






RESEARCH

Open Access



# DefElement: an encyclopedia of finite element definitions

Matthew W. Scroggs<sup>1\*</sup> , Pablo D. Brubeck<sup>2</sup> , Joseph P. Dean<sup>3</sup> , Jørgen S. Dokken<sup>4</sup>  and India Marsden<sup>2</sup> 

The results published in this paper passed the reproducibility check by the CSAE Code Review Board.

\*Correspondence: [matthew.scroggs.14@ucl.ac.uk](mailto:matthew.scroggs.14@ucl.ac.uk)

<sup>1</sup> Advanced Research Computing Centre, University College London, Gower Street, London WC1E 6BT, UK

<sup>2</sup> Department of Mathematics, Oxford University, Woodstock Road, Oxford OX2 6GG, UK

<sup>3</sup> Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK

<sup>4</sup> Simula Research Laboratory, Kristian Augusts Gate 23, 0164 Oslo, Norway

## Abstract

DefElement is an online encyclopedia of finite element definitions that was created and is maintained by the authors of this paper. DefElement aims to make information about elements defined in the literature easily available in a standard format. There are a number of open-source finite element libraries available, and it can be difficult to check that an implementation of an element in a library matches the element's definition in the literature or implementation in another library, especially when many libraries include variants of elements whose basis functions do not match exactly. In this paper, we carefully derive conditions under which elements can be considered equivalent and describe an algorithm that uses these conditions to verify that two implementations of a finite element are indeed variants of the same element. The results of scheduled runs of our implementation of this verification algorithm are included in the information available on the DefElement website.

**Keyword:** Finite element methods

## 1 Introduction

The finite element method has its origins in the 1940s, when methods involving the meshing of a domain began to be used by engineers [1–3]. The name *finite element method* was coined in a 1960 paper by Ray Clough [4]. Throughout the 1960s, significant progress was made in analysing the finite element method. The use of a triple to represent a finite element was introduced in 1975 in a set of lecture notes by Philippe Ciarlet [5], before appearing in his popular 1978 book [6]. This definition was first written with scalar-valued elements in mind, but it was used shortly after its introduction by Pierre-Arnaud Raviart and Jean-Marie Thomas [7] and Jean-Claude Nédélec [8] to define vector-valued H(div)-conforming and H(curl)-conforming elements.

Since its introduction, Ciarlet's triple—with some small notable adjustments—has become the widely accepted definition of a finite element, and it has been used to define a huge range of elements designed for specific applications including fluid dynamics [9, 10], plate deformation [11], and solid mechanics [12]. Details of Ciarlet's triple and an overview of different element types is presented in sect 2.

Often, only the lowest-order version of an element on a single cell type is introduced initially, with higher-order versions and versions on other cell types presented

© The Author(s) 2026. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

in subsequent papers. This can make tracking down appropriate elements that would be suitable for solving a particular problem a challenging task. To simplify this task, we have created DefElement, an online encyclopedia of finite element definitions hosted at [defelement.org](http://defelement.org)<sup>1</sup>. This old URL now forwards to [defelement.org](http://defelement.org). The name DefElement is a shortening of *definitions of elements*, and partially inspired by the use of the `def` keyword in Python [13]. DefElement contains the definitions of a large number of finite elements and useful information about each one. Our encyclopedia was partly inspired by the periodic table of finite elements [14], which displays the elements in four common de Rham families, and the Online Encyclopedia of Integer Sequences (OEIS) [15]. DefElement includes a page with information about each finite element. As shown in Fig. 1 for the Raviart–Thomas element, this information includes:

- alternative names of the element;
- details of the element’s definition;
- the polynomial and Lagrange subdegree and superdegree of the element, as we will define in section 2.4;
- the number of degrees of freedom (DOFs) that the element has in terms of the degree  $k$ ;
- details of any variants of the element, as we will define in section 3;
- details of libraries in which the element is implemented. Currently, the libraries Basix [16], Bempp-cl [17], FIAT [18], ndelement [19], and Symfem [20] are included
- some example low-degree elements, with functions and plots generated using Symfem. These plots, and all other diagrams on DefElement, are licensed under a Creative Commons license, so that they can be reused in papers and elsewhere as long as an attribution is included;
- a list of references where the full definition of the element can be found.

Alongside the implementation detail, the DefElement page shows the status of DefElement’s element verification—in Fig. 1, this status is shown via a green tick or orange or red cross next to a library’s name. The aim of element verification is to confirm that the implementation of the element in each library matches the definition given in the literature. Element verification will be discussed in more detail in Sect. 4.

We encourage readers of this paper to consider contributing to DefElement by updating the information currently available or adding new information and new elements. In Sect. 5, we outline how contributions can be made to DefElement. We finish with some concluding remarks in Sect. 6.

## 2 Defining finite elements

In this section, we give an overview of the definition of finite elements. First, we define the names we use for the sub-entities of a reference cell. These definitions are summarised for cells of topological dimensions 0 to 4 in Table 1, although these could trivially be extended to higher dimensional cells.

---

<sup>1</sup> DefElement was originally hosted at [defelement.com](http://defelement.com)

**Fig. 1** Information about the Raviart–Thomas element, as shown on its DefElement page at <https://defelement.org/elements/raviart-thomas.html> (accessed 11 December 2025). This information includes details of the element’s definition (top left), implementations that include the element and example low-degree elements (top right), and references to literature where full details of the element can be found. For Raviart–Thomas elements, the included references are [7, 8, 14, 21, 22]

**Definition 1 (Sub-entities)** The sub-entities of a polytope  $R \subset \mathbb{R}^d$  of dimension 0, 1, 2 and 3 are called vertices, edges, faces, and volumes (respectively). The *codimension* of a sub-entity is computed by subtracting the dimension of the sub-entity from the topological dimension  $d$ . Sub-entities of codimension 0, 1, 2 and 3 are called *the cell, facets, ridges* and *peaks* (respectively).

For  $k > 3$ , we do not introduce specific names for  $k$ -dimensional and  $k$ -codimensional sub-entities.

In the most general terms, a finite element can be defined as follows.

**Table 1** The entities of cells with topological dimensions 0 to 4

| Topological dimension | Entities by dimension |              |              |                |          |
|-----------------------|-----------------------|--------------|--------------|----------------|----------|
|                       | 0                     | 1            | 2            | 3              | 4        |
| 0 (a vertex)          | The cell              | –            | –            | –              | –        |
| 1 (an interval)       | Vertices/facets       | The cell     | –            | –              | –        |
| 2 (a polygon)         | Vertices/ridges       | Edges/facets | The cell     | –              | –        |
| 3 (a polyhedron)      | Vertices/peaks        | Edges/ridges | Faces/facets | The cell       | –        |
| 4                     | Vertices              | Edges/peaks  | Faces/ridges | Volumes/facets | The cell |

| Topological dimension | Entities by codimension |                 |                 |                |          |
|-----------------------|-------------------------|-----------------|-----------------|----------------|----------|
|                       | 0                       | 1               | 2               | 3              | 4        |
| 0 (a vertex)          | The cell                | –               | –               | –              | –        |
| 1 (an interval)       | The cell                | Vertices/facets | –               | –              | –        |
| 2 (a polygon)         | The cell                | Edges/facets    | Vertices/ridges | –              | –        |
| 3 (a polyhedron)      | The cell                | Faces/facets    | Edges/ridges    | Vertices/peaks | –        |
| 4                     | The cell                | Volumes/facets  | Faces/ridges    | Edges/peaks    | Vertices |

**Definition 2** (*Direct finite element*) A (direct) finite element is defined by the triple  $(C, \mathcal{V}, \mathcal{L})$ , where

- $C \subset \mathbb{R}^{d_g}$  is a  $d$ -dimensional cell in a mesh;
- $\mathcal{V}$  is a finite dimensional space on  $C$  of dimension  $n$ , usually a space of polynomials;
- $\mathcal{L} := \{l_0, \dots, l_{n-1}\}$  is a basis of the dual space  $\mathcal{V}^* := \{f : \mathcal{V} \rightarrow \mathbb{R} \mid f \text{ is linear}\}$ . Each functional  $l_i$  is associated with a sub-entity of  $C$ . A functional  $f \in \mathcal{L}$  associated with a sub-entity  $E \subseteq C$  only depends on a function's value restricted to  $E$ , i.e. if  $p|_E = q|_E$  then  $f(p) = f(q)$ .

The basis functions  $\{\phi_0, \dots, \phi_{n-1}\}$  of the finite element are defined by

$$l_i(\phi_j) = \delta_{ij} := \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases}$$

Using this definition, a different finite element can be defined for each physical cell in the mesh. This allows for a wide range of approaches, such as  $p$ -refinement where different polynomial degrees are used in different parts of the mesh.

When the element type and polynomial degree are the same for each cell, it is common to define a single finite element on a reference cell and use a map to transform basis functions from the reference cell to each physical cell in the mesh. A *reference-mapped* finite element that is defined for use in this case is defined as follows. In the literature, these are often referred to as Ciarlet elements, due to their definition first appearing in a set of lecture notes [5] and book [6] by Philippe Ciarlet.

**Definition 3** (*Reference-mapped finite element*) A reference-mapped finite element is defined by the triple  $(R, \mathcal{V}, \mathcal{L})$ , where

- $R \subset \mathbb{R}^d$  is the reference cell, usually an interval, polygon or polyhedron;
- $\mathcal{V}$  is a finite dimensional space on  $R$  of dimension  $n$ , usually a space of polynomials;
- $\mathcal{L} := \{l_0, \dots, l_{n-1}\}$  is a basis of the dual space  $\mathcal{V}^* := \{f : \mathcal{V} \rightarrow \mathbb{R} \mid f \text{ is linear}\}$ . As in Definition 2, each functional  $l_i$  is associated with a sub-entity of the reference cell  $R$ , with the value of any functional  $f \in \mathcal{L}$  associated with a sub-entity  $E \subseteq R$  only depending on a function's value restricted to  $E$ .

The reference basis functions  $\{\phi_0, \dots, \phi_{n-1}\}$  of the finite element are defined by

$$l_i(\phi_j) = \delta_{ij}.$$

In Definitions 1 and 3, the value  $d$  is called the topological dimension. This may differ from the geometric dimension  $d_g$  if, for example, a mesh of triangular cells is embedded in three-dimensional space. The *value shape* of a finite element is defined to be the shape of the range space of the polynomials in  $\mathcal{V}$ . For example, if  $\mathcal{V}$  is a set of vector-valued functions in  $\{f \mid f : R \rightarrow \mathbb{R}^3\}$  then the value shape is (3); if  $\mathcal{V}$  is a set of matrix-valued functions in  $\{f \mid f : R \rightarrow \mathbb{R}^{2 \times 2}\}$  then the value shape is (2, 2). The *value size* of a finite element is defined to be the product of all the entries of the value shape, and equal to 1 if the element is scalar-valued.

For reference-mapped elements, a geometric map  $g$  is defined that maps points on the reference cell to points on a physical cell and a push forward map is defined that maps the basis functions on the reference cell to functions on a physical cell [23–25]. The push forward map for an element is chosen so that it preserves specific properties of the element's basis functions. The inverse map that maps function values from a physical cell to the reference cell is called the pull back map. For some elements (such as serendipity [12]), better convergence is achieved if the geometric map is affine, as defined as follows.

**Definition 4** (*affine transformation*) Let  $g : R \rightarrow \mathbb{R}^{d_g}$  be a geometric map.  $g$  is affine if there exist a translation  $g_1 : R \rightarrow \mathbb{R}^d$  and a linear transformation  $g_2 : \mathbb{R}^d \rightarrow \mathbb{R}^{d_g}$  such that  $g = g_2 \circ g_1$ .

The functionals  $l_i \in \mathcal{L}$  are the (local) degrees-of-freedom (DOFs) of the finite element (note that we enumerate functionals and basis functions starting at 0). When a finite element function space is defined on a mesh, we associate a global DOF number with each local DOF on each cell. To ensure that the mapped space has the desired continuity properties, corresponding local DOFs on neighbouring cells that are associated with a shared sub-entity are assigned the same global DOF.

We note that associating each functional with a sub-entity of the reference cell was not included in Ciarlet's original definition of a finite element, but it is now common to include this as this association is an important part of most finite element implementations, as it allows for continuity between cells to be naturally implemented. For continuity to be effectively imposed in this way, it is common to make some additional assumptions about the finite element. Before stating these assumptions, we first define equivalent sets of functionals.

**Definition 5** (equivalent sets of functionals) Let  $\mathcal{L} = \{l_0, \dots, l_{a-1}\}$  and  $\tilde{\mathcal{L}} = \{\tilde{l}_0, \dots, \tilde{l}_{b-1}\}$  be two sets of functionals associated with entities  $E$  and  $\tilde{E}$ . Let  $\{\phi_0, \dots, \phi_{a-1}\}$  and  $\{\tilde{\phi}_0, \dots, \tilde{\phi}_{b-1}\}$  be the finite element basis functions corresponding to the functionals in  $\mathcal{L}$  and  $\tilde{\mathcal{L}}$ . The sets of functionals  $\mathcal{L}$  and  $\tilde{\mathcal{L}}$  are equivalent if:

- they have the same size (i.e.  $a = b$ );
- there exists a push forward map  $\mathcal{F}$  such that for all  $i = 0, \dots, a - 1$ ,

$$\tilde{l}_i(\mathcal{F}\phi_j) = \delta_{ij}.$$

For sets of functionals associated with the same sub-entity, the following lemma will be useful.

**Lemma 1** Let  $\mathcal{L} = \{l_0, \dots, l_{a-1}\}$  and  $\tilde{\mathcal{L}} = \{\tilde{l}_0, \dots, \tilde{l}_{a-1}\}$  be two sets of functionals associated with entity  $E$ . If  $\text{span } \mathcal{L} = \text{span } \tilde{\mathcal{L}}$ , then  $\mathcal{L}$  and  $\tilde{\mathcal{L}}$  are equivalent sets of functionals.

**Proof** As  $\text{span } \mathcal{L} = \text{span } \tilde{\mathcal{L}}$ , there exists a non-singular matrix  $\mathbf{A} \in \mathbb{R}^{a \times a}$  such that

$$\begin{pmatrix} \tilde{l}_0 \\ \vdots \\ \tilde{l}_{a-1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} l_0 \\ \vdots \\ l_{a-1} \end{pmatrix}. \tag{1}$$

Note that as  $\mathcal{L}$  and  $\tilde{\mathcal{L}}$  are not necessarily orthogonal bases,  $\mathbf{A}$  is not necessarily an orthogonal matrix.

Let  $\{\phi_0, \dots, \phi_{a-1}\}$  be the finite element basis functions corresponding to the functionals in  $\mathcal{L}$ . The aim is to show that the second bullet point in Definition 5 is true: we will see that this holds if we define  $\mathcal{F}$  as

$$\begin{pmatrix} \mathcal{F}\phi_0 \\ \vdots \\ \mathcal{F}\phi_{a-1} \end{pmatrix} = \mathbf{A}^{-\top} \begin{pmatrix} \phi_0 \\ \vdots \\ \phi_{a-1} \end{pmatrix}.$$

As the functionals  $\tilde{l}_i$  are linear, we can write

$$\begin{pmatrix} \tilde{l}_0(\mathcal{F}\phi_0) & \dots & \tilde{l}_{a-1}(\mathcal{F}\phi_0) \\ \vdots & \ddots & \vdots \\ \tilde{l}_0(\mathcal{F}\phi_{a-1}) & \dots & \tilde{l}_{a-1}(\mathcal{F}\phi_{a-1}) \end{pmatrix} = \mathbf{A}^{-\top} \begin{pmatrix} \tilde{l}_0(\phi_0) & \dots & \tilde{l}_{a-1}(\phi_0) \\ \vdots & \ddots & \vdots \\ \tilde{l}_0(\phi_{a-1}) & \dots & \tilde{l}_{a-1}(\phi_{a-1}) \end{pmatrix}. \tag{2}$$

Using (1), we see that

$$\begin{pmatrix} \tilde{l}_0(\phi_0) & \dots & \tilde{l}_{a-1}(\phi_0) \\ \vdots & \ddots & \vdots \\ \tilde{l}_0(\phi_{a-1}) & \dots & \tilde{l}_{a-1}(\phi_{a-1}) \end{pmatrix} = \begin{pmatrix} l_0(\phi_0) & \dots & l_{a-1}(\phi_0) \\ \vdots & \ddots & \vdots \\ l_0(\phi_{a-1}) & \dots & l_{a-1}(\phi_{a-1}) \end{pmatrix} \mathbf{A}^\top. \tag{3}$$

Combining (2) and (3), we see that

$$\begin{aligned}
 \begin{pmatrix} \tilde{l}_0(\mathcal{F}\phi_0) & \dots & \tilde{l}_{a-1}(\mathcal{F}\phi_0) \\ \vdots & \ddots & \vdots \\ \tilde{l}_0(\mathcal{F}\phi_{a-1}) & \dots & \tilde{l}_{a-1}(\mathcal{F}\phi_{a-1}) \end{pmatrix} &= \mathbf{A}^{-\top} \begin{pmatrix} l_0(\phi_0) & \dots & l_{a-1}(\phi_0) \\ \vdots & \ddots & \vdots \\ l_0(\phi_{a-1}) & \dots & l_{a-1}(\phi_{a-1}) \end{pmatrix} \mathbf{A}^\top \\
 &= \mathbf{A}^{-\top} \mathbf{I} \mathbf{A}^\top \\
 &= \mathbf{A}^{-\top} \mathbf{A}^\top \\
 &= \mathbf{I}.
 \end{aligned}$$

□

We now present the additional assumptions about reference-mapped finite elements that are commonly made to allow the elements to be mapped from the reference cell to every physical cell in an unstructured mesh. First, the following assumption is often made:

**Assumption 1** Each sub-entity of the same type (e.g. each sub-entity that is a triangle) has an equivalent set of functionals associated with it.

This assumption allows neighbouring cells to be attached by any sub-entity of the same type, as each sub-entity of the given type will have an equivalent set of DOFs. Assumption 1 is true of the vast majority of finite elements, but it is not true of some such as transition elements, where different polynomial degrees are used on different sub-entities, and Fortin–Soulie elements [26], where there are two DOFs associates with two edges of the reference triangle but only one DOF associated with the third edge.

Note that in some finite element libraries—in particular those that use *p*- or *hp*-refinement or hierarchical basis functions—functionals can be associated with sub-entities where no corresponding functional is associated in a neighbouring cell. Where the functionals in question are point evaluations, these are referred to as *hanging nodes*. In such cases, continuity must be enforced in a less trivial manner, and Assumption 1 cannot be made.

A second common assumption is as follows:

**Assumption 2** Let  $E$  be a sub-entity of  $R$ , let  $g : E \rightarrow E$  be an affine bijection, and  $\mathcal{F}_g$  be the push forward corresponding to  $g$ . If  $\phi_a, \dots, \phi_b$  are the basis functions associated with the sub-entity  $E$ , then

$$\text{span} \{ \mathcal{F}_g(\phi_a|_E), \dots, \mathcal{F}_g(\phi_b|_E) \} = \text{span} \{ \phi_a|_E, \dots, \phi_b|_E \},$$

where  $|_E$  denotes the restriction of a function to the sub-entity  $E$ .

This assumption allows neighbouring cells to be attached in any orientation, as any map that corresponds to a rotation or reflection of a sub-entity is affine and a bijection [27, lemma 20.6, exercise 20.1], and so Assumption 2 tells us that the space that needs to be continuous between cells is not affected by rotating or reflecting a sub-entity. Note, however, that the differences in the orientation of neighbouring cells must be taken into account in implementations of finite elements on unordered meshes [28].

In software implementations, it is common to compute basis functions by first computing the dual matrix [18, section 2.2]

$$\mathbf{D} = \begin{pmatrix} l_0(p_0) & \dots & l_{n-1}(p_0) \\ \vdots & \ddots & \vdots \\ l_0(p_{n-1}) & \dots & l_{n-1}(p_{n-1}) \end{pmatrix},$$

where  $\{p_0, \dots, p_{n-1}\}$  is any basis of  $\mathcal{V}$  (often chosen to be an orthogonal basis). Multiplying by the vector of basis functions, we see that

$$\mathbf{D} \begin{pmatrix} \phi_0 \\ \vdots \\ \phi_{n-1} \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{n-1} l_i(p_0)\phi_i \\ \vdots \\ \sum_{i=0}^{n-1} l_i(p_{n-1})\phi_i \end{pmatrix} = \begin{pmatrix} p_0 \\ \vdots \\ p_{n-1} \end{pmatrix},$$

and so the basis functions can be computed via

$$\begin{pmatrix} \phi_0 \\ \vdots \\ \phi_{n-1} \end{pmatrix} = \mathbf{D}^{-1} \begin{pmatrix} p_0 \\ \vdots \\ p_{n-1} \end{pmatrix}. \tag{4}$$

Note that if the functions are not being computed symbolically, the values of  $p_0(\mathbf{x}), \dots, p_{n-1}(\mathbf{x})$  at a given point  $\mathbf{x} \in R$  can be used on the right-hand side of (4) to compute the values of the basis functions at  $\mathbf{x}$ .

Note that if the basis  $\{p_0, \dots, p_{n-1}\}$  was chosen to be the set of monomials and the functionals  $l_0, \dots, l_{n-1}$  were all point evaluations, then  $\mathbf{D}$  would be a Vandermonde matrix with an ill-conditioned inverse [29]. The matrix  $\mathbf{D}$  for a general finite element is sometimes called the generalised Vandermonde matrix [27, chapter 5].

We define conforming and nonconforming finite elements as follows.

**Definition 6** (*conforming and nonconforming*) Let  $V$  be a function space and let  $M_h$  be a mesh of the domain of  $V$ . A finite element  $(R, \mathcal{V}, \mathcal{L})$  is  $V$ -conforming if the basis functions of the element on  $M_h$  are in  $V$ . A finite element  $(R, \mathcal{V}, \mathcal{L})$  is called  $V$ -nonconforming if it is not  $V$ -conforming.

## 2.1 Element types

In this section, we present a brief overview of commonly used types of finite element.

### 2.1.1 Scalar-valued elements

Scalar-valued elements are most commonly used to discretise the  $H^1$  Sobolev space, defined for  $\Omega \subset \mathbb{R}^d$  by

$$H^1(\Omega) := \left\{ f \in L^2(\Omega) \mid \frac{\partial f}{\partial x_0}, \dots, \frac{\partial f}{\partial x_{d-1}} \in L^2(\Omega) \right\},$$

where  $x_0 = x, x_1 = y, x_2 = z$ , and

$$L^2(\Omega) := H^0(\Omega) := \left\{ f : \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} f^2 < \infty \right\}.$$

The scalar-valued Lagrange element is probably the most widely used element and is the first element introduced in most finite element courses. The polynomial space  $\mathcal{V}$  of a Lagrange element on a simplex cell is a complete polynomial space (the polynomial spaces for other cells are later defined in Definition 10), and its dual basis  $\mathcal{L}$  is a set of point evaluations. On quadrilateral and hexahedral cells, serendipity [12] elements are a second widely-used  $H^1$ -conforming finite element: their polynomial spaces  $\mathcal{V}$  are a smaller set of polynomials than a Lagrange element and their dual bases  $\mathcal{L}$  contain point evaluations at vertices and integral moments on other sub-entities. Serendipity elements achieve the same accuracy as Lagrange elements on affine cells, but their accuracy degrades on non-affine cells [30].

By including derivatives in the definitions of functionals, we can enforce continuity on some derivatives of an element and create elements for discretising  $H^2$  and higher-order spaces such as the Argyris [31], Bell [32], Hermite [33], Morley [34], Morley–Wang–Xu [35] and Wu–Xu elements [36]. Note that in some older finite element literature (e.g. [6]), Hermite was used as the name for *all* finite elements with at least one derivative included in their functionals whereas it is now widely used as the name for one specific element.

By associating all the DOFs of an element with the interior of the cell, we can create an element that is discontinuous between cells that can be used to discretise  $L^2$ . The discontinuous Galerkin (DG) finite element method [37, 38] is often based on use of the discontinuous Lagrange element. A number of elements can also be created that are  $H^1$ -nonconforming—they are neither fully discontinuous nor fully  $C^0$  continuous. The most widely used of these is probably the Crouzeix–Raviart [39] element alongside its extensions to higher polynomial degrees and quadrilaterals [10, 26, 40].

$H^1$ -conforming scalar-valued elements typically use the identity push-forward map, also known as the push-forward by the geometric transformation. This is defined, for a scalar-valued function  $\phi$ , by

$$\left(\mathcal{F}^{\text{id}}\phi\right)(\mathbf{x}) := \phi(g^{-1}(\mathbf{x})),$$

where  $g : R \rightarrow \mathbb{R}^{d_s}$  is the geometric map from the reference cell  $R \subset \mathbb{R}^d$  to a physical cell. The term  $g^{-1}(\mathbf{x})$  is the point on the reference cell corresponding to the point  $\mathbf{x}$ , so this mapping maps a value of the function on the reference cell to the same value at the corresponding point.

$L^2$ -conforming elements—such as discontinuous Lagrange elements—use the  $L^2$  Piola push-forward that is defined, for a scalar-valued function  $\phi$ , by

$$\left(\mathcal{F}^{L^2}\phi\right)(\mathbf{x}) := \frac{1}{\det \mathbf{J}} \phi(g^{-1}(\mathbf{x})),$$

where  $\mathbf{J}$  is the Jacobian of the transformation  $g$ . When  $g$  is affine,  $\det \mathbf{J}$  will be constant and the identity push forward can be used (and is often used) instead of the  $L^2$  Piola push forward. However, if  $g$  is not affine, then in some applications it is necessary to use

the  $L^2$  Piola map so that the basis functions maintain properties of the de Rham complex (see Sect. 2.2).

The mapping of elements involving derivatives in their functionals from the reference cell to a physical cell is a more involved process [25], whose detailed discussion is beyond the scope of this paper.

### 2.1.2 Vector-valued elements

Many problems, including those in electromagnetics [41] and fluids [42], involve vectors. For such problems, vector-valued finite elements can be used. Vector-valued elements can be created by using a separate scalar-valued element for each component of the vector: such elements are typically used to represent the geometry of a mesh, with each component of the position of a cell typically represented by a Lagrange element. Using higher-degree elements in the representation of geometry allows curved cells to be represented.

Vector-valued elements built from multiple copies of a scalar-valued element copy the continuity of the scalar element—for most scalar-valued elements, this leads to vector functions whose components are all continuous across boundaries between cells. In some cases—for example, when representing geometry—this  $C^0$  continuity is desired, but in other cases, the discontinuity of some components is more appropriate.

The spaces  $H(\text{div})$  and  $H(\text{curl})$ [27, chapter 4] are defined for  $\Omega \subset \mathbb{R}^d$  by

$$H(\text{div}, \Omega) := \left\{ \mathbf{f} \in \left( L^2(\Omega) \right)^d \mid \text{div}(\mathbf{f}) \in L^2(\Omega) \right\},$$

$$H(\text{curl}, \Omega) := \left\{ \mathbf{f} \in \left( L^2(\Omega) \right)^d \mid \text{curl}(\mathbf{f}) \in \tilde{L} \right\},$$

where

$$\tilde{L} := \begin{cases} \left( L^2(\Omega) \right)^3 & \text{if } d = 3, \\ L^2(\Omega) & \text{if } d = 2, \end{cases}$$

and for  $d = 2$ , curl is defined by

$$\text{curl} \left( \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} \right) = \frac{\partial f_1}{\partial x} - \frac{\partial f_0}{\partial y}.$$

When discretising  $H(\text{div})$ , we only require that components of the vector normal to each facet are continuous between cells. Raviart–Thomas [7] and Brezzi–Douglas–Marini [43] elements are the most widely used  $H(\text{div})$ -conforming elements. To enforce the appropriate continuity, the DOFs associated with each facet are integral moments against vector functions normal to the faces. When discretising  $H(\text{curl})$ , we only require that the components of the vector tangent to edges and faces are continuous. Nédelec first kind [8] and second kind [44] elements are the most widely used  $H(\text{curl})$ -conforming elements. To enforce the appropriate continuity, the DOFs associated with each edge and face are integral moments against vector functions tangential to the sub-entity.

$H(\text{div})$ -conforming and  $H(\text{curl})$ -conforming elements use the contravariant Piola ( $\mathcal{F}^{\text{div}}$ ) and covariant Piola ( $\mathcal{F}^{\text{curl}}$ ) push-forward maps (respectively). These are defined by

$$\begin{aligned} (\mathcal{F}^{\text{div}} \boldsymbol{\phi})(\mathbf{x}) &:= \frac{1}{\det \mathbf{J}} \mathbf{J} \boldsymbol{\phi}(g^{-1}(\mathbf{x})), \\ (\mathcal{F}^{\text{curl}} \boldsymbol{\phi})(\mathbf{x}) &:= \mathbf{J}^{-T} \boldsymbol{\phi}(g^{-1}(\mathbf{x})). \end{aligned}$$

These maps preserve the continuity of normal and tangential components of  $H(\text{div})$ -conforming and  $H(\text{curl})$ -conforming elements (respectively).

### 2.1.3 Matrix-valued elements

Some problems involve matrix-valued unknowns, such as the strain and stress in elasticity [45]. For such problems, matrix-valued finite elements can be defined. As in the vector case, these can be created by using a scalar-valued element for each entry of the matrix. Depending on the application, symmetric or non-symmetric matrices may be desired for the unknown: both cases can easily be created from scalar-valued element by either using a separate element for each entry, or by using fewer elements and setting the appropriate matrix entries to be equal.

As in the vector-valued case, for many applications we desire elements with weaker continuity than the  $C^0$  continuity usually obtained from the scalar element approach. For any matrix-valued function  $\mathbf{F}$ , we define  $\text{div}(\mathbf{F})$  to be the vector containing the results of applying  $\text{div}$  to each row of  $\mathbf{F}$ . If  $d = 2$ , we define  $\text{curl}(\mathbf{F})$  to be the vector containing the results of applying  $\text{curl}$  to each row of  $\mathbf{F}$ ; if  $d = 3$ , we define  $\text{curl}(\mathbf{F})$  to be the matrix where the  $i$ th column is the result of applying  $\text{curl}$  to the  $i$ th row of  $\mathbf{F}$ . We can now define the appropriate Sobolev spaces for matrix-valued elements:  $H(\text{div div})$ ,  $H(\text{curl curl})$  and  $H(\text{curl div})$  [46, 47]. These are defined for  $\Omega \subset \mathbb{R}^d$  by

$$\begin{aligned} H(\text{div div}, \Omega) &:= \left\{ \mathbf{F} \in \left( L^2(\Omega) \right)^{d \times d} \mid \mathbf{F}^T = \mathbf{F} \text{ and } \text{div}(\text{div}(\mathbf{F})) \in \left( H_0^1(\Omega) \right)^* \right\}, \\ H(\text{curl curl}, \Omega) &:= \left\{ \mathbf{F} \in \left( L^2(\Omega) \right)^{d \times d} \mid \mathbf{F}^T = \mathbf{F} \text{ and } \text{curl}(\text{curl}(\mathbf{F})) \in \left( \tilde{H}_2 \right)^* \right\}, \\ H(\text{curl div}, \Omega) &:= \left\{ \mathbf{F} \in \left( L^2(\Omega) \right)^{d \times d} \mid \text{curl}(\text{div}(\mathbf{F})) \in \left( \tilde{H}_1 \right)^* \right\}, \end{aligned}$$

where

$$\begin{aligned} H_0^1(\Omega) &:= \left\{ f \in H^1(\Omega) \mid f = 0 \text{ on } \partial\Omega \right\}, \\ \tilde{H}_1 &:= \begin{cases} \left( H_0^1(\Omega) \right)^3 & \text{if } d = 3, \\ H_0^1(\Omega) & \text{if } d = 2, \end{cases} \\ \tilde{H}_2 &:= \begin{cases} \left( H_0^1(\Omega) \right)^{3 \times 3} & \text{if } d = 3, \\ H_0^1(\Omega) & \text{if } d = 2, \end{cases} \end{aligned}$$

and  $V^*$  denotes the dual of a space  $V$ .

Note that elements in  $H(\text{div div})$  and  $H(\text{curl curl})$  are symmetric matrices, while elements in  $H(\text{curl div})$  are non-symmetric. When discretising  $H(\text{div div})$ , we require that the normal-normal component

$$\mathbf{n}^T \Phi \mathbf{n}$$

is continuous across each facet, where  $\mathbf{n}$  is normal to the facet. Regge elements [45, 48] are the most widely used  $H(\text{div div})$ -conforming elements. When discretising  $H(\text{curl curl})$ , we require that the tangent-tangent component

$$\mathbf{t}^T \Phi \mathbf{t}$$

is continuous across each sub-entity, where  $\mathbf{t}$  is tangential to the sub-entity. Hellan–Herrmann–Johnson elements [49] are the most widely used  $H(\text{curl curl})$ -conforming elements. When discretising  $H(\text{curl div})$ , we require that the tangent-normal component

$$\mathbf{t}^T \Phi \mathbf{n}$$

is continuous across each facet. Sets of  $H(\text{curl div})$ -conforming elements are defined in [50, 51]. Gopalakrishnan–Lederer–Schöberl elements [47][52, section 5.3] can be used to discretise  $H(\text{curl div})$ , although they are slightly nonconforming in  $H(\text{curl div})$ .

$H(\text{div div})$ -conforming,  $H(\text{curl curl})$ -conforming and  $H(\text{curl div})$ -conforming elements use the double contravariant Piola ( $\mathcal{F}^{\text{div div}}$ ), double covariant Piola ( $\mathcal{F}^{\text{curl curl}}$ ) and covariant-contravariant Piola ( $\mathcal{F}^{\text{curl div}}$ ) push-forward maps (respectively). These are defined by

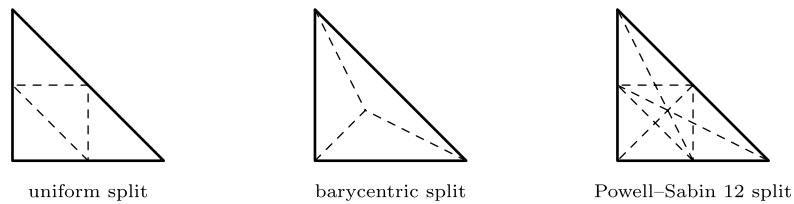
$$\begin{aligned} (\mathcal{F}^{\text{div div}} \Phi)(\mathbf{x}) &:= \frac{1}{(\det \mathbf{J})^2} \mathbf{J} \Phi(g^{-1}(\mathbf{x})) \mathbf{J}^T, \\ (\mathcal{F}^{\text{curl curl}} \Phi)(\mathbf{x}) &:= \mathbf{J}^{-T} \Phi(g^{-1}(\mathbf{x})) \mathbf{J}^{-1}, \\ (\mathcal{F}^{\text{curl div}} \Phi)(\mathbf{x}) &:= \frac{1}{(\det \mathbf{J})} \mathbf{J}^{-T} \Phi(g^{-1}(\mathbf{x})) \mathbf{J}^T. \end{aligned}$$

These maps preserve the appropriate continuity for each element type.

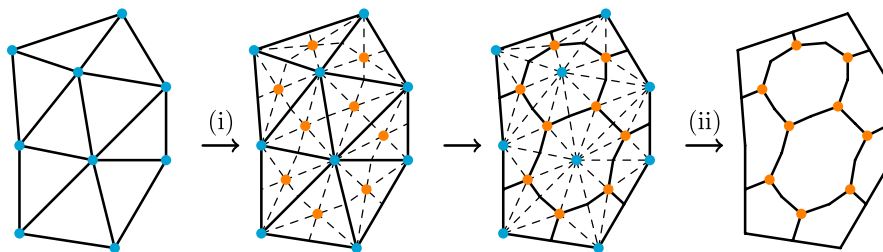
### 2.1.4 Macroelements

The elements listed so far in this section are all defined using polynomials on each cell. There are, however, many desired properties of elements that can be achieved more efficiently by using macroelements, particularly when enforcing higher-order continuity and divergence-free constraints. A macroelement is defined by subdividing the cell  $R$  into a number of sub-cells, and then defining  $\mathcal{V}$  to be polynomial on each subcell with some continuity weaker than  $C^\infty$  between subcells [53]. There are a large number of ways of splitting a cell into sub-cells that used when defining macroelements. A selection of these are shown in Fig. 2.

Popular macroelements include the P1-iso-P2 element [54], which achieves some of the desired properties of a quadratic polynomial element while being linear on each sub-cell; Hsieh–Clough–Tocher elements [56–58], which achieve  $C^1$  continuity using fewer degrees-of-freedom than other  $C^1$  elements; and Guzmán–Neilan elements [9], which are divergence-free at a lower polynomial degree than is otherwise possible.



**Fig. 2** Three commonly used methods for splitting a triangular cell into sub-cells when defining macroelements. The uniform split subdivides the triangle into four congruent sub-triangles and is used by P1-iso-P2 elements [54]; the barycentric split subdivides the triangle into three sub-triangles by adding a vertex at the barycentre and is used by Alfeld–Sorokina elements [55], Guzmán–Neilan elements [9] and Hsieh–Clough–Tocher elements [56–58] among others; the Powell–Sabin 12 split subdivides the triangle into 12 sub-triangles using the barycentre and the midpoints of each side and is used by some Powell–Sabin elements [59]



**Fig. 3** Creating a barycentric dual grid by refining a triangular grid. The barycentric dual grid is used in boundary element applications to define dual polynomial and Buffa–Christiansen elements [60]

### 2.1.5 Dual elements

In boundary element applications, elements defined on the barycentric dual grid are commonly used when constructing operator preconditioners [60]. The barycentric dual grid is constructed from a triangular cell by (i) barycentrically refining each cell (by connecting each vertex with the midpoint of its opposite edge) then (ii) forming all the sub-triangles adjacent to a vertex in the unrefined grid into a single cell. This process is shown for a small example grid in Fig. 3.

Dual elements can be treated as direct finite elements (Definition 2), as they are defined on physical cells rather than being mapped from a reference cell. It is common, however, to define each dual basis function as a linear combination of basis functions on the barycentrically refined grid, where these barycentric basis functions are defined via a reference-mapped element.

In [60], degree 0 and 1 scalar-valued barycentric dual elements were defined alongside lowest degree  $H(\text{div})$ -conforming Buffa–Christiansen elements for triangular cells.

### 2.2 Element families and complexes

When doing finite element computations on meshes that contain a mixture of cell types, it is important to ensure that the elements used have the correct continuity between neighbouring cells of different types. To classify elements where this continuity is achieved, we define compatible elements as follows.

**Definition 7** (*compatible elements*) Two finite elements  $(R, \mathcal{V}, \mathcal{L})$  and  $(\tilde{R}, \tilde{\mathcal{V}}, \tilde{\mathcal{L}})$  are compatible if:

- the topological dimensions of  $R$  and  $\tilde{R}$  are equal;
- for any sub-entities  $E \subset R$  and  $\tilde{E} \subset \tilde{R}$  of the same type that are not equal to the entire reference cell,  $\mathcal{L}|_E$  and  $\tilde{\mathcal{L}}|_{\tilde{E}}$  contain an equivalent set of functionals.

We note that Assumption 1 can be re-expressed as the assumption that an element is compatible with itself. Finite elements are often defined in families, with elements of the same degree within the family being compatible with each other. For example, Lagrange elements of the same degree on different cell types are compatible.

Many PDEs of interest in mathematics and science have rich mathematical structures that are key to their well-posedness [61]. Often, this structure can be encoded in a cochain complex. One such example is the de Rham complex, which describes the relationship between the spaces  $H^1$ ,  $H(\text{curl})$ ,  $H(\text{div})$ , and  $L^2$ , and the differential operators  $\nabla$ ,  $\nabla \times$ , and  $\nabla \cdot$ . In three dimensions, it is given by

$$\mathbb{R} \hookrightarrow H^1 \xrightarrow{\nabla} H(\text{curl}) \xrightarrow{\nabla \times} H(\text{div}) \xrightarrow{\nabla \cdot} L^2 \rightarrow 0.$$

Preserving this structure at the discrete level is often key to obtaining stable and accurate discretisations, and there are several finite element families that replicate it. For example, the following family forms a discrete De Rham complex in three dimensions:

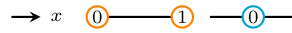
$$\begin{aligned} &\text{Lagrange degree } k \xrightarrow{\nabla} \text{Nédélec (first kind) degree } k \xrightarrow{\nabla \times} \\ &\text{Raviart-Thomas degree } k \xrightarrow{\nabla \cdot} \text{discontinuous Lagrange degree } k - 1 \end{aligned}$$

Many other finite element subcomplexes exist, both for the de Rham complex as well as others. Two naming conventions for these discrete complexes come from exterior calculus [14, 62] and the work of Bernardo Cockburn and Guosheng Fu [22]: in these conventions, the name of the example discrete complex above is  $\mathcal{P}_k^- \Lambda^r$  and  $S_{2,k}^{\text{db}}$  (respectively) for a tetrahedron  $\mathcal{Q}_k^- \Lambda^r$  and  $S_{4,k}^{\text{db}}$  (respectively) for a hexahedron, where  $k$  is the degree of the element and the exterior derivative acts on  $r$ -forms (i.e.  $r = 0$  is Lagrange,  $r = 1$  is the Raviart–Thomas, and so on).

### 2.3 Reference cell sub-entity numbering

In this section, we define the reference cells that we use on DefElement, and the numbering that we assign to their sub-entities. We note that the choice of reference cell and numbering of its sub-entities is arbitrary, and the definitions and results that follow could be adapted to apply to any other choice of reference cell. The reference cells that we use on DefElement are given in the following definition.

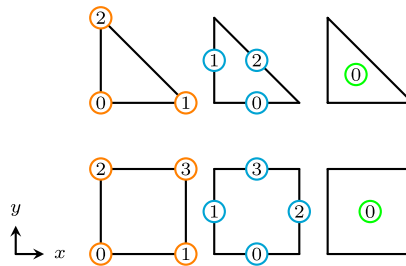
**Definition 8** The reference interval  $(R^-)$ , triangle  $(R^{\triangle})$ , quadrilateral  $(R^{\square})$ , tetrahedron  $(R^{\text{db}})$ , hexahedron  $(R^{\text{db}})$ , prism  $(R^{\text{db}})$ , and pyramid  $(R^{\text{db}})$  are defined by



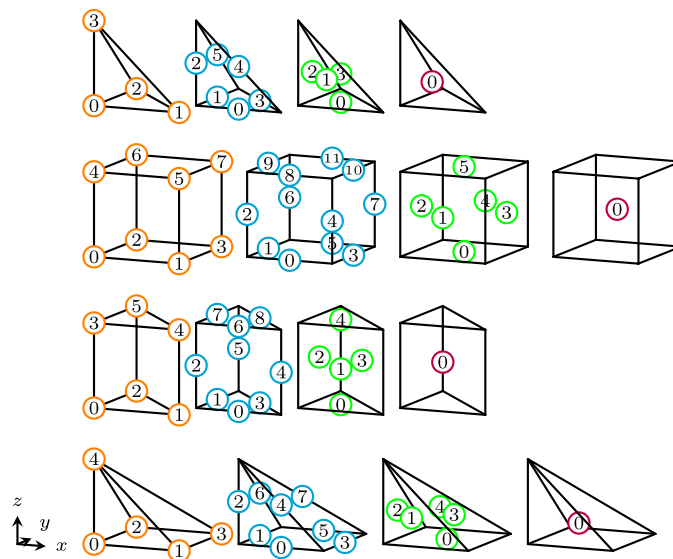
**Fig. 4** The numbering of the sub-entities that DefElement uses for the reference interval

$$\begin{aligned}
 R^- &= \{x \mid 0 \leq x \leq 1\}, \\
 R^\triangle &= \{(x, y) \mid 0 \leq x, 0 \leq y, x + y \leq 1\}, \\
 R^\square &= \{(x, y) \mid 0 \leq x \leq 1, 0 \leq y \leq 1\}, \\
 R^\triangleright &= \{(x, y, z) \mid 0 \leq x, 0 \leq y, 0 \leq z, x + y + z \leq 1\}, \\
 R^{\boxplus} &= \{(x, y, z) \mid 0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1\}, \\
 R^{\boxminus} &= \{(x, y, z) \mid 0 \leq x, 0 \leq y, x + y \leq 1, 0 \leq z \leq 1\}, \\
 R^{\boxtimes} &= \{(x, y, z) \mid 0 \leq x, 0 \leq y, 0 \leq z \leq 1, x + z \leq 1, y + z \leq 1\}.
 \end{aligned}$$

The numbering of their sub-entities used in DefElement is shown in Figs. 4, 5 and 6.



**Fig. 5** The numbering of the sub-entities that DefElement uses for two-dimensional reference cells



**Fig. 6** The numbering of the sub-entities that DefElement uses for three-dimensional reference cells

For  $d \in \mathbb{N}$ , the reference  $k$ -dimensional reference simplex and cube cells are defined by

$$R^{(\text{simplex},d)} = \left\{ (x_0, \dots, x_{d-1}) \mid 0 \leq \min_{i=0}^{d-1} x_i \text{ and } \sum_{i=0}^{d-1} x_i \leq 1 \right\},$$

$$R^{(\text{cube},d)} = \left\{ (x_0, \dots, x_{d-1}) \mid 0 \leq \min_{i=0}^{d-1} x_i \text{ and } \max_{i=0}^{d-1} x_i \leq 1 \right\}.$$

We note that for  $d = 1$  to  $d = 3$ , the simplex reference cells correspond to  $R^-$ ,  $R^\triangle$  and  $R^\triangleleft$ , and the cube cells correspond to  $R^-$ ,  $R^\square$  and  $R^\boxplus$ .

In order to define a consistent convention to use for numbering the sub-entities of reference cells on DefElement, we use the following definition of  $<$  for tuples of values, which corresponds to the behaviour of the  $<$  operator in Python.

**Definition 9** For two tuples  $a$  and  $b$ ,  $<$  is defined by:

- For two length 1 tuples  $a = (a_0)$  and  $b = (b_0)$ ,  $a < b$  if and only if  $a_0 < b_0$
- For two length  $n$  tuples  $a = (a_0, \dots, a_{n-1})$  and  $b = (b_0, \dots, b_{n-1})$ ,  $a < b$  if and only if one of the following conditions holds:
  - $a_0 < b_0$ ,
  - $a_0 = b_0$  and  $(a_1, \dots, a_{n-1}) < (b_1, \dots, b_{n-1})$ .
- For two tuples  $a$  and  $b$  of length  $n_a$  and  $n_b$ ,  $a < b$  if and only if one of the following conditions holds:
  - $(a_0, \dots, a_{\min(n_a, n_b)}) < (b_0, \dots, b_{\min(n_a, n_b)})$ ,
  - $(a_0, \dots, a_{\min(n_a, n_b)}) = (b_0, \dots, b_{\min(n_a, n_b)})$  and  $n_a < n_b$ .

We now present the sub-entity numbering conventions used by DefElement. The aim of these is to ensure consistency in the choices of numbering. These conventions lead the numbering shown in Figs. 4, 5 and 6.

*Convention 1* (Numbering of vertices) Let  $a = (a_0, \dots, a_{d-1}), b = (b_0, \dots, b_{d-1}) \in \mathbb{R}^d$  be two vertices of a reference cell  $R$ . The index of vertex  $a$  is less than the index of vertex  $b$  if and only if  $(a_{d-1}, \dots, a_0) < (b_{d-1}, \dots, b_0)$ .

*Convention 2* (Numbering of sub-entities) Let  $a, b \subseteq R$  be two sub-entities of the same topological dimension. Let  $v_a$  and  $v_b$  be tuples containing the indices of vertices connected to  $a$  and  $b$  (respectively), with vertices with lower indices appearing earlier in the tuples. The index of sub-entity  $a$  is less than the index of sub-entity  $b$  if and only if  $v_a < v_b$ .

### 2.4 The degree of an element

In this section, we define the degree of a finite element. We define two spaces on each of the cells in Definition 8: the natural degree  $k$  polynomial (or on a pyramid rational polynomial, or *rationalomial*) space; and the complete polynomial space.

**Definition 10** The natural polynomial spaces of degree  $k$  on an interval  $(\mathbb{P}_k^-)$ , a triangle  $(\mathbb{P}_k^\Delta)$ , a quadrilateral  $(\mathbb{P}_k^\square)$ , a tetrahedron  $(\mathbb{P}_k^\Delta)$ , a hexahedron  $(\mathbb{P}_k^{\boxplus})$ , and a prism  $(\mathbb{P}_k^\boxminus)$ , and the natural rationalomial space of degree  $k$  on a pyramid  $(\mathbb{P}_k^{\boxtimes})$  are defined by

$$\begin{aligned} \mathbb{P}_k^- &= \text{span} \{x^{p_0} \mid p_0 \in \mathbb{N}, p_0 \leq k\}, \\ \mathbb{P}_k^\Delta &= \text{span} \{x^{p_0}y^{p_1} \mid p_0, p_1 \in \mathbb{N}_0, p_0 + p_1 \leq k\}, \\ \mathbb{P}_k^\square &= \text{span} \{x^{p_0}y^{p_1} \mid p_0, p_1 \in \mathbb{N}_0, p_0 \leq k, p_1 \leq k\}, \\ \mathbb{P}_k^\Delta &= \text{span} \{x^{p_0}y^{p_1}z^{p_2} \mid p_0, p_1, p_2 \in \mathbb{N}_0, p_0 + p_1 + p_2 \leq k\}, \\ \mathbb{P}_k^{\boxplus} &= \text{span} \{x^{p_0}y^{p_1}z^{p_2} \mid p_0, p_1, p_2 \in \mathbb{N}_0, p_0 \leq k, p_1 \leq k, p_2 \leq k\}, \\ \mathbb{P}_k^\boxminus &= \text{span} \{x^{p_0}y^{p_1}z^{p_2} \mid p_0, p_1, p_2 \in \mathbb{N}_0, p_0 + p_1 \leq k, p_2 \leq k\}, \\ \mathbb{P}_k^{\boxtimes} &= \text{span} \left\{ \frac{x^{p_0}y^{p_1}z^{p_2}}{(1-z)^{p_0+p_1}} \mid p_0, p_1, p_2 \in \mathbb{N}_0, p_0 \leq k, p_1 \leq k, p_2 \leq k \right\}, \end{aligned}$$

where  $\mathbb{N}_0$  is the set of non-negative integers.

For  $d \in \mathbb{N}$ , the natural polynomial spaces of degree  $k$  on a simplex  $(\mathbb{P}_k^{(\text{simplex},d)})$  and a cube  $(\mathbb{P}_k^{(\text{cube},d)})$  are defined by

$$\begin{aligned} \mathbb{P}_k^{(\text{simplex},d)} &= \text{span} \left\{ \prod_{i=0}^{d-1} x_i^{p_i} \mid p_0, p_1, \dots, p_{d-1} \in \mathbb{N}_0 \text{ and } \sum_{i=0}^{d-1} p_i \leq k \right\}, \\ \mathbb{P}_k^{(\text{cube},d)} &= \text{span} \left\{ \prod_{i=0}^{d-1} x_i^{p_i} \mid p_0, p_1, \dots, p_{d-1} \in \mathbb{N}_0 \text{ and } \max_{i=0}^{d-1} p_i \leq k \right\}, \end{aligned}$$

**Definition 11** For a cell with topological dimension  $d$ , the complete polynomial space degree  $k$  is defined by  $\mathbb{P}_k = \mathbb{P}_k^{(\text{simplex},d)}$ . For  $d = 1$  to  $d = 3$ , we note that

$$\mathbb{P}_k = \begin{cases} \mathbb{P}_k^- & \text{if the topological dimension of the cell is 1,} \\ \mathbb{P}_k^\Delta & \text{if the topological dimension of the cell is 2,} \\ \mathbb{P}_k^{\boxplus} & \text{if the topological dimension of the cell is 3.} \end{cases}$$

On every cell except the pyramid, the natural polynomials spaces are the spaces spanned by the degree  $k$  Lagrange finite element on the cell. On the pyramid, however, the Lagrange element spans the smaller space [22, 63]

$$\mathbb{P}_k^{\triangleright, \text{Lagrange}} = \text{span} \left\{ \frac{x^{p_0} y^{p_1} z^{p_2}}{(1-z)^{\min(p_0, p_1)}} \mid p_0, p_1, p_2 \in \mathbb{N}_0, p_0 + p_2 \leq k, p_1 + p_2 \leq k \right\}.$$

This close relationship with the Lagrange element inspires the naming convention we use for our notions of the degree of an element: we now define the polynomial and Lagrange subdegrees and superdegrees of a finite element.

**Definition 12** (*polynomial subdegree*) The polynomial subdegree of a finite element  $(R, \mathcal{V}, \mathcal{L})$  is the largest value of  $k$  such that

$$\mathbb{P}_k \subseteq \mathcal{V}.$$

If  $\mathbb{P}_0 \not\subseteq \mathcal{V}$ , then the polynomial subdegree is undefined.

**Definition 13** (*polynomial superdegree*) The polynomial superdegree of a finite element  $(R, \mathcal{V}, \mathcal{L})$  is the smallest value of  $k$  such that

$$\mathcal{V} \subseteq \mathbb{P}_k.$$

If  $\forall k \in \mathbb{N}_0, \mathcal{V} \not\subseteq \mathbb{P}_k$ , then the polynomial superdegree is undefined.

**Definition 14** (*Lagrange subdegree*) The Lagrange subdegree of a finite element  $(R, \mathcal{V}, \mathcal{L})$  is the largest value of  $k$  such that

$$\mathbb{P}_k^R \subseteq \mathcal{V}.$$

If  $\mathbb{P}_0^R \not\subseteq \mathcal{V}$ , then the Lagrange subdegree is undefined.

**Definition 15** (*Lagrange superdegree*) The Lagrange superdegree of a finite element  $(R, \mathcal{V}, \mathcal{L})$  is the smallest value of  $k$  such that

$$\mathcal{V} \subseteq \mathbb{P}_k^R.$$

If  $\forall k \in \mathbb{N}_0, \mathcal{V} \not\subseteq \mathbb{P}_k^R$ , then the Lagrange superdegree is undefined.

We note that on simplex cells the Lagrange and polynomial degrees will always be equal, but they will often disagree on other cell types.

For many elements, the polynomial superdegree and Lagrange subdegree are not well-behaved: for example, for serendipity elements [12] on a quadrilateral cell, the Lagrange subdegree is always 1; and for Arnold–Boffi–Falk elements [24] on a quadrilateral cell, the polynomial superdegree is  $2k + 2$ , where  $k$  is the polynomial subdegree, and so jumps by 2 every time the element’s degree is increased. It is therefore typical to use either the polynomial subdegree or the Lagrange superdegree as the canonical degree of an element. In general on DefElement, we index elements by their polynomial subdegree, unless this is undefined in which case a sensible alternative is chosen (for example, bubble elements have an undefined polynomial subdegree, so are indexed using the Lagrange superdegree instead). Note that for some elements (including Nédélec

first kind and Raviart–Thomas), this indexing is different to that used by some libraries, including FEniCS [16, 64].

Let  $(R, \mathcal{V}, \mathcal{L})$  be a finite element and let  $\{p_0, \dots, p_{n-1}\}$  be a basis of  $\mathcal{V}$ . If the finite element has Lagrange superdegree  $k$ , then the basis can be represented by

$$\begin{pmatrix} p_0 \\ \vdots \\ p_{n-1} \end{pmatrix} = \mathbf{C} \begin{pmatrix} q_0 \\ \vdots \\ q_{m-1} \end{pmatrix} \tag{5}$$

for some  $\mathbf{C} = \mathbb{R}^{n \times m}$ , where  $\{q_0, \dots, q_{m-1}\}$  is a basis of  $\mathbb{P}_k^R$ . Using (4), we can now represent the finite element basis functions as

$$\begin{pmatrix} \phi_0 \\ \vdots \\ \phi_{n-1} \end{pmatrix} = \mathbf{D}^{-1} \mathbf{C} \begin{pmatrix} q_0 \\ \vdots \\ q_{m-1} \end{pmatrix}. \tag{6}$$

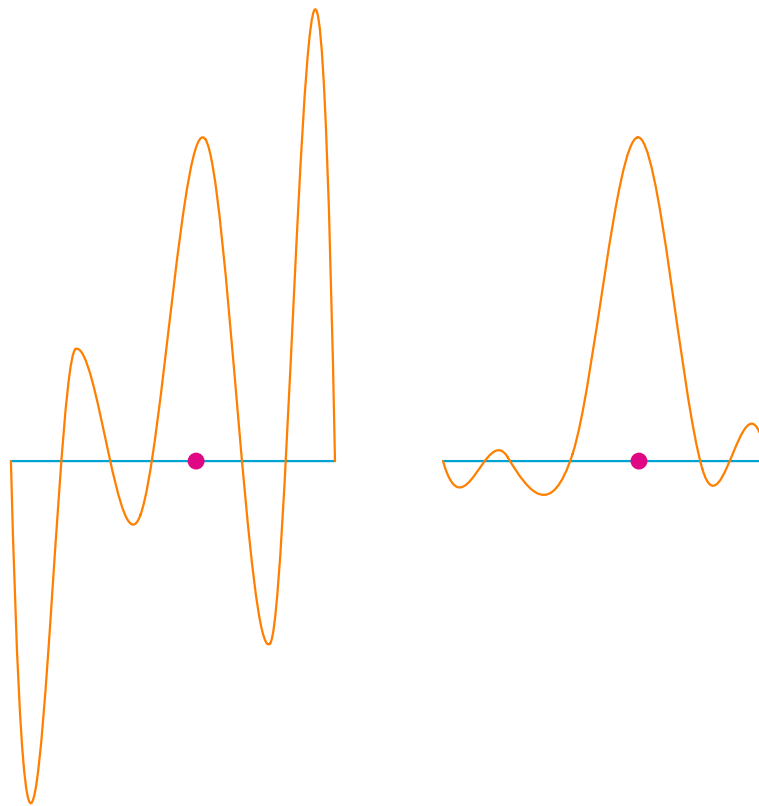
Many implementations store the basis functions by computing and storing the matrix of expansion coefficients  $\mathbf{D}^{-1} \mathbf{C} \in \mathbb{R}^{n \times m}$  [16, 18, 64–66]. The basis functions can then be evaluated at a set of points by evaluating the functions  $q_0, \dots, q_{m-1}$  at these points, and then recombining these with the stored matrix of expansion coefficients. As the functions  $q_0, \dots, q_{m-1}$  are a basis of the general space  $\mathbb{P}_k^R$ , the code to evaluate them is not specific for each finite element, and so the same implementation of the spaces can be used for all elements. It is common in implementations to take  $\{q_0, \dots, q_{m-1}\}$  to be a basis for which recurrence formulae exist that can be used to quickly evaluate  $\{q_0, \dots, q_{m-1}\}$  at a set of points [67–69]: for example, the Legendre polynomials (mapped to  $[0, 1]$  if the reference cells in definition 8 are used) could be used for interval cells. In order to ensure that the basis functions can be stored in this way, many finite element implementations implicitly rely on every element having a well-defined Lagrange superdegree.

The definitions of degree can be extended to macroelements by defining spaces on the macroelement that are equal to those defined in definition 10 on each sub-cell, with appropriate continuity between sub-cells.

### 3 Element variants

For many elements, it is possible to define a number of different variants of the element. In many cases, there is no single variant that is always the best variant, and so implementations may support multiple variants. For example, for Lagrange elements, it is common to use equally spaced points to define low-degree elements, and mesh storage formats that use Lagrange elements to represent mesh geometry often assume this equal spacing. However, if equally spaced points are used for high-degree elements, the basis functions will exhibit Runge’s phenomenon and become numerically unstable [70], as shown in Fig. 7. Hence, for higher-degree elements, it is best to use an alternative set of points, such as Gauss–Lobatto–Legendre (GLL) points or Chebyshev points [71, chapter 13].

In general, variants of finite elements are defined as follows.



**Fig. 7** A basis function of a degree 7 Lagrange element on an interval when defined using equispaced points (left) and GLL points (right). The basis function defined using equispaced points exhibits Runge’s phenomenon: there are large function values far from the marked point where the basis function is equal to one. Similar issues can be observed when defining the basis on other cell types

**Definition 16** (Variant of an element) Let  $(R, \mathcal{V}, \mathcal{L})$  and  $(\tilde{R}, \tilde{\mathcal{V}}, \tilde{\mathcal{L}})$  be reference-mapped finite elements.  $(\tilde{R}, \tilde{\mathcal{V}}, \tilde{\mathcal{L}})$  is a variant of  $(R, \mathcal{V}, \mathcal{L})$  if:

- $\tilde{R} = R$ ;
- $\tilde{\mathcal{V}} = \mathcal{V}$ ;
- For all sub-entities  $E \subseteq R$ ,  $\mathcal{L}|_E$  and  $\tilde{\mathcal{L}}|_E$  are equivalent sets of functionals, where  $\mathcal{L}|_E$  and  $\tilde{\mathcal{L}}|_E$  are the sets of functionals in  $\mathcal{L}$  and  $\tilde{\mathcal{L}}$  that are associated with  $E$ .

We will now show that a set of alternative criteria is equivalent to Definition 16, but we must first define the controlled and uncontrolled trace of a finite element’s polynomial space on a sub-entity as follows.

**Definition 17** Let  $(R, \mathcal{V}, \mathcal{L})$  be a reference-mapped finite element. Let  $E \subseteq R$  be a sub-entity of  $R$  (i.e. a vertex, edge, or face of  $R$ , or  $R$  itself). Let  $\mathcal{L} = \{l_0, \dots, l_{n-1}\}$  and let  $\phi_0, \dots, \phi_{n-1} \in \mathcal{V}$  be the basis functions of the finite element. The uncontrolled trace space  $\gamma^\perp(\mathcal{V}, E, \mathcal{L})$  is defined by

$$\gamma^\perp(\mathcal{V}, E, \mathcal{L}) = \{p|_E : p \in \mathcal{V} \text{ and } \forall l_i \in \mathcal{L}|_E l_i(p) = 0\},$$

where  $\mathcal{L}|_{\bar{E}}$  is the set of all functionals in  $\mathcal{L}$  associated with the closure of  $E$ . The controlled trace space  $\gamma^{\parallel}(\mathcal{V}, E, \mathcal{L})$  is defined by

$$\begin{aligned} \gamma^{\parallel}(\mathcal{V}, E, \mathcal{L}) &= \left[ \gamma^{\perp}(\mathcal{V}, E, \mathcal{L}) \right]^{\perp} \\ &= \left\{ p|_E : p \in \mathcal{V} \text{ and } \forall q \in \gamma^{\perp}(\mathcal{V}, E, \mathcal{L}) \langle p, q \rangle_E = 0 \right\}, \end{aligned}$$

where  $\langle p, q \rangle_E := \int_E p \cdot q$ .

Note that it is a standard result in functional analysis (see e.g. [72, Theorem 3.3–4]) that if  $V$  is a function space and  $W$  is a subspace of  $V$ , then  $V = W \oplus W^{\perp}$ . It follows that

$$\mathcal{V}|_E = \gamma^{\perp}(\mathcal{V}, E, \mathcal{L}) \oplus \gamma^{\parallel}(\mathcal{V}, E, \mathcal{L}). \tag{7}$$

As an example, let  $(R, \mathcal{V}, \mathcal{L})$  be a degree  $k$  Lagrange element. For any sub-entity  $E \subset R$ , the restricted space  $\mathcal{V}|_E$  is the Lagrange space  $\mathbb{P}_k^E$ . The uncontrolled trace space  $\gamma^{\perp}(\mathcal{V}, E, \mathcal{L})$  is equal to  $\{0\}$ , as any non-zero polynomial on  $E$  would make at least one functional associated with  $\bar{E}$  non-zero. It follows from (7) that the controlled trace space  $\gamma^{\parallel}(\mathcal{V}, E, \mathcal{L})$  is  $\mathbb{P}_k^E$ .

For a less trivial example, let  $\triangle, \mathcal{V}, \mathcal{L}$  be a degree 1 Raviart–Thomas element [7] on a triangle and let  $E$  be the edge with vertices at  $(0, 0)$  and  $(1, 0)$ . The polynomial space for this element is

$$\mathcal{V} = \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} y \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix}, \begin{pmatrix} x^2 \\ xy \end{pmatrix}, \begin{pmatrix} xy \\ y^2 \end{pmatrix} \right\},$$

and so

$$\mathcal{V}|_E = \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} x^2 \\ 0 \end{pmatrix} \right\}.$$

There are no functionals associated with vertices in this element, and so the only functionals associated with  $\bar{E}$  are the integral moments of the normal components that are associated with  $E$ . These functionals will all be zero if applied to functions that only have a tangential component and so

$$\begin{aligned} \gamma^{\perp}(\mathcal{V}, E, \mathcal{L}) &= \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} x^2 \\ 0 \end{pmatrix} \right\}, \\ \gamma^{\parallel}(\mathcal{V}, E, \mathcal{L}) &= \text{span} \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix} \right\}. \end{aligned}$$

For this element, the normal components are continuous between cells: the controlled space  $\gamma^{\parallel}(\mathcal{V}, E, \mathcal{L})$  is the space of functions that are continuous between cells, while the uncontrolled space contains the functions that may jump between cells.

For the remainder of this section, we will aim to prove that if the uncontrolled trace spaces of two elements are equal for two elements with the same reference cell and polynomial space, then they are variants of each other (Theorem 1). We begin by proving some preliminary lemmas that we will use.

**Lemma 2** *Let  $(R, \mathcal{V}, \mathcal{L})$  be a reference-mapped finite element. If  $E$  is a sub-entity of  $R$ , then*

$$\gamma^\perp(\mathcal{V}, E, \mathcal{L}) = \text{span} \{ \phi_i|_E : l_i \notin \mathcal{L}|_{\bar{E}} \}.$$

**Proof** Let  $p \in \gamma^\perp(\mathcal{V}, E, \mathcal{L})$ . As  $p \in \mathcal{V}|_E := \text{span} \{ \phi_0|_E, \dots, \phi_{n-1}|_E \}$ , we can write

$$p = \sum_{j=0}^{n-1} \alpha_j \phi_j|_E,$$

for some  $\alpha_0, \dots, \alpha_{n-1} \in \mathbb{R}$ . If  $l_i \in \mathcal{L}|_{\bar{E}}$ , then

$$0 = l_i(p) = \sum_{j=0}^{n-1} \alpha_j l_i(\phi_j|_E) = \alpha_i.$$

This means that  $p \in \text{span} \{ \phi_i|_E : l_i \notin \mathcal{L}|_{\bar{E}} \}$ , and so

$$\gamma^\perp(\mathcal{V}, E, \mathcal{L}) \subseteq \text{span} \{ \phi_i|_E : l_i \notin \mathcal{L}|_{\bar{E}} \}.$$

From the definition of the uncontrolled trace (Definition 17), it can be seen that if  $l_i \notin \mathcal{L}|_{\bar{E}}$  then  $\phi_i|_E \in \gamma^\perp(\mathcal{V}, E, \mathcal{L})$ , and so

$$\gamma^\perp(\mathcal{V}, E, \mathcal{L}) \supseteq \text{span} \{ \phi_i|_E : l_i \notin \mathcal{L}|_{\bar{E}} \}.$$

□

**Lemma 3** *Let  $\mathcal{V}|_E := \text{span} \{ \phi_0|_E, \dots, \phi_{n-1}|_E \}$ . Given  $f \in \mathcal{V}|_E$ , there are unique functions  $f^\parallel \in \gamma^\parallel(\mathcal{V}, E, \mathcal{L})$  and  $f^\perp \in \gamma^\perp(\mathcal{V}, E, \mathcal{L})$  such that*

$$f = f^\perp + f^\parallel.$$

Moreover, the operator  $P_E : \mathcal{V}|_E \rightarrow \gamma^\parallel(\mathcal{V}, E, \mathcal{L})$  defined by  $P_E : f \mapsto f^\parallel$  is a linear projection.

**Proof** See [72, theorem 3.3–4], and note that  $P_E$  is an orthogonal projection operator, as defined in [72, section 3.3]. □

**Lemma 4** *Let  $(R, \mathcal{V}, \mathcal{L})$  be a reference-mapped finite element. If  $E$  is a sub-entity of  $R$ , then*

$$\gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \text{span} \{ P_E(\phi_i|_E) : l_i \in \mathcal{L}|_{\bar{E}} \},$$

where  $P_E$  is defined as in Lemma 3.

**Proof** Using (7), we see that

$$\begin{aligned} \gamma^\perp(\mathcal{V}, E, \mathcal{L}) \oplus \gamma^\parallel(\mathcal{V}, E, \mathcal{L}) &= \mathcal{V}|_E \\ &= \text{span} \{ \phi_0|_E, \dots, \phi_{n-1}|_E \} \\ &= \text{span} \left( \{ \phi_i|_E : l_i \notin \mathcal{L}|_{\bar{E}} \} \cup \{ \phi_i|_E : l_i \in \mathcal{L}|_{\bar{E}} \} \right). \end{aligned}$$

Using Lemma 2, we therefore see that

$$\gamma^\perp(\mathcal{V}, E, \mathcal{L}) \oplus \gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \gamma^\perp(\mathcal{V}, E, \mathcal{L}) + \text{span} \{ \phi_{i|E} : l_i \in \mathcal{L}|_{\bar{E}} \}.$$

By Lemma 3, we know that for all  $l_i \in \mathcal{L}|_{\bar{E}}$ , there exists  $g_i \in \gamma^\perp(\mathcal{V}, E, \mathcal{L})$  such that  $\phi_{i|E} = g_i + P_E(\phi_{i|E})$ . It follows that

$$\gamma^\perp(\mathcal{V}, E, \mathcal{L}) \oplus \gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \gamma^\perp(\mathcal{V}, E, \mathcal{L}) + \text{span} \{ P_E(\phi_{i|E}) : l_i \in \mathcal{L}|_{\bar{E}} \}. \tag{8}$$

As (for all  $l_i \in \mathcal{L}|_{\bar{E}}$ )  $P_E(\phi_{i|E}) \in \gamma^\parallel(\mathcal{V}, E, \mathcal{L})$ , we see due to Lemma 3 that

$$\gamma^\perp(\mathcal{V}, E, \mathcal{L}) \cap \text{span} \{ P_E(\phi_{i|E}) : l_i \in \mathcal{L}|_{\bar{E}} \} = \emptyset,$$

and so the sum in (8) is disjoint, ie

$$\gamma^\perp(\mathcal{V}, E, \mathcal{L}) \oplus \gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \gamma^\perp(\mathcal{V}, E, \mathcal{L}) \oplus \text{span} \{ P_E(\phi_{i|E}) : l_i \in \mathcal{L}|_{\bar{E}} \}.$$

From this, we conclude that

$$\gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \text{span} \{ P_E(\phi_{i|E}) : l_i \in \mathcal{L}|_{\bar{E}} \}.$$

□

**Lemma 5** For all  $l_i, l_j \in \mathcal{L}|_{\bar{E}}$ ,

$$l_i(P_E(\phi_{j|E})) = \delta_{ij}.$$

**Proof** By lemma 3, we know that for all  $l_i \in \mathcal{L}|_{\bar{E}}$ , there exists  $g_i \in \gamma^\perp(\mathcal{V}, E, \mathcal{L})$  such that  $\phi_{i|E} = g_i + P_E(\phi_{i|E})$ . By definition 17,  $l_j(g_i) = 0$  and so

$$l_j(P_E(\phi_{i|E})) = l_j(\phi_{i|E} - g_i) = l_j(\phi_{i|E}) - l_j(g_i) = l_j(\phi_{i|E}).$$

The result then follows from Definition 3. □

**Lemma 6** Let  $(R, \mathcal{V}, \mathcal{L})$  be a reference-mapped finite element. If  $E$  be a sub-entity of  $R$ , then

$$\{ P_E(\phi_{i|E}) : l_i \in \mathcal{L}|_{\bar{E}} \}$$

is a basis of  $\gamma^\parallel(\mathcal{V}, E, \mathcal{L})$ , where  $P_E$  is defined as in Lemma 3.

**Proof** By Lemma 4, we know that

$$\gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \text{span} \{ P_E(\phi_{i|E}) : l_i \in \mathcal{L}|_{\bar{E}} \},$$

and so there exists a basis of  $\gamma^\parallel(\mathcal{V}, E, \mathcal{L})$  that is a subset of  $\{ P_E(\phi_{i|E}) : l_i \in \mathcal{L}|_{\bar{E}} \}$ . Suppose that

$$\sum_{\substack{i \text{ s.t.} \\ l_i \in \mathcal{L}|_{\bar{E}}}} \alpha_i P_E(\phi_{i|E}) = 0,$$

and let  $l_j \in \mathcal{L}|_{\bar{E}}$ . We can see, using Lemma 5, that

$$0 = l_j(0) = l_j \left( \sum_{\substack{i \text{ s.t.} \\ l_i \in \mathcal{L}|_{\bar{E}}}} \alpha_i P_E(\phi_i|_E) \right) = \sum_{\substack{i \text{ s.t.} \\ l_i \in \mathcal{L}|_{\bar{E}}}} \alpha_i l_j(P_E(\phi_i|_E)) = \alpha_j,$$

and so the set  $\{P_E(\phi_i|_E) : l_i \in \mathcal{L}|_{\bar{E}}\}$  is linearly independent. □

**Lemma 7** *Let  $(R, \mathcal{V}, \mathcal{L})$  and  $(R, \mathcal{V}, \tilde{\mathcal{L}})$  be reference-mapped finite elements where for all sub-entities  $E \subseteq R$ , there are the same number of functionals in  $\mathcal{L}$  and  $\tilde{\mathcal{L}}$  that are associated with  $E$ . If for all sub-entities  $E \subseteq R$ ,  $\gamma^\perp(\mathcal{V}, E, \mathcal{L}) = \gamma^\perp(\mathcal{V}, E, \tilde{\mathcal{L}})$ ; then for all sub-entities  $E \subseteq R$ ,  $\gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \gamma^\parallel(\mathcal{V}, E, \tilde{\mathcal{L}})$ ,*

**Proof** This is follows from [72, theorem 3.3–4]. □

**Lemma 8** *Let  $(R, \mathcal{V}, \mathcal{L})$  and  $(R, \mathcal{V}, \tilde{\mathcal{L}})$  be reference-mapped finite elements where for all sub-entities  $E \subseteq R$ , there are the same number of functionals in  $\mathcal{L}$  and  $\tilde{\mathcal{L}}$  that are associated with  $E$ . Let  $E \subseteq R$  be a sub-entity of  $R$ . If  $\gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \gamma^\parallel(\mathcal{V}, E, \tilde{\mathcal{L}})$ , then  $\mathcal{L}|_{\bar{E}}$  and  $\tilde{\mathcal{L}}|_{\bar{E}}$  are equivalent sets of functionals.*

**Proof** Let  $E \subseteq R$  by a sub-entity of  $R$ , and suppose that  $\gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \gamma^\parallel(\mathcal{V}, E, \tilde{\mathcal{L}})$ . Let  $\mathcal{I}_E$  and  $\tilde{\mathcal{I}}_E$  be sets of indices such that  $\{l_i : i \in \mathcal{I}_E\} = \mathcal{L}|_{\bar{E}}$  and  $\{\tilde{l}_i : i \in \tilde{\mathcal{I}}_E\} = \tilde{\mathcal{L}}|_{\bar{E}}$ .

By Lemma 6, we know there exist  $\{p_i : i \in \mathcal{I}_E\} \subset \gamma^\parallel(\mathcal{V}, E, \mathcal{L})$  and  $\{q_i : i \in \tilde{\mathcal{I}}_E\} \subset \gamma^\parallel(\mathcal{V}, E, \tilde{\mathcal{L}})$  such that  $l_i(p_j) = \delta_{ij}$  and (for  $i, j \in \tilde{\mathcal{I}}_E$ )  $\tilde{l}_i(q_j) = \delta_{ij}$ . As  $\gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \gamma^\parallel(\mathcal{V}, E, \tilde{\mathcal{L}})$ , we see from Lemma 6, that  $\{p_i : i \in \mathcal{I}_E\}$  and  $\{q_i : i \in \tilde{\mathcal{I}}_E\}$  are bases of the space  $\gamma^\parallel(\mathcal{V}, E, \mathcal{L})$ . Hence there exist  $\alpha_{ij} \in \mathbb{R}$  such that (for  $i \in \mathcal{I}_E$ )

$$q_i = \sum_{j \in \tilde{\mathcal{I}}_E} \alpha_{ij} p_j.$$

Using this, we see that (for  $j \in \mathcal{I}_E$ )

$$l_j(q_i) = l_j \left( \sum_{k=a}^b \alpha_{ik} p_k \right) = \sum_{k=a}^b \alpha_{ik} l_j(p_k) = \alpha_{ij}. \tag{9}$$

Let  $f \in \gamma^\parallel(\mathcal{V}, E, \mathcal{L})$ . As  $\{q_i : i \in \tilde{\mathcal{I}}_E\}$  is a basis, there exist  $c_j \in \mathbb{R}$  such that  $f = \sum_{j \in \tilde{\mathcal{I}}_E} c_j q_j$ . We see that (for  $i \in \tilde{\mathcal{I}}_E$ )

$$\tilde{l}_i(f) = \tilde{l}_i \left( \sum_{j \in \tilde{\mathcal{I}}_E} c_j q_j \right) = \sum_{j \in \tilde{\mathcal{I}}_E} c_j \tilde{l}_i(q_j) = c_i. \tag{10}$$

Using (9) and (10), we see that

$$l_j(f) = \sum_{i=a}^b c_i l_j(q_i) = \sum_{i=a}^b c_i \alpha_{ij} = \sum_{i=a}^b \alpha_{ij} l_i(f),$$

and so

$$l_j = \sum_{i=a}^b \alpha_{ij} \tilde{l}_i.$$

It follows that  $\mathcal{L}|_{\bar{E}}$  and  $\tilde{\mathcal{L}}|_{\bar{E}}$  are equivalent. □

**Lemma 9** *Let  $(R, \mathcal{V}, \mathcal{L})$  and  $(R, \mathcal{V}, \tilde{\mathcal{L}})$  be reference-mapped finite elements where for all sub-entities  $E \subseteq R$ , there are the same number of functionals in  $\mathcal{L}$  and  $\tilde{\mathcal{L}}$  that are associated with  $E$ . If for all sub-entities  $E \subseteq R$ ,  $\mathcal{L}|_{\bar{E}}$  and  $\tilde{\mathcal{L}}|_{\bar{E}}$  are equivalent sets of functionals; then for all sub-entities  $E \subseteq R$ ,  $\mathcal{L}|_E$  and  $\tilde{\mathcal{L}}|_E$  are equivalent sets of functionals.*

**Proof** We can show this by induction on the dimension of the sub-entity. If the sub-entity is of dimension 0 (i.e. a vertex), then  $\bar{E} = E$  and the result trivially holds.

If the result holds for sub-entities of dimension  $< k$ , then let  $E$  be an entity of dimension  $k$ . We can write  $\bar{E} = E \cup E_0 \cup \dots \cup E_{m-1}$ , where  $E_0, \dots, E_{m-1}$  are sub-entities of dimension  $< k$ . We can see that

$$\begin{aligned} \text{span } \mathcal{L}|_{\bar{E}} &= \text{span } \mathcal{L}|_E \oplus \text{span } \mathcal{L}|_{E_0} \oplus \dots \oplus \text{span } \mathcal{L}|_{E_{m-1}} \\ &= \text{span } \mathcal{L}|_E \oplus \text{span } \tilde{\mathcal{L}}|_{E_0} \oplus \dots \oplus \text{span } \tilde{\mathcal{L}}|_{E_{m-1}}, \end{aligned}$$

and also

$$\begin{aligned} \text{span } \tilde{\mathcal{L}}|_{\bar{E}} &= \text{span } \tilde{\mathcal{L}}|_E \\ &= \text{span } \tilde{\mathcal{L}}|_E \oplus \text{span } \tilde{\mathcal{L}}|_{E_0} \oplus \dots \oplus \text{span } \tilde{\mathcal{L}}|_{E_{m-1}}. \end{aligned}$$

Therefore  $\text{span } \tilde{\mathcal{L}}|_E = \text{span } \mathcal{L}|_E$ . By Lemma 1, it follows that  $\mathcal{L}|_E$  and  $\tilde{\mathcal{L}}|_E$  are equivalent sets of functionals. □

We can now show the main result of this section.

**Theorem 1** *Let  $(R, \mathcal{V}, \mathcal{L})$  and  $(R, \mathcal{V}, \tilde{\mathcal{L}})$  be finite elements. If for all sub-entities  $E \subseteq R$ ,  $\gamma^\perp(\mathcal{V}, E, \mathcal{L}) = \gamma^\perp(\mathcal{V}, E, \tilde{\mathcal{L}})$ ; then  $(\tilde{R}, \tilde{\mathcal{V}}, \tilde{\mathcal{L}})$  is a variant of  $(R, \mathcal{V}, \mathcal{L})$ .*

**Proof** The first two bullet points in Definition 16 trivially hold, so we only need to show the third bullet point.

First, Lemma 7 tells us that the assumptions of the theorem imply that for all sub-entities  $E \subseteq R$ ,  $\gamma^\parallel(\mathcal{V}, E, \mathcal{L}) = \gamma^\parallel(\mathcal{V}, E, \tilde{\mathcal{L}})$ . Applying Lemma 8 to every sub-entity  $E \subseteq R$ , we see that  $\mathcal{L}|_{\bar{E}}$  and  $\tilde{\mathcal{L}}|_{\bar{E}}$  are equivalent sets of functionals. We can then apply Lemma 9 to see that  $\mathcal{L}|_E$  and  $\tilde{\mathcal{L}}|_E$  are equivalent sets of functionals. □

Theorem 1 gives us a way to show that two elements are variants using only their basis functions. We will use it in the following section as the basis of our algorithm to verify that elements from different implementations are variants of each other.

#### 4 Element verification

There are a large number of open source libraries available that include implementations of finite elements. Without a significant amount of work by the user, it is difficult to be certain that the implementations of the same element in different libraries are indeed the implementations of the same element, and that the way the element is implemented agrees with the element's definition in the literature. For less simple elements in particular, it is quite possible that a bug in the implementation leads to an element that “looks ok” but does not solve problems to the accuracy that it should.

In order to confirm that implementations of the same element in different libraries are actually the same element, DefElement performs verification of finite elements. This verification is performed monthly, with the latest results displayed at [defelement.org/verification/](https://defelement.org/verification/). It is currently performed for elements implemented in Basix [16], FIAT [18], and ndelement [19], with Symfem [20] used as the verification baseline. DefElement's element verification has led the authors of this paper to find and correct a number of bugs in FIAT and Symfem.

When verifying elements, we do not require the basis functions in the two implementations to be exactly the same, as two libraries may use different variants of the same element, as defined in Definition 16. These variants will have different basis functions, but their important properties that the error of a problem solution depends on will be the same. We therefore designed an algorithm to verify that two elements are variants of each other: this is shown in Algorithm 1.

**Algorithm 1** Verifying that two finite elements  $(R_0, \mathcal{V}_0, \mathcal{L}_0)$  and  $(R_1, \mathcal{V}_1, \mathcal{L}_1)$  are variants of each other. In this algorithm,  $|S|$  is the cardinality of a set  $S$ .

---

```

1: input  $(R_0, \mathcal{V}_0, \mathcal{L}_0), (R_1, \mathcal{V}_1, \mathcal{L}_1)$ 
2: if  $R_0 \neq R_1$  then
3:   update  $\mathcal{V}_1$  by applying push forward from  $R_1$  to  $R_0$ 
4: end if
5: if  $\mathcal{V}_0 \neq \mathcal{V}_1$  then
6:   return false
7: end if
8: for  $E \subseteq R_0$  do
9:   if  $|\mathcal{L}_0|_E| \neq |\mathcal{L}_1|_E|$  then
10:    return false
11:   end if
12: end for
13: for  $E \subseteq R_0$  do
14:   if  $\gamma^\perp(\mathcal{V}_0, E, \mathcal{L}_0) \neq \gamma^\perp(\mathcal{V}_1, E, \mathcal{L}_1)$  then
15:    return false
16:   end if
17: end for
18: return true

```

---

In Algorithm 1, we first check whether the two implementations of the element have the same reference cell: if not we apply the push forward map (lines 2–4). We then check that they have the same polynomial space (lines 5–7). Next, we check that in each implementation, the same number of DOFs are associated with each sub-entity of the reference cell (lines 8–12). Lastly, we check that the uncontrolled trace spaces for each sub-entity of the reference cell are the same (lines 13–17). If these checks all pass, then by Theorem 1 the two elements are variants of each other.

When implementing this algorithm, however, one cannot simply check that the polynomial spaces in line 5 and the uncontrolled trace spaces in 14 are equal. Hence to test if these spaces are the same, we apply Algorithm 2 to the bases of the polynomial spaces  $\mathcal{V}_0$  and  $\mathcal{V}_1$  (for line 5) and the basis functions not associated with  $E$  restricted to  $E$  (for line 14). For the polynomial spaces in line 5, we additionally check that the spaces have the same dimension.

**Algorithm 2** Checking if two sets of functions  $\mathcal{A} = \{a_0, \dots, a_{n_a-1}\}$  and  $\mathcal{B} = \{b_0, \dots, b_{n_b-1}\}$  span the same space on the entity  $E$ . In this algorithm,  $s_a$  and  $s_b$  are the value sizes of the functions in  $\mathcal{A}$  and  $\mathcal{B}$  and  $[\hat{f}]_k$  is equal to  $f$  if  $f$  is a scalar-valued function, or the  $k$ th component of  $f$  if  $f$  is vector- or tensor-valued.

---

```

1: input ( $\mathcal{A}, \mathcal{B}, E$ )
2: if  $s_a \neq s_b$  then
3:   return false
4: end if
5: let  $P = \{p_0, \dots, p_{n_P}\}$  be a set of points on  $E$ 
6: let  $M_a = (m_{i,j}^a) \in \mathbb{R}^{n_P \times s_a n_a}$ 
7: let  $M_b = (m_{i,j}^b) \in \mathbb{R}^{n_P \times s_b n_b}$ 
8: for  $0 \leq i < n_P$  do
9:   for  $0 \leq j < s_a$  do
10:    for  $0 \leq k < n_a$  do
11:      let  $m_{i,ks_a+j} = [a_k(p_i)]_j$ 
12:    end for
13:    for  $0 \leq k < n_b$  do
14:      let  $m_{i,ks_b+j} = [b_k(p_i)]_j$ 
15:    end for
16:  end for
17: end for
18: if  $\text{rank}(M_a) \neq \text{rank}(M_b)$  then
19:   return false
20: end if
21: return  $\text{rank}(M_a) = \text{rank}\left(\begin{pmatrix} M_a \\ M_b \end{pmatrix}\right)$ 

```

---

Algorithm 2 first checks that the two sets of functions have the same value shape (lines 2–4). In line 5, we then define a set of points on the entity  $E$ . The number of points  $n_P$  must be large enough that the full behaviour of the highest degree polynomial in the input functions is captured: for the DefElement verification, we use a regular lattice of points with the number of points in each direction much larger than the degree of the highest degree space that we verify (with points outside triangle, tetrahedron, prism and pyramid cells removed). We then evaluate each input function at each of these points, and form two matrices that have the values for one input function on each row (lines 6–17). The row spans of these two matrices will be representative of the spans of the

two sets of functions, so we conclude the algorithm by checking that the two matrices have the same rank (lines 18–20), and the rank of the two matrices stacked vertically is the same as the span of one matrix (line 21)—this final check confirms that the row span of the two matrices is the same, as if it was not, the rank of the stacked matrix would be larger.

## 5 Contributing to DefElement

Each element on DefElement is defined by a `.def` file which contains information about the element in yaml format. The full encyclopedia is generated from these files by scripts written in Python. The full code used to generate DefElement is available under an MIT license in the DefElement Github repository [73], with code that is not finite element specific separated into a separate website-build-tools repository for easier reuse by other projects [74].

The adding and updating of information on DefElement is done using a variety of Github features. Changes can be suggested, and errors and bugs can be reported using the issue tracker at [github.com/DefElement/DefElement/issues](https://github.com/DefElement/DefElement/issues). More detailed discussion about DefElement and its features can be held on the discussions board at [github.com/DefElement/DefElement/discussions](https://github.com/DefElement/DefElement/discussions).

More direct proposal of changes to DefElement is encouraged, and can be done by forking the DefElement repository, making the changes, then opening a pull request. The authors of the paper act as the editors of DefElement and are responsible for checking the correctness of new contributions, both those made by other editors and people outside the team. This checking is done via the pull request review feature on Github, and this allows the editors to comment on specific lines have changed and suggest direct corrections to lines for more minor issues like typographical errors.

We strongly encourage readers of this paper to contribute to DefElement, by opening issues for elements that are not yet included in the encyclopedia, suggesting other changes, or contributing directly via pull requests.

### 5.1 Adding a finite element library

Readers of this paper who are involved in the development of a finite element library are encouraged to consider adding their library to the set of libraries that DefElement displays information about. This can be done by adding a file to the folder `defelement/implementations`, this file should contain the definition of a subclass of DefElement's `Implementation` class. This class includes methods that define how the information for the library will appear on the element's page.

The subclass of `Implementation` can optionally include a `verify` function that takes the element's string for the implementation (as set in the `.def` file), the reference cell, the element's degree and a dictionary of additional parameters (as set in the `.def` file) and returns lists of entities associated with each subentity and a function that evaluates the basis functions of the element at some input points. If this function is implemented, then verification will be performed for the library.

A full walkthrough guide to adding a new implementation to DefElement can be found on the DefElement website at [defelement.org/adding-an-implementation.html](https://defelement.org/adding-an-implementation.html). Currently only implementations with a Python interface are supported on DefElement,

although we hope to widen the range of supported languages in the future. Details about making other changes can be found on the DefElement contributing page at [defelement.org/contributing.html](https://defelement.org/contributing.html).

## 6 Concluding remarks

As the number of finite elements that exist in the literature continues to grow, we believe that it is increasingly important that information about elements can be found in a concise and consistent format. We hope that you find the information we have made available on DefElement useful, and that you consider contributing to encyclopedia by editing existing information and adding more details and elements.

### Author contributions

All authors contributed to the manuscript text and reviewed the manuscript.

### Funding

Support from the Engineering and Physical Sciences Research Council for JPD (EP/W026635/1) is gratefully acknowledged. PDB was supported by EPSRC grant EP/W026260/1. IM was supported by EPSRC grant EP/W524311/1.

### Availability of data and materials

DefElement's source code is available from the DefElement Github repository at [github.com/DefElement/DefElement](https://github.com/DefElement/DefElement). An archived snapshot of the code used to build DefElement and the built HTML files is available on Zenodo [75]. This archive is automatically updated every three months.

### Code availability

The full code used to generate DefElement is available under an MIT license in the DefElement Github repository [73]. All website text content and images are available for reuse under a Creative Commons license. Three-monthly snapshots of the source code to generate DefElement and the website HTML are archived on Zenodo at <https://doi.org/10.5281/zenodo.17904468>.

## Declarations

### Ethics approval and consent to participate

Not applicable.

### Competing interests

The authors declare no conflict of interest.

Received: 25 June 2025 Revised: 23 December 2025 Accepted: 11 March 2026

Published online: 14 April 2026

## References

1. Courant R. Variational methods for the solution of problems of equilibrium and vibrations. *Bull Am Math Soc.* 1943;49(1):1–23.
2. Liu WK, Li S, Park HS. Eighty years of the finite element method: birth, evolution, and future. *Archiv Comput Methods Eng.* 2022;29(6):4431–53. <https://doi.org/10.1007/s11831-022-09740-9>.
3. Clough RW. The finite element method after twenty-five years: a personal view. *Comput Struct.* 1980;12(4):361–70. [https://doi.org/10.1016/0045-7949\(80\)90113-3](https://doi.org/10.1016/0045-7949(80)90113-3).
4. Clough RW. The finite element method in plane stress analysis. In: *Proceedings of 2nd American Society of civil engineers conference on electronic computation*, 1960; 345–378
5. Ciarlet PG. *Lectures on the finite element method*. Bombay: Tata Institute of Fundamental Research; 1975.
6. Ciarlet PG. *The finite element method for elliptic problems*. Amsterdam: North-Holland; 1978.
7. Raviart P-A, Thomas J-M. A mixed finite element method for 2nd order elliptic problems. In: Galligani I, Magenes E, editors. *Mathematical aspects of finite element methods*, vol. 606. 1977. p. 292–315. <https://doi.org/10.1007/BFb0064470>.
8. Nédélec J-C. Mixed finite elements in  $\mathbb{R}^3$ . *Numer Math.* 1980;35(3):315–41. <https://doi.org/10.1007/BF01396415>.
9. Guzmán J, Neilan M. Inf-sup stable finite elements on barycentric refinements producing divergence-free approximations in arbitrary dimensions. *SIAM J Numer Anal.* 2018;56:2826–44. <https://doi.org/10.1137/17M1153467>.
10. Rannacher R, Turek S. Simple nonconforming quadrilateral Stokes element. *Num Methods Partial Differ Equ.* 1992;8:97–111. <https://doi.org/10.1002/num.1690080202>.
11. Pechstein AS, Schöberl J. The TDNNS method for Reissner–Mildlin plates. *Numer Math.* 2017;137:713–40. <https://doi.org/10.1007/s00211-017-0883-9>.

12. Arnold DN, Awanou G. The serendipity family of finite elements. *Found Comput Math*. 2011;11(3):337–44. <https://doi.org/10.1007/s10208-011-9087-3>.
13. Python Software Foundation: Python Language Reference, version 2.7. Available at <http://www.python.org>
14. Arnold DN, Logg A. Periodic table of the finite elements. *SIAM News* 2014:47(9)
15. OEIS Foundation Inc. The on-line encyclopedia of integer sequences. [Online; Accessed 11-December-2025] (1996–). <https://oeis.org/>
16. Scroggs MW, Baratta IA, Richardson CN, Wells GN. Basix: a runtime finite element basis evaluation library. *J Open Source Softw*. 2022;7(73):3982. <https://doi.org/10.21105/joss.03982>.
17. Betcke T, Scroggs MW. Bempp-cl: a fast python based just-in-time compiling boundary element library. *J Open Source Softw*. 2021;6(59):2879. <https://doi.org/10.21105/joss.02879>.
18. Kirby RC. Algorithm 839: FIAT, a new paradigm for computing finite element basis functions. *ACM Trans Math Softw*. 2004;30(4):502–16. <https://doi.org/10.1145/1039813.1039820>.
19. Scroggs, M.W.: ndelement: Rust finite element definition library (2024). <https://github.com/bempp/ndelement>
20. Scroggs MW. Symfem: a symbolic finite element definition library. *J Open Source Softw*. 2021;6(64):3556. <https://doi.org/10.21105/joss.03556>.
21. Rao SSM, Wilton DR, Glisson AW. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans Antennas Propag*. 1982;30:409–18. <https://doi.org/10.1109/TAP.1982.1142818>.
22. Cockburn B, Fu G. A systematic construction of finite element commuting exact sequences. *SIAM J Numer Anal*. 2017;55:1650–88. <https://doi.org/10.1137/16M1073352>.
23. Rognes ME, Kirby RC, Logg A. Efficient assembly of  $h(\text{div})$  and  $h(\text{curl})$  conforming finite elements. *SIAM J Sci Comput*. 2009;31(6):4130–51. <https://doi.org/10.1137/08073901X>.
24. Arnold DN, Boffi D, Falk RS. Quadrilateral  $h(\text{div})$  finite elements. *SIAM J Numer Anal*. 2005;42:2429–51. <https://doi.org/10.1137/S0036142903431924>.
25. Kirby RC. A general approach to transforming finite elements. *The SMAI J Comput Math*. 2018;4:197–224. <https://doi.org/10.5802/smai-jcm.33>.
26. Fortin M, Soulie M. A non-conforming piecewise quadratic finite element on triangles. *Int J Numer Meth Eng*. 1983;19:505–20. <https://doi.org/10.1002/nme.1620190405>.
27. Ern A, Guermond J-L. *Finite elements I: approximation and interpolation*. Switzerland: Springer; 2021. <https://doi.org/10.1007/978-3-030-56341-7>.
28. Scroggs MW, Dokken JS, Richardson CN, Wells GN. Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes. *ACM Trans Math Softw*. 2022;48(2):18–11823. <https://doi.org/10.1145/3524456>.
29. Pan VY. How bad are Vandermonde matrices? *SIAM J Matrix Anal Appl*. 2016;37(2):676–94. <https://doi.org/10.1137/15M1030170>.
30. Arnold DN, Boffi D, Falk RS. Approximation by quadrilateral finite elements. *Math Comput*. 2002;71:909–22. <https://doi.org/10.1090/S0025-5718-02-01439-4>.
31. Argyris JH, Fried I, Scharpf DW. The TUBA family of plate elements for the matrix displacement method. *The Aeronautical J*. 1968;72(692):701–9. <https://doi.org/10.1017/S000192400008489X>.
32. Bell K. A refined triangular plate bending finite element. *Int J Numer Meth Eng*. 1969;1(1):101–22. <https://doi.org/10.1002/nme.1620010108>.
33. Ciarlet PG, Raviart P-A. Interpolation theory over curved elements, with applications to finite element methods. *Comput Methods Appl Mech Eng*. 1972;1(2):217–49. [https://doi.org/10.1016/0045-7825\(72\)90006-0](https://doi.org/10.1016/0045-7825(72)90006-0).
34. Morley LSD. The triangular equilibrium element in the solution of plate bending problems. *The Aeronautical Q*. 1968;19(2):149–69. <https://doi.org/10.1017/S0001925900004546>.
35. Wang M, Xu J. Minimal finite element spaces for 2m-th-order partial differential equations in  $\mathbb{R}^n$ . *Math Comput*. 2013;82(281):25–43. <https://doi.org/10.1090/S0025-5718-2012-02611-1>.
36. Wu S, Xu J. Nonconforming finite element spaces for 2mth order partial differential equations on  $\mathbb{R}^n$  simplicial grids when  $m=n+1$ . *Math Comput*. 2019;88:531–51. <https://doi.org/10.1090/mcom/3361>.
37. Arnold DN, Brezzi F, Cockburn B, Marini LD. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J Numer Anal*. 2002;39(5):1749–79. <https://doi.org/10.1137/S0036142901384162>.
38. Reed WH, Hill TR. *Triangular mesh methods for the neutron transport equation*. Technical report LA-UR-73-479, Los Alamos Scientific Laboratory 1973. <https://www.osti.gov/servlets/purl/4491151>
39. Crouzeix M, Raviart P-A. Conforming and nonconforming finite element methods for solving the stationary Stokes equations. *Revue Française d'Automatique Informatique et Recherche Opérationnelle*. 1973;3:33–75. <https://doi.org/10.1051/m2an/197307R300331>.
40. Crouzeix M, Falk RS. Nonconforming finite elements for the Stokes problem. *Math Comput*. 1989;52:437–56. <https://doi.org/10.2307/2008475>.
41. Monk P. *Finite element methods for Maxwell's equations*. Oxford: Oxford University Press; 2003. <https://doi.org/10.1093/acprof:oso/9780198508885.001.0001>.
42. Zienkiewicz OC, Taylor RL, Nithiarasu P. *The finite element method for fluid dynamics*. 7th ed. Oxford: Butterworth-Heinemann; 2014. <https://doi.org/10.1016/C2009-0-26328-8>.
43. Brezzi F, Douglas J, Marini LD. Two families of mixed finite elements for second order elliptic problems. *Numer Math*. 1985;47:217–35. <https://doi.org/10.1007/BF01389710>.
44. Nédélec J-C. A new family of mixed finite elements in  $\mathbb{R}^3$ . *Numer Math*. 1986;50(1):57–81. <https://doi.org/10.1007/BF01389668>.
45. Christiansen SH. On the linearization of Regge calculus. *Numer Math*. 2011;119(4):613–40. <https://doi.org/10.1007/s00211-011-0394-z>.
46. Sky A, Neunteufel M, Hale JS, Zilian A. Formulae and transformations for simplicial tensorial finite elements via polytopal templates. 2024. <https://doi.org/10.48550/arXiv.2405.10402>.
47. Gopalakrishnan J, Lederer PL, Schöberl J. A mass conserving mixed stress formulation for Stokes flow with weakly imposed stress symmetry. *SIAM J Numer Anal*. 2020;58(1):706–32. <https://doi.org/10.1137/19M1248960>.

48. Regge T. General relativity without coordinates. *Il Nuovo Cimento*. 1961;19(3):558–71. <https://doi.org/10.1007/BF02733251>.
49. Arnold DN, Walker SW. The Hellan-Herrmann-Johnson method with curved elements. *SIAM J Numerical Anal*. 2020;58(5):2829–55. <https://doi.org/10.1137/19M1288723>.
50. Hu K, Lin T. Finite element form-valued forms (I): construction 2025. <https://doi.org/10.48550/arXiv.2503.03243>
51. Hu K, Lin T, Zhang Q. Distributional Hessian and divdiv complexes on triangulation and cohomology. *SIAM J Appl Algebra Geometry*. 2025;9(1):108–53. <https://doi.org/10.1137/23M1623860>.
52. Lederer PL. A mass conserving mixed stress formulation for incompressible flows. PhD thesis, Technische Universität Wien 2019. <https://doi.org/10.34726/hss.2019.62042>.
53. Brubeck PD, Kirby RC. FIAT: enabling classical and modern macroelements 2025. <https://doi.org/10.48550/arXiv.2501.14599>
54. Bercovier M, Pironneau O. Error estimates for finite element method solution of the Stokes problem in the primitive variables. *Numer Math*. 1979;33:211–24. <https://doi.org/10.1007/BF01399555>.
55. Alfeld P, Sorokina T. Linear differential operators on bivariate spline spaces and spline vector fields. *BIT Numer Math*. 2016;56:15–32. <https://doi.org/10.1007/s10543-015-0557-x>.
56. Clough RW, Tocher JL. Finite element stiffness matrices for analysis of plate bending. In: Proceedings of the first conference on matrix methods in structural mechanics, 1965:515–546
57. Ciarlet PG. Interpolation error estimates for the reduced Hsieh-Clough-Tocher triangle. *Math Comput*. 1978;32:335–44. <https://doi.org/10.1090/S0025-5718-1978-0482249-1>.
58. Grošelj J, Knez M. Generalized C1 Clough-Tocher splines for CAGD and FEM. *Comput Methods Appl Mech Eng*. 2022. <https://doi.org/10.1016/j.cma.2022.114983>.
59. Powell MJD, Sabin MA. Piecewise quadratic approximations on triangles. *ACM Trans Math Softw*. 1977;3(4):316–25. <https://doi.org/10.1145/355759.355761>.
60. Buffa A, Christiansen SH. A dual finite element complex on the barycentric refinement. *Math Comput*. 2007;76:1743–69. <https://doi.org/10.1016/j.crma.2004.12.022>.
61. Arnold DN. Differential complexes and numerical stability 2002. <https://doi.org/10.48550/arXiv.math/0212391>
62. Arnold DN, Falk RS, Winther R. Finite element exterior calculus, homological techniques, and applications. *Acta Numer*. 2006;15:1–155. <https://doi.org/10.1017/S0962492906210018>.
63. Scroggs M. Deriving recurrence formulae for orthogonal rationally on a pyramid, version 2 2025. <https://doi.org/10.5281/zenodo.15641641>
64. Baratta IA, Dean JP, Dokken JS, Habera M, Hale JS, Richardson CN, Rognes ME, Scroggs MW, Sime N, Wells GN. DOLFINx: the next generation FEniCS problem solving environment 2023. <https://doi.org/10.5281/zenodo.10447666>
65. Ham DA, Kelly PHJ, Mitchell L, Cotter CJ, Kirby RC, Sagiyama K, Bouziani N, Vorderwuelbecke S, Gregory TJ, Betteridge J, Shapero DR, Nixon-Hill RW, Ward CJ, Farrell PE, Brubeck PD, Marsden I, Gibson TH, Homolya M, Sun T, McRae ATT, Luporini F, Gregory A, Lange M, Funke SW, Rathgeber F, Bercea G-T, Markall GR. Firedrake user manual, First edition edn. Imperial College London and University of Oxford and Baylor University and University of Washington, 2023. <https://doi.org/10.25561/104839> Imperial College London and University of Oxford and Baylor University and University of Washington
66. Logg A, Mardal K-A, Wells GN, et al. Automated solution of differential equations by the finite element method. Heidelberg: The FEniCS Book. Springer; 2012. <https://doi.org/10.1007/978-3-642-23099-8>.
67. Dubiner M. Spectral methods on triangles and other domains. *J Sci Comput*. 1991;6(4):345–90. <https://doi.org/10.1007/BF01060030>.
68. Karniadakis G, Sherwin S. Spectral/HP element methods for computational fluid dynamics. Oxford: Oxford University Press; 2005. <https://doi.org/10.1093/acprof:oso/9780198528692.001.0001>.
69. Kirby RC. Singularity-free evaluation of collapsed-coordinate orthogonal polynomials. *ACM Trans Math Softw*. 2010;37(1):1–16. <https://doi.org/10.1145/1644001.1644006>.
70. Runge C. Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. In: *Zeitschrift Für Mathematik und Physik*, 1901;46:224–243
71. Trefethen LN. Approximation theory and approximation practice, Extended Edition. Society for Industrial and Applied Mathematics, Philadelphia, USA 2019. <https://doi.org/10.1137/1.9781611975949>
72. Kreyszig E. Introductory functional analysis with applications. New York: Wiley; 1978.
73. Scroggs MW, Brubeck PD, Dean JP, Dokken JS, Marsden I, Nobre N, et al. DefElement: an encyclopedia of finite element definitions 2020–2025. <https://github.com/DefElement/DefElement>
74. Scroggs MW. website-build-tools 2025. <https://github.com/DefElement/website-build-tools>
75. Scroggs MW, Brubeck PD, Dean JP, Dokken JS, Marsden I. DefElement: an encyclopedia of finite element definitions. Zenodo. 2025. <https://doi.org/10.5281/zenodo.17904469>.