

UKAEA-CCFE-PR(22)17

S.J.P.Pamela G.T.A.Huijsmans M.Hoelzl

# **A Generalised Formulation of G-continuous Bezier Elements Applied to Non-linear MHD Simulations**

Enquiries about copyright and reproduction should in the first instance be addressed to the UKAEA Publications Officer, Culham Science Centre, Building K1/O/83 Abingdon, Oxfordshire, OX14 3DB, UK. The United Kingdom Atomic Energy Authority is the copyright holder.

The contents of this document and all other UKAEA Preprints, Reports and Conference Papers are available to view online free at [scientific-publications.ukaea.uk/](https://scientific-publications.ukaea.uk/)

# **A Generalised Formulation of G-continuous Bezier Elements Applied to Non-linear MHD Simulations**

S.J.P.Pamela G.T.A.Huijsmans M.Hoelzl



# A Generalised Formulation of G-continuous Bezier Elements Applied to Non-linear MHD Simulations

S.J.P.Pamela<sup>1</sup>, G.T.A.Huijsmans<sup>2,3</sup>, M.Hoelzl<sup>4</sup> and the JOEREK Team<sup>5</sup>

*EUROfusion Consortium, JET, Culham Science Centre, Abingdon, OX14 3DB, UK.*

<sup>1</sup>*CCFE, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK.*

<sup>2</sup>*CEA, IRFM, F-13108 Saint-Paul-lez-Durance, France.*

<sup>3</sup>*Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands.*

<sup>4</sup>*Max-Planck Institute for Plasma Physics, 85748 Garching, Germany.*

<sup>5</sup>*See author list of M Hoelzl et al, Nucl. Fusion (submitted); [<https://arxiv.org/abs/2011.09120>].*

*Corresponding author: Stanislas Pamela, stanislas.pamela@hotmail.com*

## Abstract

As the international tokamak ITER is being built, non-linear MHD simulations are playing an essential role in active research, understanding, and prediction of tokamak plasmas for the realisation of a fusion power plant. The development of MHD codes like JOEREK is a key aspect of this research effort, and provides invaluable insight into the plasma stability and the control of global and localised plasma events, like Edge-Localised-Mode and disruptions. In this paper, we present an operational implementation of a new, generalised formulation of Bezier finite-elements applied to the JOEREK code, a significant advancement from the previously C1-continuous bi-cubic Bezier elements. This new mathematical method enables any polynomial order of Bezier elements, with a guarantee of C-continuity at the level of  $(n - 1)/2$ , where  $n$  is the order of the Bezier polynomials. The generalised method is defined, and a rigorous mathematical proof is provided for the C-continuity requirement. Key details on the code implementation are mentioned, together with a suite of tests to demonstrate the mathematical reliability of the finite-element method, as well as the practical usability for typical non-linear tokamak MHD simulations. A demonstration for a state-of-the-art simulation of an Edge-Localised-Mode instability in the future ITER tokamak, with complex grid geometry, finalises the study.

## 1 Introduction

Industrial electricity production using nuclear fusion power would greatly contribute to the reduction of greenhouse gas emissions and of long-lived radioactive nuclear waste, while providing electricity to society without the limit of an exhaustible natural resource. A favorable candidate

for industrial fusion reactors is the tokamak device. Tokamaks use a helical magnetic field that winds itself around a toroidal vacuum chamber. The periodic nature of the torus ensures that charged particles, which approximately follow magnetic field lines, are not lost at the end of open field lines like in linear plasma devices. However, this periodicity can be subject to resonance and instabilities. Large-scale instabilities typically involve both the plasma and the magnetic field, and are often studied in the fluid picture using magnetohydrodynamics (MHD).

There is a wide variety of MHD instabilities in tokamak plasmas, some of which can reduce or limit the operational capabilities of the machine. Edge-Localised-Modes (ELMs) are instabilities that eject plasma from the confined region onto the material surfaces of the first wall of the machine; these instabilities can lead to large heat-fluxes that may reduce the life-time of the material surfaces [1–4]. Toroidal Alfvén Eigenmodes (TAEs) can be excited by the 3.5MeV alpha-particles born from fusion reactions; these can significantly limit the plasma pressure, and are a concern for future reactors where the burning plasma will produce a large amount of alpha-particles [5–8]. Global MHD instabilities, during which the entire plasma is affected, can lead to disruptions; during such events the kinetic and magnetic energy of the plasma can be transferred to the wall, leading to material heat-fluxes and/or wall-currents that can damage the machine and its structure [9–14]. In order to study, understand and predict these instabilities, non-linear MHD simulations are performed using numerical codes like JOEAK [15–18], M3D-C1 [19, 20], NIMROD [21, 22], XTOR [23], BOUT++ [24, 25], MEGA [26–28], HALO [7] and many others.

There is a wide range of finite-element methods used in the community of fusion modelling. The use of C1-continuous finite-elements has proven essential for the numerical stability of many codes, in particular JOEAK [16, 18] and M3D-C1 [19, 20]. C1-continuity is particularly essential when dealing with physics models that include high-order spatial derivatives, typically order-2 derivatives for diffusive terms like viscosity, resistivity and particle/heat diffusion, as well as order-4 derivatives for hyper-diffusion terms. In addition, the magnetohydrodynamic equations are based on an equilibrium between two very large pressure terms: the kinetic pressure  $\nabla p$ , and the magnetic pressure  $\vec{J} \times \vec{B}$ . At equilibrium, these two terms in the momentum equation cancel each other out, but when dealing with instabilities, it is precisely this balance that breaks, leading to spatially finely localised differences arising from the  $\nabla p - \vec{J} \times \vec{B}$  imbalance. In this highly-nonlinear environment, precision is key, such that high-order finite-element methods can provide definite advantages.

In this paper, we present a generalised method for high-order Bezier finite-elements, which ensure that G-continuity increases linearly with the polynomial order of the elements. The method is defined, proven, and implemented in the JOEAK code with several numerical tests and benchmark cases. Note that in the remainder of this paper the term C-continuity is often used for simplicity since it is more frequently used in literature, but in all instances, this means G-continuity. Section-2 introduces the C2-continuous Bezier finite-elements, which serves as an introduction to the generalised method. Section-3 presents the definition the generalised Bezier formulation, together with the mathematical proofs to demonstrate C-continuity. Section-4 describes some of the key details that were required for the implementation of this new method in the JOEAK code, and Section-5

presents the series of tests conducted to demonstrate the mathematical accuracy of the new method, together with the practical usability and advantages of using higher-order finite-element methods for non-linear MHD instabilities in tokamaks. Finally Section-6 summarises the work and lays out the further improvements required for future studies of tokamak instabilities.

## 2 C2-continuous Bi-Quintic Bezier Elements

### 2.1 Bezier curves and Bezier elements

Iso-parametric Bezier curves of order  $n$  are described by the formulation of any function (including spatial coordinates) as

$$F(s) = \sum_{i=0}^n B_i(s)P_i \quad (1)$$

Where  $s$  is the element's local coordinate, the points  $P_i$  are the control points of the curve, and the Bernstein polynomials are defined as

$$B_i(s) = \frac{n!}{i!(n-i)!} s^i (1-s)^{n-i} \quad (2)$$

This is represented for Bezier segments of order 3 and 5 in Figure-1. For cubic curves, the end control points  $P_0$  and  $P_3$  determine the position of the segment vertices (or nodes), while the vectors  $[P_0, P_1]$  and  $[P_3, P_2]$  control the first derivatives (or directions) of the segments. Cubic Bezier segments can have up to two inflexion points. For quintic curves, the end control points  $P_0$  and  $P_5$  determine the position of the segment vertices, the first vectors  $[P_0, P_1]$  and  $[P_5, P_4]$  control the first derivatives, and the second vectors  $[P_1, P_2]$  and  $[P_4, P_3]$  control the second derivatives. Quintic Bezier segments can have up to four inflexion points.

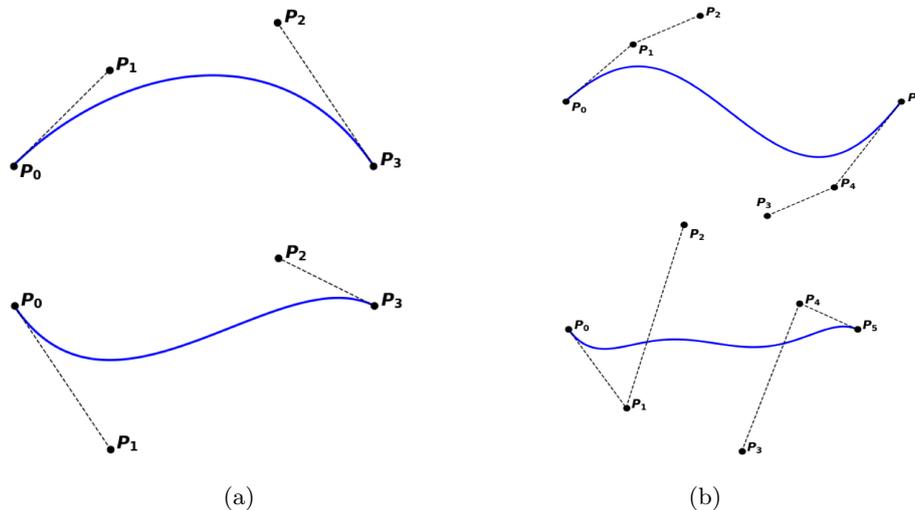


Figure 1:(a) Cubic Bezier segments and (b) quintic Bezier segments, showing control points.

Similarly, iso-parametric Bezier finite elements of order  $n$  are described by the formulation of

any function (including spatial coordinates) as

$$F(s, t) = \sum_{i=0}^n \sum_{j=0}^n B_i(s) B_j(t) P_{ij} \quad (3)$$

Where  $s$  and  $t$  are the element's local coordinates, the points  $P_{ij}$  are the control points of the elements, and the Bezier polynomials are the same as defined above by (2). Bezier elements of order 3 and 5 are represented in Figure-1. Note that for simplicity the elements are shown in 2D here, but this can be generalised to 3D. The iso-parametric property of Bezier elements means that the spatial coordinates have the same formulation (3) as any other variable.

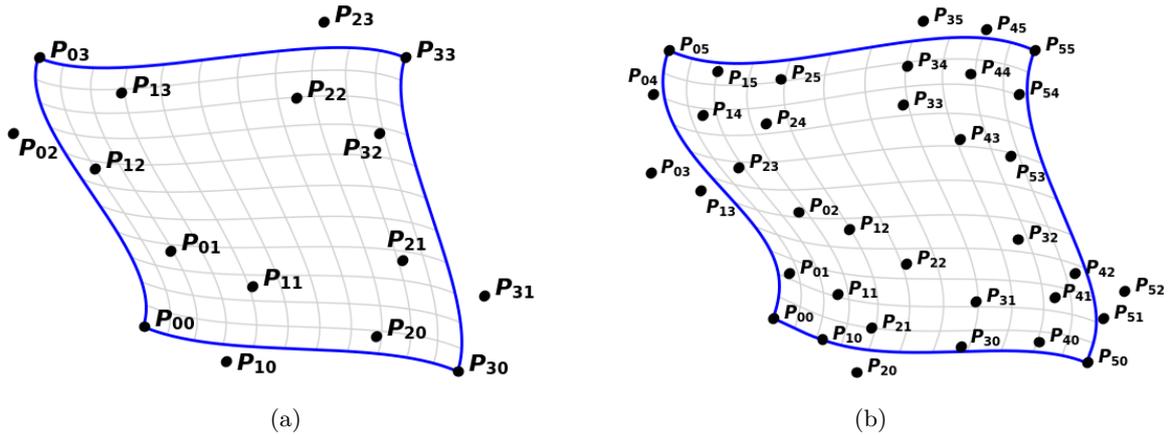


Figure 2:(a) Bi-cubic Bezier element and (b) bi-quintic Bezier element, showing control points.

## 2.2 Continuity between Bezier curves or elements

Consider two cubic Bezier curves  $\zeta^1$  and  $\zeta^2$ , with control point  $[P_0, P_1, P_2, P_3]$  and  $[Q_0, Q_1, Q_2, Q_3]$  respectively. The two curves are continuous, or C0-continuous, provided the two control points  $P_3$  and  $Q_0$  are identical. The curves are smooth, or C1-continuous, provided the vectors  $[P_4, P_3]$  and  $[Q_0, Q_1]$  are aligned, such that  $(P_3 - P_2) = \alpha(Q_1 - Q_0)$  for any non-zero positive scalar  $\alpha$ . This is represented by Figure-3, where  $\alpha$  has been chosen to be  $\alpha = 1$ . This condition is important with respect to the Bezier formalism of finite elements: the freedom of allowing  $\alpha \neq 1$  means that finite elements can have different sizes on each side of a node, implying that the derivatives along the local coordinate are not continuous, only derivatives in real space are (ie. G1 continuity).

In order to achieve C2-continuity, such that the curvature (or second derivative) of the curves is continuous at the point  $P_3 = Q_0$ , the following rule must be satisfied:

$$(P_1 - P_2) + (P_3 - P_2) = (Q_2 - Q_1) + (Q_0 - Q_1) \quad (4)$$

There is an important implication from this rule for cubic Bezier lines/elements. The C2 continuity at vertex  $P_3$  imposes the choice of  $P_1$ , such that if the position of  $P_0$  is fixed, then the first derivative value at  $P_0$ , controlled by  $P_1$  is also imposed by the C2-continuity at  $P_3$ . In other words, C2-continuity at one vertex uses the control points from another vertex. While this may be acceptable

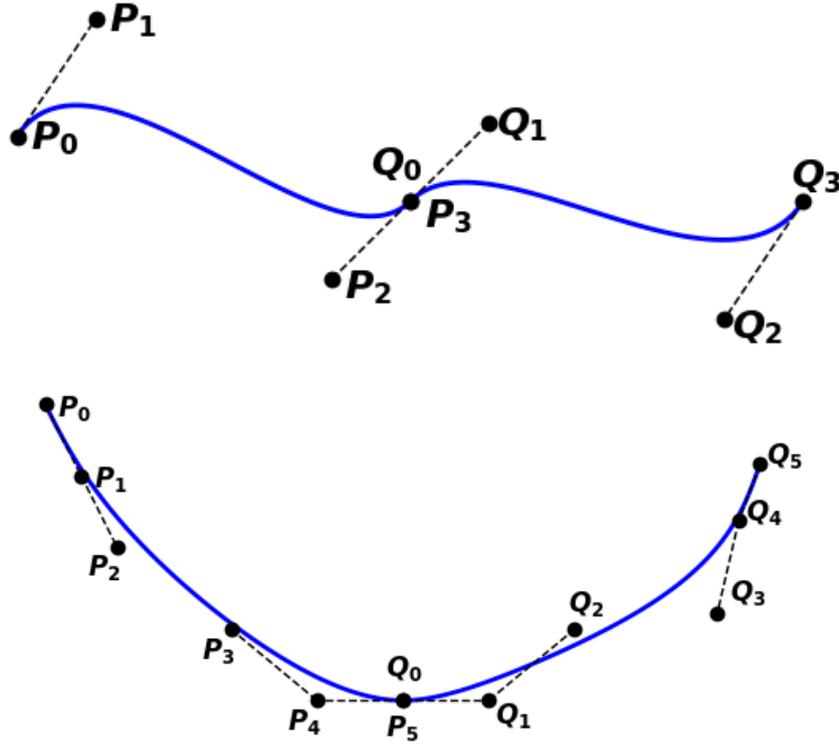


Figure 3:  $C1$ -continuity between two cubic Bezier curves (top).  $C2$ -continuity between two quintic Bezier curves (bottom).

in some simple symmetrical geometries, it greatly diminishes the flexibility of finite elements nodes to have their own 1st and 2nd derivatives, regardless of neighbouring nodes. This is the main reason to increase from cubic to quintic finite elements: since quintic curves have 6 control points, each node has its own set of 3 control points to determine 2nd order derivatives and  $C2$ -continuity with the next curve.

### 2.3 Nodal formulation of bi-quintic elements

As in [17], it is ideal to use a nodal formulation of the finite elements, and set constraints that will guarantee  $C0$ ,  $C1$  and  $C2$  continuity of all variables for any solution of the system to be solved. To achieve this, we define a vector basis used to locate all control points. In bi-cubic elements, the node value plus 3 vectors are necessary to define the 4 control points associated to each node. For bi-quintic elements, the node value plus 8 vectors are required. The nodal formulation, at node  $P_{00}$  in Figure-2b, is defined as

$$\begin{aligned}
P_{00} & \\
P_{10} &= P_{00} + h_{00}^u \vec{u}_{00} \\
P_{01} &= P_{00} + h_{00}^v \vec{v}_{00} \\
P_{11} &= P_{00} + h_{00}^u \vec{u}_{00} + h_{00}^v \vec{v}_{00} + h_{00}^w \vec{w}_{00} \\
P_{20} &= P_{00} + 2h_{00}^u \vec{u}_{00} + h_{00}^i \vec{i}_{00} \\
P_{02} &= P_{00} + 2h_{00}^v \vec{v}_{00} + h_{00}^j \vec{j}_{00} \\
P_{21} &= P_{00} + 2h_{00}^u \vec{u}_{00} + h_{00}^v \vec{v}_{00} + h_{00}^i \vec{i}_{00} + 2h_{00}^w \vec{w}_{00} + h_{00}^m \vec{m}_{00} \\
P_{12} &= P_{00} + 2h_{00}^v \vec{v}_{00} + h_{00}^u \vec{u}_{00} + h_{00}^j \vec{j}_{00} + 2h_{00}^w \vec{w}_{00} + h_{00}^n \vec{n}_{00} \\
P_{22} &= P_{00} + 2h_{00}^u \vec{u}_{00} + 2h_{00}^v \vec{v}_{00} + h_{00}^i \vec{i}_{00} + h_{00}^j \vec{j}_{00} + 2h_{00}^m \vec{m}_{00} + 2h_{00}^n \vec{n}_{00} + 4h_{00}^w \vec{w}_{00} + h_{00}^k \vec{k}_{00}
\end{aligned} \tag{5}$$

where the vectors  $\vec{u}_{00}$ ,  $\vec{v}_{00}$ ,  $\vec{w}_{00}$ ,  $\vec{i}_{00}$ ,  $\vec{j}_{00}$ ,  $\vec{m}_{00}$ ,  $\vec{n}_{00}$ ,  $\vec{k}_{00}$ , and the element sizes  $h_{00}^u$ ,  $h_{00}^v$ ,  $h_{00}^w$ ,  $h_{00}^i$ ,  $h_{00}^j$ ,  $h_{00}^m$ ,  $h_{00}^n$ ,  $h_{00}^k$  have been introduced. A representation of this formulation is shown in Figure-4.

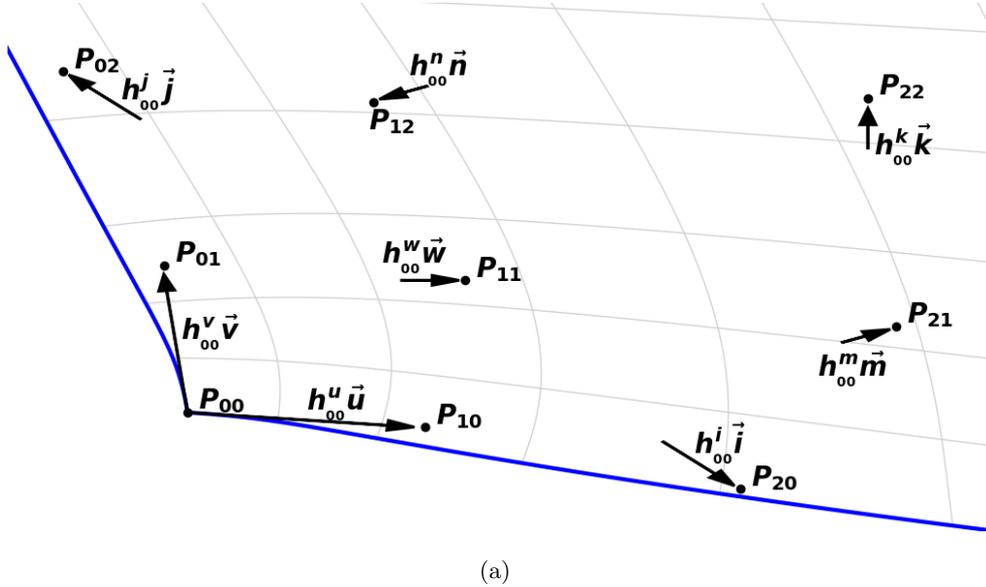


Figure 4: Nodal formulation of bi-quintic elements, focused on the first node  $P_{00}$  of an element similar to that represented in Figure-2b.

Similar definitions are used for the nodal formulation at nodes  $P_{05}$ ,  $P_{50}$  and  $P_{55}$ , to locate all control points of an element. It is worth noting that, as will become evident later, the choice of this particular formulation is not random. For example, point  $P_{22}$  could simply have been defined as  $P_{00} + h_{00}^k \vec{k}$ . However, using this mixture of vectors to represent various points will be essential in order to define C2-continuity rules. In addition, as will be shown further on, this definition has the advantage that each vector corresponds to a derivative with respect to the local coordinates  $(s, t)$  of the element.

Using the nodal formulation (5) at the four nodes of an element, all control points (except  $P_{00}$ ,  $P_{50}$ ,  $P_{05}$  and  $P_{55}$ ) can now be substituted into the Bezier definition of a finite element (3), to write

$F(s, t)$  as

$$\begin{aligned}
F(s, t) = \sum_{xx} \left[ \begin{aligned}
& F_{xx}^P P_{xx} + F_{xx}^u h_{xx}^u \vec{u}_{xx} + F_{xx}^v h_{xx}^v \vec{v}_{xx} \\
& + F_{xx}^w h_{xx}^w \vec{w}_{xx} + F_{xx}^i h_{xx}^i \vec{i}_{xx} + F_{xx}^j h_{xx}^j \vec{j}_{xx} \\
& + F_{xx}^m h_{xx}^m \vec{m}_{xx} + F_{xx}^n h_{xx}^n \vec{n}_{xx} + F_{xx}^k h_{xx}^k \vec{k}_{xx}
\end{aligned} \right] \tag{6}
\end{aligned}$$

with  $xx = [00, 05, 50, 55]$ .

where a new set of basis functions  $F_{xx}^P, F_{xx}^u, \dots, F_{xx}^k$ , has been obtained by factorizing  $F(s, t)$  with respect to each vector. Thus, the vectors  $\vec{u}, \vec{v}$ , etc. represent the degrees of freedom that are used and solved in the system of linearised equations, while the element sizes  $h^u, h^v$ , etc. are fixed in time, and chosen to ensure C2-continuity between elements, for any values of the vectors  $\vec{u}, \vec{v}$ , etc.

Note that, in this case, the Bezier function  $F(s, t)$  is actually a 2D vector, but this can simply be considered to be two separate functions  $X(s, t)$  and  $Y(s, t)$ , with each set of vectors being actual scalar degrees of freedom for each function. This is true for any variable  $\psi$  of the system to be evaluated. It is written here as vectors just to simplify the visualisation with the use of plots of 2D elements like Figure-4.

## 2.4 C0-, C1- and C2-Continuity Constraints

Finally, constraints must be set on the elements sizes in order to ensure C2-continuity for any set of node vectors. Rather than simply defining those constraints and showing that C0-, C1- and C2-continuity is thus satisfied, it is better to start from the continuity requirements, and deduce the constraints from those. Despite being more lengthy, it will provide a clearer understanding why each set of constraints is chosen. To describe these constraints, let the 4 Bezier elements  $\xi_{11}, \xi_{-11}, \xi_{1-1}$ , and  $\xi_{-1-1}$  have the common node  $P_{00}$ . The control points of the 4 parent elements around node  $P_{00}$  can be defined, similarly to (5), using the subscripts “11”, “1-1”, “-1-1” and “-11” for each of the 4 parent elements. Note that, by definition, all elements use the same vectors for the nodal formulation, but have different element sizes. Hence, on element  $\xi_{11}$ , the nodal formulation is

$$\begin{aligned}
P_{10} &= P_{00} + h_{11}^u \vec{u} \\
P_{01} &= P_{00} + h_{11}^v \vec{v} \\
P_{11} &= P_{00} + h_{11}^u \vec{u} + h_{11}^v \vec{v} + h_{11}^w \vec{w} \\
P_{20} &= P_{00} + 2h_{11}^u \vec{u} + h_{11}^i \vec{i} \\
P_{02} &= P_{00} + 2h_{11}^v \vec{v} + h_{11}^j \vec{j} \\
P_{21} &= P_{00} + 2h_{11}^u \vec{u} + h_{11}^v \vec{v} + h_{11}^i \vec{i} + 2h_{11}^w \vec{w} + h_{11}^m \vec{m} \\
P_{12} &= P_{00} + 2h_{11}^v \vec{v} + h_{11}^u \vec{u} + h_{11}^j \vec{j} + 2h_{11}^w \vec{w} + h_{11}^n \vec{n} \\
P_{22} &= P_{00} + 2h_{11}^u \vec{u} + 2h_{11}^v \vec{v} + h_{11}^i \vec{i} + h_{11}^j \vec{j} \\
&\quad + 2h_{11}^m \vec{m} + 2h_{11}^n \vec{n} + 4h_{11}^w \vec{w} + h_{11}^k \vec{k}
\end{aligned} \tag{7}$$

while on the other three elements, minus subscripts will be used for the corresponding points and element sizes. For example, point  $P_{-21}$  (on element  $\xi_{-11}$ ), point  $P_{2-1}$  (on element  $\xi_{1-1}$ ), and point  $P_{-2-1}$  (on element  $\xi_{-1-1}$ ), are defined as

$$\begin{aligned}
P_{-21} &= P_{00} + 2h_{-11}^u \vec{u} + h_{-11}^v \vec{v} + h_{-11}^i \vec{i} + 2h_{-11}^w \vec{w} + h_{-11}^m \vec{m} \\
P_{2-1} &= P_{00} + 2h_{1-1}^u \vec{u} + h_{1-1}^v \vec{v} + h_{1-1}^i \vec{i} + 2h_{1-1}^w \vec{w} + h_{1-1}^m \vec{m} \\
P_{-2-1} &= P_{00} + 2h_{-1-1}^u \vec{u} + h_{-1-1}^v \vec{v} + h_{-1-1}^i \vec{i} + 2h_{-1-1}^w \vec{w} + h_{-1-1}^m \vec{m}
\end{aligned} \tag{8}$$

## 2.5 C0-continuity

C0 continuity requires that, apart from the common node  $P_{00}$ , all control points along the element sides must coincide on both sides of an element, such that vector sizes used to define these common points must be equal on either sides of an element boundary, namely

$$\begin{aligned}
h_{1-1}^u &= h_{11}^u & \text{and} & & h_{-1-1}^u &= h_{-11}^u \\
h_{-11}^v &= h_{11}^v & \text{and} & & h_{-1-1}^v &= h_{1-1}^v \\
h_{1-1}^i &= h_{11}^i & \text{and} & & h_{-1-1}^i &= h_{-11}^i \\
h_{-11}^j &= h_{11}^j & \text{and} & & h_{-1-1}^j &= h_{1-1}^j
\end{aligned} \tag{9}$$

## 2.6 C1-continuity

Firstly, considering C1-continuity at the node itself,  $P_{00}$ , the requirement, as mentioned above, is that opposite vectors are aligned, but this is already achieved by the nodal formulation (5), since we use the same vectors  $\vec{u}$  and  $\vec{v}$  for all elements. Therefore, the first rules are

$$\begin{aligned}
h_{-11}^u &= -\alpha h_{11}^u & \text{with} & & \alpha > 0 \\
h_{1-1}^v &= -\beta h_{11}^v & \text{with} & & \beta > 0
\end{aligned} \tag{10}$$

Secondly, considering C1-continuity at the points directly adjacent to the node  $P_{00}$ , namely the control points  $P_{01}$ ,  $P_{10}$ ,  $P_{0-1}$  and  $P_{-10}$ , this requires the alignment of the points

$$\begin{aligned}
(P_{11} - P_{01}) &= -\delta_1(P_{-11} - P_{01}) & \text{with } \delta_1 > 0 \\
(P_{-1-1} - P_{0-1}) &= -\delta_2(P_{-1-1} - P_{0-1}) & \text{with } \delta_2 > 0 \\
(P_{11} - P_{10}) &= -\delta_3(P_{-1-1} - P_{10}) & \text{with } \delta_3 > 0 \\
(P_{-11} - P_{-10}) &= -\delta_4(P_{-1-1} - P_{-10}) & \text{with } \delta_4 > 0
\end{aligned} \tag{11}$$

Substituting the formulations of control points from (7,8), and using the identities (9) and (10), gives

$$\begin{aligned}
(1 - \alpha\delta_1)h_{11}^u\vec{u} + (h_{11}^w + \delta_1h_{-11}^w)\vec{w} &= 0 \\
(1 - \alpha\delta_2)h_{11}^u\vec{u} - (h_{-1-1}^w + \delta_2h_{-1-1}^w)\vec{w} &= 0 \\
(1 - \beta\delta_3)h_{11}^v\vec{v} + (h_{11}^w + \delta_3h_{-1-1}^w)\vec{w} &= 0 \\
(1 - \beta\delta_4)h_{11}^v\vec{v} + (h_{-11}^w + \delta_4h_{-1-1}^w)\vec{w} &= 0
\end{aligned}$$

Hence, introducing the following constraints on the  $w$  element sizes:

$$\begin{aligned}
h_{xx}^w &= h_{xx}^u h_{xx}^v \\
\text{with } xx &= [11, 1-1, -11, -1-1],
\end{aligned} \tag{12}$$

and using again (9) and (10), the above simplifies to give

$$\begin{aligned}
(1 - \alpha\delta_1)h_{11}^u\vec{u} + h_{11}^u h_{11}^v(1 - \alpha\delta_1)\vec{w} &= 0 \\
(1 - \alpha\delta_2)h_{11}^u\vec{u} - \beta h_{11}^u h_{11}^v(1 - \alpha\delta_2)\vec{w} &= 0 \\
(1 - \beta\delta_3)h_{11}^v\vec{v} + h_{11}^u h_{11}^v(1 - \beta\delta_3)\vec{w} &= 0 \\
(1 - \beta\delta_4)h_{11}^v\vec{v} - \alpha h_{11}^u h_{11}^v(1 - \beta\delta_4)\vec{w} &= 0
\end{aligned}$$

which is trivially satisfied for  $\delta_1 = \delta_2 = \alpha^{-1}$  and  $\delta_3 = \delta_4 = \beta^{-1}$ . Finally, C1-continuity is ensured at the points  $P_{20}$ ,  $P_{02}$ ,  $P_{-20}$  and  $P_{0-2}$  by aligning the points

$$\begin{aligned}
(P_{12} - P_{02}) &= -\lambda_1(P_{-12} - P_{02}) & \text{with } \lambda_1 > 0 \\
(P_{-1-2} - P_{0-2}) &= -\lambda_2(P_{-1-2} - P_{0-2}) & \text{with } \lambda_2 > 0 \\
(P_{21} - P_{20}) &= -\lambda_3(P_{2-1} - P_{20}) & \text{with } \lambda_3 > 0 \\
(P_{-21} - P_{-20}) &= -\lambda_4(P_{-2-1} - P_{-20}) & \text{with } \lambda_4 > 0
\end{aligned} \tag{13}$$

The constraints to be imposed here are

$$\begin{aligned}
h_{-11}^n &= -\alpha h_{11}^n & \text{and } h_{-1-1}^n &= -\alpha h_{1-1}^n \\
h_{-1-1}^m &= -\beta h_{11}^m & \text{and } h_{-1-1}^m &= -\beta h_{-11}^m
\end{aligned} \tag{14}$$

Indeed, as before, substituting points definitions from (7,8), and using the constraints (9), (10), (12) and (14), gives

$$\begin{aligned}
h_{11}^u(1 - \alpha\lambda_1)\vec{u} + 2h_{11}^u h_{11}^v(1 - \alpha\lambda_1)\vec{w} + h_{11}^n(1 - \alpha\lambda_1)\vec{n} &= 0 \\
h_{11}^u(1 - \alpha\lambda_2)\vec{u} + -2\beta h_{11}^u h_{11}^v(1 - \alpha\lambda_2)\vec{w} + h_{1-1}^n(1 - \alpha\lambda_2)\vec{n} &= 0 \\
h_{11}^v(1 - \beta\lambda_3)\vec{v} + 2h_{11}^u h_{11}^v(1 - \beta\lambda_3)\vec{w} + h_{11}^m(1 - \beta\lambda_3)\vec{m} &= 0 \\
h_{11}^v(1 - \beta\lambda_4)\vec{v} + -2\alpha h_{11}^u h_{11}^v(1 - \beta\lambda_4)\vec{w} + h_{-11}^m(1 - \beta\lambda_4)\vec{m} &= 0
\end{aligned}$$

which holds for  $\lambda_1 = \lambda_2 = \alpha^{-1}$  and  $\lambda_3 = \lambda_4 = \beta^{-1}$ .

## 2.7 C2-continuity

As for C1-continuity, the C2-continuity constraints is first considered at the node itself, where the control points must satisfy the conditions

$$\begin{aligned}
P_{00} + P_{20} - 2P_{10} &= P_{00} + P_{-20} - 2P_{-10} \\
P_{00} + P_{02} - 2P_{01} &= P_{00} + P_{0-2} - 2P_{0-1}
\end{aligned} \tag{15}$$

which can be expanded, using (7,8) and the C0- and C1-constraints derived above, to give

$$\begin{aligned}
h_{11}^i \vec{i} &= h_{-1-1}^i \vec{i} \\
h_{11}^j \vec{j} &= h_{-1-1}^j \vec{j}
\end{aligned} \tag{16}$$

implying that the C0-constraint (9) must be extended to

$$\begin{aligned}
h_{1-1}^i &= h_{-11}^i = h_{-1-1}^i = h_{11}^i \\
h_{1-1}^j &= h_{-11}^j = h_{-1-1}^j = h_{11}^j
\end{aligned} \tag{17}$$

Now, consider the C2-continuity conditions at the four control points adjacent to  $P_{00}$ , namely

$$\begin{aligned}
P_{01} + P_{21} - 2P_{11} &= P_{01} + P_{-21} - 2P_{-11} \\
P_{0-1} + P_{2-1} - 2P_{1-1} &= P_{0-1} + P_{-2-1} - 2P_{-1-1} \\
P_{10} + P_{12} - 2P_{11} &= P_{10} + P_{1-2} - 2P_{1-1} \\
P_{-10} + P_{-12} - 2P_{-11} &= P_{-10} + P_{-1-2} - 2P_{-1-1}
\end{aligned} \tag{18}$$

which, once expanded, gives only the remaining terms

$$\begin{aligned}
h_{11}^m \vec{m} &= h_{-11}^m \vec{m} \\
h_{1-1}^m \vec{m} &= h_{-1-1}^m \vec{m} \\
h_{11}^n \vec{n} &= h_{-1-1}^n \vec{n} \\
h_{-11}^n \vec{n} &= h_{-1-1}^n \vec{n}
\end{aligned} \tag{19}$$

Hence, the set of constraints (14) must be extended to

$$\begin{aligned}
h_{-11}^n &= h_{-1-1}^n = -\alpha h_{11}^n = -\alpha h_{1-1}^n \\
h_{1-1}^m &= h_{-1-1}^m = -\beta h_{11}^m = -\beta h_{-11}^m
\end{aligned} \tag{20}$$

At last, the C2-continuity at the remaining nodes is expressed as

$$\begin{aligned}
P_{02} + P_{22} - 2P_{12} &= P_{02} + P_{-22} - 2P_{-12} \\
P_{0-2} + P_{2-2} - 2P_{1-2} &= P_{0-2} + P_{-2-2} - 2P_{-1-2} \\
P_{20} + P_{22} - 2P_{21} &= P_{20} + P_{2-2} - 2P_{2-1} \\
P_{-20} + P_{-22} - 2P_{-21} &= P_{-20} + P_{-2-2} - 2P_{-2-1}
\end{aligned} \tag{21}$$

which, once expanded, gives the remaining terms

$$\begin{aligned}
h_{11}^k \vec{k} &= h_{-11}^k \vec{k} \\
h_{1-1}^k \vec{k} &= h_{-1-1}^k \vec{k} \\
h_{11}^k \vec{k} &= h_{1-1}^k \vec{k} \\
h_{-11}^k \vec{k} &= h_{-1-1}^k \vec{k}
\end{aligned} \tag{22}$$

Now, this leads to the final constraint on the element size for the  $\vec{k}$  vectors:

$$\begin{aligned}
h_{xx}^k &= h_{xx}^i h_{xx}^j \\
\text{with } xx &= [11, 1-1, -11, -1-1],
\end{aligned} \tag{23}$$

Using this rule, together with (17), the above is trivially satisfied.

For convenience, the C2-continuity constraints are summarised here:

$$h_{-1-1}^u = h_{-11}^u = -\alpha h_{1-1}^u = -\alpha h_{11}^u \tag{24}$$

$$h_{-1-1}^v = h_{1-1}^v = -\beta h_{-11}^v = -\beta h_{11}^v \tag{25}$$

$$h_{1-1}^i = h_{11}^i = h_{-1-1}^i = h_{-11}^i \tag{26}$$

$$h_{-11}^j = h_{11}^j = h_{-1-1}^j = h_{1-1}^j \tag{27}$$

$$h_{1-1}^m = h_{-1-1}^m = -\beta h_{11}^m = -\beta h_{-11}^m \tag{28}$$

$$h_{-11}^n = h_{-1-1}^n = -\alpha h_{11}^n = -\alpha h_{1-1}^n \tag{29}$$

$$h_{xx}^w = h_{xx}^u h_{xx}^v \tag{30}$$

$$h_{xx}^k = h_{xx}^i h_{xx}^j \tag{31}$$

$$\text{with } xx = [11, 1-1, -11, -1-1],$$

### 3 Generalised formulation: beyond C2 continuity

The nodal formulation (5), and the element size rules (24-31) can be generalised to provide Bezier elements of arbitrary odd order  $n$ , with C-continuity of order  $(n-1)/2$ . This generalisation requires the definition of a nodal formulation at any odd order  $n$ . The order  $n$  must be odd such that each node have the same number of associated control points. Note that continuity of order  $(n-1)/2$  is obtained since in a given direction, there are  $(n+1)$  control points, including the end nodes, hence  $(n+1)/2$  control points for each node, one being the value of a variable, and then one for each derivative, such that  $(n-1)/2$  derivatives can be controlled. At each node, there are  $(n+1)^2/4$  degrees of freedom.

### 3.1 Notation for the derivatives of control points

Before defining the general formulation, it helps to understand the structure of C-continuity requirements.

**Notation-1 :**

**C1-continuity**

Let direction  $x$  denote the first index of control points  $P_{ij}$ , and direction  $y$  the second index. The first derivative, in the direction  $x$ , at a border point  $P_{0j}$  is defined by

$$\Delta_x^1 P_{0j} = P_{0j} - P_{1j}, \text{ for any } j. \quad (32)$$

On the opposite element, it is defined as

$$\Delta_x^{-1} P_{0j} = P_{0j} - P_{-1j}, \text{ for any } j. \quad (33)$$

Likewise, the first derivative in the  $y$  direction is defined by

$$\Delta_y^1 P_{i0} = P_{i0} - P_{i1}, \text{ for any } i \quad (34)$$

and, on the opposite element,

$$\Delta_y^{-1} P_{i0} = P_{i0} - P_{i-1}, \text{ for any } i. \quad (35)$$

The four parent elements of node  $P_{00}$  are C1-continuous provided

$$\Delta_x^{-1} P_{0j} = -\lambda_1 \Delta_x^1 P_{0j}, \quad (36)$$

$$\Delta_y^{-1} P_{i0} = -\lambda_2 \Delta_y^1 P_{i0}, \quad (37)$$

for any indices  $i, j$  in interval  $[-(n-1)/2, (n-1)/2]$ , and any positive real scalars  $\lambda_1, \lambda_2$ .

**Notation-2 :****C-continuity of order  $m$** 

The  $m$ -th derivative, in direction  $x$ , at point  $P_{0j}$ , is defined by

$$\Delta_x^m P_{0j} = \Delta_x^{m-1} P_{0j} - \Delta_x^{m-1} P_{1j} \quad (38)$$

Just like standard derivatives, it is trivial to demonstrate by induction that this is equivalent to

$$\Delta_x^m P_{0j} = \sum_{i=0}^m (-1)^i \binom{m}{i} P_{ij} \quad (39)$$

Likewise, on the opposite element, the  $m$ -th derivative can be written as

$$\Delta_x^{-m} P_{0j} = \sum_{i=0}^m (-1)^i \binom{m}{i} P_{i-j} \quad (40)$$

Similarly, at point  $P_{i0}$ , the  $m$ -th derivatives, in direction  $y$ , on two opposite elements, are defined as

$$\Delta_y^m P_{i0} = \sum_{j=0}^m (-1)^j \binom{m}{j} P_{ij} \quad (41)$$

$$\Delta_y^{-m} P_{i0} = \sum_{j=0}^m (-1)^j \binom{m}{j} P_{i-j} \quad (42)$$

Unlike the C1-continuity requirement (36-37), C-continuity of order  $m$  requires equality of the  $m$ -th derivatives on either sides of a node, namely

$$\Delta_x^m P_{0j} = \Delta_x^{-m} P_{0j}, \quad (43)$$

$$\Delta_y^m P_{i0} = \Delta_y^{-m} P_{i0}, \quad (44)$$

It is important to note that we use a sub-case version of C-continuity definition. Generic C1- and C2-continuity definitions can be found in [29, 30] respectively. In this paper, a special case of these conditions is used. In the most general case, C2-continuity involves both the second, first and zero derivatives of control points, but in this study, we neglect lower-order derivatives, which is equivalent to satisfying C-continuity by aligning the control points accordingly. The main advantage of the more generic continuity definition is that they allow the degrees of freedom to be shifted from the control points to the polynomial representation of boundaries between elements. One of the main outcomes of such a generic method is that it allows control points to be “misaligned”, enabling strong angles between elements at a given node. Such a generic method is, however, beyond the scope of this paper and could be considered in future studies.

The notation of control point derivatives (39,41), together with the continuity rules (43,44) can be derived simply by evaluating the derivative of the Bezier formula (3) at  $s=0$  or  $s=1$  (or  $t=0$  and  $t=1$ ). This derivation is not detailed here, but its method can be seen in [17] for C1-continuity, and is identical for higher-order derivatives.

Now, consider the generic Bezier 1D-derivative identity (39) as a function of the control points. The ideal nodal formulation to satisfy these conditions is one that automatically simplifies the derivative at any order. This should be, of course, in both directions  $i$  and  $j$ . As will become clear below, the generic formulation is nothing less than a construction similar to mixed 2D-derivatives.

### 3.2 Generalised nodal formulation of Bezier elements

#### Definition-1 :

For Bezier elements of any odd polynomial order  $n$ , the nodal formulation of control points  $P_{ij}$  is defined as

$$P_{ij} = h^{ij} \vec{u}^{ij} + \sum_{k=0}^i \sum_{l=0}^j (-1)^{1+i+j+k+l} (1 - \delta_{ki} \delta_{lj}) \binom{i}{k} \binom{j}{l} P_{kl} \quad (45)$$

where

$$0 \leq i < (n+1)/2$$

$$0 \leq j < (n+1)/2$$

$\delta_{ab}$  is the Kronecker delta

$h^{ij}$  are the sizes of each component of the nodal vectorial basis.

$$P_{00} = h^{00} \vec{u}^{00} := P_{00}$$

#### Definition-2 :

The notation of the element sizes for the 4 parent elements of a node are expressed as  $h^{ij}$ ,  $h^{-ij}$ ,  $h^{i-j}$  and  $h^{-i-j}$ , and they are constrained with the following rules

for any  $j$ ,  $h_{-ij}$  is constrained by

$$h^{-ij} = \begin{cases} -\alpha h^{ij} & \text{for } i = 1 \text{ and } \alpha > 0 \\ h^{ij} & \text{for } i \neq 1 \end{cases} \quad (46)$$

for any  $i$ ,  $h_{i-j}$  is constrained by

$$h^{i-j} = \begin{cases} -\beta h^{ij} & \text{for } j = 1 \text{ and } \beta > 0 \\ h^{ij} & \text{for } j \neq 1 \end{cases} \quad (47)$$

Note that Definition-1 is clearly just a mixture of the derivatives (39) in two dimensions. The Kronecker-delta functions are simply required to remove the re-occurrence of the point  $P_{ij}$ . Using these two definitions, it can be demonstrated that C-continuity is ensured at order  $(n-1)/2$ . Before this can be achieved, some necessary properties of the definitions must first be obtained.

### 3.3 C1- and C2-continuity demonstration

The final proof of C-continuity is done by induction, and therefore C1- and C2-continuity must first be proven.

**Proposition-1 :**

The nodal formulation  $P_{ij}$  defined in Definition-1, together with the element size constraints defined in Definition-2, ensures C1-continuity between Bezier elements.

**Proof :**

Consider C1-continuity in the  $x$  direction. C1-continuity at the node  $P_{00}$  itself (ie. at  $j=0$ ) is trivial, since expanding the nodal formulation (45) at points  $P_{10}$  and  $P_{-10}$  gives

$$P_{10} = P_{00} + h^{10}\vec{u}^{10} \quad (48)$$

$$P_{-10} = P_{00} + h^{-10}\vec{u}^{10} \quad (49)$$

where of course, by definition, the nodal formulation implies that  $\vec{u}^{-10} = \vec{u}^{10}$ . The above can be rearranged and written as

$$\Delta_x^1 P_{00} = -h^{10}\vec{u}^{10} \quad (50)$$

$$\Delta_x^{-1} P_{00} = -h^{-10}\vec{u}^{10} = \alpha h^{10}\vec{u}^{10} = -\alpha \Delta_x^1 P_{00} \quad (51)$$

where rule (46) of Definition-2 was used. Thus, the C1-continuity rule (36) is satisfied with  $\lambda_1 = \alpha$ . Similarly, at  $j=1$ , points  $P_{11}$  and  $P_{-11}$  can be expressed, using (45), as

$$P_{11} = h^{11}\vec{u}^{11} - P_{00} + P_{01} + P_{10} = h^{11}\vec{u}^{11} - \Delta_x^1 P_{00} + P_{01} \quad (52)$$

$$P_{-11} = h^{-11}\vec{u}^{11} - P_{00} + P_{01} + P_{-10} = -\alpha h^{11}\vec{u}^{11} - \Delta_x^{-1} P_{00} + P_{01} \quad (53)$$

where rule (46) of Definition-2 was used. However, C1-continuity at  $P_{00}$  already provides an expression for  $\Delta_x^{-1} P_{00}$  in (51), such that the above can be rearranged as

$$\Delta_x^1 P_{01} = -h^{11}\vec{u}^{11} + \Delta_x^1 P_{00} \quad (54)$$

$$\Delta_x^{-1} P_{01} = +\alpha h^{11}\vec{u}^{11} - \alpha \Delta_x^1 P_{00} \quad (55)$$

Thus, the C1-continuity rule (36) is satisfied at  $P_{01}$  with  $\lambda_1 = \alpha$ .

We now proceed by induction. Assume C1-continuity holds at point  $P_{0,(j-1)}$ , such that  $\Delta_x^{-1} P_{0l} =$

$-\alpha\Delta_x^1 P_{0l}$  for all  $l \leq j-1$ . Then, expressing  $P_{1j}$  using the nodal formulation (45) gives

$$\begin{aligned}
P_{1j} &= h^{1j}\bar{u}^{1j} + \sum_{k=0}^1 \sum_{l=0}^j (-1)^{j+k+l} (1 - \delta_{k1}\delta_{lj}) \binom{j}{l} P_{kl} \\
&= h^{1j}\bar{u}^{1j} + \sum_{l=0}^j (-1)^{j+l} \binom{j}{l} P_{0l} + \sum_{l=0}^j (-1)^{1+j+l} (1 - \delta_{lj}) \binom{j}{l} P_{1l} \\
&= h^{1j}\bar{u}^{1j} + P_{0j} + \sum_{l=0}^{j-1} (-1)^{j+l} \binom{j}{l} P_{0l} + \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} P_{1l} \\
&= h^{1j}\bar{u}^{1j} + P_{0j} + \sum_{l=0}^{j-1} (-1)^{j+l} \binom{j}{l} (P_{0l} - P_{1l}) \\
&= h^{1j}\bar{u}^{1j} + P_{0j} + \sum_{l=0}^{j-1} (-1)^{j+l} \binom{j}{l} \Delta_x^1 P_{0l}
\end{aligned} \tag{56}$$

Similarly, point  $P_{-1j}$  is expanded to give

$$\begin{aligned}
P_{-1j} &= h^{-1j}\bar{u}^{1j} + P_{0j} + \sum_{l=0}^{j-1} (-1)^{j+l} \binom{j}{l} (P_{0l} - P_{-1l}) \\
&= -\alpha h^{1j}\bar{u}^{1j} + P_{0j} + \sum_{l=0}^{j-1} (-1)^{j+l} \Delta_x^{-1} P_{0l}
\end{aligned} \tag{57}$$

where rule (46) of Definition-2 was used. However, since the sum term in (57) is only up to  $j-1$ , C1-continuity applies for  $\Delta_x^{-1} P_{0l}$  by assumption, which can be replaced by  $-\alpha\Delta_x^1 P_{0l}$ . Therefore, (56) and (57) can be rearranged to write

$$\Delta_x^1 P_{0j} = -h^{1j}\bar{u}^{1j} - \sum_{l=0}^{j-1} (-1)^{j+l} \binom{j}{l} \Delta_x^1 P_{0l} \tag{58}$$

$$\Delta_x^{-1} P_{0j} = +\alpha h^{1j}\bar{u}^{1j} + \alpha \sum_{l=0}^{j-1} (-1)^{j+l} \binom{j}{l} \Delta_x^1 P_{0l} \tag{59}$$

In other words, the C1-continuity rule (36) is satisfied at  $P_{0j}$  with  $\lambda_1 = \alpha$ . Hence, by induction, since we know that C1-continuity is valid at point  $P_{01}$ , it follows that C1-continuity is valid at all points  $P_{0j}$  for  $j \leq (n-1)/2$ . The demonstration for negative  $j$  is almost identical. Likewise, the above demonstration for C1-continuity in the  $y$  direction, at any point  $P_{i0}$ , is also identical, by swapping indices. This completes the proof, that the nodal formulation of Definition-1 and Definition-2 satisfies C1-continuity between Bezier elements.

□

### **Proposition-2 :**

The nodal formulation  $P_{ij}$  defined in Definition-1, together with the element size constraints defined in Definition-2, ensures C2-continuity between Bezier elements.

### **Proof :**

As with C1-continuity proof, consider C2-continuity in the  $x$  direction. Expand point  $P_{20}$  using the

nodal formulation (45),

$$\begin{aligned} P_{20} &= h^{20} \bar{u}^{20} + \sum_{k=0}^2 (-1)^{1+k} (1 - \delta_{k2}) \binom{2}{k} P_{k0} \\ &= h^{20} \bar{u}^{20} - P_{00} + 2P_{10} \end{aligned} \quad (60)$$

However, the second derivative is defined as  $\Delta_x^2 P_{00} = P_{00} - 2P_{10} + P_{20}$ , so that the above can be rearranged as

$$\Delta_x^2 P_{00} = h^{20} \bar{u}^{20}$$

Likewise, on the opposite side,

$$\begin{aligned} \Delta_x^{-2} P_{00} &= h^{-20} \bar{u}^{20} \\ &= h^{20} \bar{u}^{20} \end{aligned}$$

where rule (46) of Definition-2 was used. Thus,  $\Delta_x^2 P_{00} = \Delta_x^{-2} P_{00}$ , so that C2-continuity is satisfied at  $P_{00}$ .

Next, we proceed by induction. Assume C2-continuity holds at point  $P_{0,(j-1)}$ , such that  $\Delta_x^2 P_{0l} = \Delta_x^{-2} P_{0l}$  for all  $l \leq j-1$ . Then, expressing  $P_{2j}$  using the nodal formulation,

$$\begin{aligned} P_{2j} &= h^{2j} \bar{u}^{2j} + \sum_{k=0}^2 \sum_{l=0}^j (-1)^{1+j+k+l} (1 - \delta_{k2} \delta_{lj}) \binom{2}{k} \binom{j}{l} P_{kl} \\ &= h^{2j} \bar{u}^{2j} + \sum_{l=0}^j (-1)^{1+j+l} \binom{j}{l} P_{0l} + 2 \sum_{l=0}^j (-1)^{j+l} \binom{j}{l} P_{1l} + \sum_{l=0}^j (-1)^{1+j+l} (1 - \delta_{lj}) \binom{j}{l} P_{2l} \\ &= h^{2j} \bar{u}^{2j} - P_{0j} + 2P_{1j} \\ &\quad + \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} P_{0l} + 2 \sum_{l=0}^{j-1} (-1)^{j+l} \binom{j}{l} P_{1l} + \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} P_{2l} \end{aligned} \quad (61)$$

which can simply be rearranged as

$$\begin{aligned} \Delta_x^2 P_{0j} &= h^{2j} \bar{u}^{2j} + \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} (P_{0l} - 2P_{1l} + P_{2l}) \\ &= h^{2j} \bar{u}^{2j} + \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} \Delta_x^2 P_{0l} \end{aligned} \quad (62)$$

Likewise, on the opposite side,

$$\begin{aligned} \Delta_x^{-2} P_{0j} &= h^{-2j} \bar{u}^{2j} + \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} (P_{0l} - 2P_{-1l} + P_{-2l}) \\ &= h^{2j} \bar{u}^{2j} + \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} \Delta_x^{-2} P_{0l} \end{aligned} \quad (63)$$

where rule (46) of Definition-2 was used for  $h^{-2j} = h^{2j}$ . However, since C2-continuity holds for  $l \leq j-1$ , the sum term in (63) is equivalent to the sum term in (62), such that  $\Delta_x^2 P_{0j} = \Delta_x^{-2} P_{0j}$ . Hence, by induction, C2-continuity in the  $x$  direction is satisfied for all  $j$  between 0 and  $(n-1)/2$ . For negative  $j$ , the demonstration is identical. Likewise, in the  $y$  direction, the demonstration is identical, by switching indices.

□

### 3.4 C-continuity demonstration

**Proposition-3 :**

If the Bezier elements are C-continuous at order  $m$ , in direction  $x$ , then for any index  $i$  such that  $1 < i \leq m$ , the following identity holds,

$$P_{ij} - P_{-ij} = -i(\alpha + 1)(P_{0j} - P_{1j}) \quad (64)$$

for any index  $j$ , and where  $\alpha$  is the positive scalar of the C1-continuity rule in the  $x$  direction from (46).

Proof :

First, consider C1-continuity at point  $P_{0j}$ , which implies that

$$\begin{aligned} \Delta_x^{-1} P_{0j} &= -\alpha \Delta_x^1 P_{0j} \\ \implies P_{0j} - P_{-1j} &= -\alpha(P_{0j} - P_{1j}) \end{aligned} \quad (65)$$

Then, consider C2-continuity at point  $P_{0j}$ , so that

$$\begin{aligned} \Delta_x^2 P_{0j} &= \Delta_x^{-2} P_{0j} \\ \implies P_{0j} - 2P_{1j} + P_{2j} &= P_{0j} - 2P_{-1j} + P_{-2j} \\ \implies P_{2j} - P_{-2j} &= 2P_{1j} - 2P_{-1j} \\ \implies P_{2j} - P_{-2j} &= 2(P_{1j} - P_{0j}) - 2(P_{-1j} - P_{0j}) \end{aligned} \quad (66)$$

Substituting for (65) into (66) gives

$$\begin{aligned} P_{2j} - P_{-2j} &= 2(P_{1j} - P_{0j}) + 2\alpha(P_{-1j} - P_{0j}) \\ \implies P_{2j} - P_{-2j} &= -2(\alpha + 1)(P_{0j} - P_{1j}) \end{aligned} \quad (67)$$

Next, we proceed by induction. Assume that identity (64) holds for all values of  $i - 1$  up to some  $i - 1 < m$ . Then, since C-continuity holds at order  $i$ , the following identity can be written,

$$\begin{aligned} \Delta_x^i P_{0j} &= \Delta_x^{-i} P_{0j} \\ \implies \sum_{k=0}^i (-1)^k \binom{i}{k} P_{kj} &= \sum_{k=0}^i (-1)^k \binom{i}{k} P_{-kj} \\ \implies (-1)^i P_{ij} + \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} P_{kj} &= (-1)^i P_{-ij} + \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} P_{-kj} \\ \implies P_{ij} - P_{-ij} &= -(-1)^i \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} (P_{kj} - P_{-kj}) \end{aligned} \quad (68)$$

However, since the right-hand side sum term only goes up to  $i - 1$ , by assumption, we can apply identity (64), to write (68) as

$$P_{ij} - P_{-ij} = +(-1)^i \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} k(\alpha + 1)(P_{0j} - P_{1j}) \quad (69)$$

Now, most of the terms in the right-hand side sum term are independent of the summation index  $k$ , so they can simply be extracted from the sum, such that (69) becomes

$$P_{ij} - P_{-ij} = + (-1)^i (\alpha + 1) (P_{0j} - P_{1j}) \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} k \quad (70)$$

The sum term is now close to a known binomial identity [31], which states that

$$\begin{aligned} \sum_{k=0}^i (-1)^k \binom{i}{k} k &= 0 \\ \implies \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} k &= -(-1)^i i \end{aligned} \quad (71)$$

Hence, (70) becomes

$$\begin{aligned} P_{ij} - P_{-ij} &= - (-1)^{2i} i (\alpha + 1) (P_{0j} - P_{1j}) \\ &= - i (\alpha + 1) (P_{0j} - P_{1j}) \end{aligned} \quad (72)$$

By induction, since identity (64) was shown for  $i = 2$ , this completes the proof up to  $i = m$ .

□

**Theorem-1 :**

With the nodal formulation of  $P_{ij}$  defined in Definition-1, together with the element size constraints defined in Definition-2, Bezier elements are C-continuous at order  $(n - 1)/2$  for any value of the vectors  $\vec{u}^{ij}$ .

**Proof :**

This proof is a little intricate, because it contains a proof-by-induction within a proof-by-induction. Since Proposition-2 already shows that C-continuity holds at order = 2, the first induction is on the order of the C-continuity, proving that if C-continuity holds for  $i - 1$ , then it also does for  $i$ . However, this procedure is also done by induction for the second index  $j$  of the point  $P_{0j}$ , showing that if C-continuity of order  $i$  holds for  $P_{0,(j-1)}$ , then it also does for  $P_{ij}$ .

Hence, assume that C-continuity holds up to order  $i - 1$  for some  $2 \leq i - 1 < (n - 1)/2$ . Consider the C-continuity requirement, in direction  $x$ , at order  $i$ . Exactly like the above (68), this gives

$$\begin{aligned} \Delta_x^i P_{0j} &= \Delta_x^{-i} P_{0j} \\ \implies P_{ij} - P_{-ij} &= -(-1)^i \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} (P_{kj} - P_{-kj}) \end{aligned} \quad (73)$$

Again, since the right-hand side sum term only goes up to  $i - 1$ , by assumption, we can apply identity (64), to write (73) as

$$P_{ij} - P_{-ij} = + (-1)^i \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} k (\alpha + 1) (P_{0j} - P_{1j}) \quad (74)$$

which, as in the demonstration of Proposition-3, gives

$$P_{ij} - P_{-ij} = -i(\alpha + 1)(P_{0j} - P_{1j}) \quad (75)$$

Thus, in order to prove that C-continuity holds at order  $i$ , we need to show that the above identity (75) is true. First, consider  $P_{ij} - P_{-ij}$  for  $j = 0$ , by expanding using the nodal formulation (45) of Definition-1,

$$\begin{aligned} P_{i0} - P_{-i0} &= \bar{u}^{i0}(h^{i0} - h^{-i0}) + \sum_{k=0}^i (-1)^{1+i+k}(1 - \delta_{ki}) \binom{i}{k} (P_{k0} - P_{-k0}) \\ &= \sum_{k=0}^{i-1} (-1)^{1+i+k} \binom{i}{k} (P_{k0} - P_{-k0}) \end{aligned} \quad (76)$$

where rule (46) of Definition-2 was used for  $h^{-i0} = h^{i0}$ , for any  $i > 1$ . Now, since the sum term in (76) only goes up to  $i - 1$ , we can use identity (64) to write it as

$$P_{i0} - P_{-i0} = -(-1)^{1+i}(\alpha + 1)(P_{00} - P_{10}) \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} k \quad (77)$$

which, using (71), is simply

$$\begin{aligned} P_{i0} - P_{-i0} &= (-1)^{1+2i} i(\alpha + 1)(P_{00} - P_{10}) \\ &= -i(\alpha + 1)(P_{00} - P_{10}) \end{aligned} \quad (78)$$

which demonstrates C-continuity for  $j = 0$ . Next, consider  $j = 1$ , and expand  $P_{i1} - P_{-i1}$  using the nodal formulation (45) of Definition-1,

$$\begin{aligned} &P_{i1} - P_{-i1} \\ &= \bar{u}^{i1}(h^{i1} - h^{-i1}) + \sum_{k=0}^i \sum_{l=0}^1 (-1)^{i+k+l}(1 - \delta_{ki}\delta_{l1}) \binom{i}{k} (P_{kl} - P_{-kl}) \\ &= \sum_{k=0}^i (-1)^{i+k} \binom{i}{k} (P_{k0} - P_{-k0}) + \sum_{k=0}^i (-1)^{1+i+k}(1 - \delta_{ki}) \binom{i}{k} (P_{k1} - P_{-k1}) \\ &= \sum_{k=0}^i (-1)^{i+k} \binom{i}{k} (P_{k0} - P_{-k0}) + \sum_{k=0}^{i-1} (-1)^{1+i+k} \binom{i}{k} (P_{k1} - P_{-k1}) \\ &= (P_{i0} - P_{-i0}) + \sum_{k=0}^{i-1} (-1)^{i+k} \binom{i}{k} (P_{k0} - P_{-k0}) + \sum_{k=0}^{i-1} (-1)^{1+i+k} \binom{i}{k} (P_{k1} - P_{-k1}) \\ &= (P_{i0} - P_{-i0}) \end{aligned} \quad (79)$$

where the cancellation of the two sum terms is obvious since  $(-1)^1 = -1$  can simply be extracted from the sum. The cancellation of  $(h^{i1} - h^{-i1})$  comes from the definition of element sizes (46) for  $i > 1$ . Next, (78) implies that the above can be written as

$$P_{i1} - P_{-i1} = -i(\alpha + 1)(P_{00} - P_{10}) \quad (80)$$

which demonstrates C-continuity for  $j = 1$ . Now, we proceed by induction. Assume that identity (75) holds up to some  $j - 1$ . Then, expanding  $P_{ij} - P_{-ij}$  with the nodal formulation (45) of Definition-1,

$$\begin{aligned}
& P_{ij} - P_{-ij} \\
&= \cancel{\bar{u}^{ij}(h^{ij} - \bar{h}^{-ij})} + \sum_{k=0}^i \sum_{l=0}^j (-1)^{1+i+j+k+l} (1 - \delta_{ki} \delta_{lj}) \binom{i}{k} \binom{j}{l} (P_{kl} - P_{-kl}) \\
&= \sum_{l=0}^j (-1)^{1+j+l} (1 - \delta_{lj}) \binom{j}{l} (P_{il} - P_{-il}) + \sum_{k=0}^{i-1} \sum_{l=0}^j (-1)^{1+i+j+k+l} \binom{i}{k} \binom{j}{l} (P_{kl} - P_{-kl}) \\
&= \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} (P_{il} - P_{-il}) + \sum_{k=0}^{i-1} \sum_{l=0}^j (-1)^{1+i+j+k+l} \binom{i}{k} \binom{j}{l} (P_{kl} - P_{-kl}) \tag{81}
\end{aligned}$$

Now, by assumption, since the first term only goes up to  $j - 1$ , identity (75) can be used. Likewise, by assumption, since we have assumed C-continuity up to  $i - 1$ , identity (64) can be used for the second term. Thus, (81) can be written as

$$\begin{aligned}
& P_{ij} - P_{-ij} \\
&= -(\alpha + 1)i \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} (P_{0l} - P_{1l}) \\
&\quad - (\alpha + 1) \sum_{k=0}^{i-1} \sum_{l=0}^j (-1)^{1+i+j+k+l} \binom{i}{k} \binom{j}{l} i (P_{0l} - P_{1l}) \tag{82}
\end{aligned}$$

Now, the second term is separated to isolate the  $k$ -sum as much as possible, such that

$$\begin{aligned}
& P_{ij} - P_{-ij} \\
&= -(\alpha + 1)i \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} (P_{0l} - P_{1l}) \\
&\quad - (\alpha + 1)i \sum_{l=0}^j (-1)^{1+i+j+l} \binom{j}{l} (P_{0l} - P_{1l}) \left[ \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} \right] \tag{83}
\end{aligned}$$

However, this is also a well known binomial identity, such that

$$\begin{aligned}
& \sum_{k=0}^i (-1)^k \binom{i}{k} = 0 \\
\implies & \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} = -(-1)^i
\end{aligned}$$

Hence, (83) can be written as

$$\begin{aligned}
& P_{ij} - P_{-ij} \\
&= -(\alpha + 1)i \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} (P_{0l} - P_{1l}) \\
&\quad + (\alpha + 1)i \sum_{l=0}^j (-1)^{1+j+l} \binom{j}{l} (P_{0l} - P_{1l}) \\
&= \cancel{-(\alpha + 1)i \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} (P_{0l} - P_{1l})} \\
&\quad - i(\alpha + 1)(P_{0j} - P_{1j}) + \cancel{(\alpha + 1)i \sum_{l=0}^{j-1} (-1)^{1+j+l} \binom{j}{l} (P_{0l} - P_{1l})} \\
&= -i(\alpha + 1)(P_{0j} - P_{1j}) \tag{84}
\end{aligned}$$

which demonstrates, by induction, that identity (75) holds for all positive  $j \leq (n-1)/2$ . The proof for all negative  $j$  is identical.

Thus, this completes the proof that C-continuity of order  $i$  holds if it is assumed at order  $i-1$ , in the  $x$  direction. Therefore, C-continuity holds for all  $i$  up to  $(n-1)/2$ , since C-continuity was shown for  $i=2$ . The same holds for the  $y$ -direction, for which an analogue of Proposition-3 can be obtained in the  $y$ -direction, such that the above proof can be repeated, with switched indices, in the  $y$ -direction.

□

### 3.5 Derivatives identity

Note that the generalised nodal formulation of Definition-1 is very similar to the derivatives definition (39), but in two dimensions. In fact, a notable property of Definition-1 is the following corollary.

#### Corollary-1 :

For the nodal formulation of  $P_{ij}$  defined in Definition-1, any derivative of order  $(f, g)$  at the element node  $P_{00}$ , with  $f + g > 0$ , corresponds to the vector  $\vec{u}^{fg}$ , namely

$$\partial_s^f \partial_t^g F(0, 0) = \frac{n!}{(n-f)!} \frac{n!}{(n-g)!} h^{fg} \vec{u}^{fg} \tag{85}$$

Proof :

Expanding the Bezier formulation (3) gives

$$F(s, t) = \sum_{i=0}^n \sum_{j=0}^m \binom{n}{i} \binom{m}{j} s^i (1-s)^{n-i} t^j (1-t)^{m-j} P_{ij} \tag{86}$$

where the polynomial orders have been separated between  $n$  and  $m$ , because it is important to distinguish between the two for this derivation, but in reality,  $m=n$ . Now, consider the derivative of order  $f$  for the two composites of a Bernstein polynomial,

$$\begin{aligned}\partial_s^f [s^i] &= \begin{cases} \frac{i!}{(i-f)!} s^{i-f} & \text{for } 0 \leq f \leq i \\ 0 & \text{for } f > i \end{cases} \\ \partial_s^f [(1-s)^{n-i}] &= \begin{cases} \frac{(n-i)!}{(n-i-f)!} (-1)^f (1-s)^{n-i-f} & \text{for } 0 \leq f \leq n-i \\ 0 & \text{for } f > n-i \end{cases}\end{aligned}$$

Evaluated at  $s=0$ , this gives

$$\left. \partial_s^f [s^i] \right|_{s=0} = \begin{cases} i! & \text{for } f = i \\ 0 & \text{for } f \neq i \end{cases} \quad (87)$$

$$\left. \partial_s^f [(1-s)^{n-i}] \right|_{s=0} = \begin{cases} \frac{(n-i)!}{(n-i-f)!} (-1)^f & \text{for } 0 \leq f \leq n-i \\ 0 & \text{for } f > n-i \end{cases} \quad (88)$$

Next, consider the derivative of the factor of two functions of  $I(s)$  and  $J(s)$ , namely,

$$\partial_s^f [IJ] = \sum_{a=0}^f \binom{f}{a} (\partial_s^a I) (\partial_s^{f-a} J)$$

and substitute the functions with the Bernstein polynomial components, such that  $I = s^i$  and  $J = (1-s)^{n-i}$ . Then

$$\partial_s^f [s^i (1-s)^{n-i}] = \sum_{a=0}^f \binom{f}{a} (\partial_s^a [s^i]) (\partial_s^{f-a} [(1-s)^{n-i}]) \quad (89)$$

From (87), it follows that the above is zero at  $s=0$  unless  $a=i$ . In addition, since the summation goes from  $a=0$  to  $a=f$ , the above is also zero if  $i > f$ . Hence, evaluating (89) at  $s=0$  gives

$$\left. \partial_s^f [s^i (1-s)^{n-i}] \right|_{s=0} = \begin{cases} \binom{f}{i} i! (\partial_s^{f-i} [(1-s)^{n-i}]) & \text{for } i \leq f \\ 0 & \text{for } i > f \end{cases} \quad (90)$$

However, from (88), it also follows that the above is zero if  $f-i > n-i$ , such that (90) becomes

$$\left. \partial_s^f [s^i (1-s)^{n-i}] \right|_{s=0} = \begin{cases} \binom{f}{i} i! (-1)^{f-i} \frac{(n-i)!}{(n-f)!} & \text{for } i \leq f \leq n \\ 0 & \text{otherwise} \end{cases} \quad (91)$$

Now, the Bezier function (86) can be derived with respect to  $s$  and  $t$ , such that

$$\partial_s^f \partial_t^g F(s, t) = \sum_{i=0}^n \sum_{j=0}^m \binom{n}{i} \binom{m}{j} \partial_s^f [s^i (1-s)^{n-i}] \partial_t^g [t^j (1-t)^{m-j}] P_{ij}$$

Thus, evaluating at  $s=0$  and  $t=0$ , and using (91), the above becomes

$$\begin{aligned}
& \left. \partial_s^f \partial_t^g F(s, t) \right|_{s=0, t=0} \\
&= \sum_{i=0}^n \sum_{j=0}^m \binom{n}{i} \binom{m}{j} \left[ \binom{f}{i} i! (-1)^{f-i} \frac{(n-i)!}{(n-f)!} \right] \left[ \binom{g}{j} j! (-1)^{g-j} \frac{(m-j)!}{(m-g)!} \right] P_{ij} \\
&= \sum_{i=0}^n \sum_{j=0}^m (-1)^{f+g-i-j} \frac{n!}{(n-f)!} \frac{m!}{(m-g)!} \binom{f}{i} \binom{g}{j} P_{ij} \\
&= \frac{n!}{(n-f)!} \frac{m!}{(m-g)!} \sum_{i=0}^n \sum_{j=0}^m (-1)^{f+g-i-j} \binom{f}{i} \binom{g}{j} P_{ij} \tag{92}
\end{aligned}$$

for  $i \leq f \leq n$  and  $j \leq g \leq m$  (zero otherwise)

However, since the above is valid only for  $i \leq f$  and  $j \leq g$ , the two sums are valid only up to  $f$  and  $g$  respectively, so that (92) simplifies to

$$\left. \partial_s^f \partial_t^g F(s, t) \right|_{s=0, t=0} = \frac{n!}{(n-f)!} \frac{m!}{(m-g)!} \sum_{i=0}^f \sum_{j=0}^g (-1)^{f+g+i+j} \binom{f}{i} \binom{g}{j} P_{ij} \tag{93}$$

where, of course, the identity  $(-1)^{-i} = (-1)^i$  was used. Now, note that at  $i=f$  and  $j=g$  the summation term in (93) simply gives  $P_{fg}$ . By using the Kronecker delta function, (93) can thus be written as

$$\begin{aligned}
& \left. \partial_s^f \partial_t^g F(s, t) \right|_{s=0, t=0} \\
&= \frac{n!}{(n-f)!} \frac{m!}{(m-g)!} \left[ P_{fg} - \sum_{i=0}^f \sum_{j=0}^g (-1)^{1+f+g+i+j} (1 - \delta_{if} \delta_{jg}) \binom{f}{i} \binom{g}{j} P_{ij} \right] \tag{94}
\end{aligned}$$

This completes the proof, since the bracket term is none other than the nodal formulation (45) of Definition-1, namely

$$\left. \partial_s^f \partial_t^g F(s, t) \right|_{s=0, t=0} = \frac{n!}{(n-f)!} \frac{m!}{(m-g)!} h^{fg} \bar{u}^{fg}$$

□

## 4 Implementation in the non-linear full-MHD code JOREK

### 4.1 The JOREK code

The above formulation is implemented in the JOREK code [15–18]. JOREK uses a finite-element grid in the poloidal plane, with finite Fourier series in the toroidal direction. The finite elements are

quadrangular Bezier elements using a nodal formulation equivalent to the generalised formulation in Definition-1 and Definition-2 with polynomials of order 3, and thus C-continuity of order 1 [17]. The Fourier series used in the toroidal direction can be chosen with arbitrary toroidal periodicity, meaning that with 3 Fourier modes and a toroidal periodicity of 4, the physical toroidal mode numbers simulated would be  $n=4,8,12$ . Typically, when producing linear stability scans, a single Fourier mode is used, while scanning the toroidal periodicity. For non-linear simulations, multiple Fourier modes are used, with a periodicity of 1 or 2.

The time discretisation used in JOREK is fully implicit, with a choice between the Crank-Nicolson or the Gear's scheme. This implicit method results in a linearised system of equations, for which a sparse matrix needs to be inverted. This is done either with a direct solve, or using a pre-conditioned GMRES iterative solver. In practice, the direct solve approach is only used for the toroidally axisymmetric kinetic equilibrium  $n=0$ , while for non-linear cases, the GMRES method is employed. The GMRES pre-conditioner is equivalent to a Block-Jacobi pre-conditioner, which is obtained by solving each toroidal harmonic block of the matrix individually (ie. without the harmonic coupling), which is done in parallel with a direct solve. The direct solves (either for the pre-conditioner harmonic-blocks, or for the whole system) is done using open-source solvers like MUMPS [32–34], PASTIX [35, 36] or STRUMPACK [37, 38].

The Weak Formulation method [39] is used for all equations, which are multiplied by a test-function and integrated over the element surfaces. The test-functions are chosen to be the basis functions obtained from the nodal formulation, as in (6), or as described in the next section. One of the main advantages of the weak formulation is that it allows integration by parts of divergence terms, which allows the removal of all 2nd order derivatives from the system. This was a significant advantage when using C1-continuous finite-elements in the past, like those implemented in JOREK [17], since all terms in the equations were guaranteed to be smoothly represented, thus improving numerical stability.

The physics models of JOREK includes visco-resistive models for both full-MHD and reduced-MHD [40]. The equations are normalised using two constants: vacuum permeability  $\mu_0$ , and the central density  $\rho_0$ . This normalisation is similar to the Alfvén time normalisation, such that for a deuterium plasma with central particle density  $n_o = 6 \times 10^{19} \text{m}^{-3}$ , a normalized time unit corresponds to approximately  $0.5 \mu\text{s}$ . Note that this is a pseudo-normalisation, where not all variables are dimensionless in the final formulation. In particular, the magnetic field is not normalised.

## 4.2 The basis functions

The full formulation of any variable  $\psi$  (as well as spatial variables  $R$  and  $Z$ ) on the finite elements, including the Fourier series, is given by

$$\psi(s, t) = \sum_{j=1}^4 \sum_{k=0}^{\frac{n-1}{2}} \sum_{l=0}^{\frac{n-1}{2}} \sum_{m=0}^N \left[ \psi_{jm}^{kl} F_{klj}(s, t) h_j^{kl} e^{im\phi} \right] \quad (95)$$

$j$  : sum on 4 element nodes

$k$  : sum on first vector index

$l$  : sum on second vector index

$m$  : sum on N Fourier modes

$F_{j,k,l}(s, t)$  : nodal basis functions

$h_j^{kl}$  : element sizes

$\phi$  : toroidal angle

$\psi_{jm}^{kl}$  : variable values for each degree of freedom

Note, with respect to the generalised formulation (45), the  $\psi_{jm}^{kl}$  values are the equivalent of the vectorial basis  $\vec{u}^{ij}$ , but for any variable  $\psi$ , including spatial coordinates. Once the nodal formulation (45) has been calculated for a given polynomial order  $n$ , the basis functions  $F_{jkl}$  of this nodal basis need to be calculated as the coefficients obtained from the factorization of the Bezier formula (3) with respect to each  $\vec{u}^{ij}$  term. For bi-cubic C1-continuous elements, this is easily done by hand. For bi-quintic C2-continuous elements, the basis functions and their derivatives with respect to  $s$  and  $t$  can be calculated by hand, although this is already tedious. The  $s$ - and  $t$ -derivatives are required up to order 2 in the code, which means 5 derivatives in total (plus the absolute value), for each basis function component. For bi-quintic C2-continuous elements, this gives a total of: 4 nodes, times  $[(n+1)/2]^2$  basis vectors, times 6 derivatives (including zeroth derivative); hence 216 functions to calculate and include in the code. For elements of order  $n = 7$ , there are 384, and for  $n = 11$ , there are 864.

Thus, although the basis functions and their derivatives can be calculated by hand for bi-quintic elements, at higher order this is not feasible without a considerable risk of human error. It is much safer to do this analytical calculation numerically, which can be done relatively easily with analytical algebra libraries like Python's symbols library, Sympy [41]. Using this library, the Fortran module of JOREK that includes the basis function formulas is auto-generated for any polynomial order  $n$ .

### 4.3 Gaussian integration

The weak-formulation of JOREK requires integration over each element. This is done using Gaussian quadrature, for which the number of Gaussian integration points (and weights) needs to correspond to the polynomial order of the Bezier elements. Although trivially obtained from Python libraries like Numpy [42], this also requires the auto-generation of a small module in the JOREK code (for polynomial orders  $n > 7$ ), to register these Gaussian integration points and weights.

## 4.4 The mesh generators and initial conditions

The JOEREK code includes a large variety of mesh generators, adapted to complex magnetic configurations in tokamaks, and with adaptive extensions to arbitrary wall structures [43]. These grid generators had to be adapted to include higher order polynomials. The alignment of the grids, along the background magnetic flux surfaces, is done up to second derivatives, which is largely sufficient for most applications, and thus for polynomial orders  $n \geq 7$ , the derivatives of order  $\geq 3$  are set to zero. In the future, if higher accuracy is required, for Bezier elements of order  $n \geq 7$ , these higher-order derivatives could be included in the grid generators.

Initial conditions must be set on the generated grids, before time-evolution is simulated. This is done by projecting the initial condition on the node coefficients, simply by solving the weak-formulated system of the equation  $\psi = \psi_{init}$ , where  $\psi_{init}$  are the known initial values (Right-Hand Side), and  $\psi$  represents the linearised matrix with contributions from all degrees of freedom  $\psi_{j,m=0}^{kl}$ .

## 4.5 Replacing Root Solvers

Several root-solver routines inside JOEREK cannot be used with bi-quintic (or higher-order) elements. These root-solvers are used to find values along the sides of elements, either to determine the spatial minima/maxima of an element, or to find the intersections of flux-surfaces along element edges, used for flux-aligned grid construction. For example, with bi-cubic elements, finding the location of a  $\psi$ -value along the side of an element simply requires the analytical formula for the solution of a 3rd order polynomial. Since there are no analytical solution for polynomials of order 5 or higher, locating root values along element sides requires converging algorithms like Newton's method. Of course Newton's method can only be used after first determining the location of minima/maxima along the side of an element; once this is achieved, Newton's method can be used to locate roots between consecutive minima/maxima. Note that locating minima/maxima along the side of a bi-quintic element requires the zero solutions of the derivative of a given variable  $\psi$ , thus a 4th order polynomial. While there exists a formulation for the roots of 4th order polynomials, since this cannot be done for polynomial orders  $n \geq 7$ , Newton's method is also used to find these roots to determine the minima/maxima.

## 4.6 Long-integer matrix solvers

In theory, for a given simulation grid, if higher polynomial Bezier elements are used, the spatial resolution of the grid can be diminished. However, in practice, particularly when addressing performance and convergence properties of the elements, long-integers (64 bit) may be needed for the sparse-matrix representation and the solvers. For large non-linear problems, the matrix sent to direct solvers like PastiX is determined by the grid size, ie. the number of nodes  $n_n$ , the number of degrees of freedom per node  $n_d$  polynomial basis, and the number of variables  $n_v$  in the physics system. Indeed, since the GMRES preconditioner is obtained by solving each Fourier harmonic

individually, the size of these matrices is independent of the number of Fourier modes. Thus, each matrix size is, at most

$$m \times m = (2 n_n n_d n_v)^2 \quad (96)$$

where the factor 2 comes from the *sine* and *cosine* component of each Fourier mode. Of course, these are sparse matrices, but in practice, they will scale with the square of the number of degrees of freedom,  $n_d^2$ . Since the number of degrees of freedom per node scales as the square of the polynomial basis order,  $n^2$ , it means the size of the matrices will scale as  $n^4$ . In order to enable simulations with large spatial resolutions and large polynomial basis orders, parts of the JOREK code required modification to allow for a compressed sparse row (CSR) representation of the matrices using long-integers. Although straight-forward in most instances of the code, this also requires MPI communications of the matrix to be split into batches, since MPI communication indexing is restricted to short integers.

## 5 Testing the JOREK code

### 5.1 Linear MHD tests

Three linear benchmarks are conducted for core and edge MHD instabilities in toroidal geometry, using the reduced-MHD model. The first two linear benchmarks are a low- $\beta$   $m=n_\phi=1$  internal kink mode, and a low- $\beta$   $m=2$ ,  $n_\phi=1$  tearing-mode (where  $m$  is the dominant poloidal mode, and  $n_\phi$  the toroidal Fourier mode). Both instabilities were studied in previous publications [40], and they are included here for completeness and clarity. The two equilibria are similar, but differ in safety  $q$ -profile: the internal kink mode has a  $q$ -profile in the range [0.7, 1.6], crossing  $q=1$  at  $\psi_n=0.5$ , while the tearing mode has a  $q$ -profile in the range [1.7, 3.9], crossing  $q=2$  at  $\psi_n=0.3$ . The Grad-Shafranov equilibrium quantities and profiles for these two cases are described in more details in [40]. Since the equilibria both have circular poloidal cross-sections, polar grids are used.

The first two benchmark cases are run for a scan in resistivity. The kink mode is run with resistivity alone (without viscosity, and without particle or thermal diffusion), while the tearing mode is run including all diffusions, with  $\mu_0=10^{-8}\text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$ ,  $D_\perp=0.7\text{m}^2\cdot\text{s}^{-1}$ , and  $\kappa_\perp=1.7\times 10^{-8}\text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$ . For simplicity, the resistivity and viscosity are taken to be spatially constant for both cases. Note that only the toroidal mode  $n_\phi=1$  is simulated here, such that coupling with higher toroidal modes is not present in these linear benchmarks.

The third benchmark case is an X-point plasma with peeling-ballooning instabilities. This is an artificial equilibrium similar to the JET tokamak. It is run using the kinetic stationary background equilibrium flows ( $n_\phi=0$ ), together with a single toroidal harmonic, which is ranged from  $n_\phi=1$  up to  $n_\phi=20$ . The equilibrium characteristics, together with the diffusive parameters, are identical to those described in [44].

Figure-5 show the benchmark of the internal kink mode, the tearing mode, and the X-point peeling-ballooning modes, for various polynomial orders up to  $n=7$  with C3-continuity (and up

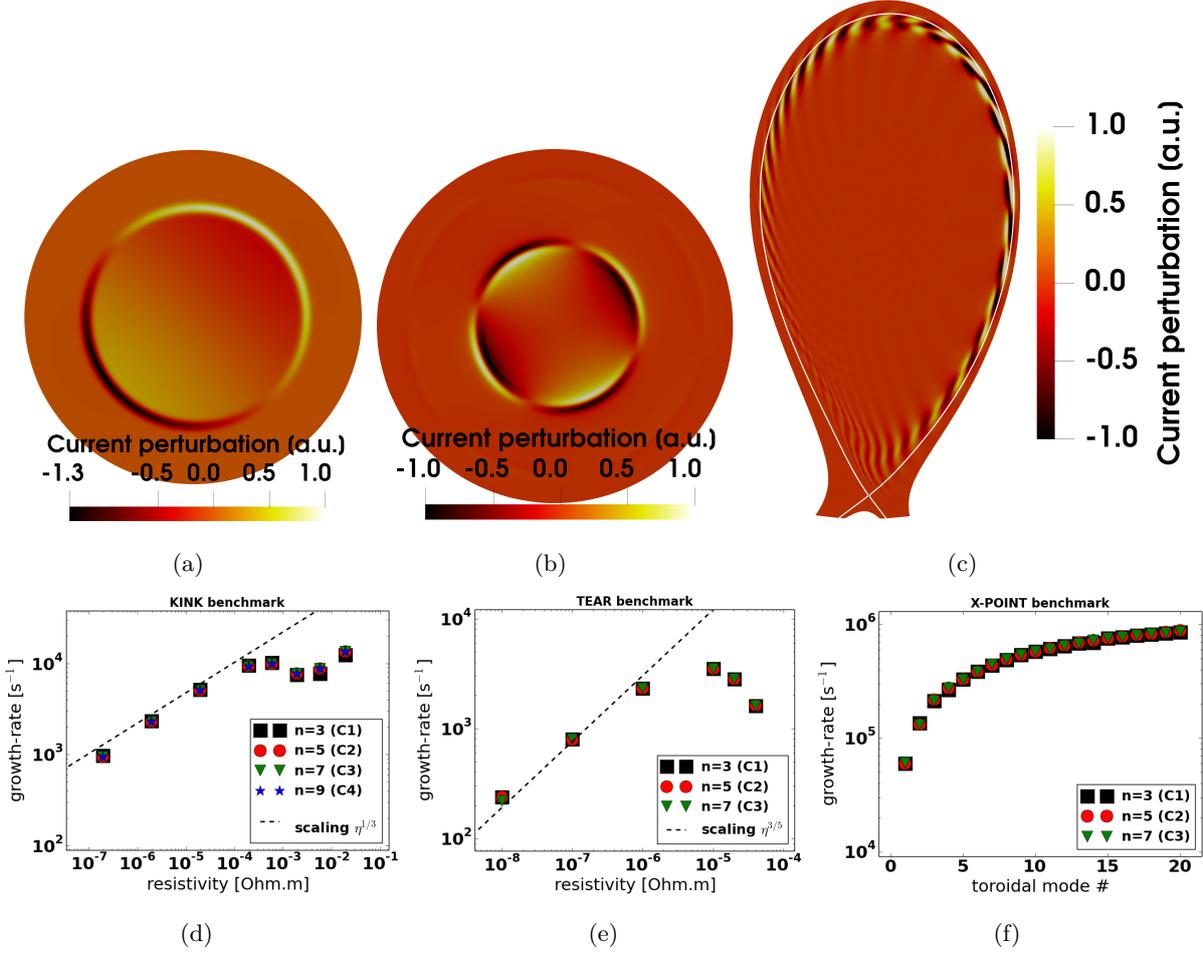


Figure 5:

$m=1$ ,  $n_\phi=1$  internal kink mode benchmark: (a) Poloidal cross-sections of the normalised perturbation of the toroidal current  $j_\phi$ , and (d) comparison of the growth rates between various polynomial orders, ranging from  $n=3$  to  $n=9$ , as a function of resistivity  $\eta$ .

$m=2$ ,  $n_\phi=1$  low- $\beta$  tearing mode benchmark: (b) Poloidal cross-sections of the normalised perturbation of the toroidal current  $j_\phi$ , and (e) comparison of the growth rates between various polynomial orders, ranging from  $n=3$  to  $n=9$ , as a function of resistivity  $\eta$ .

Peeling-ballooning mode benchmark with X-point equilibrium: (c) Poloidal cross-sections of the normalised perturbation of the toroidal current  $j_\phi$  for the Fourier mode  $n_\phi=17$ , and (f) comparison of the growth rates between various polynomial orders, ranging from  $n=3$  to  $n=7$ , as a function of the toroidal mode number  $n_\phi$ .

to  $n=9$ , C4-continuity for the internal kink). Poloidal cross-sections of the main Fourier mode perturbation are shown for the toroidal current  $j_\phi$ , together with the growth rates of the modes, plotted as a function of resistivity for the internal kink and tearing mode cases, and as a function of toroidal mode number for the peeling-ballooning mode. The linear theory in the ideal (low resistivity) regimes is also plotted, with a  $\eta^{1/3}$  scaling for the internal kink mode [45], and a  $\eta^{3/5}$  scaling for the tearing mode [46]. For the peeling-ballooning mode, the poloidal cross-section pictures the toroidal Fourier mode  $n_\phi=17$ . The benchmarks are, as expected, identical in all cases.

## 5.2 Non-linear MHD tests

A simple non-linear simulation is run to test that results are well reproduced at higher polynomial orders in non-linear regimes. This case is an Edge-Localised-Mode in a discharge from the MAST tokamak. The particularity of this case, also described in details in [44], is that plasma filaments are expelled from the edge plasma, travelling at large radial velocities (several km/s) into the far Scrape-Off Layer. For this benchmark, the C1- and C2-continuous versions are run (ie. with polynomial orders  $n=3$  and  $n=5$  respectively) and compared after the filaments have crossed the separatrix. Higher spatial resolution is used for the C1 case than for the C2 case, which is required to resolve the filaments accurately.

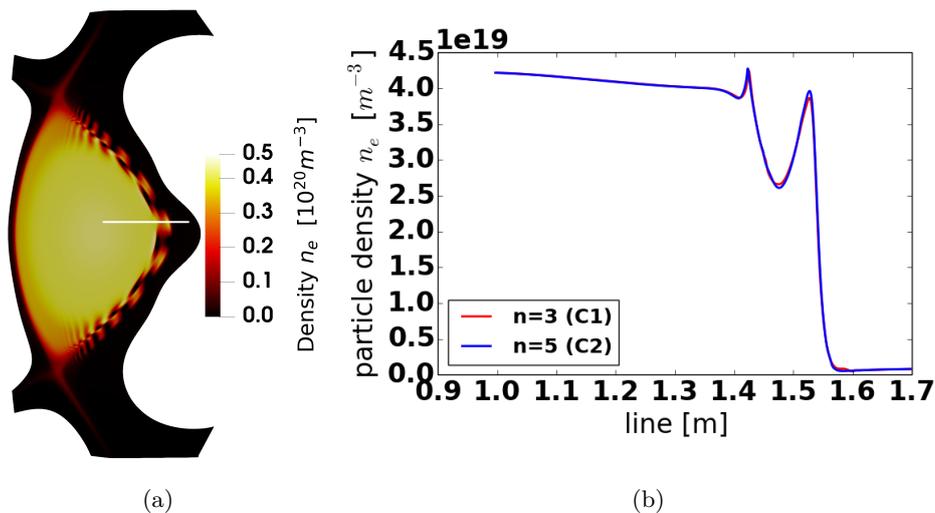


Figure 6:

(a) Poloidal cross-section picturing the particle density  $n_e$  during an Edge-Localised-Mode (ELM) in the MAST tokamak, during which plasma filaments are expelled at several km/s into the Scrape-Off Layer. (b) Particle density profiles across a filament while it is moving away from the plasma, with a comparison between the C1-continuous ( $n=3$ ) and C2-continuous ( $n=5$ ) finite elements. These radial profiles are taken along the white horizontal line pictured in (a).

Figure-6 shows a poloidal cross-section of the MAST tokamak, as the filaments travel into the SOL. A radial profile of density is taken across the filament, and compared between the C1 and C2 runs. Negligible differences are found between the two profiles, which is to be expected due to the accuracy differences between the two cases, and the resulting effect this may have due to the non-linear nature of the run. Nevertheless, this benchmark is also extremely positive.

## 5.3 C-continuity Verification

Verification of C-continuity is done up to polynomial order  $n=7$  (C3-continuous), by comparing between subsequent polynomial orders of finite elements, at fixed spatial resolution (ie. fixed element size). This is done by measuring linear profiles across a sharp perturbation of the toroidal current  $j_\phi$ .

Figure-7 shows the comparison for 2nd and 3rd derivatives between corresponding polynomial orders for the 1D profiles. This result clearly shows that the discontinuity of derivatives at element

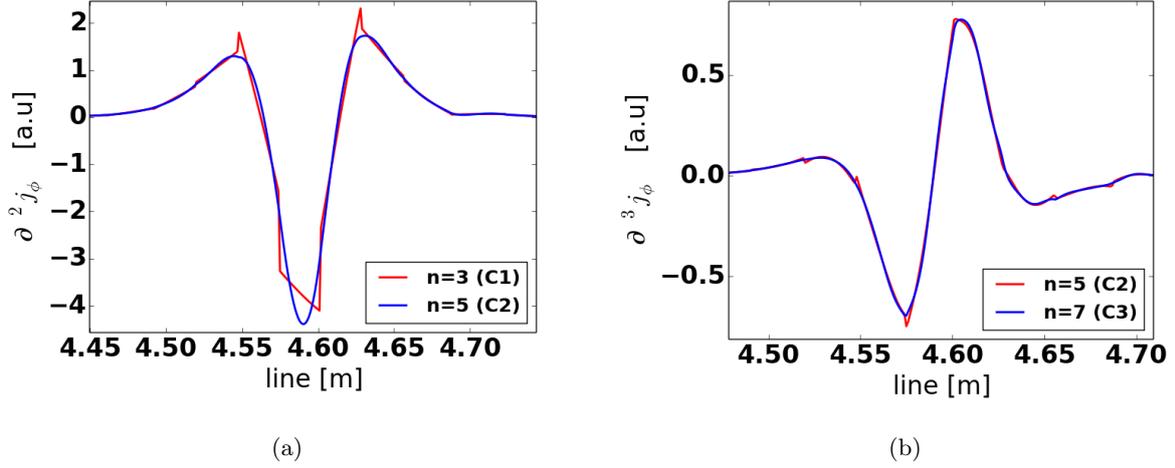


Figure 7:

(a) Comparison of second derivatives across a sharp perturbation of the toroidal current  $j_\phi$ , between the C1-continuous ( $n=3$ ) and C2-continuous ( $n=5$ ) finite elements. (b) Comparison of third derivatives across a sharp perturbation of the current  $j$ , between the C2-continuous ( $n=5$ ) and C3-continuous ( $n=7$ ) finite elements. Note that all cases are done at equal spatial resolution (ie. elements are the same size in all cases). The plots reveal the discontinuity of the current derivatives (red curves), which becomes smooth at the relevant C-continuous polynomial order (blue curves).

boundaries (represented by jumps in the red profiles), disappears entirely at the relevant polynomial order. Figure-8 shows 3D warp surfaces of the corresponding quantities  $\partial_R j_\phi$  and  $\partial_R^2 j_\phi$  for the C2-continuous case, revealing the smoothness of derivatives.

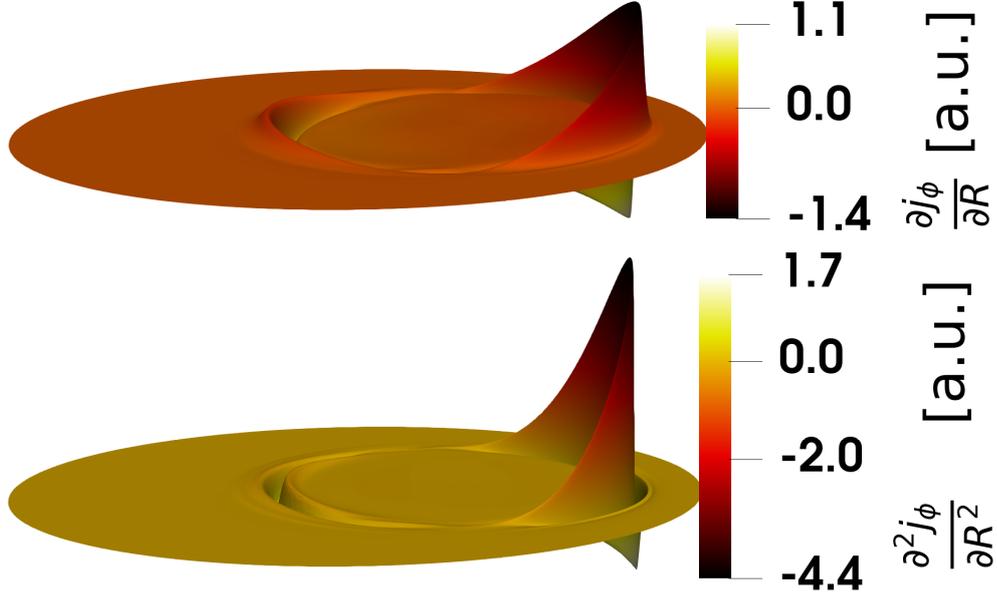


Figure 8: 3D warp surfaces of  $\partial_R j_\phi$  (top) and  $\partial_R^2 j_\phi$  (bottom).

## 5.4 Convergence Tests

One of the main interests of using higher-order polynomials with finite elements is that the local numerical errors diminish significantly. Convergence of growth rates as a function of spatial grid resolution is tested for a standard ballooning benchmark, the so-called CBM18 ballooning case, also described in details in [40]. The grid resolution is scanned homogeneously in the radial and poloidal directions, scanning from  $(n_{flux}, n_{tht})=(54, 180)$  to  $(144, 480)$ , where  $n_{flux}$  and  $n_{tht}$  are the number of radial (flux surfaces) and poloidal (theta) grid-points, respectively, both of which are equidistant in real-space for this case. With finite elements, the local error is estimated as  $E \sim h^p$  [39], where  $h$  is the element size, and  $p = n + 1$  is the polynomial order of the finite elements. In this case,  $(\sqrt{n_{flux}n_{tht}})^{-1}$  is used as an approximation of the element size  $h$ . Note that these tests evaluate the growth rates of toroidal modes, which are obtained by integrating the mode energies over the domain (ie. over the elements), hence adding another factor  $(\sqrt{n_{flux}n_{tht}})^{-1}$  to the error estimate. Thus, the error of the growth rates should scale with the  $(p + 1)$  power of the spatial resolution, ie.  $(\sqrt{n_{flux}n_{tht}})^{-5}$  for bi-cubic elements,  $(\sqrt{n_{flux}n_{tht}})^{-7}$  for bi-quintic elements, etc.

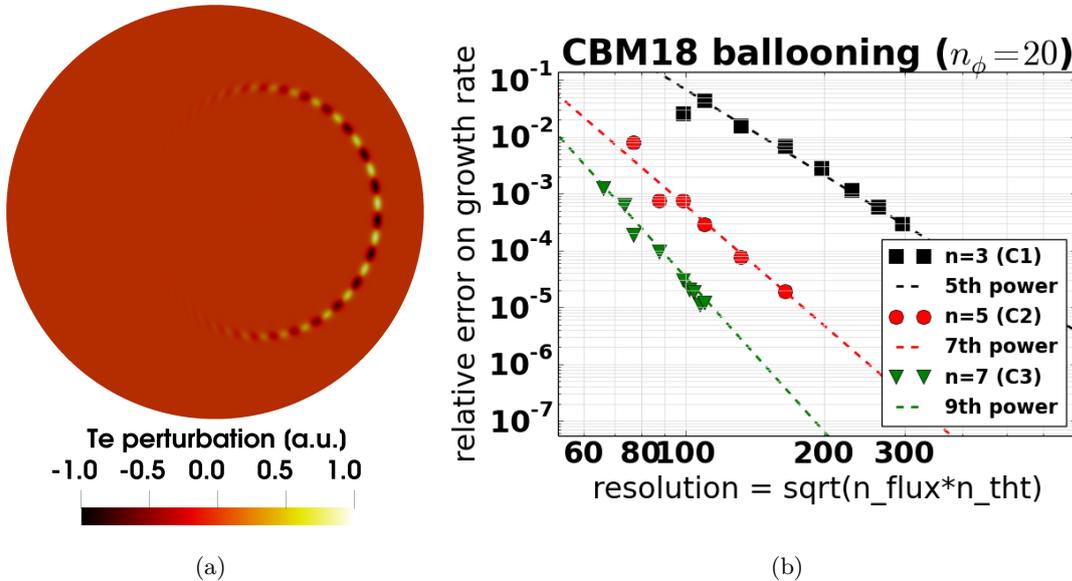


Figure 9:

(a) Poloidal cross section of the ballooning perturbations (mode number  $n_\phi=20$ ) of the electron temperature  $T_e$  for the CBM18 test-case used to verify convergence of the error with respect to resolution. (b) Convergence of the error on the growth rates, for the CBM18 test-case with the ballooning mode number  $n_\phi=20$ , as a function of spatial grid resolution. The error converges according to the estimate  $E \sim h^p$  [39].

Figure-9a shows the convergence of the growth rate error, as a function of spatial resolution. The scaling of this convergence is, as expected, dependent on the corresponding p-refinement level of the finite-elements, as long as the resolution is above a certain minimum (vertical dashed line). The converged growth rates, used to evaluate the errors in Figure-9, are obtained using the highest resolution cases available. A least-square fit is done for the expected convergence, such that the converged growth rate is estimated at “infinite” resolution.

This CBM18 test case is a representative example of both the advantages and the limitations of higher-order finite-elements. As can be seen in Figure-9b, below a certain poloidal resolution, the fine ballooning structure of Figure-9a cannot be resolved properly, and the error starts deviating from the scaling. Both high radial and poloidal resolution is required to resolve the ballooning filaments, and this requirement increases with the toroidal mode number  $n_\phi$ . Depending on the physics addressed, higher-order finite-elements may not always be the preferred choice. For example if addressing turbulence, which requires mode numbers upwards of  $n_\phi=40$ , the high minimal resolution required may become computationally too expensive when combined with high-order polynomial elements. Users need to find the balance between the numerical stability of the physics model, the spatial scale of non-linear structures to be resolved, and the computational cost of simulations. The highest resolution case with C3-continuity required close to 12,000 cores on the CINECA-Marconi cluster, due to the memory requirement for solving the larger sparse matrices at higher polynomial order. However the lowest C3-continuous resolution case required only 192 cores, and still provides reasonable accuracy. The next subsection provides a more detailed evaluation of the computational gains versus accuracy.

## 5.5 Performance Tests

Figure-10a shows the computation time as a function of the polynomial order of the finite elements, up to  $n=13$  (C6-continuous). As expected, the computation time scales as the square of the number of degrees of freedom per node (because the matrix dimension scales as  $n_{degrees}$  in both directions), hence a 4th order scaling with the order of the polynomial basis. Figure-10b shows the computation time as a function of the error on the growth-rate, corresponding to the CBM18 test-case shown in Figure-9. The alignment of different orders is very clear, and higher accuracy requires higher computational cost, as expected. However, the important point to notice here is that at a given error level, the higher-order finite-element cases are at least as fast, in several cases cheaper in terms of CPU-hours.

Note that an additional modifications, left for future development, could further reduce the cost of higher-order finite elements. Firstly, the reduced-MHD model, frequently used in JOREK, uses separate equations for the variables  $w_\phi$  and  $j_\phi$ . The reason for this is that these variables are themselves 2nd order derivatives of other variables, such that solving these separately reduces numerical errors in C1-continuous elements. With higher C-continuity, these could be removed, reducing the number of variables from 7 to 5, which would significantly reduce the size of the problem. Secondly, using localised p-refinement would also result in lower computational costs, while keeping the desired accuracy where it is needed (for example in the region where ballooning structures are dominant in Figure-9a). These modifications would however require significant development and testing, which is beyond the scope of this paper and left for future work.

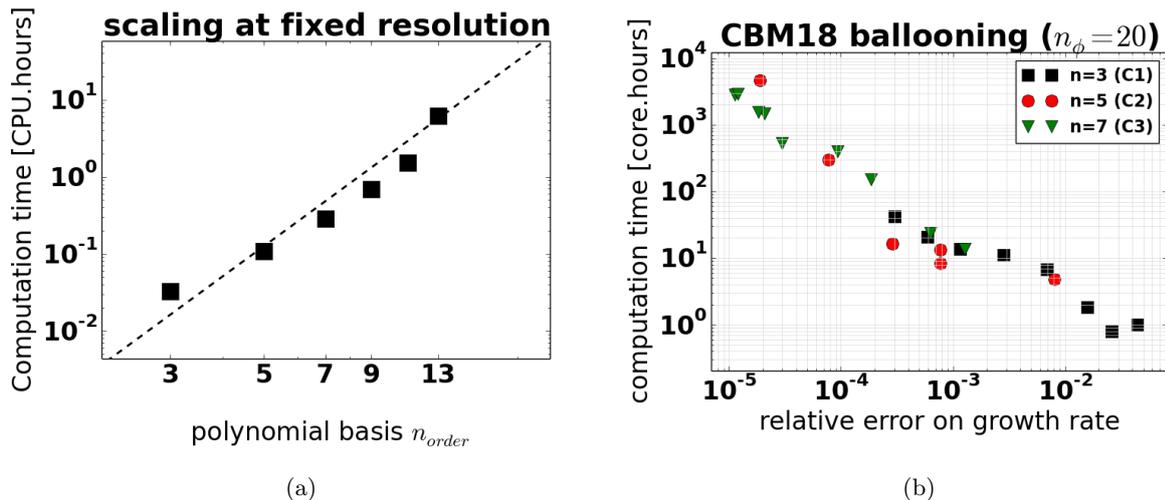


Figure 10:

(a) Scaling of the computation time, at fixed spatial grid resolution, as function of the polynomial basis, up to  $n=13$  (C6-continuous). The dashed line shows the scaling of  $n^4$ , which matches the calculations as expected. (b) Computation cost of the convergence test case from Figure-9, as a function of error. This clearly shows that, between C1- and C2- cases, at equivalent accuracy, the C2 runs are computationally cheaper.

## 5.6 Advanced simulation demonstration

Finally, to demonstrate the practical usability of the new finite-element method implemented in JOREK, a non-linear simulation is run, with multiple Fourier mode numbers  $n_\phi=(1,2,3,\dots,8)$ , for an Edge-Localised-Mode in ITER. This is done for a 15MA equilibrium, with a pedestal electron temperature of  $T_e^{ped} = 8\text{keV}$ . An advanced grid geometry is used, aligned to the first-wall of ITER, including the complex Dome geometry below the X-point, based on the grid-patches method developed in [43]. The simulation is run with conservative physical diffusivity coefficients, since its purpose is simply to demonstrate numerical feasibility of violent MHD non-linear instabilities. The resistivity is taken to be 300 times the neoclassical resistivity, and diamagnetic drifts are ignored. More advanced studies of ITER plasmas, at experimental resistivity and with diamagnetic effects, are the current focus of other experts in the JOREK team, and are well beyond the scope and purpose of this paper.

Figure-11 shows the density and electron temperature as ELM filaments are expelled from the plasma. Fine, well-resolved filamentary structures can be observed reaching the divertor. The finite-elements are also pictures, in the divertor and X-point region, in Figure-11b. The kinetic energy of the toroidal modes is pictured in Figure-11c, where the ELM starts with the toroidal mode number  $n_\phi=8$ , followed by coupling with lower harmonics, as is typical of non-linear ELM dynamics [47].

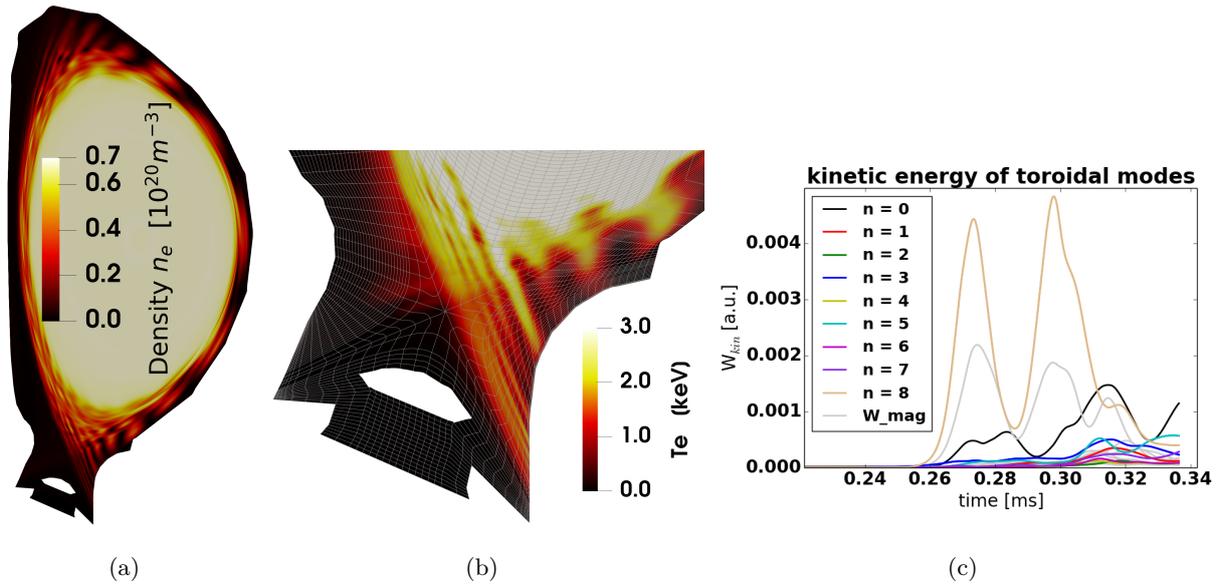


Figure 11:

ITER simulation with  $C2$ -continuous elements, with a wall-aligned grid, including the divertor Dome below the X-point. (a) Poloidal cross-section of the density, showing filaments traveling radially in the Scrape-Off Layer. (b) Zoom on the divertor region, showing thin stripes of temperature reaching the divertor target, due to heat conduction from upstream filaments along magnetic field-lines. (c) Kinetic energies of the ballooning modes as a function of time, where  $n_\phi=8$  dominates the early phase of the ELM, followed by non-linear coupling with lower modes in the later phase.

## 6 Conclusion

### 6.1 Summary

This paper presents the significant development and generalisation of the finite-element methods used in the JOREK code. A generalised formulation was developed for a nodal representation of Bezier elements, which ensures that C-continuity of order  $(n - 1)/2$  is respected, where  $n$  is the order of the polynomial basis of the Bezier elements. Rigorous proof is provided to show that the definition of our nodal formulation ensures C-continuity. Some details about the key aspects of the implementation into the code are provided, followed by a series of tests and benchmark studies. Beyond the fact that the C-continuity of the method is demonstrated numerically, and that all benchmark tests provide identical results to the previous JOREK version, one of the key achievements of this study is that, at equivalent numerical precision, having higher-order finite-elements can be computationally cheaper. As such, it is expected that the  $C2$ -continuous version of the JOREK code, possibly the  $C3$ -version, will become the default for all JOREK studies in the near future.

### 6.2 Further Work

There are, nevertheless, several areas where improvements can be brought to the JOREK code, following the implementation of this new method.

The first aspect is that a more generic definition of continuity, as in [29, 30], could be considered

at least for the C1- and C2-continuous versions. This is far more elaborate than the mathematical methods presented here, and would require significant code developments. However, there is an invaluable advantage to such generic methods, because they allow sharp angles between elements at a given node, which would enable more advanced element constructions, with arbitrary numbers of elements meeting at a given node, and the construction of triangular element sub-structures. This great flexibility could prove beneficial in the context of arbitrary grid construction, particularly in the context of alignment to arbitrary wall structure like in Figure-11.

The second point that could be improved, and of significant impact, would be the elimination of the two equations for the variables  $w_\phi$  and  $j_\phi$  in the reduced-MHD model. These equations are simply the identity definitions of  $w_\phi = \nabla^2\Phi$  and  $j_\phi = \nabla^2\psi$ , where  $\Phi$  and  $\psi$  are the electric and magnetic potentials respectively. These identities are time-independent, but they are included as separate equations in the model because this brings improved numerical stability to the model. If higher-order continuity is available, these equations could be removed, in principle, such that 3rd and 4th order derivatives of  $\Phi$  and  $\psi$  would then be directly used in the momentum and induction equations (as opposed to 1st and 2nd derivatives of  $w_\phi$  and  $j_\phi$  at present). This would reduce the number of equations/variables from 7 to 5 for the standard reduced-MHD model, which would significantly reduce the overall cost of simulation.

The third aspect concerns boundary conditions for Mach-1 Sheath conditions for boundaries with incident magnetic field lines. At present, the Sheath boundary conditions are constrained with the weak-formulation, such that no assumption is made for higher-order derivatives, but the Mach-1 boundary conditions are enforced by a penalisation method projected onto specific node degrees of freedom. With higher-order finite-elements, this is done for first-order derivatives, but not 2nd or higher derivative degrees of freedom. It would be highly desirable to implement this feature in the code, but numerical difficulties were found in our initial attempts to implement this, and so this will require further investigation. However, the current version of the code, with Mach-1 boundary conditions applied to the values and first derivatives of the plasma velocity, is numerically very stable and sufficient for production studies. It is, of course, at least as precise as the previous version.

The fourth area, where additional development could bring significant flexibility, is the possibility of having so-called “localised p-refinement”. The degrees of freedom per node on a given element is entirely independent of other neighbouring elements, such that different polynomial orders could be used on different elements. This would require more advanced developments in the code, but it is not that complex. It can be viewed in two ways: adding higher p-refinement locally means increasing the degrees of freedom compared to the one used initially, which may be difficult. However, approaching the issue from the other direction is very trivial: it simply requires to throw away unwanted degrees of freedom. The only major difference, in the code, is that different basis functions would be required for different elements. Nevertheless, this is an appealing route to explore, and one that is used in many other state-of-the-art modelling tools like Nektar++ [48, 49].

In terms of the mathematical method itself, the existence of such a generalised method for

triangular Bezier elements would need to be demonstrated. It may also be of interest to address more generic finite elements than Bezier, like B-splines, and using basis functions other than the Bernstein polynomials.

Finally, now that this new method is available in the JOREK code, extensive studies and routine usability still remains to be demonstrated by the JOREK team, hopefully leading to cheaper, more precise simulations in the future.

## **Acknowledgement**

This work was performed with the support of the JOREK Team [See <https://www.jorek.eu> for the present list of team members].

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. To obtain further information on the data and models underlying this paper please contact [PublicationsManager@ccfe.ac.uk](mailto:PublicationsManager@ccfe.ac.uk). The views and opinions expressed herein do not necessarily reflect those of the European Commission or the ITER Organization.

This work has been carried out within the framework of the RCUK Energy Programme [grant number EP/I501045].

This work was performed using the MARCONI computer at CINECA in Italy, within the EUROfusion framework.

This work was performed using the Cambridge Service for Data Driven Discovery (CSD3), part of which is operated by the University of Cambridge Research Computing on behalf of the STFC DiRAC HPC Facility ([www.dirac.ac.uk](http://www.dirac.ac.uk)). The DiRAC component of CSD3 was funded by BEIS capital funding via STFC capital grants ST/P002307/1 and ST/R002452/1 and STFC operations grant ST/R00689X/1. DiRAC is part of the National e-Infrastructure

## References

- [1] T. Eich, A. Herrmann, and J. Neuhauser. *Phys. Rev. Lett.* *91*, 195003, 2003.
- [2] B Sieglin, M Faitsch, T Eich, A Herrmann, and W Suttrop. *Physica Scripta* *2017*, T170, 2017.
- [3] C Ham, A Kirk, S Pamela, and H Wilson. *Nature Rev. Phys.* *2*, 159-167, 2020.
- [4] A.W.Leonard. *Phys. Plasmas* *21*, 090501., 2014.
- [5] C.Cheng, L Chen, and M S Chance. *Ann. Phys.* *161*, 21, 1985.
- [6] A.Dvornova, G. T. A. Huijsmans, S. Sharapov, F. J. Artola Such, P. Puglia, M. Hoelzl, S. Pamela, A. Fasoli, and D. Testa. *Phys. Plasmas* *27*, 012507., 2020.
- [7] M.Fitzgerald, J.Buchanan, R.J.Akers, B.N.Breizman, and S.E.Sharapov. *Comp. Phys. Comm.* *252*, 106773, 2020.
- [8] S. D. Pinches, I. T. Chapman, Ph. W. Lauber, H. J. C. Oliver, S. E. Sharapov, K. Shinohara, and K. Tani. *Phys. Plasmas* *22*, 021807, 2015.
- [9] A.H.Boozer. *Phys. Plasmas* *19*, 058101, 2012.
- [10] P. C. de Vries, G. Pautasso, D. Humphreys, M. Lehnen, S. Maruyama, J. A. Snipes, A. Vergara, and L. Zabeo. *Fus. Sci. Technol.* *69*, 471-484, 2016.
- [11] M.Lehnen, K.Aleynikova, P.B.Aleynikov, D.J.Campbell, P.Drewelow, N.W.Eidietis, Yu.Gasparyan, R.S.Granetz, Y.Gribov, N.Hartmann, E.M.Hollmann, and V.A.Izzo et al. *J. Nucl. Mater.* *463*, 39-48, 2015.
- [12] F. J. Artola, K. Lackner, G. T. A. Huijsmans, M. Hoelzl, E. Nardon, , and A. Loarte. *Phys. Plasmas* *27*, 032501, 2020.
- [13] D. Hu, E. Nardon, M. Lehnen, G.T.A. Huijsmans, and D.C. van Vugt. *Nucl. Fusion* *58*, 12, 2018.
- [14] V. Bandaru, M. Hoelzl, F. J. Artola, G. Papp, and G. T. A. Huijsmans. *Phys. Rev. E* *99*, 063317, 2019.
- [15] M.Hoelzl, G.T.A.Huijsmans, S.J.P.Pamela, M.Becoulet, E.Nardon, F.J.Artola, and B.Nkonga et al. *Nucl. Fusion (submitted); preprint at: <https://arxiv.org/abs/2011.09120>*, 2021.
- [16] G.T.A.Huysmans and O.Czarny. *Nucl. Fusion* *47*, 659, 2007.
- [17] O.Czarny and G.T.A.Huysmans. *J. Computational Phys.* *227*, 7423, 2008.
- [18] JOREK. <https://www.jorek.eu/>, 2020.
- [19] M3D-C1. <https://w3.pppl.gov/nferraro/m3dc1.html>, 2020.
- [20] S.C.Jardin. *J. Comp. Phys.* *200-133*, 2004.
- [21] NIMROD. <https://nimrodteam.org/>, 2020.
- [22] 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.Chufth, and the NIMROD Team. *J Comp. Phys.* *195*, 355, 2004.
- [23] H.Lutjens and JF Luciani. *J Comp. Phys.* *227*, 14, 6944-6966, 2008.
- [24] BOUT++. <https://boutproject.github.io/>, 2020.
- [25] B.D.Dudson, M.V.Umansky, X.Q.Xu, P.B.Snyder, and H.R.Wilson. *Comp. Phys. Comm.* *180*, 1467-1480, 2009.

- [26] Y. Todo and T. Sato. *Phys. Plasmas* 5 1321, 1998.
- [27] Y. Todo, H.L. Berk, and B.N. Breizman. *Nucl. Fusion* 52 094018, 2012.
- [28] A. Konies, S. Briguglio, N. Gorelenkov, T. Feher, M. Isaev, Ph. Lauber, A. Mishchenko, D.A. Spong, Y. Todo, W.A. Cooper, R. Hatzky, R. Kleiber, M. Borchardt, G. Vlad, A. Biancalani, and A. Bottino. *Nucl. Fusion* 58, 12, 2018.
- [29] W-H. Du and F.J.M. Schmitt. *Computer Aided Design, Butterworth-Heinemann Ltd, 0010-4485/90/090556-18*, 1990.
- [30] J. Zheng, G. Wang, and Y. Liang. *Computer Aided Geometric Design* 9, 321-335, 1992.
- [31] Binomial Coefficient Identities. Binomial coefficient identities. [https://en.wikipedia.org/wiki/Binomial\\_coefficient](https://en.wikipedia.org/wiki/Binomial_coefficient).
- [32] MUMPS website: <http://mumps.enseiht.fr/>, .
- [33] P.R. Amestoy, I.S. Duff, J. Koster, and J.-Y. L'Excellent. *SIAM Journal on Matrix Analysis and Applications* 23, 1, 15-41, 2001. doi: 10.1137/S0895479899358194.
- [34] P.R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet. *Parallel Computing* 32, 2, 136-156, 2006. doi: 10.1016/j.parco.2005.07.004.
- [35] P. Henon, P. Ramet, and J. Roman. *Parallel Comp.* 34 345-362, 2008.
- [36] G. Pichon, E. Darve, M. Faverge, P. Ramet, and J. Roman. *Journal of Computational Science* 27, 255-270, 2018. doi: 10.1016/j.jocs.2018.06.007.
- [37] STRUMPACK website: <https://portal.nersc.gov/project/sparse/strumpack/index.html>, .
- [38] Ieee international parallel and distributed processing symposium (ipdps), pp-897-906.
- [39] G. Strang and G. Fix. An analysis of the finite element method. *Wellesley-Cambridge Press, 2nd ed., ISBN-10: 0980232708, ISBN-13: 978-0980232707*, 2008.
- [40] S.J.P. Pamela, A. Bhole, G.T.A. Huijsmans, B. Nkonga, M. Hoelzl, I. Krebs, and E. Strumberger. *Phys. Plasmas* 27, 102510, 2020.
- [41] Sympy Python library. <https://www.sympy.org/en/index.html>, .
- [42] Numpy Python library. <https://numpy.org/doc/stable/contents.html>, .
- [43] S. Pamela, G. Huijsmans, A.J. Thornton, A. Kirk, S.F. Smith, M. Hoelzl, and T. Eich. *Comp. Phys. Comm.* 243, 41-50, 2019.
- [44] S. Pamela, G. Huysmans, and S. Benkadda. *Plasma Phys. Contr. Fusion* 52, 7, 2010.
- [45] R.J. Hastie, T.C. Hender, B.A. Carreras, L.A. Charlton, and J.A. Holmes. *Phys. Fluids* 30, 1756, 1987.
- [46] P.L. Pritchett, Y.C. Lee, and J.F. Drake. *Phys. Fluids* 23, 1368, 1980.
- [47] A. Cathey, M. Hoelzl, K. Lackner, G. Huijsmans, M. Dunne, E. Wolfrum, S. Pamela, F. Orain, and S. Gunter. *Nucl. Fusion* 60, 124007, 2020.
- [48] C.D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R.M. Kirby, and S.J. Sherwin. *Computer physics communications, vol. 192, pp. 205-219*, 2015.

- [49] D.Moxey, C.D.Cantwell, Y.Bao, A.Cassinelli, G.Castiglioni, S.Chun, E.Juda, E.Kazemi, K.Lackhove, J.Marcon, G.Mengaldo, D.Serson, M.Turner, H.Xu, J.Peiro, R.M.Kirby, and S.J.Sherwin. *Computer physics communications*, vol. 249, 107110, 2020.