

Discontinuous Galerkin method with Taylor basis functions

Aditya Kashi

March 28, 2017

Abstract

Discontinuous Galerkin (DG) finite element method (FEM) with Taylor basis theoretically has several advantages over traditional nodal Lagrange basis FEM for the numerical solution of hyperbolic problems. We wish to investigate theoretically and numerically a Taylor basis DG FEM. One of the advantages of using Taylor basis functions is that reconstructing higher order derivatives becomes straightforward, which is the key idea in the ‘reconstructed’ DG (RDG) method. We would like to investigate the effectiveness of this method as well.

1 Introduction

We wish to investigate the solution of the compressible Euler equations in two-dimensions using Taylor basis functions in a discontinuous Galerkin (DG) finite element method (FEM). Nodal Lagrange finite elements are by far the most commonly used as the polynomial basis of choice in fields like solid mechanics and fluid mechanics. However, an alternative exists in modal basis functions, such as Legendre basis and Taylor basis.

There are some advantages that the hierarchical, modal, Taylor basis functions have over nodal Lagrange basis functions [5, 1].

- The set of basis functions of a certain polynomial degree contains the basis functions of all lower-degree sets of basis functions, ie., the basis is *hierarchical*. This makes easier the implementation of p-adaptation (dynamically changing polynomial degree locally based on accuracy requirements during the simulation) and p-multigrid solvers (utilizing corrections to the solution computed from lower-order solves).
- The Taylor basis expansion remains same and the basis functions retain the same form irrespective of the geometric type of elements. This makes for easier implementation of codes that work both on triangles and quadrangles, or all three of tetrahedra, prisms and hexahedra. Another aspect of this is that the number of degrees of freedom in case of Taylor basis is smaller than that in case of nodal basis for non-simplicial elements, while maintaining optimal order of accuracy.
- Thirdly, they make implementation of reconstruction DG methods (described later) efficient and elegant. This is because the spatial derivatives of the unknowns are readily available as those are the degrees of freedom. Using them, higher derivatives can be reconstructed to improve the accuracy of the scheme, as done in high-order (higher than 1st order) finite volume methods.

2 Governing equations

The compressible Euler equations, in n_d spatial dimensions, are

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{j=1}^{n_d} \frac{\partial \mathbf{F}_j}{\partial x_j} = \mathbf{0} \quad \mathbf{x} \in \Omega, t \in [0, T] \quad (1)$$

with some initial condition

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad (2)$$

and some combination of several possible types of boundary conditions. $\mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^m$ is the vector of the m conserved variables ($m = 4$ for the 2D Euler equations), and $\mathbf{F}_j(\mathbf{u}(\mathbf{x}, t)) \in \mathbb{R}^m$ are the flux functions. For two-dimensional flows, these are

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho e \end{bmatrix}, \quad \mathbf{F}_1 = \begin{bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(\rho e + p) \end{bmatrix}, \quad \mathbf{F}_2 = \begin{bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 \\ u_2(\rho e + p) \end{bmatrix}. \quad (3)$$

In the above, ρ is the fluid density, u_j are the velocity components, e is the specific energy, and p is the pressure. The governing equations are closed by the ideal gas equation of state

$$p = \rho(e - \frac{1}{2}|\mathbf{u}|^2). \quad (4)$$

For a compact notation, we define a flux ‘matrix’ \mathbf{F} as the matrix with the x- and y- fluxes as the columns. We can then write (1) as

$$\mathbf{u}_t + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{0} \quad (5)$$

where $\nabla \cdot$ is now the tensor divergence and $\mathbf{F} \in \mathbb{R}^{m \times n_d}$. This means $\mathbf{F} : \mathbb{R}^{n_d} \rightarrow \mathbb{R}^m$ maps any vector (direction) in space to the flux in that direction. Note that the operator \mathbf{F} is a function of the state \mathbf{u} .

3 Finite element formulation

A weak formulation is derived in a ‘broken’ Sobolev space and discretized by finite element method [5]. For $w \in W$, a suitable test function space, we can multiply (5) by w and integrate over an element Ω_e . Taking μ as the domain measure and s as the boundary measure,

$$\int_{\Omega_e} \frac{\partial \mathbf{u}}{\partial t} w \, d\mu + \int_{\Omega_e} \nabla \cdot \mathbf{F}(\mathbf{u}) w \, d\mu = 0. \quad (6)$$

Integrating the flux term by parts, we have

$$\int_{\Omega_e} \frac{\partial \mathbf{u}}{\partial t} w \, d\mu + \int_{\Gamma_e} \mathbf{F}(\mathbf{u}) \hat{\mathbf{n}} w \, ds - \int_{\Omega_e} \mathbf{F}(\mathbf{u}) \nabla w \, d\mu = 0 \quad (7)$$

where Γ_e is the boundary of element Ω_e , and $\hat{\mathbf{n}}$ is the unit normal vector to Γ_e .

Let \mathcal{T}_h be a set of subsets of Ω_h such that $\bigcup_{K \in \mathcal{T}_h} K = \Omega_h$ and $K_i \cap K_j = \emptyset \, \forall K_i, K_j \in \mathcal{T}_h, i \neq j$, and ∂K is a Lipschitz boundary for all $K \in \mathcal{T}_h$. One choice of the discrete function space could be, for some positive integer k ,

$$W_h := \{v \in L^2(\Omega) \mid v|_K \in C^k(K) \, \forall K \in \mathcal{T}\}. \quad (8)$$

We can then state the discrete weak formulation on each element as follows. Find $\mathbf{u}_h \in W_h^m$ such that

$$\int_K \frac{\partial \mathbf{u}_h}{\partial t} w d\mu + \int_{\partial K} \mathbf{F}(\mathbf{u}_h) \hat{\mathbf{n}} w ds - \int_K \mathbf{F}(\mathbf{u}_h) \nabla w d\mu = \mathbf{0} \quad \forall w \in W_h, \forall K \in \mathcal{T}. \quad (9)$$

Consider basis $B_i(\mathbf{x})$ on an element Ω_e of the mesh \mathcal{T}_h . We can then write $\mathbf{u}_h = \sum_{j=1}^n \mathbf{u}_j B_j(\mathbf{x})$. Further, we introduce the numerical flux \mathbf{h} defined such that for a face f between two elements Ω_L and Ω_R with states (conserved variables) \mathbf{u}_L and \mathbf{u}_R ,

$$\mathbf{F}(\mathbf{u})|_f \hat{\mathbf{n}} \approx \mathbf{h}(\mathbf{u}_L, \mathbf{u}_R, \hat{\mathbf{n}}). \quad (10)$$

This is necessary because \mathbf{u} is, in general, discontinuous across element interfaces f and the analytical flux $\mathbf{F}(\mathbf{u})|_f$ is not uniquely defined.

The discrete form becomes: find the n degrees of freedom (DOFs) u_{lj} of each conserved variable $\mathbf{u}_l \in \mathbb{R}^n, l = 1, \dots, m$ such that (note: $\mathbf{u}_L, \mathbf{u}_R$ and \mathbf{u}_h still denote functions in the discrete space W_h^m)

$$\sum_{j=1}^n \int_{\Omega_e} B_i B_j d\mu \frac{du_{lj}}{dt} + \int_{\Gamma_e} \mathbf{h}(\mathbf{u}_L(\mathbf{x}), \mathbf{u}_R(\mathbf{x}), \hat{\mathbf{n}}) B_i ds - \int_{\Omega_e} \mathbf{F}(\mathbf{u}_h(\mathbf{x})) \nabla B_i d\mu = \mathbf{0}, \quad i \in 1, \dots, n, \forall \Omega_e \in \mathcal{T}. \quad (11)$$

We denote by $\mathbf{U} := [\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_m]^T \in \mathbb{R}^{mn}$ the stacked vector of all degrees of freedom of all conserved variables on the element Ω_e ; note that all DOFs of any conserved variable are put consecutively together before starting the next conserved variable. We thus need to solve the following ODE on each element.

$$M \frac{d\mathbf{U}}{dt} = \mathbf{r}(\mathbf{U}(t)). \quad (12)$$

where $M \in \mathbb{R}^{mn \times mn}$ is a block-diagonal matrix having the m diagonal blocks $[M_D]_{ij} = \int_{\Omega_e} B_i B_j d\mu$ and $\mathbf{r}(\mathbf{U})$ is a stacked vector of the same form as \mathbf{U} , with the i th sub-vector having DOFs corresponding to the i th conserved variable: $\mathbf{r}_i(\mathbf{u}) := - \int_{\Gamma_e} \mathbf{h}(\mathbf{u}_{hL}, \mathbf{u}_{hR}, \hat{\mathbf{n}}) B_i ds + \int_{\Omega_e} \mathbf{F}(\mathbf{u}_h) \nabla B_i d\mu$.

4 Taylor basis functions

In a Taylor basis, a function is expressed as a Taylor expansion about the element center. In 2D, a quadratic or P2 expansion in an element e would be written as

$$u_h = u_c + \frac{\partial u}{\partial x} \Big|_c (x - x_c) + \frac{\partial u}{\partial y} \Big|_c (y - y_c) + \frac{\partial^2 u}{\partial x^2} \Big|_c \frac{(x - x_c)^2}{2} + \frac{\partial^2 u}{\partial y^2} \Big|_c \frac{(y - y_c)^2}{2} + \frac{\partial^2 u}{\partial x \partial y} \Big|_c (x - x_c)(y - y_c) \quad (13)$$

where the subscript 'c' indicates the quantity at the element's geometric center. Let A_e be the element area. Then we can take an average of both sides over the element to get

$$\begin{aligned} \tilde{u} &= u_c + \frac{\partial u}{\partial x} \Big|_c \frac{1}{A_e} \int_{\Omega_e} (x - x_c) d\mu + \frac{\partial u}{\partial y} \Big|_c \frac{1}{A_e} \int_{\Omega_e} (y - y_c) d\mu \\ &\quad + \frac{\partial^2 u}{\partial x^2} \Big|_c \frac{1}{A_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2} d\mu + \frac{\partial^2 u}{\partial y^2} \Big|_c \frac{1}{A_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2} d\mu + \frac{\partial^2 u}{\partial x \partial y} \Big|_c \frac{1}{A_e} \int_{\Omega_e} (x - x_c)(y - y_c) d\mu \\ &= u_c + \frac{\partial^2 u}{\partial x^2} \Big|_c \frac{1}{A_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2} d\mu + \frac{\partial^2 u}{\partial y^2} \Big|_c \frac{1}{A_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2} d\mu + \frac{\partial^2 u}{\partial x \partial y} \Big|_c \frac{1}{A_e} \int_{\Omega_e} (x - x_c)(y - y_c) d\mu \end{aligned} \quad (14)$$

where \tilde{u} is the average value of the unknown function over the element, and μ is the area measure in 2D. This last equation can be subtracted from (13) to get

$$u_h = \tilde{u} + \frac{\partial u}{\partial x}\Big|_c (x - x_c) + \frac{\partial u}{\partial y}\Big|_c (y - y_c) + \frac{\partial^2 u}{\partial x^2}\Big|_c \left(\frac{(x - x_c)^2}{2} - \frac{1}{A_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2} d\mu \right) + \frac{\partial^2 u}{\partial y^2}\Big|_c \left(\frac{(y - y_c)^2}{2} - \frac{1}{A_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2} d\mu \right) + \frac{\partial^2 u}{\partial x \partial y}\Big|_c \left((x - x_c)(y - y_c) - \frac{1}{A_e} \int_{\Omega_e} (x - x_c)(y - y_c) d\mu \right). \quad (15)$$

This gives us our basis functions and corresponding degrees of freedom on a physical element Ω_e . We also normalize the basis functions to get a better-conditioned discrete problem. Finally, the basis functions used for a P2 ‘Taylor finite element’ are

$$B_1(\mathbf{x}) = 1 \quad (16)$$

$$B_2(\mathbf{x}) = \frac{(x - x_c)}{\Delta x} \quad (17)$$

$$B_3(\mathbf{x}) = \frac{(y - y_c)}{\Delta y} \quad (18)$$

$$B_4(\mathbf{x}) = \left(\frac{(x - x_c)^2}{2\Delta x^2} - \frac{1}{A_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2} d\mu \right) \quad (19)$$

$$B_5(\mathbf{x}) = \left(\frac{(y - y_c)^2}{2\Delta y^2} - \frac{1}{A_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2} d\mu \right) \quad (20)$$

$$B_6(\mathbf{x}) = \left(\frac{(x - x_c)(y - y_c)}{\Delta x \Delta y} - \frac{1}{A_e} \int_{\Omega_e} (x - x_c)(y - y_c) d\mu \right) \quad (21)$$

with the corresponding coefficients, the degrees of freedom, being

$$\tilde{U} := \tilde{u} \quad (22)$$

$$U_x := \frac{\partial u}{\partial x}\Big|_c \Delta x \quad (23)$$

$$U_y := \frac{\partial u}{\partial y}\Big|_c \Delta y \quad (24)$$

$$U_{xx} := \frac{\partial^2 u}{\partial x^2}\Big|_c 2\Delta x^2 \quad (25)$$

$$U_{yy} := \frac{\partial^2 u}{\partial y^2}\Big|_c 2\Delta y^2 \quad (26)$$

$$U_{xy} := \frac{\partial^2 u}{\partial x \partial y}\Big|_c \Delta x \Delta y \quad (27)$$

where Δx and Δy are $\frac{1}{2}(x_{max} - x_{min})$ and $\frac{1}{2}(y_{max} - y_{min})$ respectively, and subscripts max and min indicate maximum and minimum values over the element. Note that symbols such as U_x include their respective normalization factors.

This can be similarly extended to any desired polynomial degree in any number of dimensions.

An interesting point is that the Taylor basis functions above are defined on the physical element. Thus, any polynomial function of these functions or their gradients remains polynomial. This means that terms in the mass matrices, stiffness matrices etc. can be integrated exactly by Gaussian quadrature of high-enough order, *even for curved elements*. This may be contrasted to how Lagrange

basis functions are defined on physical elements (see, for example, chapter 12 in [4]). The finite-dimensional function space on element K is defined as

$$P_K = \{p : p(\mathbf{x}) = \hat{p}(F^{-1}(\mathbf{x})), \mathbf{x} \in K, \hat{p} \in P_{\hat{K}}\} \quad (28)$$

where $F : \hat{K} \rightarrow K$ is a diffeomorphism from the reference element \hat{K} to the physical element K . For curved elements, F^{-1} is not a polynomial in general. Thus, integrands in a weak form will, in general, not be polynomials and thus cannot be integrated exactly.

5 Reconstruction DG scheme

One issue with the DG FEM method is its high cost compared to finite volume schemes for low to intermediate levels of desired accuracy. They become competitive only when a high-enough level of accuracy is required. DG FEM is also more expensive than a continuous FEM scheme of the same order on the same grid, since degrees of freedom are not shared among elements. One way of reducing the cost is the reconstruction DG (RDG) FEM scheme [6]. A Taylor basis RDG scheme can be implemented in a reasonably straightforward manner to give high-order accuracy using less degrees of freedom than a DG scheme.

Consider a DG P1 scheme with Taylor elements. The degrees of freedom are the mean value over the element and the first spatial derivatives at the element centre. Using this data, the three second spatial derivatives can be reconstructed using data from face-neighbouring elements, assuming a smooth enough solution in the region. Consider the union of a given element Ω_i and its face-neighbouring elements Ω_j . In 2D, $j = 1, 2, 3$ or $j = 1, 2, 3, 4$ when Ω_i is a triangle or quadrangle respectively. We write the unknown $u(\mathbf{x})$ as a Taylor series on this union (using the nomenclature introduced in the previous section).

$$u(\mathbf{x}) = \tilde{U}_i + U_{xi}B_{i_2}(\mathbf{x}) + U_{yi}B_{i_3}(\mathbf{x}) + U_{xxi}B_{i_4}(\mathbf{x}) + U_{yyi}B_{i_5}(\mathbf{x}) + U_{xyi}B_{i_6}(\mathbf{x}) \quad (29)$$

where B are the basis functions on element i and the derivatives are those at the centre of element i . We can also Taylor-expand the first derivatives.

$$\frac{\partial u}{\partial x} = U_{xi}/\Delta x_i + U_{xxi}B_2/\Delta x_i + U_{xyi}/\Delta x_i \quad (30)$$

$$\frac{\partial u}{\partial y} = U_{yi}/\Delta y_i + U_{xyi}B_2/\Delta y_i + U_{yyi}/\Delta y_i \quad (31)$$

Note that we already know the first-order derivatives as those are our degrees of freedom. The second derivatives are the unknowns we wish to reconstruct. To this end, the above functions can be evaluated at the element-centres \mathbf{x}_j of each of the face-neighbouring elements j .

$$u_j = \tilde{U}_i + U_{xi}B_{i_2}(\mathbf{x}_j) + U_{yi}B_{i_3}(\mathbf{x}_j) + U_{xxi}B_{i_4}(\mathbf{x}_j) + U_{yyi}B_{i_5}(\mathbf{x}_j) + U_{xyi}B_{i_6}(\mathbf{x}_j) \quad (32)$$

$$\left. \frac{\partial u}{\partial x} \right|_j = U_{xi}/\Delta x_i + U_{xxi}B_{i_2}(\mathbf{x}_j)/\Delta x_i + U_{xyi}B_{i_3}(\mathbf{x}_j)/\Delta x_i \quad (33)$$

$$\left. \frac{\partial u}{\partial y} \right|_j = U_{yi}/\Delta y_i + U_{xyi}B_{i_2}(\mathbf{x}_j)/\Delta y_i + U_{yyi}B_{i_3}(\mathbf{x}_j)/\Delta y_i \quad (34)$$

This can be expressed as the following linear system.

$$\begin{bmatrix} B_4^j & B_5^j & B_6^j \\ B_2^j & 0 & B_3^j \\ 0 & B_3^j & B_2^j \end{bmatrix} \begin{bmatrix} U_{xxi} \\ U_{yyi} \\ U_{xyi} \end{bmatrix} = \begin{bmatrix} u_j - (\tilde{U}_i + U_{xi}B_2^j + U_{yi}B_3^j) \\ \frac{\Delta x_i}{\Delta x_j} U_{xj} - U_{xi} \\ \frac{\Delta y_i}{\Delta y_j} U_{yj} - U_{yi} \end{bmatrix} =: \begin{bmatrix} R_1^j \\ R_2^j \\ R_3^j \end{bmatrix} \quad (35)$$

where the superscript j indicates that the function is evaluated at the element-centre of j . Writing these equations for all face-neighbours j , we get a 9x3 system for a triangle and a 12x3 system for a quadrilateral, in the three variables - the three second derivatives at the centre of cell i . This can be solved in a least squares sense to give a unique solution. Further, a WENO reconstruction can be used to suppress spurious oscillations to preserve nonlinear stability [7].

6 DG vs SUPG

The ‘streamline upwind Petrov-Galerkin’ method is a technique for solving hyperbolic PDEs that uses continuous trial and test function spaces, as opposed to DG methods. This means that degrees of freedom are shared between surrounding elements, and thus the number of degrees of freedom required to attain a given error level is lower than DG, at least for smooth solutions. SUPG is stable in presence shocks as well. However, it has its issues, as described nicely in section 2.5 of [3].

7 Outline of the project

One can think of the following questions that need answering.

7.1 Taylor basis DG

- (AW) For what H^k functions are these degrees of freedom well-defined? Recall for a Lagrange interpolant, we need at least continuous functions for prescribe nodal values. How would this case be different? Solobev embedding theorem might be useful here.
- Show that the degrees of freedom described are unisolvent, and thus that a ‘Taylor element’ is a valid finite element.
- (AW) How can we extend error estimates for standard DG FE space to DG Taylor basis FE? A starting point would depend on the answer to the first point above.

7.2 RDG method

- (AW) Given the example of reconstructing second derivatives, can one estimate the H^2 semi-norm of the local error between a generic H^2 function and its reconstructed interpolant over a triangle? If it can be done then it justifies that the reconstructed interpolant is a good approximation for the second derivative.
- (AW) Why solve a least-square system? If we are able to generate 9 equations for the reconstruction, why not include more higher derivatives in the basis and solve uniquely for 9 unknown coefficients?
- Prove that the least-squares system will have a unique solution, ie., that the 9x3 or 12x3 matrix has full column rank.

On the implementation side, the tentative plan is as follows. First implement Taylor basis DG FEM for the compressible Euler equations on 2D hybrid (triangles and quadrangles) grids. Total variation diminishing Runge-Kutta explicit time-stepping will be used.

Two test cases are planned - the isentropic vortex problem on a periodic domain, for which an analytical solution is available, and subsonic inviscid flow over a cylinder, for which error can

computed by measuring the entropy generated. Note that entropy generation should be zero for an ideal inviscid subsonic flow. Also, curved elements are necessary for this case [2].

Once that is accomplished, tentatively, the reconstruction DG scheme will be implemented to investigate the order of accuracy achieved and the computing resources needed as compared to a regular DG method of same formal order of accuracy.

References

- [1] Vadym Aizinger, Dmitri Kuzmin, and Lukas Koros. Scale separation in fast hierarchical solvers for discontinuous Galerkin methods. *Applied Mathematics and Computation*, 266:838–849, 2015.
- [2] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *Journal of Computational Physics*, 128:251–285, 1997.
- [3] Ralf Hartmann. *Adaptive Finite Element Methods for the Compressible Euler Equations*. PhD thesis, University of Heidelberg, 2002.
- [4] Claes Johnson. *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, 1987.
- [5] Hong Luo, Joseph D. Baum, and Rainald Löhner. A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids. *Journal of Computational Physics*, 227:8875–8893, 2008.
- [6] Hong Luo, Luqing Luo, and Robert Nourgaliev. A reconstructed discontinuous Galerkin method for the Euler equations on arbitrary grids. *Communications in Computational Physics*, 12(5):1495–1519, 2012.
- [7] Hong Luo, Yidong Xia, Shujie Li, Robert Nourgaliev, and Chunpei Cai. A Hermite WENO reconstruction based discontinuous Galerkin method for the Euler equations on tetrahedral grids. *Journal of Computational Physics*, 231:5489–5503, 2012.