Journal of Computational Physics 227 (2008) 9612–9627

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Superconvergence and time evolution of discontinuous Galerkin finite element solutions

Yingda Cheng [a], Chi-Wang Shu [b,*]

[a] Department of Mathematics and ICES, University of Texas, Austin, TX 78712, USA
[b] Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

ARTICLE INFO

Article history:
Received 23 November 2007
Received in revised form 31 March 2008
Accepted 10 July 2008
Available online 25 July 2008

Keywords:
Discontinuous Galerkin method
Superconvergence
Upwind flux
Projection
Error estimates

ABSTRACT

In this paper, we study the convergence and time evolution of the error between the discontinuous Galerkin (DG) finite element solution and the exact solution for conservation laws when upwind fluxes are used. We prove that if we apply piecewise linear polynomials to a linear scalar equation, the DG solution will be superconvergent towards a particular projection of the exact solution. Thus, the error of the DG scheme will not grow for fine grids over a long time period. We give numerical examples of P^k polynomials, with $1 \leqslant k \leqslant 3$, to demonstrate the superconvergence property, as well as the long time behavior of the error. Nonlinear equations, one-dimensional systems and two-dimensional equations are numerically investigated to demonstrate that the conclusions hold true for very general cases.

1. Introduction

In this paper, we consider smooth solutions of the following hyperbolic conservation laws

$$u_t + f(u)_x = b(x,t), \tag{1.1}$$

and

$$u_t + f(u)_x + g(u)_y = b(x,y,t), \tag{1.2}$$

where $u, f(u)$ and $g(u)$ can be either scalars or vectors. We study the convergence and time evolution of the error between the discontinuous Galerkin (DG) finite element solution and the exact solution, as well as the error between the DG solution and a particular type of projection of the exact solution. In this paper, we only consider the case when the upwind fluxes (for the nonlinear case, these include, for example, the Godunov flux or the Engquist-Osher flux) are used.

The type of DG method that we will discuss here is a class of finite element methods devised to solve hyperbolic conservation laws containing only first order spatial derivatives, e.g. [6,5,4,3,7]. Using completely discontinuous polynomial space for both the test and trial functions in the spatial variables and coupled with explicit and nonlinearly stable high order Runge–Kutta time discretization, the method has the advantage of flexibility for arbitrarily unstructured meshes, with a compact stencil, and with the ability to easily accommodate arbitrary h–p adaptivity. See, for example [8–10] for recent development and application of the DG methods.

The numerical error of the DG solution to (1.1) and (1.2) has been observed to stay level or grow very slowly with respect to the time t for P^k polynomials with $k \geqslant 1$. In [12], Zhang and Shu explicitly give the formulation of the DG solution

[*] Corresponding author. Tel.: +1 401 863 2549; fax: +1 401 863 1355.
E-mail addresses: ycheng@math.utexas.edu (Y. Cheng), shu@dam.brown.edu (C.-W. Shu).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.07.010

in the case of P^1 (piecewise linear) for the linear convection equation. The leading error term is shown to be of a constant magnitude independent of the time t. This motivates us to divide the numerical error into two parts, one part being the leading term that does not grow with time, the other part being a superconvergent term that grows at most linearly with time.

In [1,2], Adjerid et al. proved the superconvergence of the DG solutions at Radau points for ordinary differential equations. They have also made numerical experiments for the partial differential equations (PDE), although they have not provided a proof for such cases and have not considered the issue of time growth of the errors. In this paper, we give a proof for a similar superconvergence in the case of piecewise linear polynomials for linear PDEs and consider its impact on the time growth of the errors. We also demonstrate numerically that the conclusions hold true for very general cases, including higher order DG, nonlinear equations, systems, and two dimensions. The wind direction $f'(u)$ is no longer assumed to be positive, and it can vary in sign in the computational domain. In this case, upwind fluxes need to be used to achieve superconvergence, and the projection of the exact solution will depend on the sign of $f'(u)$ and is not uniform in all cells.

We remark that we are concerned with the error evolution of the semi-discrete DG solution which is not discretized in time in the proof, and is discretized in time with a higher order time integrator in the numerical experiments so that the spatial error dominates. In a practical setting, the time discretization error should also be considered, although we do not address this issue in this paper.

This paper is organized as follows: in Section 2, we consider linear constant coefficient scalar equations. We prove that for P^1 the error between the DG solution and a particular projection of the exact solution is superconvergent and thus the error between the DG solution and the exact solution will not grow for fine grids over a long time period. Specifically, the proof indicates that for a grid with mesh size h, the error will not grow until the time reaches $O(\frac{1}{\sqrt{h}})$. Numerical experiments indicate even better performance, that the error does not grow until the time reaches $O(\frac{1}{h})$. We give numerical examples for the case of P^k with $0 \leqslant k \leqslant 3$ and show that for $k \geqslant 1$, this superconvergence property always holds true. In Section 3, we generalize the discussion into linear equations with variable coefficients. Numerical results for nonlinear equations, one-dimensional systems and two-dimensional equations are given in Sections 4–6, respectively. Finally, concluding remarks and remarks on future work are provided in Section 7.

2. Linear equations with constant coefficients

In this section, we consider the following equation

$$\begin{cases} u_t + u_x = 0 \\ u(x,0) = u_0(x) \\ u(0,t) = u(2\pi, t) \end{cases} . \tag{2.1}$$

Here, $u_0(x)$ is a smooth 2π-periodic function.

The usual notation of the DG method is adopted. If we want to solve this equation on the interval $I = [a,b]$, first we divide it into N cells as follows

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = b \tag{2.2}$$

We denote

$$I_j = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}), \quad x_j = \frac{1}{2}\left(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}}\right), \tag{2.3}$$

as the cells and cell centers respectively, and $V_h^k = \{v : v|_{I_j} \in P^k(I_j), j = 1, \ldots, N\}$ to be the approximation space, where $P^k(I_j)$ denotes all polynomials of degree at most k on I_j. The DG scheme using the upwind flux will become: find $u_h \in V_h^k$, such that

$$\int_{I_j} (u_h)_t v_h \mathrm{d}x - \int_{I_j} u_h (v_h)_x \mathrm{d}x + u_h^- v_h^- |_{j+\frac{1}{2}} - u_h^- v_h^+ |_{j-\frac{1}{2}} = 0 \tag{2.4}$$

holds for any $v_h \in V_h^k$. Here and below $(v_h)_{j+\frac{1}{2}}^- = v_h(x_{j+\frac{1}{2}}^-)$ denotes the left limit of the function v_h at the discontinuity point $x_{j+\frac{1}{2}}$. Likewise for v_h^+.

In addition, we define $P_h^- u$ to be a projection of u into V_h^k, such that

$$\int_{I_j} P_h^- u \; v_h \mathrm{d}x = \int_{I_j} u v_h \mathrm{d}x \tag{2.5}$$

for any $v_h \in P^{k-1}$ on I_j, where k is the polynomial degree of the DG solution, and

$$(P_h^- u)^- = u^- \quad \text{at } x_{j+1/2}. \tag{2.6}$$

Notice that this special projection is used in the error estimates of the DG methods to derive optimal L^2 error bounds in the literature, e.g. in [13]. We are going to show that indeed the numerical solution is closer to this special projection of the exact solution than to the exact solution itself. Let us denote $e = u - u_h$ to be the error between the exact solution and numerical solution, $\varepsilon = u - P_h^- u$ to be the projection error, and $\bar{e} = P_h^- u - u_h$ to be the error between the numerical solution and the projection of the exact solution.

2.1. The case of P^1

In this subsection, we consider the piecewise linear case of $k = 1$. We prove that for this particular case, the error between the numerical solution and the particular projection of the exact solution $\bar{e}$ will achieve at least $(k + \frac{3}{2})$th order of superconvergence.

Theorem 2.1. *Let u be the exact solution of the Eq. (2.1), and u_h be the DG solution of (2.4) with the initial condition $u_h(\cdot, 0) = P_h^- u_0$. Here, h denotes the mesh size. In the case of $k = 1$ and uniform meshes, we have the following error estimate:*

$$||\bar{e}(\cdot, t)||_{L^2} \leqslant C_1(t + 1)h^{5/2}, \tag{2.7}$$

and

$$||e(\cdot, t)||_{L^2} \leqslant C_1 t h^{5/2} + C_2 h^2, \tag{2.8}$$

where C_1 and C_2 are constants which do not depend on t or h.

Proof. Since u satisfies (2.1), we can easily check that

$$\int_{I_j} u_t v_h \mathrm{d}x - \int_{I_j} u(v_h)_x \mathrm{d}x + u^- v_h^-|_{j+\frac{1}{2}} - u^- v_h^+|_{j-\frac{1}{2}} = 0 \tag{2.9}$$

holds for any $v_h \in V_h^1$. Combined with (2.4), we have the error equation

$$\int_{I_j} e_t v_h \mathrm{d}x - \int_{I_j} e(v_h)_x \mathrm{d}x + e^- v_h^-|_{j+\frac{1}{2}} - e^- v_h^+|_{j-\frac{1}{2}} = 0 \tag{2.10}$$

which holds true for any $v_h \in V_h^1$. Taking $v_h = \bar{e}$, we obtain

$$\int_I (\bar{e})_t \bar{e} \mathrm{d}x + \int_I \varepsilon_t \bar{e} \mathrm{d}x - \int_{I_j} \varepsilon \bar{e}_x \mathrm{d}x - \int_{I_j} \bar{e}\bar{e}_x \mathrm{d}x + e^- \bar{e}^-|_{j+\frac{1}{2}} - e^- \bar{e}^+|_{j-\frac{1}{2}} = 0. \tag{2.11}$$

By the property (2.5) of the projection P_h^-, we have

$$\int_{I_j} \varepsilon \bar{e}_x \mathrm{d}x = 0$$

since $\bar{e}_x$ is a polynomial of degree at most $k - 1$ in I_j. By the property (2.6) of the projection P_h^-, we have

$$e_{j+\frac{1}{2}}^- = \varepsilon_{j+\frac{1}{2}}^- + \bar{e}_{j+\frac{1}{2}}^- = \bar{e}_{j+\frac{1}{2}}^-$$

for all j. (2.11) now becomes

$$\int_I (\bar{e})_t \bar{e} \mathrm{d}x + \int_I \varepsilon_t \bar{e} \mathrm{d}x - \int_{I_j} \bar{e}\bar{e}_x \mathrm{d}x + \bar{e}^- \bar{e}^-|_{j+\frac{1}{2}} - \bar{e}^- \bar{e}^+|_{j-\frac{1}{2}} = 0$$

or

$$\int_I (\bar{e})_t \bar{e} \mathrm{d}x + \int_I \varepsilon_t \bar{e} \mathrm{d}x + \widehat{F}_{j+\frac{1}{2}} - \widehat{F}_{j-\frac{1}{2}} + \frac{1}{2}[\bar{e}]_{j+\frac{1}{2}}^2 = 0 \tag{2.12}$$

where $[v] = v^+ - v^-$ denotes the jump of v, with

$$\widehat{F}_{j+\frac{1}{2}} = -\frac{1}{2}(\bar{e}_{j+\frac{1}{2}}^+)^2 + \bar{e}_{j+\frac{1}{2}}^- \bar{e}_{j+\frac{1}{2}}^+.$$

Summing the equality (2.12) over j and noticing the periodic boundary condition, we obtain

$$\int_I (\bar{e})_t \bar{e} \mathrm{d}x + \frac{1}{2}\sum_j [\bar{e}]_{j+\frac{1}{2}}^2 + \int_I \varepsilon_t \bar{e} \mathrm{d}x = 0.$$

Thus,

$$\frac{\mathrm{d}}{\mathrm{d}t}||\bar{e}||_{L^2}^2 \leqslant 2\left|\int_I \varepsilon_t \bar{e} \mathrm{d}x\right|. \tag{2.13}$$

To simplify the discussion, we first take a single Fourier mode and assume $u_0(x) = \sin(x)$. In this case, we know the exact solution of (2.1) is

$$u(x,t) = \sin(x - t).$$

Hence, we can compute the projection of u to be

$$P_h^- u = c_j + d_j \frac{x - x_j}{h}$$

on I_j, where $c_j = \frac{1}{h}(-\cos(x_{j+\frac{1}{2}} - t) + \cos(x_{j-\frac{1}{2}} - t))$, and $d_j = 2(\sin(x_{j+\frac{1}{2}} - t) - c_j)$. On the other hand, the numerical solution u_h can be expressed in the following form

$$u_h|_{I_j} = u_{j-\frac{1}{4}}\phi_{j-\frac{1}{4}}(x) + u_{j+\frac{1}{4}}\phi_{j+\frac{1}{4}},$$

where $\phi_{j-\frac{1}{4}}(x) = \frac{2}{h}(x_{j+\frac{1}{4}} - x)$ and $\phi_{j+\frac{1}{4}}(x) = \frac{2}{h}(x - x_{j-\frac{1}{4}})$ are the basis functions. The value of $u_{j\pm\frac{1}{4}}$ can be obtained using the techniques in [12]

$$u_{j-\frac{1}{4}} = \sin(x_j - t) - \frac{h}{4}\cos(x_j - t) + \frac{h^2}{96}\sin(x_j - t) + h^3\left(\frac{\cos(x_j - t)}{1152} - \frac{t}{72}\sin(x_j - t)\right) + O(h^4),$$

and

$$u_{j+\frac{1}{4}} = \sin(x_j - t) + \frac{h}{4}\cos(x_j - t) - \frac{7h^2}{96}\sin(x_j - t) + h^3\left(-\frac{11\cos(x_j - t)}{384} - \frac{t}{72}\sin(x_j - t)\right) + O(h^4).$$

After Taylor expansion, we have

$$\bar{e} = -\frac{h^2}{96}\sin(x_j - t) + \left(\frac{1}{72}(\cos(x_j - t) + t\sin(x_j - t)) + \left(\frac{5}{298}\cos(x_j - t) + \frac{1}{18}t\sin(x_j - t)\right)\frac{x - x_j}{h}\right)h^3 + O(h^4) \quad \text{on} \quad I_j.$$

By similar arguments, we can prove that if $u_0(x) = \sin(kx)$, then

$$\begin{aligned}\bar{e} = &-\frac{h^2}{96}k^2\sin(k(x_j - t)) \\ &+ \left(\frac{1}{72}(\cos(k(x_j - t)) + t\sin(k(x_j - t))) + \left(\frac{5}{298}\cos(k(x_j - t)) + \frac{1}{18}t\sin(k(x_j - t))\right)\frac{x - x_j}{h}\right)k^3h^3 + O(k^4h^4) \quad \text{on} \quad I_j.\end{aligned}$$

and if $u_0(x) = \cos(kx)$, then

$$\begin{aligned}\bar{e} = &-\frac{h^2}{96}k^2\cos(k(x_j - t)) \\ &+ \left(\frac{1}{72}(-\sin(k(x_j - t)) + t\cos(k(x_j - t))) + \left(-\frac{5}{298}\sin(k(x_j - t)) + \frac{1}{18}t\cos(k(x_j - t))\right)\frac{x - x_j}{h}\right)k^3h^3 \\ &+ O(k^4h^4) \quad \text{on } I_j.\end{aligned}$$

For any smooth initial condition $u_0(x)$, the Fourier series of u_0 is $u_0(x) = \sum_{k=0}^{\infty}[a_k\sin(kx) + b_k\cos(kx)]$ where $a_k = \frac{1}{\pi}\int_0^{2\pi}u_0(x)\sin(kx)\mathrm{d}x$, $b_k = \frac{1}{\pi}\int_0^{2\pi}u_0(x)\cos(kx)\mathrm{d}x$. Then the exact solution to (2.1) is

$$u(x,t) = u_0(x - t) = \sum_{k=0}^{\infty}[a_k\sin(k(x - t)) + b_k\cos(k(x - t))]. \tag{2.14}$$

It is well known that

$$a_k, b_k = O\left(\frac{1}{k^p}\right) \tag{2.15}$$

if $u_0 \in C^p$. Since P_h^- is a linear operator,

$$P_h^- u(x,t) = \sum_{k=0}^{\infty}[a_kP_h^-\sin(k(x - t)) + b_kP_h^-\cos(k(x - t))]. \tag{2.16}$$

We can also prove that

$$u_h(x,t) = \sum_{k=0}^{\infty}[a_ku_h^k(x,t) + b_kw_h^k(x,t)], \tag{2.17}$$

where $u_h^k(x,t)$ is the DG solution to

$$\begin{cases} u_t + u_x = 0 \\ u(x,0) = \sin(kx) \\ u(0,t) = u(2\pi,t) \end{cases},$$

and $w_h^k(x,t)$ is the DG solution to

$$\begin{cases} u_t + u_x = 0 \\ u(x,0) = \cos(kx)\,. \\ u(0,t) = u(2\pi,t) \end{cases}$$

To prove this, we only need the following equations

$$\int_{I_j} \sum_{k=0}^{\infty} a_k (u_h^k)_t v_h \mathrm{d}x = \sum_{k=0}^{\infty} a_k \int_{I_j} (u_h^k)_t v_h \mathrm{d}x$$
$$\int_{I_j} \sum_{k=0}^{\infty} a_k u_h^k (v_h)_x \mathrm{d}x = \sum_{k=0}^{\infty} a_k \int_{I_j} u_h^k (v_h)_x \mathrm{d}x$$

to hold for any $v_h \in V_h^1$. They can be easily proved using the dominated convergence theorem.

From (2.14), (2.15) with $p \geqslant 6$, (2.16) and (2.17), we have

$$\begin{aligned}\bar{e} = & -\frac{h^2}{96}\sum_{k=0}^{\infty}[a_k k^2 \sin(k(x_j - t)) + b_k k^2 \cos(k(x_j - t))] + \sum_{k=0}^{\infty}\left[a_k \frac{1}{72}\cos(k(x_j - t)) - b_k \frac{1}{72}\sin(k(x_j - t))\right] k^3 h^3 \\ & + \sum_{k=0}^{\infty}\left[a_k \frac{t}{72}\sin(k(x_j - t)) + b_k \frac{t}{72}\cos(k(x_j - t))\right] k^3 h^3 \\ & + \sum_{k=0}^{\infty}\left[a_k \frac{5}{298}\cos(k(x_j - t)) - b_k \frac{5}{298}\sin(k(x_j - t))\right] \frac{x - x_j}{h} k^3 h^3 \\ & + \sum_{k=0}^{\infty}\left[a_k \frac{t}{18}\sin(k(x_j - t)) + b_k \frac{t}{18}\cos(k(x_j - t))\right] \frac{x - x_j}{h} k^3 h^3 + O(h^4) \end{aligned} \tag{2.18}$$

By the properties of the projection P_h^-, we know that on I_j

$$\int_{I_j} \varepsilon_t \mathrm{d}x = 0$$

and $\varepsilon_t \leqslant C_1 h^2$ where the constant C_1 does not depend on t. Moreover, by (2.15) with $p \geqslant 5$, there exists C_2, such that

$$\left|\sum_{k=0}^{\infty}\left[a_k \frac{5}{298}\cos(k(x_j - t)) - b_k \frac{5}{298}\sin(k(x_j - t))\right] k^3\right| \leqslant C_2$$

and

$$\left|\sum_{k=0}^{\infty}\left[a_k \frac{1}{18}\sin(k(x_j - t)) + b_k \frac{1}{18}\cos(k(x_j - t)))\right] k^3\right| \leqslant C_2$$

for all j. Thus, by (2.18), we have

$$\left|\int_{I_j} \varepsilon_t \bar{e} \mathrm{d}x\right| \leqslant C_1 C_2 (1 + t) h^6$$

with the constants C_1, C_2 independent of t. Summing over j and using (2.13), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}||\bar{e}||_{L^2}^2 \leqslant Ch^5(1 + t)$$

where the constant C is independent of t. Thus,

$$||\bar{e}(\cdot,t)||_{L^2} \leqslant Ch^{5/2}(1 + t)$$

since the initial condition is chosen as $u_h(\cdot,0) = P_h^- u_0$. We have therefore proved (2.7), thus (2.8) follows. □

From (2.8), if $t \leqslant \frac{C}{\sqrt{h}}$, we will have the following bound for the L^2 error,

$$||e(x,t)||_{L^2} \leqslant (C_1 C + C_2)h^2 = Dh^2,$$

where D is a constant that does not depend on t or h. This is what we mean by saying that the error of the DG scheme will not grow for fine grids (small h) over a long time period (proportional to $\frac{1}{\sqrt{h}}$).

Even though our proof is provided for the simple scalar equation (2.1), the same proof applies also to any linear hyperbolic system

$$u_t + Au_x = 0$$

where A is a constant matrix which is diagonalizable with real eigenvalues. This is because the PDE as well as the DG scheme can be diagonalized into decoupled scalar equations.

Next, we use some numerical examples to demonstrate the superconvergence of $\bar{e}$ and the long time behavior of e.

Example 1. We solve the one-dimensional equation

$$\begin{cases} u_t + u_x = 0 \\ u(x,0) = \sin(x) \\ u(0,t) = u(2\pi, t) \end{cases} \quad . \tag{2.19}$$

The exact solution to this problem is

$$u(x,t) = \sin(x-t). \tag{2.20}$$

In the computation, we use the L^2 projection of the initial condition as our numerical initial condition (even though Theorem 2.1 requires the initial condition to be $u_h(\cdot,0) = P_h^- u_0$, we have observed little difference if we use the usual L^2 projection of the initial condition instead). The strong stability preserving (SSP) ninth-order time discretization from [11] is adopted as the time discretization to make the time error negligible compared to the space errors. Uniform meshes are used in the calculation.

Table 2.1 lists the numerical errors, $\bar{e}$ and e, and their orders for different final time T. We conclude that at any time, we can always observe third order accuracy for $\bar{e}$, indicating that the error bound for $\bar{e}$ obtained in (2.7) is not optimal. For longer time, for example from $T = 10$ to $T = 100, \bar{e}$ grows linearly with respect to the time t. Meanwhile, the error e is of second order, and this error does not grow much with respect to the time t for fine grids. This is a natural conclusion of the theorem, since the threshold time for the growth of e is proportional to $\frac{1}{\sqrt{h}}$. In this example, the numerical results indicate that it is actually proportional to $\frac{1}{h}$. For bigger N, namely, smaller h, this time is longer than what we have observed in the table.

If instead of the periodic boundary condition, we impose $u(0,t) = \sin(-t)$ and solve the initial-boundary value problem, we observe similar results as shown in Table 2.2. It seems to indicate that, although the proof relies on Fourier analysis and hence is restricted to periodic cases, the results are in fact more general for a broader class of initial-boundary value problems. We have also tested initial-boundary value problems for the remaining examples below and have obtained essentially the same results as for the periodic cases, however we will not list them to save space.

We have also used a non-uniform mesh which is a 10% random perturbation of the uniform mesh. Table 2.3 lists the order and time evolution of $\bar{e}$ and e in this case. We can see that all the conclusions for uniform meshes also hold true for this non-

Table 2.1
The errors $\bar{e}$ and e for Example 1 when using P^1 polynomials and SSP ninth-order time discretization on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 10$		$T = 100$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	4.60E−04	–	3.04E−03	–	2.96E−02	–
	40	5.80E−05	2.99	3.82E−04	2.99	3.79E−03	2.97
	80	7.26E−06	3.00	4.79E−05	3.00	4.75E−04	2.99
	160	9.08E−07	3.00	5.99E−06	3.00	5.95E−05	3.00
e	20	4.21E−03	–	5.16E−03	–	2.99E−02	–
	40	1.06E−03	1.99	1.12E−03	2.20	3.93E−−03	2.92
	80	2.65E−04	2.00	2.69E−04	2.06	5.44E−04	2.85
	160	6.64E−05	2.00	6.66E−05	2.02	8.91E−05	2.61

Table 2.2
The errors $\bar{e}$ and e for Example 1 solved as an initial-boundary value problem when using P^1 polynomials and SSP ninth-order time discretization on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 10$		$T = 100$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	4.74E−04	–	1.28E−03	–	1.06E−03	–
	40	6.02E−05	2.98	1.60E−04	3.00	1.33E−04	3.00
	80	7.57E−06	2.99	2.00E−05	3.00	1.65E−05	3.01
	160	9.50E−07	3.00	2.50E−06	3.00	2.16E−06	2.93
e	20	4.22E−03	–	4.43E−03	–	4.37E−03	–
	40	1.06E−03	1.99	1.07E−03	2.04	1.07E−03	2.03
	80	2.65E−04	2.00	2.66E−04	2.01	2.66E−04	2.01
	160	6.64E−05	2.00	6.64E−05	2.00	6.64E−05	2.00

Table 2.3
The errors $\bar{e}$ and e for Example 1 when using P^1 polynomials and SSP ninth-order time discretization on a random mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 10$		$T = 100$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	5.09E−04	–	3.18E−03	–	3.09E−02	–
	40	7.35E−05	2.79	4.01E−04	2.99	3.98E−03	2.96
	80	9.05E−06	3.02	4.91E−05	3.03	4.88E−04	3.03
	160	1.45E−06	2.64	6.24E−06	2.98	6.19E−05	2.98
	320	2.04E−07	2.83	7.81E−07	3.00	7.74E−06	3.00
e	20	4.40E−03	–	5.36E−03	–	3.12E−02	–
	40	1.09E−03	2.01	1.16E−03	2.21	4.13E−03	2.92
	80	2.71E−04	2.01	2.76E−04	2.07	5.58E−04	2.89
	160	6.86E−05	1.98	6.88E−05	2.00	9.23E−05	2.60
	320	1.72E−05	2.00	1.72E−05	2.00	1.88E−05	2.30

uniform mesh. Notice that for a 10% random perturbation of the uniform mesh, the ratio of mesh sizes between adjacent cells can be as large as $\frac{3}{2}$. We have also tested this example using a more severely non-uniform mesh which is a 30% random perturbation of the uniform mesh. The numerical results (which are not included to save space) still show superconvergence, although the errors for $\bar{e}$ do not achieve third order (the order is around 2.5). Notice that for a 30% random perturbation of the uniform mesh, the ratio of mesh sizes between adjacent cells can be as large as 4.

Example 2. We still solve the same equation as in the previous example but with a different initial condition.

$$\begin{cases} u_t + u_x = 0 \\ u(x,0) = e^{\sin(x)} \\ u(0,t) = u(2\pi,t) \end{cases} \tag{2.21}$$

The exact solution is

$$u(x,t) = e^{\sin(x-t)}. \tag{2.22}$$

Notice that the initial condition contains all the Fourier modes, not just a single mode as in Example 1.

Table 2.4 is obtained by a non-uniform mesh which is a 10% random perturbation of the uniform mesh. All conclusions for Example 1 seem to hold true for this example as well.

2.2. The case of P^2

While the technique of the proof for Theorem 2.1 in the previous section can be used in principle also for P^k cases with $k > 1$, the algebraic manipulations become prohibitively complicated. We have therefore only used numerical experiments to demonstrate the generality of the conclusion for the remaining cases.

In this subsection, we provide numerical tests of Examples 1 and 2 when piecewise P^2 polynomials are used.

Example 1a. Table 2.5 lists the numerical errors, $\bar{e}$ and e, obtained on a uniform mesh. When $T = 1, \bar{e}$ is of fourth order. When $T = 100$ and $T = 1000$, if we keep on refining the grids, the order seems also to converge to four. From $T = 100$ to $T = 1000$, the error $\bar{e}$ grows linearly with respect to time. In this example, e is of third order and does not grow with respect to time (until the final time $T = 1000$ that we have run) for the fine grids.

Table 2.4
The errors $\bar{e}$ and e for Example 2 when using P^1 polynomials and SSP ninth-order time discretization on a random mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 10$		$T = 100$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	2.05E−03	–	1.66E−02	–	1.08E−01	–
	40	2.80E−04	2.87	2.32E−03	2.84	2.07E−02	2.38
	80	3.43E−05	3.03	2.89E−04	3.01	2.83E−03	2.87
	160	4.53E−06	2.92	3.67E−05	2.98	3.66E−04	2.95
	320	5.79E−07	2.97	4.59E−06	3.00	4.58E−05	3.00
e	20	6.79E−03	–	1.78E−02	–	1.08E−01	–
	40	1.79E−03	1.92	2.88E−03	2.63	2.08E−02	2.37
	80	4.31E−04	2.05	5.28E−04	2.45	2.87E−03	2.86
	160	1.11E−04	1.95	1.17E−04	2.18	3.82E−04	2.91
	320	2.77E−05	2.01	2.83E−05	2.04	5.35E−05	2.84

Table 2.5
The errors $\bar{e}$ and e for Example 1a when using P^2 polynomials and SSP ninth-order time discretization on a uniform mesh of N cells ($CFL = 0.1$)

	N	$T = 1$		$T = 100$		$T = 1000$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	4.17E−06	–	3.02E−05	–	2.99E−04	–
	40	2.62E−07	3.99	9.74E−07	4.95	9.38E−06	4.99
	80	1.64E−08	4.00	3.36E−08	4.86	2.94E−07	5.00
	160	1.02E−09	4.00	1.37E−09	4.61	9.91E−09	4.89
e	20	1.07E−04	–	1.11E−04	–	3.18E−04	–
	40	1.34E−05	3.00	1.34E−05	3.05	1.63E−05	4.28
	80	1.67E−06	3.00	1.67E−06	3.00	1.70E−06	3.28
	160	2.09E−07	3.00	2.09E−07	3.00	2.09E−07	3.02

Example 2a. Table 2.6 lists the numerical errors and orders for $\bar{e}$ and e for the test case in Example 2. When $T = 1$, we observe fourth order accuracy for $\bar{e}$. When $T = 100$ and $T = 1000, \bar{e}$ seems to have fifth order accuracy. In this example, e is of third order until $T = 100$ but is still fourth order at $T = 1000$, and it does not grow with respect to time until $T = 100$ but grows between $T = 100$ and $T = 1000$, for the grids that we have run.

2.3. The case of P^3

In this subsection, we provide numerical tests of Examples 1 and 2 when piecewise P^3 polynomials are used.

Example 1b. We use piecewise P^3 polynomials and the SSP ninth order time discretization on a uniform mesh for the test problem in Example 1. Table 2.7 gives the numerical orders and time evolution for $\bar{e}$ and e. $\bar{e}$ achieves $(k+2)$th order superconvergence and grows linearly from $T = 100$ to $T = 500$. e achieves $(k + 1)$th order convergence and does not grow with respect to the time t until the time that we have computed, for fine grids.

Example 2b. We use piecewise P^3 polynomials and SSP ninth order time discretization on a uniform mesh for the test case in Example 2. The results are listed in Table 2.8. The conclusion is similar to that in Example 1b, however, we seem to get higher order accuracy here than in Example 1b for the grid sizes and times that we have used.

Table 2.6
The errors $\bar{e}$ and e for Example 2a when using P^2 polynomials and SSP ninth-order time discretization on a uniform mesh of N cells ($CFL = 0.1$)

	N	$T = 1$		$T = 100$		$T = 1000$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	2.64E−05	–	1.22E−03	–	1.01E−02	–
	40	1.59E−06	4.05	4.07E−05	4.90	4.02E−04	4.64
	80	9.77E−08	4.03	1.29E−06	4.98	1.28E−05	4.97
	160	6.08E−09	4.01	4.07E−08	4.98	4.02E−07	5.00
e	20	2.94E−04	–	1.25E−03	–	1.01E−02	–
	40	3.67E−05	3.00	5.48E−05	4.51	4.04E−04	4.64
	80	4.59E−06	3.00	4.77E−06	3.52	1.36E−05	4.89
	160	5.74E−07	3.00	5.76E−07	3.05	7.01E−07	4.28

Table 2.7
The errors $\bar{e}$ and e for Example 1b when using P^3 polynomials and SSP ninth-order time discretization on a uniform mesh of N cells ($CFL = 0.1$)

	N	$T = 10$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	5	7.87E−05	–	2.42E−04	–	1.18E−03	–
	10	1.70E−06	5.53	2.58E−06	6.55	9.72E−06	6.92
	20	5.40E−08	4.98	5.60E−08	5.53	9.27E−08	6.71
	40	1.68E−09	5.00	1.69E−09	5.05	1.79E−09	5.69
e	5	5.21E−04	–	5.74E−04	–	1.29E−03	–
	10	3.30E−05	3.98	3.30E−05	4.12	3.43E−05	5.23
	20	2.06E−06	4.00	2.06E−06	4.00	2.07E−06	4.05
	40	1.29E−07	4.00	1.29E−07	4.00	1.29E−07	4.00

Table 2.8
The errors $\bar{e}$ and e for Example 2b when using P^3 polynomials and SSP ninth-order time discretization on a uniform mesh of N cells ($CFL = 0.1$)

	N	$T = 10$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	5	5.40E−03	–	2.86E−02	–	6.70E−02	–
	10	8.70E−05	5.96	7.65E−04	5.22	3.24E−03	4.37
	20	1.11E−06	6.30	7.50E−06	6.67	3.71E−05	6.45
	40	2.61E−08	5.40	6.57E−08	6.84	3.04E−07	6.93
e	5	5.50E−03	–	2.89E−02	–	6.70E−02	–
	10	2.09E−04	4.72	7.88E−04	5.19	3.24E−03	4.37
	20	1.22E−05	4.09	1.43E−05	5.79	3.91E−05	6.38
	40	7.65E−07	4.00	7.67E−07	4.22	8.22E−07	5.57

Table 2.9
The error e for Example 1 when using P^0 polynomials and SSP ninth-order time discretization on a uniform mesh of *320* cells ($CFL = 0.5$)

T	L^2 error
1	7.99E−03
10	6.62E−02
100	4.42E−01

2.4. The case of P^0

Finally we consider the case of P^0. In this case, the projection P_h^- can no longer be defined. We compute e for Example 1 when $N = 320$ and list the L^2 errors in Table 2.9. Unlike the cases of P^1, P^2, and P^3, this time e grows with respect to time even for fine grids.

3. Linear equations with variable coefficients

In this section, we generalize the discussion of Section 2 to linear variable coefficient equations.

Example 3. We solve the following equation

$$\begin{cases} u_t + (a(x)u)_x = b(x,t) \\ u(x,0) = \sin(x) \\ u(0,t) = u(2\pi,t) \end{cases}, \tag{3.1}$$

where $a(x)$ and $b(x,t)$ are given by

$$a(x) = \sin(x) + 2,$$
$$b(x,t) = (\sin(x) + 3)\cos(x+t) + \cos(x)\sin(x+t).$$

The exact solution to this problem is

$$u(x,t) = \sin(x+t). \tag{3.2}$$

Since $a(x) > 0$, we can still use the upwind fluxes. The projection P_h^- is defined in the same way as before. We use the five stage, fourth order SSP Runge–Kutta discretization in time and take $\Delta t = CFL\ h^2$ to reduce the time errors. We test the example with both P^1 and P^2 polynomials. The results in Tables 3.1 and 3.2 show that $\bar{e}$ achieves $(k+2)$th order superconvergence,

Table 3.1
The errors $\bar{e}$ and e for Example 3 when using P^1 polynomials on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	2.55E−04	–	1.35E−04	–	1.64E−04	–
	40	3.24E−05	2.98	2.31E−05	2.54	1.94E−05	3.08
	80	4.06E−06	3.00	3.79E−06	2.61	3.08E−06	2.65
	160	5.09E−07	3.00	5.05E−07	2.91	4.31E−07	2.85
e	20	4.25E−03	–	4.26E−03	–	4.24E−03	–
	40	1.06E−03	2.00	1.06E−03	2.00	1.06E−03	2.00
	80	2.65E−04	2.00	2.66E−04	2.00	2.65E−04	2.00
	160	6.64E−05	2.00	6.64E−05	2.00	6.64E−05	2.00

Table 3.2
The errors $\bar{e}$ and e for Example 3 when using P^2 polynomials on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	4.29E–06	–	4.19E–06	–	4.20E–06	–
	40	2.64E–07	4.02	2.61E–07	4.00	2.62E–07	4.00
	80	1.65E–08	4.00	1.63E–08	4.00	1.63E–08	4.00
	160	1.03E–09	4.00	1.02E–09	4.00	1.02E–09	4.00
e	20	1.07E–04	–	1.07E–04	–	1.07E–04	–
	40	1.34E–05	3.00	1.34E–05	3.00	1.34E–05	3.00
	80	1.67E–06	3.00	1.67E–06	3.00	1.67E–06	3.00
	160	2.09E–07	3.00	2.09E–07	3.00	2.09E–07	3.00

and it does not grow with respect to time for the meshes and times that we have used. The error e achieves the expected $(k + 1)$th order accuracy and it does not grow with respect to time either. That is, the results seem to be even better in this variable coefficient case than the constant coefficient case studied in the previous section.

Example 4. We solve the Eq. (3.1) where $a(x)$ and $b(x,t)$ are given by

$$a(x) = \sin(x),$$
$$b(x,t) = (\sin(x) + 1)\cos(x + t) + \cos(x)\sin(x + t).$$

The exact solution to this problem is

$$u(x,t) = \sin(x + t). \tag{3.3}$$

In this example, $a(x)$ is no longer always positive. However, we can still use the upwind flux, namely, if $a(x_{j+\frac{1}{2}}) > 0$, then at $x_{j+\frac{1}{2}}$, we take the flux to be u_h^-; otherwise, we use u_h^+.

The projection P_h is defined as follows. If $a(x_j) > 0$, then on the cell I_j, we use P_h^-; otherwise, we use P_h^+, which is defined as the projection of u into V_h^k such that

$$\int_{I_j} P_h^+ u\ v_h \mathrm{d}x = \int_{I_j} u v_h \mathrm{d}x$$

Table 3.3
The errors $\bar{e}$ and e for Example 4 when using P^1 polynomials on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	1.31E–03	–	4.17E–03	–	2.61E–03	–
	40	2.21E–04	2.56	8.07E–04	2.37	6.01E–04	2.12
	80	3.81E–05	2.54	1.07E–04	2.91	1.11E–04	2.43
	160	6.64E–06	2.52	1.16E–05	3.20	1.64E–05	2.76
e	20	4.25E–03	–	5.66E–03	–	4.86E–03	–
	40	1.03E–03	1.98	1.30E–03	2.13	1.24E–03	1.98
	80	2.61E–04	1.98	2.85E–04	2.18	2.90E–04	2.09
	160	6.56E–05	1.99	6.77E–05	2.07	6.82E–05	2.09

Table 3.4
The errors $\bar{e}$ and e for Example 4 when using P^2 polynomials on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	4.38E–05	–	9.73E–05	–	8.53E–05	–
	40	3.96E–06	3.47	5.99E–06	4.02	9.30E–06	3.20
	80	3.53E–07	3.49	4.11E–07	3.87	5.26E–07	4.14
	160	3.13E–08	3.50	3.93E–08	3.38	4.12E–08	4.00
e	20	1.16E–04	–	1.27E–04	–	1.27E–04	–
	40	1.40E–05	3.05	1.41E–05	3.17	1.54E–05	3.05
	80	1.72E–06	3.03	1.75E–06	3.01	1.76E–06	3.13
	160	2.12E–07	3.02	2.13E–07	3.04	2.16E–07	3.00

for any $v_h \in P^{k-1}$ on I_j and

$$(P_h^+ u)^+ = u^+ \quad \text{at } x_{j-1/2}.$$

We again use the five stage, fourth order SSP Runge–Kutta discretization in time and take $\Delta t = CFL\ h^2$ to reduce time errors. $\bar{e}$ is now defined to be $P_h u - u_h$. We test the example with both P^1 and P^2 polynomials. The results in Tables 3.3 and 3.4 show that $\bar{e}$ achieves at least $(k+\frac{3}{2})$th order superconvergence, and it does not grow with respect to time for the meshes and times that we have used. The error e achieves the expected $(k + 1)$th order accuracy and it does not grow with respect to time either. That is, the results are similar to those from the previous example when the coefficient is positive.

4. Nonlinear equations

In this section, we consider nonlinear equations.

Example 5. We solve the following nonlinear equation

$$\begin{cases} u_t + (u^3)_x = b(x,t) \\ u(x,0) = \sin(x) \\ u(0,t) = u(2\pi,t) \end{cases}, \tag{4.1}$$

where $b(x,t)$ is given by

$$b(x,t) = (1 + 3\sin^2(x+t))\cos(x+t).$$

The exact solution is

$$u(x,t) = \sin(x+t). \tag{4.2}$$

Since $f'(u) = 3u^2 \geqslant 0$, we can still use the upwind fluxes. The projection P_h^- is defined in the same way as before. We use the five stage, fourth order SSP Runge–Kutta discretization in time and take $\Delta t = CFL\ h^2$ to reduce time errors. We test the example with both P^1 and P^2 polynomials. The results in Tables 4.1 and 4.2 show that $\bar{e}$ achieves at least $(k+\frac{3}{2})$th order superconvergence, and it does not grow with respect to time for the meshes and times that we have used. The error e achieves the expected $(k + 1)$th order accuracy and it does not grow with respect to time either. That is, the results are similar to those from the previous section with variable coefficient linear equations.

Table 4.1
The errors $\bar{e}$ and e for Example 5 when using P^1 polynomials on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	7.30E−04	–	6.97E−04	–	6.95E−04	–
	40	1.31E−04	2.48	1.38E−04	2.33	1.30E−04	2.42
	80	2.32E−05	2.49	2.45E−05	2.49	2.40E−05	2.43
	160	4.18E−06	2.47	4.31E−06	2.51	4.29E−06	2.48
e	20	4.27E−03	–	4.25E−03	–	4.25E−03	–
	40	1.06E−03	2.00	1.06E−03	2.00	1.06E−03	2.00
	80	2.66E−04	2.00	2.66E−04	2.00	2.66E−04	2.00
	160	6.64E−05	2.00	6.64E−05	2.00	6.64E−05	2.00

Table 4.2
The errors $\bar{e}$ and e for Example 5 when using P^2 polynomials on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	5.40E−05	–	3.45E−05	–	3.53E−05	–
	40	4.67E−06	3.53	3.02E−06	3.51	3.01E−06	3.55
	80	3.22E−07	3.86	2.57E−07	3.56	2.57E−07	3.55
	160	1.99E−08	4.02	1.91E−08	3.75	1.91E−08	3.75
e	20	1.12E−04	–	1.08E−04	–	1.08E−04	–
	40	1.34E−05	3.07	1.33E−05	3.03	1.33E−05	3.03
	80	1.65E−06	3.02	1.65E−06	3.01	1.65E−06	3.01
	160	2.07E−07	3.00	2.07E−07	2.99	2.07E−07	2.99

Example 6. We solve the following nonlinear Burgers equation

$$\begin{cases} u_t + (u^2)_x = b(x,t) \\ u(x,0) = \sin(x) \\ u(0,t) = u(2\pi,t) \end{cases}, \tag{4.3}$$

where $b(x,t)$ is given by

$$b(x,t) = (1 + 2\sin(x+t))\cos(x+t).$$

The exact solution is

$$u(x,t) = \sin(x+t). \tag{4.4}$$

In this example, $f(u)$ is no longer always positive. We choose to use the Godunov flux, which is an upwind flux. The projection P_h is defined as follows. If $u(x_j,t)$ is positive, then on the cell I_j, we use P_h^-; otherwise, we use P_h^+.

We again use the five stage, fourth order SSP Runge–Kutta discretization in time and take $\Delta t = CFLh^2 .\bar{e}$ is now defined to be $P_h u - u_h$. We test this example using both P^1 and P^2 polynomials. From Tables 4.3 and 4.4, we can see that $\bar{e}$ has at least $(k + \frac{3}{2})$th order superconvergence, and it does not grow with respect to time for most meshes. The error e achieves the expected $(k + 1)$th order accuracy and it does not grow with respect to time either. That is, the results are similar to those from the previous example when the wind direction does not change.

For Example 6, if we use the Lax-Friedrichs flux instead of the Godunov flux, then $\bar{e}$ does not achieve superconvergence, as can be seen from Table 4.5.

Table 4.3
The errors $\bar{e}$ and e for Example 6 when using P^1 polynomials on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	7.74E−04	–	1.85E−03	–	1.82E−03	–
	40	1.11E−04	2.80	2.91E−04	2.66	2.99E−04	2.60
	80	1.46E−05	2.93	4.81E−05	2.60	4.81E−05	2.64
	160	1.93E−06	2.92	7.76E−06	2.63	7.76E−06	2.62
e	20	4.29E−03	–	4.61E−03	–	4.61E−03	–
	40	1.07E−03	2.01	1.10E−03	2.00	1.10E−03	2.06
	80	2.66E−04	2.00	2.70E−04	2.03	2.70E−04	2.03
	160	6.64E−05	2.00	6.68E−05	2.01	6.68E−05	2.00

Table 4.4
The errors $\bar{e}$ and e for Example 6 when using P^2 polynomials on a uniform mesh of N cells ($CFL = 0.5$)

	N	$T = 1$		$T = 100$		$T = 500$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	7.18E−05	–	8.97E−05	–	1.14E−04	–
	40	7.56E−06	3.53	9.58E−06	3.23	9.03E−06	3.66
	80	9.19E−07	3.04	8.84E−07	3.44	8.92E−07	3.34
	160	7.76E−08	3.57	7.71E−08	3.51	7.84E−08	3.51
e	20	1.23E−04	–	1.37E−04	–	1.36E−04	–
	40	1.48E−05	3.05	1.53E−05	3.16	1.54E−05	3.14
	80	1.79E−06	3.05	1.81E−06	3.08	1.81E−06	3.09
	160	2.16E−07	3.05	2.16E−07	3.00	2.16E−07	3.07

Table 4.5
Example 6 when using the Lax-Friedrichs flux and P^1 polynomials on a uniform mesh of N cells ($T = 1$, $CFL = 0.5$)

N	$\bar{e}$		e	
	L^2 error	Order	L^2 error	Order
20	1.17E−03	–	4.00E−03	–
40	2.29E−04	2.35	9.84E−04	2.02
80	5.08E−05	2.17	2.44E−04	2.01
160	1.22E−05	2.06	6.10E−05	2.00

5. One-dimensional systems

In this section, we generalize our discussion from scalar equations to one-dimensional hyperbolic systems. We consider only linear system with constant coefficients.

Example 7. We solve the following one-dimensional system

$$\begin{cases} \begin{pmatrix} u \\ v \end{pmatrix}_t + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}_x = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ u(x,0) = \sin(x) \\ v(x,0) = \cos(x) \\ u(0,t) = u(2\pi,t) \\ v(0,t) = v(2\pi,t) \end{cases} . \tag{5.1}$$

The exact solution to this system is

$$\begin{pmatrix} u \\ v \end{pmatrix}(x,t) = \frac{1}{2}\begin{pmatrix} \sin(x-t) + \cos(x-t) + \sin(x+t) - \cos(x+t) \\ \sin(x-t) + \cos(x-t) - \sin(x+t) + \cos(x+t) \end{pmatrix}. \tag{5.2}$$

The DG scheme for this problem is formulated as: find $u_h, v_h \in V_h^k$, such that

$$\begin{cases} \int_{I_j} (u_h)_t z_h \mathrm{d}x - \int_{I_j} v_h (z_h)_x \mathrm{d}x + h_1(u_h^-, u_h^+, v_h^-, v_h^+) z_h^- |_{j+\frac{1}{2}} - h_1(u_h^-, u_h^+, v_h^-, v_h^+) z_h^+ |_{j-\frac{1}{2}} = 0 \\ \int_{I_j} (v_h)_t q_h \mathrm{d}x - \int_{I_j} u_h (q_h)_x \mathrm{d}x + h_2(u_h^-, u_h^+, v_h^-, v_h^+) q_h^- |_{j+\frac{1}{2}} - h_2(u_h^-, u_h^+, v_h^-, v_h^+) q_h^+ |_{j-\frac{1}{2}} = 0 \end{cases} \tag{5.3}$$

holds for any $z_h, q_h \in V_h^k$.

In (5.3), the numerical fluxes are taken as the upwind fluxes

$$\begin{pmatrix} h_1(u_h^-, u_h^+, v_h^-, v_h^+) \\ h_2(u_h^-, u_h^+, v_h^-, v_h^+) \end{pmatrix}(x,t) = \frac{1}{2}\left(\begin{pmatrix} v_h^- + v_h^+ \\ u_h^- + u_h^+ \end{pmatrix} - \begin{pmatrix} u_h^+ - u_h^- \\ v_h^+ + v_h^- \end{pmatrix} \right). \tag{5.4}$$

For this problem, the projection P_h of the exact solution is defined as:

$$\begin{pmatrix} P_h u \\ P_h v \end{pmatrix}(x,t) = \frac{1}{2}\begin{pmatrix} P_h^- \sin(x-t) + P_h^- \cos(x-t) + P_h^+ \sin(x+t) - P_h^+ \cos(x+t) \\ P_h^- \sin(x-t) + P_h^- \cos(x-t) - P_h^+ \sin(x+t) + P_h^+ \cos(x+t) \end{pmatrix}. \tag{5.5}$$

This projection is obtained by diagonalizing the system into

$$\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}_t + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}_x = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \tag{5.6}$$

where

$$\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}. \tag{5.7}$$

We can apply P_h^- and P_h^+ on w_1 and w_2, and then define

$$\begin{pmatrix} P_h u \\ P_h v \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} P_h^- w_1 \\ P_h^+ w_2 \end{pmatrix}. \tag{5.8}$$

Table 5.1
The errors $\bar{e}$ and e for Example 7 when using P^1 polynomials on a uniform mesh of N cells (CFL = 0.5)

	N	T = 1		T = 10		T = 100	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	4.60E−04	–	3.04E−03	–	2.96E−02	–
	40	5.80E−05	2.99	3.82E−04	2.99	3.79E−03	2.97
	80	7.26E−06	3.00	4.79E−05	3.00	4.75E−04	2.99
	160	9.08E−07	3.00	5.99E−06	3.00	5.95E−05	3.00
e	20	4.21E−03	–	5.16E−03	–	2.99E−02	–
	40	1.06E−03	1.99	1.12E−03	2.20	3.93E−03	2.92
	80	2.65E−04	2.00	2.69E−04	2.06	5.44E−04	2.85
	160	6.64E−05	2.00	6.66E−05	2.02	8.91E−05	2.61

Table 5.2
The errors $\bar{e}$ and e for Example 7 when using P^2 polynomials on a uniform mesh of N cells ($CFL = 0.1$)

	N	$T = 1$		$T = 10$		$T = 100$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	20	4.17E–06	–	5.13E–06	–	3.02E–05	–
	40	2.62E–07	3.99	2.78E–07	4.21	9.74E–07	4.95
	80	1.64E–08	4.00	1.66E–08	4.06	3.36E–08	4.86
	160	1.02E–09	4.00	1.03E–09	4.02	1.37E–09	4.61
e	20	1.07E–04	–	1.07E–04	–	1.11E–04	–
	40	1.34E–05	3.00	1.34E–05	3.00	1.34E–05	3.05
	80	1.67E–06	3.00	1.67E–06	3.00	1.67E–06	3.00
	160	2.09E–07	3.00	2.09E–07	3.00	2.09E–07	3.00

We use the SSP ninth-order discretization in time and take $\Delta t = CFL\ h$. We run numerical tests with both P^1 and P^2 polynomials. From Tables 5.1 and 5.2, we observe that $\bar{e}$ achieves $(k + 2)$th order superconvergence, and it grows at most linearly with respect to time. Meanwhile, the error e is of the expected $(k + 1)$th order accuracy, and it does not grow much with respect to the time t (for the time that we have considered) for fine grids. That is, the results are similar to those for the scalar PDEs.

6. Two-dimensional equations

In this section, we generalize our discussion to two-dimensional equations. We will only consider scalar linear equations.

Example 8. We solve the following equation

$$\begin{cases} u_t + u_x + u_y = 0 \\ u(x,y,0) = \sin(x+y) \end{cases}. \tag{6.1}$$

Periodic boundary conditions are imposed on the boundary of the domain $[0,2\pi]^2$. The exact solution is

$$u(x,t) = \sin(x+y-2t). \tag{6.2}$$

We use a rectangular mesh defined as

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N_x+\frac{1}{2}} = 2\pi, \quad 0 = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \cdots < y_{N_y+\frac{1}{2}} = 2\pi \tag{6.3}$$

and

$$I_{i,j} = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \times \left[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}\right]$$

We define the approximation space as

$$V_h^k = \{v : v|_{I_{i,j}} \in Q^k(I_{i,j}), \quad i = 1, \ldots, N_x, \quad j = 1, \ldots, N_y\} \tag{6.4}$$

where $Q^k(I_{i,j})$ denotes all the polynomials of degree at most k in x and in y on $I_{i,j}$.

The projection P_h is defined as follows:

$$\int_{I_{i,j}} P_h u\ v_h \mathrm{d}x\mathrm{d}y = \int_{I_{i,j}} u v_h \mathrm{d}x\mathrm{d}y$$

for any $v_h \in V_h^{k-1}$,

Table 6.1
The errors $\bar{e}$ and e for Example 8 when using V_h^1 space on a uniform mesh of $N \times N$ cells ($CFL = 0.5$)

	N	$T = 1$		$T = 10$		$T = 100$	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	5	4.68E–02	–	2.81E–01	–	7.01E–01	–
	10	6.64E–03	2.82	4.60E–02	2.61	3.46E–01	1.02
	20	8.63E–04	2.94	6.04E–03	2.93	5.79E–02	2.58
	40	1.09E–04	2.98	7.63E–04	2.99	7.56E–03	2.94
e	5	8.80E–02	–	2.86E–01	–	7.06E–01	–
	10	2.34E–02	1.91	5.06E–02	2.50	3.47E–01	1.03
	20	5.97E–03	1.97	8.42E–03	2.59	5.82E–02	2.57
	40	1.50E–03	1.99	1.68E–03	2.33	7.70E–03	2.92

Table 6.2
The errors $\bar{e}$ and e for Example 8 when using V_h^1 space on a random mesh of $N \times N$ cells ($CFL = 0.5$)

	N	T = 1		T = 10		T = 100	
		L^2 error	Order	L^2 error	Order	L^2 error	Order
$\bar{e}$	5	4.80E−02	–	2.86E−01	–	7.01E−01	–
	10	7.03E−03	2.77	4.73E−02	2.60	3.54E−01	0.99
	20	9.47E−04	2.89	6.32E−03	2.91	6.05E−02	2.55
	40	1.23E−04	2.94	7.97E−04	2.99	7.89E−03	2.94
e	5	9.32E−02	–	2.92E−01	–	7.06E−01	–
	10	2.76E−02	1.76	5.36E−02	2.44	3.54E−01	1.00
	20	9.98E−03	1.47	1.17E−02	2.19	6.12E−02	2.53
	40	4.17E−03	1.26	4.24E−03	1.47	8.91E−03	2.78

$$\int_{y_{j-1/2}}^{y_{j+1/2}} P_h u(x_{i+1/2}, y) w_h(y) \mathrm{d}y = \int_{y_{j-1/2}}^{y_{j+1/2}} u(x_{i+1/2}, y) w_h(y) \mathrm{d}y$$

for any $w_h \in P^k$ on $[y_{j-1/2}, y_{j+1/2}]$, and

$$\int_{x_{i-1/2}}^{x_{i+1/2}} P_h u(x, y_{j+1/2}) z_h(x) \mathrm{d}x = \int_{x_{i-1/2}}^{x_{i+1/2}} u(x, y_{j+1/2}) z_h(x) \mathrm{d}x$$

for any $z_h \in P^k$ on $[x_{i-1/2}, x_{i+1/2}]$.

We use the upwind fluxes in both spatial directions and SSP Runge–Kutta ninth-order discretization in time to reduce time errors. We test the example using V_h^1 space with both uniform and random meshes. The results in Tables 6.1 and 6.2 show that $\bar{e}$ achieves third order superconvergence and it grows at most linearly with respect to the time t, just as in the one-dimensional case.

7. Concluding remarks

We have studied the behavior of the error between the DG solution and the exact solution for smooth solutions of conservation laws when upwind fluxes are used. We prove that if we apply piecewise linear polynomials to a linear constant coefficient scalar equation, the DG solution will be superconvergent towards a particular projection of the exact solution. The error between the DG solution and the exact solution is thus decomposed into two parts, the superconvergent part which grows at most linearly in time, and another part which does not grow in time. Therefore, the error of the DG scheme will not grow for fine grids over a long time period, at least proportional to $\mathrm{O}(\frac{1}{\sqrt{h}})$ according to the proof for the P^1 case and at least to $\mathrm{O}(\frac{1}{h})$ according to the numerical results. We give numerical tests of P^k polynomials, with $1 \leqslant k \leqslant 3$, to demonstrate the superconvergence property for linear equations with variable coefficients, nonlinear equations, systems and two-dimensional equations, to demonstrate the generality of the conclusions. The most important implication of the discussion in this paper is that the DG solution does not grow with time until the time reaches $\mathrm{O}(\frac{1}{h})$ where h is the mesh size, and this further justifies the advantage of choosing DG methods for long time simulation of linear and nonlinear wave equations.

We remark that we observe high order superconvergence for L^1, L^2 and L^∞ measures of the error, even though only error tables for the L^2 norm are provided to save space. A direct conclusion is that the DG solution will be superconvergent at the outflow boundary point. This has been proved by Adjerid et al. in [1,2].

In our future work we will attempt to prove the superconvergence property for general case of P^k with $k \geqslant 1$. We will also generalize the results to other types of PDEs, for example the convection diffusion equations.

Acknowledgment

Research supported by NSF Grant DMS-0510345.

References

[1] S. Adjerid, K. Devine, J. Flaherty, L. Krivodonova, A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems, Computational Methods in Applied Mechanics and Engineering 191 (2002) 1097–1112.

[2] S. Adjerid, T. Massey, Superconvergence of discontinuous Galerkin solutions for a nonlinear scalar hyperbolic problem, Computational Methods in Applied Mechanics and Engineering 195 (2006) 3331–3346.

[3] B. Cockburn, S. Hou, C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Mathematics of Computation 54 (1990) 545–581.

[4] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, Journal of Computational Physics 84 (1989) 90–113.

[5] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Mathematics of Computation 52 (1989) 411–435.

[6] B. Cockburn, C.-W. Shu, The Runge-Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws, Mathematical Modelling and Numerical Analysis 25 (1991) 337–361.

[7] B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, Journal of Computational Physics 141 (1998) 199–224.

[8] B. Cockburn, C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, Journal of Scientific Computing 16 (2001) 173–261.

[9] B. Cockburn, C.-W. Shu, Foreword for the special issue on discontinuous Galerkin method, Journal of Scientific Computing 22–23 (2005) 1–3.

[10] C. Dawson, Foreword for the special issue on discontinuous Galerkin method, Computer Methods in Applied Mechanics and Engineering 195 (2006) 3183.

[11] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability preserving high order time discretization methods, SIAM Review 43 (2001) 89–112.

[12] M. Zhang, C.-W. Shu, An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods, Computers and Fluids 34 (2005) 581–592.

[13] Q. Zhang, C.-W. Shu, Error estimates to smooth solutions of Runge-Kutta discontinuous Galerkin methods for scalar conservation laws, SIAM Journal on Numerical Analysis 42 (2004) 641–666.