3D Partitioned Element Method: Shape-Function Formulation

Mark Rashid

23 April, 2015

1 Geometric Preliminaries

These notes cover the formulation of the shape functions on a single polyhedral finite element via the "partitioned element method" (PEM). A typical polyhedral finite element (henceforth, simply an "element") is shown in Figure 1. It is important to study this figure in order to understand the geometry-related notation and terminology that will be used throughout these notes.

Preparatory to the computation of the element's shape-function data, the element is first partitioned into *cells* ω_a, which are themselves polyhedra. (Throughout, lower-case Latin subscripts from the beginning of the alphabet, e.g. a, b, c, are used to index topes such as polyhedral cells, polygonal facets, etc.) The boundaries of the cells consist of polygonal facets σ_a, whose boundaries are, in turn, composed of line segments γ_a. Finally, each segment is bounded by a pair of vertices v_a. Some representative facets (σ's), segments (γ's), and vertices (v's) are labeled in Figure 1. Only the topes that lie on the boundary of the element can be seen in the figure; of course there are others in the interior, associated with the partition of the element into cells. The important point at this stage is that, geometrically, the element consists of inventories of cells $\{\omega_a\}$, facets $\{\sigma_a\}$, segments $\{\gamma_a\}$, and vertices $\{v_a\}$ that are logically connected via nested boundary definitions. I.e. each cell is defined by the collection of facets that define its boundary, each facet is defined by the set of segments on its boundary, and each segment is defined by the pair of vertices that constitute its endpoints.

The element's topes will be associated into other sets as well, for purposes that will be discussed shortly. By way of preview, it will prove useful, for example, to organize the facets into disjoint sets that define *element faces*. We shall also have a need for explicit "reverse connectivity" data, such as the cells that contain a given facet, and the facets that contain a given segment. These and other set associations, along with the notation used to describe them, are explained subsequently.

Vertices are indicated in Figure 1 with black dots. Some vertices have red circles around them; these circled vertices are also *nodes*, which are labeled p_a. A node is a point with which there is associated both a shape function, and a value of the function that is being interpolated. An *interpolant* on the element is simply a linear combination of nodal values times their respective shape functions. Not all vertices are nodes; indeed, nodes only occur

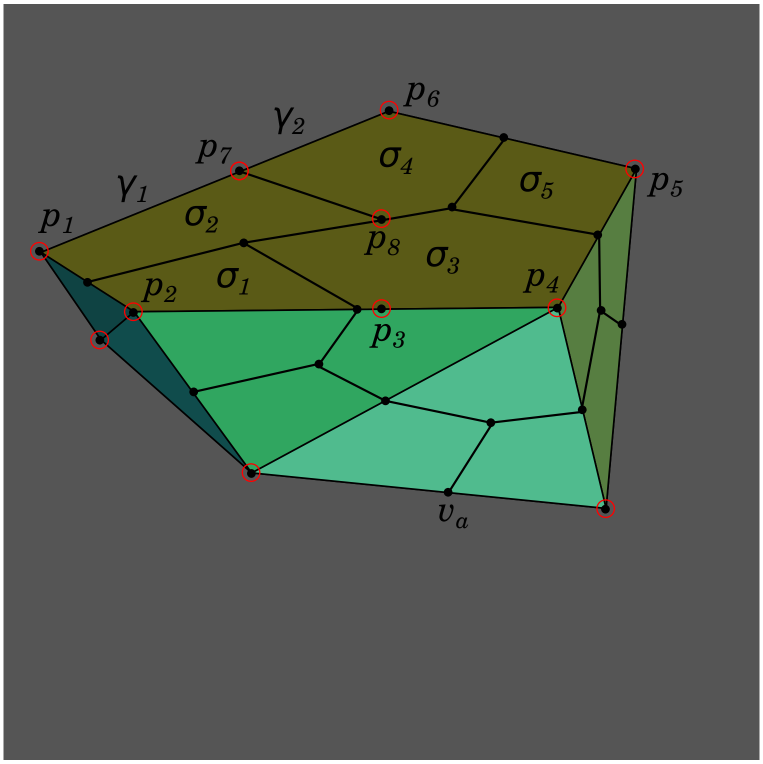

Figure 1: Element geometry.

on the boundary of the element. Even among the boundary vertices, some are nodes while others are not.

Before the element is subdivided into cells, the element geometry is defined by what will be called **source facets**. The *facets* that lie on the element's boundary are the result of subdivision of these source facets, pursuant to the partitioning of the element into cells. (Throughout, the modifier **source** will indicate that the tope in question belongs to the **source mesh** – i.e. the finite element mesh itself, prior to partitioning of any of its elements into cells.)

An element's source facets define the boundary of the element. The source facets are organized into **source faces**, as follows. A single source face consists of a contiguous collection of source facets that are shared with exactly one other element in the mesh, or with void. Note that "contiguous collection" here means that each source facet in the collection shares at least one source segment with another source facet in the collection. Two source facets that touch only at a source vertex would not be considered contiguous.

Next, **source edges** are defined as collections of **source segments**, as follows. Each source edge consists of a contiguous collection of source segments that belong to two source-face boundaries. Source edges usually consist of single, open chains of source segments that are linked together head-to-tail. It will also usually be the case that the endpoints of a source edge are nodes. Nodes may occur at other vertices along the source edge as well.

The association of source facets into source faces, and source segments into source edges, is inherited by the element's facets and segments after the element is partitioned into cells.

In other words, the facets that come from any of the source facets that belong to a single source face are collected together into a **face**. Similarly, the segments that come from any of the source segments that belong to a single source edge are collected together into an **edge**. So, **faces** and **edges** consist of sets of **facets** and **segments**, respectively, of the post-subdivision element, with the set memberships derived from the source-face and source-edge memberships.

2 Data Constructs

At the heart the PEM shape-function formulation is the idea that *some of the segments and facets of the element, and all of the cells, support their own local interpolants.* These local interpolants take the form of complete, low-order polynomials in the element-global Cartesian coordinates. ("Element-global Cartesian coordinates" refers to a fixed Cartesian frame that is aligned with the global frame for the whole problem, but with origin translated to the first node of the element.) In the "default" element formulation, these "tope-specific," or local, interpolants are simply linear polynomials, and thus require 4 coefficients to specify on each tope. However, the development here will accommodate quadratic interpolants as well (10 coefficients each). Local interpolants of even higher order are possible, but are unlikely to exhibit favorable cost-benefit economics. It is important to appreciate that these local interpolants are not strictly "compatible," e.g. the local interpolants on the collection of cells does not fit together into a *continuous* function on the element. However, the local interpolants *will* be chosen so that the discontinuities at inter-tope interfaces is, in an appropriate sense, minimized. These ideas will be elaborated and made precise in the next section.

A further central idea in the PEM shape-function formulation is that the local interpolants described above are *linear* functions of some or all of the element's nodal values. Specifically, the polynomial coefficients of each cell's local interpolant are linearly related to *all* nodal values, while the dependencies for facet and segment local interpolants are confined to particular subsets of nodes. *Which* nodal values influence which segment- and facet-interpolants is intimately related to the concepts of *element faces* and *element edges*, which are explained next.

Clearly not all of the element's facets and segments belong to a face or an edge: only the element-boundary facets, and only *some* of the element-boundary segments, have this status. However, importantly, only the facets and segments that *do* belong to a face or, respectively, to an edge support their own local interpolants. Further, a main purpose of organizing facets and segments into faces and edges is to delineate the sets of nodes on which each local interpolant depends. Specifically:

- If a segment belongs to a particular edge, then the segment has a local interpolant, and the coefficients of this interpolant depend on the nodal values at the nodes contained by the edge.

- If a facet belongs to a particular face, then the facet has a local interpolant, and the coefficients of this interpolant depend on the nodal values at the nodes contained by the face.

- All cells belong to the element, of course; this is the logical extension of facet/face and segment/edge associations. Each cell has a local interpolant, the coefficients of which depend on all nodal values of the element.

In Figure 1, segments γ_1 and γ_2 together make up a single edge. Nodes p_1, p_6, and p_7 are, correspondingly, assigned to γ_1 and γ_2. Turning to the facets, σ_1, σ_2, σ_3, σ_4, and σ_5 make a single element face. Each of these 5 facets is assigned nodes p_{1-8}. Geometrically, element faces need not be planar. However, if a face's bounding edges lie in a plane, but the facets of the face do not, then there will be one or more nodes at interior vertices of the face. In Figure 1, node p_8 is such a face-interior node. As mentioned previously, for the present purposes, node definitions, as well as the association of facets into faces and segments into edges, are assumed given.

The polynomial-coefficient/nodal-value dependencies referred to above are linear. The matrices that define these dependencies are the main objectives of the calculations described in these notes. The notation that will be used for these main results is as follows.

- $\mathbf{P}_a$: The $4 \times n$ (linear local interpolant) or $10 \times n$ (quadratic) matrix that transfers segment γ_a's n nodal values into the polynomial coefficients of its local interpolant.
- $\mathbf{Q}_a$: Similarly, for facet σ_a.
- $\mathbf{R}_a$: The transfer matrix for cell ω_a.

We will also need, for *all* segments and facets, a mapping from a local list of polynomial coefficients, to the coefficients of the corresponding polynomial in the element-global Cartesian coordinates. Consider first the case of a segment. Any complete quadratic polynomial g in the element-global Cartesian coordinates has the form

$$g = C_1 + C_2x + C_3y + C_4z + C_5x^2 + C_6xy + C_7xz + C_8y^2 + C_9yz + C_{10}z^2. \tag{1}$$

When restricted to a line segment, however, this g can be expressed by

$$g = c_0 + c_1\xi \ + \ c_2\xi^2, \tag{2}$$

where ξ is a segment coordinate that measures, e.g., distance along the segment from endpoint 1. We will need, for each segment γ_a, the matrix $\mathbf{L}_a$ that relates the vector $\mathbf{C}$ of global polynomial coefficients to the corresponding vector $\mathbf{c}$ of segment polynomial coefficients, via

$$\mathbf{C} = \mathbf{L}_a\mathbf{c}. \tag{3}$$

The $\mathbf{L}_a$ matrices have dimension 10×3 for a mapping of quadratic polynomials. The corresponding matrices for the facets will be denoted $\mathbf{M}_a$, each of which has dimension 10×6 for a mapping of quadratic polynomials. For the present purposes, all of these matrices are to be regarded as given. In any case, they are easy to derive, and depend only on the direction of the line (for segments) or normal to the plane (for facets) on which the subject tope lies.

We are now in a position to articulate the data that should be stored for each segment, facet, and cell.

- For each **segment** that lies on the boundary of the element:

1. Indices of the two facets that share the segment, followed by the index of the edge on which the segment falls, if the segment belongs to an edge (zero otherwise).
2. Nodal-value-to-polynomial-coefficient matrix $\mathbf{P}_a$. This is only needed for segments that lie on an edge. It should be an allocatable array, and left unallocated for segments that lie on the interiors of faces. It is computed based on the developments given in the next section.
3. Polynomial-coefficient map $\mathbf{L}_a$.
4. Vector $\mathbf{m}_a$ of monomial integrals on the segment. These are integrals of element-global monomials through a certain order, where the requisite order is discussed later.

All of this data should be pre-calculated and stored, with the exception of $\mathbf{P}_a$, whose calculation is explained in the next section. A derived type containing the above-described components should be defined, with one object of this type defined for each segment that lies on the boundary of the element.

- For each **facet** in the element's inventory of facets:

  1. Indices of the two cells that share the facet, followed by the index of the face on which the facet falls, if the facet belongs to a face (zero otherwise). If the facet belongs to a face, then the second (cell) index should be zero.
  2. Nodal-value-to-polynomial-coefficient matrix $\mathbf{Q}_a$. This is only needed for facets that lie on a face. It should be an allocatable array, and left unallocated for facets that lie in the element's interior. Its computation is the subject of the next section.
  3. Polynomial-coefficient map $\mathbf{M}_a$.
  4. Vector $\mathbf{m}_a$ of monomial integrals on the segment. These are integrals of element-global monomials through a certain order, which is discussed later.

  A derived type should be defined with these components, with one object if this type defined for each facet in the element's inventory of facets.

- For each **cell** in the element's inventory of cells:

  1. Nodal-value-to-polynomial-coefficient matrix $\mathbf{R}_a$. This is computed as explained in the next section.
  2. Vector $\mathbf{m}_a$ of monomial integrals on the facet. These are integrals of monomials in the element-global Cartesian coordinates, which are parallel to the mesh-global Cartesian coordinates, but offset by the first node of the element.

In addition to this tope-centric data, we will also need some data that pertains to edges and faces, as follows:

- $\mathcal{P}_a$, $\mathcal{S}_a$, $\bar{\mathbf{P}}_a$: The first of these is the set of nodes associated with edge a (and, therefore, the set of nodes associated with each segment that belongs to edge a). The second is the set of segments that compose edge a. The final item is a real matrix that gives the element-global polynomial coefficients for the linear (4 coefficients) or quadratic (10 coefficients) polynomial that is the least-squares best fit of the edge's nodal data. I.e., $\bar{\mathbf{P}}_a \times$ (vector of nodal values) = (polynomial coefficients).

- $\mathcal{Q}_a$, $\mathcal{F}_a$, $\bar{\mathbf{Q}}_a$: The first of these is the set of nodes associated with face a (and, therefore, the set of nodes associated with each facet that belongs to face a). The second is the set of facets that make up face a. The third item is a real matrix that gives the element-global polynomial coefficients for the linear (4 coefficients) or quadratic (10 coefficients) polynomial that is the least-squares best fit of the face's nodal data. I.e., $\bar{\mathbf{Q}}_a \times$ (vector of nodal values) = (polynomial coefficients).

The "set" components above could consist of allocatable arrays of integers, or of linked lists that point to the actual objects themselves. It is not clear at this stage which would be better. The best-fit matrices for edges and faces are needed for subtle reasons, which are explained in the next section. In any case, these matrices are easily formed once faces and edges have been defined and nodes assigned to them. As such, the above edge- and face-specific data should be regarded as known ahead of the calculations described in the next section.

3 Calculation

For the present purposes, we shall assume that the element's partitioned geometry is known; i.e. we have in hand the complete inventory of vertices, segments, facets, and cells. We shall also assume that all of the above-described data objects are fully pre-defined, with the exception of the $\mathbf{P}_a$, $\mathbf{Q}_a$, and $\mathbf{R}_a$ matrices. Computation of these tope-specific components is the subject of this section.

The basic pattern of the shape-function calculation is as follows. First, we consider each edge in turn. The $\mathbf{P}_a$ matrices for the segments that make up a given edge are the result of a minimization problem on the edge. After treating all edges, we will have all $\mathbf{P}_a$ matrices for the segments for which we need these matrices – i.e. all segments that belong to an edge. Next, we consider the faces, one after another. The $\mathbf{Q}_a$ matrices for the facets of a given face emerge from a minimization problem on the face. The $\mathbf{P}_a$ matrices for the segments that bound the edge also appear in this minimization problem. Solving these minimization problems gives us a $\mathbf{Q}_a$ for each facet for which we need one, namely, the facets that belong to faces. Finally, we come to the calculation of the $\mathbf{R}_a$ matrices for the element's cells, which are the outcomes of a minimization problem on the element. This minimization problem involves the $\mathbf{Q}_a$ matrices from the element's boundary facets. The calculation concludes with any cell-averaging that must occur in order to form the cell averages of the shape-function gradients.

3.1 edges and their segments

And now the details, taking the edge case first. Figure 2 shows a typical edge that contains 3 nodes, 5 vertices, and 4 segments. The nodal values φ_a of the interpolant are assumed given. The interpolant on the edge is defined by a (constant, gradient) pair $(c_a, \mathbf{g}_a)$ for each segment, such that the interpolant is continuous over the entire edge. The constant coefficient c_a for each segment is defined with reference to the element-global coordinate origin, which is taken to be located at node 1 of the element. A further convention is that the segments and vertices are indexed so that segment a extends from vertex a to vertex $a+1$, while the nodes of the edge are indexed separately from 1, and inorder by their underlying vertex indices. In the special case where the edge consists of only a single segment, then the segment's polynomial coefficients can be taken to be

$$\mathbf{g}_1 = \frac{\mathbf{x}_2 - \mathbf{x}_1}{\|\mathbf{x}_2 - \mathbf{x}_1\|}(\varphi_2 - \varphi_1), \quad c_1 = \varphi_1 - \mathbf{g}_1 \cdot \mathbf{x}_1, \tag{4}$$

where $\mathbf{x}_a$ is the position vector of vertex a relative to the element-global origin (i.e. node 1 of the element).

In the more general case, a constrained quadratic minimization problem is invoked to compel the segment coefficients $(c_a, \mathbf{g}_a)$ toward an optimally smooth variation along the edge, as follows. The jumps in gradient at the interior vertices are to be minimized via the objective function

$$\sum_{a=2}^{n} (\mathbf{g}_a - \mathbf{g}_{a-1}) \cdot (\mathbf{g}_a - \mathbf{g}_{a-1}), \tag{5}$$

where n is the number of segments for the edge. The minimum problem is subject to constraints that require, first, that the edge interpolant is continuous at interior vertices; and second, that the interpolant meets the nodal values at the nodes. The equations are

$$\begin{aligned} \text{continuity}: \quad & c_a + \mathbf{g}_a \cdot \mathbf{x}_a = c_{a-1} + \mathbf{g}_{a-1} \cdot \mathbf{x}_a, \quad a = 2, \ldots, n \\ \text{nodal conformity}: \quad & c_a + \mathbf{g}_a \cdot \mathbf{x}_a = \varphi_b, \end{aligned} \tag{6}$$

where b is the nodal index for vertex a, and n is the number of segments. In the case of the last node in the sequence of nodes for the edge, $\mathbf{x}_a$ must be replaced by $\mathbf{x}_{a+1}$. There is one "nodal conformity" constraint per node, resulting in a total of $(n-1) +$ (number of nodes) constraints for the edge.

Minimization of (5) subject to the constraints (6) leads directly to the polynomial-coefficient map $\mathbf{P}_a$ on each segment. The coefficients $(c_a, \mathbf{g}_a)$ are then recovered by forming $\mathbf{P}_a\mathbf{p}$, where $\mathbf{p}$ is the vector of edge nodal values. The calculation proceeds as follows. The objective function (5) can be cast in the form

$$\tfrac{1}{2}\, \mathbf{g}^T \mathbf{A} \mathbf{g}, \tag{7}$$

where $\mathbf{g}$ represents a column vector of the $(c_a, \mathbf{g}_a)$ coefficients for the edges, stacked one above the other. The constraints (6), on the other hand, take the form

$$\mathbf{C}\mathbf{g} = \mathbf{B}\mathbf{p}. \tag{8}$$

The $\mathbf{g}$ vector is to be found that minimizes (7), while satisfying (8) for a given vector $\mathbf{p}$ of nodal values. It is further noted that this contstrained minimum problem may not have a unique solution. For example, if the edge's segements are colinear or even coplanar, the

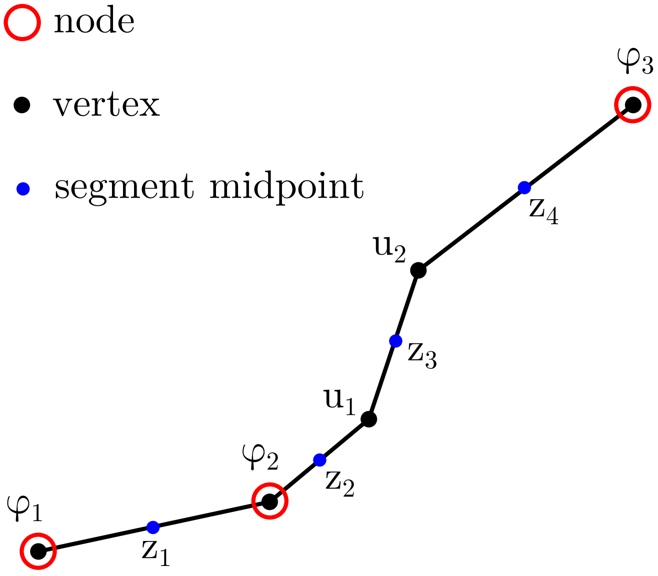

Figure 2: An element edge consisting of 4 segments.

segment gradients $\mathbf{g}_a$ are not uniquely determined. This issue is discussed in more depth below. For the moment, our aim is to formulate a solution procedure that accommodates this possible indeterminacy. To this end, we first express all possible solutions of (8) as

$$\mathbf{g} = \mathbf{C}^{+}\mathbf{B}\mathbf{p} + \mathbf{M}\mathbf{t}, \tag{9}$$

where $\mathbf{C}^{+}$ is the Moore-Penrose pseudo-inverse of $\mathbf{C}$, and $\mathbf{M}$ is a matrix whose columns consist of a basis for the nullspace of $\mathbf{C}$.[1] Substituting (9) into (7) and minimizing with respect to the parameters $\mathbf{t}$ results in a solution for $\mathbf{t}$, which can then be used in (9) to obtain

$$\mathbf{g} = \left[\mathbf{1} - \mathbf{M}(\mathbf{M}^{T}\mathbf{A}\mathbf{M})^{+}\mathbf{M}^{T}\mathbf{A}\right]\mathbf{C}^{+}\mathbf{B}\mathbf{p}. \tag{10}$$

The second appearance of a pseudo-inverse in (10) occurs because $\mathbf{M}^{T}\mathbf{A}\mathbf{M}$ will not be of full rank when the edge is colinear or coplanar. The desired polynomial-coefficient maps $\mathbf{P}_a$ for the edge's segments are recovered as groups of four rows of the matrix on the right-hand side of (10).

It is noted that the objective function (5) is independent of the geometry of the edge: the edge's geometry influences the solution for $(c_a, \mathbf{g}_a)$ only through the constraints (6). It is further noted that (5) is expressed in terms of the "whole gradient" $\mathbf{g}_a$ associated with each segment, and not, say, just the tangential component of that gradient. This choice

[1] Any matrix $\mathbf{C}$ (invertible or not, square or not) can be thought of as a bijective mapping from the row-space of $\mathbf{C}$ to its column-space. Being a bijection, this mapping has an inverse, which obviously maps the column-space of $\mathbf{C}$ to its row-space. This is the Moore-Penrose psuedo-inverse $\mathbf{C}^{+}$. It is equal to the regular inverse $\mathbf{C}^{-1}$ when $\mathbf{C}$ is invertible. If $\mathbf{C}$ is not invertible, then $\mathbf{C}\mathbf{x} = \mathbf{b}$ still has a solution, provided that $\mathbf{b}$ lies in the column space of $\mathbf{C}$. If it does, then in fact there is an infinity of solutions, all of which have the form $\mathbf{C}^{+}\mathbf{b}+$ (an arbitrary vector in the nullspace of $\mathbf{C}$). High-quality routines have already been written to compute the pseudo-inverse of an arbitrary input matrix, as well as a basis for the matrix's nullspace. These can be used in the present work.

merits careful explanation. The essence of the issue is the requirement that the eventual element interpolant exhibit *linear completeness.* Linear completeness calls for an element interpolant that exactly reproduces a linear function, whenever the element's nodal values are set consistent with the linear function. In order for the final shape functions to exhibit linear completeness, it is necessary that the facet-local interpolants and, in their turn, the segment-local interpolants do so as well. In other words, e.g., whenever the vector $\mathbf{p}$ of nodal values for a particular edge are consistent with the linear function $C + \mathbf{G} \cdot \mathbf{x}$, then the polynomial coefficients for each segment of the edge, namely $\mathbf{P}_a\mathbf{p}$, should be consistent with this same linear function. It is tempting to interpret this requirement as implying simply that $\mathbf{P}_a\mathbf{p}$ produce coefficients $(c_a, \mathbf{g}_a)$ that exactly match the original linear-function coefficients $(C, \mathbf{G})$. However, this is not possible in general, as can be seen by taking a simple example. Suppose the edge is straight, and parallel to the x_1-axis. In this case, the nodal values on the edge are insensitive to G_2 and G_3, so there is no way these two coefficients can be determined on the basis of the edge's nodal values. In this special case, linear completeness requires that $g_{1a} = G_1$, but the other cofficients could differ, so long as the segment-local interpolant matches the global linear function *on the segment.*

This insensitivity in the case of a *straight* edge is not "the problem," though it does hint at it. Indeed, linear completeness requires only that the interpolant on each segment of an edge be exactly consistent with any linear function which is used to set the edge's nodal values. In the event that multiple global linear functions produce the same pattern of nodal values, linear completeness requires only that the interpolant matches those global linear functions *evaluated on the edge.* The critical thing for linear completeness is the *consistency* between the nodal values and the interpolant for *all* globally linear functions.

A further example will provide more insight. Consider now an edge containing two segments in the form of a "vee" shape, with two nodes, one at each endpoint. Suppose that the edge lies in the x_1, x_2 plane, and that the two nodes lie on the x_1-axis. The two nodal values establish the coefficients C and G_1, but not G_2 and G_3. But, an interpolant on the edge that takes its nodal values from the linear function $C + G_1x_1 + G_2x_2 + G_3x_3$ would be sensitive to C, G_1, *and* G_2. Therefore, it is not possible for linear completeness to hold in this situation, because the edge interpolant cannot "know about" G_2 if it must depend only on the edge's nodal values.

The obvious solution here would be to place a third node at the apex of the vee. By doing so, we would bring into alignment the coefficients of the global linear function that are determinable by the nodal values – which are now C, G_1, and G_2 – and the coefficients that influence the variation on the edge. In fact, this alignment is necessary in order to achieve linear completeness, and constitutes the main factor in decisions about the placement of nodes at interior vertices of both edges and faces. And, it is the necessity of achieving this alignment that compels the form (5) for the edge objective function. If, instead of the vector gradient $\mathbf{g}_a$, there appeared the tangential derivative, then minimization of the functional *would not lead to linear completeness* on the edge, for non-straight edges. Minimization of (5), subject to the constraints (6), *does* produce linear completeness of the edge interpolant. However, for the common special cases of edges that lie on a line or in a plane, the constrained minimum problem leads to a rank-deficient system of linear equations for the segment $(c_a, \mathbf{g}_a)$ coefficients. This is merely a reflection of the fact that, in these cases, multiple linear functions are consistent with any given "linear setting" of the nodal values. Accordingly, any solution of the rank-deficient system serves as well as any other for the segment interpolant coefficients.

3.2 faces and their facets

Next, we turn to the faces and their constituent facets. The calculation follows a similar general pattern to the edge/segment procedure. Specifically, we take each face in turn and solve a minimization problem on the face, the solution to which is a set of mappings $\mathbf{Q}_a$ that take the face's nodal values into polynomial coefficients of the local interpolant on facet σ_a. Exactly the same linear-consistency issue comes up for faces as did for edges, with much the same solution: if the face's facets are non-coplanar, then the face's nodes must be too; and the minimum problem for the face must be framed in terms of a 3D gradient vector for each facet, despite the fact that only the in-plane component of the gradient influences the interpolant on its facet.

The objective function for a face will look slightly different than it did for edges, because with faces, there is no prospect for achieving a continuous interpolant over the face. Accordingly, we cannot impose constraints that enforce strict continuity at inter-facet segments. Instead, the objective function itself is made to reflect both smoothness and continuity on the interior of the face, as well as consistency with the interpolant variation on the boundary of the face and with any interior nodal value. The proposed objective function has a total of six terms:

$$
\begin{aligned}
&\frac{\alpha(1-\mu)}{L}\sum_{a\in\mathcal{L}}|\gamma_a|\left[(\mathbf{g}_{a+}-\mathbf{g}_{a-})\cdot\mathbf{\Lambda}_a(\mathbf{g}_{a+}-\mathbf{g}_{a-})\right]+\\
&\qquad\frac{\alpha\beta\mu}{L}\sum_{a\in\mathcal{L}}|\gamma_a|\left[\boldsymbol{\lambda}_a\cdot(\mathbf{g}_{a+}-\mathbf{g}_{a-})\right]^2+\\
&\frac{\alpha(1-\beta)}{AL}\sum_{a\in\mathcal{L}}\int_{\gamma_a}\left[(c_{a+}+\mathbf{g}_{a+}\cdot\mathbf{x})-(c_{a-}+\mathbf{g}_{a-}\cdot\mathbf{x})\right]^2 ds+\\
&\qquad\frac{(1-\alpha)\beta\eta}{B}\sum_{a\in\mathcal{B}}|\gamma_a|\left[\boldsymbol{\lambda}_a\cdot(\mathbf{g}_{a+}-\mathbf{g}_{a-})\right]^2+\\
&\frac{(1-\alpha)(1-\beta)\eta}{AB}\sum_{a\in\mathcal{B}}\int_{\gamma_a}\left[(c_{a+}+\mathbf{g}_{a+}\cdot\mathbf{x})-(c_{a-}+\mathbf{g}_{a-}\cdot\mathbf{x})\right]^2 ds+\\
&\qquad\frac{1-\eta}{A}\sum_{a\in\mathcal{I}}\left[c_a+\mathbf{g}_a\cdot\mathbf{x}_1-\varphi_1\right]^2,
\end{aligned}
\tag{11}
$$

where

$$
\mathbf{\Lambda}_a=\mathbf{1}-\boldsymbol{\lambda}_a\otimes\boldsymbol{\lambda}_a,\quad L=\sum_{a\in\mathcal{L}}|\gamma_a|,\quad B=\sum_{a\in\mathcal{B}}|\gamma_a|,\quad A=\sum_a|\sigma_a|,\tag{12}
$$

and where $\boldsymbol{\lambda}_a$ is the unit tangent vector to segment γ_a. Also, $\mathcal{L}$ contains indices of the inter-facet segments, and $\mathcal{B}$ consists of indices of face-boundary segments. If there is an interior node on the face, then φ_1 is the associated nodal value, $\mathbf{x}_1$ is the location of the hosting vertex, and $\mathcal{I}$ contains the indices of the facets that touch this vertex. A is clearly the total area of the face, while L and B are, respectively, the total lengths of inter-facet and face-boundary segments. The numerical parameters α, β, η, μ all lie in the range $(0, 1)$. Their meanings are as follows:

- α: Adjusts the relative importance of the boundary segments vs. the interior segments, with $\alpha \to 1$ emphasizing the interior segments.
- β: Adjusts the relative importance of value-matching on segments vs. matching of the tangential derivative along segments, with $\beta \to 1$ emphasizing the tangential derivative.
- η: Adjusts the relative importance of value-matching with the interior node vs. with the edge interpolants, with $\eta \to 1$ emphasizing the edge interpolants.
- μ: Adjust the relative importance of matching of the normal derivative vs. the tangential derivative on interior segments, with $\mu \to 1$ emphasizing the tangential derivative.

The dimensional scaling parameters L, B and A likewise each have their specific purposes. Their dimensions are length, length, and length squared, respectively. Importantly, L depends on the subdivision of the face into facets, whereas B and A are determined only by the face's geometry. L, the total length of the interior segments, is introduced into the first three terms in order to prevent these terms from dominating, in the event that the subdivision of the face into facets is very fine. The dimensions of each of the six objective-function terms is inverse length squared, such that an isotropic scaling of the element by a factor r would scale all gradients $\mathbf{g}_a$ by $1/r$, with the constants c_a remaining invariant.

The individual terms of the objective function (11) are intended to introduce the following sensitivities:

1. Interior smoothness of the interpolant, as measured by continuity of the normal derivative across inter-facet segments.
2. Interior continuity of the interpolant, as measured by continuity of the tangential derivative across inter-facet segments.
3. Interior continuity of the interpolant, as measured by continuity of the interpolant itself across inter-facet segments.
4. Consistency of the face interpolant with the adjacent edge interpolants, as measured by continuity of the tangential derivatives of the face and edge interpolants. The edge interpolants are given by linear maps of the edge nodes. Thus, this term and term 5 introduce a dependency of the face interpolants on the nodal values at nodes on the boundary of the face.
5. Consistency of the face interpolant with the adjacent edge interpolants, as measured by continuity of the facet interpolant itself with the edge interpolants.
6. Consistency of the face interpolant with an interior nodal value, if the face has an interior node (only a single interior node is allowed).

The objective function (11) is a quadratic function of the facet-local polynomial coefficients $(c_a, \mathbf{g}_a)$. Minimizing the objective function with respect to these coefficients produces a map $\mathbf{Q}_a$ for each facet σ_a of the face, such that $\mathbf{Q}_a\mathbf{p}$ is a column vector of length 4, containing $(c_a, \mathbf{g}_a)$. Here, $\mathbf{p}$ is the column vector of nodal values of the face. Taking a similar approach to the edge case, we cast the objective function (11) in the equivalent form

$$^1\!/_2\, \mathbf{q}^T\mathbf{A}\mathbf{q} + \mathbf{q}^T\mathbf{E}\mathbf{g} + \varphi_1\mathbf{k}\cdot\mathbf{q}, \tag{13}$$

where $\mathbf{q}$ is the vector of stacked coefficients $(c_a, \mathbf{g}_a)$ for the facets of the face, while $\mathbf{g}$ contains the coefficients for the segments on the boundary of the face. At this point in the calculation, the latter will be known as linear functions of the nodal values on the boundary of the face, via the edge solutions (10). The second term reflects the contributions of the fourth and fifth terms of (11), which involve the interpolant on the face's adjacent edges. The third term is only present if the face has an interior node; it comes from the sixth term of the objective function.

Unlike the edge case, the face minimum problem is unconstrained. Nevertheless, calculation of the minimizer of (13) still requires the pseudo-inverse, on account of the fact that the coefficient vector $\mathbf{q}$ is not, in general, uniquely determined by the minimum problem. Specifically, if the face's facets are coplanar, then the gradient components normal to the plane are indeterminate. In this event, any minimizer constitutes a valid solution, as the indeterminacy in $\mathbf{q}$ does not influence the in-plane variation of the interpolant. Taking $\mathbf{p}$ to be a column vector of nodal values at the nodes of the face, the final result is

$$\mathbf{q} = -\mathbf{A}^+\left[\mathbf{E}\mathbf{P} + \mathbf{K}\right]\mathbf{p}, \tag{14}$$

where $\mathbf{K}$ contains the vector $\mathbf{k}$ in the column associated with the interior nodal value φ_1 (if there is one), and is otherwise zero. The mappings $\mathbf{Q}_a$ then consist of groups of four rows of the right-hand-side matrix in (14).

3.3 the element and its cells

Finally we come to the calculation of the cell-local interpolants for the element. The main ideas are very similar to those of the face and edge calculations: a minimization problem is solved on the whole element, the solution to which is cell-local interpolants on each of the element's constituent cells. The principal difference is that the cell-local interpolants must exhibit an attribute that will be called "quadrature consistency," which must be explicitly enforced via a constraint on the minimization problem.

First we put forward the objective function for the element problem. It takes a very similar form to (11) for the faces:

$$\begin{aligned}
\frac{\alpha(1-\mu)}{L}\sum_{a\in\mathcal{L}}|\sigma_a|\left[\mathbf{n}_a\cdot(\mathbf{g}_{a+}-\mathbf{g}_{a-})\right]^2 &+ \\
\frac{\alpha\beta\mu}{L}\sum_{a\in\mathcal{L}}|\sigma_a|\left[(\mathbf{g}_{a+}-\mathbf{g}_{a-})\cdot\mathbf{N}_a(\mathbf{g}_{a+}-\mathbf{g}_{a-})\right] &+ \\
\frac{\alpha(1-\beta)}{V^{2/3}L}\sum_{a\in\mathcal{L}}\int_{\sigma_a}\left[(c_{a+}+\mathbf{g}_{a+}\cdot\mathbf{x})-(c_{a-}+\mathbf{g}_{a-}\cdot\mathbf{x})\right]^2 da &+ \qquad (15)\\
\frac{(1-\alpha)\beta}{B}\sum_{a\in\mathcal{B}}|\sigma_a|\left[(\mathbf{g}_{a+}-\mathbf{g}_{a-})\cdot\mathbf{N}_a(\mathbf{g}_{a+}-\mathbf{g}_{a-})\right] &+ \\
\frac{(1-\alpha)(1-\beta)}{V^{2/3}B}\sum_{a\in\mathcal{B}}\int_{\sigma_a}\left[(c_{a+}+\mathbf{g}_{a+}\cdot\mathbf{x})-(c_{a-}+\mathbf{g}_{a-}\cdot\mathbf{x})\right]^2 da&,
\end{aligned}$$

where now

$$\mathbf{N}_a=\mathbf{1}-\mathbf{n}_a\otimes\mathbf{n}_a,\quad L=\sum_{a\in\mathcal{L}}|\sigma_a|,\quad B=\sum_{a\in\mathcal{B}}|\sigma_a|,\quad V=\sum_{a}|\omega_a|, \qquad (16)$$

and where $\mathbf{n}_a$ is the unit normal vector to facet σ_a. Also, $\mathcal{L}$ contains indices of the inter-cell facets, and $\mathcal{B}$ consists of indices of element-bundary facets. V is the total volume of the element, while L and B are, respectively, the total areas of inter-cell and element-boundary facets. The numerical parameters α, β, μ all lie in the range $(0, 1)$, but are distinct from the counterparts in (11). Their meanings are as follows:

- α: Adjusts the relative importance of the boundary facets vs. the interior facets, with $\alpha \to 1$ emphasizing the interior facets.

- β: Adjusts the relative importance of value-matching vs. matching of the in-plane gradient on facets, with $\beta \to 1$ emphasizing the in-plane gradient.

- μ: Adjust the relative importance of matching of the normal derivative vs. the in-plane gradient on interior segments, with $\mu \to 1$ emphasizing the in-plane gradient.

Note that there is no sixth term in (15), because no interior node is needed or contemplated for the element as a whole.

The intent of each term of the objective function (15) are similar to their counterparts in the face objective function:

1. Interior smoothness of the interpolant, as measured by continuity of the normal derivative across inter-cell facets.

2. Interior continuity of the interpolant, as measured by continuity of the in-plane gradient across inter-cell facets.

3. Interior continuity of the interpolant, as measured by continuity of the interpolant itself across inter-cell facets.

4. Consistency of the element interpolant with the adjacent face interpolants, as measured by continuity of the in-plane gradients of the element and and face interpolants. The face interpolants are given by linear maps of the face nodes. Thus, this term and term 5 introduce a dependency of the cell interpolants on the element's nodal values.

5. Consistency of the element interpolant with the adjacent face interpolants, as measured by continuity of the element interpolant itself with the face interpolants.

The "quadrature consistency" requirement holds, essentially, that a constant stress field should exactly satisfy the weak form of equilibrium on a single element. This will occur if

$$\int_{\Omega} \varphi_{,i} \, dv = \int_{\partial\Omega} n_i \varphi da, \tag{17}$$

where φ is any interpolant on the element (e.g. a shape function), Ω is the element and n_i is the outward unit normal on the element's boundary $\partial\Omega$. This equation must hold when the integrals on both the left- and right-hand sides are evaluated using the element's quadrature rules. It amounts to a requirement that element interpolants satisfy the divergence theorem exactly when the element's quadrature rules are used. Satisfaction of this requirement, together with linear consistency, lead directly to passage of patch tests on arbitrary assemblages of elements.

To proceed further, quadrature rules on the element interior and on element faces must be made explicit. The weak-form integrals generally involve products of shape functions or shape-function derivatives, and some other function – e.g. a stress tensor or a given traction distribution. In the PEM, such integrals are evaluated by first forming a value of the other function for each cell; the function is assumed to be constant at this value over the cell. The integral over the complex of cells is then evaluated analytically, using the shape-function interpolant and the piecewise-constant representation for the other function. In the case of element-domain integrals, the "cells" in this description are the polyhedral cells into which the 3D element was subdivided. For boundary integrals, the cells derive from the subdivision of the domain-boundary face(s) that occurs as a by-product of subdivision of the element domain.

With the quadrature rules in hand, the quadrature-consistency constraint (17) can be cast explicitly as a constraint on the set $\{\mathbf{R}_a\}$ of interpolant maps for the element's cells. The right-hand side of (17) brings in the interpolant maps $\{\mathbf{Q}_a\}$ for the element's boundary facets, but these are considered to be fixed at this point. The $\mathbf{R}_a$'s are yet to be computed, based on minimization of (15), subject to the quadrature-consistency constraints (17).

To formulate this constrained minimization problem, let us consider a single vector $\mathbf{r}$ of polynomial coefficients for the collection of cell-local interpolants. Essentially, $\mathbf{r}$ consists of the vectors $\mathbf{R}_a\mathbf{p}$ stacked one on top of the other, where $\mathbf{p}$ is some fixed, but arbitrary, vector of nodal values for the element. Also, let $\mathbf{q}$ be a similar "stack" of polynomial coefficients $\mathbf{Q}_a\mathbf{p}$ for the facet-local interpolants on the element's boundary. With this notation, the objective function (15) can be cast in the form

$$\frac{1}{2}\,\mathbf{r}^T\mathbf{A}\mathbf{r} + \mathbf{r}^T\mathbf{G}\mathbf{q}, \tag{18}$$

whereas the constraints (17) take the form

$$\mathbf{Br} = \mathbf{Hq}. \tag{19}$$

Our problem is to minimize (18) with respect to $\mathbf{r}$, subject to the linear constraints (19). Here, $\mathbf{B}$, $\mathbf{H}$ are the constraint matrices that follow from enforcement of (17). (Essentially, multiplication of the left- and right-hand sides of (19) by the column vector $\mathbf{p}$ of nodal values produces the left- and right-hand sides of (17), respectively.) Matrices $\mathbf{B}$ and $\mathbf{H}$ each have 3 rows, corresponding to the $i = 1, 2, 3$ components of (17).

To solve this constrained-minimization problem, it is convenient to first express all possible solutions $\mathbf{r}$ of (19) as

$$\mathbf{r} = \mathbf{B}^{+}\mathbf{Hq} + \mathbf{Nt}, \tag{20}$$

where $\mathbf{B}^{+}$ is the Moore-Penrose pseudo-inverse of $\mathbf{B}$, and $\mathbf{N}$ is a matrix whose columns consist of a basis for the nullspace of $\mathbf{B}$. Substituting (20) into (18) and minimizing with respect to the parameters $\mathbf{t}$ results in a solution for $\mathbf{t}$, which can then be used in (20). The final result is

$$\mathbf{r} = \left[\mathbf{B}^{+}\mathbf{H} - \mathbf{N}(\mathbf{N}^{T}\mathbf{AN})^{-1}\mathbf{N}^{T}(\mathbf{AB}^{+}\mathbf{H} + \mathbf{G})\right] \mathbf{q}. \tag{21}$$

Remembering the meaning of $\mathbf{r}$ and $\mathbf{q}$, it is clear that (21) provides the desired relation between the $\mathbf{R}_a$ and $\mathbf{Q}_a$ matrices.