

Numerical Diffusion Control of a Space-Time Discontinuous Galerkin Method

Michel Borrel* and Juliette Ryan

ONERA, BP72 – 29 Avenue de la Division Leclerc, 92322, Chatillon Cedex, France.

Received 2 October 2007; Accepted (in revised version) 18 April 2008

Available online 1 August 2008

Abstract. Variations on space-time Discontinuous Galerkin (STDG) discretization associated to Runge-Kutta schemes are developed. These new schemes while keeping the original scheme order can improve accuracy and stability. Numerical analysis is made on academic test cases and efficiency of these schemes are shown on propagating pressure waves.

AMS subject classifications: 52B10, 65D18, 68U05, 68U07

Key words: Gas dynamics, aeroacoustics, discontinuous Galerkin discretization, Runge-Kutta schemes, numerical diffusion control.

1 Introduction

Controlling the numerical diffusion with an upwinding technique is not a new idea, but this task is rather difficult when solving the nonlinear equations of gas dynamics for compressible flows with a DG approach [2–5]. The numerical flux in space (for example the Roe flux or the Lax-Friedrichs flux) are fixed once and for all and there are no parameters to be tuned except those of the Runge-Kutta time scheme.

What we propose here in this paper is to devise variants of the DG approach. The space-time DG approach (STDG) [7–9] leads to naturally implicit schemes solved iteratively. Our first variant is to use a truncated explicit process to replace the iterative implicit solver (STDG-RK). The second variant STDG- α consists in an adapted upwinding in time of the STDG scheme when computing the time fluxes. This scheme can improve convergence for steady flows as a higher CFL can be used in the pseudo-time solver. A third variant (RKDG-NDC) consists in upwinding the Runge-Kutta space DG approach, leading to a unified formulation with the iterative implicit STDG scheme. These schemes allow control of numerical diffusion.

*Corresponding author. *Email addresses:* borrel@onera.fr (M. Borrel), ryan@onera.fr (J. Ryan)

A numerical study of precision and stability is presented for the 1D linear advection equation and the Burgers equation both for steady and unsteady problems.

DG is a nonlinear scheme. To preserve monotonicity, limiters should be used when solving nonlinear problems. But in many cases (nonlinear aeroacoustics, subsonic, transonic and vortical flows for example), they are not necessary. In this paper, all computations are done without limiters. MUSCL results are shown to provide reference results as the classical DG can be seen as an extension of the MUSCL approach [1].

Results are shown on the academic test case of a planar wave propagating upstream a subsonic flow. The RKDG-NDC result is compared to a computation without numerical diffusion control. As will be seen, the RKDG-NDC shows very little diffusion without loss of accuracy or stability.

Another result concerns the study of a planar acoustic wave interacting with a circular temperature spot on a Cartesian mesh. This study is connected to high frequency combustion instability [11] This aeroacoustic application is displayed associated with an AMR technique [12–14].

These formulations are easily extended to curvilinear grid as shown in the case of a transonic flow around the NACA0012 airfoil.

2 Numerical discretization

Following the works of many authors (see for example Cockburn [6] or van der Veegt [7–9]), DG is now standard for solving conservation law equations. This paper is a sequel of [3] in which different classical space DG variants were compared to MUSCL. We look here at a space-time DG formulation which gives better results than previously for nearly the same computational effort. In our implementation of RKDG (Space) or STDG (Space-Time) formulations in our Euler solver, only a $P1$ approximation has been used (but $P1$ and $P2$ accuracy tests have been conducted on a 1D scalar equation). The Euler implementation has been realized within an AMR platform [12].

2.1 Governing equations

We consider the Euler equations written in the compact conservative form:

$$\partial_t W + \vec{\nabla} \cdot \mathbf{F}(W) = \mathbf{0}, \quad (2.1)$$

where W is the conservation variable vector:

$$W = (\rho, \rho \vec{U}, \rho E), \quad (2.2)$$

and $\mathbf{F} = (f, g, h)^T$ the flux vector:

$$\begin{cases} f = (\rho u, \rho u^2 + p, \rho uv, \rho uw, u(\rho E + p)), \\ g = (\rho v, \rho uv, \rho v^2 + p, \rho vw, v(\rho E + p)), \\ h = (\rho w, \rho uw, \rho vw, \rho w^2 + p, w(\rho E + p)), \end{cases} \quad (2.3)$$

ρ is the density, \vec{U} the velocity, E the total energy and p the pressure given by the state law for perfect gas (the specific heat ratio γ is supposed to be constant). These equations can be re-written more conveniently for the space-time formulation into the full divergence form:

$$\nabla_t \cdot \mathcal{F}(W) = \mathbf{0}, \tag{2.4}$$

with $\nabla_t = (\partial_t, \partial_{x^1}, \partial_{x^2}, \partial_{x^3})$ and $\mathcal{F} = (W, \mathbf{F}(W))^T$.

2.2 Space-time discontinuous Galerkin formulation

At each point $(x,t) \in \Omega \times [0,T]$, an approximate solution $W_h(x,t)$ is searched from an initial solution $W(x,t=0)$ and boundary conditions. The 3D computational domain is partitioned using a structured mesh:

$$\Omega = \bigcup_{i=1}^N \Omega_i, \tag{2.5}$$

and the centroid of each cell Ω_i is noted $x_{i+1/2}$. In each cell, the solution is expanded on a local basis of polynomials of degree at most 1:

$$\forall x \in \Omega, \quad \forall t \in [0,T], \quad W_h(x,t) = \sum_{i,n} \left(\sum_{\ell=0,4} p_{i,n}^\ell(x,t) \bar{W}_i^{\ell,n} \right), \tag{2.6}$$

where $\{p_{i,n}^\ell, \ell=0,4\}$ is the local basis of Legendre polynomials:

$$\begin{cases} p_{i,n}^0(x,t) = \begin{cases} 1 & \text{if } (x,t) \in \Omega_i \times [t^n, t^{n+1}], \\ 0 & \text{elsewhere,} \end{cases} \\ p_{i,n}^\ell(x,t) = (x^\ell - x_{i+1/2}^\ell) p_{i,n}^0(x,t) \quad (\ell=1,2,3), \\ p_{i,n}^4(x,t) = (t - t^{n+1/2}) p_{i,n}^0(x,t), \end{cases} \tag{2.7}$$

\bar{W}_i^ℓ are the degrees of freedom on the space-time slab $D_i^n = \Omega_i \times [t^n, t^{n+1}]$ which approximate the mean and the gradient cell values.

In the following, all integrations are computed in the physical space, as in Allmaras's thesis [15]. Thus, element mappings are avoided, as in finite volume methods. The discretization is built from the weak formulation of (2.4) obtained by multiplying each equation with a test function φ_h in the approximation space V_h generated by our local basis and by integrating by parts on $\Omega_i \times [t^n, t^{n+1}]$:

$$\oint_{\partial D_i^n} \varphi_h \tilde{\mathcal{F}}(W_h) \cdot \vec{n}_t \, d\sigma - \int_{D_i^n} \vec{\nabla} \varphi_h \cdot \mathcal{F}(W_h) \, d\Omega dt = 0 \tag{2.8}$$

with n_t the outward normal to D_i^n and

$$V_h = \{ \varphi_h \in L^2(\Omega \times [0,T]) \mid \forall (i,n), \varphi_h|_{D_i^n} = \varphi_{i,n} \in P^1(D_i^n) \}, \tag{2.9}$$

where $P^1(D_i^n)$ represents the polynomials of degree at most 1 on D_i^n . In (2.8), $\tilde{\mathcal{F}}(W_h)$ represents the boundary values of \mathcal{F} . From the relation:

$$\partial D_i^n = \Omega_i \times \{t^n\} \cup \Omega_i \times \{t^{n+1}\} \cup \partial\Omega_i \times [t^n, t^{n+1}] \tag{2.10}$$

and our choice of Legendre polynomials, we can now introduce our first parameter α .

If we use an upwind discretization in time with $\Delta t = [t^n, t^{n+1}]$ and ($\alpha \geq 0$):

$$\begin{cases} \oint_{\Omega_i \times \{t^{n+1}\}} \varphi_h W_h \cdot \vec{\mathbf{n}}_t d\sigma \equiv \int_{\Omega_i} \varphi_h(t^{n+1}) W_h(t^{n+1/2} + \alpha\Delta t/2) d\Omega, \\ \oint_{\Omega_i \times \{t^n\}} \varphi_h W_h \cdot \vec{\mathbf{n}}_t d\sigma \equiv - \int_{\Omega_i} \varphi_h(t^n) W_h(t^{n-1/2} + \alpha\Delta t/2) d\Omega, \end{cases} \tag{2.11}$$

then, the weak formulation leads to the system:

$$\begin{cases} \bar{W}_i^{0,n} = \bar{W}_i^{0,n-1} + \alpha \frac{\Delta t}{2} \bar{W}_i^{4,n-1} + \frac{\Delta t}{2} \mathbf{R}_i^{0,n} - \frac{\Delta t^2}{12} \mathbf{R}_i^{4,n}, \\ \bar{W}_i^{l,n} = \bar{W}_i^{l,n-1} + \Delta t [\mathcal{M}_i^{-1}(\mathbf{R}_i^{m,n})_{m=1,3}]_\ell \quad (\ell = 1,2,3), \\ \bar{W}_i^{4,n} = (\mathbf{R}_i^{0,n} + \frac{\Delta t}{6} \mathbf{R}_i^{4,n}) / \alpha, \end{cases} \tag{2.12}$$

with the following notations for the residuals:

$$\mathbf{R}_i^{\ell,n} = \frac{1}{a(\ell)} \left(- \int_{t^n}^{t^{n+1}} dt \oint_{\partial\Omega_i} p_i^\ell \tilde{\mathbf{F}} \cdot \vec{\mathbf{n}}_x d\sigma_x + \int_{t^n}^{t^{n+1}} dt \int_{\Omega_i} \mathbf{F} \cdot \vec{\nabla} p_i^\ell d\Omega \right), \tag{2.13}$$

where

$$a(\ell) = \int_{t^n}^{t^{n+1}} dt \int_{\Omega_i} p_i^\ell p_i^\ell d\Omega \tag{2.14}$$

and

$$\mathcal{M}_i = \left(\frac{1}{a(\ell)} \int_{\Omega_i} p_i^\ell p_i^m d\Omega \right)_{\ell,m} \tag{2.15}$$

represents the local mass matrix which is diagonal with Cartesian grids. \mathbf{n}_x is the outward normal to Ω_i . Both \mathbf{n}_x and Ω_i (thus $d\sigma_x$) are time independent. In these expressions, $\tilde{\mathbf{F}}$ represents the interface numerical flux evaluated using a 4 point Gauss quadrature formula (in 3D) associated with a classical flux formula (Roe, LLF, ...). The numerical implicit scheme can be re-formulated as:

$$\begin{cases} \bar{W}_i^{0,n} = \bar{W}_i^{0,n-1} + \frac{\Delta t}{2} (\mathbf{R}_i^{0,n} + \mathbf{R}_i^{0,n-1}) - \frac{\Delta t^2}{12} (\mathbf{R}_i^{4,n} - \mathbf{R}_i^{4,n-1}), \\ \bar{W}_i^{l,n} = \bar{W}_i^{l,n-1} + \Delta t [\mathcal{M}_i^{-1}(\mathbf{R}_i^{m,n})_{m=1,3}]_\ell \quad (\ell = 1,2,3), \\ \bar{W}_i^{4,n} = (\mathbf{R}_i^{0,n} + \frac{\Delta t}{6} \mathbf{R}_i^{4,n}) / \alpha. \end{cases} \tag{2.16}$$

One can see that introducing the α term works as a relaxation factor on the time derivative.

The expression (2.13) requires the computation of time integrals. If we use a two-point Gauss formula to evaluate these integrals, then $\mathbf{R}_i^{4,n}$ can be seen as a time gradient of residuals and thus the term $(\mathbf{R}_i^{4,n} - \mathbf{R}_i^{4,n-1})$ in (2.16) works as a dissipative term.

If one uses a one-point integration formula to evaluate the time integrals, $\mathbf{R}_i^{4,n} = 0$ and writing the one point approximation of $\mathbf{R}_i^{l,n}$ as $\mathbf{R}_i^{l,n-1/2}$, (2.16) then becomes

$$\begin{cases} \bar{W}_i^{0,n} = \bar{W}_i^{0,n-1} + \frac{\Delta t}{2} (\mathbf{R}_i^{0,n-1/2} + \mathbf{R}_i^{0,n-3/2}), \\ \bar{W}_i^{l,n} = \bar{W}_i^{l,n-1} + \Delta t [\mathcal{M}_i^{-1}(\mathbf{R}_i^{m,n-1/2})_{m=1,3}]_\ell \\ \bar{W}_i^{4,n} = \mathbf{R}_i^{0,n-1/2} / \alpha. \end{cases} \quad (\ell = 1, 2, 3), \quad (2.17)$$

This scheme can now be compared to that of a RK2 time discretization applied to a DG discretization in space with

$$\mathbf{R}_i^{\ell,n-1/2} = -\frac{1}{b(\ell)} \oint_{\partial\Omega_i} p_i^\ell \mathbf{F}(\mathbf{W}^{n-1/2}) \cdot \vec{\mathbf{n}} \, d\sigma,$$

where a RK step is written as:

$$\begin{cases} \bar{W}_i^{0,n} = \bar{W}_i^{0,n-1} + \Delta t \mathbf{R}_i^{n-1/2}, \\ \bar{W}_i^{l,n} = \bar{W}_i^{l,n-1} + \Delta t [\mathcal{N}_i^{-1}(\mathbf{R}_i^{m,n-1/2})_{m=1,3}]_\ell \end{cases} \quad (\ell = 1, 2, 3), \quad (2.18)$$

with

$$b(\ell) = \int_{\Omega_i} p_i^\ell p_i^\ell \, d\Omega, \quad \mathcal{N} = \left(\frac{1}{b(\ell)} \int_{\Omega_i} p_i^\ell p_i^m \, d\Omega \right)_{\ell,m}.$$

Mixing these two formulations (STDG and RKDG), it is possible to devise a new class of explicit schemes, depending on one parameter β and labeled RKDG-NDC in the following:

$$\begin{cases} \bar{W}_i^{0,n} = \bar{W}_i^{0,n-1} + \Delta t (\beta \mathbf{R}_i^{0,n-1/2} + (1-\beta) \mathbf{R}_i^{0,n-3/2}), \\ \bar{W}_i^{l,n} = \bar{W}_i^{l,n-1} + \Delta t [\mathcal{M}_i^{-1}(\mathbf{R}_i^{m,n-1/2})_{m=1,3}]_\ell \\ \bar{W}_i^{4,n} = \mathbf{R}_i^{0,n-1/2} / \alpha. \end{cases} \quad (\ell = 1, 2, 3), \quad (2.19)$$

The parameter β has been introduced in order to obtain a unified formulation for the implicit STDG scheme (2.16) (with $\beta = 1/2$) and the explicit RKDG scheme (2.18) (with $\beta = 1$). For $0.5 \leq \beta \leq 1$, this variant can be considered as a variant of the RKDG approach with an extra storage for $\bar{W}_i^{4,n}$ but requiring the same time stepping; the parameter β can be viewed as a control of the numerical dissipation.

Formulation (2.19) can be written as

$$\mathcal{W}^n = \mathcal{W}^{n-1} + \mathcal{A}(\mathcal{W}^n, \mathcal{W}^{n-1}). \quad (2.20)$$

2.3 Time stepping methods

The STDG formulation (2.16) is unconditionally stable with respect to the physical time step Δt but leads to an implicit system

$$\mathcal{L}_1(\mathcal{W}^n, \mathcal{W}^{n-1}) = 0, \quad (2.21)$$

where

$$\mathcal{L}_1(\mathcal{W}^n, \mathcal{W}^{n-1}) = \mathcal{W}^n - \mathcal{W}^{n-1} - \mathcal{A}(\mathcal{W}^n, \mathcal{W}^{n-1})$$

is solved iteratively with a pseudo-CFL constraint (cf. [10]), using the following system:

$$\frac{\partial \overline{\mathcal{W}}}{\partial t} = -\frac{1}{\Delta t} \mathcal{L}_1(\overline{\mathcal{W}}, \mathcal{W}^{n-1}). \quad (2.22)$$

The explicit STDG-RKp formulations use the following time discretization:

$$\frac{d\overline{\mathcal{W}}}{dt} = \mathcal{L}_2(\overline{\mathcal{W}}, \mathcal{W}^{n-1}), \quad (2.23)$$

where

$$\mathcal{L}_2(\mathcal{W}) = \frac{1}{\Delta t} \mathcal{A}(\mathcal{W}, \mathcal{W}^{n-1})$$

is realized by means of an explicit p step Runge-Kutta method (RKp) requiring a CFL criterion on the physical time step. This variant needs an another computation of $\mathcal{A}(\mathcal{W}, \mathcal{W}^{n-1})$ so that all residues are consistent in time.

In either case, the algorithm is based on the following steps:

$$\begin{cases} \mathcal{W}_0 = \mathcal{W}^n, \\ \mathcal{W}_1 = \mathcal{W}_0 + C_1 \Delta t \mathcal{L}(\mathcal{W}_0), \\ \dots \\ \mathcal{W}_p = \mathcal{W}_0 + C_p \Delta t \mathcal{L}(\mathcal{W}_{p-1}), \\ \mathcal{W}^{n+1} = \mathcal{W}_p, \end{cases} \quad (2.24)$$

where Δt is the time step calculated by means of a CFL type criterion and $\{C_1, C_2, \dots, C_p\}$ are constants ranging between 0 and 1 classified by ascending order ($C_p = 1$ for consistency). The advantage of this method of integration, in addition to being easily implementable, is that it requires identical memory storage for any value of p .

Computing $\mathcal{A}(\mathcal{W}, \mathcal{W}^{n-1})$ is the most time consuming when dealing with 3D unsteady computations. Comparisons will be made with the most precise scheme that is STDG-RK5q where, at each time step (2.22) will be solved iteratively using q iterations of the 5-step Runge-Kutta scheme presented in [10], with coefficients $(C_1, \dots, C_5) = (.0791451, .1635551, .283663, .5, 1)$.

3 1D scalar problems: Numerical study

In this study, P1 and P2 variants are analyzed for unsteady 1D problems ($\alpha = 1$).

3.1 P2 equations

For 1D STDG-P2, the local basis is extended with the following functions:

$$\begin{cases} p_{i,n}^5(x,t) = \left(\frac{(x-x_{i+1/2})^2}{2} - \frac{\Delta x^2}{24} \right) p_{i,n}^0(x,t), \\ p_{i,n}^6(x,t) = (x-x_{i+1/2})(t-t^{n+1/2}) p_{i,n}^0(x,t), \\ p_{i,n}^7(x,t) = \left(\frac{(t-t^{n+1/2})^2}{2} - \frac{\Delta t^2}{24} \right) p_{i,n}^0(x,t), \end{cases} \quad (3.1)$$

and the formulation becomes

$$\begin{cases} \bar{W}_i^{0,n} = \bar{W}_i^{0,n-1} + \frac{\Delta t}{2} [\beta \mathbf{R}_i^{0,n} + (1-\beta) \mathbf{R}_i^{0,n-1}] - \frac{\Delta t^2}{12} (\mathbf{R}_i^{4,n} - \mathbf{R}_i^{4,n-1}), \\ \bar{W}_i^{1,n} = \bar{W}_i^{1,n-1} + \frac{\Delta t}{2} [\beta \mathbf{R}_i^{1,n} + (1-\beta) \mathbf{R}_i^{1,n-1}] - \frac{\Delta t^2}{12} (\mathbf{R}_i^{6,n} - \mathbf{R}_i^{6,n-1}), \\ \bar{W}_i^{5,n} = \bar{W}_i^{5,n-1} + \Delta t \mathbf{R}_i^{5,n}, \\ \bar{W}_i^{4,n} = (\mathbf{R}_i^{0,n} - \frac{\Delta t}{30} \mathbf{R}_i^{7,n}) / \alpha, \\ \bar{W}_i^{6,n} = (\mathbf{R}_i^{1,n} + \frac{\Delta t}{12} \mathbf{R}_i^{6,n}) / \alpha, \\ \bar{W}_i^{7,n} = (\mathbf{R}_i^{4,n} + \frac{\Delta t}{10} \mathbf{R}_i^{7,n}) / \alpha. \end{cases} \quad (3.2)$$

3.2 Accuracy analysis

First, we look at the two algorithms (2.22), (2.23) to solve the nonlinear implicit system (2.21) on the scalar 1D equation.

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = g \text{ in } \Omega \times [0, T], \\ u(x, 0) = u_0(x), \end{cases} \quad (3.3)$$

where $F(u) = u$ in the linear case and $F(u) = u^2/2$ in the non linear Burgers case. $\Omega = [0, 1]$ and g is a right hand side such that $u_{ex} = \sin 6\pi(x-t)$ is solution to these two problems ($g = 0$ in the linear case and $g(x,t) = 6\pi(-1 + \cos(6\pi(x-t)))$, $u_0(x) = \sin(6\pi x)$). Dirichlet boundary conditions are imposed at boundaries with incoming velocity and extrapolated values at boundaries with outgoing velocity. Schemes are compared at $T = .5$.

The first scheme (STDG-RK5q) has no CFL restriction at the cost of increasing q the number of inner iterations. The second scheme (STDG-RK2) has a CFL restriction of .2. Costs are measured in number of work units (WU), a work unit being the computation of $\mathcal{A}(\bar{W}, \mathcal{W}^{n-1})$. In Table 1, is shown the cost for the two schemes for 3 different numbers of discretization points N_x for a similar relative L2 norm of the error $E2 = \|u - u_{ex}\|_{L2} / \|u_{ex}\|_{L2}$ at CFL=.2 and at $T = .5$. For STDG-RK5q, this is obtained by adjusting the value of q .

Table 1: Cost comparison at CFL=.2 for the two STDG-P1 variants. Left: linear case; Right: Burgers case.

Nx	E2	RK2	RK5q	Nx	E2	RK2	RK5q
15	1.9 e-001	114	760	15	1.0e-001	114	950
30	3.5 e-002	225	1500	30	2.5e-002	225	1875
60	6.3e-003	450	3750	60	7.7e-003	450	3750

Table 2: Cost comparison at CFL=.2 for the two STDG-P2 variants. Left: linear case; Right: Burgers case.

Nx	E2	RK4	RK5q	Nx	E2	RK4	RK5q
15	4.3e-002	228	950	15	4.3 e-002	228	760
30	4.5e-003	450	1875	30	1.1 e-002	450	1875
60	8.7e-004	900	3750	60	5.7e-003	900	3750

Table 3: Comparison at CFL=.2 for the linear case between the classical RKDG and RKDG-NDC variant for P1 and P2.

Nx	E2		WU(P1)	E2		WU(P2)
	RK2DG-P1	RKDG-NDC-P1		RK4DG-P2	RKDG-NDC-P2	
15	2.8e-001	1.1 e-001	72	1.1 e-002	6.2 e-003	144
30	5.0e-002	1.5 e-002	146	1.2 e-003	5.6 e-004	292
60	9.3e-003	4.4e-003	296	1.8 e-004	1.2 e-004	592

Thus these schemes are of order close to 2.5 in the linear case and to 2 in the Burgers case. The STDG-RK5q is unconditionally stable if one increases the number of sub-iterations. For a similar precision to that of CFL=.2 computation, the numbers of work units for increasing CFL is the following:

$$WU(CFL = .2) = 760, WU(CFL = .5) = 1950, WU(CFL = .9) = 2500.$$

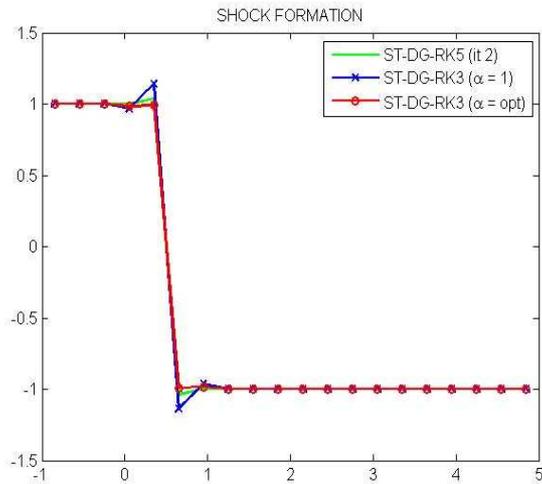
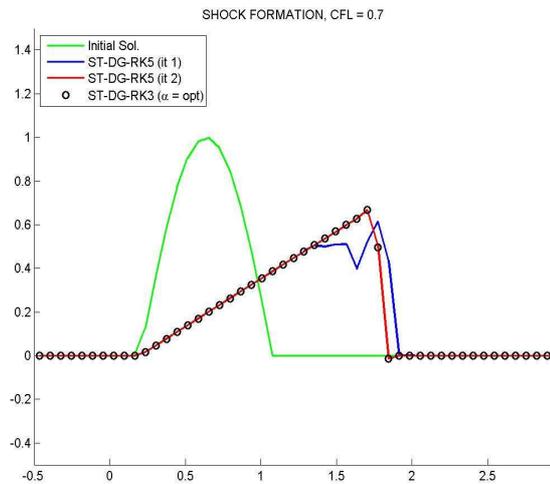
Thus for space time P1 Galerkin discretization, STDG-RK2 seems to be a good alternative for solving the nonlinear implicit problem.

A similar comparison was done for P2 Galerkin discretization in Table 2. The RK2 time scheme is replaced by the classical 4th order Runge-Kutta scheme. In this case the global order is the same as the P1 Galerkin scheme. The STDG-P2 will achieve the global order of 5 but at the cost of a great amount of sub-iterations. For these test cases pertaining to acoustics, it would seem that a Runge-Kutta scheme in time and Galerkin P2 in space is more efficient as work units per time step will never be greater than 4.

In Table 3 a comparison is made between the classical RKDG (P1 and P2) and the NDC variants. An optimal value of β which is problem dependant, divides at least by two the error.

3.3 STDG-RK2- α

The scalar non-linear Burgers equation is considered. Using MATLAB programming techniques, an optimum value of α can be found improving by a factor of 2 the STDG-

Figure 1: Effect of the optimal α coefficient.Figure 2: Effect of the optimal α coefficient.

RK2 CFL condition leaving untouched the scheme accuracy for steady cases. In Fig. 1, is shown in the case of a steady shock the effect of this optimal value (which varies with the size of the discretization). For a domain $[-1,5]$ discretized with 20 points, the optimum value is $\alpha=2$. In Fig. 2, in the case of a shock formation, comparison is made with STDG-RK5q solved at each time step by q times the 5-step Runge-Kutta scheme ($q=1$ and $q=2$).

One sub iteration of the RK5 STDG scheme is inaccurate, but for $q \geq 2$, results vary very little. With an optimal value of α , the STDG-RK2 is as precise as the STDG-RK5 at a cost ratio of 3 to 10.

This scheme is only valid for computing steady flows as for unsteady cases, precision deteriorates for any value of $\alpha \neq 1$.

3.4 RKDG- β

The case of a moving shock for the Burgers equation is considered. On Fig. 3 is shown the influence of the β parameter in the RKDG-NDC parameter on the L_2 error and the L_∞ with a minimum at $\beta=.8$. (Recall that $\beta=1$ is the original RKDG scheme). This optimum value of β though discretization dependent brings no CPU overcost, but cannot improve the original ($\beta=1$) CFL constraint.

4 Euler equations: P1 numerical results

All following computations were performed with a 3D AMR code with degenerated directions in 1D and 2D cases. This code solves the Euler equations (2.1).

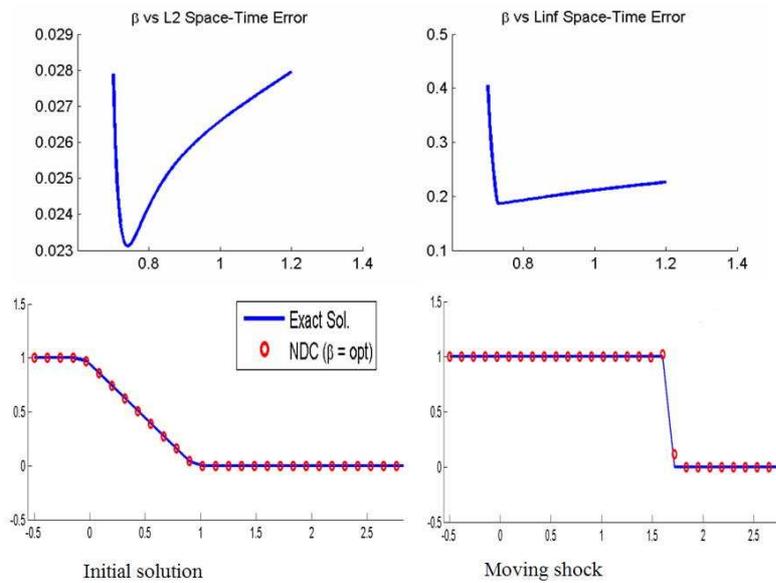


Figure 3: The scalar non-linear Burgers equation: RKDG-NDC.

4.1 Planar wave

We consider the problem of an unsteady planar wave on a Cartesian grid. The size of the cells is 2.5mm and the size of the computational domain is 2.5m . The simulation consists, starting from an initial state defined by the uniform flow at Mach number 0.4 along the x axis, in imposing a (static) pressure fluctuation at the downstream boundary. This fluctuation takes the following form

$$p(t) = \bar{p} + \Delta p \sin(2\pi t/T).$$

The amplitude Δp is 89Pa for a total pressure of 101kPa and the period defined by $T = N\Delta x/c$ with c the speed of sound and N the wave number which was set equal to 20, which corresponds to a signal of approximately 6700Hz . The wave goes upstream and a non reflection condition based on characteristics is imposed at the downstream boundary. Within this configuration, the wavelength is discretized on approximately twelve cells.

For an optimal value of β , as can be seen on Fig. 4, the RKDG-NDC can provide an efficient control of the diffusion.

4.2 Planar Gaussian wave

In Fig. 5 the different variants are compared in the case of a pulse propagation over a long distance. This isentropic pressure pulse is defined in $[0,1]$ initially in the following way:

$$\delta p/p = 10^{-3} \exp\left(-\frac{40}{\Delta x}(x-0.2)^2\right), \quad \delta u = \delta p/\rho c,$$

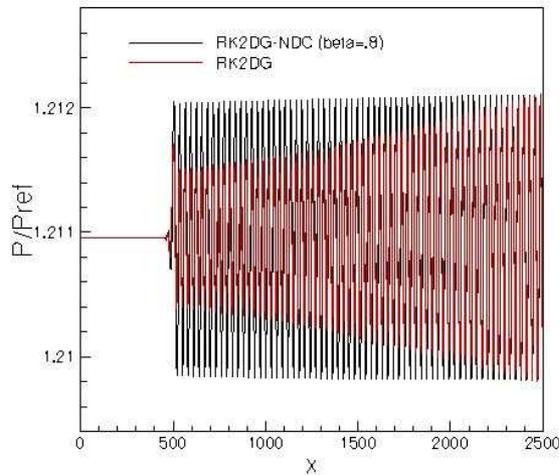


Figure 4: Planar wave propagating upstream a subsonic flow: comparison of MUSCL and DG schemes.

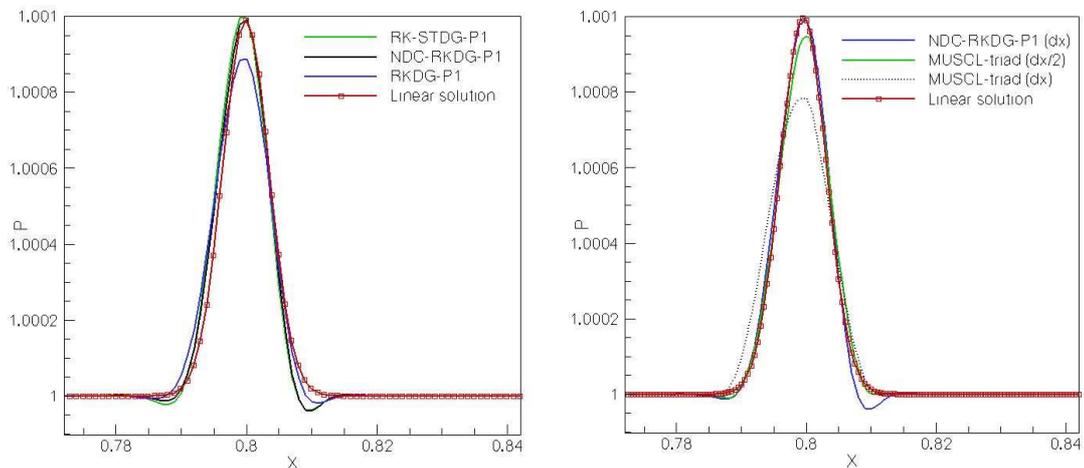


Figure 5: Propagating Gaussian acoustic wave (partial view): DG and MUSCL comparison.

with c the sound velocity. This test case measures dispersive and diffusive errors on a more complex problem than a simple sinusoidal wave.

The simulation ran with a 1000 point mesh ($\Delta x = .001$) and a CFL=.1 for 6000 time steps for the DG variants and 2000 points for a MUSCL-triad scheme [11] computation. As seen as before, the RKDG-NDC variant is better than the RKDG scheme (on the left hand side figure) and is as accurate as the STDG-RK2 scheme. It also requires 4 times less work units. These schemes show minimum diffusion and a little dispersion error. On the right hand side figure, the RKDG-NDC is compared with the MUSCL-triad scheme on a mesh twice the size so as to have comparable number of degrees of freedom. For the 1D case, work units are similar between a RKDG-NDC running on a N -size mesh with Nt time steps and MUSCL running on a $2N$ -size mesh and $2Nt$ time steps. The RKDG-

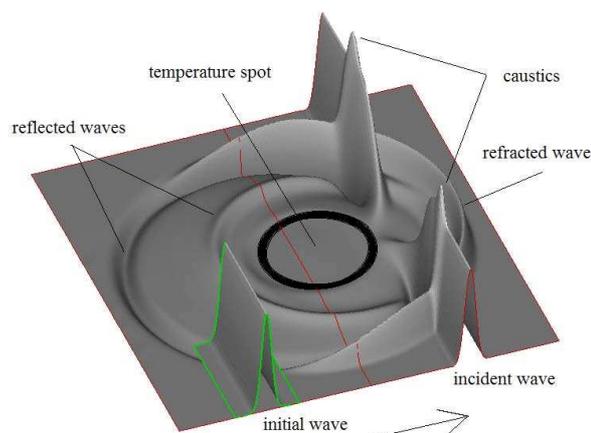


Figure 6: Pressure wave / temperature spot interaction: description of the different waves (STDG results).

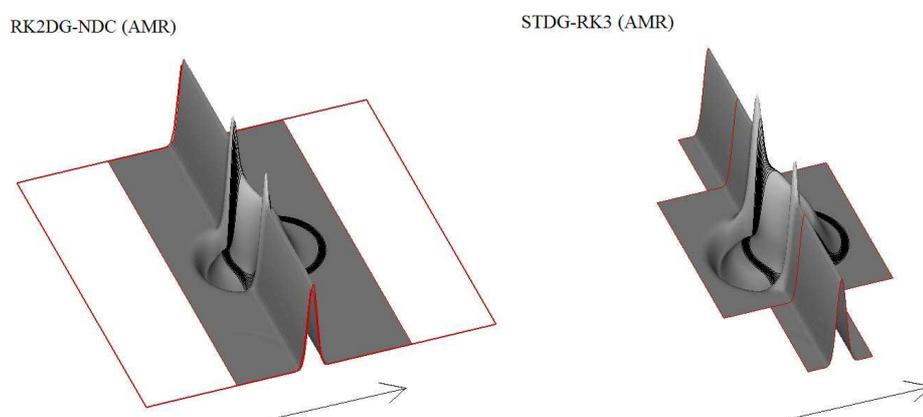


Figure 7: Pressure wave / temperature spot interaction: Comparison of the two variants with AMR.

NDC in 3D may become a little less expensive but dispersive errors are greater than the MUSCL scheme. Thus a higher order RKDG-NDC will be the subject of further study.

4.3 Acoustic wave interacting with a temperature spot

This problem is derived from [11]: the computational domain is a square of 4cm length and the dimension of the basis Cartesian grid is 100×100 . The patches of the AMR procedure are refined by a factor 3. The spot is centered at $(x = 21\text{mm}, y = 20\text{mm})$ and its radius is 5mm . Outside the spot, the temperature is 300K and inside the temperature is 1500K . The pressure is uniform and equal to 1 atm and the velocity is null. Starting from this initial flowfield, the same Gaussian acoustic wave as in the above section (positive pulse) moves forward through the computational domain along the x -direction. We have chosen a case in which the wavelength of this pressure perturbation is larger than

Transonic flow around a naca0012 airfoil ($M=0.8, \alpha=1.25^\circ$):

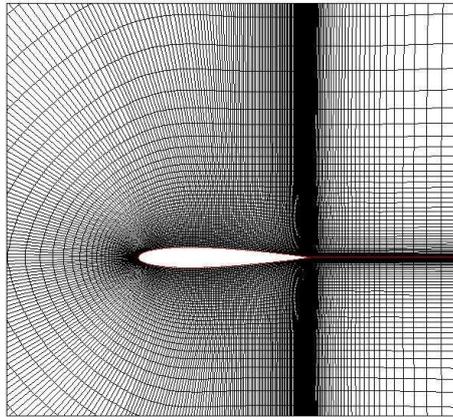


Figure 8: NACA0012: partial view of the curvilinear grid (449x65 nodes).

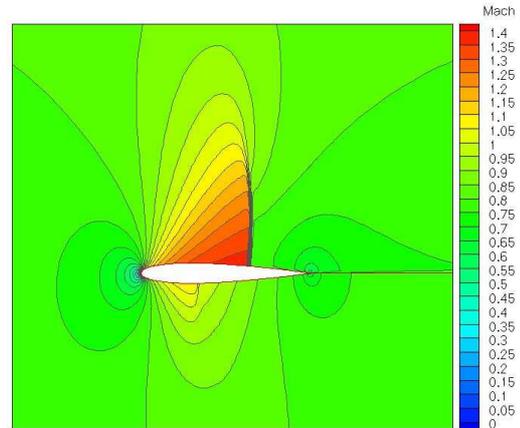


Figure 9: Partial view of the Mach contours.

Subsonic flow around a naca0012 airfoil ($M=0.5, \alpha=2^\circ$):

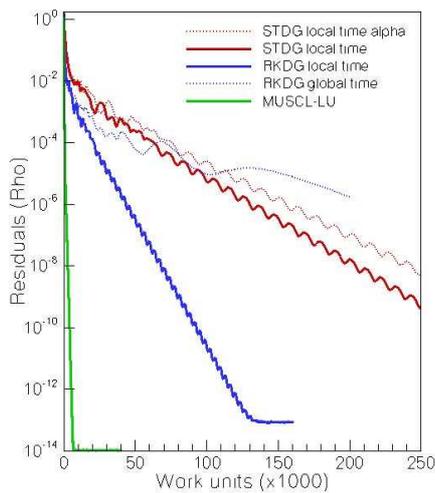


Figure 10: Convergence history of some variants.

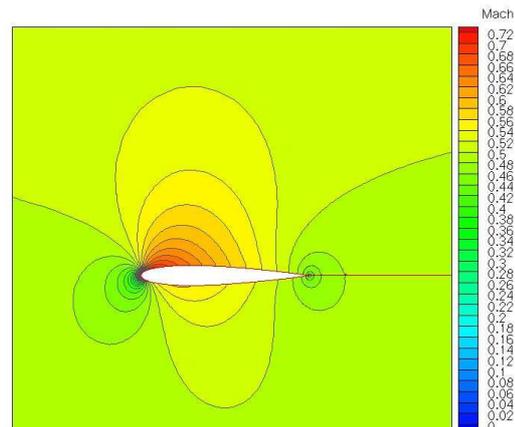


Figure 11: Partial view of the Mach contours.

the thickness of the temperature spot. The description of the different waves which occur during the interaction is presented in Fig. 6. In particular, the caustic waves due to the interaction of the planar incident wave and the curved diffracted wave originate from the non-linearity of the governing equations. An important point for this computation is not to diffuse or to amplify the incident acoustic wave.

We have tested STDG and RKDG-NDC schemes on this problem. As one could expect from the analysis, the two schemes give very close results but STDG is more stable than RKDG-NDC and so, allows smaller refinement patches (see Fig. 7).

Remark 4.1. When implemented with an AMR technique which needs refining locally the solution in time and in space, the STDG discretization has a great advantage over the RKDG as time space projection is immediate. This advantage is also available with the RKDG-NDC scheme as time gradients are easily computed.

4.4 Curvilinear grid

In order to illustrate the behaviour of these DG variants on curvilinear grids, a fine grid around a NACA0012 airfoil, composed of about 28600 elements in 2D with 320 points on the profile, (Fig. 8), has been used. Subsonic and transonic steady state results are presented in Figs. 9-11. In Fig. 10 the convergence history of some variants is presented. Residuals are plotted versus cost expressed in terms of number of work units. At convergence, all results compare well in terms of precision with classical schemes.

STDG schemes though allowing higher CFL (twice the one for RK2DG) are globally more expensive in terms of work units.

Finally these schemes are also compared with an implicit (LU) MUSCL scheme to remind us that is still much room for improvement for these explicit schemes in the case of steady flows.

5 Conclusion

Variants of space time Discontinuous Galerkin discretizations have been developed. The first replaces the implicit iterative STDG solver by a truncated explicit one. It is of interest when dealing with P1 schemes as accuracy and order are respected. This is not so much the case for STDG-P2 schemes. The second STDG- α is based on a time derivative relaxation technique. The third RKDG-NDC can be seen as a unified treatment identifying (coding wise) STDG and RKDG.

The 1D numerical study and 2D test cases confirm that introducing the upwinding parameter α allows higher CFL with no loss of accuracy but does not accelerate convergence to a steady state. Other time schemes are to be looked into.

Introducing the unifying parameter β controls numerical diffusion at no over cost preserving initial stability and accuracy.

Acknowledgments

The author would like to thank Germain Billet for his results and fruitful discussions concerning the test cases, and the reviewers for their very useful comments.

References

- [1] B. van Leer, Towards the ultimate conservative difference scheme, V. A second order sequel to Godunov's method, J. Comput. Phys., 32 (1979), 101-136.

- [2] J.-C. Jouhaud and M. Borrel, Discontinuous Galerkin and MUSCL strategies for an adaptive mesh refinement method, Proceedings of the 15th ICNMF, Monterey, Springer, 1996, pp. 400-405.
- [3] M. Borrel, DG and MUSCL approaches: Comparison of some recent variants for aeroacoustics and vortex computations, ECCOMAS CFD, Egmond aan Zee, The Netherlands, 2006.
- [4] M. Borrel and B. Berde, Moment approach for the Navier-Stokes equations, AIAA paper 95-1663, 12th CFD Conf., San Diego, 1995.
- [5] C. Drozo, M. Borrel, A. Lerat. Discontinuous Galerkin schemes for the compressible Navier-Stokes equations, Proceedings of the 16th ICNMF, Arcachon, Springer, 1998, pp. 266-271.
- [6] B. Cockburn, Discontinuous Galerkin Method for Problems Convection-Dominated. Reading Notes in Comput. Sc. and Eng., vol. 9, Springer, High-Order Methods for Computational Physics, 1999.
- [7] J. van der Vegt and H. van der Ven, Space-time discontinuous Galerkin finite element method with dynamics grid motion for inviscid compressible flows, part I. General formulation, J. Comput. Phys., 182 (2002), 546-585.
- [8] H. van der Ven and J. J. W. van der Vegt, Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows II. Efficient flux quadrature, Comput. Methods Appl. Mech. Engrg., 191 (2002), 4747-4780.
- [9] C. M. Klaij, J. J. W. van der Vegt and H. van der Ven, Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations, J. Comput. Phys., 217 (2006), 589-611.
- [10] C. M. Klaij, J. J. W. van der Vegt and H. van der Ven, Pseudo-time stepping methods for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, J. Comput. Phys., 219 (2006), 622-643.
- [11] G. Billet, Improvement of convective concentration fluxes in a one step reactive flow solver, J. Comput. Phys., 204 (2005), 319-352.
- [12] M. Borrel, J. Ryan and G. Billet, A Generalized patch AMR platform that uses cell centered or cell vertex solvers, ECCOMAS CFD, Egmond aan Zee, The Netherlands, 2006.
- [13] N. Huré and M. Borrel, A multiblock multigrid AMR method using a MUSCL or a Discontinuous Galerkin approach, Finite Volumes for Complex. Applications III, Hermes Penton Ltd, 2002.
- [14] J. Ryan and M. Borrel, Adaptive Mesh Refinement: A Coupling Framework For Direct Numerical Simulation Of Reacting Gas Flow, ICFD, Oxford, 2004.
- [15] S. R. Allmaras, A Coupled Euler/Navier-Stokes Algorithm for 2D Unsteady Transonic Shock/Boundary Layer Interaction, Ph.D. Thesis, Massachusetts Institute of Technology, 1989.