HIGH-ORDER NUMERICAL INTEGRATION OVER DISCRETE SURFACES[*]

NAVAMITA RAY[†], DUO WANG[†], XIANGMIN JIAO[‡], AND JAMES GLIMM[†]

Abstract. We consider the problem of numerical integration of a function over a discrete surface to high-order accuracy. Surface integration is a common operation in numerical computations for scientific and engineering problems. Integration over discrete surfaces (such as a surface triangulation) is typically limited to first- or second-order accuracy due to the piecewise linear approximations of the surface and the function. We present a novel method that can achieve third- and higher-order accuracy for integration over discrete surfaces. Our method combines a stabilized least squares approximation, a blending procedure based on linear shape functions, and high-degree quadrature rules. We present theoretical analysis of the accuracy of our method as well as experimental results of up to sixth-order accuracy with our method.

Key words. high-order accuracy, numerical integration, surface integrals, surface reconstruction

AMS subject classifications. 65D15, 65D30

DOI. 10.1137/110857404

1. Introduction.

Surface integration is a fundamental operation for many scientific and engineering problems. It is a core procedure for a variety of numerical methods, such as the boundary integral method, the finite element method, surface finite elements, integral transforms, the finite volume method, etc. For example, in the boundary integral method the solution is obtained by solving an integral equation, which in turn is solved by forming a linear system using collocation methods. The entries in the linear system are surface integrals. In geometric processing, computing the surface area and solid volume are fundamental primitives, both of which require surface integration. In computational fluid dynamics, the computation of the flux across a curved interface between different materials requires surface integrals. The surface finite element method, which is a special case of the finite element method for solving partial differential equations on surfaces, also requires surface integrals. The application of surface integration is also found in fluid-structure interactions, where the integral of the pressure over the structure is the force applied by the fluid to the structure. Surface integrals also appear in surface, interface and colloidal sciences as well as in the semiconductor industry and for pharmaceutical manufacturing. They are helpful in understanding various processes, such as adhesion and fracture processes as well as in the manipulation of nanoscale objects.

For these applications, the standard method for surface integration is to integrate over the individual triangles, but its accuracy is limited by the piecewise linear approximations to the geometry and the function. The relative error is reduced by either global or adaptive mesh refinement until a set tolerance is reached. A natural, fundamental question is whether we can achieve high-order accuracy given a piece-

[*] Received by the editors December 1, 2011; accepted for publication (in revised form) September 17, 2012; published electronically November 27, 2012. This work was supported in part by the National Science Foundation under award DMS-0809285, by the DOE NEUP program under contract DE-AC07-05ID14517, and by DoD-ARO under contract W911NF0910306.

http://www.siam.org/journals/sinum/50-6/85740.html

[†] Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794 (nray@ams.sunysb.edu, duowang@ams.sunysb.edu, glimm@ams.sunysb.edu).

[‡] Corresponding author. Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794 (jiao@ams.sunysb.edu).

wise linear approximation to the surface. In this paper we investigate the problem of accurate numerical integration of a function over a triangulated curved surface.

Numerical integration might appear to be an easy problem, as integration in general is a smoothing process and tends to be more stable than differentiation. Although low-order integration schemes are easy, high-order integration schemes are as difficult as high-order differentiation schemes, if not more so. For these problems, numerical integration over discrete surfaces is more than just quadrature (or cubature) rules. The reason is that accurate numerical integration requires high-order approximations (or reconstructions) of the geometry as well as that of the function. Therefore, high-order numerical integration over surfaces is still a research issue.

We review a few approaches that are generally used to get high-order approximation of surface integrals. In finite element methods, high-order approximations are achieved by using high-order isoparametric elements [13]. Unfortunately, in problems such as fluid simulations involving moving interfaces, high-order elements are rarely used due to difficulties in the generation and adaptation of high-order finite element meshes. Meshless methods, such as moving least squares, offer another set of alternatives for constructing high-order approximations, but accurate numerical integration over such surfaces is subtle [1], and meshless methods are in general expensive in terms of computational cost. In [3], Chien presented a method for high-order surface integration using a high-order interpolated function and geometry. He approximated the function and the surface based on high-order piecewise interpolants using Lagrange polynomials. The points chosen for high-order interpolation are based on a uniform subdivision of individual triangles. His method requires the function values to be available at any point on the surface, and it becomes inapplicable when only a discrete set of values of the function is given. Also, Lagrange polynomial interpolation tends to be unstable for high-degree polynomials.

Another approach is presented in [5], in which adaptive refinement of the mesh is used to minimize relative error along with simplification of the trapezoidal rule, which is second-order accurate, though the relative error can be minimized to a set tolerance using adaptive mesh refinement. An assumption of that method is that the surface is represented as an implicit function, so that a retraction can be defined from a neighborhood of the triangulated surface to the exact surface to obtain a better approximation of the exact surface.

In this paper, we propose a novel method for accurate surface integration over discrete surfaces. Our techniques can deliver higher convergence rates, both in theory and in practice, than one would expect from the given piecewise linear approximations. Our method has three key components: a stabilized least squares fitting to obtain higher-order approximation of the geometry and the integrand, a blending procedure based on linear finite element shape functions, and high-degree numerical quadrature rules. Our resulting algorithm is relatively simple compared to the previous alternatives, and it is efficient. We demonstrate experimental results of up to sixth-order accuracy with our method.

The remainder of the paper is organized as follows. Section 2 reviews some background on numerical integration. Section 3 presents the theoretical framework for high-order reconstruction of surfaces and functions over triangulated surfaces. Section 4 describes our proposed method for high-order surface integration by combining the techniques described earlier, and section 5 analyzes the accuracy of the method. Section 6 presents numerical experiments to verify the theoretical results. Section 7 concludes the paper with a discussion of future work.

2. Background.

In this section, we review concepts for smooth surfaces in differential calculus and differential geometry, which will serve as the foundation of our method for discrete surfaces.

2.1. Integration of continuous functions over smooth surfaces.

Let a parametrization of a smooth surface Γ be given as $\Gamma = \{\boldsymbol{x}(\boldsymbol{\xi}) \mid \mathrm{U}(\boldsymbol{\xi}) \subset \mathbb{R}^2 \to \boldsymbol{x} \subset \mathbb{R}^3\}$ with coordinates $\boldsymbol{\xi} \in \mathbb{R}^2$, $\boldsymbol{x} \in \mathbb{R}^3$, and Jacobian $\boldsymbol{J} = \partial \boldsymbol{x}/\partial \boldsymbol{\xi}$, where

$$\boldsymbol{\xi} \equiv \begin{bmatrix} \xi \\ \eta \end{bmatrix}, \quad \boldsymbol{x} \equiv \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \boldsymbol{J} \equiv \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} \end{bmatrix}.$$

Let g denote $\sqrt{\det(\boldsymbol{J}^T\boldsymbol{J})}$, which is analogous to the absolute value of the Jacobian determinant if the surface were in the xy-plane. The surface integral of a scalar function $\varphi : \Gamma \to \mathbb{R}$ is

$$\int_\Gamma \varphi \, da \equiv \int_U \varphi(\boldsymbol{\xi}) g(\boldsymbol{\xi}) d\xi \, d\eta. \tag{1}$$

In the following discussion, we will omit the function arguments $\boldsymbol{\xi}$ for conciseness. The simplest application of (1) is the surface area $A = \int_\Gamma \, da$, where $\varphi = 1$. Similarly, given a vector-valued function $\boldsymbol{\varphi} : \Gamma \to \mathbb{R}^3$, the surface integral of $\boldsymbol{\varphi}$ is

$$\int_\Gamma \boldsymbol{\varphi} \cdot d\boldsymbol{a} \equiv \int_\Gamma \boldsymbol{\varphi} \cdot \hat{\boldsymbol{n}} da = \int_U \boldsymbol{\varphi} \cdot \hat{\boldsymbol{n}} g d\xi \, d\eta, \tag{2}$$

where $\hat{\boldsymbol{n}}$ denotes the unit surface normal. Let $\boldsymbol{j}_k$ denote the kth column of $\boldsymbol{J}$ for $k = 1, 2$. Assume $\boldsymbol{j}_1$, $\boldsymbol{j}_2$, and $\hat{\boldsymbol{n}}$ form a right-hand system, and let $\boldsymbol{n} \equiv g\hat{\boldsymbol{n}} = \boldsymbol{j}_1 \times \boldsymbol{j}_2$, which we refer to as the *Jacobian-weighted normal*. The surface integral of $\boldsymbol{\varphi}$ can be written as

$$\int_\Gamma \boldsymbol{\varphi} \cdot d\boldsymbol{a} = \int_U \boldsymbol{\varphi} \cdot (\boldsymbol{j}_1 \times \boldsymbol{j}_2) d\xi \, d\eta. \tag{3}$$

The simplest application of (3) is the volume $V = \int_\Gamma \boldsymbol{x} \cdot d\boldsymbol{a}/3$ for a closed surface Γ, where $\boldsymbol{\varphi} = \boldsymbol{x}/3$. Generally, a global parameterization of the whole surface is computationally difficult. To overcome this difficulty, the surface may be decomposed into nonoverlapping regions. Let Γ be decomposed into nonoverlapping regions σ_i whose union is Γ, i.e., $\Gamma = \bigcup \sigma_i$. The previous formulas then become

$$\int_\Gamma \varphi da = \sum_i \int_{\sigma_i} \varphi g d\xi \, d\eta$$

and

$$\int_\Gamma \boldsymbol{\varphi} \cdot d\boldsymbol{a} = \sum_i \int_{\sigma_i} \boldsymbol{\varphi} \cdot (\boldsymbol{j}_1 \times \boldsymbol{j}_2) d\xi \, d\eta,$$

where the integral over σ_i can be computed using local parameterizations of σ_i. Our computations for triangulated surfaces will approximate these formulas based on local parameterizations.

From (1) and (3), it is clear that the surface integral of a scalar or vector-valued function requires the Jacobian of the surface. The computation of the Jacobian is significantly simplified if we transform the surface from the global xyz coordinate system onto a local uvw coordinate system. Assume both xyz and uvw coordinate frames are orthonormal right-hand systems. Let the origin of the local frame be at $\boldsymbol{x}_0$. Let $\hat{\boldsymbol{t}}_1$ and $\hat{\boldsymbol{t}}_2$ be unit vectors in the xyz coordinate system along the positive directions of the u- and v-axes, respectively, and $\hat{\boldsymbol{m}} = \hat{\boldsymbol{t}}_1 \times \hat{\boldsymbol{t}}_2$ be the unit vector along the positive w direction. Let $\boldsymbol{T} = [\,\boldsymbol{t}_1 \,|\, \boldsymbol{t}_2\,]$ denote the matrix consisting of the unit vectors in the tangent plane and $\boldsymbol{Q}$ the rotation matrix $[\,\hat{\boldsymbol{t}}_1 \,|\, \hat{\boldsymbol{t}}_2 \,|\, \hat{\boldsymbol{m}}\,]$. Any point $\boldsymbol{x}$ on Γ is then transformed to a point $\left[\begin{smallmatrix} u \\ v \\ f(u,v) \end{smallmatrix}\right] \equiv \boldsymbol{Q}^T(\boldsymbol{x} - \boldsymbol{x}_0)$. We refer to $f(u,v)$ as the *height function* in the uvw coordinate frame. In general, f is not a one-to-one mapping over the whole surface, but if the uv plane is not too far from the tangent plane at a point $\boldsymbol{x} \in \Gamma$ close to $\boldsymbol{x}_0$, f would be one-to-one in the neighborhood of $\boldsymbol{x}$.

Let $\boldsymbol{p}(u,v) = [u, v, f(u,v)]^T$ denote the points on the surface Γ in the uvw coordinate frame. Let $\nabla f \equiv [f_u, f_v]^T$ denote the gradient of f with respect to $\boldsymbol{u} \equiv (u,v)$. The Jacobian of $\boldsymbol{p}$ with respect to $\boldsymbol{u}$ is then

$$\boldsymbol{J} = [\,\boldsymbol{p}_u |\, \boldsymbol{p}_v] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ f_u & f_v \end{bmatrix}. \tag{4}$$

The vectors $\boldsymbol{p}_u$ and $\boldsymbol{p}_v$ form a basis of the tangent space of the surface at $\boldsymbol{p}$, but they may not be orthogonal to each other. Then, $g = \sqrt{1 + f_u^2 + f_v^2}$. The Jacobian-weighted normal and unit normal in the xyz coordinate system are then

$$\boldsymbol{n} = \boldsymbol{Q} \begin{bmatrix} -f_u \\ -f_v \\ 1 \end{bmatrix} = \hat{\boldsymbol{m}} - f_u \hat{\boldsymbol{t}}_1 - f_v \hat{\boldsymbol{t}}_2 \text{ and } \hat{\boldsymbol{n}} = \frac{\boldsymbol{n}}{g}. \tag{5}$$

These formulas are exact for smooth surfaces. In the next section, we will present their high-order approximations and their use in surface integration for triangulated surfaces.

2.2. Quadrature rules for integrating functions. Integration in the continuum is defined as a limiting process. However, in many practical applications numerical integration requires the use of quadrature rules,[1] which are based on polynomial interpolations and approximate an integral as a weighted sum of function values at some quadrature points. In general, a degree-d quadrature rule is exact for degree-d polynomials, and it is accurate for integrands that are continuously differentiable to dth derivatives.

As a numerical problem, integration is in general well-conditioned because of its smoothing effect. Let the integration of $\varphi : \Gamma \subseteq \mathbb{R}^d \to \mathbb{R}$ be given by

$$I\varphi = \int_\Gamma \varphi(\boldsymbol{x}) d\boldsymbol{x}. \tag{6}$$

The error in $I\varphi$ is bounded by the inequality

$$\|I\widetilde{\varphi} - I\varphi\| \le K_\varphi \|\widetilde{\varphi} - \varphi\|_\infty, \tag{7}$$

where $\widetilde{\varphi}$ denotes an approximation of φ and $K_\varphi = \text{area}(\Gamma)$ is the condition number

[1] In mathematics, the generalization of quadrature rules to two or higher dimensions is called *cubature rules*, but it is also often referred to as *high-dimensional quadrature rules*. We use the term *quadrature rules* in this paper, as the distinction between numerical integrations in one dimension and higher dimensions is immaterial algorithmically.

of $I\varphi$ with respect to changes in φ. Therefore, integrating an approximated function does not change the result drastically. The condition number of a quadrature rule is the sum of the absolute value of its weights. If all the weights are positive, then the condition number of the quadrature rule is equal to that of $I\varphi$. However, numerical integration also has an inherent uncertainty due to an infinite number of choices for $\widetilde{\varphi}$ that approximate φ. The distance between $\widetilde{\varphi}$ and φ could be arbitrary, so the error in the numerical integration may be arbitrarily large for an improper choice of $\widetilde{\varphi}$.

In one dimension it suffices to define quadrature rules on a "standard" interval, as any other interval could be transformed to this standard interval. However, this is not the case for two or higher dimensions, since an arbitrary shape may not be mapped into a "standard" shape. For example, an annulus cannot be mapped into a triangle. Therefore, high-dimensional quadrature rules are defined over some primitive shapes, such as triangles or rectangles. In [10], a survey of quadrature rules over triangles can be found. These quadrature rules are then applied to a tessellation of the domain. In this paper, we assume the domain of integration is given by a triangulated surface, so we will utilize quadrature rules for triangles.

3. Local high-order reconstructions.

To achieve high-order accuracy, we must construct high-order approximations to the geometry and the integrand. Our approach is to reconstruct such approximations within each triangle, accomplished in two steps. First, we obtain a high-order reconstruction of both the surface and the integrand by performing a weighted linear least squares fitting at each vertex of the triangle. Second, we blend these local fittings for each vertex of a triangle using a weighted averaging based on linear finite-element shape functions. These blended local polynomial fittings form high-order approximation to the discrete surface (integrand). Finally, we compute the first-order differential quantities (normals and Jacobians) as well as the integrand at the quadrature points from the local fittings, which are then used by the quadrature rules. In this section, we describe the first step, i.e., the local high-order reconstructions for both the surface and the integrand, while paying special attention to the robustness of its solutions.

3.1. High-order surface reconstruction.

We construct the polynomial fitting at each vertex using a local orthonormal *uvw* coordinate frame, where the *uv* plane is nearly tangential to the surface, and the w coordinate is orthogonal to the *uv* plane and corresponds to the *height function* f. For accuracy, we require the w-axis be at least a first-order accurate approximation to the unit normal of the surface and it is computed by averaging face normals of all incident faces at the vertex. The polynomial fitting is then derived from the Taylor (or Maclaurin) series expansion. The height function f can be approximated to dth-order accuracy about the origin $\boldsymbol{u}_0 = (0, 0)$ as

$$f(\boldsymbol{u}) = \sum_{p=0}^{d} \sum_{j,k\geq 0}^{j+k=p} c_{jk} \frac{u^j v^k}{j!k!} + O(\|\boldsymbol{u}\|^{d+1}), \tag{8}$$

where $\boldsymbol{u} = (u, v)$ and $c_{jk} = \frac{\partial^{j+k}}{\partial u^j \partial v^k} f(\boldsymbol{0})$. We assume that f has $d+1$ continuous derivatives. In addition, we assume that a set of points is given near the origin of the local coordinate system, and these points sample a smooth surface. Substituting each sample point (u_i, v_i, f_i) into (8), we obtain the equation

$$\sum_{p=0}^{d} \sum_{j,k\geq 0}^{j+k=p} c_{jk} \frac{u_i^j v_i^k}{j!k!} \approx f(\boldsymbol{u}_i). \tag{9}$$

Suppose there are m known points, and let $n = (d+1)(d+2)/2$ be the number of coefficients c_{jk}, where $m \geq n$. We obtain an $m \times n$ (rectangular) linear system by replacing the approximate equality (9) by equality. Let us denote this rectangular linear system by $\boldsymbol{Ax} = \boldsymbol{b}$, whose solution would give the coefficients c_{jk} and in turn a polynomial approximation to f. We will discuss the robust solution of this rectangular linear system in the next subsection. If the origin is at a vertex of interest, this equation is then the same as the vertex-based polynomial fitting known as osculating jets [2]. We use this exact same procedure for constructing the polynomial fitting of the integrand φ or $\boldsymbol{\varphi}$. For a vector-valued function $\boldsymbol{\varphi}$, if it does not have specific physical or geometric meaning, which we assume in this paper, its individual components can be approximated independently of each other.

After obtaining c_{jk}, we then obtain the gradient and Hessian of the height function in the neighborhood of the point from their corresponding Taylor series. In particular, the gradient at $\boldsymbol{u}$ is given by

$$f_u(\boldsymbol{u}) \approx \sum_{p=0}^{d-1} \sum_{j,k\geq 0}^{j+k=p} c_{j+1,k} \frac{u^j v^k}{j!k!}, \tag{10}$$

$$f_v(\boldsymbol{u}) \approx \sum_{p=0}^{d-1} \sum_{j,k\geq 0}^{j+k=p} c_{j,k+1} \frac{u^j v^k}{j!k!}, \tag{11}$$

where the error is $O(\|\boldsymbol{u}\|^d)$; see [8] for a proof. From (10) and (11) we can obtain the Jacobian-weighted normal to the polynomial surface in the uvw coordinate system, i.e., $[-f_u, -f_v, 1]^T$.

By construction, a polynomial fitting is obtained for each vertex. Hence, the fittings at different vertices are independent of each other. In section 4, we will describe a simple procedure to blend these local fittings into a coherent surface. Hereafter, we describe the robust solution of the least squares fittings, based on the procedure described in [8].

3.1.1. Robust solution of least squares fits. The linear system obtained from (9) in general is rectangular and must be solved in a least squares sense. We pose the problem as a minimization of a weighted norm of the residual $\boldsymbol{Ax} - \boldsymbol{b}$, i.e.,

$$\min_{\boldsymbol{x}} \|\boldsymbol{Ax} - \boldsymbol{b}\|_{\boldsymbol{S}} \equiv \min_{\boldsymbol{x}} \|\boldsymbol{S}(\boldsymbol{Ax} - \boldsymbol{b})\|_2, \tag{12}$$

where $\boldsymbol{S} = \operatorname{diag}(w_1, w_2, \ldots, w_m)$ is diagonal, which we refer to as the *weighting matrix*. The weighting matrix $\boldsymbol{S}$ assigns priorities to different rows of the linear system corresponding to different points that are being fit. Note that $\boldsymbol{S}$ has no effect on the solution if $\boldsymbol{A}$ is a nonsingular square matrix, but different $\boldsymbol{S}$ would lead to different solutions for rectangular matrices. Let $\hat{\boldsymbol{m}}_{\boldsymbol{i}}$ denote a first-order approximation to the unit normal at the ith vertex (e.g., obtained by averaging face normals). In general, for the ith row corresponding to the ith point $\boldsymbol{x}_i$, it is desirable to assign its weight w_i to some larger value if $\boldsymbol{x}_i$ is close to the origin of the local coordinate system $\boldsymbol{x}_0$ or a smaller value (or even zero) of $\boldsymbol{x}_i$ is far away from $\boldsymbol{x}_0$ or its normal $\hat{\boldsymbol{m}}_i$ is too far from the normal $\hat{\boldsymbol{m}}_0$ at $\boldsymbol{x}_0$. In particular, we choose the weight at the ith vertex as

$$w_i = \frac{\gamma_i^+}{\left(\|\boldsymbol{u}_i\|^2/h + \epsilon\right)^{d/2}}, \tag{13}$$

where $\gamma_i^+ \equiv \max(0, \hat{\boldsymbol{m}}_i^T \hat{\boldsymbol{m}}_0)$, $h \equiv \sum_{i=1}^m \|\boldsymbol{u}_i\|^2/m$, and $\epsilon \approx 0.01$. The factor γ_i^+ serves as a safeguard against drastically changing normals for coarse meshes or nonsmooth

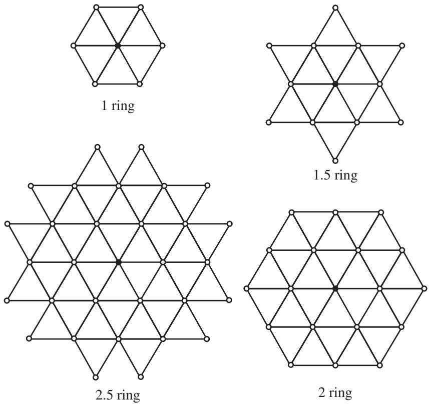

Fig. 1. *Stencil for point selection.*

areas. The denominator $(\|\boldsymbol{u}_i\|^2/h+\epsilon)^{d/2}$ prevents the weights from becoming too large at points that are too close to $\boldsymbol{u}_0$ and is approximately equal to $w_i \approx (\|\boldsymbol{u}_i\|^2/h)^{-d/2}$.

Because $\boldsymbol{S}$ allows the flexibility to underweigh (and even filter out) undesirable points, we use a simple procedure to select points based on mesh connectivity when constructing the linear system. In particular, we use a k-ring neighborhood with half-ring increments:

- The 1-*ring neighbor faces* of a vertex v are the faces incident on v, and the 1-*ring neighbor vertices* are the vertices of these faces.
- The 1.5-*ring neighbor faces* are the faces that share an edge with a 1-ring neighbor face, and the 1.5-*ring neighbor vertices* are the vertices of these faces.
- For an integer $k \geq 1$, the $(k+1)$-*ring neighborhood* of a vertex is the union of the 1-ring neighbors of its k-ring neighbor vertices, and the $(k+1.5)$-*ring neighborhood* is the union of the 1.5-ring neighbors of the k-ring neighbor vertices.

Figure 1 illustrates the definitions up to 2.5 rings. In general, for a dth-degree fitting, we use the $(d+1)/2$-ring for accurate input or the $(d/2+1)$-ring for relatively noisy input.

The coefficient matrix $\boldsymbol{A}$ in (9) is a generalized Vandermonde matrix which is ill-conditioned if u_i or v_j are all too far from unity, because its entries may vary substantially. The weighting matrix $\boldsymbol{S}$ scales the rows of $\boldsymbol{A}$ and hence cannot improve the poor scaling of the columns. This issue is resolved by introducing a column *scaling matrix* $\boldsymbol{T}$ and then imposing the minimization as

$$\min_{\boldsymbol{x}} \|\boldsymbol{SATy} - \boldsymbol{Sb}\|_2, \tag{14}$$

where $\boldsymbol{x} = \boldsymbol{Ty}$. Unlike $\boldsymbol{S}$, the scaling matrix $\boldsymbol{T}$ does not change the exact solution of $\boldsymbol{x}$. However, $\boldsymbol{T}$ can significantly improve the conditioning of the linear system and in turn improve the accuracy in the presence of rounding errors. In general, given a weighting matrix $\boldsymbol{S}$, let $\boldsymbol{v}_i$ denote the ith column vector of $\boldsymbol{SA}$. We choose $\boldsymbol{T}$ to be the diagonal matrix with entries

$$T_{ii} = 1/\|\boldsymbol{v}_i\|_2 \tag{15}$$

for $i = 1, \ldots, n$. This scaling matrix approximately minimizes the condition number of $\boldsymbol{SAT}$ (see [6, p. 265] and [11]). Note that the scaling matrix $\boldsymbol{T}$ cannot improve the condition number of $\boldsymbol{A}$ if the ill-conditioning is caused by the lack of points or some unfortunate selection of points, which is a well-known issue [9]. It can be alleviated by including additional points in the fitting if possible. However, if the number of points is fixed, solving this ill-conditioned system is similar to solving an under constrained linear system. Instead of using the truncated SVD to solve this linear system, we use a variant of the reduced QR factorization to address the problem. Let the reduced QR factorization of $\boldsymbol{SAT}$ be

$$\boldsymbol{SAT} = \boldsymbol{QR},$$

where $\boldsymbol{Q}$ is $m \times n$ with orthonormal column vectors and $\boldsymbol{R}$ is a $n \times n$ upper-triangular matrix. The condition number of $\boldsymbol{SAT}$ is the same as that of $\boldsymbol{R}$, which can be estimated accurately and efficiently using a variant of back substitution. If the condition number $\boldsymbol{R}$ is too large (e.g., $\geq 10^6$), we then reduce the degree of the fitting by removing the last few columns that correspond to the highest derivatives. Let $\tilde{\boldsymbol{Q}}$ and $\tilde{\boldsymbol{R}}$ denote the reduced matrices. The final solution of $\boldsymbol{x}$ is given by

$$\boldsymbol{x} = \boldsymbol{T}\tilde{\boldsymbol{R}}^{-1}\tilde{\boldsymbol{Q}}^{T}\boldsymbol{Sb}, \tag{16}$$

where $\tilde{\boldsymbol{R}}^{-1}$ denotes a back substitution step. Compared to the solution based on SVD, this procedure is more accurate asymptotically as it gives highest priority to the lower-order coefficients of the polynomial while maintaining good scaling of the matrix, and at the same time it is more efficient than SVD. If the degree of fitting is reduced due to either an ill-conditioned $\tilde{\boldsymbol{R}}$ or an insufficient number of points in the stencil, a cure could be increasing the size of the stencil so that the system is more stable.

3.2. High-order function reconstruction. We use a similar approach to reconstruct a smooth function defined over the surface. For a function defined at each vertex of the triangulated surface, a high-order approximation at any point inside a triangle can be obtained using local polynomial fitting at its vertices, as is done for the height function. For a function φ smooth over the surface, we can consider it as a function of the parameters $\boldsymbol{u} = (u, v)$. We expand φ into a Taylor (or Maclaurin) series about $\boldsymbol{u}_0 = (0, 0)$ similar to (8) as

$$\varphi(\boldsymbol{u}) = \sum_{p=0}^{d} \sum_{j,k\geq 0}^{j+k=p} c_{jk} \frac{u^j v^k}{j!k!} + O(\|\boldsymbol{u}\|^{d+1}). \tag{17}$$

Given a set of points in the stencil of $\boldsymbol{u}_0$ with known values for φ, we get

$$\sum_{p=0}^{d} \sum_{j,k\geq 0}^{j+k=p} c_{jk} \frac{u_i^j v_i^k}{j!k!} \approx \varphi(u_i, v_i). \tag{18}$$

Clearly, the only difference between high-order reconstruction of the height function f and the integrand φ is the right-hand side of the resulting linear system. The system given by (18) can be solved in the same manner as for the height function. In the case of a vector function, we perform high-order reconstruction for each component independently.

4. High-order numerical integration.

In this section, we describe the second step of our algorithm, i.e., to evaluate the first-order differential quantities (normals and Jacobians) and the function values at the quadrature points, which are required by the quadrature rules. The stabilized least squares fittings in the previous section are insufficient by themselves for this purpose, because the polynomial fittings at different vertices lead to ambiguities in the surface definition. We describe two approaches to overcome this ambiguity. Our first approach is to blend the polynomials at the vertices using standard linear finite element shape functions. The second approach is to take a weighted average of the estimates from each fitting. These two approaches have been used in [7] to obtain high-order reconstructions of surfaces. In this paper we extend the method for high-order estimates of geometric differential quantities and the interpolant.

4.1. High-order piecewise smooth geometry.

The key idea behind our methods is that the polynomial at each vertex gives a high-order approximation to the surface and its differential quantities in the stencil of the vertex. Therefore, any weighted average of these polynomials associated with the vertices of a triangle would also give a high-order approximation. However, the matter is complicated by the fact that different local coordinate frames are used at different vertices, so a change of coordinates is necessary.

Our approach is to combine the fittings using the shape functions of a triangle. Let ξ and η denote the natural coordinate of a triangle σ, and let N_i denote the shape function (or the barycentric coordinate) with respect to the ith vertex of σ, given by

$$\begin{aligned} N_1 &= 1 - \xi - \eta, \\ N_2 &= \xi, \\ N_3 &= \eta. \end{aligned}$$

From the polynomial fitting at each vertex of σ, we first compute a high-order approximation for an arbitrary point $\boldsymbol{q} \in \sigma$ with natural coordinate $\boldsymbol{\xi} = (\xi, \eta)$ within the triangle in the corresponding local uvw coordinate at the vertex. This is performed by interpolating the uv coordinates at $\boldsymbol{q}$ from those at the vertices and then evaluating the Taylor polynomial. The resulting point is then transformed into the global xyz coordinate system. For each point $\boldsymbol{q} \in \sigma$, let $\boldsymbol{p}_i(\boldsymbol{\xi})$ denote the reconstructed point associated with the ith vertex for $i = 1, 2, 3$. We then average these points using the shape functions. The geometry of the blended surface within the triangle is then defined by

$$\boldsymbol{p}(\boldsymbol{\xi}) = \sum_{i=1}^{3} N_i(\boldsymbol{\xi}) \boldsymbol{p}_i(\boldsymbol{\xi}). \tag{19}$$

For conciseness, we will drop out the parameters (ξ, η) in the notation. Following [7], we refer to the blended surface as the *WALF* (*weighted average of least squares fittings*) *surface*. It is clear that the blended function is infinitely differentiable within each triangle. In addition, it is C^0 continuous across the boundaries of triangles due to the continuity of the shape functions [7].

To perform integration, we need to compute the differential quantities such as the Jacobian and surface normal at the quadrature points. We compute the exact Jacobian or surface normal of the WALF surface. Again, let $\boldsymbol{Q}_i$ denote the rotation matrix $[\,\hat{\boldsymbol{t}}_{1i} \,|\, \hat{\boldsymbol{t}}_{2i} \,|\, \hat{\boldsymbol{m}}_i\,]$ for the ith vertex, where $\hat{\boldsymbol{m}}_i$ denotes the first-order approximation to the unit normal at the ith vertex, and let $\boldsymbol{T}_i$ denote the matrix composed of the first two columns of $\boldsymbol{Q}_i$. Let $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$ denote the vertices of the triangle in the xyz coordinate system, and let $\boldsymbol{J} = [\boldsymbol{x}_2 - \boldsymbol{x}_1 \mid \boldsymbol{x}_3 - \boldsymbol{x}_1]$ be the Jacobian of the linear triangle with respect to $\boldsymbol{\xi}$. Let $\boldsymbol{p}_i(\boldsymbol{\xi})$ denote the reconstructed point in the xyz coordinate system by the polynomial associated with the ith vertex. The Jacobian of the WALF-reconstructed triangle with respect to $\boldsymbol{\xi} = (\xi, \eta)$, denoted by $\boldsymbol{\nabla}_{\boldsymbol{\xi}}\boldsymbol{p}$, is then

$$\boldsymbol{\nabla}_{\boldsymbol{\xi}}\boldsymbol{p} = \sum_{i=1}^{3} \Big(N_i \left(\boldsymbol{T}_i + \hat{\boldsymbol{m}}_i \nabla_{\boldsymbol{u}} f_i \right) \boldsymbol{T}_i^T \boldsymbol{J} \Big) + [\,\boldsymbol{p}_2 - \boldsymbol{p}_1 | \,\boldsymbol{p}_3 - \boldsymbol{p}_1]. \tag{20}$$

The appendix shows the derivation of (20) for completeness. The normal of the blended surface can be obtained by taking the cross product of the two column vectors of $\boldsymbol{\nabla}_{\boldsymbol{\xi}}\boldsymbol{p}$.

4.2. High-order piecewise smooth function. When evaluating the functions at the quadrature points for numerical integration, if an analytical formula is available for the function, one can simply evaluate the analytical formula. However, in most cases the function values are available only at the vertices of the triangulated surface. The method in section 3.2 can reconstruct local polynomial fittings of the function at each vertex. Similar to surface reconstruction, we blend these local fittings using the shape function to reconstruct a piecewise smooth function. Let $\tilde{\varphi}_i(\boldsymbol{\xi})$ denote the function reconstructed from the polynomial at the ith vertex. The blended function within the triangle is then

$$\tilde{\varphi}(\boldsymbol{\xi}) = \sum_{i=1}^{3} N_i(\boldsymbol{\xi}) \tilde{\varphi}_i(\boldsymbol{\xi}).$$

When the function is a vector function, different components are blended independently. In the next section, we will analyze the accuracy of these reconstructions and also show that both high-order geometry and high-order function reconstructions are necessary for high-order accuracy of numerical integration.

4.3. Overall algorithm. After obtaining the high-order reconstructions of the surface and the integrand, the remainder of our algorithm is to apply numerical quadrature over each triangle. The general form of the quadrature over a triangle σ applied to scalar surface integrals is

$$\int_\sigma \varphi \; da = \int_\sigma \varphi(\boldsymbol{\xi}) g(\boldsymbol{\xi}) d\xi \, d\eta = \frac{1}{2} \sum_k w_k \varphi(\xi_k, \eta_k) g(\xi_k, \eta_k),$$

where (ξ_k, η_k) are the natural coordinates of the quadrature points within the triangle and w_k is the associated weight of the quadrature points. For a comprehensive list of quadrature rules, see [4]. (An online database is available at http://nines.cs.kuleuven.be/research/ecf/ecf.html.) The overall flow of our algorithm is shown in Figure 2.

The above method has many advantages. First, it is easy to implement, efficient, and robust. Second, it does not suffer from oscillations due to high-order polynomial interpolation. The least squares formulation mitigates any such oscillation that might have been present due to high-order polynomial interpolation. Another advantage of the method is that it is largely independent of mesh quality in the sense that it does not depend on the angles or Jacobian or other general quality attributes of a mesh. Thus we can still get high order for very-poor-quality meshes (subsection 6.1). There are

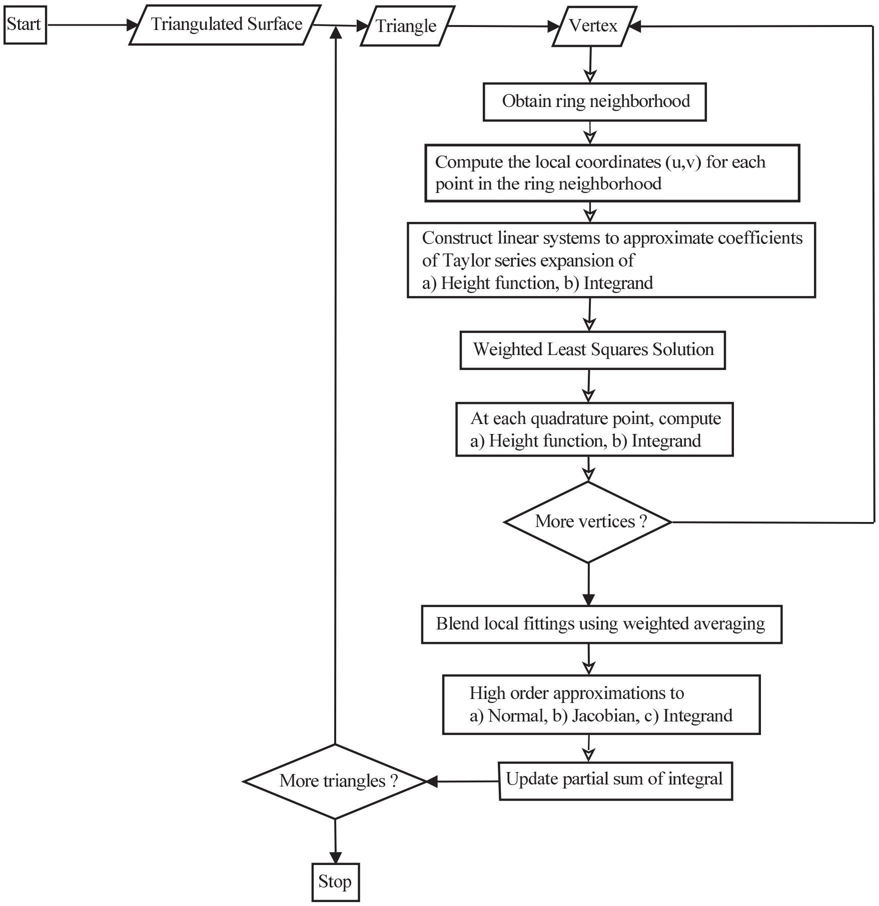
Start
Triangulated Surface
Triangle
Vertex
Obtain ring neighborhood
Compute the local coordinates (u,v) for each point in the ring neighborhood
Construct linear systems to approximate coefficients of Taylor series expansion of
a) Height function, b) Integrand
Weighted Least Squares Solution
At each quadrature point, compute
a) Height function, b) Integrand
More vertices ?
Blend local fittings using weighted averaging
High order approximations to
a) Normal, b) Jacobian, c) Integrand
Update partial sum of integral
More triangles ?
Stop

FIG. 2. *Flowchart of our algorithm for high-order surface integration, where w, φ, and g denote the weight, integrand, and Jacobian of the surface at a quadrature point. In case of vector functions, $\varphi = \boldsymbol{\varphi} \cdot \hat{\boldsymbol{n}}$.*

only two requirements imposed on the input mesh. First, the vertices approximate the exact surface up to machine precision. Second, there is a sufficient number of points in the stencils. The connectivity of the mesh is used mostly to form the stencils at each vertex about which the local polynomial fitting is to take place.

5. Convergence of proposed methods.

In this section, we analyze the order of accuracy of our proposed methods. Our results are given in terms of a local measure of the mesh resolution, such as the average edge length h in the triangulation. The main result of theoretical analysis is given by the following theorem and corollary.

THEOREM 1. *Let Γ be a smooth surface and φ be a scalar function defined over Γ. Let S denote the triangulation of Γ, and let $\widetilde{S}$ and $\tilde{\varphi}$ be the surface and function reconstructed from values at vertices on S using weighted average of local fittings of degree d, respectively. Then $|\int_\Gamma \varphi \, da - \int_{\widetilde{S}} \tilde{\varphi} \, da| = O(h^d + h^5)$.*

As a corollary of Theorem 1, we also obtain the following estimation of the errors in surface integrals of vector-valued functions.

COROLLARY 2. *Let Γ be a smooth surface and $\boldsymbol{\varphi} : \Gamma \to \mathbb{R}^3$ be a vector-valued function defined over Γ. Let S denote the triangulation of Γ, and let $\widetilde{S}$ and $\tilde{\boldsymbol{\varphi}} : \widetilde{S} \to \mathbb{R}^3$*

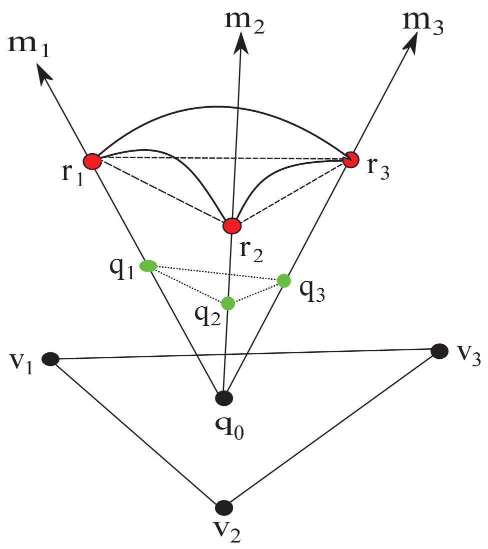

Fig. 3. *Illustration of notation used in proofs. Point $\boldsymbol{q}_0$ is a point inside a linear triangle $\triangle \boldsymbol{v_1 v_2 v_3}$. The reconstructed points corresponding to local fittings at vertices $\boldsymbol{v}_i$ are denoted by green points $\boldsymbol{q}_i$. The intersection of the first-order normals $\boldsymbol{m}_i$ with the exact surface is denoted by $\boldsymbol{r}_i$.*

be the reconstructed surface and the reconstructed function using weighted average of local fittings of degree d, respectively. Then $|\int_\Gamma \boldsymbol{\varphi} \cdot d\boldsymbol{a} - \int_{\tilde{S}} \tilde{\boldsymbol{\varphi}} \cdot d\boldsymbol{a}| = O(h^d + h^5)$.

When quadrature rules are used for the integrals numerically, in general it suffices to use degree-d quadrature rules over the triangles to preserve the accuracy. The proof of Theorem 1 is somewhat complex, as it involves analysis of the accuracies of the WALF-reconstructed surface, its approximate normal and Jacobian, and the reconstructed function. We will analyze these individual terms in the following sections and then use the results to prove the accuracy of the integration. Note that the barrier of $O(h^5)$ in the theorem above is somewhat pessimistic, due to technical difficulties in the analysis of the approximation to the Jacobian in section 5.2. In numerical experiments, we observe up to eighth-order convergence rate in our numerical experiments, as shown in section 6.

5.1. Accuracies of weighted averaging of fittings. Let Γ denote a smooth surface and S be a triangulation of Γ. Suppose the triangulation S is composed of triangles σ_i, i.e., $S = \cup_{i=1}^{K} \sigma_i$, where K is the number of triangles. Let $\tilde{\sigma}_i$ denote the high-order reconstruction of σ_i by WALF; then $\tilde{\Gamma} = \sum_{i=1}^{K} \tilde{\sigma}_i$.

For each point $\boldsymbol{q}_0 \in \sigma_i$ in S, let $\boldsymbol{v}_i$ be the ith vertex of σ_i, and let $\boldsymbol{m}_i$ denote a first-order approximation to the unit normal at $\boldsymbol{v}_i$ used in constructing the local uvw coordinate frame at $\boldsymbol{v}_i$. Let $\boldsymbol{q}_i$ denote the reconstructed point for $\boldsymbol{q}_0$ based on local fittings at the vertices $\boldsymbol{v}_i$. Let $\boldsymbol{q}$ be the reconstructed point of $\boldsymbol{q}_0$ on $\tilde{\Gamma}$, i.e., $\boldsymbol{q} = \sum_{i=1}^{3} N_i \boldsymbol{q}_i$. Let $\boldsymbol{r}_i$ denote the intersection of lines $\boldsymbol{q}_0\boldsymbol{q}_i$ with the exact surface Γ; see Figure 3 for an illustration of these points. We first have the following lemma regarding the distances between $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, which impose a lower bound for the minimum error that can be obtained by the weighted averaging scheme.

Lemma 3. *Assume that the local coordinate systems are aligned with approximate surface normals that are at least first-order accurate. The distance between points* $\|\boldsymbol{r}_i - \boldsymbol{r}_j\| = O(h^3)$, $i, j = 1, 2, 3$.

This lemma was a side product in the analysis of the order of accuracy of a WALF surface in [7]. For completeness, we give a separate proof as follows.

Proof. It suffices to prove for only $\|\boldsymbol{r}_1 - \boldsymbol{r}_2\|$. Let θ_i denote the angle at $\boldsymbol{r}_i$ in the triangle $\boldsymbol{q}_0\boldsymbol{r}_1\boldsymbol{r}_2$. We have

$$\begin{aligned}\|\boldsymbol{r}_1 - \boldsymbol{r}_2\| &= |\cos\theta_1 \|\boldsymbol{r}_1 - \boldsymbol{q}_0\| + \cos\theta_2 \|\boldsymbol{r}_2 - \boldsymbol{q}_0\||\\ &\le (|\cos\theta_1| + |\cos\theta_2|) \max_{i=1,2}\{\|\boldsymbol{r}_i - \boldsymbol{q}_0\|\}.\end{aligned}$$

Note that $|\cos\theta_1| = O(h)$, because $\boldsymbol{r}_1\boldsymbol{r}_2$ is at least a first-order approximation to the tangent direction at $\boldsymbol{r}_1$, and $\boldsymbol{m}_1$ is by assumption a first-order approximation to the normal direction at $\boldsymbol{v}_1$ and in turn also at $\boldsymbol{r}_1$. Similarly, $|\cos\theta_2| = O(h)$. Furthermore, $\|\boldsymbol{r}_i - \boldsymbol{q}_0\| = O(h^2)$, because $\boldsymbol{q}_0$ is a linear approximation to $\boldsymbol{r}_i$ in the local coordinate frame aligned with $\boldsymbol{m}_i$. Therefore, $\|\boldsymbol{r}_1 - \boldsymbol{r}_2\| \le O(h)O(h^2) = O(h^3)$. □

For error analysis, we need to define a mapping from $\tilde{\Gamma}$ to Γ. Let $\boldsymbol{r}$ be the projection of $\boldsymbol{q}$ onto Γ along the direction $\sum_{j=1}^{3} N_j\boldsymbol{m}_j$, and let Π denote the mapping from $\tilde{\Gamma}$ onto Γ, i.e., $\boldsymbol{r} = \Pi(\boldsymbol{q})$. Assume the triangulation S is a dense enough triangulation of Γ, so that the projection from $\tilde{\Gamma}$ to Γ is one-to-one and onto. Let γ_i denote the range of this projection from points in σ_i. Therefore, the triangulation S defines local parameterizations for both Γ and $\tilde{\Gamma}$ within each triangle. With this mapping, we obtain the following theorems regarding the reconstructed surface and function.

THEOREM 4. *Given a mesh whose vertices approximate a smooth surface Γ with an error of at most $O(h^{d+1})$, for each point $\boldsymbol{q}$ on the WALF surface obtained from degree-d fittings, $\|\boldsymbol{q} - \Pi(\boldsymbol{q})\| = O(h^{d+1} + h^6)$.*

Proof. Let $\boldsymbol{r} = \Pi(\boldsymbol{q})$. We have

$$\begin{aligned}\|\boldsymbol{q} - \boldsymbol{r}\| &= \left\|\sum_{i=1}^{3} N_i\boldsymbol{q}_i - \boldsymbol{r}\right\|\\ &= \left\|\sum_{i=1}^{3} N_i\boldsymbol{q}_i - \sum_{i=1}^{3} N_i\boldsymbol{r}_i + \sum_{i=1}^{3} N_i\boldsymbol{r}_i - \boldsymbol{r}\right\|\\ &\le \left(\sum_{i=1}^{3} N_i \|\boldsymbol{q}_i - \boldsymbol{r}_i\|\right) + \left\|\boldsymbol{r} - \sum_{i=1}^{3} N_i\boldsymbol{r}_i\right\|.\end{aligned}$$

Because $\boldsymbol{q}_i$ is a local fitting with dth degree polynomial at the local coordinate frame at the ith vertex of σ, $\|\boldsymbol{q}_i - \boldsymbol{r}_i\| = O(h^{d+1})$. For the second term, note that $\sum_{i=1}^{3} N_i\boldsymbol{r}_i$ is a linear approximation to $\boldsymbol{r}$ over triangle $\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{r}_3$ of length $O(h^3)$, and hence $\|\boldsymbol{r} - \sum_{i=1}^{3} N_i\boldsymbol{r}_i\| = O(h^6)$. Overall, $\|\boldsymbol{q} - \boldsymbol{r}\| = O(h^{d+1} + h^6)$. □

In the following, we generalize this result to functions reconstructed by weighted averaging of least squares fitting (WALF). We introduce some additional notation here. Let $\varphi : \Gamma \to \mathbb{R}$ be a scalar function defined on a smooth surface Γ and let $\tilde{\varphi} : \tilde{\Gamma} \to \mathbb{R}$ denote the WALF-reconstructed scalar function over $\tilde{\Gamma}$. More specifically, $\tilde{\varphi}$ is defined as follows. Let $\bar{\varphi}_i(\boldsymbol{x}) : S \to \mathbb{R}$ denote the local fitting of a point $\boldsymbol{x}$ on S at the ith vertex of a triangle σ containing $\boldsymbol{x}$. Let $\boldsymbol{\xi} = (\xi, \eta)$ denote the local parameters (i.e., the natural coordinates) of the points within σ and $\tilde{\boldsymbol{x}}$ denote the reconstructed point on $\tilde{\Gamma}$. Then,

$$\tilde{\varphi}(\tilde{\boldsymbol{x}}(\boldsymbol{\xi})) = \sum_{i=1}^{3} N_i(\boldsymbol{\xi})\bar{\varphi}_i(\boldsymbol{x}(\boldsymbol{\xi})),$$

where N_i denote the linear finite element shape function associated with the ith point. Using this notation, we have the following theorem.

THEOREM 5. *Given a mesh whose vertices approximate a smooth surface Γ with an error of at most $O(h^{d+1})$, let $\boldsymbol{r}$ be the projection of the point $\tilde{\boldsymbol{x}}$ on the reconstructed WALF surface with degree-d fittings onto the exact surface Γ. Then for the degree-d WALF-reconstructed function $\tilde{\varphi}(\tilde{\boldsymbol{x}})$, $|\tilde{\varphi}(\tilde{\boldsymbol{x}}) - \varphi(\boldsymbol{r})| = O(h^{d+1} + h^6)$.*

The proof for Theorem 5 shares some similarities with that for Theorem 4, but it is slightly more complicated because of the additional function φ. We give the proof as follows.

Proof. Note that $\boldsymbol{\xi}$ parameterizes $\tilde{x}(\boldsymbol{\xi})$ within a triangle σ. Then

$$\begin{aligned}
|\tilde{\varphi}(\tilde{\boldsymbol{x}}(\boldsymbol{\xi})) - \varphi(\boldsymbol{r})| &= \left|\sum_{i=1}^{3} N_i \bar{\varphi}_i(\boldsymbol{x}(\boldsymbol{\xi})) - \varphi(\boldsymbol{r})\right| \\
&= \left|\sum_{i=1}^{3} N_i \bar{\varphi}_i(\boldsymbol{x}(\boldsymbol{\xi})) - \sum_{i=1}^{3} N_i \varphi(\boldsymbol{r}_i) + \sum_{i=1}^{3} N_i \varphi(\boldsymbol{r}_i) - \varphi(\boldsymbol{r})\right| \\
&\le \left(\sum_{i=1}^{3} N_i \left|\bar{\varphi}_i(\boldsymbol{x}(\boldsymbol{\xi})) - \varphi(\boldsymbol{r}_i)\right|\right) + \left|\varphi(\boldsymbol{r}) - \sum_{i=1}^{3} N_i \varphi(\boldsymbol{r}_i)\right|.
\end{aligned}$$

Because $\bar{\varphi}_i$ is a local fitting with dth degree polynomial at the local coordinate frame at the ith vertex of σ, $|\bar{\varphi}_i(\boldsymbol{x}(\boldsymbol{\xi})) - \varphi(\boldsymbol{r}_i)| = O(h^{d+1})$. For the second term, note that $\sum_{i=1}^{3} N_i \varphi(\boldsymbol{r}_i)$ is a linear approximation to $\varphi(\boldsymbol{r})$ over triangle $\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{r}_3$ of length $O(h^3)$, and hence $|\varphi(\boldsymbol{r}) - \sum_{i=1}^{3} N_i \varphi(\boldsymbol{r}_i)| = O(h^6)$. □

5.2. Accuracies of Jacobian. The mapping Π from $\tilde{\Gamma}$ to Γ allows us to obtain the reference solution for the Jacobian as well as for the normals. For triangle $\tilde{\sigma}_i \in \tilde{\Gamma}$, let $\tilde{\boldsymbol{J}}_i$ denote the Jacobian matrix for the mapping from natural coordinates $\boldsymbol{\xi}$ of σ_i, i.e., $\tilde{\boldsymbol{J}}_i(\boldsymbol{q}(\boldsymbol{\xi})) = \left[\frac{\partial \boldsymbol{q}}{\partial \xi} \middle| \frac{\partial \boldsymbol{q}}{\partial \eta}\right]$. Similarly, for triangle $\gamma_i \in \Gamma$, let $\boldsymbol{J}_i$ denote the Jacobian matrix for the mapping from $\boldsymbol{\xi}$ to γ_i, i.e., $\boldsymbol{J}_i(\boldsymbol{r}(\boldsymbol{\xi})) = \left[\frac{\partial \boldsymbol{r}}{\partial \xi} \middle| \frac{\partial \boldsymbol{r}}{\partial \eta}\right]$. In numerical integration, these Jacobian matrices are used to compute the surface area as well as the surface normal. We first prove the following lemma regarding the Jacobian matrices.

LEMMA 6. *Let a_{ij} denote an entry of matrix $\tilde{\boldsymbol{J}}_{\boldsymbol{i}}(\boldsymbol{q}(\boldsymbol{\xi})) - \boldsymbol{J}_{\boldsymbol{i}}(\boldsymbol{r}(\boldsymbol{\xi}))$. Then*

$$|a_{ij}| = O(h^{d+1} + h^6).$$

Proof. Note that

$$\tilde{\boldsymbol{J}}_{\boldsymbol{i}}(\boldsymbol{q}(\boldsymbol{\xi})) - \boldsymbol{J}_{\boldsymbol{i}}(\boldsymbol{r}(\boldsymbol{\xi})) = \left[\frac{\partial(\boldsymbol{q}-\boldsymbol{r})}{\partial \xi} \middle| \frac{\partial(\boldsymbol{q}-\boldsymbol{r})}{\partial \eta}\right].$$

Due to Theorem 5, $\|\boldsymbol{q} - \boldsymbol{r}\| = O(h^{d+1} + h^6)$. The parameters ξ and η are always between 0 and 1 independently of h; therefore $\|\partial(\boldsymbol{q}-\boldsymbol{r})/\partial\xi\| = O(h^{d+1}+h^6)$. Similarly $\|\partial(\boldsymbol{q}-\boldsymbol{r})/\partial\eta\| = O(h^{d+1} + h^6)$. Hence for each entry a_{ij} in $\tilde{\boldsymbol{J}}_{\boldsymbol{i}} - \boldsymbol{J}_{\boldsymbol{i}}$, $|a_{ij}| = O(h^{d+1} + h^6)$. □

Let $\tilde{g}_i = \sqrt{\det(\tilde{\boldsymbol{J}}_i^T \tilde{\boldsymbol{J}}_{\boldsymbol{i}})}$ and $g_i = \sqrt{\det(\boldsymbol{J}_i^T \boldsymbol{J}_i)}$. Let $\tilde{\boldsymbol{n}}(\boldsymbol{q})$ denote the unit normal to $\tilde{\Gamma}$ at point $\boldsymbol{q}$, obtained by normalizing $\frac{\partial \boldsymbol{q}}{\partial \xi} \times \frac{\partial \boldsymbol{q}}{\partial \eta}$, and similarly let $\boldsymbol{n}(\boldsymbol{r})$ denote the unit normal to Γ at $\boldsymbol{r}$, obtained by normalizing $\frac{\partial \boldsymbol{r}}{\partial \xi} \times \frac{\partial \boldsymbol{r}}{\partial \eta}$. We consider $\tilde{g}_i$ as an approximation g_i and $\tilde{\boldsymbol{n}}(\boldsymbol{q})$ as an approximation to $\boldsymbol{n}(\boldsymbol{r})$. Because these quantities are obtained from simple arithmetic operations from $\tilde{\boldsymbol{J}}_i$ and $\boldsymbol{J}_i$, respectively, a direct consequence of Lemma 6 is the following theorem.

THEOREM 7. *$|\tilde{g}_i - g_i| = O(h^{d+2} + h^7)$ and $\|\tilde{\boldsymbol{n}}(\boldsymbol{q}) - \boldsymbol{n}(\boldsymbol{r})\| = O(h^d + h^5)$.*

Proof. From Lemma 6 $|a_{ij}| = O(h^{d+1} + h^6)$, we get $|\frac{\partial q_i}{\partial \xi} - \frac{\partial r_i}{\partial \xi}| = O(h^{d+1} + h^6)$ and $|\frac{\partial q_i}{\partial \eta} - \frac{\partial r_i}{\partial \eta}| = O(h^{d+1} + h^6)$, where $\boldsymbol{q} = (q_1, q_2, q_3)$ and $\boldsymbol{r} = (r_1, r_2, r_3)$. Now, $\tilde{g}_i = \|\frac{\partial \boldsymbol{q}}{\partial \xi} \times \frac{\partial \boldsymbol{q}}{\partial \eta}\|$ and $g_i = \|\frac{\partial \boldsymbol{r}}{\partial \xi} \times \frac{\partial \boldsymbol{r}}{\partial \eta}\|$. We note that each term of $\frac{\partial \boldsymbol{r}}{\partial \xi} \times \frac{\partial \boldsymbol{r}}{\partial \eta}$ is of the form $\frac{\partial r_k}{\partial \xi}\frac{\partial r_j}{\partial \eta} - \frac{\partial r_j}{\partial \xi}\frac{\partial r_k}{\partial \eta}$ for appropriate indices j and k. We also note that $\|\frac{\partial q_i}{\partial \xi}\| = O(h)$ as it should be at least a first-order approximation of a tangent direction at $\boldsymbol{q}$. Therefore,

$$\left\|\frac{\partial \boldsymbol{r}}{\partial \xi} \times \frac{\partial \boldsymbol{r}}{\partial \eta} - \frac{\partial \boldsymbol{q}}{\partial \xi} \times \frac{\partial \boldsymbol{q}}{\partial \eta}\right\| \approx O(h^{d+2} + h^7) \tag{21}$$

$$\left\|\frac{\partial \boldsymbol{r}}{\partial \xi} \times \frac{\partial \boldsymbol{r}}{\partial \eta}\right\| \approx \left\|\frac{\partial \boldsymbol{q}}{\partial \xi} \times \frac{\partial \boldsymbol{q}}{\partial \eta}\right\| + O(h^{d+2} + h^7),$$

i.e., $|\tilde{g}_i - g_i| = O(h^{d+2} + h^7)$. Similarly,

$$\begin{aligned}
\|\tilde{\boldsymbol{n}}(\boldsymbol{q}) - \boldsymbol{n}(\boldsymbol{r})\| &= \left\|\frac{1}{\tilde{g}_i}\left(\frac{\partial \boldsymbol{q}}{\partial \xi} \times \frac{\partial \boldsymbol{q}}{\partial \eta}\right) - \frac{1}{g_i}\left(\frac{\partial \boldsymbol{r}}{\partial \xi} \times \frac{\partial \boldsymbol{r}}{\partial \eta}\right)\right\| \\
&= \left\|\frac{g_i - \tilde{g}_i}{g_i\tilde{g}_i}\left(\frac{\partial \boldsymbol{q}}{\partial \xi} \times \frac{\partial \boldsymbol{q}}{\partial \eta}\right) + \frac{1}{g_i}\left(\frac{\partial \boldsymbol{q}}{\partial \xi} \times \frac{\partial \boldsymbol{q}}{\partial \eta} - \frac{\partial \boldsymbol{r}}{\partial \xi} \times \frac{\partial \boldsymbol{r}}{\partial \eta}\right)\right\| \\
&\le \left|\frac{g_i - \tilde{g}_i}{g_i\tilde{g}_i}\right| \left\|\left(\frac{\partial \boldsymbol{q}}{\partial \xi} \times \frac{\partial \boldsymbol{q}}{\partial \eta}\right)\right\| + \frac{1}{g_i}\left\|\frac{\partial \boldsymbol{q}}{\partial \xi} \times \frac{\partial \boldsymbol{q}}{\partial \eta} - \frac{\partial \boldsymbol{r}}{\partial \xi} \times \frac{\partial \boldsymbol{r}}{\partial \eta}\right\| \\
&= \frac{1}{|g_i|}\left(|\tilde{g}_i - g_i| + O(h^{d+2} + h^7)\right) \\
&= O(h^d + h^5),
\end{aligned}$$

where the last equality is because $|g_i| = O(h^2)$. □

Note that this result gives an upper bound of the errors in the approximation to the surface areas and the normal, and in turn they can bound the errors of the numerical integration. However, this approximation largely depends on our definition of the mapping Π from $\tilde{\Gamma}$ and Γ. There may be an "optimal" mapping for which the error may be bounded by $O(h^6)$ instead of $O(h^5)$. However, we are content with this result because $O(h^5)$ is a small enough error bound for almost all practical purposes.

5.3. Accuracies of integrations.

Using the above results, we can now prove our main theorem about the accuracy of integration, i.e., Theorem 1.

Proof. Suppose the triangulation S is composed of triangles σ_i, i.e., $S = \cup_{i=1}^{K}\sigma_i$, where K is the number of triangles. Let $\tilde{\sigma}_i$ denote the high-order reconstruction of σ_i by WALF; then $\tilde{S} = \sum_{i=1}^{K}\tilde{\sigma}_i$, and

$$\int_{\tilde{S}} \tilde{\varphi}\, da = \sum_{i=1}^{K}\int_{\tilde{\sigma}_i} \tilde{\varphi}\, da.$$

Using the parameterization in subsection 5.2, we have $\Gamma = \sum_{i=1}^{K}\gamma_i$, and

$$\int_{\Gamma} \varphi\, da = \sum_{i=1}^{K}\int_{\gamma_i} \varphi\, da.$$

Then

$$\int_{\Gamma} \varphi \, da = \sum_{i=1}^{K} \int_{\sigma_i} \varphi(\boldsymbol{\xi}) g_i(\boldsymbol{\xi}) \, d\xi \, d\eta,$$
$$\int_{\tilde{S}} \tilde{\varphi} \, da = \sum_{i=1}^{K} \int_{\sigma_i} \tilde{\varphi}(\boldsymbol{\xi}) \tilde{g}_i(\boldsymbol{\xi}) \, d\xi \, d\eta.$$

For conciseness, we omit the parameters $\boldsymbol{\xi}$ in the following. Therefore,

$$\begin{aligned} \left| \int_{\Gamma} \varphi \, da - \int_{\tilde{S}} \tilde{\varphi} \, da \right| &= \left| \sum_{i=1}^{K} \int_{\sigma} \left(\varphi g_i - \tilde{\varphi} \tilde{g}_i \right) \, d\xi \, d\eta \right| \\ &\le \sum_{i=1}^{K} \left| \int_{\sigma} \left(\left(\varphi - \tilde{\varphi} \right) g_i + \tilde{\varphi} \left(g_i - \tilde{g}_i \right) \right) \, d\xi \, d\eta \right| \\ &\le \sum_{i=1}^{K} \left(\int_{\sigma} \left| \left(\varphi - \tilde{\varphi} \right) g_i \right| \, d\xi \, d\eta + \int_{\sigma} \left| \tilde{\varphi} \left(g_i - \tilde{g}_i \right) \right| \, d\xi \, d\eta \right) \\ &\le \underbrace{|\varphi - \tilde{\varphi}|_{\max} \sum_{i=1}^{K} \int_{\sigma} g_i \, d\xi \, d\eta}_{1} + \underbrace{|\tilde{\varphi}|_{\max} \sum_{i=1}^{K} \int_{\sigma} \left| g_i - \tilde{g}_i \right| \, d\xi \, d\eta}_{2}. \end{aligned}$$

The first term is the condition number of the problem with a perturbation to the integrand and is bounded by $O(h^{d+1} + h^6)$ times the surface area. The error of the integral depends on the accuracy of the last integral in the second term. Before we determine its accuracy, we note that the number of triangles in the surface mesh is inversely proportional to the average area of the triangle, i.e., $K = O(1/h^2)$. Using this result along with Lemma 6, $|\tilde{g}_i - g_i| = O(h^{d+2} + h^7)$, we get

$$\begin{aligned} \sum_{i=1}^{K} \int_{\sigma} \left| \left(g_i - \tilde{g}_i \right) \right| \, d\xi \, d\eta &= \sum_{i=1}^{K} \int_{\sigma} O(h^{d+2} + h^7) \, d\xi \, d\eta \\ &= O(h^d + h^5). \end{aligned}$$

Hence, $| \int_{\Gamma} \varphi \, da - \int_{\tilde{S}} \tilde{\varphi} \, da| = O(h^d + h^5)$. □

Given Theorem 1, Corollary 2 follows naturally, because the surface integrals $\int_{\Gamma} \boldsymbol{\varphi} \cdot d\boldsymbol{a}$ can be reduced to the scalar integral of $\boldsymbol{\varphi} \cdot \hat{\boldsymbol{n}}$, and from Theorem 7, $\boldsymbol{\varphi} \cdot \hat{\boldsymbol{n}}$ can be approximated to $O(h^d + h^5)$, just as the scalar function φ in Theorem 1.

6. Numerical experiments.

We now present the numerical results with our methods, focusing on assessing the accuracy and convergence. We also demonstrate the usefulness of our method with an application and report the performance.

6.1. Convergence of high-order integrations.

The main objectives of our experiments are to verify the high-order convergence predicted by our theoretical analysis and to assess the effects of high-order reconstructions of the geometry and function. For these purposes, it suffices to use simple smooth geometries. We used a torus (with inner radius 0.7 and outer radius 1.3) as the test geometry and generated a set of high-quality meshes using the mesh generator Gambit from ANSYS, Inc. Note that by construction our method has little requirements for mesh quality. To

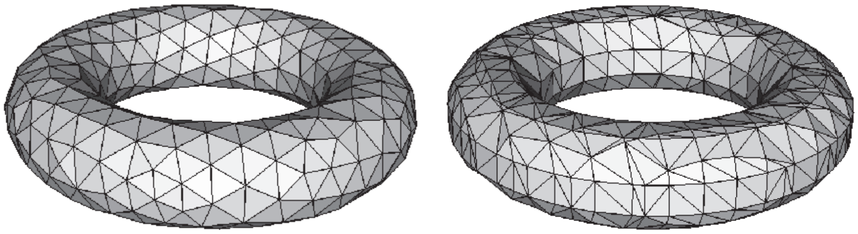

FIG. 4. *Coarsest high-quality (left) and poor-quality (right) test meshes for the torus used in our numerical experiments.*

demonstrate this, we also generated another set of poor-quality meshes using marching cubes. Some example meshes are shown in Figure 4. For studying mesh convergence, we generated five meshes of different resolutions for each set of our test meshes and numbered these meshes from the coarsest (mesh 1) to the finest (mesh 5). The average edge lengths are approximately halved between adjacent mesh resolutions. We use the finest meshes to compute the reference solutions when exact solutions are unknown and use the other four meshes to estimate convergence. We estimate the error for each mesh as

$$\text{relative error} = \frac{\|\,\text{numerical solution} - \text{reference solution}\,\|}{\|\,\text{reference solution}\,\|}$$

and compute the average convergence rate as

$$\text{convergence rate} = \frac{1}{3}\log_2\left(\frac{\text{error of mesh 1}}{\text{error of mesh 4}}\right).$$

To avoid poor convergence due to inaccurate inputs, in all cases we project the vertices onto the exact surface, so all the vertices are accurate to machine precision of double-precision floating point numbers.

6.1.1. Surface integral of scalar and vector functions. We first investigate the integration of a scalar function. A simple example is the computation of surface area, for which the integrand is $\varphi = 1$. In this case, we need to reconstruct only the geometry. For the torus, the exact surface areas can be computed analytically, so we use the exact answers as the reference solutions.

Figure 5 plots the relative errors of the computed surface areas for the torus using polynomial fittings of degrees between 1 and 6 under mesh refinement. The average convergence rates are shown on the right end of the curves in the plot. It can be seen that the order of convergence is at least as high as that predicted by the theory. The even-degree polynomials exhibited higher convergence rates than predicted, probably because of statistical error cancellation due to some symmetry of the geometry and the mesh.

For integrating vector-valued functions, the simplest example is the computation of the volume bounded by a closed surface. By the divergence theorem, the volume is equal to one third of the surface integral of the position vector of the surface, i.e.,

$$V = \int_\Gamma \boldsymbol{\varphi} \cdot d\boldsymbol{a} = \int_\Gamma \boldsymbol{\varphi} \cdot \hat{\boldsymbol{n}} da, \tag{22}$$

where $\boldsymbol{\varphi} = \boldsymbol{x}/3$ and $\hat{\boldsymbol{n}}$ is the outward unit normal to the surface Γ. For simple geometries such as a torus, the exact volumes are available analytically and hence we

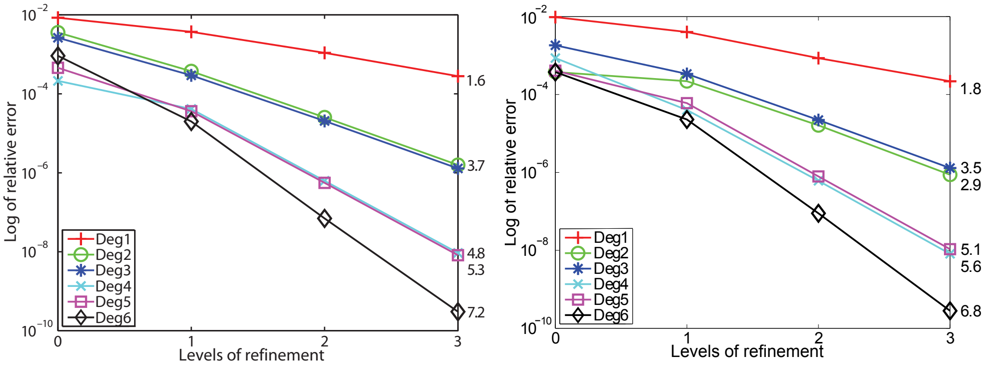

FIG. 5. *Relative errors and average convergence rates of surface area of a torus under mesh refinement for high-quality (left) and low-quality (right) meshes.*

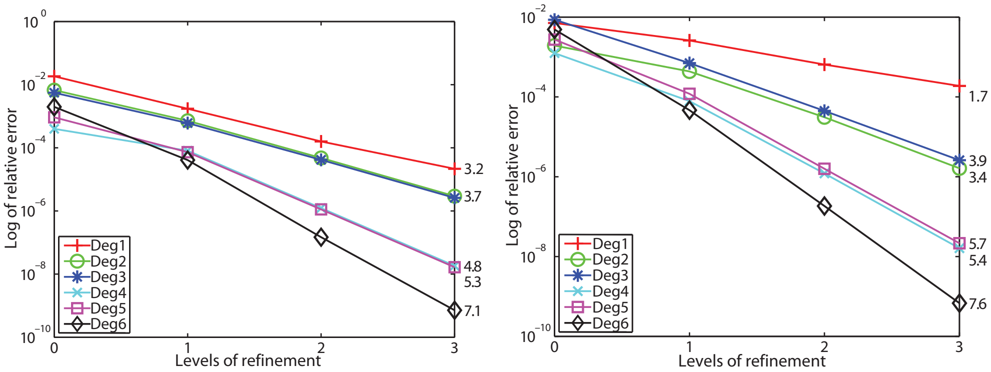

FIG. 6. *Relative errors and average convergence rates of computed volume of a torus under mesh refinement for high-quality (left) and low-quality (right) meshes.*

use them as reference solutions in our test. Figure 6 shows the relative errors of the computed volume of the torus using polynomial fittings of degrees between 1 and 6 under mesh refinement. The average convergence rates are shown on the right of the plots, which again confirm our theoretical results.

For generality, we also test integrating a scalar test function $\varphi(x,y,z) = \sin(x + yz) + e^{xy}$ and a vector-valued test function $\boldsymbol{\varphi}(x,y,z) = (x\cos(y),\ e^{y},\ z + e^{z})$ on the torus. Since the exact integrals are unavailable, we use the results for the finest meshes as reference solutions. The errors for this case are shown in Figure 7. These results are qualitatively similar to those of surface areas, and the convergence rates again confirm the theoretical results.

6.1.2. Necessity of high-order reconstructions of surface. In the previous subsection we showed that our method can deliver high-order convergence rates. It is important to note that the high-order reconstructions of both the surface and the function are necessary in assuring high convergence rates. To demonstrate this, Figure 8, left, shows the computed surface area of the torus using high-order reconstruction of the function but using piecewise linear geometry defined by the linear triangles of the meshes, and Figure 8, right, which is the same as Figure 5, left, shows the corresponding results using high-order reconstructions of both the geometry and

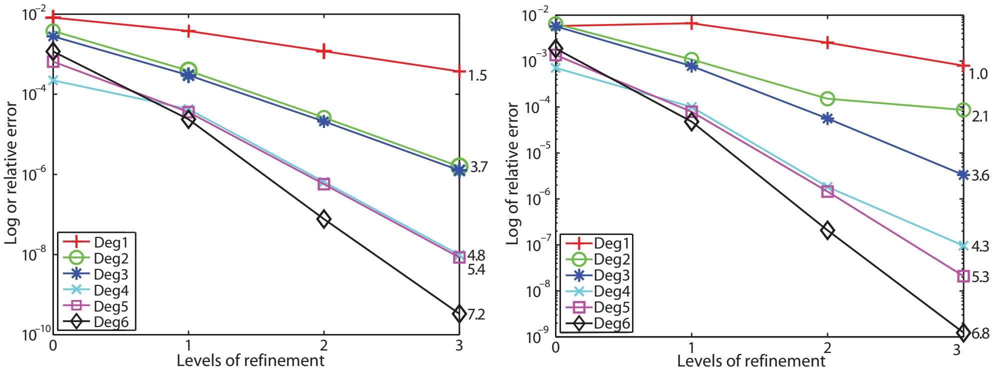

FIG. 7. *Relative errors and average convergence rates for integration of a test scalar function (left) and a test vector-valued function (right) on the torus.*

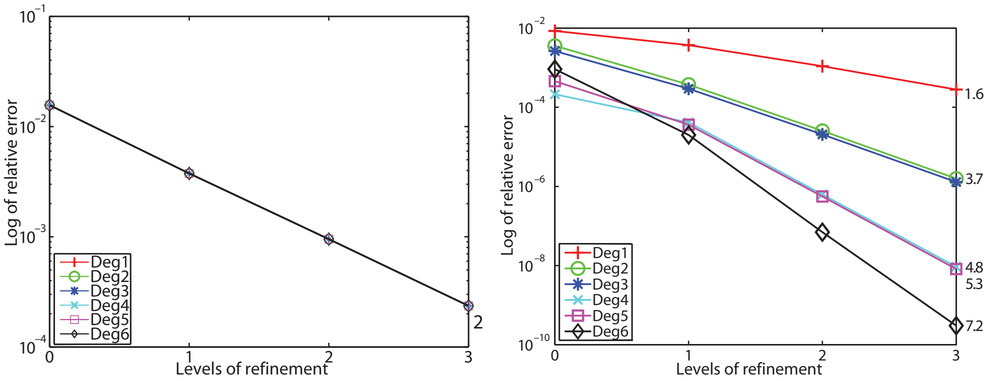

FIG. 8. *Comparison of convergence rates of computed surface area of the torus using high-order reconstructions of (left) only the function vs. (right) both the geometry and the function.*

the function. Using only high-order reconstruction of the geometry but not the function would lead to a result similar to that in Figure 8, left. From this comparison, it is evident that both the geometry and the function need to be reconstructed to high-order accuracy in order to achieve high-order surface integration.

6.1.3. Complicated geometries. We show some examples of our method applied to more complicated geometries as well as meshes with sharp features. For testing a complicated smooth domain of integration, we used a spherical surface harmonic of degree 6 and order 1 (Figure 9). The convergence of errors under mesh refinement for computing the surface area is shown in Figure 9, right. It can be seen that a high order of convergence is achieved as predicted.

To demonstrate the applicability of our method to meshes with sharp features, we apply our algorithm to a slotted sphere (Figure 10). The presence of sharp features in the input geometry requires some preprocessing of the mesh. In particular, we first identify all the sharp ridges and corners of the input geometry using a simple algorithm. Thereafter, we virtually split the mesh by duplicating the vertices along ridges and then apply our algorithm to this new mesh. The new connectivity ensures that stencils for points on the ridge edges are one-sided. Figure 10 shows the convergence results for computing the volume of the slotted sphere. We observe that for fittings of

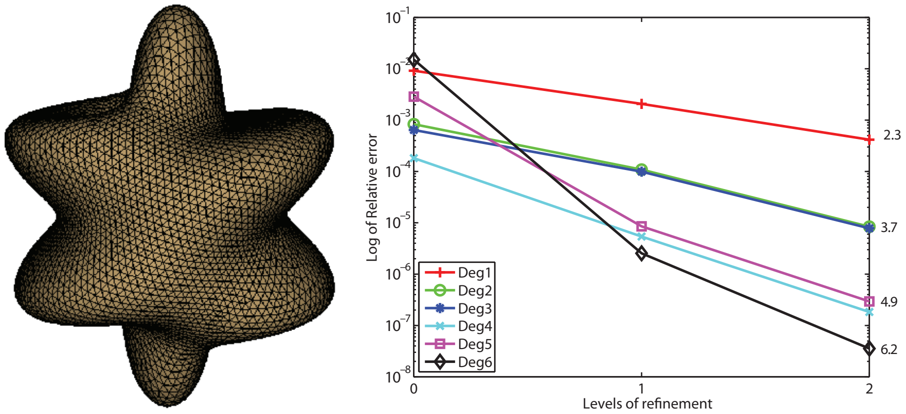

FIG. 9. *Convergence rates of computed surface area of a spherical harmonic of degree* 6 *(right). The domain of integration is shown on the left.*

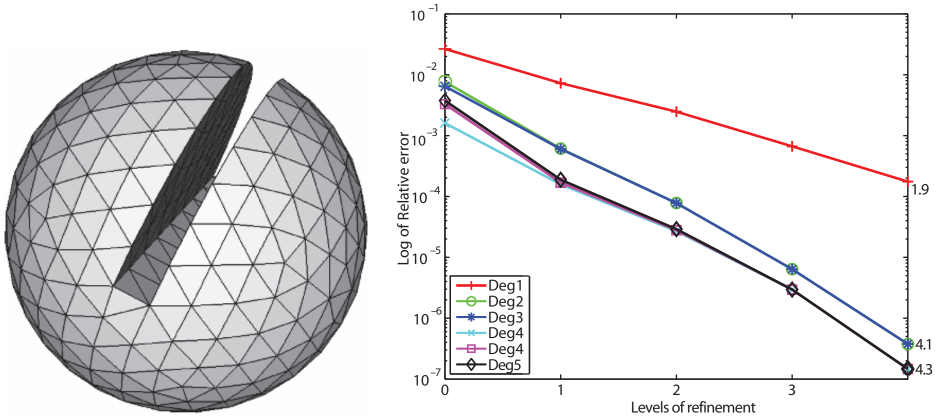

FIG. 10. *Convergence rates of computed volume of a slotted sphere (right). The domain of integration is shown on the left.*

degree 4, 5, and 6, the order of convergence is about 4. This is due to the reduction of degree of fittings for points on the sharp features, which have insufficient number points in their stencils to achieve higher order. As a result, the overall degree of the algorithm is reduced.

6.2. Applications to computation of Van der Waals force. As mentioned earlier, the surface integral has various applications. We demonstrate the usefulness of our method with an application to the computation of the Van der Waals interaction forces between two microscopic bodies and show the improvement in accuracy. The Van der Waals force between two bodies of arbitrary geometry is equal to

$$\boldsymbol{F} = C\rho_1\rho_2 \int_{V_2}\int_{V_1} \nabla\frac{1}{s^6}\, dV_1\, dV_2. \tag{23}$$

Here, ρ_i is the atomic density of body i for $i = 1, 2$, C is the London–Van der Waals constant, and s is the distance between two atoms or molecules. Defining a vector

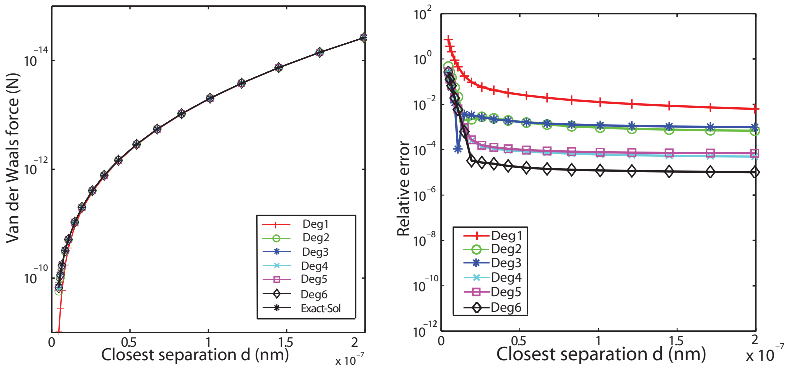

FIG. 11. *Analytical and numerical results for sphere-sphere interaction force calculation. Left: Van der Waals force at different separation distances. Right: relative errors of numerical results at different separation distances.*

field $\boldsymbol{G}$ such that $\nabla \cdot \boldsymbol{G} = C/s^6$, one can reduce (23) to

$$\boldsymbol{F} = \rho_1 \rho_2 \int_{S_2} \int_{S_1} (\boldsymbol{G} \cdot \hat{\boldsymbol{n}}_1) \cdot \hat{\boldsymbol{n}}_2 \, dS_1 \, dS_2, \tag{24}$$

where S_1 and S_2 are the surfaces of bodies 1 and 2, and $\hat{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_2$ are the unit outward normal to S_1 and S_2, respectively. In [12], the discretization of (24) is given for geometries defined by non-uniform rational B-spline (NURBS), and $\boldsymbol{G} = \frac{1}{3} C \left(\boldsymbol{s}_1 - \boldsymbol{s}_2\right) / \|\boldsymbol{s}_1 - \boldsymbol{s}_1\|^6$, where $\boldsymbol{s}_1 \in S_1$ and $\boldsymbol{s}_2 \in S_2$. Here we assume that S_1 and S_2 are given by triangulations. Let $S_1 = \bigcup_{i=1}^{n_1} \sigma_i$ and $S_2 = \bigcup_{j=1}^{n_2} \tau_j$, where n_1 and n_2 are the numbers of triangles in S_1 and S_2, respectively. Therefore, (24) can be rewritten as

$$\boldsymbol{F} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \rho_1 \rho_2 \int_{\tau_j} \int_{\sigma_i} (\boldsymbol{G} \cdot \hat{\boldsymbol{n}}_1) \cdot \hat{\boldsymbol{n}}_2 \, dS_1 \, dS_2,$$

i.e., the summation of double integrals over all pairs of triangles in S_1 and S_2. We use our algorithm to compute the double integrals using high-order reconstructions of the surface and the surface normals. Similar to [12], we show the results for the interaction force between two spherical bodies with radius $R_1 = 100$ nm and $R_2 = 100$ nm, and $\rho_1 = \rho_2 = 8.49 \times 10^{28}/\mathrm{m}^3$ and $C = 4.5639 \times 10^{-78} \mathrm{Jm}^6$. We calculate the Van der Waals force for 20 different separations, which are nonuniformly distributed from 5 nm to 200 nm. Figure 11, left, shows the computed Van der Waals force for polynomial fittings of degrees from 1 to 6, along with the exact solutions. Figure 11, right, shows the relative errors for different degrees. From these results, it is evident that the computed forces are more accurate with higher-degree fittings, and high-order reconstructions can significantly improve the accuracy compared with piecewise linear approximations.

6.3. Performance results. We present some performance results for our algorithms. We implemented our algorithm in MATLAB and then converted the code into C using MATLAB Coder. We performed the tests on a Linux machine with a 3.16 GHz Intel Core 2 Duo processor and 4 GB of RAM. Table 1 shows the execution times for the computations of the volume of the torus, using polynomials of degrees between 1 and 6. It can be seen that the runtimes scale linearly with the size of the meshes. In addition, the higher the degree of the polynomials, the more expensive the algorithms are. Compared to integration with linear triangles, it is expected that polynomial

Table 1
Runtime in seconds for computation of volume of triangulated torus.

Mesh	#Vertices	Degree 1	2	3	4	5	6
1	544	0.0927	0.172	0.34	0.785	1.45	2.44
2	1,896	0.322	0.612	1.26	2.73	5.57	11.2
3	7,528	1.27	2.35	4.87	11.0	21.0	44.3
4	31,392	5.34	9.84	20.18	47.2	89.1	168.4

fittings would be much more expensive, because the stencil for the fitting is large and least squares problems need to be solved. The computational complexity of solving the linear system is dominated by that of finding the reduced QR factorization. For our current implementation, polynomial fittings are two to three orders of magnitude more expensive than using linear triangles, but its runtime can be improved with further performance optimization.

7. Conclusions.

We proposed a novel method for high-order numerical integration over triangulated surfaces. Our method uses stabilized least squares polynomial fittings locally at each vertex to fit the geometry and the interpolant. The polynomials at the vertices are then blended using finite element shape functions. The resulting surface is a piecewise high-order approximation with C^0 continuity along triangle boundaries. Within each triangle, the blended surface is a polynomial, which is then integrated using high-degree quadrature rules. We presented numerical results for integrating both scalar functions and vector functions and demonstrated up to sixth-order convergence rate. The method is also largely independent of mesh quality in the sense that it does not depend on the angles or Jacobian or other general quality attributes of a mesh. As a result, we can still get high order for very poor quality meshes, as shown in the numerical experiments.

The present work has some limitations. Most notably, the least squares high-order reconstruction is significantly more expensive than using linear triangles. Whether the extra cost is warranted would depend on the specific applications. Nevertheless, we plan to further optimize the performance of our method.

Because of the fundamental importance of the surface integral in various applications, such as the computation of flux in fluid dynamics and the computation of swept volume of moving interfaces, this proposed method can have implications in such application areas. We plan to integrate our method with other numerical methods such as finite element methods, embedded boundary methods, and conservative front tracking for moving interfaces and apply them to applications such as fluid dynamics and fluid-structure interactions.

Appendix A. Derivation of Jacobian of WALF surface.

The Jacobian of the WALF surface is

$$\begin{aligned}\boldsymbol{\nabla}_{\boldsymbol{\xi}}\boldsymbol{p}(\boldsymbol{\xi}) &= \nabla_{\boldsymbol{\xi}}\sum_{i=1}^{3}\left(N_i(\boldsymbol{\xi})\boldsymbol{p}_i(\boldsymbol{\xi})\right)\\ &= \sum_{i=1}^{3}\left(\left(\nabla_{\boldsymbol{\xi}}N_i(\boldsymbol{\xi})\right)\boldsymbol{p}_i(\boldsymbol{\xi})\right)+\sum_{i=1}^{3}\Big(N_i(\boldsymbol{\xi})\nabla_{\boldsymbol{\xi}}\boldsymbol{p}_i(\boldsymbol{\xi})\Big).\end{aligned}$$

Note that

$$\sum_{i=1}^{3}\left(\left(\nabla_{\boldsymbol{\xi}}N_i(\boldsymbol{\xi})\right)\boldsymbol{p}_i(\boldsymbol{\xi})\right)=[\,\boldsymbol{p}_2(\boldsymbol{\xi})-\boldsymbol{p}_1(\boldsymbol{\xi})|\,\boldsymbol{p}_3(\boldsymbol{\xi})-\boldsymbol{p}_1(\boldsymbol{\xi})]. \tag{25}$$

Then, $\boldsymbol{\nabla}_{\boldsymbol{\xi}}\boldsymbol{u} = \begin{bmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial u}{\partial \eta} \\ \frac{\partial v}{\partial \xi} & \frac{\partial v}{\partial \eta} \end{bmatrix} = \boldsymbol{T}_i^T \boldsymbol{J}_f$, and

$$\begin{aligned}\boldsymbol{\nabla}_{\boldsymbol{\xi}}\boldsymbol{p}_i(\boldsymbol{\xi}) &= \nabla_{\boldsymbol{u}}\left(\boldsymbol{Q}_i\boldsymbol{p}_i(\boldsymbol{u}) + \boldsymbol{x}_i\right)\left(\nabla_{\boldsymbol{\xi}}\boldsymbol{u}\right) \\ &= \left(\boldsymbol{Q}_i \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \frac{\partial f_i}{\partial u} & \frac{\partial f_i}{\partial v} \end{bmatrix}\right)\boldsymbol{T}_i^T\boldsymbol{J}_f \\ &= \left(\boldsymbol{T}_i + \hat{\boldsymbol{m}}_i\nabla_{\boldsymbol{u}} f_i(\boldsymbol{u})\right)\boldsymbol{T}_i^T\boldsymbol{J}_f,\end{aligned}$$

where $\boldsymbol{u}$ is also functions of $\boldsymbol{\xi}$. Therefore,

$$\sum_{i=1}^{3} N_i(\boldsymbol{\xi})\boldsymbol{\nabla}_{\boldsymbol{\xi}}\boldsymbol{p}_i(\boldsymbol{\xi}) = \sum_{i=1}^{3}\Big(N_i(\boldsymbol{\xi})\left(\boldsymbol{T}_i + \hat{\boldsymbol{m}}_i\nabla_{\boldsymbol{u}} f_i(\boldsymbol{u}(\boldsymbol{\xi}))\right)\boldsymbol{T}_i^T\boldsymbol{J}_f\Big). \tag{26}$$

Acknowledgments. The authors would like to thank the anonymous reviewers for their helpful comments and thank Ms. Rebecca Conley for her help in proofreading the paper.

REFERENCES

[1] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, *Meshless methods: An overview and recent developments*, Comput. Methods Appl. Mech. Engrg., 139 (1996), pp. 3–47.

[2] F. Cazals and M. Pouget, *Estimating differential quantities using polynomial fitting of osculating jets*, Comput. Aided Geom. Design, 22 (2005), pp. 121–146.

[3] D. Chien, *Numerical evaluation of surface integrals in three dimensions*, Math. Comp., 64 (1993), pp. 727–743.

[4] R. Cools, *An encyclopaedia of cubature formulas*, J. Complexity, 19 (2003), pp. 445–453.

[5] K. Georg, *Approximation of integrals for boundary element methods*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 443–453.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computation*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[7] X. Jiao and D. Wang, *Reconstructing high-order surfaces for meshing*, Engrg. Comput., 28 (2011), pp. 361–373; DOI: 10.1007/s00366-011-0244-8.

[8] X. Jiao and H. Zha, *Consistent computation of first- and second-order differential quantities for surface meshes*, in Proceedings of the ACM Solid and Physical Modeling Symposium, ACM, 2008, pp. 159–170.

[9] P. Lancaster and K. Salkauskas, *Curve and Surface Fitting: An Introduction*, Academic Press, New York, 1986.

[10] J. N. Lyness and R. Cools, *A Survey of Numerical Cubature over Triangles*, Proc. Sympos. Appl. Math. 48, AMS, Providence, RI, 1994, pp. 127–150.

[11] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.

[12] P. Yang and X. Qian, *A general accurate procedure for calculating molecular interaction force*, J. Colloid Interface Science, 337 (2009), pp. 594–605.

[13] O. Zienkiewicz, R. Taylor, and J. Zhu, *The Finite Element Method: Its Basis & Fundamentals*, 6th ed., Elsevier, New York, 2005.