Vít Dolejší

An efficient implementation of the semi-implicit discontinuous Galerkin method for compressible flow simulation

In: Jan Chleboun and Karel Segeth and Tomáš Vejchodský (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Prague, May 28-31, 2006. Institute of Mathematics AS CR, Prague, 2006. pp. 74–79.

Persistent URL: http://dml.cz/dmlcz/702821

AN EFFICIENT IMPLEMENTATION OF THE SEMI-IMPLICIT DISCONTINUOUS GALERKIN METHOD FOR COMPRESSIBLE FLOW SIMULATION*

Vít Dolejší

Abstract

We deal with a numerical simulation of the inviscid compressible flow with the aid of the combination of the discontinuous Galerkin method (DGM) and backward difference formulae. We recall the mentioned numerical scheme and discuss implementation aspects of DGM, particularly a choice of basis functions and numerical quadratures for integrations. An illustrative numerical example is presented.

1. Introduction

Our aim is to develop a sufficiently robust, accurate and efficient numerical scheme for a simulation of compressible flows. Among several types of numerical schemes the discontinuous Galerkin method (DGM) seems to be a promising technique, see e.g., [2], [3], [5], [8], [9]. DGM is based on a piecewise polynomial but discontinuous approximation and represents a generalization of the finite element and finite volume methods. Although authors mostly claim that DGM is very suitable for the compressible flow simulation they admit one disadvantage: a high computational cost which prevents DGM from practical applications. Therefore an efficient implementation exhibits a challenging task.

In this paper we recall the semi-implicit numerical scheme proposed in [4],which is based on a combination of DGM for the space semi-discretization and the backward difference formula for the time discretization (Section 3.). Then we discuss some implementation aspects with respect to the CPU time, particularly a choice of the basis functions and numerical quadratures for integrations (Section 4.). Finally, one numerical example of an unsteady inviscid compressible flow through the forward facing step is presented for an illustration.

2. Problem formulation

The system of the *Euler equations* describing 2D inviscid compressible flow can be written in the form

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{2} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = 0 \quad \text{in } Q_T = \Omega \times (0,T), \tag{1}$$

*This work is a part of the research project MSM 0021620839 financed by the Ministry of Education of the Czech Republic and was partly supported by the Grant No. 316/2006/B-MAT/MFF of the Grant Agency of Charles University Prague.

where $\Omega \subset \mathbb{R}^2$ is a bounded polygonal domain occupied by a gas, $T > 0$ is the length of a time interval, $\boldsymbol{w} = (w_1, \dots, w_4)^{\mathrm{T}} = (\rho,\, \rho v_1,\, \rho v_2,\, e)^{\mathrm{T}}$ is the *state vector* and $\boldsymbol{f}_s(\boldsymbol{w}) = (\rho v_s,\, \rho v_s v_1 + \delta_{s1} p,\, \rho v_s v_2 + \delta_{s2} p,\, (e+p)\, v_s)^{\mathrm{T}},\ s = 1, 2$, are the *inviscid (Euler) fluxes*. We use the following notation: ρ – density, p – pressure, e – total energy, $\boldsymbol{v} = (v_1, v_2)$ – velocity, δ_{sk} – Kronecker symbol, $\gamma > 1$ – Poisson adiabatic constant. The equation of state implies that $p = (\gamma - 1)\,(e - \rho|\boldsymbol{v}|^2/2)$. The system (1) is equipped with a set of initial and boundary conditions, for details see, e.g., [7].

3. Discretization

In [4], we presented the discretization of the Euler equations (1) by the discontinuous Galerkin method (DGM). Therefore we do not derive the numerical scheme again but only present the main relations.

Let $\mathcal{T}_h \equiv \{K_i\}_{i \in I}$ denote a triangulation of the closure $\overline{\Omega}$ of the domain Ω into a finite number of closed elements (triangles or quadrilaterals) $K_i,\ i \in I$ with mutually disjoint interiors. Let $\partial K_i \equiv \cup_{j \in S(i)} \Gamma_{ij}\ \forall K_i \in \mathcal{T}_h$, where $S(i),\ i \in I$ are suitable index sets, Γ_{ij} is either a common face between neighbouring elements K_i and K_j or a boundary face (i.e. $\Gamma_{ij} \subset \partial\Omega$). Moreover, $\boldsymbol{n}_{ij} = ((n_{ij})_1, (n_{ij})_2)$ is the unit outer normal to ∂K_i on the face Γ_{ij}.

The approximate solution of (1) is sought in the space of discontinuous piecewise polynomial functions $\boldsymbol{S}_h$ defined by

$$\boldsymbol{S}_h \equiv [S_h]^4, \quad S_h \equiv S^{p,-1}(\Omega, \mathcal{T}_h) \equiv \{v; v|_K \in P^p(K)\ \forall\, K \in \mathcal{T}_h\}, \tag{2}$$

where $P^p(K)$ denotes the space of all polynomials on K of degree at most $p \geq 0$, p is an integer. For $\boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_h$ we introduce the forms

$$\begin{aligned} (\boldsymbol{w}_h, \boldsymbol{\varphi}_h) &= \int_\Omega \boldsymbol{w}_h(\boldsymbol{x}) \cdot \boldsymbol{\varphi}_h(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}, \qquad (3)\\ \boldsymbol{b}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) &= -\sum_{K \in \mathcal{T}_h} \int_K \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}_h) \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x_s}\, \mathrm{d}\boldsymbol{x} \\ &\quad + \sum_{K_i \in \mathcal{T}_h} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \boldsymbol{H}(\boldsymbol{w}_h|_{\Gamma_{ij}}, \boldsymbol{w}_h|_{\Gamma_{ji}}, \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi}_h \mathrm{d}S, \end{aligned}$$

where $\boldsymbol{H}$ is a *numerical flux*, $\boldsymbol{w}(t)|_{\Gamma_{ij}}$ and $\boldsymbol{w}(t)|_{\Gamma_{ji}}$ are the values of $\boldsymbol{w}$ on Γ_{ij} considered from the interior and the exterior of K_i, respectively, and at time t. The values of $\boldsymbol{w}(t)|_{\Gamma_{ji}}$ for $\Gamma_{ij} \subset \partial\Omega$ are given by the boundary conditions, for details, see [7]. Then we define the *semidiscrete problem*:

Definition 1: Function $\boldsymbol{w}_h$ is a *semidiscrete solution* of the problem (1), if

$$\begin{aligned} &\text{a)} \quad \boldsymbol{w}_h \in C^1([0,T]; \boldsymbol{S}_h), \qquad (4)\\ &\text{b)} \quad \left(\frac{\partial \boldsymbol{w}_h(t)}{\partial t}, \boldsymbol{\varphi}_h\right) + \boldsymbol{b}_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) = 0 \quad \forall\, \boldsymbol{\varphi}_h \in \boldsymbol{S}_h\ \forall\, t \in (0,T), \\ &\text{c)} \quad \boldsymbol{w}_h(0) = \boldsymbol{w}_h^0, \end{aligned}$$

	approximation		
#dof	P_1	P_2	P_3
FEM	n	$2.5n$	$6n$
DGM	$6n$	$12n$	$20n$

Tab. 1: *Comparison of degree of freedom of DGM and FEM for a triangular grid having n vertices.*

where $\boldsymbol{w}_h^0 \in \boldsymbol{S}_h$ denotes the initial condition. Here $C^1([0,T];\boldsymbol{S}_h)$ is the space of continuously differentiable mappings of the interval $[0,T]$ into $\boldsymbol{S}_h$.

The problem (4), a) – c) exhibits a system of ordinary differential equations for $\boldsymbol{w}_h(t)$ which has to be discretized by a suitable ODE method. In [4] we introduced the semi-implicit discretization of (4), a) – c), where the form $\boldsymbol{b}_h(\cdot,\cdot)$ was linearized and then the linear terms were treated implicitly by a multi-step backward difference formula and the nonlinear terms were approximated by a suitable higher order explicit extrapolation. Then the full space-time discretization leads to a system of linear algebraic equations at each time level, the numerical scheme is practically unconditionally stable and has a high order of accuracy with respect to the time coordinate.

Since for the purposes of this paper an exact form of the time discretization is not important we write the full space-time discretization schematically by

$$\left(\boldsymbol{M} + \tau_k \boldsymbol{C}(\boldsymbol{w}_h^k)\right) \boldsymbol{w}_h^{k+1} = \boldsymbol{g}(\boldsymbol{w}_h^k), \quad k = 0, 1, \ldots,$$

where $\boldsymbol{w}_h^k \in \boldsymbol{S}_h,\ k = 0, 1, \ldots$ represents an approximation of the solution at $t = t_k$, $\boldsymbol{M}$ is the mass matrix (6), $\boldsymbol{C}(\cdot)$ is a matrix representing the form $\boldsymbol{b}_h(\cdot,\cdot)$, $\boldsymbol{g}(\cdot)$ is a right-hand-side and $\tau_k \equiv t_{k+1} - t_k$ is a time step. For more details see [4].

4. Implementation aspects

Although DGM exhibits a very promising approach for a simulation of compressible flows, its main disadvantage is a higher number of degrees of freedom in comparison with the classical finite element method (FEM) which leads to a higher requirement on CPU time. Table 1 compares the degrees of freedom of DGM and FEM on a triangular grid with n vertices (than the number of triangles $\approx 2n$) for piecewise linear, quadratic and cubic approximations. We observe several times higher number of degrees of freedom for DGM than FEM. Therefore a very efficient implementation is a natural requirement for an industrial use of DGM. We discuss two items: choice of basis functions and numerical quadratures. Other aspects (e.g. linear solver, preconditioning,...) are a subject of the future research.

4.1. Choice of basis

For a numerical simulation of compressible flows it is suitable to use meshes consisting of triangles and quadrilaterals since numerical experiments show that quadri-

laterals are better for a resolution of effects within boundary layers around solid walls whereas triangles are more suitable for capturing of discontinuities (e.g., shock waves) with a general direction. An use of the Lagrangian basis known from FEM is not suitable for a combination of triangles and quadrilaterals. Since we have a discontinuous approximation we can employ a local basis on each element independently. A natural choice is an use of the Taylor basis on element $K_i \in \mathcal{T}_h$ in the form

$$\{\psi_j^{K_i}\}_{j=1}^{dof_{K_i}} \equiv \{(x_1 - x_1^{K_i})^{n_x}(x_2 - x_2^{K_i})^{n_y},\ n_x, n_y \geq 0,\ n_x + n_y \leq p\}, \tag{5}$$

where p is the degree of the polynomial approximation on K_i, $dof_{K_i} = (p+1)(p+2)/2$ is the number of degree of freedom on K_i and $(x_1^{K_i}, x_2^{K_i})$ is the barycentre of K_i.

However, numerical experiments show that the Taylor basis (5) is not suitable for a computations, since the *mass matrix* defined by

$$\boldsymbol{M} \equiv \left\{m_{(K_i,n_i),(K_j,n_j)}\right\}_{n_i=1,\ldots,dof_{K_i},K_i\in\mathcal{T}_h}^{n_j=1,\ldots,dof_{K_j},K_j\in\mathcal{T}_h}, \quad m_{(K_i,n_i),(K_j,n_j)} \equiv \int_\Omega \psi_{n_i}^{K_i}\,\psi_{n_j}^{K_j}\ \mathrm{d}x \tag{6}$$

has elements with very different magnitudes which causes a slow convergence of the linear algebraic problem. In order to save some CPU-time it is possible to use an approach [1] where basis (5) is replaced by the following one

$$\{\tilde{\psi}_j^{K_i}\}_{j=1}^{dof_{K_i}},\ \tilde{\psi}_j^{K_i} \equiv \frac{\psi_j^{K_i}}{\|\psi_j^{K_i}\|_{L^2(\Omega)}}, \quad j = 1, \ldots, dof_{K_i},\ K_i \in \mathcal{T}_h. \tag{7}$$

Based on numerical experiments we observed that the choice of the basis (7) saves the computational time approximately 50% in comparison with the basis (5).

We extended the idea from [1] in such a way that not only "normalization" but the full orthonormalization of the basis (5) is carried out. So that we employ the basis

$$\{\bar{\psi}_j^{K_i}\}_{j=1}^{dof_{K_i}},\ \text{ such that }\ (\bar{\psi}_j^{K_i}, \bar{\psi}_l^{K_i})_{L^2(\Omega)} = \delta_{jl}, \quad j, l = 1, \ldots, dof_{K_i},\ K_i \in \mathcal{T}_h. \tag{8}$$

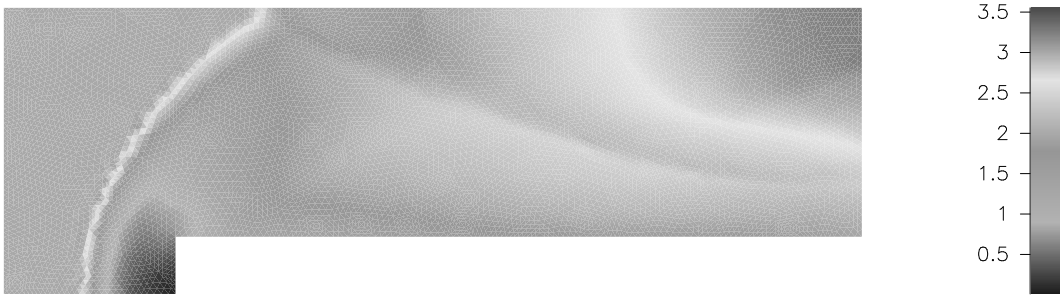
The orthonormalization is carried out by the Grant-Schmidt ortogonalization process. Although it is a known fact, that this algorithm is ill-conditioned we do not observed any problem with the stability of the Grant-Schmidt ortogonalization. It is caused by the fact that the dimension of the finite element space on each element (dof_{K_i}) is small and moreover if the basis is not (exactly) orthogonal it does not mind. We observe that the choice of the basis (8) saves the computational time approximately 90% in comparison with the basis (5).

4.2. Numerical integration

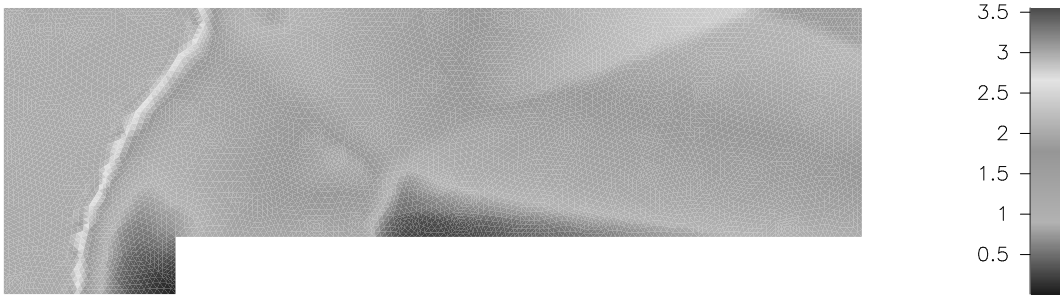
The integrals in (3) have to be evaluated with the aid of suitable numerical quadratures. An use of a numerical quadrature with a low order of accuracy can cause a loss of accuracy and on the other hand a numerical quadrature with a higher number of integration nodes requires longer CPU time. Therefore an use of some

type of integral	integ. rule	#nodes	order
edge	Gauss	$2p$	$4p-1$
quadrilateral	2D Gauss	$(2p)^2$	$4p-1$
triangle	Dunavant		$3p-1$

Tab. 2: *List of the used quadrature rules with the orders of accuracy and the number of integration nodes, p is the degree of polynomial approximation.*



$t = 1.0$



$t = 3.0$

Fig. 1: *Forward facing step, P_3 approximation, Mach number distributions.*

"optimal" numerical quadratures is necessary in order to balance the CPU-costs and the accuracy. Based on numerical experiments the classical Gauss quadrature formulas were employed for edge integrals. Concerning the volume integrals we used the 2D version of the Gauss formulas for quadrilateral elements and the Dunavant rules [6] for triangular elements. Table 2 shows the used quadrature rules with the orders of accuracy and the number of integration nodes.

In order to obtain a high efficiency of the implementation, the values of the test functions in integration nodes are evaluated a priori, so that we do not use any mappings of reference elements to physical ones. Therefore the evaluation of integrals in (3) exhibits a simple multiplicative multiplications of real arrays. This was the reason why we use the programming language Fortran 95 which is optimalized for arrays operations.

5. Numerical example

We consider a flow through the well-known forward facing step proposed in [10] with a constant initial condition given by $\rho = 1.4$, $\boldsymbol{v} = (3, 0)$, $p = 1$. Figure 1 shows Mach number distributions obtained by P_3 approximation on a grid having 1 033 triangles at $t = 0.1$ and $t = 0.3$.

References

[1] F. Bassi: Private communication, Charles University Prague, 2005.

[2] F. Bassi, S. Rebay: *High-order accurate discontinuous finite element solution of the 2D Euler equations.* J. Comput. Phys. **138**, 1997, 251–285.

[3] V. Dolejší: *On the discontinuous Galerkin method for the numerical solution of the Navier–Stokes equations.* Int. J. Numer. Methods Fluids **45**, 2004, 1083–1106.

[4] V. Dolejší: *Higher order semi-implicit discontinuous Galerkin finite element schemes for compressible flow simulation.* In: Software and Algorithms of Numerical Mathematics, 2005, (submitted).

[5] V. Dolejší, M. Feistauer: *Semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow.* J. Comput. Phys. **198**, 727–746.

[6] D.A. Dunavant: *High degree efficient symmetrical gaussian quadrature rules for the triangle.* Int. J. Numer. Methods Eng. **21**, 1985, 1129–1148.

[7] M. Feistauer, J. Felcman, I. Straškraba: *Mathematical and computational methods for compressible flow.* Oxford University Press 2003.

[8] R. Hartmann, P. Houston: *Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations.* J. Comput. Phys. **183**, 2002, 508–532.

[9] J.J.W. van der Vegt, H. van der Ven: *Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. I: General formulation.* J. Comput. Phys. **182**, 2002, 546–585.

[10] P. Woodward, P. Colella: *The numerical simulation of two-dimensional fluid flow with strong shocks.* J. of Comput. Phys. **54**, 1984, 115–173.