



Swansea University
Prifysgol Abertawe



Cronfa - Swansea University Open Access Repository

This is an author produced version of a paper published in :
International Journal for Numerical Methods in Fluids

Cronfa URL for this paper:

<http://cronfa.swan.ac.uk/Record/cronfa10751>

Paper:

Sevilla, R., FernándezMéndez, S. & Huerta, A. (in press). NURBSenhanced finite element method for Euler equations. *International Journal for Numerical Methods in Fluids*, 57(9), 1051-1069.

<http://dx.doi.org/10.1002/flid.1711>

This article is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence. Authors are personally responsible for adhering to publisher restrictions or conditions. When uploading content they are required to comply with their publisher agreement and the SHERPA RoMEO database to judge whether or not it is copyright safe to add this version of the paper to this repository.

<http://www.swansea.ac.uk/iss/researchsupport/cronfa-support/>

NURBS-Enhanced Finite Element Method for Euler equations

Ruben Sevilla, Sonia Fernández-Méndez, Antonio Huerta

Laboratori de Càlcul Numèric (www-lacan.upc.edu), Departament de Matemàtica Aplicada III, E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universitat Politècnica de Catalunya, Jordi Girona 1, E-08034 Barcelona, Spain.

Abstract

In this work the NURBS-Enhanced Finite Element Method (NEFEM) is combined with a Discontinuous Galerkin (DG) formulation for the numerical solution of Euler equations of gas dynamics. With NEFEM numerical fluxes along curved boundaries are computed much more accurately due to the exact geometric representation of the computational domain. The proper implementation of the wall boundary condition and the exact geometry provide accurate results even with a linear approximation of the solution. A detailed comparison of NEFEM in front of isoparametric finite elements is presented, demonstrating the superiority of NEFEM approach for both linear and higher order computations.

Keywords: NURBS, Discontinuous Galerkin, CAD, exact geometry representation, high-order isoparametric elements, Euler equations

1. INTRODUCTION

The importance of the geometrical model in Finite Element (FE) simulations has recently been pointed out by several authors, see [1, 2, 3, 4, 5, 6] to name a few. For instance, in [3] the error induced by the approximation of curvilinear geometries with isoparametric elements is analyzed in the context of Poisson and Maxwell problems. Using an exact mapping for the geometry

Email addresses: `ruben.sevilla@upc.edu` (Ruben Sevilla),
`sonia.fernandez@upc.edu` (Sonia Fernández-Méndez), `antonio.huerta@upc.edu`
(Antonio Huerta)

in the numerical solution of the Maxwell's equations the error is reduced by an order of magnitude.

When a DG formulation is adopted, the importance of the geometrical model is crucial in some applications, such as the numerical solution of Euler equations of gas dynamics. In [1] the authors demonstrate that using a linear approximation for the geometry it is not possible to converge to the steady state solution, even if the mesh is drastically refined near the curved boundary. In [2] a detailed study of this problem is presented to conclude that accurate results can only be obtained taking into account the curvature of the domain. More recently, in [5] a new methodology is presented for the computation of the fluxes across curved boundaries but, unfortunately, the proposed method is not conservative.

The importance of the geometrical model in the numerical solution of compressible Euler equations is not exclusive of DG methods. In [7, 8] the problem is identified in the context of Finite Volume (FV) methods, and more recent advances in this area can be found in [9, 10].

Non-Uniform Rational B-Splines (NURBS, see [11]) are widely used for geometry description in CAD (Computer Aided Design). This fact has motivated new numerical methodologies considering an exact representation of the computational domain with NURBS, such as the *isogeometric analysis* [4] and the NURBS-Enhanced Finite Element Method (NEFEM) [6]. The *isogeometric analysis* considers the same NURBS basis functions for both the description of the entire geometry and for the approximation of the solution. This idea was first introduced in [12] in the context of thin shell analysis. The NURBS-Enhanced Finite Element Method (NEFEM) also considers an exact representation of the domain but it differs from the *isogeometric analysis* in two main points: the geometry is given by the NURBS description of the boundary (i.e. the information usually provided by CAD), and standard FE polynomial interpolation is considered for the approximation of the solution. Thus, in the large majority of the domain—for elements not intersecting the boundary—a standard FE interpolation and numerical integration is used, preserving the computational efficiency of classical FE techniques. Specifically designed piecewise polynomial interpolation and numerical integration is only required for those FE along the NURBS boundary.

In [6] NEFEM is applied to the numerical solution of Poisson and electromagnetic scattering problems. In the numerical solution of Poisson problems with high-order isoparametric FE, the optimal rate for h -convergence is not achieved and, consequently, p -convergence is clearly deteriorated. In con-

trast, using NEFEM the optimal rate for h -convergence is obtained for any polynomial degree. Moreover, exponential convergence is observed when a p -refinement strategy is considered. In the context of electromagnetic scattering applications the use of NEFEM reveals an important improvement with respect to classical isoparametric FE. For the same spatial discretization an important reduction of the error is observed, in some cases more than one order of magnitude. In addition, for a desired precision NEFEM is also more efficient because it allows to compute the solution with an important reduction in number of degrees of freedom.

In this paper NEFEM is presented as a powerful method for numerical resolution of Euler equations using a DG formulation. Sections 2 and 3 recall the system of Euler equations and its DG discretization. The basic concepts of NEFEM are recalled in section 4, with special attention to the interpolation and numerical integration in those elements with one curved edge defined by NURBS. Section 5 presents a classical test for inviscid flow methods in order to evaluate the efficiency and accuracy of NEFEM in front of classical FE. Low and high-order approximations are tested, and a comparison between isoparametric FE and NEFEM in terms of the entropy error and other aerodynamic quantities of interest is presented, demonstrating the superiority of NEFEM approach for the simulation of compressible flow problems.

2. EULER EQUATIONS

Euler equations of gas dynamics express the conservation of mass, momentum and energy for a compressible, inviscid and non-conducting fluid. The strong form of these conservation laws, in the absence of external volume forces, can be written in conservative form as

$$\frac{d\mathbf{U}}{dt} + \frac{\partial \mathbf{F}_k(\mathbf{U})}{\partial x_k} = \mathbf{0}, \quad (1)$$

where Einstein notation is assumed (that is repeated indices are implicitly summed over), \mathbf{U} is the vector of conservation variables and $\mathbf{F}_k(\mathbf{U})$ are the flux vectors for each spatial dimension x_k , that is

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{pmatrix}, \quad \mathbf{F}_k(\mathbf{U}) = \begin{pmatrix} \rho v_k \\ \rho \mathbf{v} v_k + \mathbf{e}_k p \\ (\rho E + p) v_k \end{pmatrix},$$

where ρ is the density, $\rho\mathbf{v}$ is the momentum, ρE is the total energy per unit volume, \mathbf{e}_k is the unitary vector in the x_k direction, and p is the pressure, see [13] for more details.

An equation of state, relating the internal energy to pressure and density, completes this system of nonlinear hyperbolic equations. For a perfect polytropic gas the equation of state is

$$p = (\gamma - 1)\rho\left(E - \frac{1}{2}\|\mathbf{v}\|^2\right),$$

where γ is the ratio of the specific heat coefficients (specific heat at constant pressure over specific heat at constant volume), with value $\gamma = 1.4$ for air.

A usual quantity for postprocess of inviscid flow computations is the Mach number, defined as

$$M = \frac{\|\mathbf{v}\|}{c},$$

where $c = \sqrt{\gamma p/\rho}$ is the speed of sound. For a more detailed presentation of the Euler equations see for instance [14, 15, 16].

Other useful quantities for the evaluation of the accuracy are, the entropy error

$$\epsilon_{\text{ent}} = \frac{p}{p_\infty} \left(\frac{\rho_\infty}{\rho}\right)^\gamma - 1,$$

the pressure loss

$$p_{\text{loss}} = \frac{p}{p_\infty} \left(\frac{1 + 0.5(\gamma - 1)M^2}{1 + 0.5(\gamma - 1)M_\infty^2}\right)^{\frac{\gamma}{\gamma-1}},$$

and the pressure coefficient

$$C_p = \frac{p - p_\infty}{0.5\rho_\infty v_\infty^2},$$

where the subscript ∞ indicates free-stream values, see [2, 5] for more details.

3. DISCONTINUOUS GALERKIN FORMULATION

This section recalls the basic concepts of DG [17] for the solution of Euler equations in an open bounded domain $\Omega \subset \mathbb{R}^n$. A regular partition in

elements $\bar{\Omega} = \bigcup_e \bar{\Omega}_e$ is assumed, and an element by element discontinuous approximation space is considered.

The weak problem for the strong form of Euler equations (1) is stated for each element Ω_e . By multiplying by a test vector function \mathbf{W} , integrating over Ω_e and integrating by parts, the following equation is obtained

$$\int_{\Omega_e} \mathbf{W} \cdot \frac{d\mathbf{U}_e}{dt} d\Omega - \int_{\Omega_e} \frac{\partial \mathbf{W}}{\partial x_k} \cdot \mathbf{F}_k(\mathbf{U}_e) d\Omega + \int_{\partial\Omega_e} \mathbf{W} \cdot \mathbf{F}_{n_e}(\mathbf{U}_e) d\Gamma = 0 \quad \forall \mathbf{W},$$

where \mathbf{U}_e denotes the restriction of \mathbf{U} to element Ω_e , \mathbf{n}_e is the outward unit normal vector on $\partial\Omega_e$, and the normal flux is defined as

$$\mathbf{F}_n(\mathbf{U}) = \mathbf{F}_k(\mathbf{U})n_k,$$

with n_k the k -th component of \mathbf{n} . As usual in DG methods, to take into account the discontinuous nature of the approximation, the normal flux at the boundary of the element is replaced by a numerical flux, $\hat{\mathbf{F}}_{n_e}(\mathbf{U}_e, \mathbf{U}_e^{out})$, which is evaluated in terms of the solution in the current element Ω_e and the solution at neighboring elements

$$\mathbf{U}_e^{out}(\mathbf{x}) = \lim_{\varepsilon \rightarrow 0^+} \mathbf{U}(\mathbf{x} + \varepsilon \mathbf{n}_e) \quad \text{for } \mathbf{x} \in \partial\Omega_e. \quad (2)$$

The resulting DG weak formulation, to be discretized at each element Ω_e , is

$$\int_{\Omega_e} \mathbf{W} \cdot \frac{d\mathbf{U}_e}{dt} d\Omega - \int_{\Omega_e} \frac{\partial \mathbf{W}}{\partial x_k} \cdot \mathbf{F}_k(\mathbf{U}_e) d\Omega + \int_{\partial\Omega_e} \mathbf{W} \cdot \hat{\mathbf{F}}_{n_e}(\mathbf{U}_e, \mathbf{U}_e^{out}) d\Gamma = 0 \quad \forall \mathbf{W}. \quad (3)$$

Some conditions are required for the definition of a numerical flux function: it must be conservative, Lipschitz and verify some consistency conditions. Some popular flux functions for the numerical solution of the Euler equations are the exact Riemann solver, the Roe solver, the Lax-Friederichs solver or the Harten-Lax-van Leer (HLLC) solver, see [18].

Boundary conditions are implemented following the ideas in [19], initially developed in the context of FV methods. Fictitious elements are considered along the boundary, and the value of the solution is set to impose the boundary conditions through the numerical flux. As usual in the solution of Euler equations a characteristic analysis is performed at the boundary to decide the quantities to be prescribed, see for instance [14, 15, 16] or [20] for implementation details.

The DG formulation (3) is discretized in each element, leading to a system of ordinary differential equations

$$\mathbf{M} \frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = \mathbf{0}, \quad (4)$$

where \mathbf{U} is the vector of nodal values (or approximation coefficients in a more general case), \mathbf{M} is a block diagonal mass matrix and $\mathbf{R}(\mathbf{U})$ is the residual vector. As shown in the examples, the spatial discretization may be performed using isoparametric FE or NEFEM, see section 4. The ODE system (4) is advanced in time using the explicit third-order Total Variation Diminishing Runge-Kutta (TVD-RK) scheme presented in [21]. Nevertheless, it is worth noting that semi-implicit and implicit time integration schemes seem to be an efficient alternative for steady state computations, see [22, 23]. The stability condition for the TVD-RK method requires a CFL number $\leq 1/(2p+1)$, where p is the degree of the functional approximation, see [24].

The evaluation of the residual $\mathbf{R}(\mathbf{U})$, which involves the flux at the interior of the elements and their boundaries, can be carried out, as usual, with two non-equivalent options: a quadrature-free implementation or a full quadrature version, see [13]. With a quadrature-free implementation the flux at the integration points is interpolated in terms of the flux at nodal values, whereas with a full quadrature version fluxes are evaluated at integration points, in terms of the solution at each integration point. The use of a quadrature-free implementation leads to an important save in computational cost thanks to the use of elemental matrices, instead of a loop on integration points. Moreover, for triangles with straight sides (or tetrahedras with planar faces) these elemental matrices can be computed, using the jacobian, from matrices previously computed at the reference element, see [25], with an important reduction in computational time. However, numerical experiments reveal that a quadrature-free implementation for the Euler equations suffers from instability problems in the vicinity of stagnation points, see [26] for a detailed explanation. Thus, all computations presented here are obtained with a full quadrature implementation.

4. NEFEM FUNDAMENTALS

A domain $\Omega \subset \mathbb{R}^2$ is considered, whose boundary $\partial\Omega$, or a portion of its boundary, is defined by NURBS curves. A NURBS curve is a curve parametrized by a piecewise rational function, whose definition changes at



Figure 1: Physical domain with part of the boundary defined by a circular NURBS curve (left) and a valid triangulation for NEFEM (right)

the so called *breakpoints*, see [11] for a detailed definition. In fact, in practical applications CAD manipulators use *trimmed* NURBS, which are defined by a restriction of the NURBS parametrization to a subspace of the parametric space.

A triangularization of the domain $\bar{\Omega} = \bigcup_e \bar{\Omega}_e$ is also assumed, such that every triangle Ω_e has at most one side, Γ_e , on the NURBS boundary. Figure 1 shows a domain with part of the boundary described by a NURBS circular curve and a valid triangulation for NEFEM. As usual in mesh generation codes, it is also assumed that every curved boundary side belongs to a unique NURBS. That is, a side can not be defined by different NURBS curves, or equivalently, every trimmed NURBS is cut to pieces corresponding to the boundary sides. It is important to note that the breakpoints, which characterize the piecewise nature of NURBS, are independent of the mesh discretization. Thus, the NURBS parametrization can change its definition inside one side, that is breakpoints may belong to element sides and do not need to coincide with FE nodes. This is another major advantage compared to isogeometric analysis [4].

All elements not intersecting the NURBS boundary use the standard FE interpolation and numerical integration. The basis of NEFEM for an element with one side on the NURBS boundary is recalled next.

Let Ω_e be an element with two straight interior sides and one side defined by a trimmed NURBS, $\Gamma_e = \mathbf{C}([\lambda_1^e, \lambda_2^e])$. A *linear transformation* Ψ from local coordinates to physical coordinates is considered. The linear transformation Ψ^{-1} maps the element in physical space Ω_e to a curved element $I_e := \Psi^{-1}(\Omega_e)$, see Figure 2. Note that I_e plays the role of the reference element in standard FE, but here it depends on the definition of its NURBS

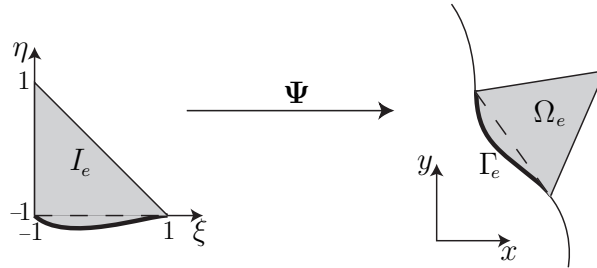


Figure 2: Linear transformation, mapping the curved element $I_e := \Psi^{-1}(\Omega_e)$ to the physic element Ω_e

side. Thus, special numerical strategies are required for every element I_e , see next sections.

Remark 1. *In order to simplify the presentation, it is assumed that the interior vertex of Ω_e is mapped to the vertex $(-1, 1)$ in I_e . The implementation of this condition only requires a proper local numbering of the vertices of the element.*

4.1. FE polynomial basis

In order to work with standard FE polynomial approximations, Lagrange polynomials (that is, standard nodal interpolation) can be considered. In fact, they can be defined on the curved triangle, I_e , in the reference domain or equivalently, in the actual element in the physical domain, Ω_e . The use of a linear transformation from the local (reference) coordinates $\boldsymbol{\xi} = (\xi, \eta)^T$ in I_e to the cartesian coordinates $\boldsymbol{x} = (x, y)^T$ in Ω_e , ensures that a complete polynomial interpolation of degree m in $\boldsymbol{\xi}$ leads to a polynomial interpolation with the same degree in \boldsymbol{x} . Thus, consistency and accuracy of the approximation is ensured even for elements Ω_e far from being a straight-sided element.

In order to make the computation of the Lagrange polynomials, $\{L_i(\boldsymbol{\xi})\}_{i=1}^{n_{en}}$, more systematic, for any order and for any distribution of nodes, the implementation proposed in [27] is adopted.

Different options can be considered to define nodal distributions in I_e . For low-order elements equally-spaced nodal distributions can be implemented. Nevertheless, high-order computations require special distributions of nodes in order to reduce the condition number of the resulting elemental matrices, see [28, 29] for details. Fekete points [30] are a good example of such distributions. For curved elements nodes can be located in the straight-sided triangle



Figure 3: 5th-order nodal distributions in I_e : Fekete points in the straight-side triangle (left) and adapted to the NURBS side (right)

given by the vertexes of I_e , see left distribution in Figure 3, or adapted to the exact geometry using the NURBS description, see right distribution in Figure 3. Adapted distributions have a positive influence on the condition number. Moreover, numerical experiments reveal that the non-adapted distribution lead to a more restrictive CFL stability condition, thus, in the numerical examples adapted distributions are used.

4.2. Numerical integration

The weak form to be solved requires integration along element sides and in the interiors. All the elements not having a side along the NURBS boundary are integrated using standard procedures. Elements with one side, Γ_e , on the NURBS boundary require special attention.

All line integrals to be computed in NEFEM can be written as

$$\int_{\Gamma_e} f \, dl = \int_{\lambda_1^e}^{\lambda_2^e} f(\mathbf{C}(\lambda_i)) |J_{\mathbf{C}}(\lambda)| \, d\lambda,$$

where f is a generic function (polynomial), the side of the element is given by a trimmed NURBS $\Gamma_e = \mathbf{C}([\lambda_1^e, \lambda_2^e])$, and $|J_{\mathbf{C}}|$ denotes the norm of the differential of the NURBS parametrization \mathbf{C} (this is not a polynomial). As usual, a 1D numerical quadrature is used for the numerical computation of the integral, namely

$$\int_{\Gamma_e} f \, dl \approx \sum_{i=1}^{n_{ip}} f(\mathbf{C}(\lambda_i)) |J_{\mathbf{C}}(\lambda_i)| \omega_i, \quad (5)$$

where λ_i and ω_i are, respectively, the coordinates and weights of the n_{ip} integration points in $[\lambda_1^e, \lambda_2^e]$. Recall that the parametrization of a trimmed

NURBS, \mathbf{C} , is a piecewise rational function whose definition changes at breakpoints. Thus, an independent numerical quadrature must be considered at every interval between breakpoints to account for the discontinuous nature of the parametrization. Numerical experiments reveal that Gauss-Legendre quadratures are a competitive choice in front of other quadrature rules such as trapezoidal and Simpson composite rules or Romberg's integration.

NEFEM also requires to compute integrals over an element Ω_e with one side Γ_e on the NURBS boundary, see Figure 2, that is

$$\int_{\Omega_e} f \, dx \, dy = |J_{\Psi}| \int_{I_e} f \, d\xi \, d\eta \quad (6)$$

where $|J_{\Psi}|$ is the determinant of the Jacobian of the linear transformation Ψ . The computation of (6) requires a numerical quadrature for every curved element I_e . Reference [31] presents different alternatives and discusses their advantages and disadvantages. The best alternative corresponds to the definition of a transformation from the rectangle $[\lambda_1^e, \lambda_2^e] \times [0, 1]$ to the curved element I_e , see Figure 4. That is, under the non restrictive assumption that nodes are numbered following Remark 1,

$$\begin{aligned} \varphi : [\lambda_1^e, \lambda_2^e] \times [0, 1] &\longrightarrow I_e \\ (\lambda, \zeta) &\longmapsto \begin{Bmatrix} \varphi_1 \\ \varphi_2 \end{Bmatrix} := \begin{Bmatrix} \phi_1(\lambda)(1 - \zeta) - \zeta \\ \phi_2(\lambda)(1 - \zeta) + \zeta \end{Bmatrix} \end{aligned} \quad (7)$$

where $\phi = (\phi_1, \phi_2)^T := \Psi^{-1} \circ \mathbf{C}$ is the parametrization of the trimmed NURBS corresponding to the curved side in I_e . Note that such a parametrization is linear in ζ ; this induces some important practical advantages.

Thus, using the transformations shown in Figure 4, integral (6) is computed as

$$\int_{\Omega_e} f \, dx \, dy = |J_{\Psi}| \int_{I_e} f \, d\xi \, d\eta \simeq |J_{\Psi}| \sum_{i=1}^{\mathbf{n}_{\text{ip}}} \sum_{j=1}^{\mathbf{m}_{\text{ip}}} f(\xi_{ij}) |J_{\varphi}(\lambda_i, \zeta_j)| \omega_i \varpi_j \quad (8)$$

where \mathbf{n}_{ip} and \mathbf{m}_{ip} are the number of integration points in the λ and ζ directions, respectively, $\xi_{ij} := \varphi(\lambda_i, \zeta_j)$, $\{\lambda_i, \omega_i\}$ and $\{\zeta_j, \varpi_j\}$ are the 1D quadrature points and weights for $[\lambda_1^e, \lambda_2^e]$ and $[0, 1]$ respectively, and $|J_{\varphi}|$ is the determinant of the Jacobian of the transformation φ .

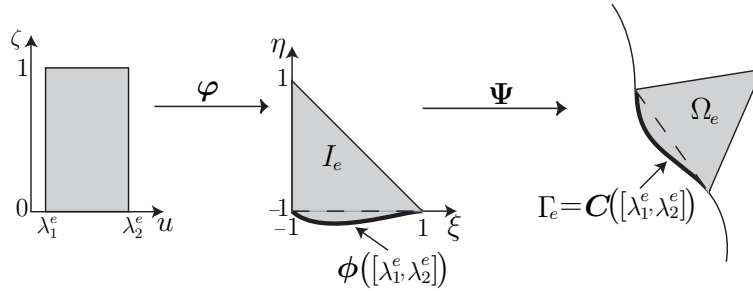


Figure 4: Transformation from $[\lambda_1^e, \lambda_2^e] \times [0, 1]$ to I_e and Ω_e

5. NUMERICAL RESULTS

To evaluate the accuracy and efficiency of the proposed methodology, a classical inviscid test case is considered: subsonic flow around a circle at free-stream Mach number $M_\infty = 0.3$. A DG formulation is adopted for every numerical computation. As first studied in [1] and later in [2, 5, 32], DG discretization of the wall boundary condition is very sensitive to the geometrical description of curved boundaries. More precisely, in [1] the authors show that it is not possible to converge to the correct physical solution if the computational boundary is approximated with piecewise linear polynomials.

The behavior of NEFEM for both linear and high-order approximations is studied and compared with isoparametric FE in the following sections. All computations are stopped when the density residual is reduced to 10^{-10} in the $\mathcal{L}_2(\Omega)$ norm. The approximate Roe solver is considered for the evaluation of the numerical flux. In fact, the Roe flux provides more accurate results than the Lax-Friederichs one for low order approximations, but no significant differences are observed for high-order approximations. However, it is very important to remark that the conclusions of the work derived from the comparison of NEFEM and FEM are exactly the same with a Lax-Friederichs flux.

5.1. Low order computations

Four O-meshes with 16×4 , 32×8 , 64×16 , and 128×32 nodes (i.e. 128, 512, 2048 and 8192 elements respectively) are considered for low-order computations. A detailed view of these meshes near the circle is represented in Figure 5, see [5] for mesh generation details.

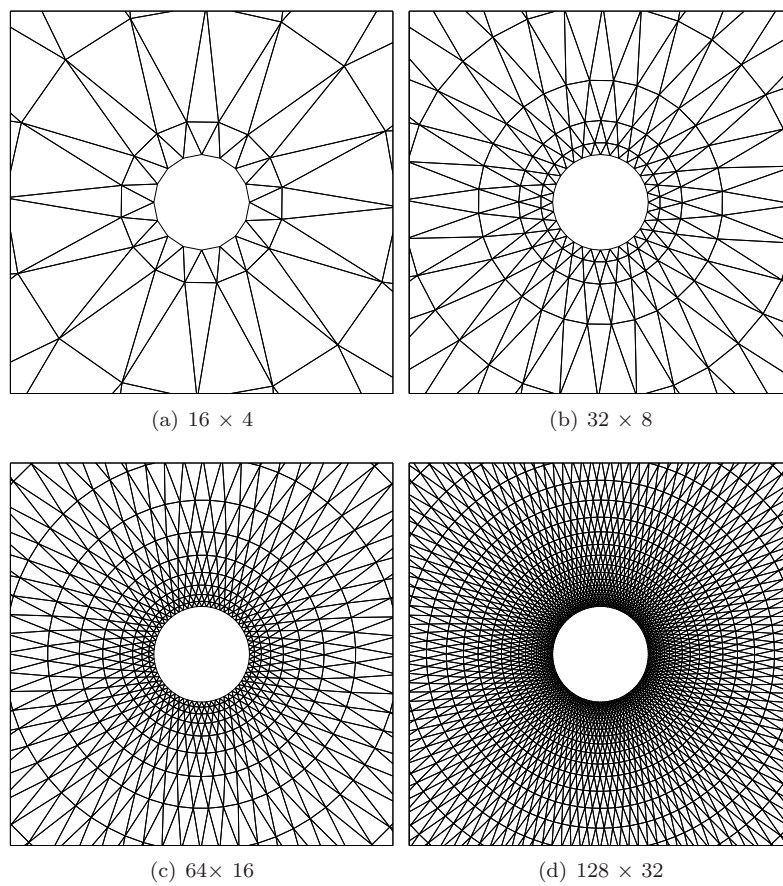


Figure 5: Detail of O-meshes for low-order computations

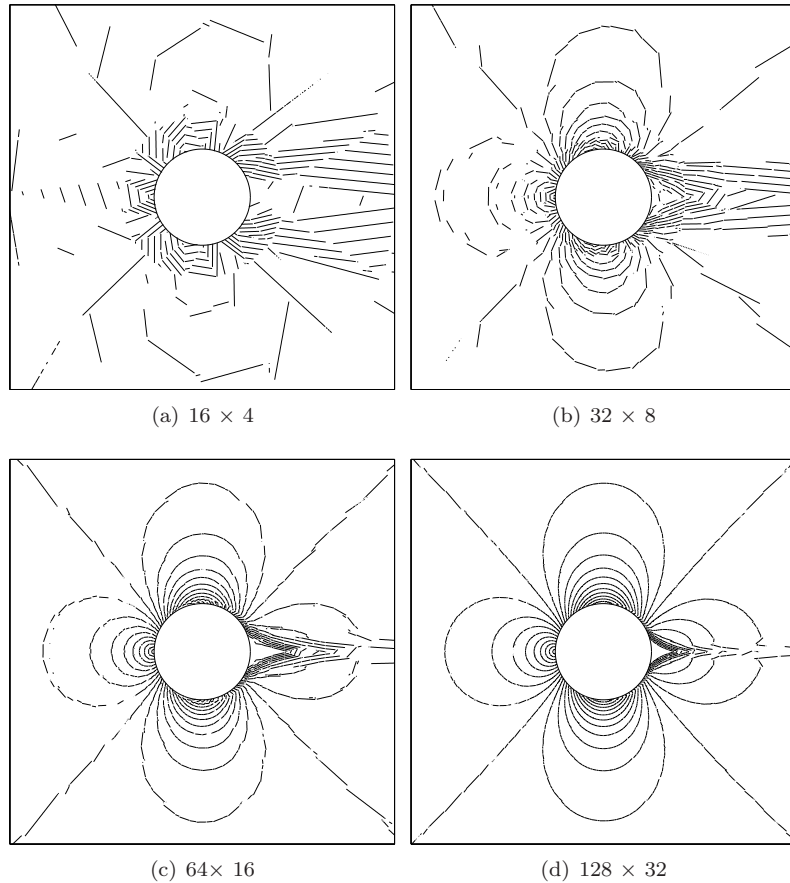


Figure 6: Mach number isolines with isoparametric FE and $p=1$

Figure 6 shows Mach number isolines for isoparametric FE with linear approximation. The results corroborate the conclusions first published by Bassi and Rebay [1] in the context of DG methods. Even if the mesh is highly refined near the circle, for instance using the fine mesh of figure 5 with 128 curved elements along the circular boundary, a non-physical entropy production is observed behind the wall. As it is commented in [33], the singularities of the polygonal approximation of the boundary generate entropy and the solution develops a non-physical wake that makes impossible the convergence to the correct solution.

Figure 7 shows Mach number isolines computed with NEFEM using lin-

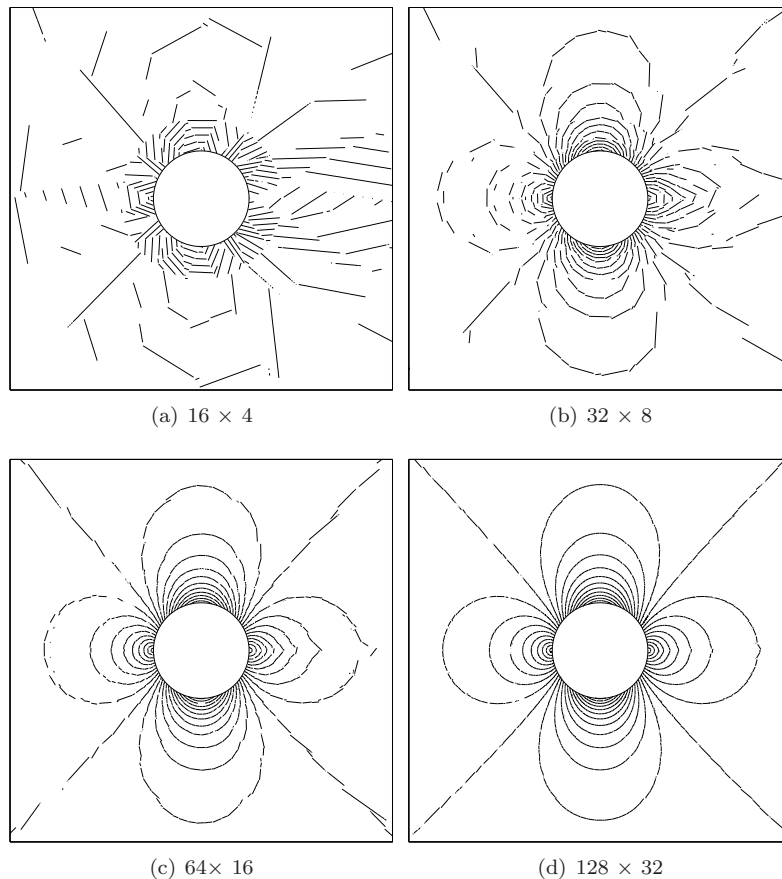


Figure 7: Mach number isolines with NEFEM and $p=1$

ear approximation. The results reveal a very good symmetry of the Mach number patterns, even if coarse meshes are used. Moreover, NEFEM allows convergence to the correct physical solution using the fine mesh with a piecewise linear approximation of the solution. The exact computation of the outward unit normal improves the imposition of the solid wall boundary condition. This issue and the exact representation of the domain drastically reduce the entropy production compared to isoparametric FE.

Entropy errors, in \mathcal{L}^2 norm, measured on the upper mid circle are reported in Table 1. For isoparametric FE, the entropy production observable in Figure 6 deteriorates the h -convergence rate. In contrast, NEFEM exhibits

Mesh	FEM		NEFEM	
	ϵ_{ent}	r	ϵ_{ent}	r
16×4	1.85E-02	-	1.18E-02	-
32×8	1.21E-02	0.61	3.07E-03	1.94
64×16	7.38E-03	0.71	6.72E-04	2.19
128×32	4.19E-03	0.82	1.53E-04	2.13

Table 1: \mathcal{L}^2 entropy error at the upper mid of the circle and convergence rate (r) for $p=1$

the optimal convergence rate for linear approximation. Results of Table 1 also show that, to achieve an entropy error of $4 \cdot 10^{-3}$ with isoparametric FE a fine mesh is necessary, whereas the second mesh suffices to obtain the same precision with NEFEM. Thus, the extra computational cost associated to the numerical integration of NEFEM is clearly surpassed by the drastic saving in number of degrees of freedom (ten times fewer degrees of freedom).

Figure 8 shows pressure loss and pressure coefficient distributions on the upper mid of the circle. At the most critical point, the stagnation point behind the circle, the maximum pressure loss error with isoparametric FE in the fine mesh is $1.8 \cdot 10^{-2}$, whereas NEFEM maximum error is reduced more than one order of magnitude, namely $8.4 \cdot 10^{-4}$. Moreover, in the fine mesh, the pressure coefficient error at the stagnation point is $2.8 \cdot 10^{-1}$ for standard FE and $4 \cdot 10^{-3}$ for NEFEM, almost two orders of magnitude more precise for the same number of degrees of freedom.

In [1] it is shown that the use of a good approximation of the normal on the solid boundary is crucial to accurately treat the DG boundary. In particular, the quality of the linear approximation is drastically improved using a quadratic approximation of the geometry. In fact, as noted by other authors [1, 5, 32, 33] using an accurate normal is crucial. If the exact normal is used with the linear approximation the results are drastically improved (similar improvement as with superparametric, i.e. quadratic, elements). Figure 9 shows Mach number isolines computed with linear approximation and exact normal at the boundary. Note the reduced entropy production behind the circle. However, although the global error (entropy error in the whole domain) is reduced using the exact normal due to the reduction in the entropy, the error measured on the circle surface is almost the same than using standard linear isoparametric elements. Moreover, it is important to note that NEFEM results are more accurate. Thus exact normal evaluation is not

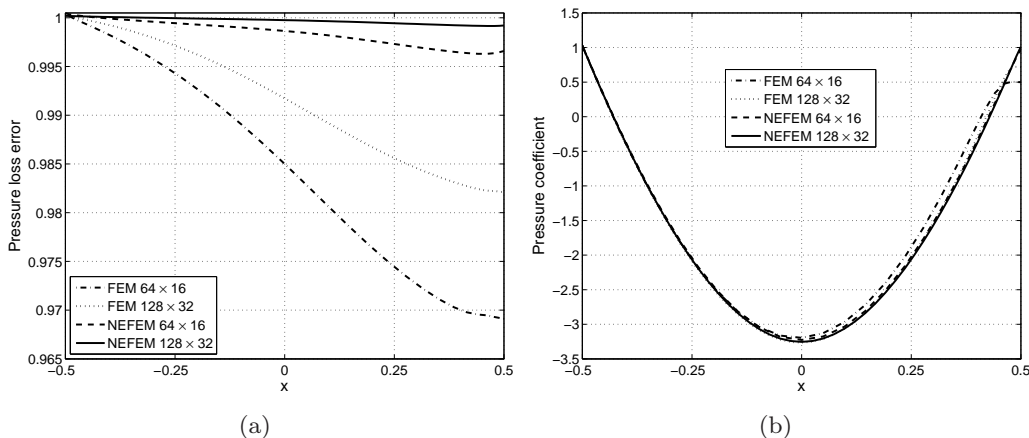


Figure 8: (a) Pressure loss distribution and (b) pressure coefficient distribution, at the upper mid of the circle for $p=1$

enough and an accurate geometrical approximation (exact approximation for NEFEM) is also important.

It is also important to note that improving the approximation of the normal (or superparametric representations) does not significantly improve the solution if high-order elements are employed [1]. Therefore this analysis will not be included in the next section for high-order elements.

To conclude, it is important to recall that the issue associated with solid wall boundary conditions is not specific to DG methods. In [8] the same problem had been observed by Barth in the FV framework. Thus, the benefits of the NURBS-Enhanced concept are extensible to FV methods.

Finally, a comparison between h and p -refinement strategies for NEFEM is presented in order to recall the superiority of high-order interpolations, see [34, 35]. For the p -refinement strategy the coarser mesh in Figure 5 is considered with an interpolation degree $p = 1, 2, 4$. Figure 10 shows the logarithm of the \mathcal{L}^2 entropy error in the upper mid of the circle as a function of the logarithm of the number of degrees of freedom. Although the h -refinement process shows the optimal rate of convergence (straight line with slope $p/2$), it is clearly surpassed by the exponential decay of the error for the p -refinement strategy. Thus, the advantage of using high-order interpolations is clear, specially in the context of a DG formulation. The performance of NEFEM for high-order computations is studied next.

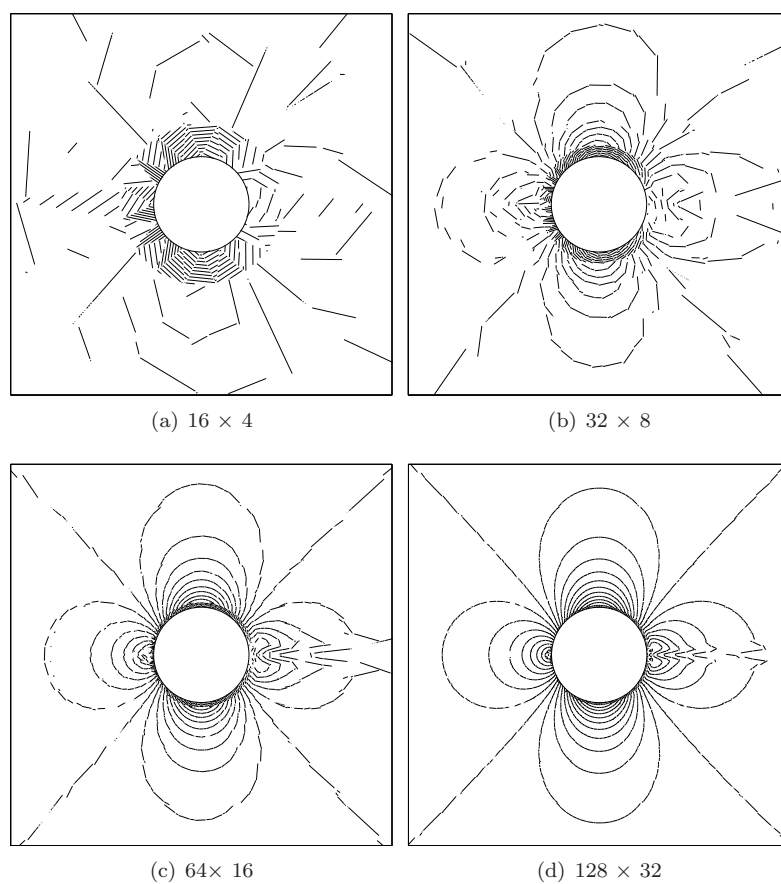


Figure 9: Mach number isolines using the exact normal for flux computations and $p=1$

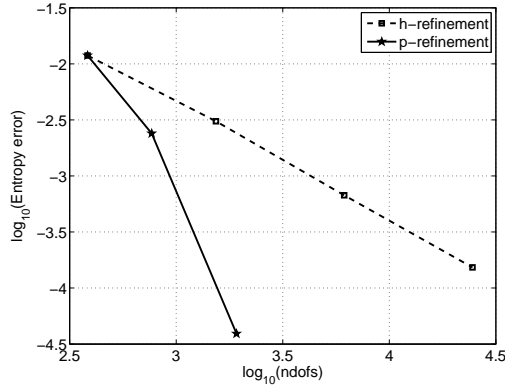


Figure 10: \mathcal{L}^2 entropy error at the upper mid of the circle for h and p -refinement using NEFEM

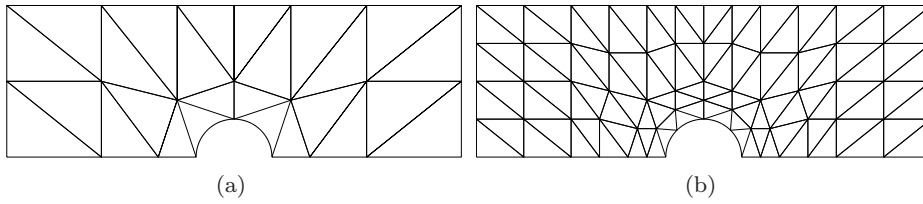


Figure 11: Computational meshes for high-order computations

5.2. High-order computations

Following [33], for high-order computations outer boundaries are placed near the obstacle, and the symmetry with respect to the x axis is used in order to reduce the number of degrees of freedom. Two computational meshes, represented in Figure 11, are considered. The coarser mesh has 26 elements and only two curved elements to describe a half of a circle, and the fine mesh is obtained by uniform refinement. A new measure is proposed in order to quantify the so-called *error in symmetry* and compare better different order or methods. Namely, the difference between the Mach number on the right part of the circle ($x > 0$) and on the left part ($x < 0$) in the \mathcal{L}^2 norm.

Figures 12 and 13 show Mach number isolines in the coarser mesh of Figure 11 using isoparametric FE and NEFEM, respectively. For both methods, a degree of $p=2$ or $p=4$ is not sufficient to properly capture the solution, and the entropy production behind the circle is clearly observed. For higher de-

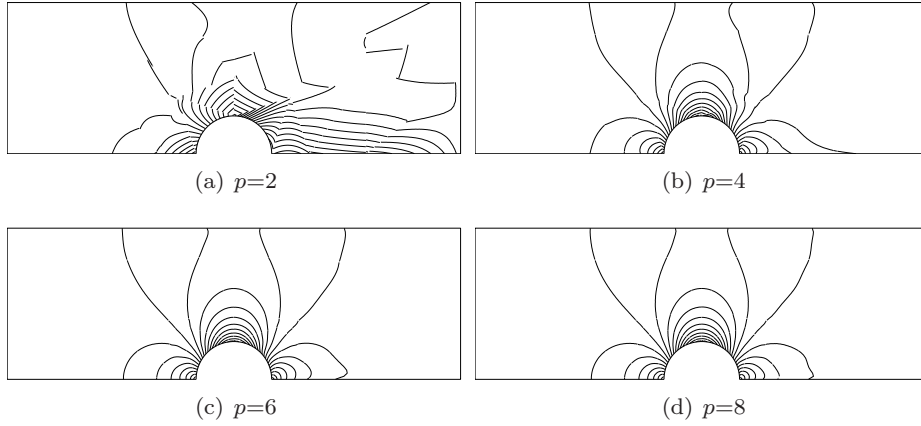


Figure 12: Mach number isolines for isoparametric FE in the coarse mesh

grees, namely for $p=6$ and $p=8$, the isoparametric FE solution shows a slight asymmetry with respect to the y axis due to the spurious entropy production. The error in symmetry is $3.7 \cdot 10^{-3}$ for $p=6$ and $3.1 \cdot 10^{-3}$ for $p=8$. NEFEM with $p=6$ exhibits better symmetry than isoparametric FE with $p=8$. The error in symmetry with NEFEM is $7.4 \cdot 10^{-4}$ for $p=6$ and $2.8 \cdot 10^{-4}$ for $p=8$.

Figures 14 and 15 show Mach number isolines for the fine mesh in Figure 11 using isoparametric FE and NEFEM, respectively. Quadratic elements, for both methods and this mesh, do not properly capture the solution. As the degree of the approximation increases results improve. However, the improvement is clearly faster with NEFEM compared to isoparametric FE. Isoparametric FE induced an error in symmetry of $2.5 \cdot 10^{-4}$ for $p=4$ and $5 \cdot 10^{-5}$ for $p=6$. Whereas, NEFEM has an error in symmetry of $5.5 \cdot 10^{-5}$ for $p=4$ and $5.2 \cdot 10^{-6}$ for $p=6$. Thus NEFEM has a comparable accuracy with $p=4$ as isoparametric FE with $p=6$. Thus, the computational overhead of NEFEM becomes negligible compared the the drastic reduction in degrees of freedom.

The accuracy in terms of the entropy error for high-order computations is analyzed next. Figure 16 compares isoparametric FE and NEFEM for the entropy error (in \mathcal{L}^2 norm and at the upper mid of the circle) as a function of the square root of the number of degrees of freedom. For isoparametric FE computations p -convergence is clearly deteriorated for high-order approximations, whereas NEFEM maintains the exponential p -convergence. The results

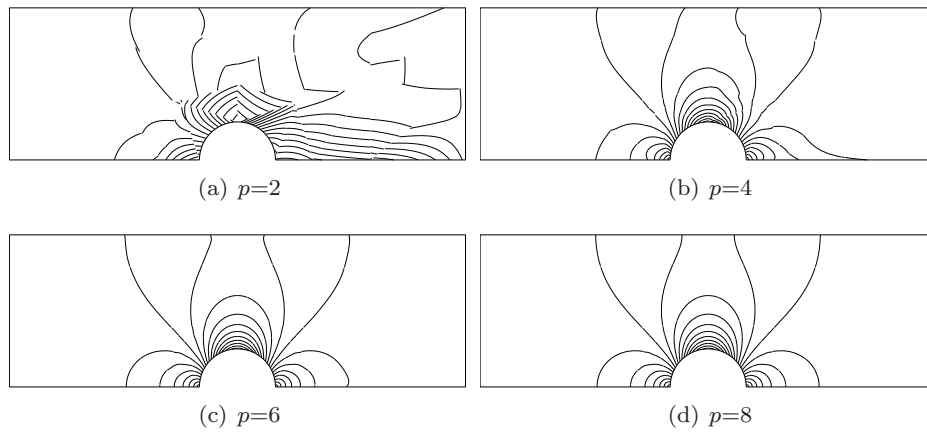


Figure 13: Mach number isolines for NEFEM in the coarse mesh

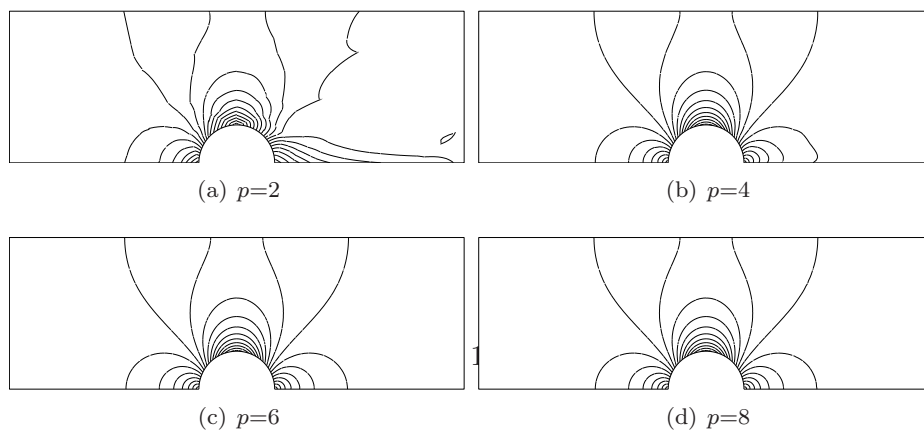


Figure 14: Mach number isolines for isoparametric FE in the fine mesh

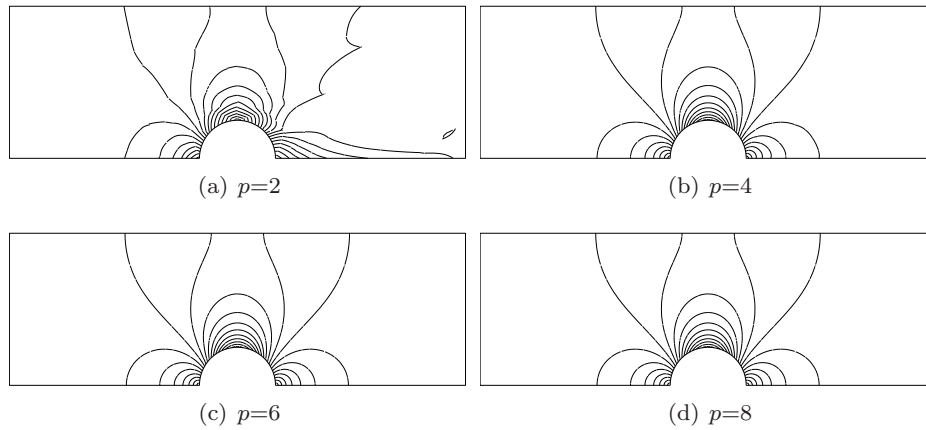


Figure 15: Mach number isolines for NEFEM in the fine mesh

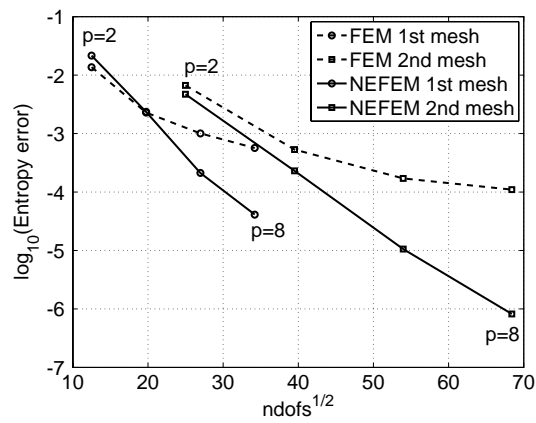


Figure 16: Entropy error on the circle surface for p -refinement

reveal that the best accuracy that can be obtained with isoparametric FE on the coarser mesh corresponds to an error of $5 \cdot 10^{-4}$. In order to improve the accuracy of the isoparametric FE computation, an hp -refinement strategy must be adopted. For instance, an error below 10^{-4} can be achieved with the second mesh shown in Figure 11. Note that for NEFEM with the coarse mesh the error with $p=8$ is already $4 \cdot 10^{-5}$, more than one order of magnitude more precise than isoparametric FE with the same number of degrees of freedom. Moreover, if the second mesh is considered, the NEFEM error with $p=8$ is 10^{-6} , two orders of magnitude more precise than standard isoparametric FE. It is also worth noting that with an approximation of degree $p=8$, similar accuracy is obtained using isoparametric FE in the fine mesh and NEFEM in the coarse one; that is, NEFEM requires four times fewer degrees of freedom.

Finally, it is important to recall that in a FE adaptive process, see [36], the computational mesh must be locally refined to properly capture both the solution and the geometry, whereas in a NEFEM context the adaptive process is controlled only by the complexity of the solution, independently of the geometrical complexity of the domain, and therefore reducing the necessary number of degrees of freedom to achieve a desired accuracy.

6. CONCLUDING REMARKS

NEFEM combined with a DG formulation is proposed for the numerical solution of compressible Euler equations. A classical test for inviscid flow solvers is considered to evaluate the accuracy and efficiency of the proposed methodology in front of isoparametric FE. The proper imposition of the wall boundary condition in NEFEM allows accurate computations even with a linear approximation of the solution. With isoparametric linear elements it is not possible to converge to the correct solution using linear approximation, even if the mesh is drastically refined near curved boundaries. Moreover, under p -refinement, exponential convergence is achieved with NEFEM and not for isoparametric FE.

Numerical results demonstrate that NEFEM is a powerful method for the solution of Euler equations of gas dynamics, more efficient than classical isoparametric FE. The extra cost of NEFEM, due to the numerical integration over elements along the NURBS boundary, is surpassed by the important saving in number of degrees of freedom. More precisely, NEFEM provides similar accuracy than isoparametric FE using between four and ten times fewer degrees of freedom.

References

- [1] F. Bassi and S. Rebay, “High-order accurate discontinuous finite element solution of the 2D Euler equations,” *J. Comput. Phys.*, vol. 138, no. 2, pp. 251–285, 1997.
- [2] 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. Eng.*, vol. 191, no. 41-42, pp. 4747–4780, 2002.
- [3] D. Xue and L. Demkowicz, “Control of geometry induced error in hp finite element (FE) simulations. I. Evaluation of FE error for curvilinear geometries,” *Int. J. Numer. Anal. Model.*, vol. 2, pp. 283–300, 2005.
- [4] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement,” *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 39–41, pp. 4135–4195, 2005.
- [5] L. Krivodonova and M. Berger, “High-order accurate implementation of solid wall boundary conditions in curved geometries,” *J. Comput. Phys.*, vol. 211, no. 2, pp. 492–512, 2006.
- [6] R. Sevilla, S. Fernández-Méndez, and A. Huerta, “NURBS-Enhanced Finite Element Method (NEFEM),” 2007. Submitted.
- [7] A. Dadone and B. Grossman, “Surface boundary conditions for the numerical solution of the Euler equations.,” *AIAA Journal*, vol. 32, no. 2, pp. 285–293, 1994.
- [8] T. Barth, *Simplified numerical methods for gas dynamics systems on triangulated domains*. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, 1998.
- [9] Z. J. Wang and Y. Sun, “A curvature-based wall boundary condition for the Euler equations on unstructured grids,” in *Proceedings of the 40th AIAA Aerospace Sciences Meeting and Exhibit*, (Nevada), AIAA, 2002.
- [10] Z. J. Wang and Y. Liu, “Extension of the spectral volume method to high-order boundary representation,” *J. Comput. Phys.*, vol. 211, pp. 154–178, 2006.

- [11] L. Piegl and W. Tiller, *The NURBS Book*. London: Springer-Verlag, 1995.
- [12] F. Cirak, M. Ortiz, and P. Schröder, “Subdivision surfaces: A new paradigm for thin-shell finite-element analysis,” *Int. J. Numer. Meth. Engrg.*, vol. 47, no. 12, pp. 2039–2072, 2000.
- [13] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*. Chichester: John Wiley & Sons, 2002.
- [14] J. D. Anderson, *Modern Compressible Flow: with Historical Perspective*. New York: McGraw-Hill, 1982.
- [15] C. Hirsch, *Numerical Computation of Internal and External Flows. Volume 2: Computational methods for inviscid and viscous flows*. Chichester: John Wiley & Sons, 1988.
- [16] C. B. Laney, *Computational gasdynamics*. Cambridge: Cambridge University Press, 1998.
- [17] B. Cockburn, *Discontinuous Galerkin methods for Computational Fluid Dynamics*, vol. 3 of *Encyclopedia of Computational Mechanics*, ch. 4. Wiley, 2004.
- [18] E. F. Toro, *Riemann solvers and numerical methods for fluid dynamics*. Berlin: Springer, second ed., 1997.
- [19] R. Leveque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge: Cambridge University Press, 2002.
- [20] D. Kuzmin and M. Möller, “Algebraic flux correction II. Compressible Euler equations,” in *Flux-corrected Transport: Principles, Algorithms, And Applications*, Springer Verlag, 2005.
- [21] B. Cockburn and C. W. Shu, “TVB Runge-Kutta local projection Discontinuous Galerkin finite-element method for conservation-laws. II. General framework,” *Math. Comp.*, vol. 52, no. 186, pp. 411–435, 1989.
- [22] V. Dolejší and M. Feistauer, “A semi-implicit Discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow,” *J. Comput. Phys.*, vol. 198, no. 2, pp. 727–746, 2004.

- [23] P. Persson and J. Peraire, “An efficient low memory implicit DG algorithm for time dependent problems,” in *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, (Nevada), AIAA, 2006.
- [24] S. Gottlieb and C. W. Shu, “Total variation diminishing Runge-Kutta schemes,” *Math. Comp.*, vol. 67, no. 221, pp. 73–85, 1998.
- [25] H. L. Atkins and C. W. Shu, “Quadrature-free implementation of Discontinuous Galerkin method for hyperbolic equations,” *AIAA Journal*, vol. 36, no. 5, pp. 775–782, 1998.
- [26] K. Hillewaert, N. Chevaugéon, P. Geuzaine, and J. Remacle, “Hierarchic multigrid iteration strategy for the Discontinuous Galerkin solution of the steady Euler equations,” *Internat. J. Numer. Methods Fluids*, vol. 51, pp. 1157–1176, 2006.
- [27] J. S. Hesthaven and T. Warburton, “Nodal high-order methods on unstructured grids. I. Time-domain solution of Maxwell’s equations,” *J. Comput. Phys.*, vol. 181, no. 1, pp. 186–221, 2002.
- [28] Q. Chen and I. Babuška, “Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle,” *Comput. Methods Appl. Mech. Eng.*, vol. 128, no. 3–4, pp. 405–417, 1995.
- [29] Q. Chen and I. Babuška, “The optimal symmetrical points for polynomial interpolation of real functions in the tetrahedron,” *Comput. Methods Appl. Mech. Eng.*, vol. 137, pp. 89–94, 1996.
- [30] M. A. Taylor, B. A. Wingate, and R. E. Vincent, “An algorithm for computing Fekete points in the triangle,” *SIAM J. Numer. Anal.*, vol. 38, no. 5, pp. 1707–1720, 2000.
- [31] R. Sevilla and S. Fernández-Méndez, “Numerical integration for the 2D NURBS-Enhanced Finite Element Method,” 2007. Submitted.
- [32] H. Luo, J. D. Baum, and R. Löhner, “A fast, p-multigrid Discontinuous Galerkin method for compressible flows at all speeds,” in *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, (Nevada), AIAA, 2006.

- [33] B. Cockburn, G. E. Karniadakis, and C. W. Shu, “The development of Discontinuous Galerkin methods,” in *Discontinuous Galerkin Methods* (B. Cockburn, G. E. Karniadakis, and C.-W. Shu, eds.), vol. 11 of *Lecture Notes in Computational Science and Engineering*, pp. 3–50, Berlin: Springer-Verlag, 2000.
- [34] B. Szabó and I. Babuška, *Finite Element Analysis*. New York: John Wiley & Sons, 1991.
- [35] B. Szabó, A. Düster, and E. Rank, *The p-version of the Finite Element Method*, vol. 1 of *Encyclopedia of Computational Mechanics*, ch. 5. Wiley, 2004.
- [36] A. Huerta, A. Rodríguez-Ferran, P. Díez, and J. Sarrate, “Adaptive finite element strategies based on error assessment,” *Int. J. Numer. Meth. Engrg.*, vol. 46, pp. 1803–1818, 1999.