

EXACT INTEGRATION FOR SINGULAR ZIENKIEWICZ AND GUZMÁN–NEILAN FINITE ELEMENTS WITH IMPLEMENTATION

LARS DIENING, JOHANNES STORN, AND TABEA TSCHERPEL

ABSTRACT. We develop a recursive integration formula for a class of rational polynomials in 2D. Based on this, we present implementations of finite elements that have rational basis functions. Specifically, we provide simple MATLAB implementations of the singular Zienkiewicz and the lowest-order Guzmán–Neilan finite element in 2D.

1. INTRODUCTION

The majority of finite elements rely on polynomial functions because they are simple to implement by use of exact integration and differentiation. However, achieving certain types of conformity, such as H^2 -conformity or exact divergence constraints, can be challenging with polynomials; ensuring these properties typically requires a large number of degrees of freedom when working with polynomial functions. By using rational functions these constraints can be satisfied with considerably fewer degrees of freedom. However, to the best of our knowledge, no exact integration formulas for rational polynomials on triangles have been established in the literature. As a result, finite elements based on rational basis functions have not been studied extensively. In this work we develop an exact integration in 2D for a class of rational polynomials and we apply it to several rational finite elements. This allows for a straightforward exact implementation, and paves the way for further investigations of rational finite elements.

Approximation with rational functions. For certain problems rational functions offer better approximation properties than polynomials, see, e.g., [New64; DY86]. Indeed, for functions with certain singularities this is evident. Designing algorithms for rational approximations is a recent topic of research [Aus+21; TNW21; BHH24]. Compared to the polynomial case many questions are still open. Indeed, exact quadrature formulas have been developed mostly for 1D problems, e.g., [Gau93; Gau01], for 2D circular domains without radial dependence [BGHN01; DBP11] and for tensorial problems. A prominent example for the approximation with rational functions are the so-called NURBS used in isogeometric analysis [HCB05], see also [BGGPV22]. Aside from few exceptions [MS16], they are based on a tensorial structure. For the quadrature usually a Duffy transform [SS97] is employed to resolve the singularities, and then quadrature formulas in 1D are applied in the tensorial form. This approach leads to inexact quadrature, but it offers exponential

2020 *Mathematics Subject Classification.* 65D32, 65N30, 76D07, 76M10.

Key words and phrases. quadrature, conforming finite elements, rational functions, singular Zienkiewicz element, Guzmán–Neilan element, MATLAB, structure preservation, exactly divergence free.

convergence in the number of quadrature points, see, e.g., [ACHRS12; Döl+20]. For complex geometries also triangular meshes and the use of rational functions on them are of interest. For those, in [BHH24] the problem of approximation has been addressed by splitting triangles into three quadrilaterals and by transforming them to the unit cube. However, for multivariate rational functions without tensor product structure, exact numerical integration seems to be an open problem, even in the case of dimension $d = 2$.

Finite element methods with rational functions. Finite element spaces based on low order polynomials fail to satisfy certain conformity properties. For example, the Zienkiewicz element is not H^2 -conforming, and for the Taylor–Hood element the corresponding space of discretely divergence-free velocity functions is not exactly divergence-free. However, in some situations such structure preservation in the form of conformity is crucial. Indeed, the lack of H^2 -conformity can cause difficulties in the approximation of solutions to the biharmonic equation; see [IR72; Shi84] for the failure of convergence of the Zienkiewicz element on certain meshes. To obtain H^2 -conformity with polynomial basis functions requires a high number of degrees of freedom. For example, both the Argyris element [AFS68], see also [Cia02, Ch. 2.2.2] and for a hierarchical version [CH21], and the Morgan–Scott element [MS75] (under certain conditions on the mesh) are C^1 -conforming. Both elements use polynomials of 5th order, and hence have 21 degrees of freedom per triangle. On the other hand, the H^2 conforming singular Zienkiewicz element [Zie71, Sec. 10.10], see also [Cia02, Ch. 6.6.1] and Section 2.1 below, has only 12 degrees of freedom per triangle due to the use of rational basis functions.

Divergence-free subspaces of H^1 and H^2 -conforming finite element spaces are related by a de Rham complex. More precisely, the curl of H^2 -conforming elements yields exactly divergence-free H^1 -conforming elements. Therefore, mixed methods for incompressible fluid equations with exact divergence constraint inherit some properties from the H^2 -conforming spaces. The importance of exact divergence constraints lies in the fact that they ensure pressure robustness of mixed methods for incompressible fluid equations, see [Lin09; GLRW12] for specific examples and [JLMNR17] for a discussion illustrating this point in the context of computational fluid dynamics. There are alternatives to ensure pressure robustness by non-conforming methods such as [LLMS17; KVZ21]. However, they inflict extra challenges for time-dependent and for non-linear equations, some of which have not been resolved to date. As for the H^2 -conforming spaces, exactly divergence-free finite elements are available for high polynomial degree, e.g., the Falk–Neilan element [FN13] and the Scott–Vogelius element [SV85; GN14b]. For the latter, using split meshes the polynomial degree can be lowered [GN18; FGNZ22], but the number of degrees of freedom increases; see [ST24] for a comparison. The Guzmán–Neilan element [GN14b], see also Section 2.2 below, uses rational basis functions for the velocity space, and requires considerably fewer degrees of freedom. Note that also a 3D version is available in [GN14a].

Quadrature for rational functions. While exact quadrature formulas can be applied for rational functions with tensor product structure, the theory is much less developed for more general rational functions. On simplices, to the best of our knowledge, no exact quadrature for rational functions is available in the literature, not even in the lowest-dimensional case $d = 2$. In practice, the Duffy transform in

combination with a quadrature of sufficiently high exactness degree is used. This approach is common in the implementation of NURBS, and was also applied to the 2D Guzmán–Neilan finite element in [Sch15]. Although the integration is not exact, geometric convergence in the number of quadrature points is available. For applications in isogeometric analysis this may be sufficient. However, for finite element spaces with exact divergence constraints the lack of an exact quadrature compromises the conformity, and thereby partially diminishes the benefit of the method.

Main contribution. In this work we present an exact quadrature for a class of 2D rational functions on triangles, based on a recursion formula. The quadrature can be computed during an offline phase and tabulated for efficient use. This opens the door for the further investigation of the singular Zienkiewicz element and the Guzmán–Neilan element, and their use in practical applications. Our objective is to demonstrate the overall procedure, and provide an easily accessible implementation in MATLAB. Specifically, in the spirit of [ACF99; ACFK02; BC05; CGH14; Bar15] we provide an implementation of the biharmonic problem discretized by singular Zienkiewicz element and of the Stokes problem discretized by the lowest-order Guzmán–Neilan element. All code is available at [DST24]. Since the focus of this work is not on highly optimized performance, any improvement in this direction is left to future work.

Outline. In Section 3 we present an exact recursive integration formula for a large class of rational polynomials in $d = 2$ dimensions. The class of rational polynomials includes basis functions occurring in singular Zienkiewicz and in Guzmán–Neilan finite elements, which are introduced in Section 2.1 and Section 2.2, respectively. Section 4.1 provides a general guideline for implementing finite element methods with such rational basis functions. This is illustrated through simple and short MATLAB implementations of the biharmonic problem discretized by the singular Zienkiewicz element and the Stokes problem discretized by the lowest-order Guzmán–Neilan element. The implementations are discussed in Section 4.2 and 4.3, respectively. We conclude our investigation with a numerical example of the impact of inexact integration on the biharmonic eigenvalue problem in Section 5, highlighting the need for exact integration rules.

2. FINITE ELEMENT SPACES

In this section we introduce the singular Zienkiewicz in Section 2.1, and the lowest-order Guzmán–Neilan element in Section 2.2.

Throughout this work we denote by \mathcal{T} a triangulation in the sense of Ciarlet [Cia02, § 2.1, p. 38] of a two-dimensional, bounded, polyhedral domain Ω . By \mathcal{N} we denote the set of all vertices in \mathcal{T} . A triangle $T = [v_0, v_1, v_2] \in \mathcal{T}$ is determined by its vertices $v_0, v_1, v_2 \in \mathbb{R}^2$. The set of all faces/edges f of T is $\mathcal{F}(T) = \{f_0, f_1, f_2\}$, where f_j denotes the edge opposite of vertex v_j for all $j = 0, 1, 2$. The outer unit normal of $f \in \mathcal{F}(T)$ is denoted by ν_f . By $\text{mid}(T)$ and $\text{mid}(f)$ we denote their respective midpoints. Whenever indices exceed 2, they are understood as modulo 3, e.g., $v_3 = v_0$. The space of polynomials on T of maximal degree $k \in \mathbb{N}_0$ is denoted by $\mathcal{P}_k(T)$. Moreover, let $\lambda_0, \lambda_1, \lambda_2 \in \mathcal{P}_1(T)$ denote the barycentric coordinates associated to the vertices v_0, v_1, v_2 , that is, $\lambda_j(v_k) = \delta_{j,k}$ for all $j, k = 0, 1, 2$. We denote the vector of barycentric coordinates by $\lambda = (\lambda_0, \lambda_1, \lambda_2)^\top \in [0, 1]^3$. We shall frequently use that these functions

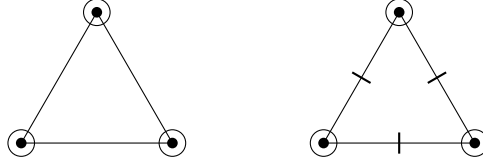


FIGURE 1. Degrees of freedom in the Zienkiewicz element and the reduced singular Zienkiewicz (left) as well as in the singular Zienkiewicz (right) element. Dots denote point evaluation, circles denote evaluation of the gradient, and straight lines denote the evaluation of the normal derivative.

form the partition of unity $\lambda_0 + \lambda_1 + \lambda_2 = 1$. The barycentric coordinates represent a mapping from T onto the reference simplex $\hat{T} = [\hat{v}_0, \hat{v}_1, \hat{v}_2] \subset \mathbb{R}^3$ with vertices

$$\hat{v}_0 = (1, 0, 0)^\top, \quad \hat{v}_1 = (0, 1, 0)^\top, \quad \hat{v}_2 = (0, 0, 1)^\top. \quad (2.1)$$

If there is no risk of misunderstanding ∇ denotes the gradient with respect to x . If a gradient is taken with respect to another variable y instead, for clarity we denote the gradient by ∇_y .

2.1. Singular Zienkiewicz element. The singular Zienkiewicz element [Cia02, Thm. 6.1.4] is a C^1 -conforming finite element. It is an enrichment of the *Zienkiewicz element*, which is a reduced cubic Hermite element, see [Cia02, Thm. 2.2.9 and Fig. 2.2.16]. For this reason, let us first recall the Zienkiewicz element. It is globally C^0 -conforming and gradients are continuous in the vertices. Its 9 degrees of freedom on each triangle T are the point values of the function and the point values of the gradient at its vertices, cf. Figure 1. The local space is a subspace of $\mathcal{P}_3(T)$, and the latter has dimension 10. Since the local space has only 9 degrees of freedom, one considers only polynomials $p \in \mathcal{P}_3(T)$ which satisfy the additional condition

$$\psi(p) := 6p(\text{mid}(T)) - 2 \sum_{j=0}^2 p(v_j) + \sum_{k=0}^2 \nabla p(v_k) \cdot (v_k - \text{mid}(T)) = 0. \quad (2.2)$$

In other words, the local space reads $Z(T) := \{p \in \mathcal{P}_3(T) : \psi(p) = 0\}$.

Lemma 2.1 (Zienkiewicz element).

(a) The local space $Z(T)$ can be represented as

$$Z(T) = \mathcal{P}_2(T) \oplus \text{span}\{\lambda_j^2 \lambda_{j+1} - \lambda_{j+1}^2 \lambda_j\}_{j=0,1,2}.$$

Its degrees of freedom are the values of p and ∇p at the vertices of T for $p \in Z(T)$, see Figure 1 (left).

(b) The corresponding global space Z is C^0 -conforming and is given by

$$Z = \{v \in H^1(\Omega) : v|_T \in Z(T) \text{ for all } T \in \mathcal{T}\}.$$

Proof. Thanks to the structure of ψ based on a Taylor expansion of second order, one can see that $\psi(p) = 0$ for any $p \in \mathcal{P}_2(T)$. Since ψ is linear and invariant under affine transformation, we obtain for any $i, j \in \{0, 1, 2\}$ with $i \neq j$

$$\psi(\lambda_i^2 \lambda_j - \lambda_i \lambda_j^2) = \psi(\lambda_i^2 \lambda_j) - \psi(\lambda_i \lambda_j^2) = \psi(\lambda_i^2 \lambda_j) - \psi(\lambda_j \lambda_i^2) = 0.$$

Hence, $\psi(p) = 0$ holds also for all $p \in \mathcal{P}_2(T) \oplus \text{span}\{\lambda_j^2 \lambda_{j+1} - \lambda_{j+1}^2 \lambda_j\}_{j=0,1,2}$. Since the basis functions are linearly independent and the dimensions agree, this

shows the representation of $Z(T)$. The second statements can be found in [Cia02, Thm. 2.2.9 and 2.2.10]. \square

There exist modifications of the polynomial space $Z(T)$ with improved properties [WSX07]. However, the following considerations show the difficulties when aiming for C^1 -conformity.

Let T_- and T_+ be two neighboring triangles with joint edge $f = T_- \cap T_+$ and normal vector ν of f . Let $p_- \in Z(T_-)$ and $p_+ \in Z(T_+)$ be local Zienkiewicz polynomials that coincide at their common degrees of freedom and set

$$p: T_- \cup T_+ \rightarrow \mathbb{R} \quad \text{with } p|_{T_-} := p_- \text{ and } p|_{T_+} := p_+. \quad (2.3)$$

Then p is continuous. However, in general p is not continuously differentiable, since the normal derivative of p at f may jump. To achieve C^1 -conformity, additional degrees of freedom are needed on the joint edge f , which in turn requires enriching the local spaces. For this purpose an extra basis functions B_f is needed that allows to adjust the normal derivative at f . More precisely, we require that

$$B_f = 0 \text{ on } \partial T_- \text{ and } \partial T_+, \quad (2.4a)$$

$$\nabla B_f = 0 \text{ on } \partial(T_- \cup T_+), \quad (2.4b)$$

$$\partial_\nu B_f \in \mathcal{P}_2(f) \text{ and } \partial_\nu B_f(\text{mid}f) \neq 0. \quad (2.4c)$$

This ensures that B_f does not interfere with the other degrees of freedom.

Remark 2.2 (Polynomial bases). *The properties in (2.4) cannot not be achieved using polynomials which we will discuss here. This is the reason, why non-polynomial basis function like rational bubbles are needed for this approach, see Lemma 2.3.*

We consider the simplices

$$T^- = \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] \quad \text{and} \quad T^+ = \left[\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right].$$

Their common face reads $f = [(\frac{1}{0}), (\frac{0}{1})]$. For contradiction, suppose that there is a polynomial B_f satisfying (2.4). Then, the first two conditions require that for some polynomial p one has

$$B_f(x, y) = x^2 y^2 (1 - x - y) p(x, y) \quad \text{for all } (x, y) \in T^-.$$

This implies that $\partial_\nu B_f(x, 1 - x) = -\sqrt{2} x^2 (1 - x) p(x, 1 - x)$ on f . The condition $\partial_\nu B_f \in \mathcal{P}_2(f)$ can only be fulfilled if $p(x, 1 - x) = 0$, but then we have $\partial_\nu B_f = 0$ on f . This contradicts (2.4c).

The singular Zienkiewicz element [Cia02, Thm. 6.1.4] uses a rational bubble function for B_f . More precisely, for $j = 0, 1, 2$ let $f_j = [v_{j+1}, v_{j+2}]$ denote the edge connecting the vertices v_{j+1} and v_{j+2} of T . In other words, f_j is the edge opposite of v_j in T . Let ν_{f_j} denote the outer unit normal vector of f_j . We denote the cubic element bubble by $b_T := \lambda_0 \lambda_1 \lambda_2$, the quadratic edge bubble by $b_{f_j} := \lambda_{j+1} \lambda_{j+2}$ and the rational bubble by

$$B_{f_j} := \frac{b_T b_{f_j}}{(\lambda_j + \lambda_{j+1})(\lambda_j + \lambda_{j+2})} = \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_{j+1} \lambda_{j+2}}{(1 - \lambda_{j+1})(1 - \lambda_{j+2})}. \quad (2.5)$$

The quintic polynomial in the numerator ensures (2.4a) and (2.4b). On f_j the denominator reduces to $\lambda_{j+1} \lambda_{j+2}$, so on f_j the function B_{f_j} is a cubic polynomial. This ensures that its derivative is quadratic on f_j . The following lemma summarizes the important properties of B_{f_j} .

Lemma 2.3 (Rational bubbles, e.g., [GN14b, Lem. 2.1]). *For all $T \in \mathcal{T}$ and all indices $j = 0, 1, 2$ with edges f_j of T , the rational bubble B_{f_j} is in $C^1(T)$ and satisfies*

- (a) $B_{f_j}|_{\partial T} = 0$,
- (b) $\nabla B_{f_j}|_{f_k} = 0$ for all $k = 0, 1, 2$ with $k \neq j$,
- (c) $\nabla B_{f_j}|_{f_j} = (\nabla b_T|_{f_j} \cdot \nu_{f_j})\nu_{f_j} = -b_{f_j}\nabla\lambda_j = -|\nabla\lambda_j|b_{f_j}\nu_{f_j} \in \mathcal{P}_2(f_j)^2$.

The local space of the singular Zienkiewicz element is defined as

$$Z_s(T) := Z(T) \oplus \text{span}\{B_{f_j} : j = 0, 1, 2\}. \quad (2.6)$$

Lemma 2.4 (Singular Zienkiewicz element [Cia02, Thm. 6.1.4]).

- (a) *Any $w \in Z_s(T)$ is uniquely determined by the values $w(v_j)$, $\nabla w(v_j)$ and $\nabla w(\text{mid}(f_j)) \cdot \nu_{f_j}$, $j = 0, 1, 2$, i.e., those are its degrees of freedom, see Figure 1 (right).*
- (b) *The corresponding global space Z_s is C^1 -conforming and is given by*

$$Z_s = \{w \in H^2(\Omega) : w|_T \in Z_s(T) \text{ for all } T \in \mathcal{T}\}.$$

Note that $w \in Z_s$ is continuous and its tangential derivatives are continuous on each face. Furthermore, the normal derivatives on each face are quadratic and coincide on the end points and the mid points of each face. Therefore, they are continuous as well.

Continuity of the normal derivatives can be achieved with a smaller local space, namely the one of the so-called reduced singular Zienkiewicz element

$$Z_r(T) := \{w \in Z_s(T) : \nabla w|_f \cdot \nu_f \in \mathcal{P}_1(f) \text{ for all edges } f \in \mathcal{F}(T)\}. \quad (2.7)$$

The functions in $Z_r(T)$ can be decomposed into quadratic functions and the functions $\lambda_j^2\lambda_i - \lambda_i^2\lambda_j$, $i, j \in \{0, 1, 2\}$ with $i \neq j$, corrected by the rational bubble function B_{f_k} , $k \in \{0, 1, 2\}$, such that its normal derivatives are linear on the faces.

Lemma 2.5 (Reduced singular Zienkiewicz element [Cia02, Sec. 6.1.6]).

- (a) *Any $w \in Z_r(T)$ is uniquely determined by the values $w(v_j)$ and $\nabla w(v_j)$ i.e., those are its degrees of freedom, see Figure 1 (left).*
- (b) *The corresponding global space Z_r is C^1 -conforming and is given by*

$$Z_r = \{w \in H^2(\Omega) : w|_T \in Z_r(T) \text{ for all } T \in \mathcal{T}\}.$$

2.2. Guzmán–Neilan element. The Guzmán–Neilan element [GN14b] is a mixed finite element consisting of a pressure space Q and a velocity space V . For those, the discretely divergence-free velocity functions are even exactly divergence-free. They are constructed in such a way that the divergence-free velocity functions are represented as the curl of some C^1 -conforming Zienkiewicz type finite element. With those they form an exact sequence with compatible projection operators.

Let us summarize the construction in 2D for the lowest-order Guzmán–Neilan element in case the singular Zienkiewicz element from Section 2.1 is used. Note that in the original contribution a slightly modified Zienkiewicz type element is used, which we present in Remark 2.8 below, but this does not change any essential properties. For functions $g: \mathbb{R}^2 \rightarrow \mathbb{R}$ we define

$$\mathbf{curl} g := \begin{pmatrix} \partial_{x_2} g \\ -\partial_{x_1} g \end{pmatrix} = R \nabla g \quad \text{with } R := \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \quad (2.8)$$

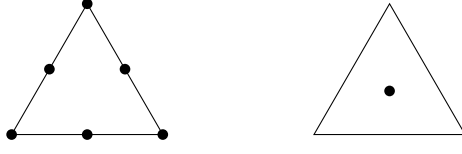


FIGURE 2. Degrees of freedom in the Guzmán–Neilan elements for the vector-valued velocity (left) and the pressure (right).

As above let $b_T := \lambda_0 \lambda_1 \lambda_2$ be the element bubble function, let $b_{f_j} := \lambda_{j+1} \lambda_{j+2}$ be the edge bubble functions, and let B_{f_j} be the rational bubble functions as in (2.5), for $j = 0, 1, 2$. The local pressure space and velocity space are given by

$$Q(T) := \mathcal{P}_0(T), \quad (2.9a)$$

$$V(T) := \mathcal{P}_1(T)^2 \oplus \text{span}\{\mathbf{curl}(\lambda_j^2 \lambda_{j+1} - \lambda_{j+1}^2 \lambda_j)\}_{j=0,1,2} \oplus \text{span}\{\mathbf{curl} B_{f_j}\}_{j=0,1,2}. \quad (2.9b)$$

Note that we have the representation (without direct sum)

$$V(T) = \mathcal{P}_1(T)^2 + \mathbf{curl} Z_s(T). \quad (2.10)$$

Since $V(T) \subset \mathcal{P}_2(T)^2 + \text{span}\{\mathbf{curl} B_{f_j}\}_{j=0,1,2}$ it follows by Lemma 2.3 (c) that

$$v|_{f_j} \in \mathcal{P}_2(f_j)^2 \quad \text{for any } v \in V(T). \quad (2.11)$$

This ensures that using the degrees of freedom of the vector-valued second-order Lagrange finite element functions, see Figure 2, one obtains globally continuous functions.

Lemma 2.6 (Guzmán–Neilan element [GN14b, Sec. 3.1–3.2]).

- (a) Any $v \in V(T)$ is uniquely determined by the values v at vertices and the midpoints of the faces, see Figure 2 (left). Any $q \in Q(T)$ is uniquely determined by its value at the barycenter, see Figure 2 (right).
- (b) The global space V is C^0 -conforming and the global spaces are given by

$$V = \{v \in H^1(\Omega)^2 : v|_T \in V(T) \text{ for all } T \in \mathcal{T}\},$$

$$Q = \{v \in L^2(\Omega) : v|_T \in Q(T) \text{ for all } T \in \mathcal{T}\}.$$

A special feature of the Guzmán–Neilan element is the fact that they are part of an exact discrete de Rham complex [GN14b, Eq. 4.5]

$$\mathbb{R} \longrightarrow Z_s \xrightarrow{\mathbf{curl}} V \xrightarrow{\text{div}} Q \longrightarrow 0.$$

Note that when including zero-traces the Guzmán–Neilan elements are inf-sup stable, as proved in [GN14b] by use of a Fortin operator.

Remark 2.7 (Divergence-free functions). *The spaces of discretely divergence-free velocity functions are defined by*

$$V_{\text{div}} := \left\{ v_h \in V : \int_{\Omega} q_h \text{div } v_h \, dx = 0 \text{ for all } q_h \in Q \right\}.$$

Due to the representation of the velocity space in (2.10) we have that $\text{div } V \subset Q$. Therefore, the discretely divergence-free velocity functions are exactly divergence-free, that is $V_{\text{div}} \subset H_{\text{div}}^1(\Omega) := \{v \in H^1(\Omega)^2 : \text{div } v = 0\}$.

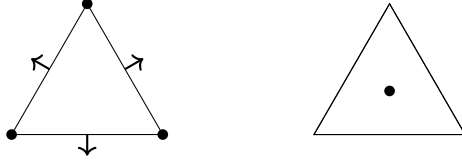


FIGURE 3. Degrees of freedom in the reduced Guzmán–Neilan elements for the vector-valued velocity (left) and the pressure (right). Dots denote point evaluations and arrows evaluation of the normal component.

Remark 2.8 (Alternative space). *Instead of using the standard Zienkiewicz elements (see Section 2.1), Guzmán and Neilan use in [GN14b] an alternative C^1 -conforming Zienkiewicz type element for their construction. It has the same degrees of freedom, but the local velocity space reads*

$$\tilde{Z}_s(T) = \mathcal{P}_2(T) \oplus \text{span}\{\lambda_j^2 \lambda_{j+1}\}_{j=0,1,2} \oplus \text{span}\{B_{f_j}\}_{j=0,1,2}.$$

This means that they use the basis functions $\lambda_j^2 \lambda_{j+1}$ instead of $\lambda_j^2 \lambda_{j+1} - \lambda_{j+1}^2 \lambda_j$, $j = 0, 1, 2$. Then, instead of $V(T)$ as in (2.9) they use the local spaces

$$\begin{aligned} \tilde{V}(T) &:= \mathcal{P}_1(T)^2 \oplus \text{span}\{\mathbf{curl}(\lambda_j^2 \lambda_{j+1})\}_{j=0,1,2} \oplus \text{span}\{\mathbf{curl} B_{f_j}\}_{j=0,1,2}, \\ \tilde{Q}(T) &:= \mathcal{P}_0(T). \end{aligned}$$

The pressure space is the same as above, but the velocity space is slightly different. The corresponding global spaces form an exact sequence. All the results cited in this section are proved in [GN14b] for this construction, but also hold for the version of the spaces presented here. This includes inf-sup stability, commuting projection properties, and the exactness of the discrete de Rham complex.

Reduced Guzmán–Neilan element. Guzmán and Neilan have also developed a reduced element in [GN14b] with fewer degrees freedom for the local velocity space, see Figure 3. The local spaces can be written with tangential unit vectors τ_f as

$$\begin{aligned} V_r(T) &:= \{v \in V(T) : v|_f \cdot \tau_f \in \mathcal{P}_1(f) \text{ for all } f \in \mathcal{F}(T)\}, \\ Q_r(T) &:= \mathcal{P}_0(T). \end{aligned}$$

Since the tangential unit vectors can be obtained by $\tau_f = R^\top \nu_f$ with rotation matrix R defined in (2.8), we have for each $v \in Z(T)$ that

$$\mathbf{curl} v|_f \cdot \tau_f = R \nabla v|_f \cdot \tau_f = \nabla v|_f \cdot R^\top \tau_f = \nabla v|_f \cdot \nu_f.$$

Since $\nabla v|_f \cdot \nu_f \in \mathcal{P}_1(f)$ with $v \in Z(T)$ if and only if $v \in Z_s(T)$, we obtain

$$V_r(T) = \mathcal{P}_1(T)^2 + \mathbf{curl} Z_r(T). \quad (2.12)$$

It is possible to determine an explicit basis of $V_r(T)$, but in our implementation this is computed on the fly.

Lemma 2.9 (Reduced Guzmán–Neilan element [GN14b, Sec. 6]).

- (a) *Any $v \in V_r(T)$ is uniquely determined by the values of v at vertices and the values of $v \cdot \nu_f$ at the edge midpoints, see Figure 3 (left). Any $q \in Q(T)$ is uniquely determined by its value at the barycenter, see Figure 3 (right).*

(b) The global space V_r is C^0 -conforming and the global spaces are given by

$$\begin{aligned} V_r &= \{v \in H^1(\Omega)^2 : v|_T \in V_r(T) \text{ for all } T \in \mathcal{T}\}, \\ Q_r &= \{v \in L^2(\Omega) : v|_T \in Q_r(T) \text{ for all } T \in \mathcal{T}\}. \end{aligned}$$

Since for the reduced Guzmán–Neilan element we have $V_r \subset V$, also the corresponding discretely divergence-free subspace of V_r is exactly divergence-free, cf. Remark 2.7.

3. RATIONAL POLYNOMIALS

Certain types of rational polynomials occur in the 2D singular Zienkiewicz and in the Guzmán–Neilan element, as introduced in Section 2. For the implementation of those elements no exact quadrature is available in the literature. Instead, [Sch15] presents an inexact quadrature that resolves the corner singularities by use of the Duffy transformation applied to the rational function, and then uses Gauß quadrature to approximate the integral. Here, in Section 3.2 we present exact recursive quadrature formulas for a class of rational polynomials that include the ones in Section 2. Furthermore, we show properties of rational polynomials in Section 3.1 that we exploit in the recursion.

Throughout this section $T = [v_0, v_1, v_2]$ denotes a triangle with barycentric coordinates $\lambda = (\lambda_0, \lambda_1, \lambda_2)$ with $\lambda_0 + \lambda_1 + \lambda_2 = 1$. For multi-indices $\alpha = (\alpha_0, \alpha_1, \alpha_2)$ and $\beta = (\beta_0, \beta_1, \beta_2)$ in \mathbb{N}_0^3 we define the rational function

$$\mathfrak{R}_\beta^\alpha := \mathfrak{R}_\beta^\alpha(\lambda) := \frac{\lambda^\alpha}{(1-\lambda)^\beta} = \frac{\prod_{j=0}^2 \lambda_j^{\alpha_j}}{\prod_{j=0}^2 (1-\lambda_j)^{\beta_j}}. \quad (3.1)$$

With this notation and the canonical basis vectors $e_j \in \mathbb{R}^3$, the rational bubble functions B_{f_j} in (2.5) read

$$B_{f_{j-1}} = \mathfrak{R}_{(1,1,1)-e_j}^{(2,2,2)-e_j} \quad \text{for } j = 1, 2, 3.$$

3.1. Differentiation and regularity. Let $\mathfrak{R}_\beta^\alpha : T \rightarrow \mathbb{R}$ be a rational polynomial with multi-indices $\alpha, \beta \in \mathbb{N}_0^3$ and let $\hat{\mathfrak{R}}_\beta^\alpha : \hat{T} \rightarrow \mathbb{R}$ be the function with $\hat{\mathfrak{R}}_\beta^\alpha \circ \lambda = \mathfrak{R}_\beta^\alpha$. To simplify the presentation, for $j = 0, 1, 2$ we use the notation

$$\nabla_\lambda \mathfrak{R}_\beta^\alpha := (\nabla_\lambda \hat{\mathfrak{R}}_\beta^\alpha) \circ \lambda \quad \text{and} \quad \partial_{\lambda_j} \mathfrak{R}_\beta^\alpha := (\partial_{\lambda_j} \hat{\mathfrak{R}}_\beta^\alpha) \circ \lambda. \quad (3.2)$$

The chain rule relates the gradients $\nabla_x \mathfrak{R}_\beta^\alpha : T \rightarrow \mathbb{R}^2$ and $\nabla_\lambda \hat{\mathfrak{R}}_\beta^\alpha : \hat{T} \rightarrow \mathbb{R}^3$ via

$$\nabla_x \mathfrak{R}_\beta^\alpha = \nabla_x (\hat{\mathfrak{R}}_\beta^\alpha \circ \lambda) = \nabla_x \lambda (\nabla_\lambda \hat{\mathfrak{R}}_\beta^\alpha) \circ \lambda = \nabla_x \lambda \nabla_\lambda \mathfrak{R}_\beta^\alpha. \quad (3.3)$$

This extends to vector-valued functions in the obvious manner. Furthermore, the Hessian matrix of $\mathfrak{R}_\beta^\alpha$ reads

$$\nabla_x^2 \mathfrak{R}_\beta^\alpha = \nabla_x \lambda ((\nabla_\lambda^2 \hat{\mathfrak{R}}_\beta^\alpha) \circ \lambda) (\nabla_x \lambda)^\top = \nabla_x \lambda (\nabla_\lambda^2 \mathfrak{R}_\beta^\alpha) (\nabla_x \lambda)^\top. \quad (3.4)$$

These formulae allow us to differentiate rational polynomials, once we know $\nabla_x \lambda$ and $\partial_{\lambda_j} \mathfrak{R}_\beta^\alpha$. The computation of $\nabla_x \lambda$ is discussed in (4.2) below, and differentiation with respect to the barycentric coordinates is subject of the following lemma.

Lemma 3.1 (Multiplication and differentiation). *For $\alpha, \beta, \sigma, \tau \in \mathbb{N}_0^3$ we have*

$$\mathfrak{R}_\beta^\alpha \cdot \mathfrak{R}_\tau^\sigma = \mathfrak{R}_{\beta+\tau}^{\alpha+\sigma}, \quad (3.5)$$

$$\partial_{\lambda_j} \mathfrak{R}_\beta^\alpha = \alpha_j \mathfrak{R}_\beta^{\alpha-e_j} - \beta_j \mathfrak{R}_{\beta+e_j}^\alpha \quad \text{for } j \in \{0, 1, 2\}. \quad (3.6)$$

Proof. The first identity follows directly by the definition of $\mathfrak{R}_\beta^\alpha$ in (3.1). The second identity is a consequence of the quotient rule. \square

Lemma 3.2 (Regularity). *For $m \in \mathbb{N}_0$ and $\alpha, \beta \in \mathbb{N}^3$ we have $\mathfrak{R}_\beta^\alpha \in W^{m,p}(T)$ for $p \in [1, \infty)$ if and only if $|\alpha| - \|\alpha + \beta\|_\infty > m - 2/p$, and for $p = \infty$ if and only if $|\alpha| - \|\alpha + \beta\|_\infty \geq m$. In particular, one has*

- (i) $\mathfrak{R}_\beta^\alpha \in W^{m,p}(T)$ for $p \in [1, 2)$ if and only if $m - 1 \leq |\alpha| - \|\alpha + \beta\|_\infty$,
- (ii) $\mathfrak{R}_\beta^\alpha \in W^{m,p}(T)$ for $p \in [2, \infty]$ if and only if $m \leq |\alpha| - \|\alpha + \beta\|_\infty$.

Proof. Let us start proving the case for $m = 0$.

Let $p \in [1, \infty)$. Note that the singularities occur only in vertices and hence it suffices to verify integrability in the neighborhoods of vertices. By symmetry it suffices to consider the vertex v_0 which is without loss of generality at the origin $v_0 = 0$. Since the terms λ_0 , $(1 - \lambda_1)$, and $(1 - \lambda_2)$ are equivalent to a constant in a neighborhood of the origin we obtain for small values $\varepsilon > 0$

$$\begin{aligned} \int_{\{x \in T: |x| \leq \varepsilon\}} \left(\frac{\lambda^\alpha}{(1 - \lambda)^\beta} \right)^p dx &\approx \int_{\{x \in T: |x| \leq \varepsilon\}} \left(\frac{\lambda_1^{\alpha_1} \lambda_2^{\alpha_2}}{(1 - \lambda_0)^{\beta_0}} \right)^p dx \\ &\approx \int_0^\varepsilon r^{p(\alpha_1 + \alpha_2 - \beta_0)} r dr. \end{aligned}$$

The integral is finite if and only if $p(\alpha_1 + \alpha_2 - \beta_0) + 1 > -1$. This condition is equivalent to $\alpha_1 + \alpha_2 - \beta_0 > -2/p$ or in other words $|\alpha| - \alpha_0 - \beta_0 > -2/p$. By symmetry in fact we require $|\alpha| - \|\alpha + \beta\|_\infty > -2/p$, which proves the claim. Similarly, for $p = \infty$, in the neighborhood of $v_0 = 0$, we have that

$$\frac{\lambda^\alpha}{(1 - \lambda)^\beta} \approx \frac{\lambda_1^{\alpha_1} \lambda_2^{\alpha_2}}{(1 - \lambda_0)^{\beta_0}} \lesssim r^{\alpha_1 + \alpha_2 - \beta_0}$$

is bounded, if $\alpha_1 + \alpha_2 - \beta_0 \geq 0$. This is equivalent to $|\alpha| - \alpha_0 - \beta_0 \geq 0$, which proves the claim for $p = \infty$. This estimate is sharp if we consider a path from the barycenter to the vertex $v_0 = 0$.

We proceed by induction over m . For $j = 0, 1, 2$ the identity (3.6) states

$$\partial_{\lambda_j} \mathfrak{R}_\beta^\alpha = \alpha_j \mathfrak{R}_{\beta - e_j}^{\alpha - e_j} - \beta_j \mathfrak{R}_{\beta + e_j}^\alpha. \quad (3.7)$$

The terms vanish for $\alpha_j = 0$ or $\beta_j = 0$, respectively. Otherwise, for the corresponding indices we obtain

$$\begin{aligned} |\alpha - e_j| - \|\alpha - e_j + \beta\|_\infty &\geq |\alpha - e_j| - \|\alpha + \beta\|_\infty = |\alpha| - \|\alpha + \beta\|_\infty - 1, \\ |\alpha| - \|\alpha + \beta + e_j\|_\infty &\geq |\alpha| - \|\alpha + \beta\|_\infty - 1. \end{aligned}$$

In both cases $p \in [1, \infty)$ and $p = \infty$ we have $\mathfrak{R}_\beta^{\alpha - e_j}, \mathfrak{R}_{\beta + e_j}^\alpha \in W^{m-1,p}(T)$ for $p \in [1, \infty]$ by induction hypothesis and the formula in (3.3). Hence, $\mathfrak{R}_\beta^\alpha \in W^{m,p}(T)$ follows by (3.3) and (3.7). The sharpness follows from the fact that the singularities of $\mathfrak{R}_\beta^{\alpha - e_j}$ and $\mathfrak{R}_{\beta + e_j}^\alpha$ are linearly independent and hence cannot cancel.

The statements in (i) and (ii) are an immediate consequence. \square

3.2. Integration formula. We are now in the position to derive recursive formulas for the (mean) integral of the rational functions $\mathfrak{R}_\beta^\alpha$. Recall, that $T \subset \mathbb{R}^2$ is a triangle. We define for $\alpha, \beta \in \mathbb{N}_0^3$

$$\mathcal{I}(\alpha, \beta) := \int_T \mathfrak{R}_\beta^\alpha(\lambda(x)) dx = \int_T \frac{\lambda^\alpha}{(1 - \lambda)^\beta} dx. \quad (3.8)$$

Note that by affine transformation this definition is independent of T , so without loss of generality we may use

$$T = [(0, 0)^\top, (1, 0)^\top, (0, 1)^\top]. \quad (3.9)$$

By Lemma 3.2 we have

$$\mathcal{I}(\alpha, \beta) < \infty \quad \text{if and only if} \quad \|\alpha + \beta\|_\infty \leq |\alpha| + 1. \quad (3.10)$$

3.2.1. *Special case* $\beta_0 = \beta_1 = 0$. The following lemma determines the value $\mathcal{I}(\alpha, 0)$, i.e., the special case of polynomial integrand, as

$$\mathcal{I}(\alpha, 0) = 2 \frac{\alpha!}{(2 + |\alpha|)!} = 2 \frac{\alpha_0! \alpha_1! \alpha_2!}{(2 + \alpha_0 + \alpha_1 + \alpha_2)!}. \quad (3.11)$$

Lemma 3.3 (Polynomials). *For any d -simplex T with $d \in \mathbb{N}$ and multi-index $\alpha = (\alpha_0, \dots, \alpha_d) \in \mathbb{N}_0^{d+1}$, one has the identity*

$$\int_T \lambda^\alpha dx = \frac{d! \alpha!}{(d + |\alpha|)!} = \frac{d! \alpha_0! \alpha_1! \alpha_2! \dots \alpha_d!}{(d + \alpha_0 + \alpha_1 + \alpha_2 + \dots + \alpha_d)!}.$$

Proof. This known identity is for example presented in [Str71, p. 222] for $\alpha_0 = 0$. The general case with proof can be found in [GM78, Eq. 2.3]. For the convenience of the reader we include a short alternative proof for polynomials on a d -simplex, for dimension $d \in \mathbb{N}$, and $\alpha \in \mathbb{N}_0^{d+1}$. We also apply the same technique later in Proposition 3.9. Let T be as in (3.9). Using the Gamma function $\Gamma(s) = \int_0^\infty t^{s-1} \exp(-t) dt$, which satisfies $\Gamma(s) = (s-1)!$ for $s \in \mathbb{N}$, we obtain

$$\begin{aligned} & \Gamma(d + |\alpha| + 1) \int_T \lambda(x)^\alpha dx \\ &= d! \int_0^\infty \int_T (1 - x_1 - x_2 - \dots - x_d)^{\alpha_0} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_d^{\alpha_d} t^{d+|\alpha|} \exp(-t) dx dt. \end{aligned}$$

The substitution $(y_0, y_1, \dots, y_d) = t(1 - x_1 - \dots - x_d, x_1, \dots, x_d)$ leads to

$$\begin{aligned} & \Gamma(d + |\alpha| + 1) \int_T \lambda(x)^\alpha dx \\ &= d! \int_0^\infty \dots \int_0^\infty y_0^{\alpha_0} \dots y_d^{\alpha_d} \exp(-y_0 - \dots - y_d) dy_0 dy_1 \dots dy_d \\ &= d! \prod_{j=0}^d \int_0^\infty z^{\alpha_j} \exp(-z) dz = d! \prod_{j=0}^d \Gamma(\alpha_j + 1) = d! \alpha!. \end{aligned}$$

This proves the claim. \square

The following lemma reveals another special case with explicit value of $\mathcal{I}(\alpha, \beta)$ generalizing the previous lemma with $\beta = (0, 0, \beta_2)$ for some $\beta_2 \in \mathbb{N}_0$ and $\alpha \in \mathbb{N}_0^3$.

Lemma 3.4. *For any triangle T , for multi-indices $\alpha = (\alpha_0, \alpha_1, \alpha_2) \in \mathbb{N}_0^3$, and $\beta = (0, 0, \beta_2)$ with $\beta_2 \in \mathbb{N}_0$ such that $\|\alpha + \beta\|_\infty \leq |\alpha| + 1$, one has the identity*

$$\int_T \frac{\lambda^\alpha}{(1 - \lambda)^\beta} dx = 2 \frac{\alpha_0! \alpha_1! \alpha_2!}{(|\alpha| - \beta_2 + 2)!} \frac{(\alpha_0 + \alpha_1 + 1 - \beta_2)!}{(\alpha_0 + \alpha_1 + 1)!}. \quad (3.12)$$

Proof. We proceed similarly as in Lemma 3.3 using the Gamma function $\Gamma(s) = \int_0^\infty t^{s-1} \exp(-t) dt$ with $\Gamma(s+1) = s!$ for $s \in \mathbb{N}_0$. Let T be as in (3.9). By the

assumption $\|\alpha + \beta\|_\infty \leq |\alpha| + 1$ the integral is finite, see Lemma 3.2. With this and the fact that $\beta_1 = 0$ it follows that

$$2 + |\alpha| - \beta_2 + 1 \geq 1 + \|\alpha + \beta\|_\infty - \beta_2 + 1 \geq 2.$$

For this reason $\Gamma(2 + |\alpha| - \beta_2 + 1)$ is well-defined. Employing the substitution $(y_0, y_1, y_2) = t(1 - x_1 - x_2, x_1, x_2)$ we obtain

$$\begin{aligned} \text{I} &:= \Gamma(2 + |\alpha| - \beta_2 + 1) \int_T \frac{\lambda(x)^\alpha}{(1 - \lambda_2(x))^{\beta_2}} dx \\ &= 2 \int_0^\infty \int_T \frac{(1 - x_1 - x_2)^{\alpha_0} x_1^{\alpha_1} x_2^{\alpha_2}}{(1 - x_2)^{\beta_2}} t^{2+|\alpha|-\beta_2} \exp(-t) dx dt \\ &= 2 \int_0^\infty \int_0^\infty \int_0^\infty \frac{y_0^{\alpha_0} y_1^{\alpha_1} y_2^{\alpha_2}}{(y_0 + y_1)^{\beta_2}} \exp(-y_0 - y_1 - y_2) dy_0 dy_1 dy_2 \\ &= 2\alpha_2! \int_0^\infty \int_0^\infty \frac{y_0^{\alpha_0} y_1^{\alpha_1}}{(y_0 + y_1)^{\beta_2}} \exp(-y_0 - y_1) dy_0 dy_1. \end{aligned}$$

We substitute $s = y_0 + y_1$, and then $z = y_0/s$ and use Lemma 3.3 for $n = 1$ to find

$$\begin{aligned} \text{I} &= 2\alpha_2! \int_0^\infty \int_0^s \frac{y_0^{\alpha_0} (s - y_0)^{\alpha_1}}{s^{\beta_2}} \exp(-s) dy_0 ds \\ &= 2\alpha_2! \int_0^\infty \int_0^1 z^{\alpha_0} (1 - z)^{\alpha_1} dz s^{\alpha_0 + \alpha_1 - \beta_2 + 1} \exp(-s) ds \\ &= 2\alpha_2! \frac{\alpha_0! \alpha_1!}{(\alpha_0 + \alpha_1 + 1)!} (\alpha_0 + \alpha_1 + 1 - \beta_2)!. \end{aligned}$$

With $\Gamma(s) = (s - 1)!$ for $s \in \mathbb{N}$ this proves the claim. \square

3.2.2. *Special case $\alpha_0 = \beta_0 = 0$.* In case that $\alpha_0 = \beta_0 = 0$ the integral $\mathcal{I}(\alpha, \beta)$ can be split by Fubini's theorem, leading to

$$\begin{aligned} \mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2) &:= \mathcal{I}((0, \alpha_1, \alpha_2), (0, \beta_1, \beta_2)) \\ &= 2 \int_0^1 \frac{y^{\alpha_2}}{(1 - y)^{\beta_2}} \int_0^{1-y} \frac{x^{\alpha_1}}{(1 - x)^{\beta_1}} dx dy. \end{aligned} \quad (3.13)$$

To evaluate $\mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2)$ we denote the inner integral by

$$\mathcal{J}^x(\alpha_1, \beta_1, y) := \int_0^{1-y} \frac{x^{\alpha_1}}{(1 - x)^{\beta_1}} dx \quad \text{for } y \in (0, 1). \quad (3.14)$$

Hence, we have the identity

$$\mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2) = 2 \int_0^1 \frac{y^{\alpha_2}}{(1 - y)^{\beta_2}} \mathcal{J}^x(\alpha_1, \beta_1, y) dy. \quad (3.15)$$

Lemma 3.5. *We have for all $\alpha_1, \beta_1 \in \mathbb{N}_0$ and $y \in (0, 1)$*

$$\mathcal{J}^x(\alpha_1, \beta_1, y) = \begin{cases} \frac{(1 - y)^{\alpha_1 + 1}}{\alpha_1 + 1} & \text{if } \beta_1 = 0, \\ -\sum_{j=1}^{\alpha_1} \frac{1}{j} (1 - y)^j - \log(y) & \text{if } \beta_1 = 1, \\ \frac{\beta_1 - \alpha_1 - 2}{\beta_1 - 1} \mathcal{J}^x(\alpha_1, \beta_1 - 1, y) + \frac{1}{\beta_1 - 1} \frac{(1 - y)^{\alpha_1 + 1}}{y^{\beta_1 - 1}} & \text{if } \beta_1 > 1. \end{cases}$$

Proof. The first identity follows from the integration of polynomials in 1D. We continue with $\beta_1 = 1$. In case $\alpha_1 = 0$ we have

$$\mathcal{J}^x(0, 1, y) = - \int_0^{1-y} \frac{1}{x-1} dx = -\log(y).$$

For $\alpha_1 > 0$ the formula for geometric sums leads to

$$\begin{aligned} \mathcal{J}^x(0, 1, y) &= \int_0^{1-y} \frac{x^{\alpha_1}}{1-x} dx = - \int_0^{1-y} \frac{x^{\alpha_1} - 1}{x-1} dx - \int_0^{1-y} \frac{1}{x-1} dx \\ &= - \sum_{j=0}^{\alpha_1-1} \int_0^{1-y} x^j dx - \int_0^{1-y} \frac{1}{x-1} dx = - \sum_{j=1}^{\alpha_1} \frac{1}{j} (1-y)^j - \log(y). \end{aligned}$$

Finally, let $\beta_1 > 1$. We use the identity

$$\begin{aligned} \mathcal{J}^x(\alpha_1, \beta_1, y) &= \int_0^{1-y} \frac{x^{\alpha_1}}{(1-x)^{\beta_1}} dx = \int_0^{1-y} \frac{x^{\alpha_1}}{(1-x)^{\beta_1-1}} dx + \int_0^{1-y} \frac{x^{\alpha_1+1}}{(1-x)^{\beta_1}} dx \\ &= \mathcal{J}^x(\alpha_1, \beta_1 - 1, y) + \mathcal{J}^x(\alpha_1 + 1, \beta_1, y). \end{aligned}$$

Integration by parts yields for the second term

$$\begin{aligned} \mathcal{J}^x(\alpha_1 + 1, \beta_1, y) &= \int_0^{1-y} \frac{x^{\alpha_1+1}}{(1-x)^{\beta_1}} dx \\ &= \left[\frac{x^{\alpha_1+1}(1-x)^{1-\beta_1}}{\beta_1 - 1} \right]_0^{1-y} - \int_0^{1-y} \frac{\alpha_1 + 1}{\beta_1 - 1} \frac{x^{\alpha_1}}{(1-x)^{\beta_1-1}} dx \\ &= \frac{(1-y)^{\alpha_1+1} y^{1-\beta_1}}{\beta_1 - 1} - \frac{\alpha_1 + 1}{\beta_1 - 1} \mathcal{J}^x(\alpha_1, \beta_1 - 1, y). \end{aligned}$$

Combining these identities concludes the proof. \square

Since expressions for \mathcal{J}^x are available by Lemma 3.5, certain integrals in y still have to be computed in order to evaluate the integral \mathcal{J} in (3.15).

Lemma 3.6. *Let $\alpha_2, \gamma \in \mathbb{N}_0$, then we have with polygamma function $\psi^{(1)}$ that*

$$\int_0^1 y^{\alpha_2} (1-y)^\gamma dy = \frac{\alpha_2! \gamma!}{(\alpha_2 + \gamma + 1)!}, \quad (3.16a)$$

$$\int_0^1 y^{\alpha_2} \log(y) dy = -\frac{1}{(\alpha_2 + 1)^2}, \quad (3.16b)$$

$$\int_0^1 \frac{y^{\alpha_2}}{1-y} \log(y) dy = -\psi^{(1)}(\alpha_2 + 1) = -\frac{\pi^2}{6} + \sum_{j=1}^{\alpha_2} \frac{1}{j^2}. \quad (3.16c)$$

Proof. The formula in (3.16a) is the 1D version of Lemma 3.3. Integration by parts leads to (3.16b), that is,

$$\int_0^1 y^{\alpha_2} \log y dy = \left[\frac{y^{\alpha_2+1}}{\alpha_2 + 1} \log y \right]_0^1 - \int_0^1 \frac{y^{\alpha_2}}{\alpha_2 + 1} dy = -\frac{1}{(\alpha_2 + 1)^2}.$$

A substitution with $y = \exp(-t)$ and the properties of the polygamma function [AS64, eq. 6.4.1 and 6.4.10] yield

$$\int_0^1 \frac{y^{\alpha_2} \log(y)}{y-1} dy = - \int_0^\infty \frac{t \exp(-\alpha_2 t)}{1 - e^{-t}} dt = -\psi^{(1)}(\alpha_2 + 1) = \frac{\pi^2}{6} - \sum_{j=1}^{\alpha_2} \frac{1}{j^2}. \quad \square$$

Combining Lemmas 3.5 and 3.6 allows us to evaluate \mathcal{J} . This is realized in Algorithm 1, which will be verified in Theorem 3.7 below.

Algorithm 1: Computation of $\mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2)$

```

ComputeJ ( $\alpha_1, \alpha_2, \beta_1, \beta_2$ )
Input: Indices  $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathbb{N}_0$ 
Output: Integral mean  $\mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2)$ 
if  $\max\{\alpha_1 + \beta_1, \alpha_2 + \beta_2\} > \alpha_1 + \alpha_2 + 1$  then
  | return  $\infty$ 
Sort  $((\alpha_j, \beta_j))_{j=1,2}$  such that  $\beta_2 \geq \beta_1$ 
if  $\beta_1 = 0$  then
  | return  $\frac{2}{\alpha_1 + 1} \frac{\alpha_2!(\alpha_1 - \beta_2 + 1)!}{(\alpha_1 + \alpha_2 - \beta_2 + 2)!}$ 
if  $\beta_1 = 1$  then
  | if  $\beta_2 = 1$  then
  | | return  $-2 \sum_{i=1}^{\alpha_2} \frac{1}{i^2} + \frac{\pi^2}{3} - 2 \sum_{j=1}^{\alpha_1} \frac{1}{j} \frac{\alpha_2!(j-1)!}{(\alpha_2 + j)!}$ 
  | else
  | | return  $\frac{\beta_2 - \alpha_2 - 2}{\beta_2 - 1} \text{ComputeJ}(\alpha_1, \alpha_2, 1, \beta_2 - 1) + \frac{2}{\beta_2 - 1} \frac{(\alpha_1 - \beta_2 + 1)! \alpha_2!}{(\alpha_1 - \beta_2 + \alpha_2 + 2)!}$ 
else
  | return  $\frac{\beta_1 - \alpha_1 - 2}{\beta_1 - 1} \text{ComputeJ}(\alpha_1, \alpha_2, \beta_1 - 1, \beta_2) + \frac{2}{\beta_1 - 1} \frac{(\alpha_2 - \beta_1 + 1)! (\alpha_1 - \beta_2 + 1)!}{(\alpha_2 - \beta_1 + \alpha_1 - \beta_2 + 3)!}$ 

```

Theorem 3.7 (Computation of \mathcal{J} in Algorithm 1). *Let $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathbb{N}_0$. Then Algorithm 1 terminates and returns $\mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2)$.*

Proof. Due to Lemma 3.2 we have $\mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2) = \infty$ if and only if we have $\max\{\alpha_1 + \beta_1, \alpha_2 + \beta_2\} > \alpha_1 + \alpha_2 + 1$. Let us consider the remaining cases. Since $\mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2) = \mathcal{J}(\alpha_2, \alpha_1, \beta_2, \beta_1)$ we can assume that $\beta_2 \geq \beta_1$. Recall

$$\mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2) = 2 \int_0^1 \frac{y^{\alpha_2}}{(1-y)^{\beta_2}} \mathcal{J}^x(\alpha_1, \beta_1, y) dy.$$

The values for \mathcal{J}^x are available by Lemma 3.5 and the integrals in y are computed in Lemma 3.6. Hence, one has

$$\begin{aligned} \mathcal{J}(\alpha_1, \alpha_2, 0, \beta_2) &= \frac{2}{\alpha_1 + 1} \int_0^1 \frac{y^{\alpha_2}}{(1-y)^{\beta_2}} (1-y)^{\alpha_1 + 1} dy \\ &= \frac{2}{\alpha_1 + 1} \frac{\alpha_2!(\alpha_1 + 1 - \beta_2)!}{(\alpha_2 + \alpha_1 - \beta_2 + 2)!}. \end{aligned}$$

This verifies Algorithm 1 for $\beta_2 \geq \beta_1 = 0$. We continue with $\beta_2 \geq \beta_1 \geq 1$. We calculate (3.16c) which yields

$$\begin{aligned} \mathcal{J}(\alpha_1, \alpha_2, 1, 1) &= -2 \int_0^1 \frac{y^{\alpha_2}}{1-y} \log(y) dy - 2 \sum_{j=1}^{\alpha_1} \frac{1}{j} \int_0^1 \frac{y^{\alpha_2}}{1-y} (1-y)^j dy \\ &= -2 \sum_{i=1}^{\alpha_2} \frac{1}{i^2} + \frac{\pi^2}{3} - 2 \sum_{j=1}^{\alpha_1} \frac{1}{j} \frac{\alpha_2!(j-1)!}{(\alpha_2 + j)!}. \end{aligned}$$

This verifies Algorithm 1 for $\beta_1 = \beta_2 = 1$.

To cover the remaining cases, we exploit Lemma 3.5 to conclude for $\beta_1 > 1$ that

$$\begin{aligned} \mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2) &= 2 \frac{\beta_1 - \alpha_1 - 2}{\beta_1 - 1} \int_0^1 \frac{y^{\alpha_2}}{(1-y)^{\beta_2}} \mathcal{J}^x(\alpha_1, \beta_1 - 1, y) dy \\ &\quad + \frac{2}{\beta_2 - 1} \int_0^1 \frac{y^{\alpha_2}}{(1-y)^{\beta_2}} \frac{(1-y)^{\alpha_1+1}}{y^{\beta_1-1}} dy \\ &= \frac{\beta_1 - \alpha_1 - 2}{\beta_1 - 1} \mathcal{J}(\alpha_1, \alpha_2, \beta_1 - 1, \beta_2) \\ &\quad + \frac{2}{\beta_1 - 1} \frac{(\alpha_2 - \beta_1 + 1)! (\alpha_1 - \beta_2 + 1)!}{(\alpha_2 - \beta_1 + \alpha_1 - \beta_2 + 3)!}. \end{aligned}$$

By symmetry, we obtain for $\beta_2 > 1$ the identity

$$\begin{aligned} \mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2) &= \frac{\beta_2 - \alpha_2 - 2}{\beta_2 - 1} \mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2 - 1) \\ &\quad + \frac{2}{\beta_2 - 1} \frac{(\alpha_1 - \beta_2 + 1)! (\alpha_2 - \beta_1 + 1)!}{(\alpha_1 - \beta_2 + \alpha_2 - \beta_1 + 3)!}. \end{aligned}$$

This yields in particular the formula for $\beta_1 = 1 < \beta_2$ and $1 < \beta_1 \leq \beta_2$ in the algorithm. Since in both cases the sum $\beta_1 + \beta_2$ is reduced by one, the recursive algorithm reaches after a finite number of steps the case $\beta_1 = 1 = \beta_2$ and terminates. \square

3.2.3. General case. In this subsection we introduce a recursive formula that successively reduces the value of $|\beta|$ or α_0 until the integrand $\mathcal{I}(\alpha, \beta)$ defined in (3.8) matches one of the special cases outlined in (3.11), in (3.12) and in Section 3.2.1. The reduction relies on the following recursive relations, which use the basis vectors $\hat{v}_0 = (1, 0, 0)$, $\hat{v}_1 = (0, 1, 0)$, and $\hat{v}_2 = (0, 0, 1)$.

Lemma 3.8 (Recursion formulae). *For any $\alpha, \beta \in \mathbb{N}_0^3$ we have the following identities for rational polynomials as defined in (3.1):*

- (a) $\mathfrak{R}_\beta^\alpha = \frac{1}{2}(\mathfrak{R}_{\beta-\hat{v}_0}^\alpha + \mathfrak{R}_{\beta-\hat{v}_1}^\alpha + \mathfrak{R}_{\beta-\hat{v}_2}^\alpha)$,
- (b) $\mathfrak{R}_\beta^\alpha = \mathfrak{R}_{\beta-\hat{v}_2}^{\alpha-\hat{v}_0} - \mathfrak{R}_\beta^{\alpha-\hat{v}_0+\hat{v}_1}$,
- (c) $\mathfrak{R}_\beta^\alpha = \frac{1}{2}(\mathfrak{R}_{\beta-\hat{v}_1}^\alpha + \mathfrak{R}_{\beta-\hat{v}_2}^\alpha + \mathfrak{R}_{\beta-\hat{v}_1}^{\alpha-\hat{v}_0+\hat{v}_1} + \mathfrak{R}_{\beta-\hat{v}_2}^{\alpha-\hat{v}_0+\hat{v}_2}) - \mathfrak{R}_\beta^{\alpha-\hat{v}_0+\hat{v}_1+\hat{v}_2}$.

Proof. The proof of (a) follows from

$$(1-\lambda)^{\hat{v}_0} + (1-\lambda)^{\hat{v}_1} + (1-\lambda)^{\hat{v}_2} = (1-\lambda_0) + (1-\lambda_1) + (1-\lambda_2) = 3 - 1 = 2.$$

The claim in (b) follows from the identity

$$\frac{\lambda^\alpha}{(1-\lambda)^\beta} = \frac{\lambda^{\alpha-\hat{v}_0}(1-\lambda_1-\lambda_2)}{(1-\lambda)^\beta} = \frac{\lambda^{\alpha-\hat{v}_0}}{(1-\lambda)^{\beta-\hat{v}_1}} - \frac{\lambda^{\alpha-\hat{v}_0+\hat{v}_2}}{(1-\lambda)^\beta}.$$

Finally, the proof of (c) is a consequence of

$$\begin{aligned} &\frac{1}{2} \left(\frac{\lambda^\alpha}{(1-\lambda)^{\beta-\hat{v}_1}} + \frac{\lambda^\alpha}{(1-\lambda)^{\beta-\hat{v}_2}} + \frac{\lambda^{\alpha-\hat{v}_0+\hat{v}_1}}{(1-\lambda)^{\beta-\hat{v}_1}} + \frac{\lambda^{\alpha-\hat{v}_0+\hat{v}_2}}{(1-\lambda)^{\beta-\hat{v}_2}} \right) - \frac{\lambda^{\alpha-\hat{v}_0+\hat{v}_1+\hat{v}_2}}{(1-\lambda)^\beta} \\ &= \frac{\lambda^{\alpha-\hat{v}_0}}{2(1-\lambda)^\beta} (\lambda_0(1-\lambda_1) + \lambda_0(1-\lambda_2) + \lambda_1(1-\lambda_1) + \lambda_2(1-\lambda_2) - 2\lambda_1\lambda_2) \\ &= \frac{\lambda^{\alpha-\hat{v}_0}}{2(1-\lambda)^\beta} 2\lambda_0 = \frac{\lambda^\alpha}{(1-\lambda)^\beta}. \end{aligned} \quad \square$$

Those recursion relations allow us to derive recursion relations for $\mathcal{I}(\alpha, \beta)$.

Proposition 3.9 (Reduction). *Let $T \subset \mathbb{R}^2$ be a triangle and let $\alpha, \beta \in \mathbb{N}_0^3$ be multi-indices such that $\mathcal{I}(\alpha, \beta) < \infty$.*

(a) *For any permutation σ of $\{0, 1, 2\}$ we have*

$$\mathcal{I}(\alpha, \beta) = \mathcal{I}(\sigma(\alpha), \sigma(\beta)).$$

(b) *If $\beta_0, \beta_1, \beta_2 \geq 1$, one has*

$$\mathcal{I}(\alpha, \beta) = \frac{1}{2}(\mathcal{I}(\alpha, \beta - \hat{v}_0) + \mathcal{I}(\alpha, \beta - \hat{v}_1) + \mathcal{I}(\alpha, \beta - \hat{v}_2)).$$

(c) *If $\beta_0 = 0$ and $\alpha_0 \geq 1$, and let $\alpha_j + \beta_j < |\alpha| + 1$ for some $j = 1, 2$, then*

$$\mathcal{I}(\alpha, \beta) = \mathcal{I}(\alpha - \hat{v}_0, \beta - \hat{v}_{3-j}) - \mathcal{I}(\alpha - \hat{v}_0 + \hat{v}_j, \beta).$$

(d) *If $\beta_0 = 0$, $\beta_1, \beta_2 \geq 1$ and $\alpha_0 > 0$, one has*

$$\begin{aligned} \mathcal{I}(\alpha, \beta) &= \frac{1}{2}(\mathcal{I}(\alpha, \beta - \hat{v}_1) + \mathcal{I}(\alpha, \beta - \hat{v}_2)) - \mathcal{I}(\alpha - \hat{v}_0 + \hat{v}_1 + \hat{v}_2, \beta) \\ &\quad + \frac{1}{2}(\mathcal{I}(\alpha - \hat{v}_0 + \hat{v}_1, \beta - \hat{v}_1) + \mathcal{I}(\alpha - \hat{v}_0 + \hat{v}_2, \beta - \hat{v}_2)). \end{aligned}$$

Proof. Recall that $\mathcal{I}(\alpha, \beta) < \infty$ holds if and only if $\|\alpha + \beta\|_\infty \leq |\alpha| + 1$, see Lemma 3.2. The proof of (a) follows by symmetry.

To prove (b) note that with $\beta_0, \beta_1, \beta_2 \geq 1$ it follows that for any $j \in \{0, 1, 2\}$ one has $\|\alpha + \beta - \hat{v}_j\|_\infty \leq \|\alpha + \beta\|_\infty \leq |\alpha| + 1$. Consequently, all terms are finite. Then the claim follows from Lemma 3.8 (a).

By symmetry, we assume that the index satisfying the assumptions of (c) is $j = 2$, that is, $\alpha_2 + \beta_2 < |\alpha| + 1$. We have $(\alpha - \hat{v}_0 + \hat{v}_2)_2 + \beta_2 = \alpha_2 + \beta_2 + 1 \leq |\alpha| + 1$. Thus, Lemma 3.2 shows that the integrals $\mathcal{I}(\alpha - \hat{v}_0, \beta - \hat{v}_1)$ and $\mathcal{I}(\alpha - \hat{v}_0 + \hat{v}_2, \beta)$ are finite. Then the identity follows from Lemma 3.8 (b).

Similarly, under the conditions on α, β as specified in (d) one can show that all terms are finite and the identity follows from Lemma 3.8 (c). \square

Algorithm 2: Computation of $\mathcal{I}(\alpha, \beta)$

ComputeI (α, β)

Input: Multi-indices $\alpha, \beta \in \mathbb{N}_0^3$ with $\alpha = (\alpha_0, \alpha_1, \alpha_2)$ and $\beta = (\beta_0, \beta_1, \beta_2)$

Output: Integral mean $\mathcal{I}(\alpha, \beta)$

if $\max\{\alpha_0 + \beta_0, \alpha_1 + \beta_1, \alpha_2 + \beta_2\} > \alpha_0 + \alpha_1 + \alpha_2 + 1$ **then**
 | **return** ∞

Sort $((\alpha_j, \beta_j))_{j=1,2}$ such that $\beta_0 \leq \beta_1 \leq \beta_2$

if $\beta_0 = \beta_1 = 0$ **then**

 | **return** $2 \frac{\alpha_0! \alpha_1! \alpha_2!}{(|\alpha| - \beta_2 + 2)!} \frac{(\alpha_0 + \alpha_1 + 1 - \beta_2)!}{(\alpha_0 + \alpha_1 + 1)!}$

if $\beta_0 \geq 1$ **then** // $\beta \geq (1, 1, 1)$

 | **return** $\frac{1}{2} \sum_{j=0}^2 \text{ComputeI}(\alpha, \beta - \hat{v}_j)$

if $\alpha_0 = 0$ **then** // $\beta_0 = 0, \beta_2 \geq \beta_1 \geq 1$

 | **return** $\text{ComputeJ}(\alpha_1, \alpha_2, \beta_1, \beta_2)$

if $\alpha_1 + \beta_1 < |\alpha| + 1$ **then** // $\beta_0 = 0, \beta_2 \geq \beta_1 \geq 1, \alpha_0 \geq 1$

 | **return** $\text{ComputeI}(\alpha - \hat{v}_0, \beta - \hat{v}_2) - \text{ComputeI}(\alpha - \hat{v}_0 + \hat{v}_1, \beta)$

if $\alpha_2 + \beta_2 < |\alpha| + 1$ **then**

 | **return** $\text{ComputeI}(\alpha - \hat{v}_0, \beta - \hat{v}_1) - \text{ComputeI}(\alpha - \hat{v}_0 + \hat{v}_2, \beta)$

else

 | **return** $\frac{1}{2} \sum_{j=1}^2 (\text{ComputeI}(\alpha, \beta - \hat{v}_j) + \text{ComputeI}(\alpha - \hat{v}_0 + \hat{v}_j, \beta - \hat{v}_j))$
 | $\quad - \text{ComputeI}(\alpha - \hat{v}_0 + \hat{v}_1 + \hat{v}_2, \beta)$

Theorem 3.10 (Computation of \mathcal{I} in Algorithm 2). *Let $\alpha, \beta \in \mathbb{N}_0^3$. Then Algorithm 2 terminates and returns $\mathcal{I}(\alpha, \beta)$.*

Proof. The first condition determines whether the integral is finite as described in Lemma 3.2. The sorting of the (α_j, β_j) guarantees that $\beta_0 \leq \beta_1 \leq \beta_2$ using Proposition 3.9 (a). In case $\beta_0 = \beta_1 = 0$ we use the explicit formula as in Lemma 3.4.

If $\beta \geq (1, 1, 1)$, the routine exploits Proposition 3.9 (b) to reduce some β_j . After that the remaining cases have $\beta_0 = 0$ and $1 \leq \beta_1 \leq \beta_2$. If also $\alpha_0 = 0$, then $\mathcal{I}(\alpha, \beta) = \mathcal{J}(\alpha_1, \alpha_2, \beta_1, \beta_2)$ and we can use Algorithm 1 to compute its value. In the remaining cases we have additionally to the conditions on β that $\alpha_0 \geq 1$. If either $\alpha_1 + \beta_1 < |\alpha| + 1$ or $\alpha_2 + \beta_2 < |\alpha| + 1$, then the three term recursion in Proposition 3.9 (c) can be applied to reduce α_0 , and on one term also one of the entries of β . Now, the only remaining case is $\beta_0 = 0, 1 \leq \beta_1 \leq \beta_2, \alpha_0 \geq 0$, and $\alpha_1 + \beta_1 = \alpha_2 + \beta_2 = |\alpha| + 1$. In this case Proposition 3.9 (d) is applicable, and in each term α_0 or $|\beta|$ is reduced. Since α_0 and $|\beta|$ are non-increasing in the iteration, and in each step α_0 or $|\beta|$ is reduced, the algorithm terminates after finitely many iterations. \square

Remark 3.11 (Performance). *Algorithm 2 is recursive and is not optimized in terms of performance. It can be improved by storing the values of $\mathcal{I}(\alpha, \beta)$ that are computed in the iteration and using symmetries. Note that in the implementation of finite elements as in Section 4, Algorithm 2 is solely used in the offline phase.*

4. IMPLEMENTATION

In this section we describe our implementation of the finite elements discussed in Section 2 with MATLAB [Mat24], which is available at [DST24]. Our code uses the implementation of a uniform mesh refinement routine from [Bar15]. The implementation can easily be transferred to other programming languages. We discuss general finite elements using rational bubble functions in Section 4.1. Then we address the special cases of the singular Zienkiewicz element in Section 4.2 and the lowest-order Guzmán–Neilan element in Section 4.3.

Before we discuss our implementation, we explain the computation of the Jacobian $D\lambda = (\nabla\lambda)^\top$ needed for the computation of gradients in (3.3) and of Hessians in (3.4). Let $T = [v_0, v_1, v_2]$ be a simplex spanned by vertices $v_0, v_1, v_2 \in \mathbb{R}^2$ and let $T^{\text{ref}} = [v_0^{\text{ref}}, v_1^{\text{ref}}, v_2^{\text{ref}}]$ denote the reference simplex with vertices $v_0^{\text{ref}} = (0, 0)^\top$, $v_1^{\text{ref}} = (1, 0)^\top$, $v_2^{\text{ref}} = (0, 1)^\top$ in \mathbb{R}^2 . The affine transformation mapping $v_i^{\text{ref}} \mapsto v_i$ for all $i = 0, 1, 2$ is denoted by $F: T^{\text{ref}} \rightarrow T$. Its Jacobian matrix is given by

$$DF = (v_1 - v_0, \quad v_2 - v_0) = (\nabla F)^\top \in \mathbb{R}^{2 \times 2}.$$

The barycentric coordinates of T^{ref} for any $x = (x_1, x_2)^\top \in \mathbb{R}^2$ are given by

$$\lambda^{\text{ref}}(x) = \begin{pmatrix} \lambda_0^{\text{ref}}(x) \\ \lambda_1^{\text{ref}}(x) \\ \lambda_2^{\text{ref}}(x) \end{pmatrix} = \begin{pmatrix} 1 - x_1 - x_2 \\ x_1 \\ x_2 \end{pmatrix}. \quad (4.1)$$

The barycentric coordinates $\lambda = (\lambda_0, \lambda_1, \lambda_2)^\top$ on T satisfy $\lambda \circ F = \lambda^{\text{ref}}$. This identity and the chain rule lead with Jacobian matrices $D\lambda^{\text{ref}}, D\lambda \in \mathbb{R}^{3 \times 2}$ to

$$D\lambda^{\text{ref}} = D(\lambda \circ F) = D\lambda DF.$$

Hence, we can compute, as for example in Figure 4, line 6, the matrix

$$G_T := D\lambda = D\lambda^{\text{ref}}(DF)^{-\top} = \begin{pmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} (DF)^{-1} \in \mathbb{R}^{3 \times 2}. \quad (4.2)$$

Remark 4.1 (Alternative). *Lemma 3.10 in [Bar15] states the alternative formula*

$$D\lambda = \begin{pmatrix} 1 & 1 & 1 \\ v_0 & v_1 & v_2 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

4.1. Finite elements using rational functions. This subsection provides a general framework to implement finite element schemes that involve rational functions of the form (3.1).

4.1.1. The class ‘Rational Function’. The local bases of the finite elements under consideration are linear combinations of rational functions. That is with scalars $\gamma^{(j)} \in \mathbb{R}$ and multi-indices $\alpha^{(j)}, \beta^{(j)} \in \mathbb{N}_0^3$ for $j = 1, \dots, m \in \mathbb{N}$ expressed in barycentric coordinates they have the form

$$\hat{b} = \sum_{j=1}^m \gamma^{(j)} \hat{\mathfrak{R}}_{\beta^{(j)}}^{\alpha^{(j)}}. \quad (4.3)$$

Given a triangle T with barycentric coordinates $\lambda: T \rightarrow \hat{T}$ we obtain the rational function $b = \hat{b} \circ \lambda$ in standard coordinates. In MATLAB we store such functions as tensors, that is, we identify b with $\mathbf{b} \in \mathbb{R}^{3 \times 3 \times m}$, via

$$b \sim \mathbf{b} \in \mathbb{R}^{3 \times 3 \times m} \quad \text{with} \quad \mathbf{b}[:, :, j] = \begin{pmatrix} \alpha_0^{(j)} & \alpha_1^{(j)} & \alpha_2^{(j)} \\ \beta_0^{(j)} & \beta_1^{(j)} & \beta_2^{(j)} \\ \gamma^{(j)} & 0 & 0 \end{pmatrix} \quad \text{for } j = 1, \dots, m.$$

Based on the results in Section 3 we implement several routines for rational functions $b \sim \mathbf{b} \in \mathbb{R}^{3 \times 3 \times m}$ and $c \sim \mathbf{c} \in \mathbb{R}^{3 \times 3 \times n}$, for $m, n \in \mathbb{N}$, including the following:

- `Multiply_Rationals(b,c)` returns the rational function bc using (3.5),
- `Integrate_Rational(b)` returns $\int_T b dx$ using Algorithm 2,
- `Diff_Rational_lam(b)` returns the gradient $\nabla_\lambda \hat{b}$ using (3.6),
- `Diff_Rational_x(DF,nabla_b)` and `Diff_Rational_y(DF,nabla_b)` with `nabla_b = \nabla_\lambda \hat{b} and DF = DF return the derivatives $\partial_{x_1} b$ and $\partial_{x_2} b$ using (3.3), (4.2).`

4.1.2. Compute local system matrices. Let $(b_\ell)_{\ell=1}^L$ with $L \in \mathbb{N}$ denote a set of rational basis functions of a local finite element space $\mathcal{P}(T)$ on a simplex $T \in \mathcal{T}$. Our routine computes local system matrices such as the mass matrix M and the stiffness matrix A , defined by

$$M = (M_{\ell,k})_{\ell,k=1}^L \quad \text{with} \quad M_{\ell,k} := \int_T b_\ell b_k dx,$$

$$A = (A_{\ell,k})_{\ell,k=1}^L \quad \text{with} \quad A_{\ell,k} := \int_T \nabla b_\ell \cdot \nabla b_k dx.$$

We can compute these local matrices directly by the routines mentioned in Section 4.1.1, but this approach appears to be rather slow. Instead, the integrals are best calculated on a reference simplex using the routines in Section 4.1.1 and then

transformed to T as exemplified in Section 4.2 and 4.3 below. Notice that the reference simplex is $\widehat{T} \subset \mathbb{R}^3$ defined in (2.1), that is, we compute quantities with respect to the barycentric coordinates. This distinguishes our ansatz from typical finite element implementations as for example in [Bra07, Chap. 2.8], where the reference element is $T^{\text{ref}} \subset \mathbb{R}^2$.

4.1.3. *Evaluation of the right-hand side.* Let $f: \Omega \rightarrow \mathbb{R}$ be a function usually representing the right-hand side of a partial differential equation, and let $T \in \mathcal{T}$ be a simplex. Let $(\varphi_j)_{j=1}^J \subset \mathcal{P}_r(T)$ denote a basis of the polynomial space $\mathcal{P}_r(T)$ for some $r \in \mathbb{N}_0$ and recall the local basis functions $(b_\ell)_{\ell=1}^L$ mentioned in Section 4.1.2. Let $\varphi_j = \widehat{\varphi}_j \circ \lambda$ and $b_\ell = \widehat{b}_\ell \circ \lambda$ and we define the affinely transformed basis on the reference triangle by $\varphi_j^{\text{ref}} = \widehat{\varphi}_j \circ \lambda^{\text{ref}}$ and $b_\ell^{\text{ref}} = \widehat{b}_\ell \circ \lambda^{\text{ref}}$ for all $j = 1, \dots, J$ and $\ell = 1, \dots, L$, with λ^{ref} as in (4.1). As described in the Section 4.1.2 we can compute the matrix

$$\widehat{C} \in \mathbb{R}^{L \times J} \quad \text{with entries} \quad \widehat{C}_{\ell,j} = \int_{T^{\text{ref}}} \varphi_j^{\text{ref}} b_\ell^{\text{ref}} dx = \int_T \varphi_j b_\ell dx.$$

Suppose we have an approximation $\mathcal{I}f = \sum_{j=1}^J f_j \varphi_j$ of f with coefficient vector $\mathbf{f} = (f_j)_{j=1}^J \in \mathbb{R}^J$, obtained for example by means of nodal interpolation. Then for all $\ell = 1, \dots, L$, one has

$$\int_T f b_\ell dx \approx \int_T \mathcal{I}f b_\ell dx = \sum_{j=1}^J f_j \int_T \varphi_j b_\ell dx = (\widehat{C} \mathbf{f})_\ell.$$

In particular, the matrix vector multiplication $\widehat{C} \mathbf{f}$ allows us to evaluate the right-hand side in our finite element scheme on each simplex $T \in \mathcal{T}$. In our implementations we precompute $\widehat{C}^\top =: \mathbf{bhat}$ and obtain \mathbf{f} by nodal interpolation, cf. line 19 in Figure 4.

4.1.4. *Change of basis.* Rather than using the basis $(b_\ell)_{\ell=1}^L$ from the previous subsections, we want to use a basis $(c_\ell)_{\ell=1}^L \subset \mathcal{P}(T) := \text{span}\{b_\ell: \ell = 1, \dots, L\}$, that corresponds to the degrees of freedom $(\psi_\ell)_{\ell=1}^L \subset \mathcal{P}(T)^*$ of the finite element in the sense that

$$\psi_\ell(c_m) = \delta_{\ell,m} \quad \text{for all } \ell, m = 1, \dots, L.$$

The representation of the basis functions $(c_\ell)_{\ell=1}^L$ might be independent of the underlying triangle $T \in \mathcal{T}$, as for example in the case of Lagrange finite elements or, more generally, for affinely equivalent elements, see [BS08, Def. 3.4.1]. However, since the elements discussed in this paper are not affinely equivalent, we compute the representation of the shape functions on the fly. For this purpose we define the generalized Vandermonde matrix

$$V = (V_{\ell,k})_{\ell,k=1}^L \in \mathbb{R}^{L \times L} \quad \text{with} \quad V_{\ell,k} := \psi_\ell(b_k) \quad \text{for all } \ell, k = 1, \dots, L.$$

Then, the coefficients $y = (y_\ell)_{\ell=1}^L \in \mathbb{R}^L$ in the expansion $c_m = \sum_{\ell=1}^L y_\ell b_\ell$, for each $m = 1, \dots, L$, are determined by

$$Vy = e_m,$$

where e_m denotes the m -th canonical unit vector in \mathbb{R}^L . In other words, the coefficients of each basis function are the columns of the inverse matrix V^{-1} . We obtain the system matrices with respect to the basis $(c_\ell)_{\ell=1}^L$ by multiplying the matrices

computed in Section 4.1.2 by the matrix V^{-1} and its transpose from the left and right, e.g., for the mass matrix

$$\widetilde{M} = (\widetilde{M}_{\ell,k})_{\ell,k=1}^L := V^{-\top} M V^{-1} \quad \text{satisfies} \quad \widetilde{M}_{\ell,k} = \int_T c_\ell c_k \, dx \quad \text{for } \ell, k = 1, \dots, L.$$

Similarly, we modify the computation of the right-hand side in Section 4.1.3 by multiplying the matrix \widehat{C} by $V^{-\top}$, i.e., we replace $\widehat{C}\mathbf{f}$ by $V^{-\top}\widehat{C}\mathbf{f}$.

4.1.5. *Assembly of the global system.* With the local integrals computed as in the previous subsections, one can assemble the global system matrices and right-hand side. With global basis functions $(B_j)_{j=1}^N$ and $N := \dim V_h$ and with $a(\bullet, \bullet): V_h \times V_h \rightarrow \mathbb{R}$ representing the bilinear form of the corresponding system matrix one has

$$\mathbf{A} = (\mathbf{A}_{j,k})_{j,k=1}^N \in \mathbb{R}^{N \times N} \quad \text{with } \mathbf{A}_{j,k} = a(B_j, B_k).$$

The basis $(B_j)_{j=1}^N$ is chosen such that $\Psi_k(B_j) = \delta_{k,j}$ for all $j, k = 1, \dots, N$ with global degrees of freedom Ψ_k . We initialize A as a $N \times N$ matrix with zero entries. Then we loop over all simplices $T \in \mathcal{T}$. For each simplex $T \in \mathcal{T}$ we compute the corresponding local system matrices as described above and add the values for the local degrees of freedom to the corresponding ones in A . Similarly, we proceed with the right-hand side, which leads to the vector $\mathbf{b} = (\mathbf{b}_j)_{j=1}^N \in \mathbb{R}^N$ with entries $\mathbf{b}_j = \int_\Omega \mathcal{I}f B_j \, dx$ for all $j = 1, \dots, N$.

Remark 4.2 (Improvement). *In Figure 4 and 6 we use a slow assembling routine that updates the sparse Matrix \mathbf{A} in each loop for the sake of a simpler presentation. For a more efficient implementation it is recommended to build the system matrix directly from an array of local system matrices and suitable index sets as described for example in [FPW11].*

4.2. **Singular Zienkiewicz element.** The previous section contains an approach to implement general finite elements that use polynomial and rational basis functions with degrees of freedom including point evaluations, evaluations of gradients, and normal derivatives on edges. However, the resulting routines are in general rather slow. We address this drawback by using the routines discussed in Section 4.1.1 to precompute certain quantities that are independent of the specific triangle $T \in \mathcal{T}$. This subsection discusses the resulting implementation for the singular Zienkiewicz element introduced in Section 2.1 for the biharmonic problem: One seeks $u_h \in V_h \cap H_0^2(\Omega)$ with $V_h := Z_s$ defined in Lemma 2.4 such that

$$\int_\Omega \Delta u_h \Delta v_h \, dx = \int_\Omega f v_h \, dx \quad \text{for all } v_h \in V_h \cap H_0^2(\Omega). \quad (4.4)$$

Let us recall the local basis of Z_s on a simplex $T = [v_0, v_1, v_2] \in \mathcal{T}$ with faces $\mathcal{F}(T) = \{f_0, f_1, f_2\}$. The local basis, see (2.6), contains the basis of $\mathcal{P}_2(T)$

$$b_1 := \lambda_2^2, \quad b_2 := \lambda_1 \lambda_2, \quad b_3 := \lambda_1^2, \quad b_4 := \lambda_0 \lambda_2, \quad b_5 := \lambda_0 \lambda_1, \quad b_6 := \lambda_0^2.$$

The remaining basis functions of the reduced cubic Hermite element are

$$b_7 := \lambda_0^2 \lambda_1 - \lambda_0 \lambda_1^2, \quad b_8 := \lambda_1^2 \lambda_2 - \lambda_1 \lambda_2^2, \quad b_9 := \lambda_2^2 \lambda_0 - \lambda_2 \lambda_0^2,$$

and the rational bubble basis functions

$$b_{10} := B_{f_0}, \quad b_{11} := B_{f_1}, \quad b_{12} := B_{f_2}.$$

The local degrees of freedom are

$$\begin{aligned}\psi_{1+j}(p) &= p(v_j), \quad \psi_{3+j}(p) = \partial_x p(v_j), \quad \psi_{7+j}(p) = \partial_y p(v_j), \\ \psi_{10+j}(p) &= \nabla p(\text{mid}(f_j)) \cdot \nu_{f_j},\end{aligned}\tag{4.5}$$

for all $p \in Z_s(T)$ and $j = 0, 1, 2$.

In our implementation the vertices $\{v_1, \dots, v_m\} = \mathcal{N}$ and edges $\{f_1, \dots, f_n\} = \mathcal{F}$ in the underlying triangulation \mathcal{T} are sorted according to their occurrence in the matrix $\mathbf{c4n} \in \mathbb{R}^{m \times 2}$ and $\mathbf{n4s} \in \mathbb{R}^{n \times 2}$. Those are computed in separate routines and contain the coordinates of each vertex and the vertex numbers of each edge as rows, respectively. The global degrees of freedom are

$$\begin{aligned}\Psi_j(w_h) &= w_h(v_j), \quad \Psi_{m+j}(w_h) = \partial_x w_h(v_j), \quad \Psi_{2m+j}(w_h) = \partial_y w_h(v_j), \\ \Psi_{3m+k}(w_h) &= \nabla w_h(\text{mid}(f_k)) \cdot \nu_{f_k},\end{aligned}\tag{4.6}$$

for all $w_h \in V_h$, $j = 1, \dots, m$, and $k = 1, \dots, n$. Let $(B_r)_{r=1}^N \subset V_h$ with dimension $N = 3m + n$ denote the corresponding basis functions in the sense of Section 4.1.5. Moreover, we define the system matrix $\mathbf{A} = (\mathbf{A}_{r,s})_{r,s=1}^N$ for the biharmonic problem (4.4) and the right-hand side $\mathbf{b} = (\mathbf{b}_r)_{r=1}^N \in \mathbb{R}^N$ by

$$\mathbf{A}_{r,s} = \int_{\Omega} \Delta B_r \Delta B_s \, dx \quad \text{and} \quad \mathbf{b}_r \approx \int_{\Omega} f B_r \, dx \quad \text{for all } r, s = 1, \dots, N.\tag{4.7}$$

While the computation of the right-hand side \mathbf{b} has been discussed in Section 4.1.3, the computation of the matrix \mathbf{A} is explained in the remainder of this subsection.

4.2.1. *Local stiffness matrix.* We start with the computation of the matrix $A_T \in \mathbb{R}^{12 \times 12}$ that represents the (local) bilinear form. Its entries read

$$(A_T)_{r,s} := \int_T \Delta b_r \Delta b_s \, dx \quad \text{for } r, s = 1, \dots, 12.\tag{4.8}$$

The identity in (3.3) yields for rational polynomials $b: T \rightarrow \mathbb{R}$ with $b = \hat{b} \circ \lambda$ and with the i -th row $\partial_{x_i} \lambda$ of $\nabla \lambda \in \mathbb{R}^{2 \times 3}$ that

$$\partial_{x_i} b = (\partial_{x_i} \lambda) (\nabla_{\lambda} \hat{b}) \circ \lambda.\tag{4.9}$$

Applying this formula twice leads to $\partial_{x_i}^2 b = (\partial_{x_i} \lambda) (\nabla_{\lambda}^2 \hat{b}) \circ \lambda (\partial_{x_i} \lambda)^{\top}$. In particular, we obtain the representation

$$\Delta b = \partial_{x_1}^2 b + \partial_{x_2}^2 b = \sum_{i=1,2} (\partial_{x_i} \lambda) (\nabla_{\lambda}^2 \hat{b}) \circ \lambda (\partial_{x_i} \lambda)^{\top}.\tag{4.10}$$

Let us rewrite the matrix A_T in (4.8). For $v, w \in \mathbb{R}^3$ and $H \in \mathbb{R}^{3 \times 3}$ the Frobenius product satisfies $v^{\top} H v = H : (v \otimes v)$ and thus

$$v^{\top} H v + w^{\top} H w = H : (v \otimes v + w \otimes w) = H : \left((v \mid w) \begin{pmatrix} v^{\top} \\ w^{\top} \end{pmatrix} \right).\tag{4.11}$$

Recall the matrix $G_T := D\lambda \in \mathbb{R}^{3 \times 2}$ computed in (4.2). Using (4.11) we can rewrite (4.10) as $\Delta b = (\nabla_{\lambda}^2 \hat{b} \circ \lambda) : G_T G_T^{\top}$. Hence, for indices $r, s = 1, \dots, 12$ the integrand in (4.8) is given by

$$\Delta b_r \Delta b_s = ((\nabla_{\lambda}^2 \hat{b}_r \circ \lambda) : G_T G_T^{\top}) ((\nabla_{\lambda}^2 \hat{b}_s \circ \lambda) : G_T G_T^{\top}).\tag{4.12}$$

Let $A, B, C \in \mathbb{R}^{3 \times 3}$ be matrices and define the tensors $[AB]$ with entries $[AB]_{i,j,k,l} = A_{i,j}B_{k,l}$ and $[CC]$ with entries $[CC]_{i,j,k,l} = C_{i,j}C_{k,l}$. One has the identity

$$\begin{aligned} (A : C)(B : C) &= \left(\sum_{i,j} A_{i,j}C_{i,j} \right) \left(\sum_{k,l} B_{k,l}C_{k,l} \right) \\ &= \sum_{i,j,k,l} A_{i,j}B_{k,l}C_{i,j}C_{k,l} = [AB] : [CC]. \end{aligned} \quad (4.13)$$

We define the tensors $\widehat{A} \in \mathbb{R}^{12 \times 12 \times 3 \times 3 \times 3 \times 3}$ and $Q_T \in \mathbb{R}^{3 \times 3 \times 3 \times 3}$ with entries

$$\begin{aligned} \widehat{A}(r, s, i, j, k, l) &:= \int_T \left((\partial_{\lambda_i} \partial_{\lambda_j} \hat{b}_r) (\partial_{\lambda_k} \partial_{\lambda_l} \hat{b}_s) \right) \circ \lambda \, dx, \\ Q_T(i, j, k, l) &:= (G_T G_T^\top)_{i,j} (G_T G_T^\top)_{k,l}. \end{aligned}$$

Notice that \widehat{A} , which is denoted by `Ahat` in Figure 4, is independent of T and thus is precomputed in our routine. Combining (4.12) and (4.13) leads to the identity

$$(A_T)_{r,s} = \int_T \Delta b_r \Delta b_s \, dx = |T| \widehat{A}(r, s, :, :, :, :): Q_T. \quad (4.14)$$

The MATLAB routine in Figure 4 performs this computation in lines 9–10.

```

1 for elem = 1 : nrElems
2   nodes = n4e(elem, :); % nodes of triangle
3   coords = c4n(nodes, :); % coordinates of the three nodes
4   sides = s4e(elem, :); % sides of triangle
5   DF = [coords(2, :)-coords(1, :); coords(3, :)-coords(1, :)]';
6   grad_lam = [-1, -1; 1, 0; 0, 1]/DF; % = [-1, -1; 1, 0; 0, 1]*inv(DF)
7   area = abs(det(DF))/2; % area of current simplex T
8   %% Compute local system matrix A_T %%
9   Q_T = tensorprod(grad_lam*grad_lam', grad_lam*grad_lam');
10  A_T = area*tensorprod(Ahat, Q_T, [3 4 5 6], [1 2 3 4]);
11  %% Local basis evaluation at dofs %%
12  Tgv = tensorprod(grad_lam, That_gv, 1, 3);
13  Tge = squeeze(pagemtimes(reshape(normal4s(sides, :)', [1, 2, 3]), permute
    (tensorprod(grad_lam, That_ge, 1, 3), [1, 3, 2])))');
14  V = [That_v; squeeze(Tgv(1, :, :)); squeeze(Tgv(2, :, :)); Tge];
15  Basis4elem(:, :, elem) = inv(V);
16  %% Update global matrix A and rhs b %%
17  l2g_Dof = [nodes, nrNodes+nodes, 2*nrNodes+nodes, 3*nrNodes+sides];
18  A(l2g_Dof, l2g_Dof) = A(l2g_Dof, l2g_Dof) + Basis4elem(:, :, elem)'*A_T*
    Basis4elem(:, :, elem);
19  b(l2g_Dof) = b(l2g_Dof) + area*Basis4elem(:, :, elem)'*bhat'*f(
    BaryCoords*coords);
20 end

```

FIGURE 4. MATLAB loop over all $T \in \mathcal{T}$ to obtain the matrix A and right-hand side b in (4.7).

4.2.2. *Local basis evaluation at dofs.* In lines 12–15 in Figure 4 our routine computes a basis $(c_j)_{j=1}^{12} \subset Z(T)$ such that with degrees of freedom ψ_j as in (4.5) we have

$$\psi_j(c_k) = \delta_{j,k} \quad \text{for all } j, k = 1, \dots, 12.$$

As described in Section 4.1.4 for this purpose we invert the generalized Vandermonde matrix $V = (V_{j,k})_{j,k=1}^{12} \in \mathbb{R}^{12 \times 12}$ with $V_{j,k} := \psi_j(b_k)$ for all $j, k = 1, \dots, 12$. Its entries are computed as follows.

The point evaluations ψ_1, ψ_2, ψ_3 are independent of the underlying triangle T and are stored in the precomputed matrix

$$\text{That_v} = (V_{j,k})_{j=1,2,3}^{k=1,\dots,12} \in \mathbb{R}^{3 \times 12}.$$

The dofs ψ_4, \dots, ψ_9 correspond to evaluations of the gradients at the vertices v_0, v_1, v_2 of T . We define the precomputed tensor $\text{That_gv} \in \mathbb{R}^{3 \times 12 \times 3}$ with

$$\text{That_gv}(i, j, :) = (\nabla_{\lambda} \hat{b}_j) \circ \lambda(v_{i-1}) \quad \text{for all } i = 1, 2, 3 \text{ and } j = 1, \dots, 12.$$

Using (4.9) this tensor allows us to compute the tensor $\text{Tgv} \in \mathbb{R}^{2 \times 3 \times 12}$ with entries $\text{Tgv}(:, i, j) = \nabla b_j(v_{i-1})$ for all $i = 1, 2, 3$ and $j = 1, \dots, 12$, cf. line 12 in Figure 4.

It remains to consider the evaluations $\psi_{10}, \psi_{11}, \psi_{12}$ of the normal derivatives at the midpoints of edges. The gradients at the face midpoints are computed similarly as Tgv . We then multiply these gradients with precomputed normal vectors. This leads to the matrix $\text{Tge} \in \mathbb{R}^{3 \times 12}$ with entries $\text{Tge}(i, j) = \nabla b_j(\text{mid}(f_{i-1})) \cdot \nu_{f_{i-1}}$ for all $i = 1, 2, 3$ and $j = 1, \dots, 12$, cf. line 13 in Figure 4. Combining these matrices leads to the matrix $V = V \in \mathbb{R}^{12 \times 12}$ in line 14 in Figure 4.

4.2.3. *Reduced Zienkiewicz element.* As discussed in Section 2.1 it is possible to reduce the number of basis functions, leading to the reduced singular Zienkiewicz element in (2.7). To obtain this reduced element, we have to modify our local basis functions b_1, \dots, b_9 by subtracting suitable scaled rational bubble functions b_{10}, b_{11}, b_{12} such that the resulting functions $b \in Z_s(T)$ satisfy $\nabla b|_{f_j} \cdot \nu_j \in \mathcal{P}_1(f_j)$ for all $j = 0, 1, 2$. This property is satisfied for the quadratic basis functions $b_1, \dots, b_6 \in \mathcal{P}_2(T)$ without any modification. For $b \in \{b_7, b_8, b_9\}$ we use the ansatz

$$\bar{b} := b - \sum_{j=0}^2 \gamma_j B_{f_j} \quad \text{with coefficients } \gamma_0, \gamma_1, \gamma_2 \in \mathbb{R} \text{ to be determined.}$$

According to Lemma 2.3 the function $\nabla \bar{b}|_{f_j}$ is a quadratic function on the edge $f_j = [v_{j+1}, v_{j+2}]$, for $j = 0, 1, 2$. Hence, the function \bar{b} satisfies $\nabla \bar{b}|_{f_j} \cdot \nu_j \in \mathcal{P}_1(f_j)$ if and only if

$$\nabla \bar{b}(\text{mid}(f_j)) \cdot \nu_{f_j} = \frac{1}{2} (\nabla \bar{b}(v_{j+1}) + \nabla \bar{b}(v_{j+2})) \cdot \nu_{f_j}.$$

Due to the properties of the rational bubble functions stated in Lemma 2.3 and by the definition of \bar{b} this is equivalent to

$$\nabla (b(\text{mid}(f_j)) - \gamma_j B_{f_j}(\text{mid}(f_j))) \cdot \nu_{f_j} = \frac{1}{2} (\nabla b(v_{j+1}) + \nabla b(v_{j+2})) \cdot \nu_{f_j}.$$

This determines the coefficients to be

$$\gamma_j = \frac{\nabla b(\text{mid}(f_j)) \cdot \nu_{f_j} - \frac{1}{2} (\nabla b(v_{j+1}) + \nabla b(v_{j+2})) \cdot \nu_{f_j}}{B_{f_j}(\text{mid}(f_j)) \cdot \nu_{f_j}} \quad \text{for } j = 0, 1, 2.$$

```

1 gamma0 = (V(10,7:9)-normal4s(sides(1),:)/2*(V([5,8],7:9)+V([6,9],7:9))
   )/V(10,10);
2 gamma1 = (V(11,7:9)-normal4s(sides(2),:)/2*(V([6,9],7:9)+V([4,7],7:9))
   )/V(11,11);
3 gamma2 = (V(12,7:9)-normal4s(sides(3),:)/2*(V([4,7],7:9)+V([5,8],7:9))
   )/V(12,12);
4 RedBasis = [eye(9); zeros(3,6), -[gamma0; gamma1; gamma2]];
5 Basis4elem(:, :, elem) = RedBasis/V(1:9,1:9);
6 12g_Dof = [nodes, nrNodes+nodes, 2*nrNodes+nodes];

```

FIGURE 5. Modification replacing lines 15–17 in Figure 4 to obtain the reduced singular Zienkiewicz element.

The values on the right-hand side have been computed and stored in the matrix $V \in \mathbb{R}^{12 \times 12}$, see Section 4.2.2, in the sense that for all $k = 1, \dots, 9$ and $j = 0, 1, 2$

$$\begin{aligned} \nabla b_k(\text{mid}(f_j)) \cdot \nu_{f_j} &= V(10 + j, k), \\ \nabla b_k(v_j) &= (V(4 + j, k), V(7 + j, k))^\top, \\ B_{f_j}(\text{mid}(f_j)) \cdot \nu_{f_j} &= b_{10+j}(\text{mid}(f_j)) \cdot \nu_{f_j} = V(10 + j, 10 + j). \end{aligned}$$

This allows us to compute the values γ_j as in line 1–3 in Figure 5. In this way we obtain the basis functions $\bar{b}_1, \dots, \bar{b}_9$ of $Z_r(T)$ with $\bar{b}_k = b_k$ for $k = 1, \dots, 6$. Since the degrees of freedom ψ_1, \dots, ψ_9 do not see the correction by the rational bubble functions according to Lemma 2.3, that is, $\psi_k(B_{f_j}) = 0$ for $k = 1, \dots, 9$ and $j = 0, 1, 2$, the Vandermonde matrix remains unchanged in the sense that $\psi_k(\bar{b}_\ell) = \psi_k(b_\ell)$ for all $k, \ell = 1, \dots, 9$. This allows us to compute the coefficients $(y_\ell)_{\ell=1}^{12} \in \mathbb{R}^{12}$ of the functions $\bar{c}_k = \sum_{\ell=1}^{12} y_\ell b_\ell \in Z_r(T)$ with $\psi_\ell(\bar{c}_k) = \delta_{\ell,k}$ for all $k, \ell = 1, \dots, 9$ as illustrated in lines 4–5 in Figure 5.

4.3. Guzmán–Neilan element. In this subsection we present an implementation of the 2D Guzmán–Neilan element for the Stokes problem. More precisely, we consider the Guzmán–Neilan element as presented in Section 2.2 with global spaces V and Q defined in Lemma 2.6 with given underlying triangulation \mathcal{T} . The discretized Stokes problems seeks $(u_h, \pi_h) \in (V \cap H_0^1(\Omega))^2 \times Q \cap L_0^2(\Omega)$ such that

$$\begin{aligned} \int_{\Omega} \nabla u_h : \nabla v_h \, dx - \int_{\Omega} \pi_h \operatorname{div} v_h \, dx &= \int_{\Omega} f v_h \, dx && \text{for all } v_h \in V \cap H_0^1(\Omega)^2, \\ \int_{\Omega} q_h \operatorname{div} u_h &= 0 && \text{for all } q_h \in Q \cap L_0^2(\Omega). \end{aligned}$$

To define our local basis functions for any simplex $T = [v_0, v_1, v_2]$, recall the functions b_{f_i} and B_{f_i} with $i = 0, 1, 2$ defined in (2.5), and set $e_1 = (1, 0)^\top$, $e_2 = (0, 1)^\top$. The local velocity space $V(T)$ is spanned by the basis functions

$$\begin{aligned} b_1 &:= \lambda_0 e_1, & b_2 &:= \lambda_1 e_1, & b_3 &:= \lambda_2 e_1, \\ b_4 &:= \lambda_0 e_2, & b_5 &:= \lambda_1 e_2, & b_6 &:= \lambda_2 e_2, \\ b_7 &:= \operatorname{curl}(\lambda_0^2 \lambda_1 - \lambda_1^2 \lambda_0), & b_8 &:= \operatorname{curl}(\lambda_1^2 \lambda_2 - \lambda_2^2 \lambda_1), & b_9 &:= \operatorname{curl}(\lambda_2^2 \lambda_0 - \lambda_0^2 \lambda_2), \\ b_{10} &:= \operatorname{curl}(B_{f_0}), & b_{11} &:= \operatorname{curl}(B_{f_1}), & b_{12} &:= \operatorname{curl}(B_{f_2}). \end{aligned} \tag{4.15}$$

Let us additionally define

$$\begin{aligned}\rho_1 &:= \lambda_0^2 \lambda_1 - \lambda_1^2 \lambda_0, & \rho_2 &:= \lambda_1^2 \lambda_2 - \lambda_2^2 \lambda_1, & \rho_3 &:= \lambda_2^2 \lambda_0 - \lambda_0^2 \lambda_2, \\ \rho_4 &:= B_{f_0}, & \rho_5 &:= B_{f_1}, & \rho_6 &:= B_{f_2}.\end{aligned}$$

The local basis of $Q(T)$ consists of the single constant function

$$q_1 = 1.$$

The local degrees of freedom are for all $p \in V(T)$ and $j = 0, 1, 2$ defined by

$$\begin{aligned}\psi_{1+j}(v) &= (p(v_j))_1, & \psi_{4+j}(p) &= (p(v_j))_2, \\ \psi_{7+j}(p) &= p(\text{mid}(f_j)) \cdot \nu_{f_j}, & \psi_{10+j}(p) &= p(\text{mid}(f_j)) \cdot \tau_{f_j}.\end{aligned}\tag{4.16}$$

In our implementation the vertices $\{v_1, \dots, v_m\} = \mathcal{N}$ and edges $\{f_1, \dots, f_n\} = \mathcal{F}$ in the underlying triangulation \mathcal{T} are sorted according to their occurrence in the matrix $\mathbf{c4n} \in \mathbb{R}^{m \times 2}$ and $\mathbf{n4s} \in \mathbb{R}^{n \times 2}$, which contains the coordinates of each vertex and the vertex numbers of each edge, respectively. Then the global degrees of freedom are, for all $w_h \in V$, $j = 1, \dots, m$, and $k = 1, \dots, n$,

$$\begin{aligned}\Psi_j(w_h) &= (w_h(v_j))_1, & \Psi_{m+j}(w_h) &= (w_h(v_j))_2, \\ \Psi_{2m+k}(w_h) &= w_h(\text{mid}(f_k)) \cdot \nu_{f_k}, & \Psi_{2m+n+k}(w_h) &= w_h(\text{mid}(f_k)) \cdot \tau_{f_k}.\end{aligned}\tag{4.17}$$

The corresponding global basis in the sense of Section 4.1.5 is denoted by $(B_r)_{r=1}^N \subset V_h$ with dimension $N = 2m + 2n$. Moreover, in our implementation the simplices $\{T_1, \dots, T_p\} = \mathcal{T}$ are sorted according to their occurrence in the matrix $\mathbf{n4e} \in \mathbb{R}^{p \times 3}$ whose j -th row contains the indices of the vertices of T_j . Then $(q_j)_{j=1}^p$ with $q_j := \mathbb{1}_{T_j}$ for all $j = 1, \dots, p$ is a basis of Q . The remainder of this subsection focuses on the computation of the local contribution of the matrices $\mathbf{A} = (\mathbf{A}_{r,s})_{r,s=1}^N \in \mathbb{R}^{N \times N}$ and $\mathbf{B} = (\mathbf{B}_{r,j})_{r=1, \dots, N}^{j=1, \dots, p} \in \mathbb{R}^{N \times p}$ as well as of the right-hand side $\mathbf{b} = (\mathbf{b}_r)_{r=1}^N \in \mathbb{R}^N$ such that for all $r, s = 1, \dots, N$ and $j = 1, \dots, p$

$$\mathbf{A}_{r,s} = \int_{\Omega} \nabla B_r : \nabla B_s \, dx, \quad \mathbf{B}_{r,j} = \int_{\Omega} q_j \operatorname{div} B_r \, dx, \quad \mathbf{b}_r \approx \int_{\Omega} f B_r \, dx.\tag{4.18}$$

4.3.1. Local stiffness matrix. First, we describe the computation of the local stiffness matrix $A_T \in \mathbb{R}^{12 \times 12}$. Given a simplex $T = [v_0, v_1, v_2] \in \mathcal{T}$ its entries read

$$(A_T)_{r,s} = \int_T \nabla b_r : \nabla b_s \, dx \quad \text{for all } r, s = 1, \dots, 12.$$

We divide this matrix into blocks $R_T, M_T, P_T \in \mathbb{R}^{6 \times 6}$ in the sense that

$$A_T = \begin{pmatrix} P_T & M_T \\ M_T^\top & R_T \end{pmatrix}.$$

This means P_T contains the entries where both basis functions are affine polynomials, R_T contains the entries where both basis functions are curls of a rational functions, and M_T contains the mixed ones.

Let us start with computing P_T . Its entries are

$$(P_T)_{r,s} := \int_T \nabla b_r : \nabla b_s \, dx \quad \text{for } r, s = 1, \dots, 6.$$

For all $j, k = 1, 2, 3$ we have the identity

$$P_{j,k} = \int_T \nabla \lambda_{j-1} \cdot \nabla \lambda_{k-1} \, dx = \int_T \nabla b_j : \nabla b_k \, dx = \int_T \nabla b_{3+j} : \nabla b_{3+k} \, dx.$$

```

1 for elem = 1 : nrElems
2   nodes = n4e(elem,:); % nodes of triangle
3   coords = c4n(nodes,:); % coordinates of the three nodes
4   sides = s4e(elem,:); % sides of triangle
5   DF = [coords(2,:)-coords(1,:); coords(3,:)-coords(1,:)]';
6   grad_lam = [-1,-1;1,0;0,1]/DF; %grad_lam = [-1,-1;1,0;0,1]*inv(DF)
7   area = abs(det(DF))/2; % area equals volume of current simplex
8   %% Compute local stiffness matrix A_T (P_T, R_T ,M_T) %%
9   GG = grad_lam*grad_lam';
10  A_T(1:3,1:3) = area*GG; A_T(4:6,4:6) = area*GG; %P_T
11  Q_T = tensorprod(GG,GG);
12  A_T(7:12,7:12) = area*tensorprod(Rhat,Q_T,[3 4 5 6],[1 2 3 4]); %R_T
13  W_T = tensorprod(R*grad_lam',GG);
14  A_T(1:6,7:12) = area*tensorprod(Mhat,W_T,[3 4 5 6],[1 2 3 4]); %M_T
15  A_T(7:12,1:6) = A_T(1:6,7:12)';
16  B_T = area*grad_lam(:);
17  %% Local basis evaluation at dofs %%
18  normals = normal4s(sides,:); %normal vector for each side
19  tangents = tangent4s(sides,:); %tangent vector for each side
20  V_left = [eye(6); normals(1,:)*val_mid1'; normals(2,:)*val_mid2';
21            normals(3,:)*val_mid3'; tangents(1,:)*val_mid1'; tangents(2,:)*
22            val_mid2'; tangents(3,:)*val_mid3'];
23  Tgv = tensorprod(grad_lam,That_gv,1,3);
24  Tge = tensorprod(grad_lam,That_ge,1,3);
25  for i=1:3
26    Tge_nt([i,i+3],:) = [normals(i,:); tangents(i,:)]*R*squeeze(Tge(:,i),:); end
27  V_right = [squeeze(Tgv(2, :, :)); -squeeze(Tgv(1, :, :)); Tge_nt];
28  V = [V_left, V_right];
29  Basis4elem(:, :, elem) = inv(V);
30  %% Update global matrix A and rhs b %%
31  l2g_Dof = [nodes, nrNodes+nodes, 2*nrNodes+sides, 2*nrNodes+nrSides+
32            sides];
33  A(l2g_Dof, l2g_Dof) = A(l2g_Dof, l2g_Dof) + Basis4elem(:, :, elem)'*A_T*
34            Basis4elem(:, :, elem);
35  B(l2g_Dof, elem) = Basis4elem(1:6, :, elem)'*B_T;
36  f_loc = f(BaryCoords*coords);
37  b2 = tensorprod(grad_lam, bhat2, 1, 2);
38  b_T1 = [bhat1'*f_loc(:, 1); bhat1'*f_loc(:, 2)];
39  b_T2 = squeeze(b2(2, :, :))'*f_loc(:, 1) - squeeze(b2(1, :, :))'*f_loc(:, 2);
40  b(l2g_Dof) = area*b(l2g_Dof)+Basis4elem(:, :, elem)'*[b_T1; b_T2];
41 end

```

FIGURE 6. Computation of the local contributions to assemble the matrices A , B , and b in (4.18).

Moreover, the off-diagonal blocks equal zero, that is, $P_{j,3+k} = 0 = P_{3+j,k}$ for all $j, k = 1, 2, 3$. Hence, with the matrix $G_T = D\lambda \in \mathbb{R}^{3 \times 2}$ computed as in (4.2) we have, cf. Figure 6, line 10, that

$$P_T = \begin{pmatrix} |T| G_T G_T^\top & 0 \\ 0 & |T| G_T G_T^\top \end{pmatrix}.$$

Let us now consider $R_T \in \mathbb{R}^{6 \times 6}$. Recall that in 2D the curl is given by

$$\mathbf{curl} f = R \nabla f \quad \text{with rotation matrix } R = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

Hence, we obtain for all $r, s = 1, \dots, 6$ that

$$\begin{aligned} (R_T)_{r,s} &:= \int_T \nabla b_{r+6} : \nabla b_{s+6} \, dx = \int_T (\nabla \mathbf{curl} \rho_r) : (\nabla \mathbf{curl} \rho_s) \, dx \\ &= \int_T (\nabla R \nabla \rho_r) : (\nabla R \nabla \rho_s) \, dx = \int_T \nabla^2 \rho_r : \nabla^2 \rho_s \, dx. \end{aligned}$$

With $G_T = D\lambda$ and (3.4) the integrand equals

$$\nabla^2 \rho_r : \nabla^2 \rho_s = (G_T^\top (\nabla_\lambda^2 \hat{\rho}_r) \circ \lambda G_T) : (G_T^\top (\nabla_\lambda^2 \hat{\rho}_s) \circ \lambda G_T).$$

We define the tensors $\widehat{R} \in \mathbb{R}^{6 \times 6 \times 3 \times 3 \times 3 \times 3}$ and $Q_T \in \mathbb{R}^{3 \times 3 \times 3 \times 3}$ as

$$\begin{aligned} \widehat{R}(r, s, i, j, k, l) &:= \int_T ((\partial_{\lambda_i} \partial_{\lambda_k} \hat{\rho}_r) (\partial_{\lambda_j} \partial_{\lambda_l} \hat{\rho}_s)) \circ \lambda \, dx, \\ Q_T(i, j, k, l) &:= (G_T G_T^\top)_{i,j} (G_T G_T^\top)_{k,l}. \end{aligned}$$

Component-wise computation reveals the identity $(R_T)_{r,s} = |T| \widehat{R}(r, s, :, :, :, :): Q_T$. In our implementation the element-independent tensor $\widehat{R} = \mathbf{Rhat}$ is precomputed and the computation of $R_T = \mathbf{R_T}$ is shown in Figure 6, line 12.

It remains to compute the submatrix M_T with entries, for $r, s = 1, \dots, 6$,

$$(M_T)_{r,s} = \int_T \nabla b_r : \nabla b_{s+6} \, dx = \int_T \nabla b_r : \nabla R \nabla \rho_s \, dx = \int_T \nabla b_r : R \nabla^2 \rho_s \, dx.$$

With the identities (3.3), (3.4), and $G_T = D\lambda$ the integrand has the form

$$\nabla b_r : R \nabla^2 \rho_s = (\nabla_\lambda \hat{b}_r) \circ \lambda G_T : R (G_T^\top (\nabla_\lambda^2 \hat{\rho}_s) \circ \lambda G_T).$$

Let us define the tensors $\widehat{M} \in \mathbb{R}^{6 \times 6 \times 2 \times 3 \times 3 \times 3}$ and $W_T \in \mathbb{R}^{2 \times 3 \times 3 \times 3}$ by

$$\begin{aligned} \widehat{M}(r, s, i, j, k, l) &:= \int_T (\partial_{\lambda_k} (\hat{b}_r)_i (\partial_{\lambda_j} \partial_{\lambda_l} \hat{\rho}_s)) \circ \lambda \, dx, \\ W_T(i, j, k, l) &:= (R G_T^\top)_{i,j} (G_T G_T^\top)_{k,l}. \end{aligned}$$

This leads to the identity $(M_T)_{r,s} = |T| \widehat{M}(r, s, :, :, :, :): W_T$. In our implementation the element-independent tensor $\widehat{M} = \mathbf{Mhat}$ is precomputed and the computation of $M_T = \mathbf{M_T}$ is shown in Figure 6, line 14.

4.3.2. *System matrix.* Since the pressure space is one-dimensional, the contributions $\int_T q_1 \operatorname{div} b_r \, dx$ with $q_1 = 1$ are represented by a vector $B_T \in \mathbb{R}^{12}$ with entries

$$(B_T)_r = \int_T \operatorname{div} b_r(x) \, dx \quad \text{for } r = 1, \dots, 12.$$

It follows by $\operatorname{div} \mathbf{curl} \rho = 0$ that $(B_T)_r = 0$ for all $r = 7, \dots, 12$. Due to the definition of the basis in (4.15) for $j = 1, 2, 3$ we have that

$$\operatorname{div} b_j = \partial_x \lambda_{j-1} \quad \text{and} \quad \operatorname{div} b_{3+j} = \partial_y \lambda_{j-1}.$$

In particular, we obtain

$$(B_T)_r = \begin{cases} |T| \partial_x \lambda_{r-1} & \text{for } r = 1, \dots, 3, \\ |T| \partial_y \lambda_{r-4} & \text{for } r = 4, \dots, 6, \\ 0 & \text{for } r = 7, \dots, 12. \end{cases}$$

This leads to the computation of $(B_T)_{r=1}^6 = \mathbf{B_T}$ as in line 16 in Figure 6.

4.3.3. *Local basis evaluation at dofs.* Since point evaluations of the basis functions b_1, \dots, b_6 at vertices and midpoints of edges are independent of the underlying simplex $T = [v_0, v_1, v_2]$, we obtain the left side $\mathbf{V_left} = (V_{r,s})_{r=1, \dots, 12}^{s=1, \dots, 6} \in \mathbb{R}^{12 \times 6}$ of the generalized Vandermonde matrix by multiplying precomputed point evaluations with the corresponding normal and tangential vectors, see Figure 6, line 20.

To obtain the remaining part $\mathbf{V_right} = (V_{r,s})_{r=1, \dots, 12}^{s=7, \dots, 12}$ of the Vandermonde matrix, we apply the considerations in Section 4.2.2 to compute in line 21–22 in Figure 6 the tensors $\mathbf{Tgv}, \mathbf{Tge} \in \mathbb{R}^{2 \times 3 \times 6}$ with entries

$$\mathbf{Tgv}(:, j, s) = \nabla \rho_s(v_{j-1}) \quad \text{and} \quad \mathbf{Tge}(:, j, s) = \nabla \rho_s(\text{mid}(f_{j-1})),$$

for all $j = 1, 2, 3$ and $s = 1, \dots, 6$. While the values in \mathbf{Tgv} can be directly added to the Vandermonde matrix V , one has to rotate and multiply the values in \mathbf{Tge} by the normal and tangential vectors as done in line 24 in Figure 6. Combining the values leads to the matrix $\mathbf{V} = \mathbf{V}$ in line 26 in Figure 6.

4.3.4. *Assembly of the right-hand side.* Let $\mathcal{P}_m(T)$ be the polynomial space of maximal degree $m \in \mathbb{N}_0$ with Lagrange basis $\{\varphi_1, \dots, \varphi_J\}$. In lines 32–36 of Figure 6 the right-hand side is assembled. The evaluation of the local components $\int_T \mathcal{I} f b_j dx$ for $j = 1, \dots, 6$ with nodal interpolation operator $\mathcal{I}: C^0(T) \rightarrow \mathcal{P}_m(T)$ is described in Section 4.1.3. To evaluate the local components $\int_T \mathcal{I} f \mathbf{curl} \rho_j dx$ for $j = 1, \dots, 6$, the code precomputes the tensor $\mathbf{bhat2} \in \mathbb{R}^{J \times 3 \times 6}$ containing the values

$$\mathbf{bhat2}(j, k, r) = \int_T (\hat{\varphi}_j \partial_{\lambda_k} \hat{\rho}_r) \circ \lambda dx \quad \text{for all } j = 1, \dots, J, k = 1, \dots, 3, r = 1, \dots, 6.$$

Exploiting the representation of $\nabla \rho_s = \nabla \lambda (\nabla_\lambda \hat{\rho}_s) \circ \lambda$ in (3.3), we obtain in line 33 in Figure 6 the tensor $\mathbf{b2}$ with entries

$$\mathbf{b2}(m, j, r) = \int_T \varphi_j \partial_{x_m} \rho_r dx \quad \text{for all } m = 1, 2, j = 1, \dots, J, r = 1, \dots, 6.$$

With this we evaluate in line 35 in Figure 6 the contributions

$$\mathbf{b_T2}(r) = \int_T \mathcal{I} f \mathbf{curl} \rho_r dx = \int_T \mathcal{I} f b_{6+r} dx \quad \text{for all } r = 1, \dots, 6.$$

These contributions are added to obtain the global vector \mathbf{b} .

4.3.5. *Reduced Guzmán–Neilan element.* To obtain the reduced Guzmán–Neilan element described in Lemma 2.9, we have to correct the local basis b_1, \dots, b_9 by the rational bubble functions $b_{10} = \mathbf{curl} B_{f_0}$, $b_{11} = \mathbf{curl} B_{f_1}$, $b_{12} = \mathbf{curl} B_{f_2}$ such that the resulting functions $\bar{b} \in V_r(T)$ satisfy

$$\bar{b}|_{f_j} \cdot \tau_{f_j} \in \mathcal{P}_1(f_j) \quad \text{for all } j = 0, 1, 2. \quad (4.19)$$

```

1 gam0=(V(10,7:9)-tangents(1,:)/2*(V([2,5],7:9)+V([3,6],7:9)))/V(10,10);
2 gam1=(V(11,7:9)-tangents(2,:)/2*(V([3,6],7:9)+V([1,4],7:9)))/V(11,11);
3 gam2=(V(12,7:9)-tangents(3,:)/2*(V([1,4],7:9)+V([2,5],7:9)))/V(12,12);
4 RedBasis = [eye(9);zeros(3,6),-[gam0;gam1;gam2]];
5 Basis4elem(:, :, elem) = RedBasis/V(1:9,1:9);
6 l2g_Dof = [nodes, nrNodes+nodes, 2*nrNodes+sides];

```

FIGURE 7. Modification replacing lines 27–29 in Figure 6 to obtain the reduced Guzmán–Neilan element.

This is trivially satisfied for the affine basis functions $b_1 = \bar{b}_1, \dots, b_6 = \bar{b}_6 \in \mathcal{P}_1(T)^2$. For the corrections $\bar{b}_k = b_k - \sum_{\ell=1}^3 \gamma_\ell b_{9+\ell}$ with $k = 7, 8, 9$ and coefficients $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}$ of the remaining basis functions we exploit the identity

$$\bar{b}_k|_{f_j} \cdot \tau_{f_j} = \mathbf{curl} \left(\rho_{k-6} - \sum_{\ell=1}^3 \gamma_\ell \rho_{3+\ell} \right) \Big|_{f_j} \cdot \tau_{f_j} = \mathbf{curl} \left(\rho_{k-6} - \gamma_j B_{f_j} \right) \Big|_{f_j} \cdot \tau_{f_j}.$$

This allows us to compute the coefficients γ_j similarly as in Section 4.2.3, which leads to the additional code displayed in Figure 7.

5. NUMERICAL EXPERIMENT

In this work we have presented an exact numerical integration for rational finite element functions. Previously, only inexact quadrature was available for rational functions on triangles. In this experiment we investigate the benefits of our exact numerical integration.

It is well known that the quadrature error may spoil the approximation order when approximating solutions to PDEs as shown for the linear elliptic PDE $-\operatorname{div}(a\nabla u) = f$ for smooth coefficient function a discretized by Lagrange elements, c.f. [Cia02, Thm. 4.1.6] and for the related eigenvalue problem [BO90; Ban92]. The results state that for smooth solutions and Lagrange elements of degree k the order of convergence remains optimal if the quadrature rule is exact for polynomials of maximal degree $2k - 2$ for the source term and of $2k - 1$ for the eigenvalue problem. For rough coefficients a , randomized quadrature rules have been investigated in [KPW19].

In the remainder of this section we numerically study the effect of quadrature errors in the singular Zienkiewicz FEM for the biharmonic eigenvalue problem: Seek an eigenfunction $\varphi \in H^2(\Omega) \setminus \{0\}$ with smallest eigenvalue $\lambda > 0$ such that

$$\begin{aligned} \Delta^2 \varphi &= \lambda \varphi && \text{in } \Omega, \\ \varphi &= \nabla \varphi \cdot \nu = 0 && \text{on } \partial\Omega. \end{aligned}$$

The domain Ω is chosen as the unit square $\Omega = (0, 1)^2$ or the L-shaped domain $\Omega = (-1, 1)^2 \setminus [0, 1)^2$. The underlying triangulations are uniform for the square and graded towards the re-entrant corner for the L-shaped domain. For the (inexact) quadrature we use Fubini's theorem and 1D Gauß quadrature, that is, for a function $g: T \rightarrow \mathbb{R}$ with $T = [(0, 0)^\top, (1, 0)^\top, (0, 1)^\top]$ we approximate the integral with suitable Gauß points $x_i, y_{i,j} \in T$ and weights $\omega_i, \omega_{i,j} \in \mathbb{R}$ for $i, j = 1, \dots, n$ with

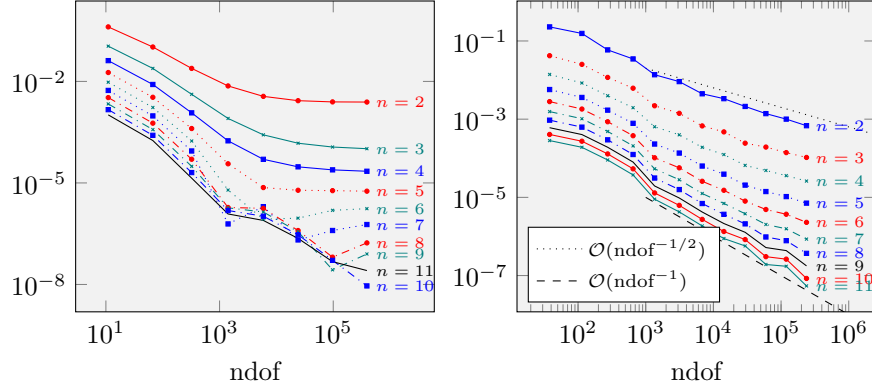


FIGURE 8. Distance $|\lambda_h - \bar{\lambda}_h|/\lambda_h$ on the square with uniform refinement (left) and L-shaped domain with adaptive refinement (right), where $\bar{\lambda}_h$ resulted from computations with the quadrature rule in (5.1) and various n .

$n \in \mathbb{N}$ by

$$\begin{aligned} \int_T g(x, y) \, d(x, y) &= \int_0^1 \int_0^{1-x} g(x, y) \, dy \, dx \approx \sum_{i=1}^n \omega_i \int_0^{1-x} g(x_i, y) \, dy \\ &\approx \sum_{i=1}^n \omega_i \sum_{j=1}^n \omega_{i,j} g(x_i, y_{i,j}). \end{aligned} \quad (5.1)$$

Notice that for polynomials $g \in \mathcal{P}_k(T)$ the integrand $\int_0^{1-x} g(x, y) \, dy$ is a polynomial in x of maximal degree $k + 1$. Thus, exploiting the exactness of Gauß quadrature for polynomials with maximal degree $2n - 1$ in 1D, this approach is exact for polynomials $g \in \mathcal{P}_{2n-2}(T)$. We use the quadrature in (5.1) to approximate the integrands in (4.8) and the local contributions of the mass matrix

$$(M_T)_{r,s} = \int_T b_r b_s \, dx \quad \text{for all } r, s = 1, \dots, 12.$$

We solve the resulting eigenvalue problem and compare the approximated eigenvalue λ_h obtained with exact integration by the implementation presented in Section 4.2 and the eigenvalue $\bar{\lambda}_h$ obtained by numerical quadrature (5.1) in the computation of the system matrices. The results are displayed in Figure 8.

Our computations with uniform mesh refinements of the square domain indicate a stagnation of the error as the mesh becomes finer, and hence suggests a failure of convergence. This effect is quite strong for small number of Gauß points n , and seems to be less prominent for larger n . However, it can be observed, that the onset of the stagnation is merely shifted for larger n .

Also for the adaptive mesh refinement there are some issues. While the solutions derived with approximated system matrices seem to exhibit convergence, for small numbers of Gauß points n one can observe a reduced convergence order $\text{ndof}^{-1/2}$, instead of ndof^{-1} as for larger n . Notably, the accelerated rate ndof^{-1} reflects the convergence of the exact scheme towards the minimal eigenvalue, that is, $\lambda_h - \lambda = \mathcal{O}(\text{ndof}^{-1})$, cf. [CP23, Sec. 4.3]. Thus, smaller values of n retard

the convergence rate of the numerical scheme, whereas larger values of n restore the rate to its expected trajectory. However, this may well be true only in the pre-asymptotic stage. With an increase in the degrees of freedom, the schemes employing inexact quadrature might still encounter a stagnation, similar to the one observed for uniform mesh refinement of the unit square domain.

Acknowledgements. We thank Professor Werner Hoffmann for the elegant proof of Lemmas 3.3 and 3.4. The work of the first two authors was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1283/2 2021 – 317210226.

REFERENCES

- [ACF99] J. Albery, C. Carstensen, and S. A. Funken. “Remarks around 50 lines of Matlab: short finite element implementation”. *Numer. Algorithms* 20.2-3 (1999), pp. 117–137. DOI: [10.1023/A:1019155918070](https://doi.org/10.1023/A:1019155918070).
- [ACFK02] J. Albery, C. Carstensen, S. A. Funken, and R. Klose. “Matlab implementation of the finite element method in elasticity”. *Computing* 69.3 (2002), pp. 239–263. DOI: [10.1007/s00607-002-1459-8](https://doi.org/10.1007/s00607-002-1459-8).
- [ACHRS12] F. Auricchio, F. Calabro, T. J. R. Hughes, A. Reali, and G. Sangalli. “A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis”. *Comput. Methods Appl. Mech. Engrg.* 249/252 (2012), pp. 15–27. DOI: [10.1016/j.cma.2012.04.014](https://doi.org/10.1016/j.cma.2012.04.014).
- [AFS68] J. H. Argyris, I. Fried, and D. W. Scharpf. “The TUBA Family of Plate Elements for the Matrix Displacement Method”. *The Aeronautical Journal* 72.692 (Aug. 1968), pp. 701–709. DOI: [10.1017/s000192400008489x](https://doi.org/10.1017/s000192400008489x).
- [AS64] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Vol. No. 55. National Bureau of Standards Applied Mathematics Series. For sale by the Superintendent of Documents. U. S. Government Printing Office, Washington, DC, 1964, pp. xiv+1046.
- [Aus+21] A. P. Austin et al. “Practical algorithms for multivariate rational approximation”. *Comput. Phys. Commun.* 261 (2021), Paper No. 107663, 17. DOI: [10.1016/j.cpc.2020.107663](https://doi.org/10.1016/j.cpc.2020.107663).
- [Ban92] U. Banerjee. “A note on the effect of numerical quadrature in finite element eigenvalue approximation”. *Numer. Math.* 61.2 (1992), pp. 145–152. DOI: [10.1007/BF01385502](https://doi.org/10.1007/BF01385502).
- [Bar15] S. Bartels. *Numerical methods for nonlinear partial differential equations*. Vol. 47. Springer Series in Computational Mathematics. Springer, Cham, 2015, pp. x+393. DOI: [10.1007/978-3-319-13797-1](https://doi.org/10.1007/978-3-319-13797-1).
- [BC05] C. Bahriawati and C. Carstensen. “Three MATLAB implementations of the lowest-order Raviart-Thomas MFEM with a posteriori error control”. *Comput. Methods Appl. Math.* 5.4 (2005), pp. 333–361. DOI: [10.2478/cmam-2005-0016](https://doi.org/10.2478/cmam-2005-0016).
- [BGGPV22] A. Buffa, G. Gantner, C. Giannelli, D. Praetorius, and R. Vázquez. “Mathematical foundations of adaptive isogeometric analysis”. *Arch. Comput. Methods Eng.* 29.7 (2022), pp. 4479–4555. DOI: [10.1007/s11831-022-09752-5](https://doi.org/10.1007/s11831-022-09752-5).
- [BGHN01] A. Bultheel, P. González-Vera, E. Hendriksen, and O. Njåstad. “Quadrature and orthogonal rational functions”. Vol. 127. 1-2. Numerical analysis 2000, Vol. V, Quadrature and orthogonal polynomials. 2001, pp. 67–91. DOI: [10.1016/S0377-0427\(00\)00493-3](https://doi.org/10.1016/S0377-0427(00)00493-3).

- [BHH24] N. Boullé, A. Herremans, and D. Huybrechs. “Multivariate Rational Approximation of Functions with Curves of Singularities”. *SIAM J. Sci. Comput.* 46.6 (2024), A3401–A3426. DOI: [10.1137/23M1626414](https://doi.org/10.1137/23M1626414).
- [BO90] U. Banerjee and J. E. Osborn. “Estimation of the effect of numerical integration in finite element eigenvalue approximation”. *Numer. Math.* 56.8 (1990), pp. 735–762. DOI: [10.1007/BF01405286](https://doi.org/10.1007/BF01405286).
- [Bra07] D. Braess. *Finite elements*. Third. Theory, fast solvers, and applications in elasticity theory, Translated from the German by Larry L. Schumaker. Cambridge University Press, Cambridge, 2007, pp. xviii+365. DOI: [10.1017/CB09780511618635](https://doi.org/10.1017/CB09780511618635).
- [BS08] S. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*. 3rd. Vol. 15. Texts in Applied Mathematics. Springer, 2008. DOI: [10.1007/978-0-387-75934-0](https://doi.org/10.1007/978-0-387-75934-0).
- [CGH14] C. Carstensen, D. Gallistl, and J. Hu. “A discrete Helmholtz decomposition with Morley finite element functions and the optimality of adaptive finite element schemes”. *Comput. Math. Appl.* 68.12 (2014), pp. 2167–2181. DOI: [10.1016/j.camwa.2014.07.019](https://doi.org/10.1016/j.camwa.2014.07.019).
- [CH21] C. Carstensen and J. Hu. “Hierarchical Argyris finite element method for adaptive and multigrid algorithms”. *Comput. Methods Appl. Math.* 21.3 (2021), pp. 529–556. DOI: [10.1515/cmam-2021-0083](https://doi.org/10.1515/cmam-2021-0083).
- [Cia02] P. Ciarlet. *The Finite Element Method for Elliptic Problems*. Vol. 40. Classics in Applied Mathematics. Reprint of the 1978 original [North-Holland, Amsterdam]. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002, pp. xxviii+530. DOI: [10.1137/1.9780898719208](https://doi.org/10.1137/1.9780898719208).
- [CP23] C. Carstensen and S. Puttkammer. “Direct guaranteed lower eigenvalue bounds with optimal a priori convergence rates for the bi-Laplacian”. *SIAM J. Numer. Anal.* 61.2 (2023), pp. 812–836. DOI: [10.1137/21M139921X](https://doi.org/10.1137/21M139921X).
- [DBP11] K. Deckers, A. Bultheel, and F. Perdomo-Pío. “Rational Gauss-Radau and rational Szegő-Lobatto quadrature on the interval and the unit circle respectively”. *Jaen J. Approx.* 3.1 (2011), pp. 15–66.
- [Döl+20] J. Dölz et al. “Bembel: The fast isogeometric boundary element C++ library for Laplace, Helmholtz, and electric wave equation”. *SoftwareX* 11 (2020), p. 100476. DOI: <https://doi.org/10.1016/j.softx.2020.100476>.
- [DST24] L. Diening, J. Storn, and T. Tscherpel. *Code - Exact Integration for singular Zienkiewicz and Guzmán—Neilan finite elements*. <https://www.codeocean.com/>. Version v1. Oct. 2024. DOI: [10.24433/CO.0647291.v1](https://doi.org/10.24433/CO.0647291.v1).
- [DY86] R. A. DeVore and X. M. Yu. “Multivariate rational approximation”. *Trans. Amer. Math. Soc.* 293.1 (1986), pp. 161–169. DOI: [10.2307/2000277](https://doi.org/10.2307/2000277).
- [FGNZ22] M. Fabien, J. Guzmán, M. Neilan, and A. Zytoon. “Low-order divergence-free approximations for the Stokes problem on Worsley-Farin and Powell-Sabin splits”. *Comput. Methods Appl. Mech. Engrg.* 390 (2022), Paper No. 114444, 21. DOI: [10.1016/j.cma.2021.114444](https://doi.org/10.1016/j.cma.2021.114444).
- [FN13] R. S. Falk and M. Neilan. “Stokes complexes and the construction of stable finite elements with pointwise mass conservation”. *SIAM J. Numer. Anal.* 51.2 (2013), pp. 1308–1326. DOI: [10.1137/120888132](https://doi.org/10.1137/120888132).
- [FPW11] S. Funken, D. Praetorius, and P. Wissgott. “Efficient implementation of adaptive P1-FEM in Matlab”. *Comput. Methods Appl. Math.* 11.4 (2011), pp. 460–490. DOI: [doi:10.2478/cmam-2011-0026](https://doi.org/10.2478/cmam-2011-0026).
- [Gau01] W. Gautschi. “The use of rational functions in numerical quadrature”. *Proceedings of the Fifth International Symposium on Orthogonal Polynomials, Special Functions and their Applications (Patras, 1999)*. Vol. 133. 1-2. 2001, pp. 111–126. DOI: [10.1016/S0377-0427\(00\)00637-3](https://doi.org/10.1016/S0377-0427(00)00637-3).

- [Gau93] W. Gautschi. “Gauss-type quadrature rules for rational functions”. *Numerical integration, IV (Oberwolfach, 1992)*. Vol. 112. Internat. Ser. Numer. Math. Birkhäuser, Basel, 1993, pp. 111–130. DOI: [10.1007/978-3-0348-6338-4_9](https://doi.org/10.1007/978-3-0348-6338-4_9).
- [GLRW12] K. J. Galvin, A. Linke, L. G. Rebholz, and N. E. Wilson. “Stabilizing poor mass conservation in incompressible flow problems with large irrotational forcing and application to thermal convection”. *Comput. Methods Appl. Mech. Engrg.* 237/240 (2012), pp. 166–176. DOI: [10.1016/j.cma.2012.05.008](https://doi.org/10.1016/j.cma.2012.05.008).
- [GM78] A. Grundmann and H. M. Möller. “Invariant integration formulas for the n -simplex by combinatorial methods”. *SIAM J. Numer. Anal.* 15.2 (1978), pp. 282–290. DOI: [10.1137/0715019](https://doi.org/10.1137/0715019).
- [GN14a] J. Guzmán and M. Neilan. “Conforming and divergence-free Stokes elements in three dimensions”. *IMA J. Numer. Anal.* 34.4 (2014), pp. 1489–1508. DOI: [10.1093/imanum/drt053](https://doi.org/10.1093/imanum/drt053).
- [GN14b] J. Guzmán and M. Neilan. “Conforming and divergence-free Stokes elements on general triangular meshes”. *Math. Comp.* 83.285 (2014), pp. 15–36. DOI: [10.1090/S0025-5718-2013-02753-6](https://doi.org/10.1090/S0025-5718-2013-02753-6).
- [GN18] J. Guzmán and M. Neilan. “inf-sup stable finite elements on barycentric refinements producing divergence-free approximations in arbitrary dimensions”. *SIAM J. Numer. Anal.* 56.5 (2018), pp. 2826–2844. DOI: [10.1137/17M1153467](https://doi.org/10.1137/17M1153467).
- [HCB05] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement”. *Comput. Methods Appl. Mech. Engrg.* 194.39-41 (2005), pp. 4135–4195. DOI: [10.1016/j.cma.2004.10.008](https://doi.org/10.1016/j.cma.2004.10.008).
- [IR72] B. M. Irons and A. Razzaque. “Experience with the patch test for convergence of finite elements”. *The mathematical foundations of the finite element method with applications to partial differential equations (Proc. Sympos., Univ. Maryland, Baltimore, Md., 1972)*. Academic Press, New York-London, 1972, pp. 557–587.
- [JLMNR17] V. John, A. Linke, C. Merdon, M. Neilan, and L. G. Rebholz. “On the divergence constraint in mixed finite element methods for incompressible flows”. *SIAM Rev.* 59.3 (2017), pp. 492–544. DOI: [10.1137/15M1047696](https://doi.org/10.1137/15M1047696).
- [KPW19] R. Kruse, N. Polydorides, and Y. Wu. *Application of Randomized Quadrature Formulas to the Finite Element Method for Elliptic Equations*. 2019. arXiv: [1908.08901](https://arxiv.org/abs/1908.08901) [[math.NA](https://arxiv.org/abs/1908.08901)].
- [KVZ21] C. Kreuzer, R. Verfürth, and P. Zanotti. “Quasi-optimal and pressure robust discretizations of the Stokes equations by moment- and divergence-preserving operators”. *Comput. Methods Appl. Math.* 21.2 (2021), pp. 423–443. DOI: [10.1515/cmam-2020-0023](https://doi.org/10.1515/cmam-2020-0023).
- [Lin09] A. Linke. “Collision in a cross-shaped domain—a steady 2d Navier-Stokes example demonstrating the importance of mass conservation in CFD”. *Comput. Methods Appl. Mech. Engrg.* 198.41-44 (2009), pp. 3278–3286. DOI: [10.1016/j.cma.2009.06.016](https://doi.org/10.1016/j.cma.2009.06.016).
- [LLMS17] P. L. Lederer, A. Linke, C. Merdon, and J. Schöberl. “Divergence-free reconstruction operators for pressure-robust Stokes discretizations with continuous pressure finite elements”. *SIAM J. Numer. Anal.* 55.3 (2017), pp. 1291–1314. DOI: [10.1137/16M1089964](https://doi.org/10.1137/16M1089964).
- [Mat24] The MathWorks Inc. *MATLAB version: 24.2.0.2712019 (R2024b)*. Natick, Massachusetts, United States, 2024.
- [MS16] C. Manni and H. Speleers. “Standard and non-standard CAGD tools for isogeometric analysis: a tutorial”. *Isogeometric analysis: a new paradigm in*

- the numerical approximation of PDEs*. Vol. 2161. Lecture Notes in Math. Springer, [Cham], 2016, pp. 1–69.
- [MS75] J. Morgan and R. Scott. “A nodal basis for C^1 piecewise polynomials of degree $n \geq 5$ ”. *Math. Comput.* 29 (1975), pp. 736–740. DOI: [10.2307/2005284](https://doi.org/10.2307/2005284).
- [New64] D. J. Newman. “Rational approximation to $|x|$ ”. *Michigan Math. J.* 11 (1964), pp. 11–14.
- [Sch15] M. Schneier. “Numerical integration of rational bubble functions with multiple singularities”. *Involve* 8.2 (2015), pp. 233–251. DOI: [10.2140/involve.2015.8.233](https://doi.org/10.2140/involve.2015.8.233).
- [Shi84] Z. C. Shi. “The generalized patch test for Zienkiewicz’s triangles”. *J. Comput. Math.* 2.3 (1984), pp. 279–286.
- [SS97] S. A. Sauter and C. Schwab. “Quadrature for hp -Galerkin BEM in \mathbf{R}^3 ”. *Numer. Math.* 78.2 (1997), pp. 211–258. DOI: [10.1007/s002110050311](https://doi.org/10.1007/s002110050311).
- [ST24] L. R. Scott and T. Tscherpel. “Dimensions of exactly divergence-free finite element spaces in 3D”. *SIAM J. Sci. Comput.* 46.2 (2024), A1102–A1131. DOI: [10.1137/22M1544579](https://doi.org/10.1137/22M1544579).
- [Str71] A. H. Stroud. *Approximate calculation of multiple integrals*. Prentice-Hall Series in Automatic Computation. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971, pp. xiii+431.
- [SV85] L. R. Scott and M. Vogelius. “Norm estimates for a maximal right inverse of the divergence operator in spaces of piecewise polynomials”. *RAIRO Modél. Math. Anal. Numér.* 19.1 (1985), pp. 111–143. DOI: [10.1051/m2an/1985190101111](https://doi.org/10.1051/m2an/1985190101111).
- [TNW21] L. N. Trefethen, Y. Nakatsukasa, and J. A. C. Weideman. “Exponential node clustering at singularities for rational approximation, quadrature, and PDEs”. *Numer. Math.* 147.1 (2021), pp. 227–254. DOI: [10.1007/s00211-020-01168-2](https://doi.org/10.1007/s00211-020-01168-2).
- [WSX07] M. Wang, Z.-c. Shi, and J. Xu. “A new class of Zienkiewicz-type non-conforming element in any dimensions”. *Numer. Math.* 106.2 (2007), pp. 335–347. DOI: [10.1007/s00211-007-0063-4](https://doi.org/10.1007/s00211-007-0063-4).
- [Zie71] O. C. Zienkiewicz. *The finite element method in engineering science*. The second, expanded and revised, edition of *The finite element method in structural and continuum mechanics*. McGraw-Hill, London-New York-Düsseldorf, 1971, pp. xiv+521.

(L. Dienes) DEPARTMENT OF MATHEMATICS, BIELEFELD UNIVERSITY, POSTFACH 10 01 31, 33501 BIELEFELD, GERMANY

Email address: lars.diening@uni-bielefeld.de

(J. Storn) FACULTY OF MATHEMATICS & COMPUTER SCIENCE, INSTITUTE OF MATHEMATICS, LEIPZIG UNIVERSITY, AUGUSTUSPLATZ 10, 04109 LEIPZIG, GERMANY

Email address: johannes.storn@uni-leipzig.de

(T. Tscherpel) DEPARTMENT OF MATHEMATICS, TECHNISCHE UNIVERSITÄT DARMSTADT, DO-LIVOSTRASSE 15, 64293 DARMSTADT, GERMANY

Email address: tscherpel@mathematik.tu-darmstadt.de