Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2008) 7423–7445

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Bézier surfaces and finite elements for MHD simulations

Olivier Czarny, Guido Huysmans *

Association Euratom-CEA, CEA/DSM/Département de Recherches sur la Fusion Contrôlée, Centre de Cadarache,
13108 Saint-Paul lez Durance, France

Received 17 July 2007; received in revised form 1 February 2008; accepted 3 April 2008
Available online 14 April 2008

Abstract

A finite element method based on bicubic Bézier surfaces is applied to the simulation of MHD instabilities relevant to magnetically confined fusion. The major advantage of the new technique is that it allows a natural way to implement mesh refinement strategy, which is not supported by a pure Hermite formulation. Compared to a Lagrangian formulation the number of degrees of freedom is significantly reduced. The use of an isoparametric representation of the space coordinates allows an accurate alignment of the finite elements to the magnetic field line geometry in a tokamak plasma. The Bézier finite elements have been implemented in a MHD code using the non-linear reduced MHD model in toroidal geometry. As an illustration, results for Soloviev equilibrium and time-dependent current-hole computations are presented and discussed.

Keywords: Bézier surfaces; Finite elements; Grid refinement; Non-linear MHD

---

1. Introduction

The understanding of magnetohydrodynamic (MHD) instabilities is quite essential for the optimization of magnetically confined plasmas. For instance, the standard ITER scenario is expected to generate oscillations in the plasma core (*internal kink* instabilities), modes around the outward limit of the plasma confinement zone (*Edge Localized Modes*, or ELMs), or local reconfiguration of the magnetic field topology into closed islands (*tearing* modes) (ITER Group [1]). When occurring, MHD instabilities are likely to degrade confinement properties and to trigger particle bursts that may significantly damage components of tokamak walls. The control of the instabilities is one of the ITER-building challenges (Evans et al. [2], Nardon et al. [3]).

Numerical simulations play an important role in the investigation of the non-linear behaviour of these instabilities and the interpretation of experimental observations. Some features should be emphasized in non-linear MHD code developments. First the MHD instabilities are radially localized and related to strong current sheets; therefore, the possibility to locally increase the number of nodes through mesh refinement

---

[*] Corresponding author. Tel.: +33 4 42 25 6362; fax: +33 4 42 25 6233.
E-mail address: guido.huysman@cea.fr (G. Huysmans).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.04.001

could be helpful. It is also necessary to handle realistic geometries that take into account both the plasma core and the plasma boundary – the "scrape-off layer"-. Moreover, the large gap between the different time scales relevant to plasmas – from the Alfvèn time[1] (10^{-6} s) to the equilibrium time scale (1 s) – makes it necessary for a code to perform integrations within acceptable computing times.

The parallelized 3D MHD code named JOREK in under development as a first step toward this objective (Huysmans [4,5]). The code is based on 3D generalized finite elements that provide some flexibility to the choice of geometries. To date, JOREK solves the reduced non-linear MHD equations, derived from the full MHD model, by means of a linear finite element interpolation in the vertical (poloidal) plane, the periodic toroidal direction being treated via a sine/cosine expansion.

While acceptable for the study of fundamental mechanisms, the reduced MHD model does not prove adequate enough for finer comparisons with realistic experiments in tokamaks. A switch to the full MHD equations is necessary, which subsequently means a significant increase in the number of variables. As memory is the main limiting factor on computing machines, the introduction of new unknowns and new equations must come with an effort to improve the code efficiency. In particular, we aim at optimizing the number of degrees of freedom required to obtain a prescribed level of accuracy. More specifically, we are looking at higher degrees of interpolation coupled to adaptive mesh refinement.

Lagrangian elements are a possible choice, like in the non-linear MHD code NIMROD (Sovinec et al. [6]). These elements are conforming, that is, continuity of order 0 between elements is verified along edges and not only at the nodes of the mesh. With Q3 elements (quadrilateral shape, cubic interpolation), each physical variable needs nine degrees of freedom per node. Providing the same degree of interpolation, bicubic Hermite quadrangles are an interesting alternative, with only four degrees of freedom per node (three of them convey information about derivatives), and continuity for gradients across element boundaries; besides regularity improvement regards, continuity of gradients takes full importance in applying stabilizing methods such as Galerkin *Least Squares techniques*, where access to second-order derivatives is necessary. Bicubic Hermite elements are implemented in the equilibrium code HELENA (Huysmans et al. [7]). On the other hand, these elements are not appropriate to mesh-refinement strategies as the Hermite formulation cannot deal with the parametric change between refined and unrefined elements.

The use of triangular finite elements with C1 continuity has been described in (Jardin [8]). Using a reduced quintic formulation yields three degrees of freedom per node. The applications to some MHD problems confirms the expected convergence rates for the higher order finite elements.

In this paper, we focus our attention on Bézier surfaces, for which Hermite patches turn out to be a special case. Coupled with additional regularity conditions, a finite element problem written under Bézier formalism can still ensure continuity for gradients across refined grids with only four degrees of freedom per node. Finite elements and Bézier interpolation have been previously used in simulations, e.g. for facial surgery (Roth et al. [9]), though in this case, the elements were triangular and the interpolation was subparametric (linear interpolation of the geometry and higher order of interpolation for the physical quantities). Here, we apply an isoparametric mapping, that is, the same order of interpolation for the geometry and the physical unknowns. The isoparametric mapping becomes important when the finite element grid is aligned with the magnetic field line structure in tokamak plasmas.

The paper is organized as follows: Section 2 reviews properties of Bézier surfaces and addresses the problem of continuity between adjacent elements. We give in Section 3 a description of the finite element discretization and establish the number of degrees of freedom required per node; the section ends with details about the mesh refinement method. Finally, after detailing time-stepping and geometric modelling, Section 4 presents two MHD benchmarks: the Soloviev equilibrium and the "current-hole" *internal kink* instability.

2. About Bézier surfaces

Pierre Bézier (1910–1999) worked as an engineer for the French car manufacturer Renault (Bézier [10,11]). In the 1960s, *Computer Aided Manufacturing* (CAM) became more and more important for car drilling and 3D

---

[1] Characteristic time for shear perturbations of the magnetic field to propagate.

milling. In this context, Bézier developed an original method to represent curves with a formalism understandable to the computers controlling the machines. This pioneering work gave birth to the UNISURF Computer Aided Design (CAD) software. Today, Bézier objects are found in most drawing software packages or, even more commonly, in vector fonts such as TrueType or Postscript.

2.1. Definitions and properties

Bézier objects are constructed with a particular family of interpolating polynomials which were first described by the Russian mathematician Sergei Bernstein (1880–1968). Bernstein dedicated part of his work to the theory of best approximation of functions. In 1911 he gave a proof of Weierstrass's theorem – which states that any continuous function defined on a real interval is the uniform limit of a polynomial sequence (Bernstein [12]). The proof, based on a probabilistic approach, calls upon polynomials (B_i^n) defined as:

$$\begin{cases} B_i^n(s) = C_n^i s^i (1-s)^{n-i}, \\ C_n^i = \frac{n!}{i!(n-i)!}, \\ 0 \leqslant i \leqslant n. \end{cases} \tag{2.1}$$

Bernstein polynomials exhibit some properties, e.g.:

- (B_i^n) is a basis of P^n, the vector subspace of polynomials whose degree is lower or equal to n.
- $0 \leqslant B_i^n(s) \leqslant 1$, for each $s \in [0;1]$.
- (B_i^n) is a partition of unity, that is: $\sum_{i=0}^n B_i^n(s) = 1, \forall s \in [0;1]$.

The simplest Bézier objects correspond to curves defined via the parametric representation:

$$P(s) = \sum_{i=0}^{n} P_i B_i^n(s), \quad 0 \leqslant s \leqslant 1, \tag{2.2}$$

where $(P_i) = (x_i, y_i, z_i)$ are control points belonging to the 3D space where coordinates are expressed with respect to the usual orthonormal frame of reference; the curve is physically 3D but one parameter only (the curvilinear coordinate s) is required to locate a point on the curve. In the most widespread cases, degree n is set to 3, yielding the following cubic basis:

$$\begin{cases} B_0^3(s) = (1-s)^3, \\ B_1^3(s) = 3(1-s)^2 s, \\ B_2^3(s) = 3(1-s)s^2, \\ B_3^3(s) = s^3. \end{cases} \tag{2.3}$$

The basis is plotted for illustration in Fig. 2.1.

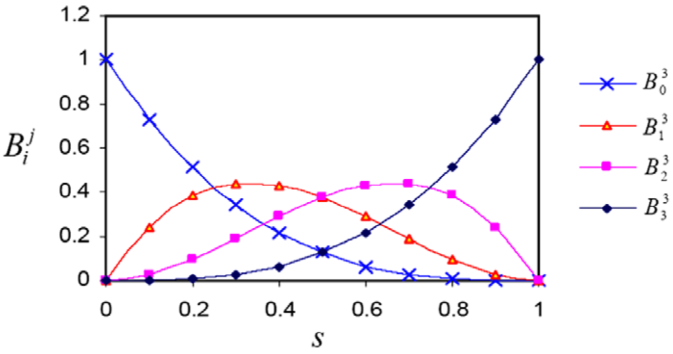

Fig. 2.1. Plots of Bernstein cubic polynomials (B_i^3).

The curve shape depends directly on the location of the control points, as shown on Fig. 2.2 for the particular case where the (P_i) belong to a common 2D plane.

Bézier curves are naturally extendable to bicubic rectangular patches via the next parameterization:

$$P(s,t) = \sum_{i=0}^{3}\sum_{j=0}^{3} P_{i,j}B_i^3(s)B_j^3(t), \quad 0 \leqslant s,\ t \leqslant 1. \tag{2.4}$$

Fig. 2.3(a) displays an example of a bicubic Bézier patch. By switching parameters s and t to 0 or 1 in Eq. (2.4), one can notice quite straightforwardly that edges of the surface are Bézier curves.

The definition of a bicubic Bézier surface involves 16 control points which can be evenly distributed into four sets, each set associated with one of the four corners of the patch. Fig. 2.3(b). zooms around one corner of the surface: apart from the vertex $(P_{0,0})$, one can identify *tangent points* $(P_{1,0}$ and $P_{0,1})$, and a *twist point* $(P_{1,1})$. Denominations of non-vertex points stem from direct relations between the points and parametric derivatives calculated at the vertex. Still considering $P_{0,0}$ corner for instance, these relations – derived from (2.4) – write:

$$\begin{aligned}
\left(\frac{\partial P}{\partial s}\right)_{(s,t)=(0,0)} &= 3(P_{1,0} - P_{0,0}),\\
\left(\frac{\partial P}{\partial t}\right)_{(s,t)=(0,0)} &= 3(P_{0,1} - P_{0,0}),\\
\left(\frac{\partial^2 P}{\partial s \partial t}\right)_{(s,t)=(0,0)} &= 9(P_{1,1} + P_{0,0} - P_{0,1} - P_{1,0}).
\end{aligned} \tag{2.5}$$

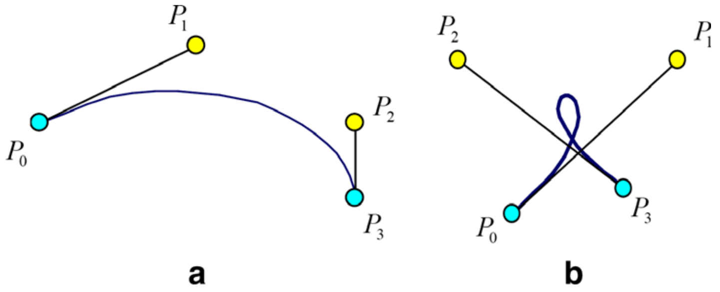

Fig. 2.2. Two examples of Bézier curves, where $P_i \in \mathbb{R}^2, 0 \leqslant i \leqslant 3$.

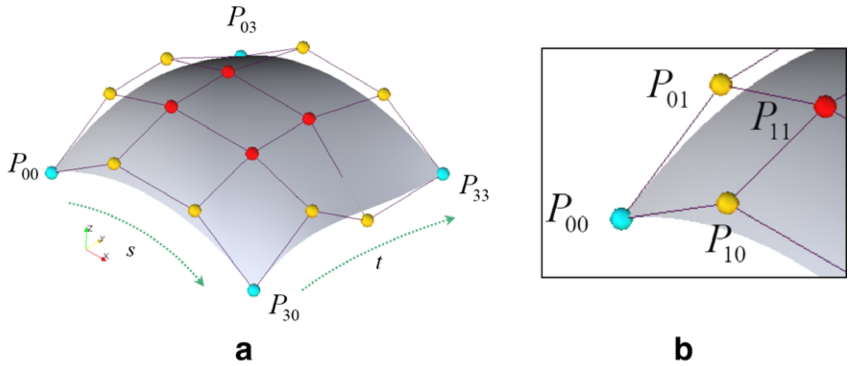

Fig. 2.3. Bicubic Bézier surface, where $P_{ij} \in \mathbb{R}^3$. (a) General view of the surface showing the net of 16 control points; for clarity, only corners are labelled. (b) Close-up around vertex $P_{0,0}$.

In other words, edges $P_{0,0}P_{3,0}$ and $P_{0,0}P_{0,3}$ of the patch are tangent at $P_{0,0}$ to $(P_{0,1} - P_{0,0})$ and $(P_{1,0} - P_{0,0})$, respectively. $P_{1,1}$ as for it is associated with the cross-derivative $(\partial^2 P/\partial s \partial t)$ which plays an important role for continuity across edges, as detailed in Section 2.2.

2.2. Continuity between adjacent Bézier patches

Since the 1970s, many papers have been devoted to the topic of edge continuity (Ueshiba & Roth, [13] Ma & Peng [14], Liang et al. [15], Kiciak [16], De Rose [17]). This issue is quite important for image smoothing applications, it is moreover crucial to ensure some regularity for computed fields within the frame of finite elements simulations.

Let us consider two Bézier surfaces S and S' as depicted on Fig. 2.4. Using parameterization of surface S, the equation of the common edge writes:

$$E(t) = P(1,t) = \sum_{j=0}^{3} P_{3,j} B_j^3(t), \tag{2.6}$$

rewritten as follows with surface S' notation:

$$E'(t') = P'(0,t') = \sum_{j=0}^{3} P'_{0,j} B_j^3(t'). \tag{2.7}$$

A first level of continuity is reached when $E(t) = E'(t')$, $\forall t = t'$. Equating (2.6) and (2.7) and using linear independence of Bernstein polynomials leads to:

$$P_{3,j} = P'_{0,j}, \quad \forall j \tag{2.8}$$

that is: surfaces S and S' share the same control points along the common edge. In this case, continuity is said to be *geometric of order 0* (G^0). The join is equivalently said to be *parametric* (written C^0) as equality of parameters t and t' leads to $P = P'$ on the edge. Through (2.8) we also have equality of the tangent vectors associated to direction t:

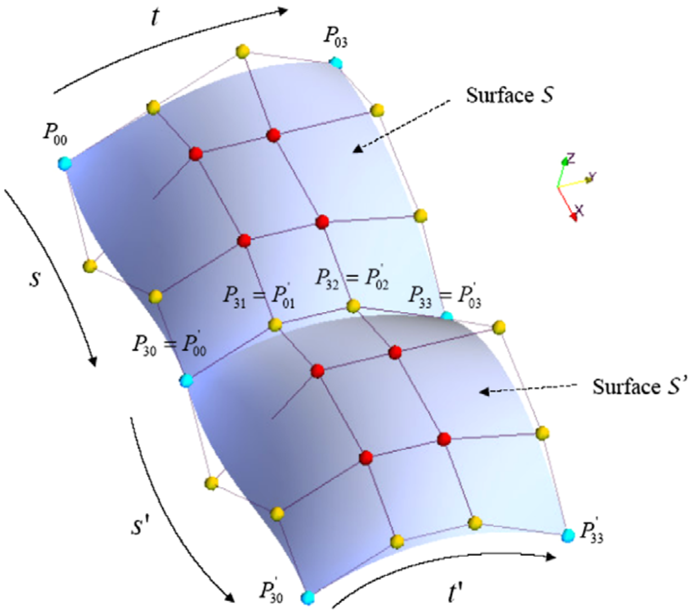

Fig. 2.4. $G^0(C^0)$ continuity between two adjacent Bézier patches. Continuity is achieved when surfaces share the same control points ($P_{3,j}$ for surface S, $P'_{0,j}$ for surface S') along the boundary.

$$\left(\frac{\partial P}{\partial t}\right)_{s=1}(t) = \left(\frac{\partial P'}{\partial t'}\right)_{s'=0}(t'), \quad \forall t = t'. \tag{2.9}$$

The next order of continuity (G^1) needs continuity of surface gradients in addition to G^0 assumptions. This requirement is also known as "*tangent plane condition*" (Ueshiba & Roth [13], De Rose [17]). Let us consider a point $P(=P')$ located on the common boundary $E(=E')$. The plane Π tangent to surface S at P is generated by a basis of two vectors: $(\partial P/\partial s)$ and $(\partial P/\partial t)$. Similarly, the plane Π' tangent to surface S' at $P'(=P)$ is generated by $(\partial P'/\partial s')$ and $(\partial P'/\partial t')$. The join between the patches fulfils the tangent plane condition if $\Pi = \Pi'$ along the edge. As G^0 continuity provides $(\partial P/\partial t) = (\partial P'/\partial t')$ – see (2.9) – the condition reduces to enforcing coplanarity of $(\partial P/\partial s)$, $(\partial P/\partial t)$ and $(\partial P'/\partial s')$ at any point of the common edge: for each $P(t)$ on the edge, one can find real numbers $k_1(t), k_2(t)k_3(t)$ so that:

$$k_1(t)\left(\frac{\partial P}{\partial s}\right) + k_2(t)\left(\frac{\partial P}{\partial t}\right) + k_3(t)\left(\frac{\partial P'}{\partial s'}\right) = 0. \tag{2.10}$$

Various formulations of necessary and sufficient conditions to meet requirement (2.10) can be found in Liang et al. [15], Liu [18]. For the present study, we reduce the practical sufficient condition detailed by Liu [18] to the specific case where control points are aligned across the boundary, as shown on Fig. 2.5. In this setting, the following homothetic relations hold:

$$\exists \lambda_j \in \mathbb{R}/P_{3,j} - P'_{1,j} = \lambda_j P_{2,j} - P_{3,j}. \tag{2.11}$$

Each derivative of Eq. (2.10) are consequently rewritten as:

$$\left(\frac{\partial P}{\partial s}\right)_{s=1} = \sum_{j=0}^{3} 3(P_{3,j} - P_{2,j})B_j^3(t), \tag{2.12}$$

$$\left(\frac{\partial P'}{\partial s'}\right)_{s'=0} = \sum_{j=0}^{3} 3\lambda_j(P_{3,j} - P_{2,j})B_j^3(t), \tag{2.13}$$

$$\left(\frac{\partial P}{\partial t}\right)_{s=1} = \sum_{j=0}^{3} P_{3,j}\frac{\mathrm{d}B_j^3}{\mathrm{d}t}(t). \tag{2.14}$$

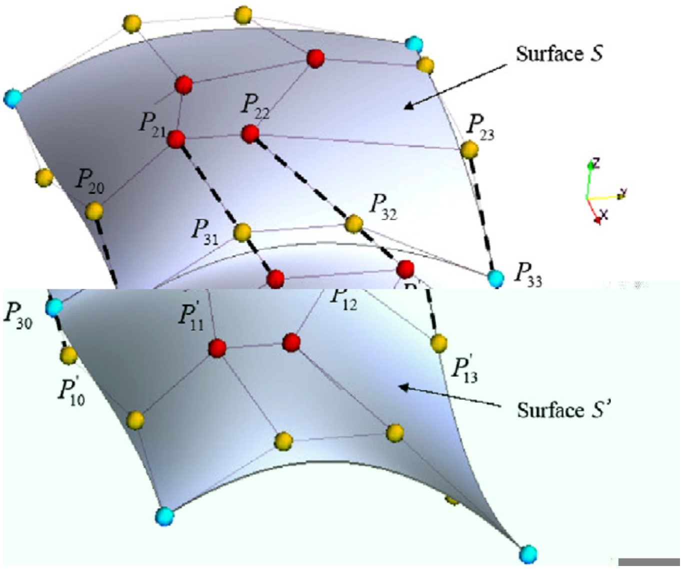

Fig. 2.5. Alignment condition for G^1 continuity between surfaces S and S'. Dashed lines illustrate how points must be aligned across the boundary. Homothetic parameter λ is constant along the boundary ($\lambda = -2$ in this example).

Choosing $k_2(t) = 0$ and invoking again the linear independence of the polynomial basis yields:

$$\lambda_j = \lambda, \quad \forall j \tag{2.15}$$

with λ an arbitrary constant coefficient. Finally, the alignment condition to get G^1 continuity writes:

$$P_{3,j} - P'_{1,j} = \lambda P_{2,j} - P_{3,j} \quad \forall j \in \{0, \ldots, 3\}. \tag{2.16}$$

We retain subsequently this condition for our formulation of the finite element problem. At this point, one may remark that setting $\lambda = 1$ leads to the Hermite formulation. In this case, continuity of order 1 is parametric – quantities (2.5) are continuous across the edge.

3. Finite element method

3.1. Bézier formalism

3.1.1. Degrees of freedom required per node

The problem at stake in the present section is that of a field $\psi(x,y)$ defined over a 2D geometrical domain $\mathcal{D}$ and G^1-interpolated through Bézier surfaces. Let us focus at a corner, where 4 Bézier patches meet (see Fig. 3.1). It is therefore possible to define for the corner a set of 9 control points that can be described in a frame of reference local to the corner.

We consider the parametric surface P as defined in (2.4), taking $P = (x, y, \psi)$ in a Cartesian frame of reference $(0, e_x, e_y, e_\zeta)$. Direction ζ is that of the values for ψ and therefore is not a physical space coordinate like x or y. We shall note that the Bézier interpolation for P apply with the same coefficients to space coordinates (x, y) and field ψ, in this case the finite element method is said to be iso-parametric.

Let $(x_{i,j}, y_{i,j}, \psi_{i,j})$ be the coordinates of $P_{i,j}$, $(i,j) \in \{-1;0;+1\}^2$. We call $M_{i,j}$ the projection of $P_{i,j}$ on plane (O, x, y) (see Fig. 3.1(a)): $M_{i,j} = (x_{i,j}, y_{i,j}, 0)$. The $M_{i,j}$ are the control points of a Bézier grid on which the finite element method will be performed.

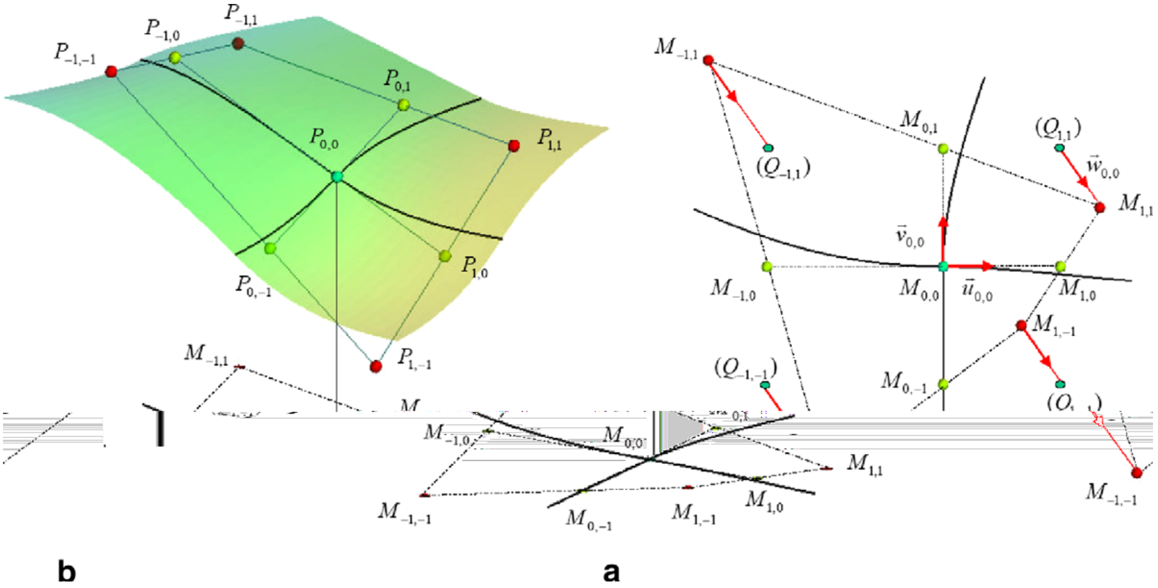

Fig. 3.1. Corner between four Bézier patches with G^1 continuity. (a) 3D view of the surface $\psi(x,y)$. The control points of the surface $(P_{i,j})$ are aligned according to condition (2.16). The thick black lines show the boundaries between the patches. Below the surface, the $M_{i,j}$ are the projection of the $P_{i,j}$ onto plane (x,y). (b) View of plane (x,y) from the top, showing the local vector basis. $Q_{i,j}$ are defined as $Q_{i,j} = M_{i,0} + M_{0,j}$, $i,j = \pm 1$. The twist points $M_{\pm 1,\pm 1}$ are translated from the Q points, with respect to the direction fixed by vector $\vec{w}_{0,0}$.

From that point, we assume that the mesh is given, i.e. the $(x_{i,j}, y_{i,j})$ are known. The nine values $(\psi_{i,j})$ are the degrees of freedom (dof) for ψ at one node. The number of dof can be decreased by adding the G^1-continuity constraint (2.16), which writes – using notations of Fig. 3.1(a):

$$\begin{aligned} &P_{1,j} - P_{0,j} = \lambda_s P_{0,j} - P_{-1,j} \ \forall j \in \{-1;0;1\},\\ &P_{i,1} - P_{i,0} = \lambda_t P_{i,0} - P_{i,-1} \ \forall i \in \{-1;0;1\}. \end{aligned} \tag{3.1}$$

Due to points alignment and using geometrical properties, we can write:

$$\begin{aligned} &\psi_{1,0} - \psi_{0,0} = \alpha_{0,0} d_{u,1} \quad \text{with } d_{u,1} = \|M_{1,0} - M_{0,0}\|,\\ &\psi_{-1,0} - \psi_{0,0} = \alpha_{0,0} d_{u,-1} \quad \text{with } d_{u,-1} = -\|M_{-1,0} - M_{0,0}\|,\\ &\psi_{1,0} - \psi_{0,0} = \lambda_s \psi_{-1,0} - \psi_{0,0}, \end{aligned} \tag{3.2}$$

where $\alpha_{0,0}$ is the tangent of angle $(P_{1,0} - P_{0,0}, M_{1,0} - M_{0,0})$. Similarly:

$$\begin{aligned} &\psi_{0,1} - \psi_{0,0} = \beta_{0,0} d_{v,1} \quad \text{with } d_{v,1} = \|M_{0,1} - M_{0,0}\|\\ &\psi_{0,-1} - \psi_{0,0} = \beta_{0,0} d_{v,-1} \quad \text{with } d_{v,-1} = -\|M_{0,-1} - M_{0,0}\|\\ &\psi_{0,1} - \psi_{0,0} = \lambda_t \psi_{0,-1} - \psi_{0,0} \end{aligned} \tag{3.3}$$

with $\beta_{0,0}$ as the tangent of angle $(P_{0,1} - P_{0,0}, M_{0,1} - M_{0,0})$.

As far as the twist points are concerned, we define $\vec{w}_{i,j} = P_{i,j} + P_{0,0} - P_{i,0} - P_{0,j} (i,j = \pm 1)$. Still referring to geometrical properties, one can prove that:

$$\vec{w}_{i,j} = \frac{d_{u,i}}{d_{u,k}} \vec{w}_{k,j}, \quad \vec{w}_{k,j} = \frac{d_{v,i}}{d_{v,l}} \vec{w}_{k,l}, \quad \forall i,j,k,l = \pm 1, \tag{3.4}$$

yielding the existence at the corner of a vector independent from the local twist points:

$$\frac{1}{d_{u,i} d_{v,j}} \vec{w}_{i,j} = \frac{1}{d_{u,k} d_{v,l}} \vec{w}_{k,l} = \vec{\Gamma}, \quad \forall i,j,k,l = \pm 1. \tag{3.5}$$

Therefore, calling $\gamma_{0,0}$ the third coordinate of $\vec{\Gamma}$, we obtain:

$$\psi_{i,j} + \psi_{0,0} - \psi_{i,0} - \psi_{0,j} = \gamma_{0,0} d_{u,i} d_{v,j} \quad \forall i,j \pm 1. \tag{3.6}$$

Consequently, only four degrees of freedom $(\psi_{0,0}, \alpha_{0,0}, \beta_{0,0}, \gamma_{0,0})$ (instead of nine) are necessary to determine all the control points associated with vertex $P_{0,0}$. This result is not surprising given the equivalence with the bicubic Hermite method (see end of Section 2.2). The extrapolation from the dof. to values other than $\psi_{0,0}$ is performed through the relation summarized below for convenience:

$$\begin{cases} \psi_{i,0} = \psi_{0,0} + \alpha_{0,0} d_{u,i} \\ \psi_{0,j} = \psi_{0,0} + \beta_{0,0} d_{v,j} & \forall i,j \pm 1. \\ \psi_{i,j} = \psi_{i,0} + \psi_{0,j} - \psi_{0,0} + \gamma_{0,0} d_{u,i} d_{v,j} \end{cases} \tag{3.7}$$

We define afterwards a basis of normalized vectors at vertex $P_{0,0}$:

$$\vec{U}_{0,0} = \begin{pmatrix} u_{0,0,x} \\ u_{0,0,y} \\ \alpha_{0,0} \end{pmatrix}, \quad \vec{V}_{0,0} = \begin{pmatrix} v_{0,0,x} \\ v_{0,0,y} \\ \beta_{0,0} \end{pmatrix}, \quad \vec{W}_{0,0} = \begin{pmatrix} w_{0,0,x} \\ w_{0,0,y} \\ \gamma_{0,0} \end{pmatrix}, \tag{3.8}$$

where subvectors $\vec{u}_{0,0}, \vec{v}_{0,0}$ and $\vec{w}_{0,0}$ (see Fig. 3.1(b)) are given by:

$$\begin{cases} \vec{u}_{0,0} = \frac{M_{1,0} - M_{0,0}}{d_{u,1}}, \\ \vec{v}_{0,0} = \frac{M_{0,1} - M_{0,0}}{d_{v,1}}, \\ \vec{w}_{0,0} = \frac{M_{1,1} + M_{0,0} - M_{0,1} - M_{1,0}}{d_{u,1} d_{v,1}}. \end{cases} \tag{3.9}$$

Vector family (3.9) is not linearly independent, as $\vec{u}_{0,0}, \vec{v}_{0,0}$ and $\vec{w}_{0,0}$ are coplanar.

From the numerical point of view, the data structure of the code is such that the basis (3.8) is taken as an intrinsic property of the node that is considered, while the coordinates of $M_{i,j}$ expressed in this basis-that is, $(d_{u,i}, d_{v,j})$ – turn into properties of the element to which the control point belongs. For each node, $(\psi_{0,0}, \alpha_{0,0}, \beta_{0,0}, \gamma_{0,0})$ are the solved unknowns of the problem.

3.1.2. Formulation

We suppose that domain $\mathcal{D}$ is decomposed into N_E quadrilateral cells parameterized with a Bézier approach. We suppose also that all nodes of the mesh are numbered from 1 to N_{nodes}. Let us consider now the element of index K, $1 \leqslant K \leqslant N_E$ (see Fig. 3.2). We define I_K as the ensemble that contains the four indices of the nodes belonging to $E_K{:}I_K = \{i_1, i_2, i_3, i_4\}$. The labelling of the nodes is done counterclockwisely as indicated on the figure.

The basis (3.8) is written at each vertex: Using (2.4), the Bézier parameterization of $P = (x, y, \psi)$ on element E_K can be rewritten as:

$$P(s,t) = \sum_{k=1}^{4} \widetilde{P}_{i_k}(s,t) = \sum_{i=1}^{4} \begin{pmatrix} \tilde{x}_{i_k}(s,t) \\ \tilde{y}_{i_k}(s,t) \\ \tilde{\psi}_{i_k}(s,t) \end{pmatrix}, \tag{3.10}$$

where index i denotes vertex i. The four components $\widetilde{P}_i$are given by:

$$\begin{cases} \widetilde{P}_{i_1}(s,t) = (1-s)^2(1-t)^2[(1+2s)(1+2t)P_{i_1} + 3s(1+2t)d_{u,i_1}U_{i_1} \\ \qquad\qquad +3t(1+2s)d_{v,i_1}V_{i_1} + 9std_{u,i_1}d_{v,i_1}W_{i_1}], \\ \widetilde{P}_{i_2}(s,t) = s^2(1-t)^2[(3-2s)(1+2t)P_{i_2} + 3(1-s)(1+2t)d_{u,i_2}U_{i_2} \\ \qquad\qquad +3t(3-2s)d_{v,i_2}V_{i_2} + 9(1-s)td_{u,i_2}d_{v,i_2}W_{i_2}], \\ \widetilde{P}_{i_3}(s,t) = s^2t^2[(3-2s)(3-2t)P_{i_3} + 3(1-s)(3-2t)d_{u,i_3}U_{i_3} \\ \qquad\qquad +3(3-2s)(1-t)d_{v,i_3}V_{i_3} + 9(1-s)(1-t)d_{u,i_3}d_{v,i_3}W_{i_3}], \\ \widetilde{P}_{i_4}(s,t) = (1-s)^2t^2[(1+2s)(3-2t)P_{i_4} + 3s(3-2t)d_{u,i_4}U_{i_4} \\ \qquad\qquad +3(1+2s)(1-t)d_{v,i_4}V_{i_4} + 9s(1-t)d_{u,i_4}d_{v,i_4}W_{i_4}], \end{cases} \tag{3.11}$$

Each of the $(\widetilde{P}_i)$ detailed in (3.11) can be set into the general form:

$$\widetilde{P}_i(s,t) = b_{i,1}|_K(s,t)P_i + b_{i,2}|_K(s,t)U_i + b_{i,3}|_K(s,t)V_i + b_{i,4}|_K(s,t)W_i, \tag{3.12}$$

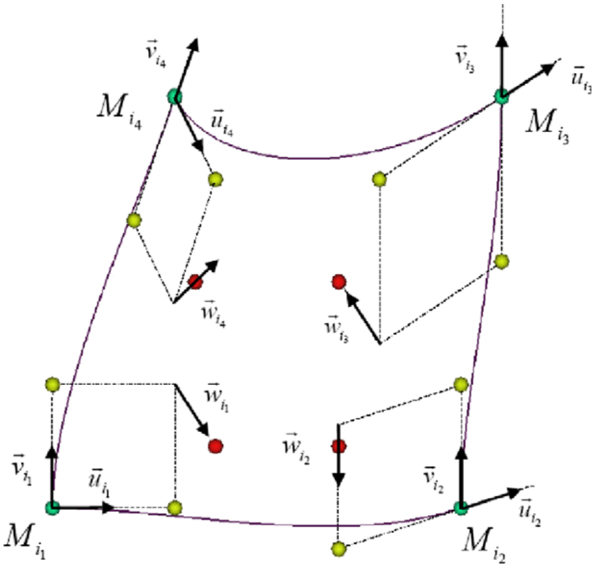

Fig. 3.2. Labelling of nodes and vector basis on element E_K.

where $b_{i,j}|_K(s,t)$ is the *shape function* related to degree of freedom j at node i and restricted to element E_K. The global shape function for dof j of node i is:

$$b_{i,j} = \bigcup_K b_{i,j}|_K \tag{3.13}$$

with $b_{i,j}|_K(s,t) = 0\ \forall j$ if node i does not belong to the element E_K

3.2. The finite element method

Using the space discretization of $\mathcal{D}$ detailed in Section 3.1.2, we illustrate the different stages of the finite element method considering the problem defined below:

$$\begin{cases} \Delta\psi = S & \text{on } \mathcal{D}, \\ \nabla\psi\cdot\vec{n} = 0 & \text{on } \partial\mathcal{D}. \end{cases} \tag{3.14}$$

The standard procedure described in this section can be applied to other types of equations.

3.2.1. Weak form of the problem

The weak form of (3.14) provides an equivalent form of the problem:

$$(3.14) \iff \begin{cases} \int_D f\Delta\psi\mathrm{d}\sigma = \int_D fS\mathrm{d}\sigma \\ \nabla\psi\cdot\vec{n} = 0 \end{cases} \quad \forall f \in H^1(\mathcal{D}) \tag{3.15}$$

with f a "*test function*" belonging to the Sobolev space of functions vanishing at the boundaries and whose derivatives are integrable. The order of derivatives can be lowered through integration by parts; one can show that:

$$\int_D f\Delta\psi\mathrm{d}\sigma = -\int_D \nabla f\cdot\nabla\psi\mathrm{d}\sigma + \int_{\partial D} f\nabla\psi\cdot\vec{n}\mathrm{d}l. \tag{3.16}$$

Taking advantage of the boundary condition, last term in (3.16) is 0, leading to:

$$(3.14) \iff -\int_D \nabla f\cdot\nabla\psi\mathrm{d}\sigma = \int_D fS\mathrm{d}\sigma. \tag{3.17}$$

Problem (3.17) cannot be solved in a general manner. Instead, we want (3.17) to be true for a limited number of functions f. The Bubnow–Galerkin approach (Hirsch [20]) consists in taking the shape functions as the test functions. This method is a straightforward way to get a closed system with as many equations as unknowns.

3.2.2. The assemble stage

We have: $D = \bigcup_K D_K$, where D_K is the subdomain of (O, e_x, e_y) associated with the element E_K. Integrals in (3.17) are evaluated for each E_K separately, requiring a change of variables to display the Bézier parameterization. Let Φ be the diffeomorphism from D_K into the reference square element $[0;1]^2$ and defined as:

$$\begin{aligned} &\Phi : D_K \to [0;1]^2, \\ &(x,y) \mapsto s,t. \end{aligned} \tag{3.18}$$

The Jacobian matrix and determinant of Φ, J_Φ and J, respectively, are given by:

$$J_\Phi = \begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial s} & \frac{\partial y}{\partial t} \end{bmatrix} \quad \text{and} \quad J = \frac{\partial x}{\partial s}\frac{\partial y}{\partial t} - \frac{\partial x}{\partial t}\frac{\partial y}{\partial s}.$$

We write $b^*_{i,j}$ the expression of the shape function $b_{i,j}$ defined in (3.13) as a function of (x,y). Using:

$$\psi(x,y) = \sum_{\substack{i\in I_K \\ 1\leqslant j\leqslant 4}} \psi_{i,j} b^*_{i,j}(x,y),$$

(3.17) yields:

$$-\int_{D_K} \sum_{\substack{i\in I_K\\ 1\leqslant j\leqslant 4}} \psi_{i,j}\nabla b^*_{i,j}\cdot\nabla b^*_{k,l}\,\mathrm{d}x\,\mathrm{d}y = \int_{D_K} b^*_{k,l} S\,\mathrm{d}x\,\mathrm{d}y \tag{3.19}$$

or, switching to the parameter space:

$$-\sum_{\substack{i\in I_K\\ 1\leqslant j\leqslant 4}} \psi_{i,j}\int_{[0;1]^2} \left({}^t\nabla b_{i,j}\big|_K\cdot J_{\Phi^{-1}}\right)\cdot\left({}^t\nabla b_{k,l}\big|_K\cdot J_{\Phi^{-1}}\right)|J|\,\mathrm{d}s\,\mathrm{d}t = \int_{[0;1]^2} b_{k,l}\big|_K S|J|\,\mathrm{d}s\,\mathrm{d}t, \tag{3.20}$$

where $J_{\Phi^{-1}}$, the jacobian matrix of the reciprocal function Φ^{-1}, is given by: $J_{\Phi^{-1}} = \frac{1}{J}\begin{bmatrix} \frac{\partial y}{\partial t} & -\frac{\partial x}{\partial t} \\ -\frac{\partial y}{\partial s} & \frac{\partial x}{\partial s} \end{bmatrix}$ $(J_{\Phi^{-1}}J_{\Phi} = Id$, the identity matrix). (3.20) can be rewritten under the following matrix form:

$$\sum_{\substack{i\in I_K\\ 1\leqslant j\leqslant 4}} A^K_{k,l,i,j}\psi_{i,j} = L^K_{k,l}. \tag{3.21}$$

We sort the degrees of freedom of the system in a list so that dof l of node k ranks at position m of the list, and dof j of node i at position n. The list contains therefore N_{dof} values. (3.21) is equivalent to:

$$\sum_{n=1}^{N_{\mathrm{dof}}} A^K_{m,n}\psi_n = L^K_m. \tag{3.22}$$

Integrals are evaluated by means of a Gauss quadrature:

$$\begin{aligned} \widetilde{A}^K_{m,n} &= \sum_{p,q=1}^{N_{\mathrm{G}}} \omega_p\omega_q A^K_{m,n}(s_p,t_q),\\ \widetilde{L}^K_m &= \sum_{p=1}^{N_{\mathrm{G}}} \omega_p L^K_m(s_p,t_p). \end{aligned} \tag{3.23}$$

N_{G} is the order of the Gauss quadrature into both directions s and t. ω_p, ω_q stand for the Gauss weights and s_p, t_q the Gauss points. Due to the cubic degree of the Bernstein polynomials, one must have $N_{\mathrm{G}} \geqslant 4$. Finally, the procedure is performed over all elements, yielding the following linear system:

$$\sum_{n=1}^{N_{\mathrm{dof}}} \widetilde{A}_{m,n}\psi_n = \widetilde{L}_m, \quad \text{with} \begin{cases} \widetilde{A}_{m,n} = \sum_K \widetilde{A}^K_{m,n},\\ \widetilde{L}_m = \sum_K \widetilde{L}^K_m. \end{cases} \tag{3.24}$$

$\widetilde{A}$ is the "*stiffness matrix* ", and $\widetilde{L}$ the "*load vector*" (names are given with reference to the field of structural analysis from which the finite element method originates).

3.3. "*h*"-refinement and constrained degrees of freedom

The "*h*"-refinement technique consists in the addition of extra nodes either at the beginning of the simulation (simple refinement) or during the simulation itself (adaptive mesh) by subdividing initial element cells (ref). It aims at increasing the accuracy of the space discretization in areas of the computational domain where space scales are insufficiently resolved.

3.3.1. Refining a Bézier patch

We illustrate the refining procedure on Fig. 3.3. The surface labelled S_2 – the "father" surface – is subdivided into subsurfaces $\mathrm{S}'_1, \mathrm{S}'_2, \mathrm{S}'_3, \mathrm{S}'_4$ through two cutting edges whose equations are $P(s=\lambda, t)$ and $P(s, t=\mu)$ respectively, as indicated on the figure; $(\lambda,\mu)\in[0;1]^2$ are arbitrary constants chosen by the user. Doing this, five nodes $(Q_1, Q_2, Q_3, Q_4$ and $Q_5)$ are generated. One interesting feature is that the new surfaces (S'_i) are also Bézier surfaces whose coefficients and vector basis can be calculated from the characteristics of the father surface S_2.

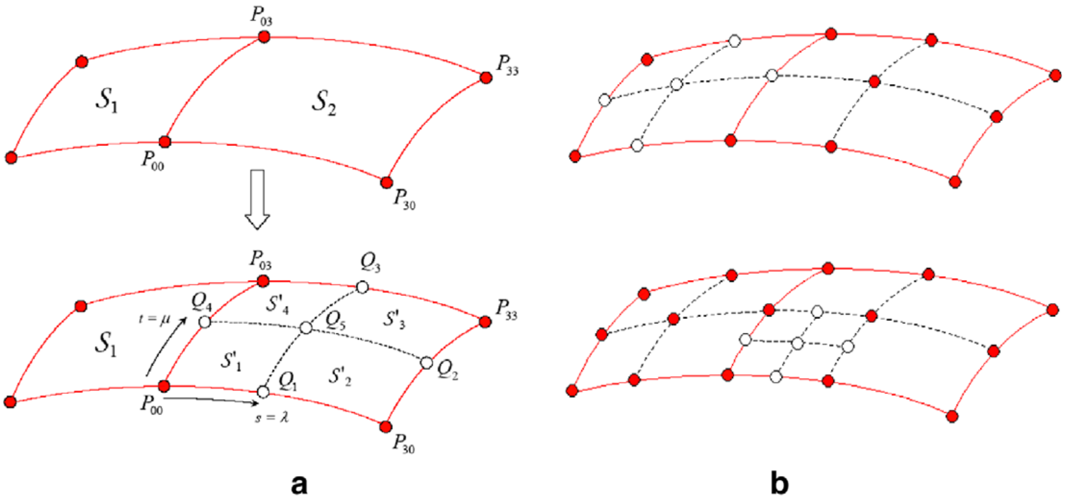

Fig. 3.3. Refinement of a Bézier surface. (a) Initial surface S_2, delimited by vertices $P_{0,0}, P_{3,0}, P_{3,3}$ and $P_{0,3}$, is subdivided into $\mathrm{S}'_1, \mathrm{S}'_2, \mathrm{S}'_3$ and S'_4. (b) to refine S'_1, S_1 is refined first so that Q_4 is no longer a constrained node. Open circles are the new nodes created at the end of one refinement stage.

Let's consider for example point Q_5, whose local coordinates are $(s_5, t_5) = (\lambda, \mu)$. Using S_2 parameterization, we compute the parametric derivatives at Q_5:

$$\left(\frac{\partial P}{\partial s}\right)_{Q_5}, \quad \left(\frac{\partial P}{\partial t}\right)_{Q_5}, \quad \left(\frac{\partial^2 P}{\partial s \partial t}\right)_{Q_5}. \tag{3.25}$$

Reminding that M is the projection of P on (x, y), we define the following lengths:

$$l_{u,Q_5} = \left|\left(\frac{\partial M}{\partial s}\right)_{Q_5}\right| = \sqrt{\left(\frac{\partial x}{\partial s}\right)^2_{Q_5} + \left(\frac{\partial y}{\partial s}\right)^2_{Q_5}}, \quad l_{v,Q} = \left|\left(\frac{\partial M}{\partial t}\right)_{Q_5}\right| = \sqrt{\left(\frac{\partial x}{\partial t}\right)^2_{Q} + \left(\frac{\partial y}{\partial t}\right)^2_{Q}}, \quad l_{w,Q_5} = l_{u,Q_5} l_{v,Q_5}. \tag{3.26}$$

The vector basis defined in (3.8) and corresponding to node Q_5 writes:

$$U_{Q_5} = \frac{1}{3 l_{u,Q_5}} \left(\frac{\partial P}{\partial s}\right)_{Q_5}, \quad V_{Q_5} = \frac{1}{3 l_{v,Q_5}} \left(\frac{\partial P}{\partial t}\right)_{Q_5}, \quad W_{Q_5} = \frac{1}{9 l_{w,Q_5}} \left(\frac{\partial^2 P}{\partial s \partial t}\right)_{Q_5} \tag{3.27}$$

and unit subvectors detailed in (3.9) are:

$$\vec{u}_{Q_5} = \frac{1}{3 l_{u,Q_5}} \left(\frac{\partial M}{\partial s}\right)_{Q_5}, \quad \vec{v}_{Q_5} = \frac{1}{3 l_{v,Q_5}} \left(\frac{\partial M}{\partial t}\right)_{Q_5}, \quad \vec{w}_{Q_5} = \frac{1}{9 l_{w,Q_5}} \left(\frac{\partial^2 M}{\partial s \partial t}\right)_{Q_5}. \tag{3.28}$$

We now compute the Bézier coefficients which must be calculated for each subsurface to which Q_5 belongs, in this case $\mathrm{S}'_1, \mathrm{S}'_2, \mathrm{S}'_3$ and S'_4. Let's suppose that subsurface S'_i is considered. With respect to the labelling on Fig. 3.2, Q_5 corresponds to vertex j $(1 \leqslant j \leqslant 4)$ of S'_i. We denote $d^i_{u,Q_5}, d^i_{v,Q_5}, d^i_{w,Q_5}$ the Bézier coefficients corresponding to directions u, v and w on the subsurface S'_i. These coefficients are given by:

$$d^i_{u,Q_5} = \sigma_u(j) k_u(i) l_{u,Q_5}, \quad d^i_{v,Q_5} = \sigma_v(j) k_v(i) l_{v,Q_5}, \quad d^i_{w,Q_5} = \sigma_u(j) \sigma_v(j) d^i_{u,Q_5} d^i_{v,Q_5} \tag{3.29}$$

(3.29) calls upon some extra functions depending on i and j. The sign functions σ_u and σ_v give the sign of the Bézier coefficients, i.e. the control points coordinates with regards to the unit vectors, and depend on the vertex index j (Fig. 3.2) as follows:

$$\begin{aligned} &\sigma_u(1) = 1, \quad \sigma_u(2) = -1, \quad \sigma_u(3) = -1, \quad \sigma_u(4) = 1, \\ &\sigma_v(1) = 1, \quad \sigma_v(2) = 1, \quad \sigma_v(3) = -1, \quad \sigma_v(4) = -1. \end{aligned} \tag{3.30}$$

The k_u and k_v functions account for the size change from the father surface to the subsurfaces. They depend only on the index i of the subsurface and on (λ, μ), but not on the vertex index j. They are given in Table 1.

The same procedure applies for nodes Q_1, Q_2, Q_3, Q_4.

3.3.2. Constrained degrees of freedom and connectivity matrix

At the end of one refinement stage, some of the created nodes may not provide additional degrees of freedom to the problem. We move back to Fig. 3.3 to illustrate this particular situation. In order to ensure G^1 continuity between $\mathtt{S}_1$, $\mathtt{S}_1'$ and $\mathtt{S}_4'$, values along edges $P_{0,0}Q_4$ or $Q_4P_{0,3}$ (referring to the local description of surfaces $\mathtt{S}_1'$ or $\mathtt{S}_4'$, respectively) must be consistent with that obtained from $\mathtt{S}_1$ with regards to the tangent plane condition. Therefore, values at Q_4 must be prescribed by values at $P_{0,0}$ and $P_{0,3}$. Node Q_4 and corresponding degrees of freedom are said to be *constrained.*

The refinement follows a basic rule based on the algorithm described in Demkowicz et al. [21]: at the end of a refinement stage, any additional node cannot have a constrained node as a parent; if necessary, neighbours of any refined element are refined first to eliminate undesired constrained nodes (Fig. 3.3(b)). This arbitrary choice limits complexity for the handling of constrained nodes during the assemble stage (see next paragraph). It helps also to reach at a faster rate more extra-dof, as – by definition – constrained nodes are useless in increasing the accuracy.

The assemble stage needs special care when constrained nodes are found. If (3.22) is performed on element K which contains constrained nodes, constrained degrees of freedom in ψ are removed and replaced in a new vector ψ' by the parent dof (by construction one of the parents necessarily belongs to element K). Formally the operation writes as a matrix product:

$$\psi = C\psi'. \tag{3.31}$$

The switch between ψ and ψ' is done through the *connectivity matrix C* whose coefficients are provided by the Bézier parameterization.

We illustrate the procedure through the following example. We consider the subsurface $\mathtt{S}_1'$ on Fig. 3.3. We suppose that the only constrained node is Q_4. Contributions of each node of the father patch can be decomposed as in (3.10) and (3.11). The local coordinates of Q_4 being $(s = 0, t = \mu)$. It is easy to notice that only $\widetilde{P}_{0,0}$ and $\widetilde{P}_{0,3}$ contribute to the parametric derivatives.

Using (3.27), and after some algebra, one obtains:

$$\begin{aligned}
\psi_{Q_4} &= (1-\mu)^2[(1+2\mu)\psi_{P_{0,0}} + 3\mu d_{v,P_{0,0}}\beta_{P_{0,0}}],\\
&\quad + \mu^2[(3-2\mu)\psi_{P_{0,0}} + 3(1-\mu)d_{v,P_{0,3}}\beta_{P_{0,3}}],\\
\alpha_{Q_4} &= \frac{1}{3l_{u,Q_4}}[3(1-\mu)^2(1+2\mu)d_{u,P_{0,0}}\alpha_{P_{0,0}} + 9\mu d_{w,P_{0,0}}\gamma_{P_{0,0}}\\
&\quad + 3\mu^2(3-2\mu)d_{u,P_{0,3}}\alpha_{P_{0,3}} + 9(1-\mu)d_{w,P_{0,3}}\gamma_{P_{0,3}}],\\
\beta_{Q_4} &= \frac{1}{3l_{v,Q_4}}[2\mu(\mu-4)\psi_{P_{0,3}} + 3(1-6\mu+\mu^2)d_{v,P_{0,3}}\beta_{P_{0,3}}\\
&\quad + 6\mu(1-\mu)\psi_{P_{0,3}} + 3\mu(1-\mu)\ d_{v,P_{0,3}}\beta_{P_{0,3}}],\\
\gamma_{Q_4} &= \frac{1}{9l_{w,Q_4}}[-18\mu(1-\mu)d_{u,P_{0,0}}\alpha_{P_{0,0}} - 9(1-\mu)(3\mu-1)d_{w,P_{0,0}}\gamma_{P_{0,0}}\\
&\quad + 18\mu(1-\mu)d_{u,P_{0,3}}\alpha_{P_{0,3}} + 9\mu(-3\mu+2)d_{w,P_{0,0}}\gamma_{P_{0,3}}].
\end{aligned} \tag{3.32}$$

Table 1
Size coefficients necessary to switch from the father surface to the subsurfaces; (λ, μ) are the (s, t) coordinates of the dividing edges on the initial patch (see Fig. 3.3)

Sub-surface index i	1	2	3	4
$k_u(i)$	λ	$1-\lambda$	$1-\lambda$	λ
$k_v(i)$	μ	μ	$1-\mu$	$1-\mu$

So that dof of Q_5 depend linearly on the dof of the father nodes $P_{0,0}$ and $P_{0,3}$. Vector ψ contains all the dof on S'_1: nodes $P_{0,0}$, Q_1, Q_5 and Q_4 (we give ψ under transposed shape for writing convenience):

$$\psi={}^t\left(\psi_{P_{0,0}}\quad \alpha_{P_{0,0}}\quad \beta_{P_{0,0}}\quad \gamma_{P_{0,0}}\quad \psi_{Q_1}\quad \alpha_{Q_1}\quad \beta_{Q_1}\quad \gamma_{Q_1}\quad \psi_{Q_5}\quad \alpha_{Q_5}\quad \beta_{Q_5}\quad \gamma_{Q_5}\quad \psi_{Q_4}\quad \alpha_{Q_4}\quad \beta_{Q_4}\quad \gamma_{Q_4}\right).$$

The dof of the first father node $P_{0,0}$ are already present in ψ; dof of node Q_4 are replaced by dof of node $P_{0,3}$, forming the new vector ψ'.

$$\psi'={}^t\left(\psi_{P_{0,0}}\quad \alpha_{P_{0,0}}\quad \beta_{P_{0,0}}\quad \gamma_{P_{0,0}}\quad \psi_{Q_1}\quad \alpha_{Q_1}\quad \beta_{Q_1}\quad \gamma_{Q_1}\quad \psi_{Q_5}\quad \alpha_{Q_5}\quad \beta_{Q_5}\quad \gamma_{Q_5}\quad \psi_{P_{0,3}}\quad \alpha_{P_{0,3}}\quad \beta_{P_{0,3}}\quad \gamma_{P_{0,3}}\right).$$

The matrix C is basically the identity matrix, but rows of C corresponding to the dof of Q_4 (here rows from 13 to 16) are reset and filled in with the right-hand side coefficients in (3.32) so that (3.31) is achieved.

The local stiffness matrix and load vector are modified and (3.22) is rewritten as:

$$A'^K\psi' = L'^K, \quad \text{with } A'^K={}^tCA^KC \quad \text{and} \quad L'^K={}^tCL^K. \tag{3.33}$$

4. Simulations

4.1. Numerical toolbox

4.1.1. Geometry and boundary conditions

Inside a tokamak, the general shape of the plasma is that of a torus of major radius $R_{O'}$ (Fig. 4.1). The meridian section – also named poloidal section – is characterized by an ellipticity $e = b/a$, where a and b are the horizontal and vertical minor radii respectively. The section boundary is either circular or given via an analytical solution of the Soloviev equilibrium problem to allow D-shaped configurations (see Section 4.2, equation (4.16)). We define the aspect ratio as $\varepsilon = a/R_{O'}$. For the need of some simulations (see current hole, Section 4.3), the configuration may be that of a cylinder, which can be considered as a degenerated torus with $R_{O'} \to +\infty$; in this case, $\varepsilon = 0$.

A point in the plasma is usually located with respect to a toroidal frame of reference $(\vec{e}_x, \vec{e}_y, \vec{e}_\phi)$: the Cartesian coordinates (R,Z) apply for the poloidal section while the toroidal direction is labelled with angle ϕ. R is the distance from the revolution axis of the torus (see Fig. 4.1). Further on the space coordinates (R, Z) are changed into dimensionless coordinates (x, y) centered in the plasma:

$$x = \frac{R - R_{O'}}{a} = \frac{R/R_{O'} - 1}{\varepsilon}, \quad x \in [-1; +1], \quad y = \frac{Z}{a}, \quad y \in \left[-\frac{b}{a}; \frac{b}{a}\right]. \tag{4.1}$$

The MHD equations are scaled with the major radius R_O.

For all simulations, walls are supposed fixed and perfectly conductible: variables are consequently set to 0 at boundaries.

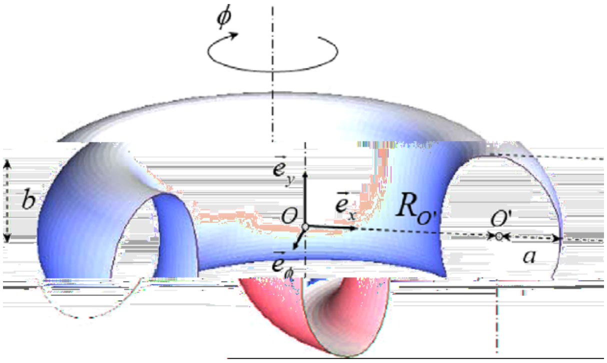

Fig. 4.1. Toroidal configuration of a tokamak. $R_{O'} = OO'$ stands for the big radius of the torus, a and b the horizontal and vertical minor radii in the poloidal section.ϕ is the toroidal angle.

4.1.2. Time-stepping

Both steady equilibrium and unstationary non-linear MHD equations are tested in the present paper (Sections 4.2 and 4.3).

A wide range of time scales (with several orders of magnitude) are found in time-dependent MHD problems, from the Alfvèn time to the equilibrium time. This may lead to quite prohibitive computing times. In order to surmount this difficulty, the time-stepping is handled with a fully implicit Crank–Nicholson approach which makes it possible to use large time-steps. We write the system of equations under the following general form:

$$\frac{\partial A}{\partial t} = B(\xi, t). \tag{4.2}$$

A and B are vectors of length l, where l is the number of independent equations. $\xi = (\xi_i)$ are the physical unknows of the problem. A and B can depend non-linearly on ξ and t. Treated via the Crank–Nicholson scheme, (4.2) becomes:

$$\delta A^n = A^{n+1} - A^n = \frac{\delta t}{2}[B^{n+1} + B^n], \tag{4.3}$$

where δt is the time-step, and superscript n denotes a value at time iteration n. Implicity of the time scheme coupled to non-linearity of A and B lead usually to performing an additional sub-iterative method. This can be avoided through a linearization technique which offers the advantage of keeping the second-order accuracy in time, as proposed by Beam and Warming (Hirsch [20]). The key point is the linearization at first-order of A and B with respect to ξ:

$$H^{n+1} - H^n = H(\xi^{n+1}, t^{n+1}) - H(\xi^n, t^n) \approx \left(\frac{\partial H}{\partial \xi}\right)^n \cdot \delta\xi^n \tag{4.4}$$

with $H = A$ or B, $\delta\xi^n = \xi^{n+1} - \xi^n$, and $\left(\frac{\partial H}{\partial \xi}\right)^n$ the jacobian matrix of H with respect to variables ξ. Scheme (4.3) writes finally:

$$\left[\left(\frac{\partial A}{\partial \xi}\right)^n - \frac{1}{2}\delta t\left(\frac{\partial B}{\partial \xi}\right)^n\right] \cdot \delta\xi^n = \delta t B^n. \tag{4.5}$$

The system (4.4) is solved via the solvers MUMPS (Amestoy et al. [22]) or PaStiX (Hénon et al. [23]). These parallel libraries are specially adapted to handle large sparse matrices as those which are intrinsically generated by the finite element discretizations.

4.1.3. Meshing

From a general standpoint, the meshes used in this paper are polar grids. We denote N_r the number of points into the radial direction, while N_θ is the number of points into the polar direction. The grid shows N_r concentric closed contours (the center is included in this count), and N_θ radial line segments. The polar coordinates (r, θ) of node M_i writes:

$$M_i = \begin{pmatrix} r_i \cos\theta_i \\ r_i \sin\theta_i \end{pmatrix}. \tag{4.6}$$

Let's look at a Bézier cell. In the case of a circular section, we identify local and global coordinates by assuming $r = f(t)$, $\theta = \theta s)$, where (s, t) are the surface parameters. Thus:

$$\frac{\partial M}{\partial s} = f(t)\theta'(s)\begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix}; \quad \frac{\partial M}{\partial t} = f'(t)\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}; \quad \frac{\partial^2 M}{\partial s \partial t} = \theta'(s)f'(t)\begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix}. \tag{4.7}$$

This writing displays directly the unit subvectors defined in (3.9) that are required by the Bézier description:

$$\vec{u} = \vec{w} = \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix}; \quad \vec{v} = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}. \tag{4.8}$$

For a circular point distribution (see Fig. 4.2), the curves between M_{i1} and M_{i4}, or M_{i2} and M_{i3}, are line segments, while $M_{i1}M_{i2}$ and $M_{i3}M_{i4}$ are circle arcs. One requirement here is that r (respectively θ) must be a smooth function of the parameter t (respectively s) in order to avoid distortions of the local mapping.

The control points of $[M_{i_1}M_{i_4}]$ (or $[M_{i_2}M_{i_3}]$) line segment are aligned. Setting the Bézier coefficients to:

$$\begin{aligned} d_{u,i_1} &= \frac{1}{3}\left\|M_{i_4} - M_{i_1}\right\|; \quad d_{u,i_4} = -\frac{1}{3}\left\|M_{i_4} - M_{i_1}\right\|, \\ d_{u,i_2} &= \frac{1}{3}\left\|M_{i_3} - M_{i_2}\right\|; \quad d_{u,i_3} = -\frac{1}{3}\left\|M_{i_3} - M_{i_2}\right\|, \end{aligned} \tag{4.9}$$

leads to a point on the segment depending linearly on parameter t:

$$\begin{aligned} M(t) &= M_{i_1} + (M_{i_4} - M_{i_1})t \quad \text{on } \lfloor M_{i_1}M_{i_4} \rfloor, \\ M(t) &= M_{i_2} + (M_{i_3} - M_{i_2})t \quad \text{on } \lfloor M_{i_2}M_{i_3} \rfloor. \end{aligned} \tag{4.10}$$

As a circle arc is analytically defined as the square root of a polynomial, it cannot be decomposed on the Bernstein basis which can deal only with polynomials of integer degree ranging from 0 to 3. Anyway it can be approximated with quite a convenient accuracy using the following formulas (Goldapp [24]):

$$\begin{aligned} d_{v,i_1} &= \frac{4}{3}\tan\left(\frac{\Delta\theta}{4}\right); \quad d_{v,i_2} = -\frac{4}{3}\tan\left(\frac{\Delta\theta}{4}\right), \\ d_{v,i_4} &= \frac{4}{3}\tan\left(\frac{\Delta\theta}{4}\right); \quad d_{v,i_3} = -\frac{4}{3}\tan\left(\frac{\Delta\theta}{4}\right), \end{aligned} \tag{4.11}$$

with $\Delta\theta = \theta_{i_2} - \theta_{i_1} = \theta_{i_3} - \theta_{i_4}$. Let's consider the circle arc M_kM_l of radius r, and M a point on its Bézier approximation. It is shown (Goldapp [24]) that in this case, the relative error δ_E, defined as:

$$\delta_E = \max_{0 \leqslant s \leqslant 1} \left| \frac{\mathrm{OM}^2(t) - r^2}{r^2} \right| \tag{4.12}$$

is equal to:

$$\delta_E = \frac{4}{27} \frac{\sin^6\left(\frac{\Delta\theta}{4}\right)}{\cos^2\left(\frac{\Delta\theta}{4}\right)}. \tag{4.13}$$

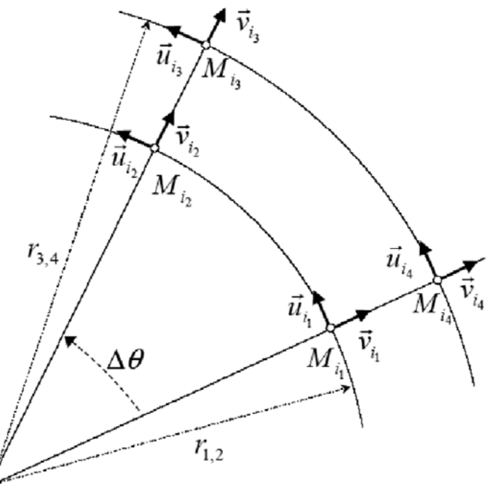

Fig. 4.2. Definition of a cell for a circular mesh. Please note that here $\vec{w}_i = \vec{u}_i, i = i_1, i_2, i_3$ or i_4.

As an example, one obtains $\delta_E \sim 8.5 \times 10^{-6}$ for $\Delta\theta = \pi/4$, and $\delta_E \sim 3.5 \times 10^{-8}$ for $\Delta\theta = \pi/10$. This shows that the error due to the Bézier approximation is quite acceptable given the range of $\Delta\theta$ values, – usually lower than $\pi/4$ – that are used for the simulations.

For the mesh generation and for the sake of simplicity, we apply $\lambda_s = \lambda_t = 1$ in condition (3.1) over the entire grid, which makes the elements Hermite-like; but the "hermicity" is broken when applying the refining algorithm. Of course, one could have chosen a pure Bézier distribution instead.

Non-circular meshes are based on the Soloviev solution described in part 4.2 (Huysmans et al. [7]). The analytical equation of the domain boundary is obtained by fixing $\psi(x, y) = 0$ in (4.17). In this case, the boundary is θ-dependent, making the expression of $r(s, t)$ more complicated. In one Bézier cell, M_{i1} and M_{i2} lie on the jth closed contour, while M_{i3} and M_{i4} lie on the $(j + 1)$th contour. The radius function is chosen as follows:

$$r(s,t) = \sum_m r_m \chi_j^m(t) e^{im\theta(s)}. \tag{4.14}$$

χ_j is a normalized radius defined by: $\chi_j(t) = (j - 1 + t)/(N_r - 1)$. To treat the polar direction, θ is chosen to be a linear function of $s : \theta = \theta_{i_1} + s\Delta\theta$. The coefficients r_m are obtained by considering elements on the boundary $(j = N_r - 1,\ t = 1)$ and adjusting (4.14) to the equation of the Soloviev profile.

The choice of (4.14) allows contours to morph smoothly from Soloviev boundary shape to circles when getting closer to the center (Fig. 4.3(a)). This choice stems from physical observations reporting a similar behaviour of the flux surfaces.

Unit vectors and Bézier coefficients are straightforwardly deduced from the parametric derivatives of M_i, using (4.6) and (4.14)b.

Similarly to the circular grid, the Soloviev mesh is initially generated so that the elements are Hermite-like, and once again, the refining procedure breaks the hermicity when used.

4.2. Soloviev equilibrium

In this part, the plasma is described by the monofluid ideal magnetohydrodynamics equations under the assumption of axisymmetry and stationarity. The Grad–Shafranov equation models a plasma equilibrium where the pressure gradient balances the Lorentz force (Biskamp [25], Friedberg [26], Wesson [27]) and writes in dimensioned form:

$$\Delta^*\tilde{\psi} = -\left(R^2 \frac{d\tilde{p}}{d\tilde{\psi}} + \widetilde{F}\frac{d\widetilde{F}}{d\tilde{\psi}}\right), \quad \text{with } \Delta^*\tilde{\psi} = R^2 \operatorname{div}\left(\frac{1}{R^2}\operatorname{grad}\tilde{\psi}\right), \tag{4.15}$$

where $\tilde{\psi}$ is the magnetic flux taken as a radial coordinate, $\tilde{p}$ the pressure and $\widetilde{F} = R\widetilde{B}_\phi$, with $\widetilde{B}_\phi$ the toroidal component of the magnetic field. The Soloviev equilibrium derives from (4.15) when adding conditions (4.16):

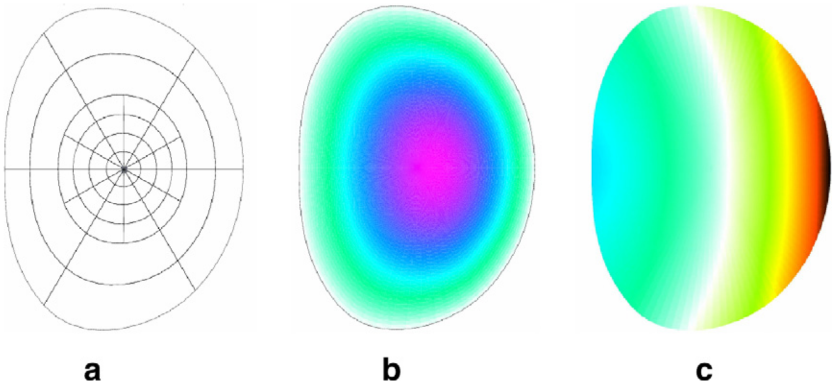

Fig. 4.3. Contour plots in a poloidal section, with $\lambda = 0, b/a = 1.4$, $e = 0.3$. (a) the refined grid; (b) magnetic flux ψ; and (c) $\partial\psi/\partial x$.

$$\frac{\mathrm{d}\tilde{p}}{\mathrm{d}\tilde{\psi}} = \mathrm{Cst}, \quad \widetilde{F}\frac{\mathrm{d}\widetilde{F}}{\mathrm{d}\tilde{\psi}} = \mathrm{Cst}. \tag{4.16}$$

The Soloviev equation admits a class of analytical solutions which, under the assumption of "$z = 0$" planar symmetry, can be written under dimensionless form as (Soloviev [28]):

$$\psi(x,y) = \left(x - \frac{1}{2}\varepsilon(1 - x^2)\right)^2 + \left(\left(1 - \frac{1}{4}\varepsilon^2\right)(1 + \varepsilon x)^2 + \lambda x\left(1 + \frac{1}{2}\varepsilon x\right)\right)\frac{y^2}{\left(\frac{b}{a}\right)^2}. \tag{4.17}$$

In addition to the ellipticity e, the plasma shape is also controlled here by a triangularity parameter λ. (x, y) are the coordinates in the poloidal plane normalized as described in Section 4.1.1. Setting $\psi(x, y) = 0$ in (4.17) leads to the equation that defines the boundary of the numerical domain.

To test the geometric smoothness of the solutions, it is necessary to avoid a purely Hermite grid (cf Section 4.1.3); to this purpose we chose arbitrarily some elements to refine (Fig. 4.3(a)). Fig. 4.3(b) shows the magnetic flux ψ and (c) the components of grad ψ in the poloidal plane for the special case (λ 0, $b/a = 1.4$, ε0.3). From the regularity of the plotted quantities across the grid – no matter if elements are refined or not – it is clear that the numerical computation is successful in providing G^1 continuity.

We define the error $E(\psi)$ as the maximum difference encountered over the domain between the analytical and the computed solutions ψ. It is shown (Strang & Fix [19]) that the error on the sth derivative of ψ scales as:

$$h^{p+1-s} + h^{2p+2-2m}, \tag{4.18}$$

where h is the characteristic length of one element, p the order of the approximation functions and $2m$ the order of the solved equation. In our problem, $p = 3$ (cubic functions) and the operator Δ^* in (4.15) is second-order (leading to $2m = 2$). Consequently, $E(\psi)$ should scale as h^4, and $E(\partial\psi/\partial x)$ or $E(\partial\psi/\partial y)$ as h^3.

Fig. 4.4 shows $E(\psi)$ and $E(\partial\psi/\partial x)$ as a function of $N = 1/h = N_r - 1$, the number of intervals along the radial direction. The grid is regular.

Numerical results match perfectly the theoretical expectations: cubic elements provide h^4-convergence rate for the magnetic flux (Fig. 4.4(a)), while gradients are approximated by one order of magnitude less (Fig. 4.4(b)). We carried out similar computations using bi-linear elements; the corresponding error curves are included on the plots. Consistently with (4.18), convergence rates for linear interpolation are lower than for cubic elements by two orders of magnitude.

It is noteworthy that for a very course grid ($N = 3$), Bézier elements give $E(\psi) = 5.56 \times 10^{-2}$, while linear elements provide $E(\psi) = 5.66 \times 10^{-1}$ only. We take advantage here of the ability of Bézier curves to fit the geometry of the flux surfaces better than linear elements.

4.3. Current hole

In this part, we consider the time-dependent reduced resistive MHD equations (Biskamp [25], Briguglio et al. [29]). Supposing invariance along the toroidal direction ϕ, the dimensionless non-linear equations write:

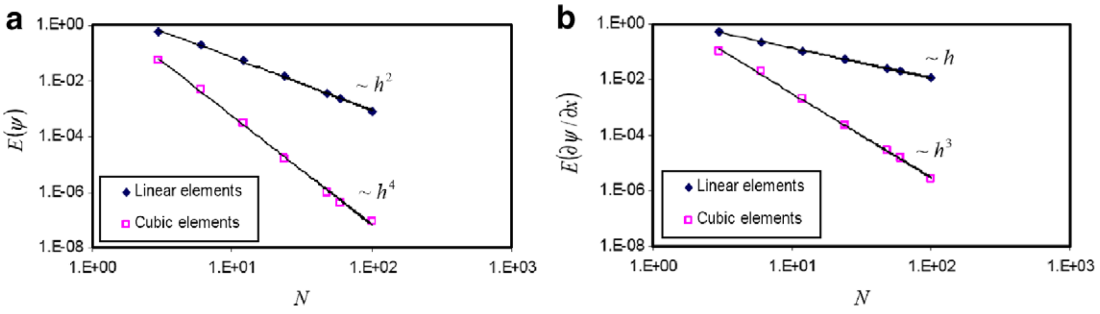

Fig. 4.4. Accuracy E on (a) ψ and (b) $\partial\psi/\partial x$, as a function of the number of radial intervals N. Both figures, drawn with log/log axis, show comparison between linear and cubic elements.

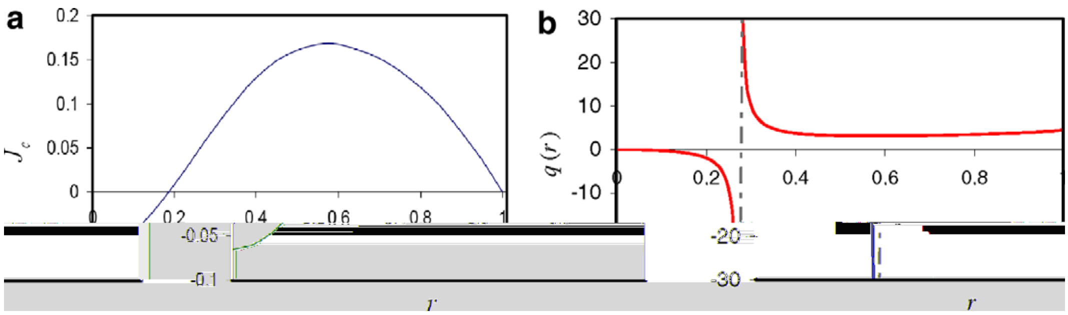

Fig. 4.5. (a) Plot of radial profile for the driven current density given by (4.20), with $j_1 = 0.2$ and $j_2 = 0.266$. r is the radius normalized by the minor radius a; and (b) corresponding profile of the safety factor q; the dashed vertical line marks the radial location of the "$q = \infty$" resonant surface.

$$\begin{cases} \frac{\partial\psi}{\partial t} = (1+\varepsilon x)[\psi,\varphi] + \eta(J - J_c), \\ \frac{\partial\varphi}{\partial t} = 2\varepsilon\frac{\partial\varphi}{\partial y}\omega + (1+\varepsilon x)[\omega,\varphi] + \frac{1}{1+\varepsilon x}[\psi,J] + \nu\Delta_\perp\omega, \\ J = \Delta^*\psi, \\ \omega = \Delta_\perp\varphi, \end{cases} \tag{4.19}$$

three new unknowns are added to ψ: velocity potential φ, toroidal current density J and vorticity ω. $[\cdot,\cdot]$ is the Poisson bracket[2], and Δ^* the Grad–Shafranov operator defined in (4.15). $\Delta_\perp$ is the laplacian restricted to the poloidal section. η and ν stand for the resistivity and the viscosity respectively. The non-ohmic driven current density J_c sets a constant current profile likely to be perturbed with fluctuations. Attention should be paid on that in equations (4.19), J does not stand for the real current which is defined as $J' = R\,J = (1+\varepsilon x)J$. The toroidal component of the magnetic field is assumed to be constant (B_0) and of leading order compared to the poloidal component $B_\perp$.

For the need of further comparisons with analytical work, the geometry is taken cylindrical ($\varepsilon = 0$). Notice that in this case, $\Delta^* = \Delta_\perp$ and the pseudo-current J is equal to the real current as $R = 1$. Consistently with [30] we chose the following radial profile for J_c.:

$$J_c = j_1(1 - r^4) - j_2(1 - r^2)^8. \tag{4.20}$$

Fig. 4.5-plots $J_c(r)$ for $j_1 = 0.2$ and $j_2 = 0.266$. The profile exhibits a zone of negative current density close to the axis – namely the "current hole"-. We take the Grad–Shafranov equilibrium based on current profile (4.20) as the initial condition for the time-dependent simulations: thus $J = J_c$ at $t = 0$.

The safety factor q is related to the pitch of the magnetic field lines and corresponds to the poloidal angle covered by the field line during one toroidal round (Freidberg [26]). For a cylindrical geometry, the safety factor is given by:

$$q(r) = \frac{2\pi r B_0}{L_0 B_\perp}, \tag{4.21}$$

where L_0 is the axial length of the cylinder. From Fig. 4.5(b), one can see that q takes negative values inside a '$q = \infty$' surface which includes the current hole itself. The resonant surface corresponds to $q = m/n$, with the toroidal wave number n equal to 0 because of the axisymmetry. The m=1 internal kink instability is responsible for sawtooth crashes in tokamaks (Wesson [27]).

Fig. 4.6 shows snapshots of the 3D current profile at different stages of the internal kink evolution. The current surface, initially axisymmetric (Fig. 4.6(a)), remains identical during the linear stage in the early part of the transient, where non-linear effects are still negligible. At the end of this stage, the profile is progressively deformed: current density is expelled outward from the central axis, generating a current sheet at the resonant surface (Fig. 4.6(b)). At the end of the crash, the profile close to the center is flattened (Fig. 4.6(c)), leaving

---

[2] With the Poisson bracket defined as: $[a,b] = \frac{\partial a}{\partial x}\frac{\partial b}{\partial y} - \frac{\partial a}{\partial y}\frac{\partial b}{\partial x}$.

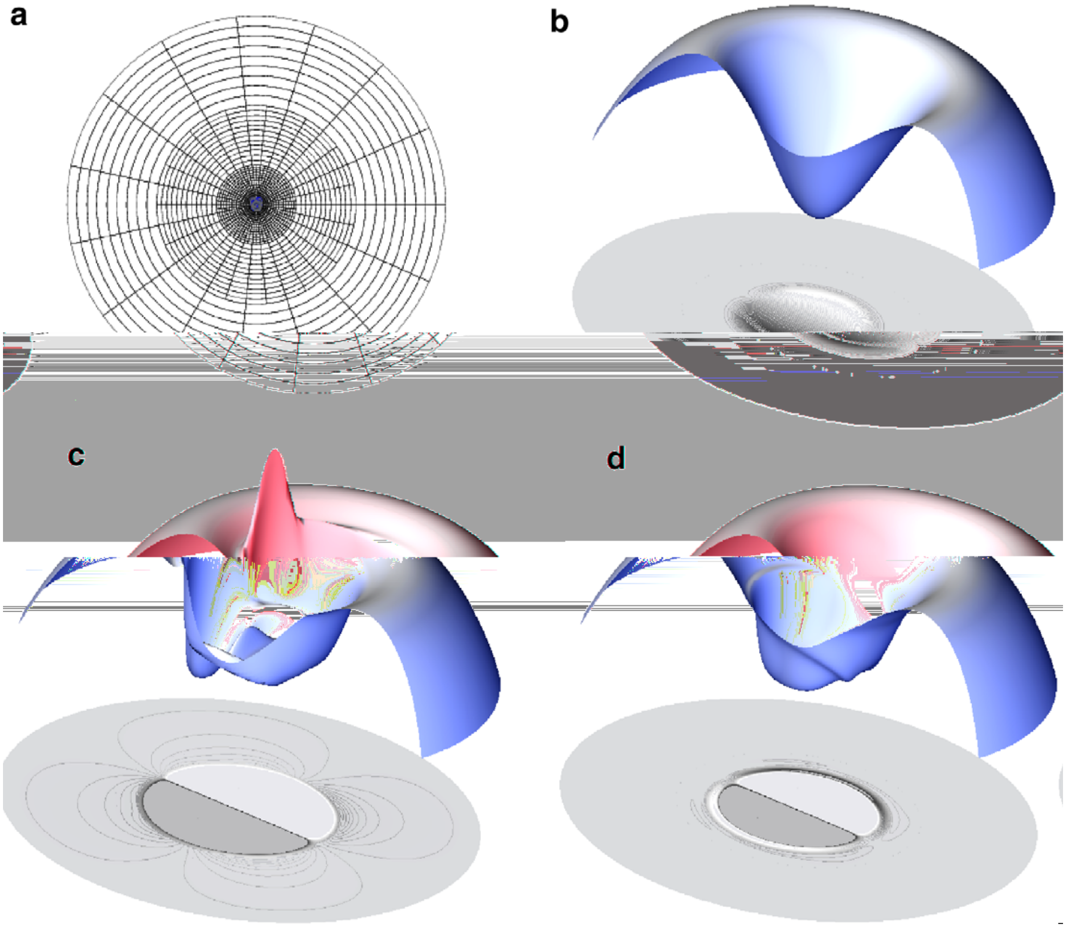

Fig. 4.6. Internal resistive kink in a cylindrical tokamak, with $\eta = 10^{-5}$ and $\nu = 10^{-6}$. The figure plots current profile as a 3D surface clipped for clarity. The cylindrical section below the current surface displays isocontours of velocity potential φ; (a) refined grid; (b) Initial axisymmetric current profile given in Fig. 4.5, $t = 0$; (c) $t = 5.9 \times 10^3\ \tau_A$; and (d)$t = 6.9 \times 10^3 \tau_A$. ($\tau_A$ is the Alfvén time).

only residual fluctuations around $J = 0$. The semi-log plot of the plasma kinetic energy (Fig. 4.7) clearly shows the linear growth stage of the kink mode, overtaken by non-linearities close to $t = 6 \times 10^3$.

As a validation test, we measured the growth rate γ of the $m = 1$ mode during the linear stage. Fig. 4.8 plots γ as a function of resistivity η. The behaviour of the curve when η tends to 0 is asymptotically that of $\eta^{1/3}$, as obtained in [30]. This result is in accordance with theoretical expectations (Wesson [27] or Ara et al. [31]): it is shown that an approximation of the growth rate is $\tau_H^{-2/3}\tau_R^{-1/3}$, where τ_H is the Alfvén time, $\tau_R = (\mu_0 r_0^2/\eta)$ the resistive diffusion time, and r_0 the radial location of the '$q = \infty$' surface. This makes clear the 1/3-dependence on resistivity.

We define $\widetilde{E}(\gamma)$ as the error between γ measured for N intervals and its asymptotic value when N tends to ∞. Fig. 4.9(a)) shows that $\widetilde{E}(\gamma)$ varies as h^2 for linear interpolation while Bézier elements provide h^6 convergence. Bézier technique proves to enhance accuracy even on evaluating growth rates, which are dynamic quantities.

At that point, one question that comes up is the computational cost required to reach a given accuracy. The CPU time required per iteration to reach a particular $\widetilde{E}(\gamma)$ is given on Fig. 4.9(b)). $t_{\mathrm{CPU}} \sim t^{-3}$ (Bézier) and $t_{\mathrm{CPU}} \sim t^{-1}$ (linear). From the figure, one can see that when both curves are defined for one value of $\widetilde{E}(\gamma)$, Bézier elements are cheaper in giving this error level.

The current sheet thickness scales as $\eta^{1/3}$ (Biskamp [25]), which means that at low resistivities, the thickness of the resistive layer is quite small. Through mesh adaptivity, it is possible to increase the mesh density around

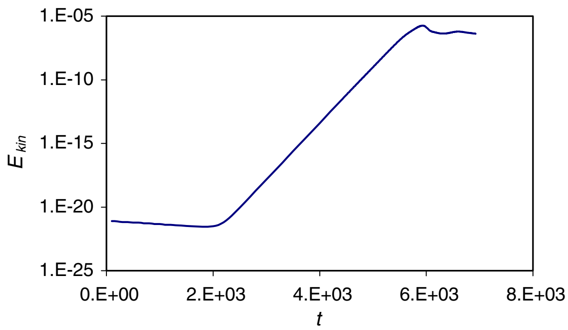

Fig. 4.7. Kinetic energy of the plasma with $\eta = 10^{-5}$ and $\nu = 10^{-6}$.

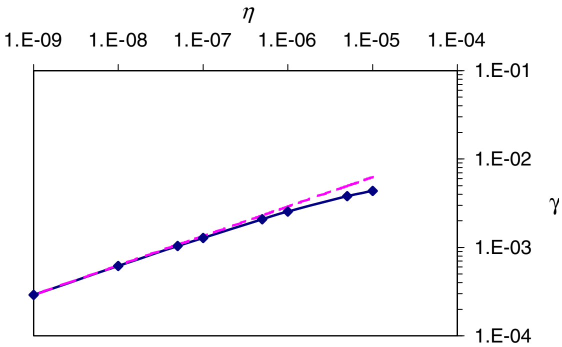

Fig. 4.8. Growth rate γ of $m = 1$ resistive kink mode vs. the resistivity η. The dashed line indicates the $\eta^{1/3}$ asymptote.

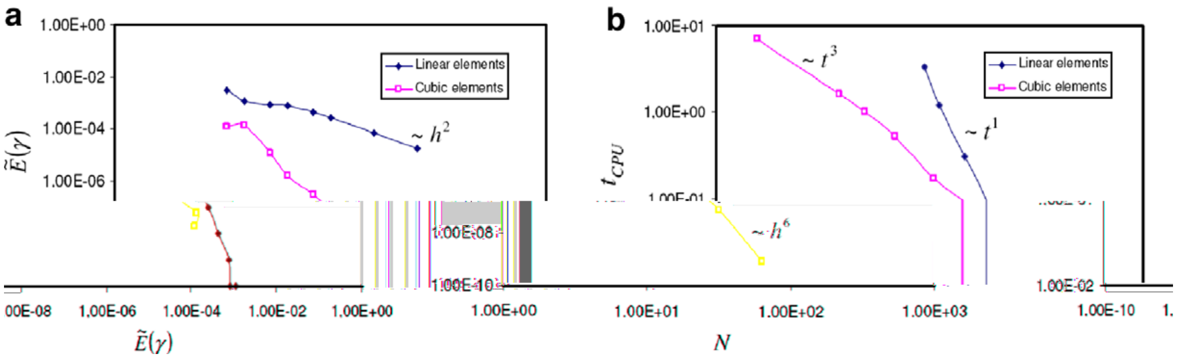

Fig. 4.9. Comparison between linear and cubic elements: (a) error on growth rate γ as a function of N; and (b) CPU time per iteration vs. error on growth rate.

the current layer during the simulation itself, as depicted on Fig. 4.10. The figure shows a wave of current density propagating toward the resonant surface. When the current gradient is strong enough, a secondary instability grows inside the layer under the shape of a tearing bubble, which is progressively swept away into the longitudinal direction (camber line) of the sheet. The occurrence of a'tearing' bubble is consistent with previous results of Biskamp [32] who derives a rough criterion from linear stability considerations: the aspect ratio A being defined as the length of the sheet divided by its thickness, the layer becomes unstable if $A > 100$ approximately. Though we report instability for values of A lower than 100, probably because for the considered regimes, non-linearities are strong enough to upset linear stability predictions [32].

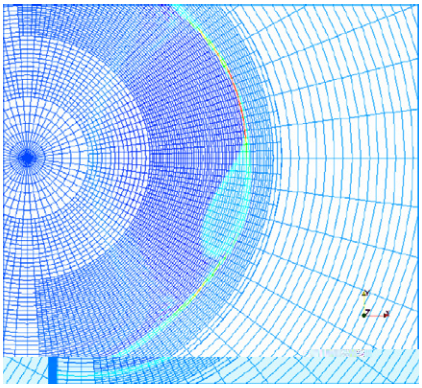

Fig. 4.10. Close-up around the resonant surface, $\eta = 10^{-7}$. The mesh density is increased to allow a sufficient description of the tearing mode that occurs in the current sheet.

5. Conclusion

The present paper describes a finite element method using Bézier rectangles. This development appeared as a necessary step to the enhancement of the code JOREK, adapted to realistic tokamak configurations and previously based on linear interpolating functions. The switch to bicubic Bernstein polynomials provides a fourth-order accuracy in space while requiring four degrees of freedom per node and per physical variable as in the Hermite approach. Nonetheless, the method provides continuity of the gradients between the elements, a large flexibility for the element shapes and allows for mesh refinements.

The method was applied to the solving of the reduced resistive non-linear MHD equations. Two axisymmetric cases have been studied: the academic Soloviev equilibrium in a torus, and the time-dependent problem of a current hole in a cylindrical geometry. In all cases, results are consistent with theoretical expectations. For the current hole experiment, we report a sixth order space accuracy in the measurements of the growth rates – while of only second-order with linear interpolation. Mesh refinement allows the tracking of secondary tearing modes in the current sheet resulting from the internal kink crash.

The 2D discretisation described above can easily be extended to 3D toroidal geometry by using a Fourier expansion in the periodic direction. Alternatively an extension of the Bézier description to 3 space coordinates can also be envisaged. The next step will be the implementation of the full compressible MHD model and the construction of a Bézier finite element mesh aligned with the magnetic geometry of plasmas with an X-point.

Acknowledgements

Parts of the simulations were carried out on the clusters of the Centre de Calcul Recherche et Technologie, CEA/Bruyères-Le-Châtel -. We thank the technical team for its precious support. Thanks also to Patrick Maget (CEA/DRFC), for fruitful theoretical discussions.

References

[1] ITER Physics Expert Group on Disruptions, Plasma Control, and MHD and ITER Physics Basis Editors. Chapter 3: MHD stability, operational limits and disruptions. Nuclear Fusion 39(12) (1999) 2251–2389.

[2] T.E. Evans, R.A. Moyer, K.H. Burrell, M.E. Fenstermacher, I. Joseph, A.W. Leonard, T.H. Osborne, G.D. Porter, M.J. Schaffer, P.B. Snyder, P.R. Thomas, J.G. Watkins, W.P. West, Edge stability and transport control with resonant magnetic perturbations in collisionless tokamak plasmas, Nature Physics 2 (2006) 419–423.

[3] E. Nardon, M. Bécoulet, G. Huysmans, O. Czarny, P.R. Thomas, M. Lipa, R.A. Moyer, T.E. Evans, G. Federici, Y. Gribov, A. Polevoi, G. Saibene, A. Portone, A. Loarte, Edge localized modes control by resonant magnetic perturbations. Journal of Nuclear Materials in press.

[4] G.T.A. Huysmans, ELMs: MHD instabilities at the transport barrier, Plasma Physics and Controlled Fusion 47 (2005) B165–B178.

[5] G.T.A. Huysmans, External kink (peeling) modes in X-point geometry, Plasma Physics and Controlled Fusion 47 (2005) 2107–2121.

[6] C.R. Sovinec, A.H. Glasser, T.A. Gianakon, D.C. Barnes, R.A. Nebel, S.E. Kruger, D.D. Schnack, S.J. Plimpton, A. Tarditi, M.S. Chu, NIMROD Team, Nonlinear magnetohydrodynamics simulation using high-order finite elements, Journal of Computational Physics 195 (2004) 355–386.

[7] G.T.A. Huysmans, J.P. Goedbloed, W. Kerner, Isoparametric bicubic Hermite elements for solution of the Grad–Shafranov equation, in: Proceedings of the CP90 – Conference on Computational Physics (1991), World Scientific Publishing Co., p. 371.

[8] S.C. Jardin, A triangular finite element with first-order continuity applied to fusion MHD applications, Journal of Computational Physics 200 (2004) 133–152.

[9] S.H.M. Roth, M.H. Gross, S. Turello, F.R. Carls. A Bernstein–Bézier approach to soft tissue simulation, in: Proceedings of Eurographics'98, Computer Graphics Forum, vol. 17–3, 1998, pp. 285–294.

[10] P. Bézier, Petite histoire d'une idée bizarre (1), Bulletin de la section d'Histoire des Usines Renault, Tome 4, Juin 1982, N°24, pp. 256–268.

[11] P. Bézier, Petite histoire d'une idée bizarre (2), Bulletin de la section d'Histoire des Usines Renault, Tome 4, December 1982, N°25, pp. 319–331.

[12] S.N. Bernstein, Démonstration du théorème de Weierstrass, fondée sur le calcul des probabilités, Communications of the Society of Mathematics, Kharkow (2) 13 (1912–1913) 1–2.

[13] T. Ueshiba, G. Roth, Generating smooth surfaces with bicubic splines over triangular meshes, in: Proceedings of the Second International Conference on Recent Advances in 3-D Digital Imaging and Modeling, 1999, pp. 302–311.

[14] L. Ma, Q. Peng, Smoothing of free-form surfaces with Bézier patches, Computer Aided Geometric Design 12 (1995) 231–249.

[15] Y. Liang, X. Ye, S. Fang, G^1 smoothing solid objects by bicubic Bézier patches, in: Eurographics Conference Proceedings, Computer Graphic Forum, vol. 7, 1988, pp. 343–355.

[16] P. Kiciak, Constructions of G^1 continuous joins of rational Bézier patches, Computer Aided Geometric Design 12 (1995) 283–303.

[17] T. De Rose, Necessary and sufficient conditions for tangent plane continuity of Bézier surfaces, Computer Aided Geometric Design 7 (1990) 179–195.

[18] D. Liu, GC1 continuity conditions between two adjacent rational Bézier surface patches, Computer Aided Geometric Design 7 (1990) 151–163.

[19] G. Strang, G.J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, 1973.

[20] C. Hirsch, Numerical computation of internal and external flows, Fundamentals of numerical discretization, vol. 1, John Wiley & Sons, 1988.

[21] L. Demkowicz, K. Gerdes, C. Schwab, A. Bajer, T. Walsh, HP90: A general and flexible Fortran 90 hp-FE code, Computing and Vizualization in Science 1 (1998) 145–163.

[22] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, Multifrontal parallel distributed symmetric and unsymmetric solvers, Computer Methods in Applied Mechanics and Engineering 184 (2000) 501–520.

[23] P. Hénon, P. Ramet, J. Roman, PaStiX: a high-performance parallel direct solver for sparse symmetric definite systems, Parallel Computing 28 (2) (2002) 301–321.

[24] M. Goldapp, Approximation of circular arcs by cubic polynomials, Computer Aided Geometric Design 8 (1991) 227–238.

[25] D. Biskamp, Nonlinear Magnetohydrodynamics, Cambridge Monographs on Plasma Physics, Cambridge University Press, 1993.

[26] J.P. Freidberg, Ideal Magnetohydrodynamics, Plenum Press, New York, 1987.

[27] J. Wesson, Tokamaks, Clarendon Press, Oxford, 2004.

[28] L.S. Soloviev, in: M.A. Leontovitch (Ed.), Reviews of Plasma Physics, 6, New York, Consultant Bureau, 1975, p. 257.

[29] S. Briguglio, F. Zonca, G. Vlad, Hybrid magnetohydrodynamic-particle simulation of linear and nonlinear evolution of Alfvén modes in tokamaks, Physics of Plasmas 5 (1998) 3287–3301.

[30] G.T.A. Huysmans, T.C. Hender, N.C. Hawkes, X. Litaudon, MHD stability of advanced tokamak scenarios with reversed central current: an explanation of the "current hole", Physical Review Letters 87 (24) (2001) 245002.

[31] G. Ara, B. Basu, B. Coppi, Magnetic reconnection and $m = 1$ oscillations in current carrying plasmas, Annals of Physics 112 (1978) 443–476.

[32] D. Biskamp, Magnetic reconnection via current sheets, Physics of Fluids 29 (5) (1986) 1520–1531.