INTERNATIONAL CENTER FOR NUMERICAL METHODS IN ENGINEERING

NURBS-Enhanced Finite Element Methods (NEFEM)

R. Sevilla
S. Fernández-Méndez
A. Huerta

Publication CIMNE Nº-308, Juny 2007

NURBS-Enhanced Finite Element Methods (NEFEM)

R. Sevilla
S. Fernández-Méndez
A. Huerta

Publication CIMNE Nº-308, Juny 2007

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, Spain

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2000; **00**:1–6 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

NURBS-Enhanced Finite Element Method (NEFEM)

Ruben Sevilla, Sonia Fernández-Méndez and Antonio Huerta*

Laboratori de Càlcul Numèric (LaCàN), 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
e-mail: {ruben.sevilla,sonia.fernandez,antonio.huerta}@upc.es, web http://www-lacan.upc.es

SUMMARY

An improvement of the classical finite element method is proposed. It is able to exactly represent the geometry by means of the usual CAD description of the boundary with Non-Uniform Rational B-Splines (NURBS). Here, the two-dimensional case is presented. For elements not intersecting the boundary, a standard finite element interpolation and numerical integration is used. But elements intersecting the NURBS boundary need a specifically designed piecewise polynomial interpolation and numerical integration. A priori error estimates are also presented. Finally, some examples demonstrate the applicability and benefits of the proposed methodology. NEFEM is at least one order of magnitude more precise than the corresponding isoparametric finite element in every numerical example shown. This is the case for both continuous and discontinuous Galerkin formulations. Moreover, for a desired precision NEFEM is also more computational efficient, as shown in the numerical examples. The use of NEFEM is strongly recommended in the presence of curved boundaries and/or when the boundary of the domain has complex geometric details. The possibility of computing accurate solution with coarse meshes and high order interpolations, makes NEFEM a more efficient strategy than classical isoparametric FE. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: NURBS; Finite Elements; CAD; Discontinuous Galerkin; exact geometry representation; high-order isoparametric finite elements; transient Maxwell equations

1. INTRODUCTION

The relevance of an accurate representation of the domain and its boundary has been pointed out by several authors, see [1, 2, 3, 4, 5, 6] among others. In some applications, such as compressible flow problems, if a Discontinuous Galerkin (DG) formulation is adopted, see [7], an important loss of accuracy is observed when a linear approximation of the boundary

*Correspondence to: Laboratori de Càlcul Numèric (LaCàN), E.T.S. Ingenieros de Caminos, Universitat Politècnica de Catalunya, Jordi Girona 1, E-08034 Barcelona, Spain.

Contract/grant sponsor: Ministerio de Ciencia y Tecnología; contract/grant number: BIA2007-66965 and DPI2007-62395

Received September 2007

is used, see [1, 2]. Bassi and Rebay [1] show that, in the presence of curved boundaries, a meaningful high-order accurate solution can only be obtained if the corresponding high-order approximation of the geometry is employed (i.e. isoparametric finite elements). In fact, it is necessary to take into account the boundary curvature effect in order to have a consistent boundary discretization, see [5]. In [6] the same problem is studied, and a new method is proposed for computing the flux across a curved face. Using a parametrization of the curved boundary the flux definition is modified but the resulting method is, unfortunately, non-conservative. The importance of the geometrical model in the numerical solution of the compressible Euler equations is not exclusive of DG methods. In [8, 2] 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].

An accurate representation of the geometry is not a prerogative of fluid mechanics. For instance, similar conclusions are derived in [3] for linear elasticity problems: sizable errors are present in the numerical solution when the order for the geometric approximation is lower than the order of functional interpolation, even for geometries as simple as a sphere. Isoparametric finite elements (FE) or superparametric FE are necessary in order to ensure an accurate enough representation of the geometry.

Obviously, Maxwell equations are also very sensitive to the quality of the boundary representation. Reference [4] studies the error induced by the approximation of curvilinear geometries with isoparametric elements. The 3D Maxwell equations are solved in a sphere with isoparametric FE and with an exact mapping of the geometry. The exact mapping provides more accurate results with errors differing by an order in magnitud. Thus, in some applications, an isoparametric representation of the geometry is far from providing an optimal numerical solution for a given FE discretization.

Recently, [11] proposes a new methodology: the *isogeometric analysis*. Its goal is to consider an exact representation of the geometry, with no dependency on the spatial discretization. Note that in the *isogeometric analysis* the solution of the boundary value problem is also approximated with the same NURBS (*Non-Uniform Rational B-Splines*, [12]) basis used for the description of the geometry.

The methodology proposed in this paper has a similar goal: an exact representation of the geometry, but it is simpler because NURBS are restricted to the boundary of the computational domain. Only the boundary of the computational domain is directly related to a CAD. Thus, *NURBS-Enhanced Finite Element Method* (NEFEM) considers the exact NURBS description for the boundary of the computational domain while the solution is approximated with a standard piecewise polynomial interpolation. Moreover, in the large majority of the domain (namely in the interior, that is for elements not intersecting the boundary) a standard FE interpolation and numerical integration is used, preserving the computational efficiency of classical FE techniques.

The use of a piecewise polynomial approximation represents an important advantage in front of the NURBS functional approximation used in the *isogeometric analysis*. NEFEM ensures local reproducibility of polynomials and, therefore, it preserves the classical FE convergence properties and allows a seamless coupling with the standard FE of the domain interior.

Section 2 introduces the basic concepts on NURBS. In section 3 the fundamentals of NEFEM are presented. Special attention is paid to the interpolation and numerical integration in those elements affected by the NURBS description on the boundary. To simplify the presentation and show its capabilities NEFEM is presented for 2D domains. The generalization to 3D domains

is conceptually easy but requires some extra attention to geometrical aspects and it will be the scope of a forthcoming publication. Some comments on the implementation of NEFEM are given and a priori error estimates for are also presented in section 3. Numerical examples are discussed in section 4. NEFEM can be implemented with a standard piecewise continuous (standard FE) or discontinuous (DG) formulation. Thus, a Poisson problem is solved first in a continuous framework and several electromagnetic scattering problems are solved using DG.

2. BASIC CONCEPTS ON NURBS

A qth-degree NURBS curve [12] is a piecewise rational function defined in parametric form as

$$\boldsymbol{C}(\lambda) = \left(\sum_{i=0}^{q} \nu_i \, \boldsymbol{B}_i \, C_{i,q}(\lambda)\right) \Big/ \left(\sum_{i=0}^{q} \nu_i \, C_{i,q}(\lambda)\right) \qquad 0 \le \lambda \le 1, \tag{1}$$

where $\{\boldsymbol{B}_i\}$ are the coordinates of the *control points* (forming the *control polygon*), $\{\nu_i\}$ are the control weights, and the $\{C_{i,q}(\lambda)\}$ are the normalized B-spline basis functions of degree q, which are defined recursively as:

$$C_{i,0}(\lambda) = \begin{cases} 1 & \text{if } \lambda \in [\lambda_i, \lambda_{i+1}[, \\ 0 & \text{elsewhere,} \end{cases}$$

$$C_{i,k}(\lambda) = \frac{\lambda - \lambda_i}{\lambda_{i+k} - \lambda_i} C_{i,k-1}(\lambda) + \frac{\lambda_{i+k+1} - \lambda}{\lambda_{i+k+1} - \lambda_{i+1}} C_{i+1,k-1}(\lambda),$$

for $k = 1 \dots q$, where λ_i, for $i = 0, \dots, m$, are the *knots* or *breakpoints*, which are assumed ordered $0 \le \lambda_i \le \lambda_{i+1} \le 1$. They form the so-called *knot vector*:

$$\Lambda = \{\underbrace{0, \dots, 0}_{q+1}, \lambda_{q+1}, \dots, \lambda_{m-q-1}, \underbrace{1, \dots, 1}_{q+1}\},$$

which uniquely describes the B-spline basis functions. The multiplicity of a knot, when it is larger than one, determines the decrease in the number of continuous derivatives. The number of control points, $\mathtt{n_{cp}}$, and knots, $\mathtt{n_k}$, is related to the degree of the parametrization, q, by the relation $\mathtt{n_k} = \mathtt{n_{cp}} + q + 1$, see [12] for more details. Figure 1 shows the B-spline basis functions for the knot vector

$$\Lambda = \{0, 0, 0, 0.2, 0.4, 0.6, 0.8, 0.8, 1, 1, 1\}. \tag{2}$$

Note that NURBS are piecewise (rational) functions and their definition changes at knots.

An example of a NURBS curve is represented in Figure 2 with the corresponding control polygon. The image of the breakpoints or knots by the NURBS are depicted in order to stress the discontinuous definition of the parametrization.

The *parametric space* for the NURBS where λ belongs, see (1), is chosen to be $[0, 1]$ for simplicity. However, in practice CAD manipulators work with *trimmed* NURBS, which are defined as the initial parametrization restricted to a subspace of the parametric space. Figure 3 shows the NURBS curve of Figure 2 trimmed to the subinterval $[0.05, 0.75]$.

Prepared using nmeauth.cls

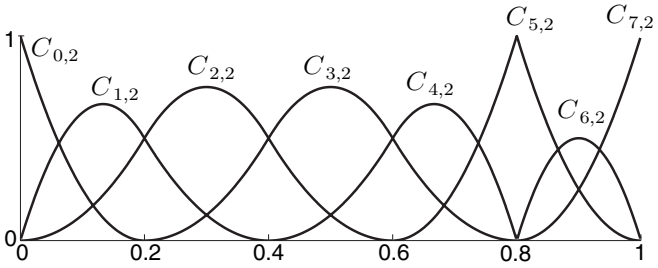


Figure 1. B-spline basis functions for the knot vector (2)

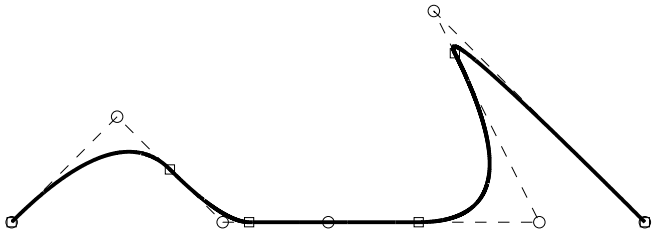


Figure 2. NURBS curve (solid line), control points (denoted by ∘), control polygon (dashed line) and breakpoints (denoted by □)

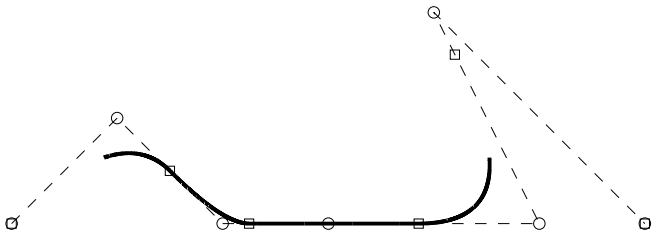


Figure 3. Trimmed NURBS curve with $\lambda \in [0.05, 0.75]$ (solid line), control points (denoted by ∘), control polygon (dashed line) and breakpoints (denoted by □)

3. NURBS-ENHANCED FINITE ELEMENT METHOD (NEFEM)

Consider a physical domain $\Omega \subset \mathbb{R}^2$ whose boundary, $\partial\Omega$, or a portion of it is defined by NURBS curves. Every NURBS is assumed to be parameterized by

$$\boldsymbol{C} : [0,1] \longrightarrow \boldsymbol{C}([0,1]) \subseteq \partial\Omega \subset \mathbb{R}^2.$$

A regular partition of the domain $\overline{\Omega} = \bigcup_e \overline{\Omega}_e$ in triangles is assumed such that every element Ω_e has at most one side, Γ_e, on the NURBS boundary. Figure 4 shows a domain with part of the boundary described by a NURBS curve corresponding to the NACA 0012 airfoil, and a valid triangulation for NEFEM.

As usual in FE mesh generation codes, it is assumed that every curved boundary side belongs to a unique NURBS, $\Gamma_e \subseteq \boldsymbol{C}([0,1])$. That is, one element edge can not be defined by portions

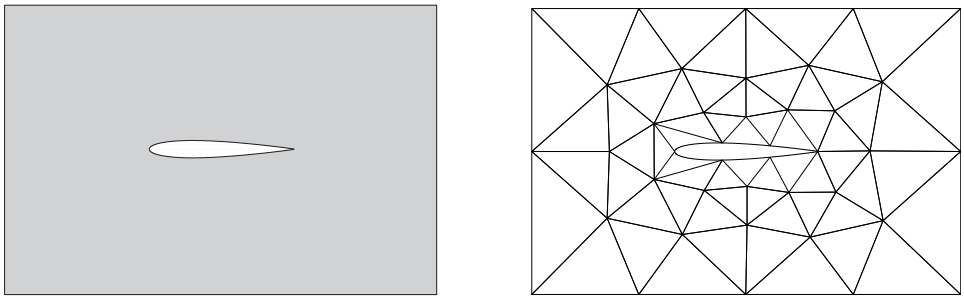


Figure 4. Physical domain with part of the boundary defined by a NURBS curve (left) and a valid triangulation for the NEFEM (right)

of two (or more) different NURBS curves. But on the contrary, it is important to note that 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 with respect to the isogeometric analysis [11].

Every *interior* element (i.e. elements not having an edge that coincides with the NURBS boundary) can be defined and treated as standard FE or DG elements. Therefore, in the vast majority of the domain, interpolation and numerical integration are standard. This section is devoted to the definition of the interpolation and the numerical integration at an element with one side, Γ_e, along the NURBS boundary. Say Ω_e is an element with two straight interior edges and one side defined by a trimmed NURBS,

$$\Gamma_e = \boldsymbol{C}([\lambda_1^e, \lambda_2^e]),$$

where λ_1^e and λ_2^e are the parametric coordinates (in the parametric space of the NURBS) of the end points of Γ_e; obviously they must verify $0 \leq \lambda_1^e < \lambda_2^e \leq 1$.

For each element Ω_e, a triangle T_e is defined using its vertices, see Figure 5. A linear mapping $\boldsymbol{\Psi} : I \longrightarrow T_e$ is used, which goes from the reference triangle I to the triangle T_e, see Figure 6. The inverse of this linear transformation maps the triangle T_e into the reference triangle I and, more important, also maps the actual element Ω_e, which is in the physical domain, into a curved element in local coordinates with two straight sides, namely

$$I_e := \boldsymbol{\Psi}^{-1}(\Omega_e),$$

see Figure 7. I_e is called the *local curved element* for the actual element Ω_e.

Note that the reference triangle I is the same for all elements Ω_e. However, the local curved element I_e depends on the trimmed NURBS defining the curved side Γ_e of Ω_e, and therefore it is different for every element Ω_e intersecting the NURBS boundary.

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

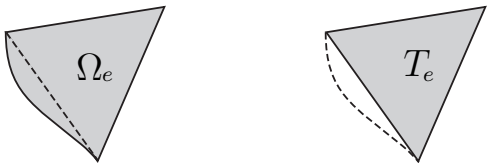


Figure 5. Actual element Ω_e (left) and triangle T_e defined using its vertices (right) in the physical domain

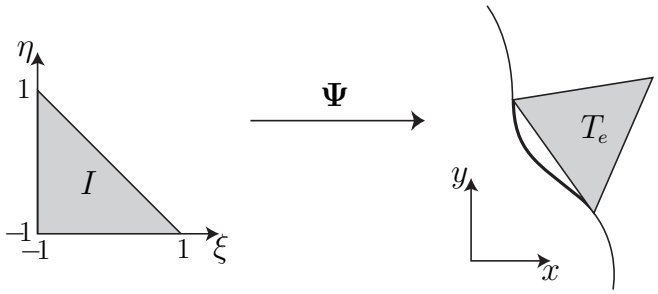


Figure 6. Mapping $\boldsymbol{\Psi}$ defined as a linear transformation from the reference triangle I to the triangle T_e in the physical domain

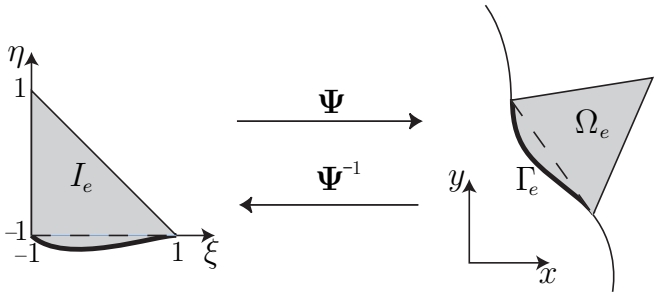


Figure 7. Linear transformation mapping the local curved element I_e to the actual element Ω_e in the physical domain

3.1. 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, the 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, $\{L_i(\boldsymbol{\xi})\}_{i=1}^{\mathtt{n_{en}}}$, more systematic, for any order and for any distribution of nodes, the implementation proposed in [13] is adopted. An orthogonal polynomial basis $\{P_i(\boldsymbol{\xi})\}_{i=1}^{\mathtt{n_{en}}}$ is considered in I. This basis is independent of the

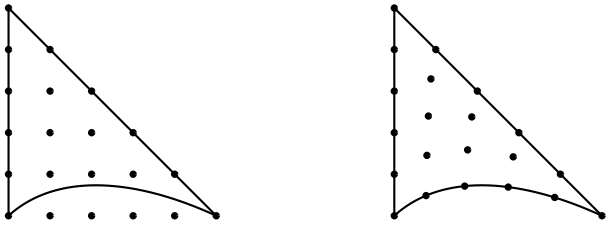


Figure 8. 5th-order nodal distributions in I_e: for equally-spaced nodes in the straight-side triangle (left) and adapted to the NURBS side (right)

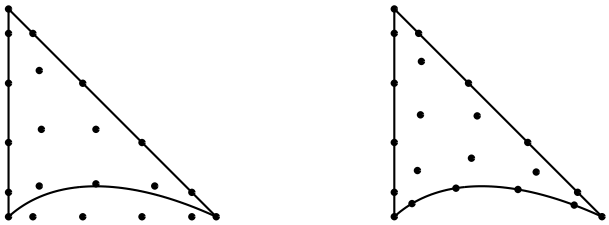


Figure 9. 5th-order nodal distributions in I_e: Fekete points in the straight-side triangle (left) and adapted to the NURBS side (right)

nodal coordinates $\{\boldsymbol{\xi}_i\}_{i=1}^{\mathtt{n_{en}}}$ and the Lagrange polynomial basis is then obtained as

$$L_i(\boldsymbol{\xi}) = \sum_{j=1}^{\mathtt{n_{en}}} \left[\boldsymbol{V}^{-1}\right]_{ji} P_j(\boldsymbol{\xi}), \tag{3}$$

where $\mathtt{n_{en}}$ is the number of element nodes and the multidimensional Vandermonde matrix is defined as $V_{ij} := P_j(\boldsymbol{\xi}_i)$, for $i, j = 1, \ldots, \mathtt{n_{en}}$.

Remark 2. *Note that equation* (3) *holds for any polynomial basis* $\{P_i(\boldsymbol{\xi})\}_{i=1}^{\mathtt{n_{en}}}$. *However, if* $\{P_i(\boldsymbol{\xi})\}_{i=1}^{\mathtt{n_{en}}}$ *is an orthogonal polynomial basis, such as the one derived from the Jacobi polynomials [13, 14, 15, 16], a moderate condition number for the Vandermonde matrix* $\boldsymbol{V}$ *is ensured and some inner products may be evaluated analytically.*

From an implementation point of view, it is worth noting that all element matrices can be first computed for the orthogonal polynomial basis, and then transformed with the Vandermonde matrix. That is, let $\boldsymbol{M}_e^p$ be an element matrix computed in terms of the orthogonal polynomial basis, then $\boldsymbol{M}_e = \boldsymbol{V}^{-T}\boldsymbol{M}_e^p\boldsymbol{V}^{-1}$ is the corresponding element matrix for the Lagrange nodal basis.

Different options can be considered for the definition of a nodal distribution in I_e. If low-order elements are used, which is the standard approach in FE, equally-spaced nodal distributions on the straight-side triangle can be implemented directly, see Figure 8. When high-order elements are used for high-fidelity computations, as it is standard in DG methods, it is more convenient to use special distributions of nodes in order to reduce the condition number of the resulting elemental matrices, see [17, 18] for details. Fekete points [19] are a good example of such distributions, see Figure 9.

Defining a nodal distribution on the straight-side triangle induces a marginal extra efficiency. Because there is both a unique definition of nodal coordinates and only one computation of the

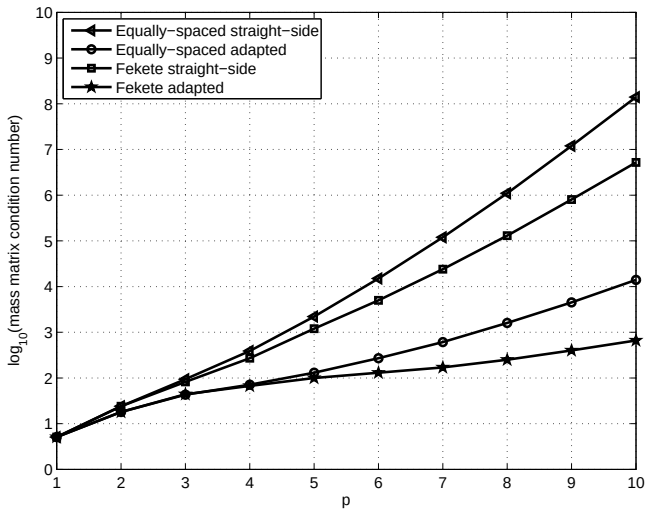
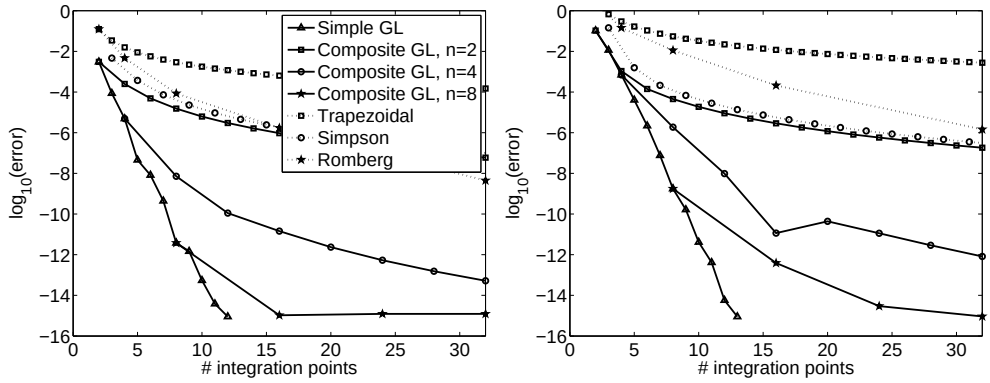


Figure 10. Condition number of the mass matrix as a function of the interpolation degree (p), using an equally-spaced nodal distributions and Fekete points

Vandermonde matrix (independently of the curved element). Another alternative is to adapt the nodal distribution to the exact geometry, see also Figures 8 and 9. This option is more reasonable when nodal values are prescribed along the boundary; but, in principle, it does not represent any advantage if boundary conditions are imposed in weak form, as usual in DG formulations. Note however, the evolution of the condition number shown in Figure 10 for the element mass matrix as a function of the degree of the interpolation. As expected, Fekete points clearly decrease the condition number but adapted distributions of nodes also have a major influence on the condition number.

3.2. Numerical integration

The weak form that must be solved requires both integrations along element edges and in the elements interiors. All integrals in elements not having an edge along the NURBS boundary are computed using standard procedures. The elements Ω_e with one side, Γ_e, on the NURBS boundary require special attention. Two cases must be considered: line integrals (usually related to the implementation of natural boundary conditions or to flux evaluation along Γ_e in a DG context) or surface integrals (standard integrals in the element Ω_e). As discussed in the previous section, since NEFEM uses polynomials to approximate the solution, the difficulties in numerical integration are only restricted to the definition of a proper numerical quadrature in the curved element $I_e = \mathbf{\Psi}^{-1}(\Omega_e)$ or its corresponding curved face $\mathbf{\Psi}^{-1}(\Gamma_e)$. This, as will be observed below, reduces the complexity in the accurate evaluation of integrals, which are not as costly as in standard mesh-free methods [20], or in isogeometric analysis [11].



(a) xy (b) $x^{11}y^9$

Figure 11. Relative error for the integration of polynomials xy and $x^{11}y^9$ along the front part of the NACA 0012 airfoil.

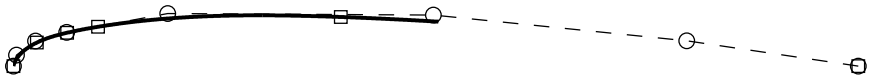


Figure 12. Trimmed NURBS curve describing the front part of the NACA 0012 airfoil (solid line), control points (○), control polygon (dashed line) and image of the breakpoints (□)

3.2.1. Line integrals A line integral to be computed along a curved boundary side given by a trimmed NURBS, $\Gamma_e = \boldsymbol{C}([\lambda_1^e, \lambda_2^e])$, can be written as

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

where f is a generic function (usually polynomial), and $|J_{\boldsymbol{C}}|$ denotes the norm of the differential of the NURBS parametrization $\boldsymbol{C}$ (which, in general, is not a polynomial). As usual, a 1D numerical quadrature is used for the numerical computation of the integral, namely

$$\int_{\Gamma_e} f \, d\ell \approx \sum_{i=1}^{\mathtt{n_{ip}}} f\big(\boldsymbol{C}(\lambda_i)\big) \, |J_{\boldsymbol{C}}(\lambda_i)| \, \omega_i, \tag{4}$$

where λ_i and ω_i are the coordinates and weights of the $\mathtt{n_{ip}}$ integration points in $[\lambda_1^e, \lambda_2^e]$.

Recall that the parametrization of a trimmed NURBS, $\boldsymbol{C}$, is a piecewise rational function whose definition changes at breakpoints. Thus, an independent numerical quadrature must be considered at every interval between breakpoints (patch) to account for the discontinuous nature of the parametrization. In [21] a detailed comparison and discussion on different alternatives to evaluate (4) is presented. 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. For instance, Figure 11 shows the evolution of the relative error for the integration of polynomials xy and $x^{11}y^9$ along the trimmed NURBS corresponding to the front part of the NACA 0012 airfoil, see Figure 12.

For the Gauss-Legendre composite rules, $\mathtt{n}$ denotes the number of integration points in every subinterval. Note that a composite quadrature with 2 intervals and $\mathtt{n} = 8$ Gauss-Legendre integration points in each interval achieves almost machine precision. Thus, although the faster convergence is obviously obtained for high-order simple quadratures (in each patch), the use of composite rules is very attractive, because it allows the definition of adaptive quadratures to control the integration error and ensure reliable computations for any NURBS and any order of polynomial interpolation.

Nevertheless, it is worth noting that in practical applications it is not necessary to compute numerical integrals with machine precision. Numerical experiments simulating the scattering of a planar electromagnetic wave [21] reveal that, in practice, if the NURBS is parametrized such that velocity has smooth variations, only one extra integration point, compared to standard isoparametric FE, is required. That is, for a given discretization with polynomials of order p, a numerical quadrature with $p+1$ Gauss-Legendre integration points in each patch provides the maximum accuracy.

3.2.2. Element integrals NEFEM also requires to compute integrals in an element Ω_e with one side Γ_e on the NURBS boundary, see Figure 7, that is

$$\int_{\Omega_e} f \, dx\, dy = |J_{\boldsymbol{\Psi}}| \int_{I_e} f \, d\xi \, d\eta \tag{5}$$

where $|J_{\boldsymbol{\Psi}}|$ is the determinant of the Jacobian of the linear transformation $\boldsymbol{\Psi}$. Thus, a numerical quadrature for every curved element I_e is needed. In [21] different alternatives are presented and discussed. The best alternative, see Figure 13, is to define a transformation from the rectangle $[\lambda_1^e, \lambda_2^e] \times [0,1]$ to the curved element I_e, namely,

$$\begin{aligned} \boldsymbol{\varphi} \, : [\lambda_1^e, \lambda_2^e] \times [0,1] &\longrightarrow I_e \\ (\lambda, \zeta) &\longmapsto \begin{Bmatrix} \varphi_1 \\ \varphi_2 \end{Bmatrix} := \begin{Bmatrix} \phi_1(\lambda)(1-\zeta) - \zeta \\ \phi_2(\lambda)(1-\zeta) + \zeta \end{Bmatrix} \end{aligned} \tag{6}$$

where $\boldsymbol{\phi} = (\phi_1, \phi_2)^T := \boldsymbol{\Psi}^{-1} \circ \boldsymbol{C}$ is the parametrization of the trimmed NURBS corresponding to the curved side in I_e. Note that this transformation requires that nodes are numbered following the non restrictive assumption presented in Remark 1. Note also, that such a parametrization is linear in ζ and as discussed in Remark 3 this implies important practical advantages. Thus, using the transformations shown in Figure 13, integral (5) is computed as

$$\int_{\Omega_e} f \, dx\, dy = |J_{\boldsymbol{\Psi}}| \int_{I_e} f \, d\xi \, d\eta \simeq |J_{\boldsymbol{\Psi}}| \sum_{i=1}^{\mathtt{n_{ip}}} \sum_{j=1}^{\mathtt{m_{ip}}} f(\boldsymbol{\xi}_{ij}) |J_{\boldsymbol{\varphi}}(\lambda_i, \zeta_j)| \, \omega_i \varpi_j \tag{7}$$

where $\mathtt{n_{ip}}$ and $\mathtt{m_{ip}}$ are the number of integration points in the λ and ζ directions, respectively, $\boldsymbol{\xi}_{ij} := \boldsymbol{\varphi}(\lambda_i, \zeta_j)$, $\{\lambda_i, \omega_i\}$ and $\{\omega_i, \varpi_i\}$ are the 1D quadrature points and weights for $[\lambda_1^e, \lambda_2^e]$ and $[0,1]$ respectively, and $|J_{\boldsymbol{\varphi}}|$ is the determinant of the Jacobian of the transformation $\boldsymbol{\varphi}$. Note that to integrate a polynomial f of degree k in Ω_e, given the transformations shown in Figure 13, the (non polynomial) function

$$\widetilde{f}(\lambda, \zeta) = f\big(\boldsymbol{\varphi}(\lambda, \zeta)\big) |J_{\boldsymbol{\varphi}}(\lambda, \zeta)|. \tag{8}$$

must be integrated in $[\lambda_1^e, \lambda_2^e] \times [0,1]$. Recall that $\boldsymbol{\varphi}$, see (6), is linear in ζ and, therefore, $f\big(\boldsymbol{\varphi}(\lambda, \zeta)\big)$ is a polynomial of degree k in ζ. Moreover, $|J_{\boldsymbol{\varphi}}(\lambda, \zeta)|$ is linear in ζ. Thus, $\widetilde{f}(\lambda, \zeta)$

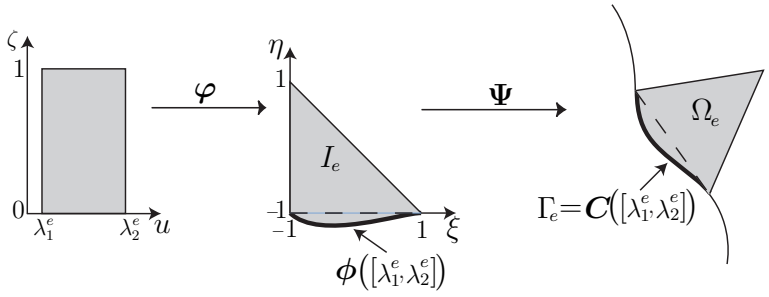


Figure 13. Transformation from $[\lambda_1^e, \lambda_2^e] \times [0, 1]$ to I_e and Ω_e

is a polynomial of degree $k + 1$ in ζ and consequently a Gauss-Legendre quadrature of order k (or $k + 1$ for even k) is optimal for the 1D quadrature along ζ.

Remark 3. *When the transformation from the rectangle $\boldsymbol{\varphi}$ is considered, the integrals involved in the elemental matrices, for a NEFEM solution with interpolation of degree p, can be exactly computed for one of the parameters, ζ, using a Gauss-Legendre quadrature with $p+1$ integration points. The other dimension, λ, can be integrated using the same quadrature considered in section 3.2.1 for line integrals over NURBS. Moreover, if the geometry is described with a q-th degree B-spline, the elemental matrices can be exactly computed with Gauss-Legendre quadratures with $p + 1$ integration points for the ζ parameter, and $q(p + 1)$ integration points for the NURBS parameter λ.*

Note that the rational definition of application $\boldsymbol{\varphi}$ is only due to the rational definition of the boundary. Then, if $\boldsymbol{C}$ is a B-spline, i.e. a piecewise polynomial parametrization, $\boldsymbol{\varphi}$ is also a piecewise polynomial function, and therefore, the function $\widetilde{f}$ is a piecewise polynomial that can be exactly integrated using Gauss-Legendre quadratures. For instance, the NACA 0012 geometry is described by a B-Spline of degree $q = 3$, and therefore element integrals can be exactly computed with Gauss-Legendre quadratures with $p+1$ and $3(p+1)$ integration points in each direction.

Another obvious option instead of using $\boldsymbol{\varphi}$ to transform a rectangle into I_e, is to define another transformation from I to a curved triangle I_e and then use quadratures specifically designed for triangles. This is also discussed in [21]. For standard FE these triangle quadratures require less integration points than other quadrature rules to achieve the same accuracy, but this is not the case here. The use of a transformation depending on the NURBS parametrization (from I to a curved triangle) leads to expensive triangle quadratures. The integration strategy proposed in this section is much more competitive due to the good behavior of parameter ζ commented in Remark 3.

The efficiency of the proposed quadrature is illustrated in Figure 14. It shows the integration points required to integrate x over a curved triangle with an error of 0.5%, using the transformation from a rectangle proposed in this section (with 30 integration points) and a symmetric triangle quadrature [22] adapted to the curved element (with 54 integration points), see [21] for details.

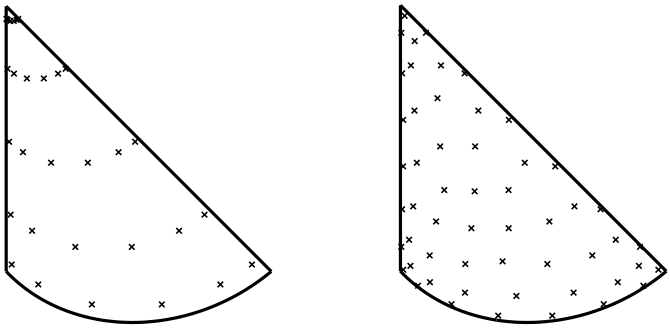


Figure 14. Two numerical quadratures in a curved element for the same accuracy

3.3. Some comments on the implementation of the NEFEM

The enhancement of a standard FE code with the NEFEM methodology requires little effort. Note that the main difference of a NEFEM code with a standard FE code is at the level of the computation of elemental matrices and vectors for curved elements and edges. In fact, fortunately, the usual routines for the computation of elemental matrices and vectors for straight-sided elements can be directly used, without any modification, just using a modified definition of the reference element. That is, the integration points and the shape functions evaluated at these points, which are usually an input of these routines, have to be particularized for every curved element I_e or side Γ_e. Thus, most of the routines usual in a standard FE code (routines for assembly, computation of elemental matrices and vectors, etc) can be directly used.

In fact, the most crucial point in the implementation may be the inclusion of the NURBS boundary information. The information for the evaluation of all the NURBS describing the boundary has to be stored. Moreover, for every curved side $\Gamma_e = \mathbf{C}([\lambda_1^e, \lambda_2^e])$ the information of the corresponding trimmed NURBS is also necessary, that is, the extremes of the interval λ_1^e and λ_2^e and a pointer to the information of the NURBS $\mathbf{C}$. Nowadays this is not an information usually provided by standard mesh generators but, it is worth noting that routines for the evaluation of the NURBS can be easily obtained or implemented [12].

On the other hand, in the context of DG formulations, NEFEM is a natural option for the implementation of high-order approximations in domains with curved boundaries. In DG codes it is usual to store only the vertices of a triangle mesh, and their connectivities, usually obtained with a linear triangle mesh generator. For high-order computations with straight-sided elements, if needed, all nodal coordinates are determined from the vertices coordinates. Under these circumstances NEFEM allows a straightforward implementation of curved boundaries, with no need of a high-order mesh generator, because the nodal coordinates at elements with one side on the NURBS boundary can be easily determined from the vertices of the triangles and the NURBS information.

3.4. A priori error estimates

Since NEFEM considers the usual FE polynomial interpolation, see section 3.1, a priori error estimates are exactly the same as those for standard FE. For instance, the result for the

solution of second order elliptic problems is recalled in the following theorem.

Theorem 1. *Let $\mathcal{T}_h$ a non-degenerate triangulation (i.e. there is a positive constant β such that $\varrho_e/h_e \geq \beta$, for all $\Omega_e \in \mathcal{T}_h$, where h_e and ϱ_e are the diameters of Ω_e and of the circle inscribed in Ω_e, respectively). Then, the following a priori estimate holds*

$$\|u - u_h\|_{\mathcal{L}^2(\Omega)} \leq K h^{p+1} |u|_{\mathcal{H}^{p+1}(\Omega)}, \tag{9}$$

where u and u_h are the exact and the FE solution respectively, K is a constant, h is the mesh size, and p is the polynomial degree of interpolation.

Moreover, for p-refinement convergence the following estimate also holds,

$$\|u - u_h\|_{E(\Omega)} \leq C \exp(-kN^r), \tag{10}$$

where $\|\cdot\|_{E(\Omega)}$ is the energy norm, C and k are positive constants, N is the number of nodes, and $r \gtrsim 1/2$ for 2D problems.

The same arguments used in the proof of standard FE error estimates are valid for the proof of Theorem 1. In fact, the derivation of a priori error estimates for NEFEM is identical to FE a priori estimates in polygonal domains, which can be found in [23, 24, 25] for h-refinement and in [26] for p-refinement.

It is worth noting that contrary to NEFEM, the proof of a priori error estimates for FE in the presence of curved boundaries requires a special attention. First, the use of isoparametric FE induces geometric errors because the computational domain is, in fact, a piecewise polynomial approximation of the physical geometry. Thus, to obtain optimal a priori error estimates, the maximum distance between the computational and the exact boundary should be bounded by γh^p, where γ is a constant, h is the mesh size and p is the polynomial degree. Moreover, a bound of the jacobian of the isoparametric transformation is also necessary.

But for NEFEM (as well as for FE with polygonal boundaries) the spatial discretization does not introduce geometric errors. Moreover, NEFEM uses a linear mapping to relate local and cartesian coordinates, see section 3. In fact, the linear transformation used by NEFEM is exactly the same mapping used by standard FE on a domain with polygonal boundaries. Consequently, all the a priori error estimates demonstrated in FE for domains with polygonal boundaries can be reproduced exactly for NEFEM, even in the presence of elements far from having straight edges.

4. NUMERICAL EXAMPLES

The application of the proposed methodology is illustrated using several numerical examples. The first example is an elliptic problem and it is solved using a continuous Galerkin formulation. More complex problems, concerning the numerical solution of the transient Maxwell's equations, are also considered in order to illustrate the efficiency of NEFEM in a DG framework.

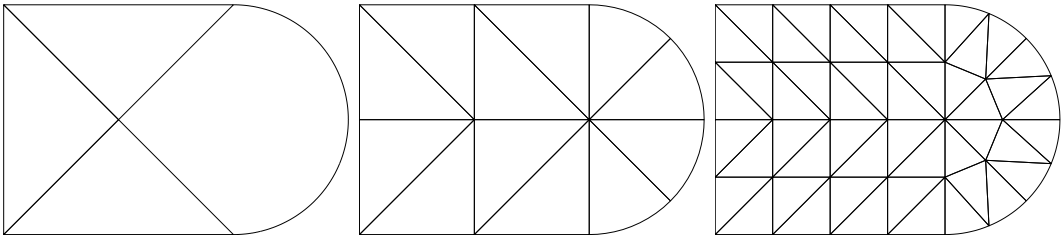
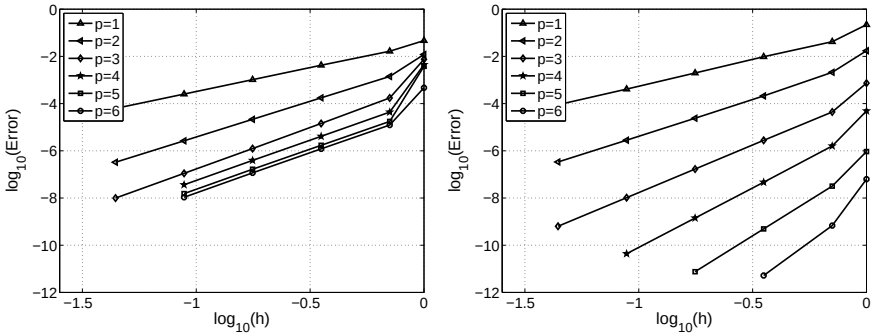


Figure 15. Coarse meshes for the h-refinement test. Nested remeshing is used for further refinement.



(a) FEM (b) NEFEM

Figure 16. h-convergence in the $\mathcal{L}^2(\Omega)$ norm for the Poisson example

4.1. Poisson problem

The following model problem is solved in two dimensions

$$\left\{\begin{array}{rcll} -\Delta u & = & f & \text{in } \Omega \\ u & = & u_d & \text{on } \Gamma_d \\ \nabla u \cdot \boldsymbol{n} & = & g_n & \text{on } \Gamma_n \end{array}\right. \tag{11}$$

where Ω is the domain (see three NEFEM meshes in Figure 15), $\overline{\Gamma}_d \cup \overline{\Gamma}_n = \partial\Omega$ and $\boldsymbol{n}$ is the outward unit normal vector on $\partial\Omega$. The source is given by $f(x,y) = x\cos(y) + y\sin(x)$, in such a way that the analytical solution of the problem is known and smooth,

$$u(x,y) = x\cos(y) + y\sin(x).$$

A Dirichlet boundary condition, corresponding to the analytical solution, is imposed in the polygonal part of the boundary, and a Neumann boundary condition, also corresponding to the analytical normal flux, is imposed in the curved part of the boundary. The curved part of the boundary is described exactly with NEFEM using one quadratic trimmed NURBS, whereas isoparametric FE induces a piecewise polynomial approximation of the boundary.

In order to check the theoretical convergence rates of Theorem 1 an h-refinement is first explored. Figure 15 shows the first three computational meshes, nested remeshing is used for further refinement. The $\mathcal{L}^2(\Omega)$ error is plotted in Figure 16 for polynomials of degree up to p=6 both for isoparametric FE and NEFEM.

	FEM		
p	$\mathcal{L}^2$	$\mathcal{L}^\infty$	Energy
1	2.02	1.61	1.00
2	3.02	2.97	2.02
3	3.48	3.02	2.46
4	3.42	2.90	2.42
5	3.42	2.92	2.42
6	3.43	2.94	2.43

	NEFEM		
p	$\mathcal{L}^2$	$\mathcal{L}^\infty$	Energy
1	2.21	1.63	1.21
2	3.06	2.98	2.12
3	4.02	4.00	3.04
4	5.02	4.95	4.08
5	6.02	5.94	5.02
6	7.02	6.97	6.09

Table I. FEM and NEFEM h-refinement rates of convergence for the Poisson example

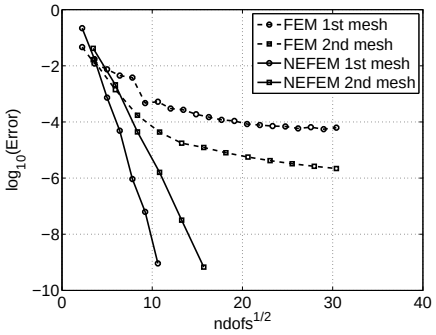


Figure 17. p-convergence in the energy norm for Poisson example

Table I also reports the rate of convergence in the $\mathcal{L}^2(\Omega)$ norm, the $\mathcal{L}^\infty(\Omega)$ norm and the energy norm. These results show as expected, see [23], that isoparametric FE only attain the optimal rate of convergence for linear and quadratic polynomials. For higher order approximations, the error introduced by the isoparametric transformation clearly deteriorates the optimal rate of convergence. NEFEM provides optimal convergence rates even for high-order computations.

Next, a p-convergence study is performed using the first two computational meshes represented in Figure 15. As usual in the context of p and hp versions of the FEM, see [26, 27], the error in the energy norm is represented in Figure 17 as a function of the square root of the number of degrees of freedom. NEFEM, as expected from Theorem 1, presents optimal convergence. However, convergence for isoparametric FE computations is clearly deteriorated for high-order approximations. The best accuracy that can be obtained for the first mesh in Figure 15 with isoparametric FE corresponds to an error in energy norm of about 10^{-4}. In order to improve accuracy with isoparametric FE an hp-refinement strategy must be adopted. For instance, errors of order 10^{-6} can be achieved with the second mesh shown in Figure 15.

No differences between the isoparametric FEM and NEFEM results are observed in the solution of (11) when Dirichlet boundary conditions are strongly imposed along the whole boundary. Thus, for this problem, the source of the extra error present in the isoparametric FE solution is due to the implementation of Neumann boundary conditions on the curved boundary. It is worth noting that when Dirichlet boundary conditions are imposed in weak form relevant differences between FE and NEFEM results are expected. This is standard in

DG methods and recently it has also been proposed in fluid mechanics for other techniques [28].

To conclude, NEFEM exhibits important advantages in the numerical solution of Poisson problems in the presence of curved boundaries. An important reduction of the error is observed when the NEFEM approach is adopted. NEFEM is, in some cases, four orders of magnitude more precise than isoparametric FE. Moreover, numerical examples corroborate that NEFEM achieves the optimal convergence rates for any polynomial degree.

Finally, it is worth noting that in a FE adaptive process, see [29], the computational mesh must be locally refined to properly account for 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.

4.2. Electromagnetic scattering simulations

In this section a DG formulation is considered for the simulation of 2D scattering of a single plane electromagnetic wave by a *Perfect Electric Conductor* (PEC) obstacle, assumed to be surrounded by free space. For a linear isotropic material of relative permittivity ε and relative permeability μ, the 2D Maxwell's equations (which are decoupled in the *Transverse Electric*, TE, and *Transverse Magnetic*, TM, modes) can be written in dimensionless conservative form as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_k(\boldsymbol{U})}{\partial x_k} = \boldsymbol{0} \qquad \text{in } \Omega, \tag{12}$$

where Einstein notation is assumed. The vector of conserved quantities $\boldsymbol{U}$ and the fluxes $\boldsymbol{F}_k$ are

$$\boldsymbol{U} = \begin{pmatrix} \varepsilon E_1 \\ \varepsilon E_2 \\ \mu H_3 \end{pmatrix}, \quad \boldsymbol{F}_1 = \begin{pmatrix} 0 \\ H_3 \\ E_2 \end{pmatrix}, \quad \boldsymbol{F}_2 = \begin{pmatrix} -H_3 \\ 0 \\ -E_1 \end{pmatrix},$$

for the TE mode, and

$$\boldsymbol{U} = \begin{pmatrix} \varepsilon E_1 \\ \varepsilon E_2 \\ \mu H_3 \end{pmatrix}, \quad \boldsymbol{F}_1 = \begin{pmatrix} 0 \\ H_3 \\ E_2 \end{pmatrix}, \quad \boldsymbol{F}_2 = \begin{pmatrix} -H_3 \\ 0 \\ -E_1 \end{pmatrix},$$

for the TM mode, where $\boldsymbol{E} = (E_1, E_2, E_3)^T$ and $\boldsymbol{H} = (H_1, H_2, H_3)^T$ are the *scattered* electric and magnetic field intensity vectors. It is assumed that there are no electric sources in the material.

In the DG implementation, numerical fluxes at the interior edges are defined from the Rankine-Hugoniot jump conditions [30], which can also be interpreted as a *flux splitting* technique, see [31]. Artificial absorbing boundaries are implemented with a first-order Silver-Müller condition, see for instance [32], and in all the examples computations are stopped before scattered waves reach the artificial boundary. Otherwise, more accurate artificial absorbing boundaries should be considered, see for instance [33]. At a PEC boundary, the tangential component of the total electric field (scattered plus incident) and the normal component of the total magnetic field vanish, that is

$$\boldsymbol{n} \times \boldsymbol{E} + \boldsymbol{n} \times \boldsymbol{E}^I = 0, \qquad \boldsymbol{n} \cdot \boldsymbol{H} + \boldsymbol{n} \cdot \boldsymbol{H}^I = 0,$$

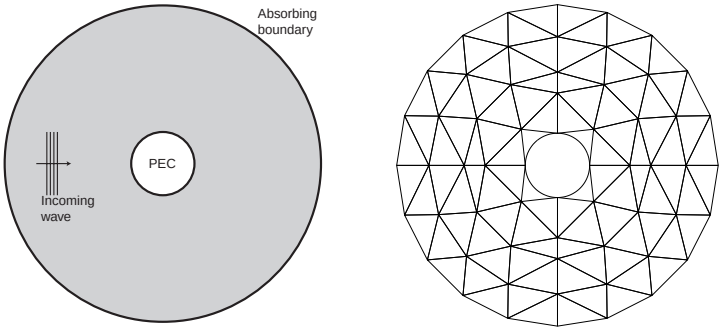


Figure 18. Scattering of a planar wave by a PEC cylinder: problem setup and computational mesh

where the superscript I refers to the incident wave. Using Rankine-Hugoniot jump conditions, PEC boundary conditions lead to the following numerical flux

$$\widehat{\boldsymbol{F}}_n^{\texttt{PEC}}(\boldsymbol{U}) = \begin{pmatrix} -n_2 H_3^{\texttt{PEC}} \\ n_1 H_3^{\texttt{PEC}} \\ -\alpha^I \end{pmatrix}, \qquad \text{where} \quad H_3^{\texttt{PEC}} = H_3 + \sqrt{\varepsilon/\mu}\left(\alpha + \alpha^I\right).$$

for the TE mode, and

$$\widehat{\boldsymbol{F}}_n^{\texttt{PEC}}(\boldsymbol{U}) = \begin{pmatrix} n_2 E_3^{\texttt{PEC}} \\ -n_1 E_3^{\texttt{PEC}} \\ -\beta^I \end{pmatrix}, \qquad \text{where} \quad E_3^{\texttt{PEC}} = E_3 + \sqrt{\mu/\varepsilon}\left(\beta + \beta^I\right).$$

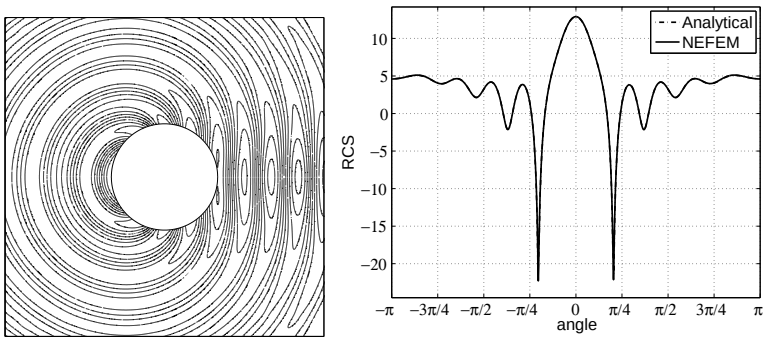
for the TM mode, where $\alpha := n_1 E_2 - n_2 E_1 \equiv \boldsymbol{n} \times \boldsymbol{E}$ and $\beta := -n_1 H_2 + n_2 H_1 \equiv -\boldsymbol{n} \times \boldsymbol{H}$. All computations are performed using a fourth-order explicit Runge-Kutta time-marching scheme.

4.2.1. Circle: The first example considers an incident plane wave travelling in the x^+ direction and scattered by a circle, which is exactly described with a NURBS curve, see Figure 18. The diameter of the circle is two wave lengths. A coarse mesh with only four elements for the discretization of the NURBS boundary is considered and high-order approximations are used to properly capture the solution.

Figure 19 shows the transverse field H_3 and the *Radar Cross Section* (RCS) for the TE mode, after four cycles, with a NEFEM approximation of degree 7. The NEFEM solution and the analytical solution [34] are overlapped; more precisely the $\mathcal{L}^2([-\pi, \pi])$ error in the RCS is $1.6\ 10^{-2}$, whereas the error for isoparametric FE is $6.1\ 10^{-2}$. Figure 20 shows a more detailed comparison: the RSC error distribution for isoparametric FE and for NEFEM using high-order approximation, namely $p = 5, 6$. In both cases, NEFEM clearly improves the solution.

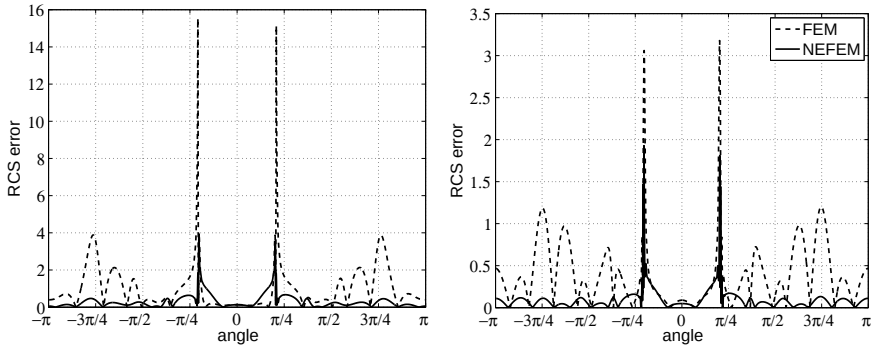
The same analysis is performed for the TM mode. Figure 21 shows the E_3 distribution and the RCS for $p = 7$ on the coarse mesh depicted in Figure 18. Again, the numerical and analytical solution are overlapped. Moreover, the comparison with isoparametric FE in Figure 22 demonstrates, once more, the better performance of NEFEM.

To compare accuracy, the error evolution for increasing p is depicted in Figure 23. For the same discretization (i.e. same degree of interpolation) NEFEM results are more accurate than the isoparametric FE ones. For instance, NEFEM with degree $p = 5$, provides a RCS with an



(a) H_3 (b) RCS

Figure 19. H_3 distribution and TE RCS after four cycles using NEFEM and $p = 7$



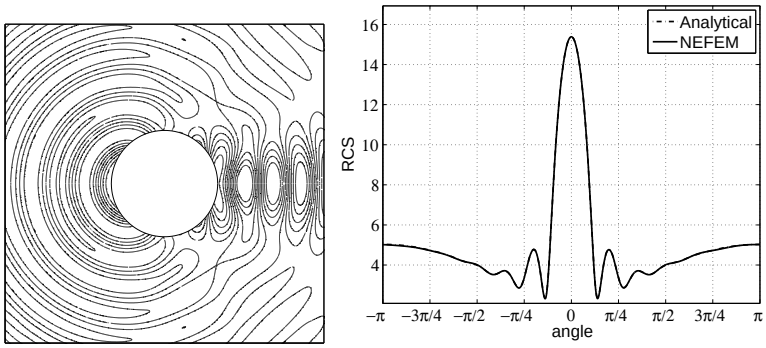
(a) 5th order (b) 6th order

Figure 20. Comparison of TE RCS error distribution for FE and NEFEM with $p = 5, 6$

error of about 2%, whereas isoparametric FE require $p = 7$ to achieve a comparable accuracy (2.4 times more degrees of freedom than NEFEM). Therefore, although integration in elements along the NURBS boundary represents higher computational cost, NEFEM is more efficient than isoparametric FE because of the drastic reduction in the number of degrees of freedom (42% in this example).

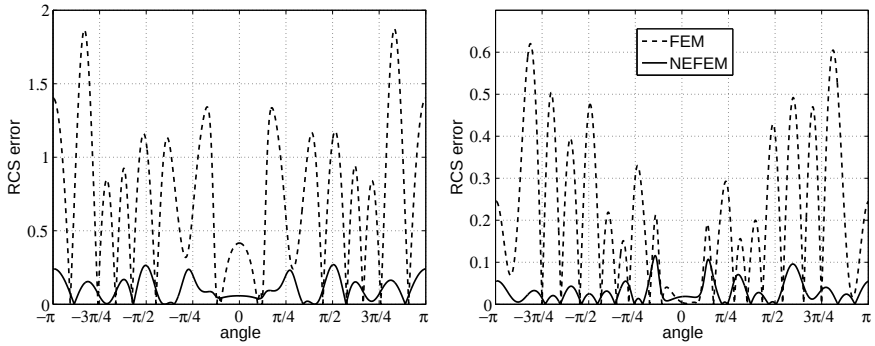
It is worth noting that low-order approximations ($p = 1$ and 2) induce similar performances for NEFEM and isoparametric FE. Note that similar behavior and accuracy between NEFEM and isoparametric FE was already observed for the Poisson example, see section 4.1, at low-order approximations. Thus, NEFEM is really competitive for high-order approximations.

To further understand the difference in accuracy between isopametric FE and NEFEM, a *cartesian FE* approximation is also considered and its accuracy depicted in Figure 23. That is, the usual reference element is used for integration purposes, but the polynomial base for the approximation of the solution is defined with cartesian coordinates directly in the physical domain. This means that an approximate (piecewise polynomial) description of the boundary is maintained, but the isoparametric transformation for the approximation is avoided. Thus,



(a) E_3 (b) RCS

Figure 21. E_3 distribution and TM RCS after four cycles using NEFEM and $p = 7$



(a) 5th order (b) 6th order

Figure 22. Comparison of TM RCS error distribution for FE and NEFEM with $p = 5, 6$

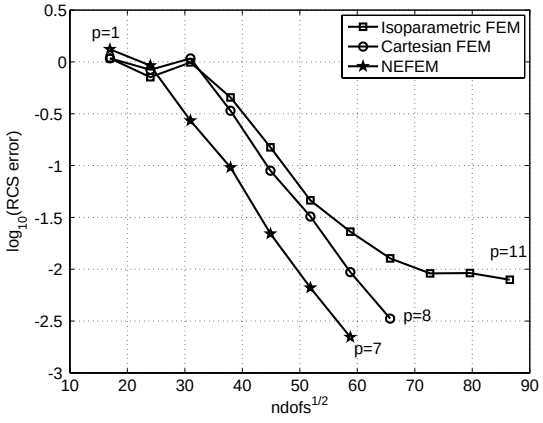


Figure 23. p-refinement convergence with the mesh in Figure 18. Markers are located for $p = 1, 2, 3, \ldots$

cartesian FE reproduces exactly polynomials in the physical domain. *Cartesian FE* results reveal that the improvement of NEFEM compared to isoparametric FEM is due to two main facts: the exact description of the boundary, and the use of a polynomial base with cartesian coordinates for the interpolation of the solution.

On one hand, the error due to the piecewise polynomial approximation of the boundary, which also affect the *cartesian FE* solution, are removed in the NEFEM solution. More precisely, with NEFEM the outward normal vector is computed exactly in terms of the NURBS boundary parametrization, improving the flux computation at the PEC boundary.

On the other hand, the difference between isoparametric and *cartesian FE* evidences the error due to the isoparametric transformation used to map the polynomial base in the reference element; which, for high-order approximation leads to a base in cartesian coordinates far from being a polynomial.

It is important to note that this convergence analysis also reveals that to overcome the deficiencies of the isoparametric FE in high-order computations [23], NEFEM is a more efficient strategy than *cartesian FE*. Moreover, the increase in the computational cost, due to the specific numerical treatment of curved elements along the boundary, is similar in NEFEM and *cartesian FE*, but from an accuracy point of view NEFEM provides much better results. For instance, the curves in Figure 23 show that to attain an accuracy comparable to a NEFEM computation with degree $p = 5$ (with an error of 2.2% in the RCS), *cartesian FE* require a discretization with degree $p = 6$ (with an error of 3.2%). That is, NEFEM provides similar accuracy to *cartesian FE* with a 63% reduction in the number of degrees of freedom.

4.2.2. NACA airfoil: The second example consists on the scattering of a planar wave by the NACA 0012 airfoil. The NACA 0012 is a symmetric airfoil with analytical expression [35], that can not be exactly described with a NURBS curve. As usual in the context of airfoil shape optimization, an approximation of the upper part of the airfoil using a B-Spline with eight control points is considered here, see for instance [21] for the B-spline data. Thus, integration is therefore exact using the numerical quadrature, see Remark 3.

Figure 24 show a detail of the computational mesh and the solution of the TE mode with degree $p = 8$, for an airfoil with a chord length of 2 wave lengths. The angle of incidence is 0 rad. The NEFEM E_3 distribution and the RCS are in good agreement with a reference numerical solution, see for instance [33], with an $\mathcal{L}^2([-\pi, \pi])$ error of 10^{-2} in the RCS. Figure 24 also shows the distribution of the error in the RCS for NEFEM and isoparametric FE. Again, NEFEM provides much more accurate results than isoparametric FE.

Next NEFEM performance for a computation with higher frequency is tested: the NACA 0012 airfoil with a chord length of 10 wave lengths and angle of incidence of $\pi/2$ rad is considered, see for instance [36]. A detail of the computational mesh, and the NEFEM solution obtained with an approximation of degree 13 is shown in Figure 25. The E_3 field and the RCS are in good agreement with the reference solution, demonstrating the applicability of the NEFEM methodology for the computation with high-degree approximation in coarse meshes (only 8 elements for the description of the airfoil boundary). The error distribution is also plotted in Figure 25. In this example the errors in the isoparametric FE solution are clearly unacceptable, whereas NEFEM demonstrates its good performance for high-order computations.

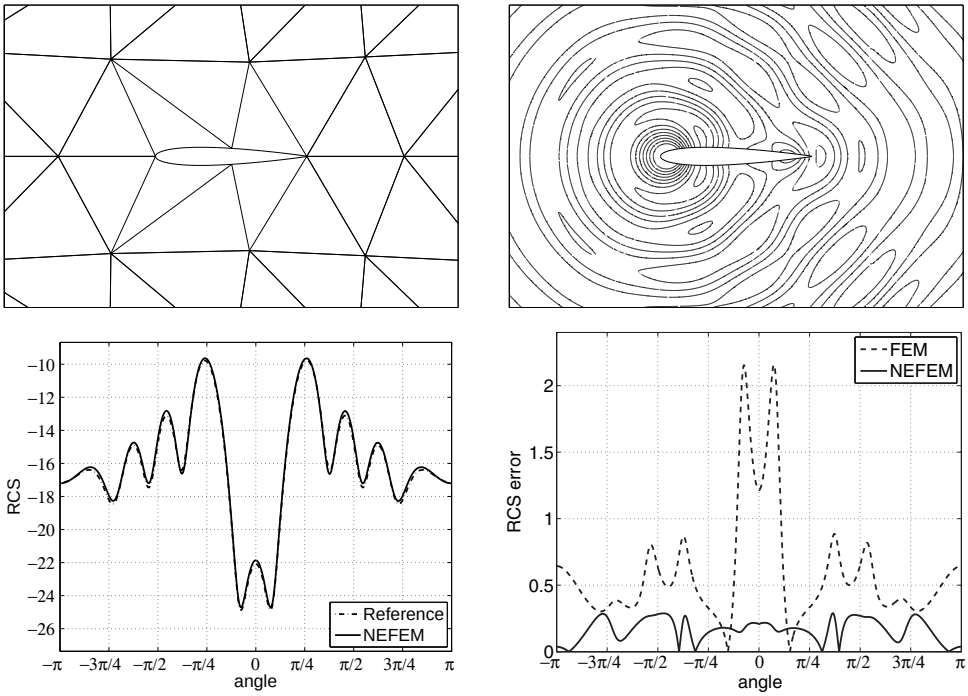


Figure 24. NACA 0012 airfoil with chord length of 2 wave lengths and angle of incidence of 0 rad ($p = 8$): detail of the computational mesh, H_3 distribution, RCS and error distribution.

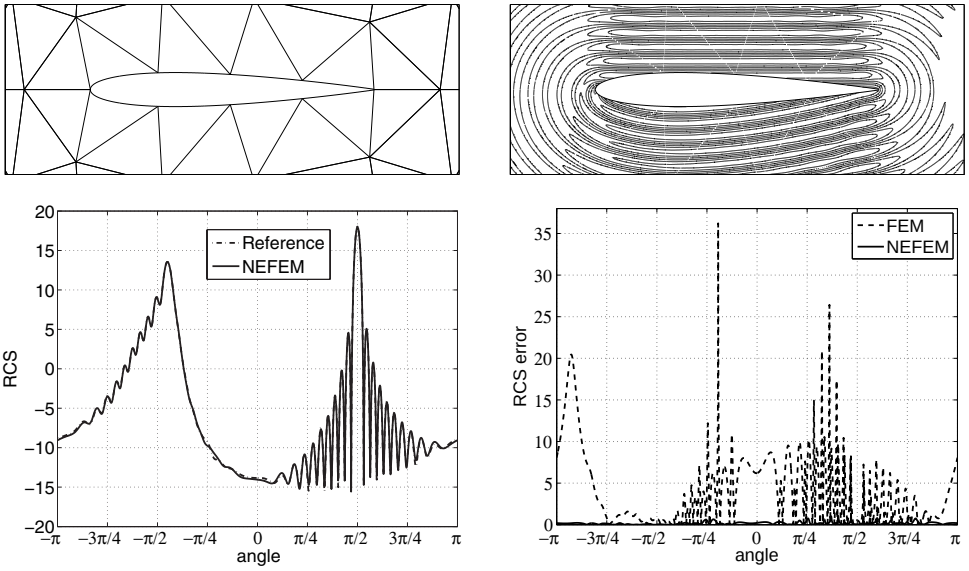


Figure 25. NACA 0012 airfoil with chord length of 10 wave lengths angle of incidence of $\pi/2$ rad ($p = 13$): detail of the computational mesh, H_3 distribution, RCS and error distribution.

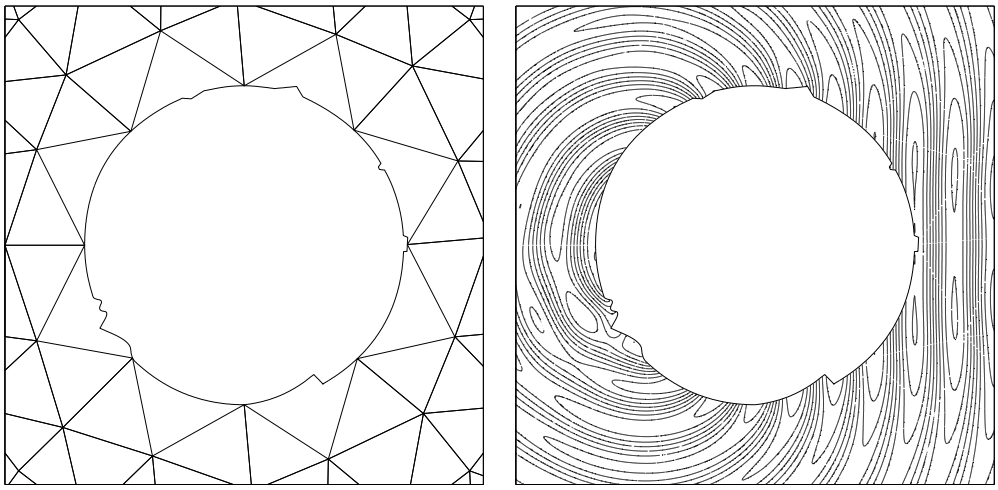


Figure 26. Irregular circle: detail of the computational mesh and NEFEM solution for $p = 9$.

4.2.3. Complex scatterers: Previous examples show the advantages of the NEFEM formulation in front of isoparametric FE for the numerical solution of some classical test cases. To illustrate all the capabilities of the NEFEM more complex scatterers are considered next. The first example is the scattering induced by an *irregular* circle with diameter of four wave lengths. The geometry of the obstacle is described *exactly* using one NURBS and a coarse mesh with only eight elements for the representation of the boundary is considered, see Figure 26.

As noted earlier, see section 3, it is important to remark that the only restriction for a NEFEM triangle is that the curved edge belongs to one NURBS. The computational mesh is chosen to emphasize the possibilities of the NEFEM. It is not necessary to locate nodes at boundary corners (boundary points with $\mathcal{C}^0$ continuity), nor to refine the mesh near the boundary to capture the geometry (it is exactly represented in the NEFEM independently of the spatial discretization!).

Figure 26 also shows the transverse field H_3 with $p = 9$ after four cycles. Even for elements with corners in its NURBS edge the quality of the solution is not deteriorated. Details showing the transverse field near the irregularities are represented in Figure 27.

The last example consists on the scattering of an electromagnetic wave by a real aircraft profile of 10 wave lengths. The geometry of this aircraft has several critical zones, in particular, a small irregularity on the upper part and the rear part of this 2D section. Figure 28 shows the computational mesh used for the NEFEM simulation, with only 44 elements on the curved boundary. Some details of the mesh are also represented, showing that it is not necessary to refine the mesh to capture exactly the geometry. Figure 29 shows the transverse field H_3 after 10 cycles and some details near the most critical zones of the aircraft.

It is worth noting that using classical isoparametric FE it is not possible to compute accurate solutions for these problems with the computational meshes used by NEFEM, see Figures 26 and 28. To properly capture the geometry of the domain with isoparametric FE it is necessary to discretize accounting for the the corners nodes (boundary points with only $\mathcal{C}^0$

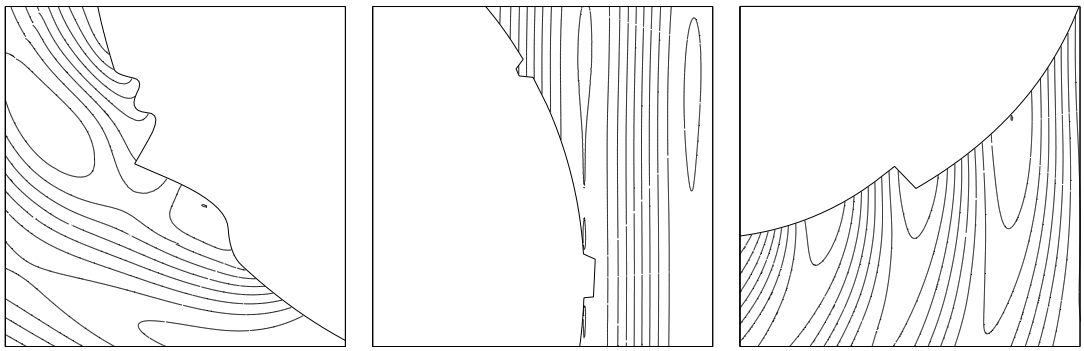


Figure 27. Irregular circle: details of the NEFEM solution near some irregularities.

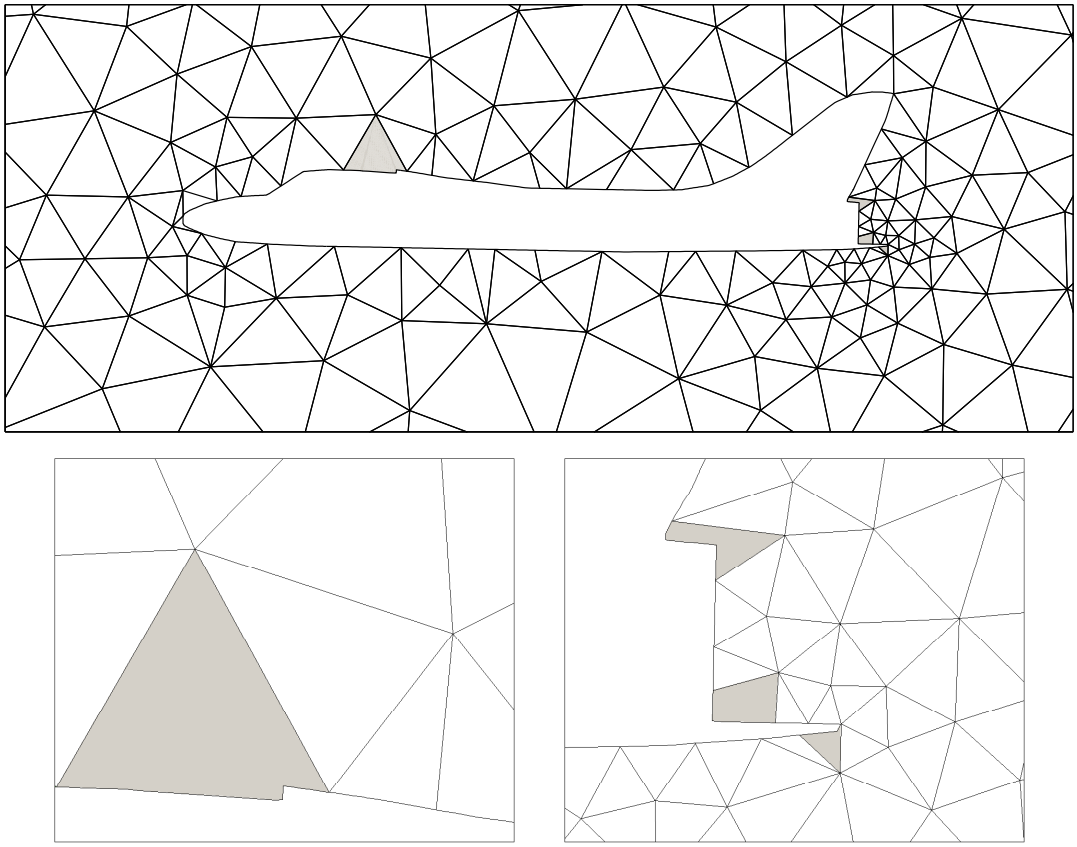


Figure 28. Aircraft profile: details of the NEFEM computational mesh.

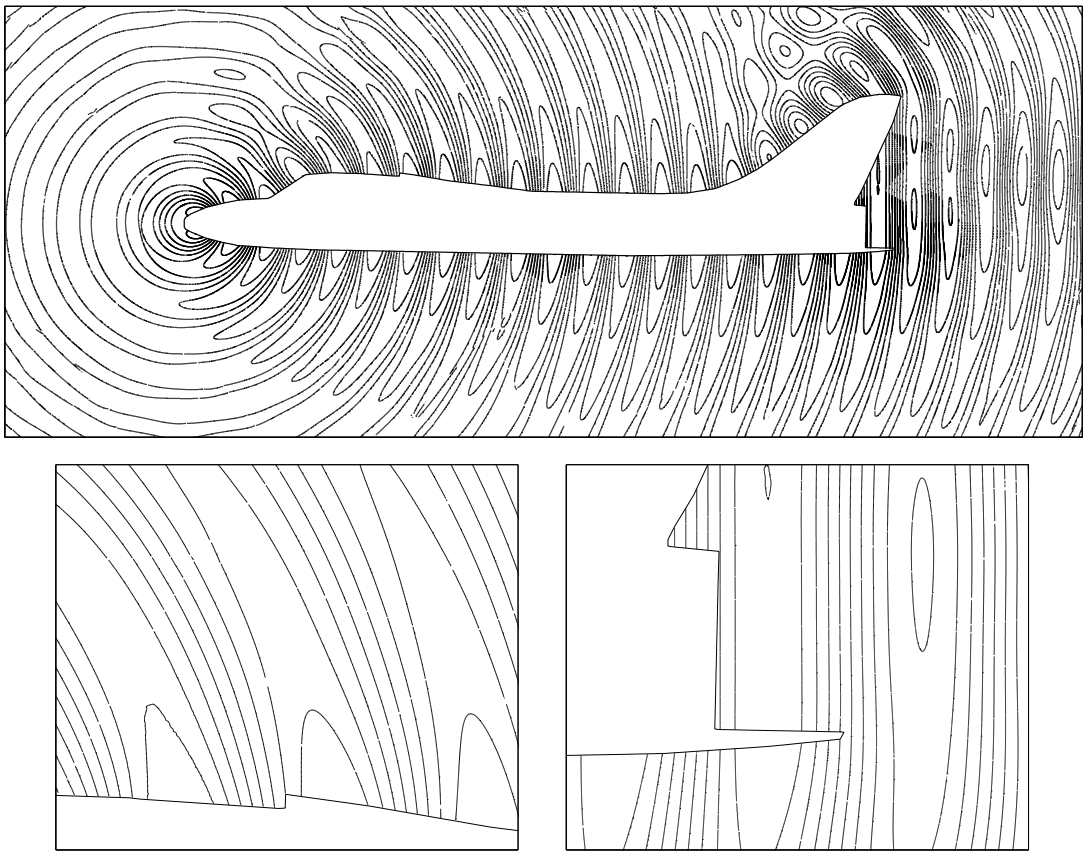


Figure 29. Aircraft profile: details of the NEFEM solution for $p = 6$ (H_3 distribution).

continuity). Thus, the minimum element size is controlled by the size of these irregularities, increasing the number of degrees of freedom in comparison with the NEFEM. For instance, Figure 30 represents a computational mesh adapted to use isoparametric FE. Detailed views near critical zones of the aircraft show that h-refinement is mandatory to properly capture the slope discontinuities in the aircraft profile. The minimum mesh size for the FE mesh in Figure 30 is $3\ 10^{-3}$ whereas the minimum mesh size for the NEFEM mesh in Figure 28 is 0.2. Obviously, this drastic difference between minimum mesh sizes induces important differences in the time-step size when explicit time integrators are used and, therefore, shows another advantage of NEFEM.

5. Concluding remarks

An improvement of standard finite elements is proposed in this work. The exact CAD description of the geometrical model is considered, but only for the boundary of the

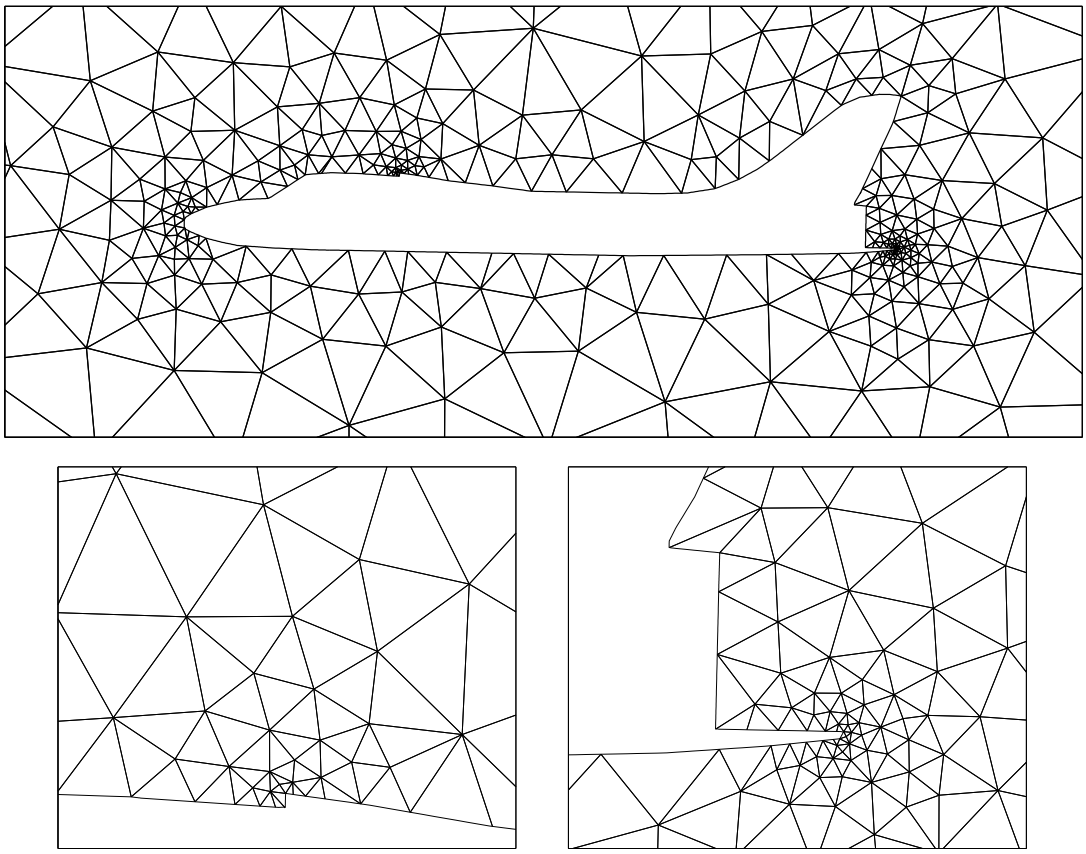


Figure 30. FE mesh around the aircraft profile and detailed view near critical zones

computational domain. At elements intersecting the NURBS boundary specific interpolation and numerical integration are proposed and, at elements not intersecting the boundary, classical finite elements are used, preserving the efficiency of the finite element method. A priori error estimates are given and comments on the implementation of the NURBS-enhanced finite element method (NEFEM) are detailed. It is worth noting that a little effort is needed to enhance a usual finite element code with the NEFEM concept.

Numerical examples demonstrate the advantages of the NEFEM in front of classical isoparametric and cartesian finite elements. A Poisson example demonstrate the applicability of the proposed method in a continuous Galerkin framework. Moreover, it allows to corroborate the a priori error estimates. The numerical results show that, in some cases, NEFEM is four orders of magnitude more precise than the corresponding isoparametric finite element. Some electromagnetic scattering applications are used to show the benefits of the proposed method combined with a Discontinuous Galerkin formulation. Even if the geometry of the boundary of the domain is simple, like a circle, NEFEM is, at least, one order of magnitude more precise than isoparametric finite elements. When the geometric model is complex, like and aircraft profile, the NEFEM is able to compute accurate solutions using coarse meshes. The exact

representation of the boundary allows to mesh the domain independently of the geometric complexity of the boundary whereas classical isoparametric finite elements need h-refinement to properly capture the geometry.

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. T.J. Barth, *Simplified numerical methods for gas dynamics systems on triangulated domains.* PhD thesis, Department of Aeronautics and Astronautics, Stanford University, 1998.
3. X.J. Luo, M.S. Shephard, and J.F. Remacle, "Influence of geometric approximation on the accuracy of higher order methods," *SCOREC report*, vol. 1, 2001.
4. 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, no. 3, pp. 283–300, 2005.
5. 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.
6. 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.
7. B. Cockburn, *Discontinuous Galerkin methods for Computational Fluid Dynamics*, vol. 3 (Fluids) of *Encyclopedia of Computational Mechanics*, ch. 4. John Wiley & Sons, 2004.
8. 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.
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. 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.
12. L. Piegl and W. Tiller, *The NURBS Book*. London: Springer-Verlag, 1995.
13. 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.
14. G.E. Karniadakis and S.J. Sherwin, *Spectral/hp Element methods for CFD (Numerical Mathematics and Scientific computation)*. Oxford: Oxford University Press, 1999.
15. G. Szegö, *Orthogonal Polynomials (fourth edition)*. Providence: Americam Mathematical Society, 1975.
16. M. Dubiner, "Spectral methods on triangles and other domains," *J. Sci. Comput.*, vol. 6, no. 4, pp. 345–390, 1991.
17. 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.
18. 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, no. 1, pp. 89–94, 1996.
19. 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.
20. T. Belytschko, A. Huerta, S. Fernández-Méndez, and T. Rabczuk, *Meshfree methods*, vol. 1 (Fundamentals) of *Encyclopedia of Computational Mechanics*, ch. 10. John Wiley & Sons, 2004.
21. R. Sevilla and S. Fernández-Méndez, "Numerical integration for the NURBS Enhanced Finite Element Method," 2007. Submitted.
22. S. Wandzura and H. Xiao, "Symmetric quadrature rules on a triangle," *Comput. Math. Appl.*, vol. 45, no. 12, pp. 1829–1840, 2003.
23. S.C. Brenner and L.R. Scott, *The Mathematical Theory of Finite Element Methods*. New York: Springer, 1994.
24. P.A. Raviart and J.-M. Thomas, *Introduction à l'analyse numérique des équations aux dérivées partielles*. Paris: Dunod, 1998.
25. C. Johnson, *Numerical solution of partial differential equations by the finite element method*. Cambridge: Cambridge University Press, 1987.

26. B. Szabó and I. Babuška, *Finite Element Analysis*. New York: John Wiley & Sons, 1991.
27. B. Szabó, A. Düster, and E. Rank, *The p-version of the Finite Element Method*, vol. 1 (Fundamentals) of *Encyclopedia of Computational Mechanics*, ch. 5. John Wiley & Sons, 2004.
28. Y. Bazilevs and T.J.R. Hughes, "Weak imposition of Dirichlet boundary conditions in fluid mechanics," *Comp. Fluids.*, vol. 36, no. 1, pp. 12–26, 2007.
29. 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, no. 10, pp. 1803–1818, 1999.
30. R.J. LeVeque, *Numerical methods for conservation laws*. Lectures in Mathematics ETH Zürich, Basel: Birkhäuser Verlag, second ed., 1992.
31. J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*. Chichester: John Wiley & Sons, 2002.
32. D. Givoli, *Numerical Methods for Problems in Infinite Domains*. Amsterdam: Elsevier, 1992.
33. P.D. Ledger, K. Morgan, O. Hassan, and N.P. Weatherill, "Arbitrary order edge elements for electromagnetic scattering simulations using hybrid meshes and a PML," *Int. J. Numer. Meth. Engrg.*, vol. 55, no. 3, pp. 339–358, 2002.
34. C.A. Balanis, *Advanced engineering electromagnetics*. New York: John Wiley & Sons, 1989.
35. C.L. Ladson, C.W. Brooks, A.S. Hill, and D.W. Sproles, "Computer program to obtain ordinates for naca airfoils," Tech. Rep. NASA TM-4741, NASA Langley Research Center, 1996.
36. J. Wu and B. Jiang, "A least-squares finite element mehod for electromagnetic scattering problems," Tech. Rep. ICOMP-96-12, NASA Lewis Research Center, 1996.