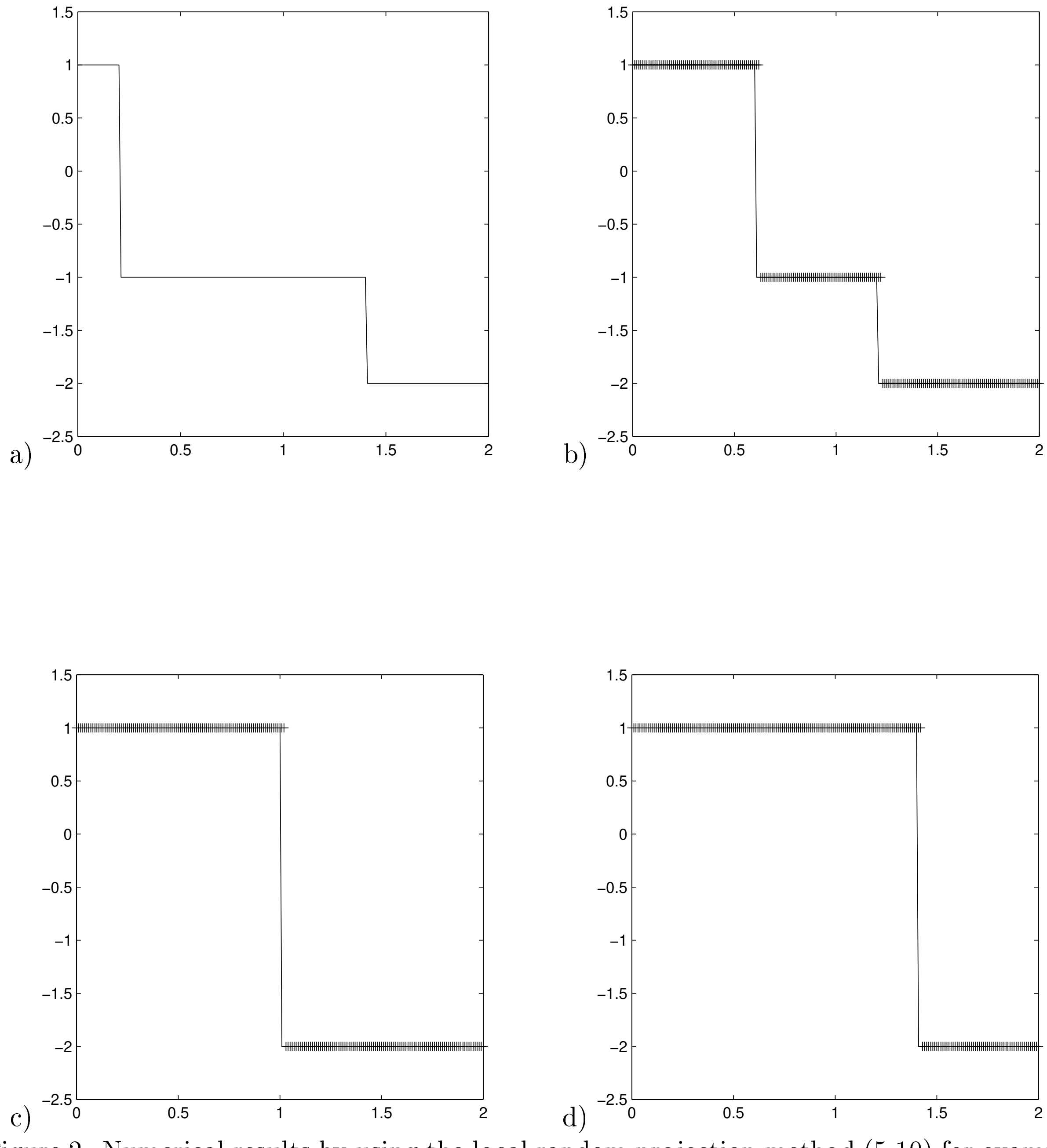

Figure 2. Numerical results by using the local random projection method (5.10) for example 5.2. $-$: true solution; $++$: computed solutions. a). $t = 0.0$, b). $t = 0.4$, c). $t = 0.8$ and d). $t = 1.6$.

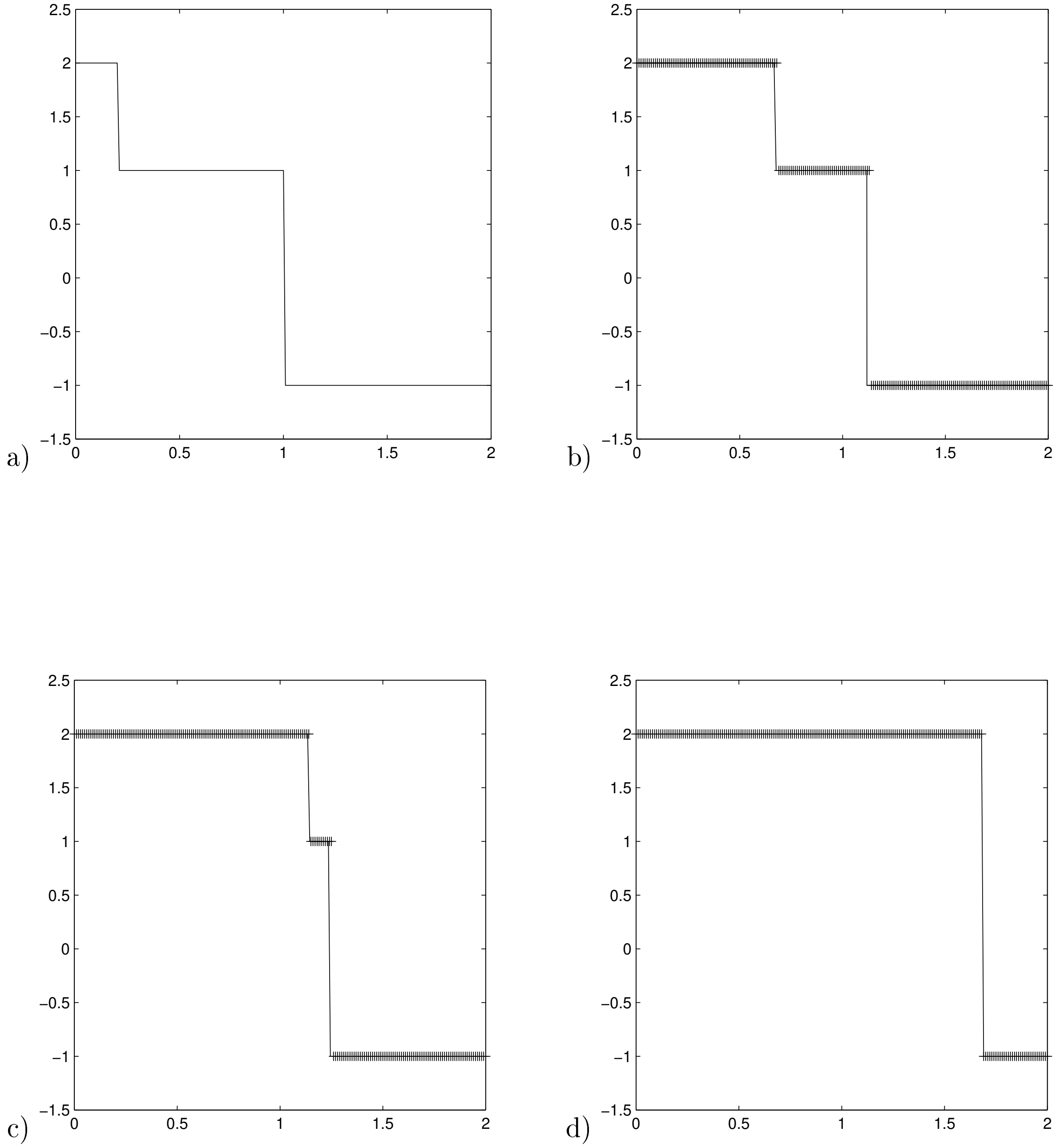

Figure 3. Numerical results by using the local random projection method (5.10) for example 5.3. $-$: true solution; $++$: computed solutions. a). $t = 0.0$, b). $t = 0.1$, c). $t = 0.2$ and d). $t = 0.4$.