Discontinuous Galerkin Method (DGM): from classical to isogeometric

hyperbolic problems

Asma Gdhami* — Regis Duvigneau** — Maher Moakher*

* Laboratoire de Modélisation Mathématique et Numérique dans les Sciences de l'Ingénieur
ENIT
Tunis
Tunisie

** INRIA
Université Côte dAzur
Sophia-Antipolis
FRANCE

RÉSUMÉ. L'Analyse isogéométrique (AIG) est une stratégie moderne de résolution numérique des équations différentielles, proposée à l'origine par Tom Hughes, Austin Cottrell et Yuri Bazilevs en 2005. Cette technique de dicretisation est une généralisation de lanalyse par éléments finis classique (AEF), conçue pour intégrer la conception assistée par ordinateur (CAO) et AEF, afin de combler l'écart entre la description géométrique et l'analyse des problèmes d'ingénierie [1]. Le but de ce travail est d'examiner et d'évaluer la méthode de Galerkin discontinue (GD) classique et la méthode de GD dans le contexte isogéométrique (IG) pour résoudre le problème d'advection. Ces deux méthodes sont basées sur le choix d'une base lagrangienne locale et d'une base de Bernstein respectivement.

ABSTRACT. Isogeometric analysis (IGA) is a generalization of classical finite element analysis (FEA) with the main aim of closing the gap between the geometrical description and the analysis of engineering problems. The basic IGA concept, based on the isoparametric paradigm, consisted of using basis functions commonly found in CAD geometries, such as B-spline, to represent both the geometry and the physical fields in the solution of problems governed by partial differential equations (PDEs) [1]. The purpose of this work is to examine and evaluate classical discontinuous Galerkin (CDG)method and discontinuous Galerkin method in the isogeometric context (IGDGM) for solving time dependent, advection problem. These two methods are based on the choice of a local Lagrangian basis and Bernstein basis respectively.

MOTS-CLÉS : Galerkin Discontinu, analyse isogéométrique, flus de Lax-Friedrichs, extraction de Bézier.

KEYWORDS : Discontinuous Galerkin, isogeometric analysis, Lax-Friedrichs flux, Bézier extraction.

1. Introduction & background

The CDG method was originally introduced in 1971 by Reed and Hill [5], for the numerical solution of the nuclear transport PDE problem. Subsequently the method has found far greater use in broad application in large-scale data intensive science and engineering problems. In contrast to the stabilized continuous Galerkin FEM, DG method produce stable discretizations without the need for stabilization parameters. However, this method combine the best properties of the finite volume (FV) method and continuous Galerkin FEM. In the fact, FV method can only use lower degree polynomials, and continuous FEM require higher regularity due to the continuity requirements, therefore, the idea of this method is to decompose the original problem into a set of subproblems that are connected using an appropriate transmission condition (known as the numerical flux). Though DG methods have gained increasing traction in large-scale application modeling and analysis, a shortcoming in the DG methodology is the inability to fully recover complex underlying geometries in the meshing domain. To overcome this problem, we combine IGA with the DG method to get IGDG method. As mentioned before, IGA is a computational technique that improves on and generalizes the classical FE method, the main benefit of thismethod is the exact representation of the geometry in the language of computer aided design (CAD) tools. This simplifies the meshing as the computational mesh is implicitly created by the engineer using the CAD tool. The IGDG method combines the best properties of the FV method and IGA, in fact FV method can only use lower degree polynomials, and IGFE method require to use functions from CAD like Bernstein (B-spline, NURBS) to determine the field where the PDE takes place and to numerically solve it. Therefore, The IGDG method is the DG method formulated on element that exactly preserve the geometries generated by CAD tools. An important property of B-spline in the context of IGA is the ability to perform Bézier extraction. Bézier extraction provides the capability of recovring a local Bernstein-Bézier representation of the geometry from the global B-spline CAD. In this work we will discusses specific details of implementation of IGDG method for the advection problem.

2. Bernstein basis

Definition (Univariate Bernstein).
The Bernstein polynomials of degree p are defined explicitly over the interval $[0, 1]$ by :

$$B_p^k(\zeta) = C_p^k \zeta^k (1-\zeta)^{p-k} \qquad \forall \quad k = 0, ..., p$$

Definition (Multivariate Bernstein).
In order to define Bernstein in higher dimensions, we make use of the tensor product.
Let $p = (p_1, p_2, ..., p_d)$ be a vector in N^d. The d-dimensional Bernstein polynomials are

defined by a tensor product of d univariate Bernstein polynomials with possibly different degrees $p_1, p_2, ..., p_d$ and multi-indices $k_1, k_2, ..., k_d$. Therefore, $\forall \zeta = (\zeta_1, \zeta_2, ..., \zeta_d) \in [0,1]^d$ we get :

$$B_p^k(\zeta) = B_{p_1}^{k_1}(\zeta_1) \otimes B_{p_2}^{k_2}(\zeta_2) \otimes ... \otimes B_{p_d}^{k_d}(\zeta_d)$$

where, the multi-indices $k = (k_1, k_2, ..., k_d)$.

3. B-spline functions

Univariate B-spline functions are defined in parametric space using a so-called vector denoted Ξ, in unit size (1D) is a set of m non-decreasing coordinates : $\Xi = \{\xi_1, \xi_2, ..., \xi_m\}$.
The univariate B-spline function $\mathcal{N}_{i,p}$ of degree p is defined according to the Coxde Boor recursion formula [2] :
for $p = 0$:

$$\mathcal{N}_{i,0}(\xi) = \begin{cases} 1 & if \quad \xi_i \leq \xi < \xi_{i+1} \qquad \forall i = 1, ..., m-1 \\ 0 & otherwise \end{cases} \qquad [1]$$

for $p \geq 1$:

$$\mathcal{N}_{i,p}(\xi) = \Big(\frac{\xi - \xi_i}{\xi_{i+p} - \xi_i}\Big)\mathcal{N}_{i,p-1}(\xi) + \Big(\frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}}\Big)\mathcal{N}_{i+1,p-1}(\xi) \qquad [2]$$

In order to define **multivariate B-splines functions in higher dimensions**, we make use of the tensor product.
Let $p = (p_1, p_2, ..., p_d)$ be a vector in N^d and let for all $j = 1, ..., d$, Ξ_j is a $1D$ knot vector defined by :

$$\Xi_j = \{\xi_1^j, \xi_2^j, ..., \xi_{n_1+p_1+1}^j\}$$

Furthermore, we denote the i_j univariate B-spline of degree p_j defined on the knot vector Ξ_j by $\mathcal{N}_{i_j,p_j}(\xi^j)$. Then, with the multi-indices $i = (i_1, i_2, ..., i_d)$, $p = (p_1, p_2, ..., p_d)$ and $n = (n_1, n_2, ..., n_d)$ the d-dimensional tensor product B-spline is defined by :

$$\mathcal{N}_{i,p}(\xi) = \mathcal{N}_{i_1,p_1}(\xi^1) \otimes \mathcal{N}_{i_2,p_2}(\xi^2) \otimes ... \otimes \mathcal{N}_{i_d,p_d}(\xi^d)$$

3.1. B-spline curves

Given n basis functions $\mathcal{N}_{i,p}$, $i = 1, ..., n$ and corresponding control points $P_i \in R$, $i = 1, ..., n$
thus a piecewise-polynomial B-spline curve is given as :

$$\mathcal{C}_p(\xi) = \sum_{i=1}^{n} \mathcal{N}_{i,p}(\xi) P_i$$

3.2. Extracting Bézier curves from B-splines

To decompose a set of B-spline basis functions to its Bézier elements, called Bézier decomposition, a straightforward approach consists in using the knot-insertion procedure p times, for each of the existing interior knots $(\xi_{p+2}, ..., \xi_n)$. Theoretically, the interior knots should have multiplicity of $(p+1)$ to form truly separated Bézier elements. By doing so, the multiplicity of p is sufficient to represent the Bernstein polynomials, which in this context are also referred to as Bézier basis functions. It is important to point out that the Bézier patch is a particular case of B-spline patch, for which the number n of functions (and control points) is equal to $p+1$.

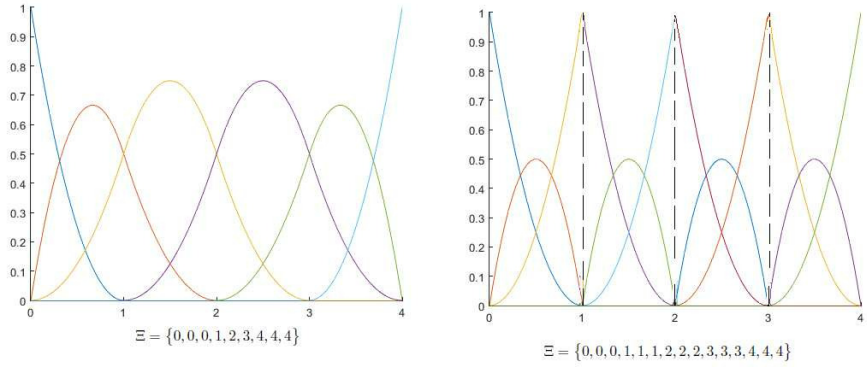

Figure 1. *Bézier decomposition (right) from a quadratic B-spline basis (lef) by knot insertion.*

4. Classical discontinuous Galerkin method

DG is a class of FEM using completely discontinuous basis functions. In contrast to the stabilized continuous Galerkin FEM, DG method produce stable discretizations without the need for stabilization parameters, due to their flexibility in local approximation they offer, together with their good stability properties [3] [4].
In the following, we describe the discretization of the advection problem by the classical DG method :

$$\begin{cases} \partial_t u(X,t) + \nabla.\left(\overrightarrow{c} u(X,t)\right) & = 0 & \forall (X,t) \in \Omega \times [0,T] \\ u(X,0) & = u_0(X) & \forall X \in \Omega \end{cases} \qquad [3]$$

where $u(X,t)$ is a scalar quantity transported by a continuous velocity field $\overrightarrow{c}$.
In the DG method, the domain Ω is subdivided into a union of finite number N_{el} of cells $\{D_k\}_{k=1}^{N_{el}}$, such that :

$$\Omega = \bigcup_{k=1}^{N_{el}} D_k \qquad with \quad D_k \bigcap D_l = \emptyset \quad \forall 1 \leq k \neq l \leq N_{el}$$

Thus, we denote by $\mathcal{T}$ a subdivision of Ω into N_{el} elements D_k.

$$\mathcal{T} = \Big\{ D_k, \quad 1 \leq k \leq N_{el} \Big\}$$

So on each cell D_k, the discrete unknown u_h^k is represented as a linear combination of well chosen basis functions of the space of polynomials of degree p. Then, the finite-dimensional subspace $\mathcal{V}_h^p$ is defined as :

$$\mathcal{V}_h^p = \Big\{ \mathbf{v} \in L^2(\Omega) \quad | \, \mathbf{v}_{|_{D_k}} \in P_p(D_k) \quad \forall 1 \leq k \leq N_{el}, \quad D_k \in \mathcal{T} \Big\}$$

where $P_p(D_k)$ represents the space of polynomials of degree up to p defined on the element D_k. By applying Greens formula and introducing the numerical flux $f^\star$ (in the present work, we use the local Lax- Friedrichs recipe), the weak formulation can be written as :
For each element $D_k \in \mathcal{T}$:

$$\int_{D_k} \frac{\partial u_h^k(X,t)}{\partial t} \mathbf{v}_h(X) dX = \int_{D_k} u_h^k(X,t) \vec{c}.\nabla \mathbf{v}_h(X) dX - \int_{\Gamma^k} u_h^k(X,t) \mathbf{v}_h(X) \vec{c}.\vec{n}^k d\Gamma^k \quad \forall t \in [0,T] \quad \forall \mathbf{v}_h \in \mathcal{V}_h^p \quad [4]$$

We denote $\overrightarrow{n}^k$ the outer unit normal to Γ^k of the element D_k.
Therefore, the local problem takes the form of a linear system, which can be written in the following matrix form :

$$M^k \partial_t u^k = \mathbf{R}^k(u^k) + f^\star(u^k) \qquad \forall t \in [0,T] \quad k = 2, ..., N_{el} - 1 \qquad [5]$$

Therefore, in the present work, a $RK2$ and $RK4$ are used for time integration. Because we are focusing on DG schemes, we discuss the limits for the C_{cfl} number when the DG method is used in conjunction with the RK time integration approach. An extra condition on the size of the timestep must also be satisfied, a Courant Friedrichs-Lewy (CFL) condition :

$$| \, c \, | \, \frac{\Delta t}{h_x} \leq \frac{1}{2p+1}$$

where $| \, c \, |$ is the largest wave speed, h_x is the smallest element width, Δt is the length of the time step and p is the degree of the approximating polynomial.

The L^2 error of the numerical approximations are depicted in Fig. (2) which indicate that the rates of convergence are of the type $O(h_x^{p+1})$.

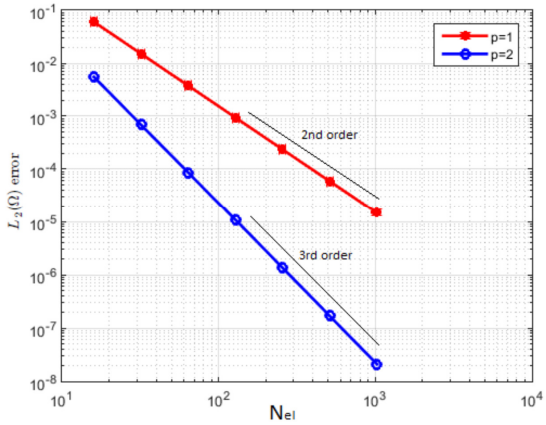

Figure 2. *$1D$ advection problem - L^2-errors from DGFE method in conjunction with RK method for a sinusoidal initial condition and Lax-Friedrichs flux.*

5. Isogeometric discontinuous Galerkin method

In this section, we present a method that combines isogeometric analysis (IGA) with the discontinuous Galerkin (DG) method for solving hyperbolic equations. The basis functions are continuous within each patch, and discontinuous only on patch boundaries. We also highlight that IGA space is local to patches rather than elements, in comparison with FEA. Therefore, the DG application in IGA is a patch to patch relation instead of an element to element. This fact is important to remember, since every time we mention about partitions in the domain, we are referring to patches that consist of elements. In order to apply the IGA methodology, the physical domain Ω is subdivided into patches Ω^e,

$$\mathcal{S}(\Omega) := \left\{\Omega^e\right\}_{e=1}^{N_{el}}$$

such that :

$$\overline{\Omega} = \bigcup_{e=1}^{N_{el}} \overline{\Omega}^e \quad with \quad \Omega^e \bigcap \Omega^l = \emptyset \quad \forall \quad 1 \leq e \neq l \leq N_{el}$$

Then, we define the test functions in the physical domain Ω^e such as :

$$\Phi^{p,q}(x,y)_{|\Omega^e} = \left(\Phi^{p,q}(x,y)\right)^e = \left(B^{p,q}(x,y)\right)^e = \left(B^{p,q}(T(\xi,\eta))\right)^e = \left(B^{p,q}(\xi,\eta)\right)^e$$
$$= \left(B^{p}(\xi)\right)^e \otimes \left(B^{q}(\eta)\right)^e = \left(\widetilde{\Phi}^{p}(\xi)\right)^e \otimes \left(\widetilde{\Phi}^{q}(\eta)\right)^e$$

where, T is the transformation of the parametric domain $\widetilde{\Omega}$ to the physical domain Ω :

$$T : \widetilde{\Omega} \longmapsto \Omega, \qquad (\xi,\eta) \longmapsto \big(x(\xi,\eta), y(\xi,\eta)\big)$$

Applying a IGDG method, the solution u is approximated by $u_h \in V^p$, we can postulate the following approximation to the solution :

$$u_h^e(x,y) = \sum_{i=1}^{p+1}\sum_{j=1}^{p+1} \left(B_i^p(\xi)\right)^e \left(B_j^p(\eta)\right)^e u_{ij}^e$$

where $u_{ij}^e : [0,T] \longmapsto R^2, \quad \forall \quad 1 \leq i,j \leq p+1$ are local unknown functions.
Therefore, the local problem takes the form of a linear system of size $(p+1)^2 \times (p+1)^2$, which can be written in the following matrix form :

$$\mathbf{M}^e \partial_t u^e = \mathbf{R}^e u^e + F^e \qquad \forall t \in [0,T] \quad \forall 1 \leq e \leq N_{el} \qquad [6]$$

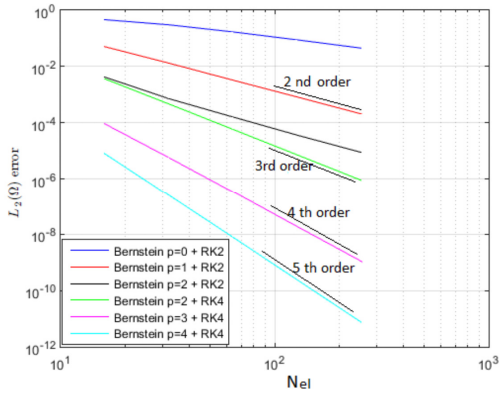

Figure 3. *$1D$ advection problem - L^2-errors for a sinusoidal initial condition, $RK2$ and $RK4$.*

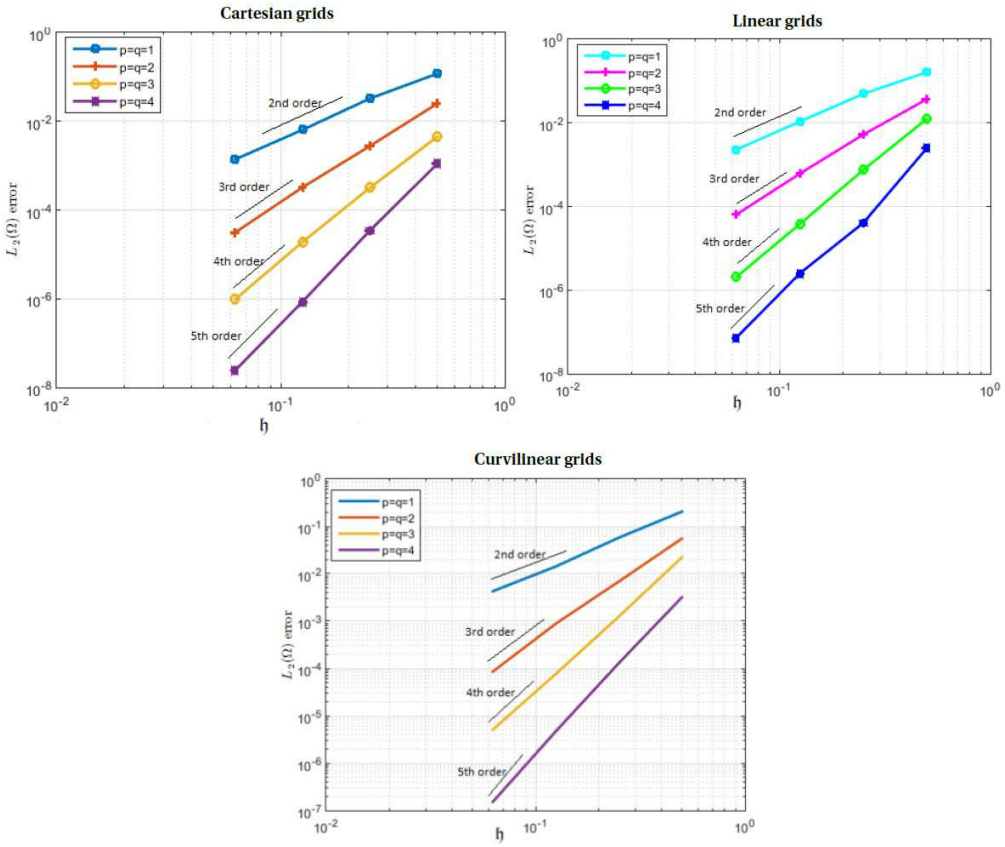

Figure 4. *$2D$ advection problem - L^2 errors for the IGDG method in conjunction with $RK4$ for different grids.*

An optimal convergence rate is found, the method being of order $p+1$ with respect to L^2-norm.

6. Conclusion

As mentioned before, the major reason for using DG methods lies with their ability to provide stable numerical methods for first order PDEs problems, for which classical FEM is well known to perform poorly. However, for geometric partitioning of the computational domain, the DG method uses standard disjoint finite element meshes, each element determines a single subproblem, the solution is calculated separately for each element of the computational mesh. The solution for the whole computational domain is achieved by summing over all the elements of the mesh. In this work a new family of discontinuous Galerkin methods which combines the IGA with the DG method, called IGDG method has been developed for the advection problem, our method takes advantage of both IGA and the DG method. In the fact, DG ideology is adopted at patch level, i.e., we employ the traditional IGA within each patch, and employ the DG method across the patch interfaces to glue the multiple patches. Obviously, due to IGA (NURBS), all conic sections can be represented exactly, thus eliminating the geometrical errors at the beginning.

7. Bibliographie

[1] *Hughes, Thomas JR and Cottrell, John A and Bazilevs, Yuri.* Isogeometric analysis : CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering.

[2] *De Boor, Carl.* On calculating with B-splines. Journal of Approximation Theory.

[3] *Shu, Chi-Wang.* Discontinuous Galerkin methods : general approach and stability. Numerical Solutions of Partial Differential Equations.

[4] *Xu, Qinwu and Hesthaven, Jan S.* Discontinuous Galerkin method for fractional convection-diffusion equations. SIAM Journal on Numerical Analysis.

[5] *Reed, William H and Hill, TR.* Triangular mesh methods for the neutron transport equation. Los Alamos Scientific Lab., N. Mex.(USA).