

Solving Boundary Handling Analytically in Two Dimensions for Smoothed Particle Hydrodynamics

Rene Winchenbach Andreas Kolb
Technical University Munich University of Siegen

July 30, 2025

Abstract

We present a fully analytic approach for evaluating boundary integrals in two dimensions for Smoothed Particle Hydrodynamics (SPH). Conventional methods often rely on boundary particles or wall re-normalization approaches derived from applying the divergence theorem, whereas our method directly evaluates the area integrals for SPH kernels and gradients over triangular boundaries. This direct integration strategy inherently accommodates higher-order boundary conditions, such as piecewise cubic fields defined via Finite Element stencils, enabling analytic and flexible coupling with mesh-based solvers. At the core of our approach is a general solution for compact polynomials of arbitrary degree over triangles by decomposing the boundary elements into elementary integrals that can be solved with closed-form solutions. We provide a complete, closed-form solution for these generalized integrals, derived by relating the angular components to Chebyshev polynomials and solving the resulting radial integral via a numerically stable evaluation of the Gaussian hypergeometric function ${}_2F_1$. Our solution is robust and adaptable and works regardless of triangle geometries and kernel functions. We validate the accuracy against high-precision numerical quadrature rules, as well as in problems with known exact solutions. We provide an open-source implementation of our general solution using differentiable programming to facilitate the adoption of our approach to SPH and other contexts that require analytic integration over polygonal domains. Our analytic solution outperforms existing numerical quadrature rules for this problem by up to five orders of magnitude, for integrals and their gradients, while providing a flexible framework to couple arbitrary triangular meshes analytically to Lagrangian schemes, building a strong foundation for addressing several grand challenges in SPH and beyond.

1 Introduction

Smoothed Particle Hydrodynamics (SPH) is a Lagrangian simulation method originally proposed for the simulation of stellar phenomena in astrophysical simulations [1]. SPH has since found a wide range of applications, including in computational fluid dynamics [2, 3], astrophysics [4, 5], and computer animation [6, 7], due to the inherent Lagrangian nature of SPH leading to significant benefits for free-surface flows and complex deformations. However, the Lagrangian nature of SPH does not lend itself to incorporating rigid boundaries or to the coupling to Eulerian mesh-based schemes, as rigid objects and mesh elements are not Lagrangian in nature. That is, they are often described using meshes, and integrating a rigid body into an SPH simulation requires either finding an accurate Lagrangian representation for the rigid body or

incorporating non-Lagrangian aspects into SPH. Resolving this impedance mismatch between the Lagrangian nature of SPH and the discrete element-based description of other schemes and boundaries is of significant importance to both the numerical accuracy and versatility of SPH in computational fluid dynamics and beyond.

Particle-based rigid boundary handling methods can be distinguished into ghost particle methods [8, 9, 10, 11, 12], which either place virtual particles inside the boundary that cover the entire support domain of a particle, or on the surface of the boundary [13, 14, 15], where either approach may utilize a consistent set of particles or generate them dynamically. Particle-based methods generally suffer from sampling problems, e.g., placing particles on the surface in a uniform manner, and struggle with representing flat geometries as the particle sampling can introduce artifacts in the density field [14]. Consequently, corrective terms are required to counteract these problems [13, 16], e.g., pseudo-volumes accounting for deficiencies in the particle sampling or to represent the rigid body using boundary surface particles.

Instead of finding a Lagrangian representation of the boundary, integral formulations directly incorporate a non-Lagrangian boundary representation into the SPH formulation. The most common approach for boundary integrals is determining a wall renormalization factor γ , originally introduced by Kulasegaram et al. [17], where the integrals themselves are commonly evaluated using the divergence theorem. There has been significant research in this direction by incorporating gradient operators [18], using semi-analytic [19, 20], numerical [20, 21] or fully analytic solutions to the integrals [22, 23, 24], for both non-viscous and viscous flows in both 2D and 3D. Instead of using wall renormalization approaches, an alternative approach is the direct evaluation of the contribution of a boundary object to a fluid particle, enabling a straightforward integration into fluid simulation models. These methods work by either numerically evaluating the contributed density [25] or a representative volume term [26] or by semi-analytically evaluating the kernel function over the boundary domain [7]. However, these schemes do not generally provide exact solutions and rely on approximations that may not always be applicable.

In this paper, we overcome these limitations by introducing a novel and generalized framework for the fully analytic, direct evaluation of 2D SPH boundary integrals over triangular domains. Our solution can be applied not just to handling rigid boundaries, but is also capable of coupling SPH exactly to mesh-based simulation schemes, removing a significant gap in the fundamental difference in discretization. Using our general solution, we can find these closed-form integrals for any compact polynomial kernel, e.g., B-Splines and Wendland kernels, of arbitrary order. The core of our approach relies on a decomposition of triangles into fundamental forms that can be analytically computed for only a small set of terms, i.e., r^k , $r^k \cos n\theta$, and $r^k \cos n\theta \sin m\theta$. We then establish the closed-form analytic solutions for these general integrals by relating the angular components to Chebyshev polynomials and solving the resulting radial integrals through a numerically stable evaluation of the Gaussian hypergeometric function ${}_2F_1$. This approach inherently supports higher-order boundary conditions, such as piecewise linear fields defined by barycentric interpolation, enabling more sophisticated and accurate physical interactions.

The remainder of this paper is organized as follows. Section 2 briefly reviews the SPH formulation and defines our general boundary integral formulation. Section 3 describes the integration of piecewise polynomial fields using barycentric interpolation into the boundary integral formulation and defines the fundamental integral terms that need to be solved. Section 4 covers the decomposition of arbitrary triangles into fundamental forms, i.e., sectors, segments, and stubs, with well-defined integration bounds and provides the general algorithm of combining these fundamental terms into integrals over actual triangles. Section 5

describes the fully-analytic closed-form solution of the integral terms described in Section 4 and the evaluation of the ${}_2F_1$ function for piecewise constant and polynomial boundary terms. Section 6 evaluates the numerical accuracy of our approach by comparing the results to high-order numerical quadratures, which rely on the same description as in Section 3, as well as evaluating the accuracy against problems with known exact solutions. This section also demonstrates the application of our scheme to a simple SPH simulation. Finally, Section 7 provides concluding remarks and discusses future work. Note that we provide an open-source implementation of the framework both in C++ and PyTorch, enabling the use of our solution in SPH and beyond, with the PyTorch approach enabling novel applications using differentiable, exact, higher-order coupling to boundaries and mesh-based solvers.

2 Governing Equations and Boundary Integral Formulation

SPH fundamentally is based on an identity of a function, where a field quantity A at any point $\mathbf{x} \in \mathbb{R}^2$ can be evaluated as an integral over the entire input space [27]:

$$\langle A(\mathbf{x}) \rangle = \int_{\mathbb{R}^2} A(\mathbf{x}') \delta(\mathbf{x}, \mathbf{x}') d\mathbf{x}', \quad (1)$$

where δ is the Dirac delta function. Note that this definition works in any number of spatial dimensions, but we only discuss two-dimensional SPH here, as our method is only applicable in 2D. The Dirac delta function is then replaced by a compact kernel function W with a support radius h [27] yielding an approximation for a field quantity A at any point \mathbf{x} based on a compact spherical domain Ω , with cutoff radius h and centered at \mathbf{x} , as

$$\langle A(\mathbf{x}) \rangle \approx \int_{\Omega} A(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (2)$$

This can also be applied to gradients of generic field quantities to yield a naïve continuous gradient formulation as

$$\langle \nabla A(\mathbf{x}) \rangle \approx \int_{\Omega} A(\mathbf{x}') \nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (3)$$

SPH then uses particles to discretize the fluid domain, which carry discrete values of the field quantities, where the apparent volume $V_j = \frac{m_j}{\rho_j}$ of a particle is used as weights [28], yielding

$$\langle \nabla A(\mathbf{x}) \rangle = \sum_j \frac{m_j}{\rho_j} A_j W(\mathbf{x} - \mathbf{x}_j, h). \quad (4)$$

Here, j refers to the neighboring particles around \mathbf{x} , i.e., all particles j with $|\mathbf{x} - \mathbf{x}_j| \leq h$, with m_j and ρ_j being the mass and density of the particle, respectively. Gradient terms can analogously be determined, however, such terms would neither result in symmetric forces nor be exact for constant functions [27] in the presence of particle disorder. A gradient formulation that is exact for constant functions, but not symmetric, can be derived as

$$\nabla A = \frac{1}{\rho} [\langle \nabla(\rho A) \rangle - A \langle \nabla \rho \rangle], \quad (5)$$

also called the difference formulation. A gradient formulation that is symmetric, but not exact for constant

functions, can be derived as

$$\nabla A = \rho \left[\left\langle \nabla \left(\frac{A}{\rho} \right) \right\rangle + \frac{A}{\rho^2} \langle \nabla \rho \rangle \right], \quad (6)$$

also called symmetric formulation [27]. The discretized forms of Eq. (5) and Eq. (6) yield:

$$\begin{aligned} \langle \nabla A(\mathbf{x}) \rangle &\approx \frac{1}{\rho(\mathbf{x})} \sum_j m_j (A_j - A(\mathbf{x})) \nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}_j, h) \\ \langle \nabla A(\mathbf{x}) \rangle &\approx \rho(\mathbf{x}) \sum_j m_j \left(\frac{A_j}{\rho_j^2} - \frac{A(\mathbf{x})}{\rho(\mathbf{x})^2} \right) \nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}_j, h). \end{aligned} \quad (7)$$

Note that analogous formulations exist for both divergence $\nabla \cdot A(\mathbf{x})$ and curl $\nabla \times A(\mathbf{x})$ [15]. For our boundary integral evaluation, we require the continuous forms of these terms, which can be directly defined using the continuous form of the SPH gradient operator as

$$\nabla A \approx \frac{1}{\rho(\mathbf{x})} [\langle \nabla(\rho A) \rangle - A \langle \nabla \rho \rangle] = \frac{1}{\rho(\mathbf{x})} \int_{\Omega} \rho(\mathbf{x}') [A(\mathbf{x}') - A(\mathbf{x})] \nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \quad (8)$$

$$\nabla A \approx \rho(\mathbf{x}) \left[\left\langle \nabla \left(\frac{A}{\rho} \right) \right\rangle + \frac{A}{\rho^2} \langle \nabla \rho \rangle \right] = \rho(\mathbf{x}) \int_{\Omega} \rho(\mathbf{x}') \left[\frac{A(\mathbf{x}')}{\rho^2(\mathbf{x}')} - \frac{A(\mathbf{x})}{\rho^2(\mathbf{x})} \right] \nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (9)$$

A general definition of the kernel function, based on Dehnen and Aly [29], is defined as

$$W(\mathbf{x}, h) = \frac{1}{h^d} C_d k \left(\frac{|\mathbf{x}|}{h} \right), \quad (10)$$

with $k(q)$ being the actual kernel, d the dimension and C_d a normalization factor. Note that, without loss of generality for the derivation, we can assume the smoothing scale and the cut-off radius to be identical. Consequently, the derivative of the kernel function is given as

$$\nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}', h) = \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|} \frac{1}{h^{d+1}} C_d \frac{\partial k \left(\frac{|\mathbf{x} - \mathbf{x}'|}{h} \right)}{\partial \frac{|\mathbf{x} - \mathbf{x}'|}{h}}. \quad (11)$$

We use the Wendland 4 kernel with $C_2 = \frac{27}{16\pi}$ and $k(q) = [1 - q]_+^6 (1 + 6q + \frac{35}{3}q^2)$, with $[\cdot]_+ = \max\{\cdot, 0\}$. As the kernel can be solely expressed using integer powers of the radial distance q within the support domain Ω , i.e., it can be written as $k(q) = \sum_i b_k q^k$, solving the boundary integral formulation for an arbitrary power q^k yields a solution to the entire kernel function. This can be extended to piecewise kernel functions trivially, such as the B-Splines, through careful evaluation of the integral bounds for each piece of the kernel function.

We now assume that there is a boundary domain Ω_b , consisting of individual triangle elements \mathcal{T} , with no requirements placed on the individual triangles' shapes and connectivity. Note that our approach still requires disjoint triangles, i.e., triangle meshes with no overlap, to yield physically plausible answers. Given a set of neighboring triangles $\mathcal{N}_{\mathcal{T}}$, i.e., triangles that overlap the support radius h relative to a point in space \mathbf{x} , the integral over the entire boundary domain is evaluated as

$$\langle A(\mathbf{x}) \rangle \approx \sum_{\mathcal{T} \in \mathcal{N}_{\mathcal{T}}} \int_{\mathcal{T}} A(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (12)$$

As we assume triangles to be disjointed, we can treat each triangle that is to be integerated independently as their connectivity to neighboring triangles does not affect our scheme. To further simplify the integral solutions, we first transform this equation into a normalized form, i.e., we remove unnecessary degrees of freedom from the equation, without loss of generality. This involves transforming the coordinate system such that the cutoff radius is equal to 1, note that this does not necessarily mean that the smoothing length is 1, and that the position \mathbf{x} at which this integral is evaluated is at the origin $\mathbf{0}$. Each individual triangle \mathcal{T} consisting of three vertices located at positions ν_1, ν_2 and $\nu_3 \in \mathbb{R}^2$ is thus shifted by \mathbf{x} and inversely scaled by h to ensure that $h = 1$ for the purposes of integration by applying a homogeneous transformation matrix

$$C = \begin{bmatrix} 1/h & 0 & -\frac{\mathbf{x}_x}{h} \\ 0 & 1/h & -\frac{\mathbf{x}_y}{h} \\ 0 & 0 & 1 \end{bmatrix},$$

which yields a transformed triangle \mathcal{T}^* , and vertices $\nu_i^* = C\nu_i$, with

$$\begin{aligned} \langle A(\mathbf{0}) \rangle &\approx \int_{\mathcal{T}^*} A(\mathbf{x}') W(-\mathbf{x}', 1) d\mathbf{x}', \\ \langle \nabla A(\mathbf{0}) \rangle &\approx \frac{1}{h} \int_{\mathcal{T}^*} A(\mathbf{x}') \nabla_{\mathbf{x}} W(-\mathbf{x}', 1) d\mathbf{x}', \end{aligned} \quad (13)$$

Note that this does not change the result of the kernel integral as the kernel function W is invariant to translation, i.e., only relative positions matter. For the gradient term, the additional division by the support radius h is required as part of the u-substitution. Referring back to the general kernel formulation [29], this means that the kernel function $W(\mathbf{x}, 1)$ can be replace with $C_d k(|\mathbf{x}|)$, where we drop the C_d as this term can simply be multiplied back onto the result after integration, without loss of generality. This yields an integral for the boundary contribution as

$$\langle A(\mathbf{0}) \rangle \approx \sum_{\mathcal{T}^* \in \mathcal{N}_{\mathcal{T}}} \int_{\mathcal{T}^*} A(\mathbf{x}') w(|-\mathbf{x}'|) d\mathbf{x}', \quad (14)$$

with $w(|\mathbf{x}|) = W(\mathbf{x}, 1)$. To simplify the integration process later on, we apply an improper rotation to a triangle to transform a given triangle into a triangle for which a solution is known. While Eq. 14 is invariant to rotations of the coordinate system, integrals of gradient terms need to be adjusted when an improper rotation matrix R is applied to the local coordinate system, after scaling and shifting, to yield a new triangle \mathcal{T}^R , and vertices $\nu_i^r = R\nu_i^*$. This, after dropping the subscript of ∇ for readability, yields a term that needs to be transformed with the inverse of R , i.e.,

$$\int_{\mathcal{T}^*} A(\mathbf{x}') \nabla w(|-\mathbf{x}'|) d\mathbf{x}' = R^{-1} \int_{\mathcal{T}^R} A(\mathbf{x}') \nabla w(|-\mathbf{x}'|) d\mathbf{x}'. \quad (15)$$

To evaluate the integral for difference gradient formulations (Eq. 8), we first evaluate quantities $f_i = \rho_i(A_i - A(\mathbf{0}))$ for each vertex of a triangle and then perform a barycentric interpolation of these quantities across the triangle during integration. Consequently, the overall integrals remain unchanged and only require multiplication by a different factor. Symmetric gradient formulations can be evaluated analogously by evaluating $f_i = \rho_i \left[\frac{A_i}{\rho_i^2} + \frac{A(\mathbf{0})}{\rho(\mathbf{0})^2} \right]$ prior to integration. Finally, divergence terms and curl terms for vector-

valued fields becomes

$$\begin{aligned}\langle \nabla \cdot A(\mathbf{x}) \rangle &= \langle \nabla A_x(\mathbf{x}) \rangle_x + \langle \nabla A_y(\mathbf{x}) \rangle_y, \\ \langle \nabla \times A(\mathbf{x}) \rangle &= \langle \nabla A_y(\mathbf{x}) \rangle_x - \langle \nabla A_x(\mathbf{x}) \rangle_y.\end{aligned}\tag{16}$$

3 Piecewise Polynomial Field Quantities

In Eulerian simulations using triangular meshes the two most common forms of quantities are either per-element information, i.e., piece-wise constant, or per-vertex information, i.e., piece-wise linear information. Note that in many prior works, a common choice in SPH is assuming A to be constant over each boundary element, i.e., $A(\mathbf{x}) = \text{const} : \forall \mathbf{x} \in \mathcal{T}$, i.e., limiting A to be piecewise constant. Alternatively, using barycentric interpolation to evaluate A across a triangle \mathcal{T} , the fields defined over boundaries can be piecewise polynomial using Finite Element Method stencils. Replacing the generic field $A(\mathbf{x})$ with a barycentric evaluation $A_{\mathcal{T}}(\mathbf{x})$, over the triangle \mathcal{T} , and by transforming from cartesian to polar coordinates yields

$$\langle A(\mathbf{0}) \rangle \approx \int_{\mathcal{T}} A_{\mathcal{T}}(r \cos \theta, r \sin \theta) w(r) r d\theta dr. \tag{17}$$

$A_{\mathcal{T}}$ is based on a barycentric interpolation using the vertex positions $\nu_i = [x_i, y_i]^T$ and their representative field values A_i . Barycentric interpolation can be defined using the canonical interpolation weights

$$\lambda_0(x, y) = \frac{(y_1 - y_2)(x - x_2) + (x_2 - x_1)(y - y_2)}{(y_1 - y_2)(x_0 - x_2) + (x_2 - x_1)(y_0 - y_2)}, \quad \lambda_1(x, y) = \frac{(y_2 - y_0)(x - x_2) + (x_0 - x_2)(y - y_2)}{(y_1 - y_2)(x_0 - x_2) + (x_2 - x_1)(y_0 - y_2)}, \tag{18}$$

with the third weight being $\lambda_2(x, y) = 1 - \lambda_0 - \lambda_1$. This then yields the canonical barycentric interpolation for a triangle as

$$A_{\mathcal{T}}(x, y) = \lambda_0(x, y) A_0 + \lambda_1(x, y) A_1 + \lambda_2(x, y) A_2 = A_2 + \lambda_0(x, y) [A_0 - A_2] + \lambda_1(x, y) [A_1 - A_2], \tag{19}$$

Note that this formulation is correct even outside the triangle and can also be used to perform point in triangle checks that will later be required for some steps of the algorithm to compute an integral. Inserting this back into eq. (17) yields

$$\langle A(\mathbf{0}) \rangle \approx A_2 \int_{\mathcal{T}} w(r) r d\theta dr + [A_0 - A_2] \int_{\mathcal{T}} \lambda_0(x, y) w(r) r d\theta dr + [A_1 - A_2] \int_{\mathcal{T}} \lambda_1(x, y) w(r) r d\theta dr, \tag{20}$$

where the first term is identical to the term that arises when assuming that the field A is constant over the boundary element, i.e., $A_0 = A_1 = A_2 = \text{const}$. Back inserting the definition of the barycentric interpolation with $D = (y_1 - y_2)(x_0 - x_2) + (x_2 - x_1)(y_0 - y_2)$ and refactoring common terms then yields

$$\langle A(\mathbf{0}) \rangle \approx A_2 \int_{\mathcal{T}} w(r) r d\theta dr + [\Lambda_0^A + \Lambda_2^A] \int_{\mathcal{T}} [r \cos \theta - x_2] w(r) r d\theta dr + [\Lambda_1^A + \Lambda_3^A] \int_{\mathcal{T}} [r \sin \theta - y_2] w(r) r d\theta dr, \tag{21}$$

with a new set of factors, which are independent of the coordinates, as

$$\Lambda_0^A = \frac{(y_1 - y_2)(A_0 - A_2)}{D}, \quad \Lambda_1^A = \frac{(x_2 - x_1)(A_0 - A_2)}{D}, \quad \Lambda_2^A = \frac{(y_2 - y_0)(A_1 - A_2)}{D}, \quad \Lambda_3^A = \frac{(x_0 - x_2)(A_1 - A_2)}{D}. \tag{22}$$

This can be further refactored into

$$\begin{aligned}\langle A(\mathbf{0}) \rangle &\approx \underbrace{[\Lambda_0^A + \Lambda_2^A]}_{\tau_1^A} \int_{\mathcal{T}} w(r) r^2 \cos \theta d\theta dr + \underbrace{[\Lambda_1^A + \Lambda_3^A]}_{\tau_2^A} \int_{\mathcal{T}} w(r) r^2 \sin \theta d\theta dr \\ &+ \underbrace{[A_2 - (\Lambda_0^A x_2 + \Lambda_1^A y_2 + \Lambda_2^A x_2 + \Lambda_3^A y_2)]}_{\tau_3^A} \int_{\mathcal{T}} w(r) r d\theta dr.\end{aligned}\quad (23)$$

Using the earlier observation that we can replace a kernel for polynomial kernel functions with a summation over polynomials and their corresponding factors, we can then write

$$\langle A(\mathbf{0}) \rangle \approx \sum_{k=0} b_k \left(\tau_1^A \int_{\mathcal{T}} r^{k+2} \cos \theta d\theta dr + \tau_2^A \int_{\mathcal{T}} r^{k+2} \sin \theta d\theta dr + \tau_3^A \int_{\mathcal{T}} r^{k+1} d\theta dr \right). \quad (24)$$

Note that for this term only the third integral is required for piecewise-constant boundary integrals with $\tau_3^A = A$. To determine gradient terms, we first consider a basic gradient formulation, where after inserting the kernel derivative, we get for a single triangle

$$\langle \nabla A \rangle \approx - \int_{\mathcal{T}} A_{\mathcal{T}}(r \cos \theta, r \sin \theta) \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \frac{\partial w(r)}{\partial r} r d\theta dr, \quad (25)$$

Note that in polar coordinates the gradient, i.e., $\nabla|\mathbf{x}|$, becomes the cos and sin of the angle. Using a derivation for the components of the gradient, similar to Eq. 23, we get equivalent integral terms by simply replacing $w(r)$ with $\cos \theta \frac{\partial w(r)}{\partial r}$ and $\sin \theta \frac{\partial w(r)}{\partial r}$. Accordingly, we find the following component-wise integrals for the basic gradient operator eq. (25) as

$$\langle \nabla A \rangle_x \approx \sum_{k=0} \frac{k}{h} b_k \left(\tau_1^A \int_{\mathcal{T}} r^{k+1} \cos^2 \theta d\theta dr + \tau_2^A \int_{\mathcal{T}} r^{k+1} \cos \theta \sin \theta d\theta dr + \tau_3^A \int_{\mathcal{T}} r^k \cos \theta d\theta dr \right), \quad (26)$$

and

$$\langle \nabla A \rangle_y \approx \sum_{k=0} \frac{k}{h} b_k \left(\tau_1^A \int_{\mathcal{T}} r^{k+1} \cos \theta \sin \theta d\theta dr + \tau_2^A \int_{\mathcal{T}} r^{k+1} \sin^2 \theta d\theta dr + \tau_3^A \int_{\mathcal{T}} r^k \sin \theta d\theta dr \right). \quad (27)$$

Note that using canonical trigonometric identities we can replace $\cos^2 \theta$ with $\frac{1}{2}(1 + \cos 2\theta)$ and $\sin^2 \theta$ with $\frac{1}{2}(1 - \cos 2\theta)$, which means that in total there are nine integrals, see Table 1. Considering this set of nine integrals, there are only 4 unique integrals:

$$g_k = \int_{\mathcal{T}} x^k d\theta dx, \quad k \geq 0, \quad (28)$$

$$g_k^{n,0} = \int_{\mathcal{T}} x^k \cos(n\theta) d\theta dx, \quad k \geq 0, n \geq 1, \quad (29)$$

$$g_k^{0,m} = \int_{\mathcal{T}} x^k \sin(m\theta) d\theta dx, \quad k \geq 0, m \geq 1, \quad (30)$$

$$g_k^{n,m} = \int_{\mathcal{T}} x^k \cos(n\theta) \sin(m\theta) d\theta dx, \quad k \geq 0, n \geq 1, m \geq 1. \quad (31)$$

	I	II	III
.	$\int_{\mathcal{T}} r^{k+2} \cos \theta d\theta dr$	$\int_{\mathcal{T}} r^{k+2} \sin \theta d\theta dr$	$\int_{\mathcal{T}} r^{k+1} d\theta dr$
$\frac{\partial}{\partial x}$	$\frac{1}{2} (\int_{\mathcal{T}} r^{k+1} d\theta dr + \int_{\mathcal{T}} r^{k+1} \cos 2\theta d\theta dr)$	$\int_{\mathcal{T}} r^{k+1} \cos \theta \sin \theta d\theta dr$	$\int_{\mathcal{T}} r^k \cos \theta d\theta dr$
$\frac{\partial}{\partial y}$	$\frac{1}{2} (\int_{\mathcal{T}} r^{k+1} d\theta dr - \int_{\mathcal{T}} r^{k+1} \cos 2\theta d\theta dr)$	$\int_{\mathcal{T}} r^{k+1} \sin^2 \theta d\theta dr$	$\int_{\mathcal{T}} r^k \sin \theta d\theta dr$

Table 1: Integral terms for normal and gradient terms, for piecewise-constant (III) and piecewise-polynomial (I-III) interpolations

Note that in a general context, $g_k^{n,0}$ for $k = 0$ would equal g_k , however, in practice, this leads to issues in automated integration, and they require significantly different treatment, so we chose to keep both terms. However, in all cases we investigated, $g_k^{0,m}$ can be treated identically to the more general $g_k^{n,m}$, reducing the number of integrals that need to be solved to 3. It is also important to note, that, if evaluated naively, integrals like $g_k^{n,m}$ can lead to inconsistencies if $n = m$; here, special care needs to be taken. Finally, this process can be done for arbitrary derivatives of the integrals, which, using trigonometric identities, will always break down to the same set of three basic integral terms. Furthermore, this process can be applied to any polynomial that is defined in barycentric coordinates, which includes many widely utilized Finite Element Method (FEM) stencils. Consequently, this means that our approach can be trivially expanded from piecewise-linear to piecewise-n boundary integrals using FEM approaches and, furthermore, this directly implies that our solution allows for the exact (both analytic and algebraic) coupling of an FEM method to SPH. The challenge now becomes integrating these three unique terms over arbitrary triangles, which will be done in the next Section.

4 Integral Terms

In the previous section, we identified 3 integral terms which need to be evaluated for each boundary triangle. However, directly evaluating these terms over an arbitrary triangle is not feasible. Instead, we propose a decomposition approach that combines integrals for geometries which can be integrated directly. The basic geometries in our case are (a) cones, (b) segments and (c) triangles with one vertex at the origin and at least one other vertex in the support domain, referred to as stubs. These basic geometries are then combined to create (a) arbitrary wedges, i.e., triangles with none or one vertex inside the support domain, (b) arbitrary triangles with two vertices in the support domain and (c) arbitrary triangles with three vertices in the support domain. To simplify the integrals, it is helpful to only treat a single spatial configuration, e.g., circular regions where one edge of the triangle coincides with the x-axis. In this regard, the $g_k^{0,0}$ term can be treated straightforwardly, as it is radially symmetric, whereas all other terms require an improper rotation first to transform the spatial configuration to a single base case; see eq. (15). Consequently, we can focus our derivations on a single spatial configuration, as long as any arbitrary shape of the same kind can be transformed into this configuration using an improper rotation. For brevity, we will discuss the evaluation of the integrals for an arbitrary function $g(r, \theta)$, i.e., any choice from the functions in Tab. 1 as the important aspect in this section is finding the right integration bounds. Later on, Section 5 will cover evaluating the integrals over the given bounds and numerical challenges in evaluating the analytic forms. Accordingly, we will consider an integral of the form

$$\int_{\mathcal{T}} g(r, \theta) d\theta dr. \quad (32)$$

Algorithm 1: Procedure to evaluate integrals over discs. This procedure requires as input the radius d of the disc, the reference triangle \mathcal{T} , and the scalar field f . The procedure returns the integral over the region and the gradient.

- 1 **Function** sphericalIntegral(d, \mathcal{T}, f):
 - 2 Calculate $\tau_1^f, \tau_2^f, \tau_3^f$ for \mathcal{T} ; see eq. (23)
 - 3 **Return** $\mathcal{C}^{\mathcal{T}}(d), [\mathcal{C}_x^{\mathcal{T}}(d), \mathcal{C}_y^{\mathcal{T}}(d)]^T$
-

Note that this formulation is independent of the barycentric interpolation, as these are just constant pre-factors of the integrals, and independent of the used kernel function, as we treat g as a black-box function for the purposes of this section. However, a very important caveat to this is that the weights of the barycentric interpolation and the results of gradient terms need to be potentially rotated using a given improper rotation matrix. Accordingly, the processes developed here need to perform the corresponding transformations, even though the actual integrals are not.

4.1 Integrals Over Discs

The first set of integral bounds we need to consider is fundamental integrals over circular sectors, i.e., cases where one edge of the triangle is completely outside the circle and one vertex is at the origin. This case occurs as a subproblem in some other cases and in its most general form requires the integration over an opening from angle α to β and over a limited radius $d \leq 1$. There are two noteworthy special cases of this for integrating over the entire support domain, i.e., from 0 to 2π , and integrals over half the domain, i.e., $-\pi/2$ to $\pi/2$. The former case never requires an adjustment as the result is correct even for all potential geometries, whereas the case of a half support integral and the more general integral between two angles requires an adjustment of the input such that the angle is symmetric with respect to the x-axis. This means that our first integral of concern is

$$\iint_C g(r, \theta) r d\theta dr = \int_0^d \int_0^{2\pi} g(r, \theta) d\theta dr. \quad (33)$$

which leads to a straightforward evaluation, see Algorithm 1. The second case, an integral over half the support domain, simply requires changing the integration bounds to yield

$$\iint_{\text{Half}} g(r, \theta) r d\theta dr = \int_0^d \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} g(r, \theta) d\theta dr. \quad (34)$$

Note that this integral is only required as part of a later integral term and does not require an explicit evaluation process. The final variant of a general sector, the integral covers the range $[\gamma^0, \gamma^0 + \gamma]$ and the radius $[0, \min\{d, 1\}]$. To simplify the integral terms, we assume that the lower angle γ^0 is equal to 0, which can be ensured for any sector through rotation by $-\gamma^0$. Consequently, we can find a function $\mathcal{S}^{\mathcal{T}}$ dependent on an opening angle of the cone γ and a radial limit d as

$$K^{\text{sector}}(\gamma, d) = \mathcal{S}^{\mathcal{T}}(\gamma, d) = \int_0^d \int_0^\gamma g(r, \theta) d\theta dr. \quad (35)$$

For a given triangle \mathcal{T} , whose intersection with the support domain results in a sector, we first determine

Algorithm 2: Procedure to evaluate integrals over sectors. This procedure requires as an input the radius of the sector region d , the reference triangle \mathcal{T} , the scalar field f and two vertices ν_1 and ν_2 that describe the sector, note that ν_0 is implied to be $\mathbf{0}$. The procedure returns the integral over the region and the gradient.

```

1 Function sectorIntegral( $\nu_1, \nu_2, d, \mathcal{T}, f$ ):
2    $\alpha_1 \leftarrow \text{atan2}(y_1, x_1)$ 
3    $s \leftarrow \text{sgn}(x_2 \sin \alpha_1 + y_2 \cos -\alpha_1)$ 
4    $C \leftarrow \begin{bmatrix} \cos(-\alpha_1) & \sin(\alpha_1) & 0 \\ -s \sin(\alpha_1) & s \cos(\alpha_1) & 0 \\ 0 & 0 & 1 \end{bmatrix},$ 
5    $\mathcal{T}' \leftarrow C \cdot \mathcal{T}, \nu'_1 \leftarrow C \cdot \nu_1, \nu'_2 \leftarrow C \cdot \nu_2$ 
6    $\gamma \leftarrow \text{atan2}(\nu'_{2,y}, \nu'_{2,x})$ 
7   Calculate  $\tau_1^f, \tau_2^f, \tau_3^f$  for  $\mathcal{T}'$ ; see eq. (23)
8   Return  $S^{\mathcal{T}'}(d, \gamma), T^{-1} \cdot [S_x^{\mathcal{T}'}(d, \gamma), S_y^{\mathcal{T}'}(d, \gamma)]^T$ 

```

the orientation of ν_1 , i.e., $\alpha = \text{atan2}(y_1, x_1)$ and the sign of the y-coordinate of rotating ν_2 by α as $s = \text{sgn}(x_2 \sin -\alpha + y_2 \cos -\alpha)$ to construct an improper rotation matrix C with an angle $-\alpha$ and a mirroring of the y-axis afterwards if $s < 0$. Accordingly, in the transformed coordinate system the integral bounds go angle-wise from 0 to γ with $\gamma = \text{atan2}(\nu'_{2,y}, \nu'_{2,x})$, for ν'_2 being the result of applying C to ν_2 , and radially from 0 to d . Note that the reference triangle \mathcal{T} also needs to be transformed with C . Algorithm 2 shows how this process can be realized. To actually solve the integrals, we can simply evaluate the most general form, i.e.,

$$\iint_{\text{Sector}} g(k, \theta) d\theta dr = \int_0^d \int_\alpha^\beta g(k, \theta) d\theta dr. \quad (36)$$

4.2 Integrals Over Segments

For circle segments, which are the result of one edge, $\overline{\nu_1 \nu_2}$ by convention, intersecting the support domain and no other edge intersecting the domain, the integral depends on the signed distance of the segment, from the origin, and the relative orientation of the segment. We simplify this configuration by assuming that the segment is orthogonal to the x-axis and that its x-coordinates are equal to the signed distance. We determine the opening angle of the segment with distance r , $d \leq r \leq 1$, as $\gamma(r) = 2 \arccos \frac{r}{d}$ using canonical trigonometry. The angular limits of integration for a given distance is $-\gamma(d)/2$ to $\gamma(d)/2$. Using the *proper* integration bounds, i.e., integrating from d to 1 for the radial limits, we leads to a large number of cases that need to be handled separately. We, therefore, chose to follow [7] and integrate from 0 to 1, leading to complex-valued integrals for any part of the integral that would be *outside* of the segment. Note that for most terms it is necessary to treat $d = 0$ as a special case of integrating angular-wise from $-\pi/2$ to $\pi/2$ and radially from 0 to 1, as the integral terms tend to involve divisions by d . Overall this yields

$$\mathcal{P}^{\mathcal{T}}(d) = \begin{cases} \int_0^1 \int_{\gamma(d)/2}^{\gamma(d)/2} g(r, \theta) d\theta dr, & d \neq 0 \\ \int_0^1 \int_{-\pi/2}^{\pi/2} \tilde{k}(r, \theta) d\theta dr, & \text{else,} \end{cases} \quad (37)$$

Algorithm 3: Procedure to evaluate integrals over segments. This procedure requires as an input the edge that forms the segment described through two vertices t_0 and t_1 , as well as a triangle consisting of vertices v_0 , v_1 and v_2 where, usually, $v_1 = t_0$, $v_2 = t_1$. Furthermore, the reference triangle \mathcal{T} and the scalar field f are required. The procedure returns the integral over the region and the gradient. $\text{lci}(\mathbf{o}, \mathbf{d}, \mathbf{c}, r)$ is a function that evaluates the intersection of a line originating from \mathbf{o} in direction \mathbf{d} with a circle at location \mathbf{c} with radius r and returns an integer describing the number of intersections and the distances along \mathbf{d} at which the intersections occurred.

```

1 Function planarIntegral( $\nu_0, \nu_1, \nu_2, t_0, t_1, \mathcal{T}, f$ ):
2   Calculate signed distance of  $\overline{t_0 t_1}$  using eq. (38)
3   hit,  $\lambda_1, \lambda_2 \leftarrow \text{lci}(t_0, t_1 - t_0, \mathbf{0}, 1)$ 
4    $a \leftarrow t_0 + \lambda_1(t_1 - t_0)$ ,  $b \leftarrow t_0 + \lambda_2(t_1 - t_0)$ 
5    $\theta \leftarrow \arccos d$ ,  $\theta_1 = \text{atan2}(a_x, a_y)$ ,  $\theta_2 = \text{atan2}(b_x, b_y)$ 
6   If  $|(\mathcal{R}_{\theta-\theta_1} a)_x - (\mathcal{R}_{\theta-\theta_1} b)_x| \leq |(\mathcal{R}_{\theta-\theta_1} a)_y - (\mathcal{R}_{\theta-\theta_1} b)_y|$ :
7      $C = \mathcal{R}_{\theta-\theta_1}$ 
8   Else:  $C = \mathcal{R}_{\theta-\theta_2}$ 
9    $\mathcal{T}' \leftarrow C\mathcal{T}$ 
10  Calculate  $\tau_1^f, \tau_2^f, \tau_3^f$  for  $\mathcal{T}'$ ; see eq. (23)
11  Return  $\mathcal{P}^{\mathcal{T}'}(d), C^{-1} \cdot [\mathcal{P}_x^{\mathcal{T}'}(d), \mathcal{P}_y^{\mathcal{T}'}(d)]^T$ 

```

with the signed distance d of the origin $\mathbf{0}$ to the intersecting edge $\overline{\nu_1 \nu_2}$ is given as

$$d = \frac{|(x_2 - x_1)(y_1) - (x_1)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \cdot \begin{cases} -1, & \mathbf{0} \text{ inside} \\ 1, & \text{else.} \end{cases} \quad (38)$$

To ensure the required spatial configuration, we first evaluate the intersections of the edge that forms the segment, i.e., the intersection of $\overline{\nu_1 \nu_2}$ with a circle of radius 1 centered at $\mathbf{0}$, which yields two intersection points \mathbf{x}^p and \mathbf{x}^n . Based on these positions, we then determine their angle of intersections, i.e., $\theta_1 = \text{atan2}(x_y^p, x_x^p)$ and $\theta_2 = \text{atan2}(x_y^n, x_x^n)$, and a third angle using the signed distance $\theta = \text{acos}(d)$, which yields two transformation matrices $\mathcal{R}_{\theta-\theta_1}$ and $\mathcal{R}_{\theta-\theta_2}$. To simplify the integration we now ensure that the intersecting edge is orthogonal, relative to the x-axis, by checking the x-coordinate of the transformed points ν_1 and ν_2 for both transformation matrices and choosing the transformation matrix C that ensures that $[C\nu_1]_x = [C\nu_2]_x = d$. Accordingly, we also need to evaluate a transformed reference triangle $\mathcal{T}' = C\mathcal{T}$. Algorithm 3 describes this overall process.

4.3 Integrals Over Triangle Stubs

A triangle stub is a triangle where one vertex ν_0 , with an associated angle γ , is at the origin $\mathbf{0}$, a second vertex ν_1 , with an associated angle β , is on the x-axis and furthest from the origin relative to the other vertices, and the last vertex ν_2 , with an associated angle α is inside the support domain and above the x-axis. Initially we limit the set of triangles to those where $\alpha \leq \frac{\pi}{2}$, as this case significantly simplifies the integral. In this specific case we can split the integral over the triangle stub into first integrating radially from 0 to $|\nu_2|$ using a sector with angle γ , i.e., by utilizing $\mathcal{S}^{\mathcal{T}}(\gamma, |\nu_2|)$, and then integrating radially from $r_{\min} = |\nu_2|$ to $r_{\max} = \min\{|\nu_1|, 1\}$ with an angle $\omega(r)$ based on the intersection of a circle of radius r with $\overline{\nu_1 \nu_2}$. For this term we first construct a parametric line connecting ν_1 and ν_2 as $g(x) = l \tan \beta - x \tan \beta$ and a line through the origin $h(x) = x \tan \omega$, with an unknown angle ω , which should intersect $g(x)$, i.e., we can

solve $h(x) = g(x)$ for omega to find ω in terms of x as

$$\omega(x) = \arctan \left[\frac{(l-x) \tan \beta}{x} \right] + n\pi, n \in \mathbb{Z}. \quad (39)$$

However, we are interested in ω in terms of r . As we want the intersection to be r away from the origin, we can find an orthogonal triangle with sides x , y , and r with y being either $g(x)$ or $h(x)$. Consequently, as x is the adjacent and r the hypotenuse, this yields $\cos \omega = \frac{x}{r}$, with $x = r \cos \omega$. Using eq. (39) and solving for ω then yields two solutions

$$\omega_1(r) = \arcsin \left[\frac{l \sin \beta}{r} \right] - \beta + 2\pi n, \omega_2(r) = -\arcsin \left[\frac{l \sin \beta}{r} \right] - \beta + 2\pi n + \pi, \quad (40)$$

with $n \in \mathbb{Z}$, where we utilize $\omega_1(r)$ with $n = 0$, i.e.,

$$\omega(r) = \arcsin \left[\frac{l \sin \beta}{r} \right] - \beta. \quad (41)$$

Accordingly, this yields an equation \mathcal{R} as

$$\mathcal{R}^T(d, l, \beta, \gamma) = \mathcal{S}^T(\gamma, |\nu_2|) + \int_d^l \int_0^{\omega(r)} g(r, \theta) d\theta dr, \quad (42)$$

with analogous terms as before for \tilde{k}_x and \tilde{k}_y . Given an arbitrary triangle consisting of vertices ν_0 , at the origin, and two other vertices ν_1 and ν_2 , with at least one of them within the support domain and with the correct range of angles, we first rotate the coordinate system such that the edge connecting ν_0 with the furthest vertex ν_f is in line with the x-axis and then flipping the coordinate system if the closer vertex ν_c is below the x-axis. This can be done using an improper rotation matrix T , which also needs to be applied to \mathcal{T} to yield a transformed triangle \mathcal{T}^T , and yields the following function

$$\mathcal{R}^T(\nu_0, \nu_1, \nu_2) = \mathcal{R}^{\mathcal{T}^T}(|\nu_c|, |\nu_f|, \beta, \gamma), \begin{bmatrix} \mathcal{R}_x^T(\nu_0, \nu_1, \nu_2) \\ \mathcal{R}_y^T(\nu_0, \nu_1, \nu_2) \end{bmatrix} = T^{-1} \begin{bmatrix} \mathcal{R}_x^{\mathcal{T}^T}(|\nu_c|, |\nu_f|, \beta, \gamma) \\ \mathcal{R}_y^{\mathcal{T}^T}(|\nu_c|, |\nu_f|, \beta, \gamma) \end{bmatrix}, \quad (43)$$

with $|\nu_c|, |\nu_f|$ and β as above. This can then be utilized to expand the range of acceptable triangles to any triangle with $\nu_0 = \mathbf{0} \wedge |\nu_2| \leq 1 \wedge |\nu_1| > |\nu_2|$, by subdividing the triangle into two sub-triangles where the greatest angle is $\pi/2$. This can be done straight forwardly by determining the closest point ν_c on the edge connecting $\overline{\nu_1 \nu_2}$, using standard linear algebra, and evaluating \mathcal{R} for a triangle ν_0, ν_c, ν_1 and a triangle ν_0, ν_c, ν_2 to yield

$$\mathcal{G}^T(\nu_0, \nu_1, \nu_2) = \begin{cases} \mathcal{R}^T(\nu_0, \nu_1, \nu_2), & \gamma \leq \frac{\pi}{2}, \\ \mathcal{R}^T(\nu_0, \nu_c, \nu_1) + \mathcal{R}^T(\nu_0, \nu_c, \nu_2), & \text{else,} \end{cases} \quad (44)$$

with analogous terms for \mathcal{G}_x^T and \mathcal{G}_y^T , see Algorithm 4.

4.4 Integrals Over Arbitrary Wedges

An arbitrary wedge is any area that results from the intersection of a triangle with a circle where 0 or 1 vertices are inside of the circle. Note that this also includes cases where there is no overlap, i.e., the integral

Algorithm 4: Procedure to evaluate integrals over triangle stubs. This procedure requires two vertices ν_1 and ν_2 of a triangle as the third vertex ν_0 is assumed to lie at the origin. Furthermore, the reference triangle \mathcal{T} and the scalar field f are required. The procedure returns the integral over the region and the gradient.

```

1 Function triangularIntegral( $\nu_1, \nu_2, \mathcal{T}, f$ ):
2   If  $|\nu_1| < |\nu_2|$ :  $\nu_1, \nu_2 \leftarrow \nu_2, \nu_1$ 
3    $\alpha \leftarrow \arccos \frac{(\nu_1 - \nu_2) \cdot (\nu_0 - \nu_2)}{|\nu_1 - \nu_2| |\nu_0 - \nu_2|}, \gamma \leftarrow \arccos \frac{(\nu_1 - \nu_0) \cdot (\nu_2 - \nu_0)}{|\nu_1 - \nu_0| |\nu_2 - \nu_0|}$ 
4   If  $\gamma \leq \pi/2 \wedge \alpha > \pi/2$ :
5     Return integrateR( $\nu_1, \nu_2, \mathcal{T}, f$ )
6   Else:
7      $\nu_c \leftarrow \text{closestPoint}(\mathbf{0}, \overline{\nu_1 \nu_2})$ 
8      $a, \nabla a = \text{integrateR}(\nu_1, \nu_c, \mathcal{T}, f)$ 
9      $b, \nabla b = \text{integrateR}(\nu_2, \nu_c, \mathcal{T}, f)$ 
10    Return  $a + b, \nabla a + \nabla b$ 
11 Function integrateR( $\nu_1, \nu_2, \mathcal{T}, f$ ):
12    $\nu_0 \leftarrow \mathbf{0}$ 
13    $l \leftarrow |\nu_1|, d \leftarrow |\nu_2|, \theta = \text{atan2}(\nu_1^x, \nu_1^y)$ 
14    $s \leftarrow \text{sgn}(\nu_2^x \sin(-\theta) + \nu_2^y \cos(-\theta)) \geq 0?1:-1$ 
15    $\beta \leftarrow \arccos \frac{(\nu_2 - \nu_1) \cdot (\nu_0 - \nu_1)}{|\nu_2 - \nu_1| |\nu_0 - \nu_1|}, \gamma \leftarrow \arccos \frac{(\nu_1 - \nu_0) \cdot (\nu_2 - \nu_0)}{|\nu_1 - \nu_0| |\nu_2 - \nu_0|}$ 
16    $T \leftarrow \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -s \sin \theta & s \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 
17   Return  $\mathcal{R}^T(l, d, \beta, \gamma)$ 

```

is 0, or the triangle completely overlaps the circle, i.e., the integral is based on the integral over the entire support domain; however, these cases can readily be detected and handled. Furthermore, there being only one edge intersecting the circle, with no vertices inside, is identical to the segment case discussed prior. By convention, we assume that either no vertex is inside of the circle or that ν_0 is in the circle, with the triangle being integrated having positive area, which can both be ensured by reordering the vertex indices accordingly. To handle the special cases, we first evaluate the intersections of all edges with the circle and if no edge intersects the circle we either return 0, if the origin is not in the triangle, an integral over the entire support domain if the origin is in the triangle and an integral over a segment if only a single edge intersects the support domain. Consequently, we only need to handle cases where at least 2 edges intersect the support domain and at most 1 vertex, ν_0 , is inside of the support domain.

Accordingly, we can find the intersections of the adjacent edges, of ν_0 , with the circle as $\lambda_n^1, \lambda_p^1, \lambda_n^2$ and λ_p^2 with $\lambda_n < \lambda_p$. As ν_1 and ν_2 both lie outside the circle, λ_p will always be on the corresponding edge and lie on the circle, whereas λ_n will only be positive if ν_0 was also outside the circle. Next, we determine the two intersecting points $\sigma^1 = \nu_0 + \lambda_p^1 [\nu_1 - \nu_0]$ and $\sigma^2 = \nu_0 + \lambda_p^2 [\nu_2 - \nu_0]$, which both lie on the circle, with $|\sigma^1| = |\sigma^2| = 1$. Using these intersecting points, we can then split the integral over the original triangle \mathcal{T} as an integral over the arc between σ^1 and σ_2 and the integral over the generic triangle stubs $\mathcal{T}_1 = \{\mathbf{0}, \nu_r, \sigma^1\}$ and $\mathcal{T}_2 = \{\mathbf{0}, \sigma^2, \nu_r\}$, i.e., we find

$$\iint_{\mathcal{T}} \tilde{k}(r, \theta) d\theta dr = C^T(\angle \sigma^1 \sigma^2, 1) + G^T(\mathbf{0}, \nu_0, \sigma^1) + G^T(\mathbf{0}, \sigma^2, \nu_0). \quad (45)$$

However, this argumentation only works if the origin $\mathbf{0}$ is within the triangle. To handle cases where the

origin is not within the triangle \mathcal{T} , we multiply each integral result by the sign of the area that is being used for the integration, i.e., we instead use

$$\begin{aligned} \iint_{\mathcal{T}} \tilde{k}(r, \theta) d\theta dr &= \operatorname{sgn}(A(\{\mathbf{0}, \sigma^1, \sigma^2\})) C^{\mathcal{T}}(\angle \sigma^1 \sigma^2, 1) \\ &\quad + \operatorname{sgn}(A(\mathbf{0}, \nu_0, \sigma^1)) G^{\mathcal{T}}(\mathbf{0}, \nu_0, \sigma^1) + \operatorname{sgn}(A(\mathbf{0}, \sigma^2, \nu_0)) G^{\mathcal{T}}(\mathbf{0}, \sigma^2, \nu_0). \end{aligned} \quad (46)$$

If the edge opposite of the reference vertex ν_r , i.e., $\overline{\nu_1 \nu_2}$ intersects the circle then we find two intersection points along this line λ_n^3 and λ_p^3 , which describe a segment that was included, when it shouldn't have, during the integration over the cone from σ^1 to σ^2 and, accordingly, this region needs to be removed from the integral. This region can be described as a segment that has been solved before, where we only need to consider how we can formulate the problem at hand in a way that makes reusing the prior results possible. To achieve this, we first determine a new vertex ν_c that lies outside of the circle and on the opposite side, relative to the origin, of $\overline{\nu_1 \nu_2}$, and using $\mathcal{P}^{\mathcal{T}}$ with the triangle ν_c, ν_1, ν_2 . We then determine the sign of the signed area of the triangle ν_c, ν_1, ν_2 and subtract $\operatorname{sgn}(A(\{\nu_c, \nu_1, \nu_2\})) \mathcal{P}^{\mathcal{T}}(\nu_c, \nu_1, \nu_2)$ from the integral over the cone, i.e., $C^{\mathcal{T}}(\angle \sigma^1 \sigma^2, 1)$ in eq. (46). If the signed area of the sector described by σ_1 and σ_2 was positive then the integral over the planar region has to be flipped, i.e., the integral has to be evaluated over everything but this region, which can readily be achieved by calculating $\mathcal{S}^{\mathcal{T}}(1) - \mathcal{P}^{\mathcal{T}}(\nu_c, \nu_1, \nu_2)$. Accordingly, we can find a function $\mathcal{W}^{\mathcal{T}}(\nu_0, \nu_1, \nu_2)$ that works for all potential cases with 0 or 1 vertex inside the circle and 0 to 3 edges intersecting the circle; see Algorithm 5.

4.5 Integrals Over Triangles

Considering the previous section, the only remaining cases that need to be handled are cases in which either 2 or 3 vertices are within the circle. These cases can be constructed straight-forwardly using the arbitrary wedge function $\mathcal{W}^{\mathcal{T}}(\nu_0, \nu_1, \nu_2)$ by constructing arbitrary wedges that resemble these cases. In the case of two vertices inside the circle, i.e., ν_0 and ν_1 we can construct a point ν_x that lies along the extended edge $\overline{\nu_0 \nu_1}$ and is outside of the circle, e.g., $\nu_x = \nu_0 + 3 \frac{\nu_1 - \nu_0}{|\nu_1 - \nu_0|}$, which yields two triangles $\mathcal{T}_1 = \{\nu_0, \nu_x, \nu_1\}$ and $\mathcal{T}_2 = \{\nu_0, \nu_x, \nu_2\}$, which are wedge shaped, where the overall integral then is

$$T_2^{\mathcal{T}}(\nu_0, \nu_1, \nu_2) = \mathcal{W}^{\mathcal{T}}(\nu_0, \nu_x, \nu_1) - \mathcal{W}^{\mathcal{T}}(\nu_0, \nu_x, \nu_2). \quad (47)$$

Note that analogous cases can be constructed if ν_1 and ν_2 are inside the circle and if ν_0 and ν_2 are inside the circle. In case of three vertices inside of the circle we construct a similar point ν_x^1 along the extended edge $\overline{\nu_0 \nu_1}$ and ν_x^2 along the extended edge $\overline{\nu_0 \nu_2}$. The integral can then be evaluated as

$$T_3^{\mathcal{T}}(\nu_0, \nu_1, \nu_2) = \mathcal{W}^{\mathcal{T}}(\nu_0, \nu_x^1, \nu_x^2) - T_2^{\mathcal{T}}(\nu_1, \nu_2, \nu_x^1) - \mathcal{W}^{\mathcal{T}}(\nu_2, \nu_x^1, \nu_x^2) \quad (48)$$

4.6 Overall integral

Based on the results for wedges \mathcal{W} , i.e., triangles with 0 or 1 vertex inside of the triangle, and the results for 2, T_2 , and 3, T_3 , vertices in the circle the overall integral for an arbitrary triangle $\mathcal{T} = \{\nu_0, \nu_1, \nu_2\}$ can be

Algorithm 5: Procedure to evaluate integrals over arbitrary wedges. This procedure requires three vertices ν_0, ν_1 and ν_2 of a triangle, the reference triangle \mathcal{T} and the scalar field f . The procedure returns the integral over the region and the gradient, with lci defined as in 2

```

1 Function integrateWedge( $\nu_0, \nu_1, \nu_2, \mathcal{T}, f$ ):
2   If no edges intersect:
3     If  $\mathbf{O} \in \text{Triangle}\{\nu_0, \nu_1, \nu_2\}$ : Return  $0, [0, 0]^T$ 
4     Else: Return  $\mathcal{S}(1)$ 
5   If one edge intersects: Return segment integral
6
7    $h_1, \lambda_n^1, \lambda_p^1 \leftarrow \text{lci}(\nu_0, \nu_1 - \nu_0, \mathbf{0}, 1)$ 
8    $\lambda_1 \leftarrow \lambda_p^1 \leq 1? \lambda_p^1 : \lambda_n^1$ 
9    $h_2, \lambda_n^2, \lambda_p^2 \leftarrow \text{lci}(\nu_0, \nu_2 - \nu_0, \mathbf{0}, 1)$ 
10   $\lambda_2 \leftarrow \lambda_p^2 \leq 1? \lambda_p^2 : \lambda_n^2$ 
11   $\sigma_1 \leftarrow \nu_0 + \lambda_1(\nu_1 - \nu_0), \sigma_2 \leftarrow \nu_0 + \lambda_2(\nu_2 - \nu_0)$ 
12   $q \leftarrow \text{Area}(\text{tri}) \geq 0? 1 : 0, p \leftarrow \mathbf{0} \in \text{tri? True : False}$ 
13
14   $a, \nabla a \leftarrow \text{triangularIntegral}(\nu_0, \sigma_1, \mathcal{T}, f)$ 
15   $s_a = \text{sgn}(\text{Area}(\{\mathbf{0}, \nu_0, \sigma_1\}))$ 
16   $b, \nabla b \leftarrow \text{triangularIntegral}(\sigma_2, \nu_0, \mathcal{T}, f)$ 
17   $s_b = \text{sgn}(\text{Area}(\{\mathbf{0}, \sigma_2, \nu_0\}))$ 
18   $c, \nabla c \leftarrow \text{sectorIntegral}(\sigma_1, \sigma_2, 1, \mathcal{T}, f)$ 
19  If third edge intersects:
20     $s, \nabla s \leftarrow \text{sphericalIntegral}(1, \mathcal{T}, f)$ 
21     $\nu_c \leftarrow [\nu_1^x - 2(\nu_2^y - \nu_1^y), \nu_1^y + 2(\nu_2^x - \nu_1^x)]^T$ 
22    If  $\nu_c$  on same side as  $\mathbf{0}$  relative to  $\overline{\nu_1 \nu_2}$ :
23       $\nu_c \leftarrow [\nu_1^x + 2(\nu_2^y - \nu_1^y), \nu_1^y - 2(\nu_2^x - \nu_1^x)]^T$ 
24       $p, \nabla p \leftarrow \text{planarIntegral}(\nu_c, \nu_1, \nu_2, \nu_2, \nu_1)$ 
25      If  $q \geq 0 : p, \nabla p \leftarrow s - p, \nabla s - \nabla p$ 
26       $c, \nabla c \leftarrow c - s_t p, \nabla c - s_t \nabla p$ 
27       $p \leftarrow qc + s_a a + s_b b$ 
28       $\nabla p \leftarrow q \nabla c + s_a \nabla a + s_b \nabla b$ 
29  Return  $p, \nabla p$ 

```

evaluated based on the number of vertices inside the triangle $\#\nu$ as

$$\iint_{\mathcal{T}} g(r, \theta) d\theta dr = \begin{cases} \mathcal{W}^{\mathcal{T}}(\nu_0, \nu_1, \nu_2), & \#\nu \leq 1, \\ T_2^{\mathcal{T}}(\nu_0, \nu_1, \nu_2), & \#\nu = 2, \\ T_3^{\mathcal{T}}(\nu_0, \nu_1, \nu_2), & \#\nu = 3. \end{cases} \quad (49)$$

Finally, it is important to note that utilizing gradient terms that are not using the straightforward formulation, see eq. (25), the methodology is unchanged but requires calculating the weights τ_i^f using modified terms that were discussed prior at the end of Section 5.

5 Closed Form Solutions

Based on the previous observations in Section 3, we can reduce the set of functions that we need to integrate to the following set of three terms

$$g_n^{0,0} = x^n, \quad (50)$$

$$g_n^{a,0} = x^n \cos(a\theta), \quad (51)$$

$$g_n^{a,b} = x^n \cos(a\theta) \sin(b\theta). \quad (52)$$

$$(53)$$

We can further simplify this result using a trigonometric transformation for the third term to remove the cosine term, i.e., we can use $\cos(a)\sin(b) = 1/2\sin(a+b) - 1/2\sin(a-b)$, which in turn yields

$$\frac{1}{2} \sin((a+b)\theta) x^n - \frac{1}{2} \sin((a-b)\theta) x^n. \quad (54)$$

However there are two important notes about this function when integrating it as (a) if $a-b=0$ the second term should not be evaluated, i.e., even though we can compute an integral that is valid for this term, it would not lead to meaningful results and should be avoided, and (b) both terms are more general forms of $\sin(b\theta)$. This means we only need to integrate

$$f_p(x) = x^n \quad (55)$$

$$f_c(x, \theta) = x^n \cos(a\theta) \quad (56)$$

$$f_s(x, \theta) = x^n \sin(b\theta). \quad (57)$$

Before discussing the actual solutions for this integral, we will first discuss more general closed form terms that will be at the core of the integral terms in Section 5.1. After that we will discuss integrals for conical segments briefly, see Sec. 5.2, followed by a more in-depth derivation of the terms for segments, in Sec. 5.3, and stubs in Sec. 5.4.

5.1 Fundamental Closed Form Terms

The first general integral we will consider is for the term

$$\int x^n \sqrt{1 - \frac{d^2}{x^2}}^m dx, \quad (58)$$

which occurs in the majority of terms for segments and stubs. This term has a special solution for most inputs we consider and reads

$$\int x^n \sqrt{1 - \frac{d^2}{x^2}}^m dx = -\frac{x^{n-m+1} d^{m-1} \sqrt{x^2 - d^2}}{\sqrt{1 - \frac{x^2}{d^2}} (m-1-n)} {}_2F_1 \left(-\frac{m}{2}, \frac{n-2(m-1)}{2}, \frac{2+n-2(m-1)}{2}; \frac{x^2}{d^2} \right). \quad (59)$$

There are several special cases where this function becomes ill-defined, e.g., for $n=1 \wedge m=2$. However, the most crucial challenging term is $\sqrt{\frac{x^2-d^2}{1-\frac{x^2}{d^2}}}$ for $x=d$, which is a common lower bound, becomes 0/0. However,

by looking at the limits we find that

$$\lim_{x \rightarrow d^-} \frac{\sqrt{x^2 - d^2}}{\sqrt{1 - \frac{x^2}{d^2}}} = \frac{\sqrt{-d}}{\sqrt{\frac{1}{d}}} = \sqrt{-d}\sqrt{d} = id; \quad \lim_{x \rightarrow d^+} \frac{\sqrt{x^2 - d^2}}{\sqrt{1 - \frac{x^2}{d^2}}} = \frac{\sqrt{d}}{\sqrt{-\frac{1}{d}}} = -\sqrt{-\frac{1}{d}}d^{3/2} = -id. \quad (60)$$

As this term specifically occurs on the lower end of the integral bounds, we pick the right limit of the fraction by evaluating $x + \epsilon$, with an arbitrary small epsilon, if $x = d$. There are six other special cases that we need to treat separately, and are given as

$$n = 1, m = 0 \rightarrow \log(x), \quad (61)$$

$$n = 1, m \neq 0 \rightarrow \frac{1}{n+1}x^{n+1}, \quad (62)$$

$$n = 0, m = 1 \rightarrow \frac{x\sqrt{-d^2 + x^2} - \frac{\text{atan}(\sqrt{-d^2 + x^2})}{x}}{|x|}, \quad (63)$$

$$n = 1, m = 2 \rightarrow \frac{1}{2} \left(-\frac{d^4}{x^2} + x^2 - 4d^2 \log x \right) \quad (64)$$

$$n = 3, m = 2 \rightarrow -d^2 x^2 + \frac{x^4}{4} + d^4 \log x \quad (65)$$

$$n \geq 3, m = 2 \rightarrow x^{n-3} \left(\frac{d^4}{n-3} - \frac{2d^2 x^2}{n-1} + \frac{x^4}{n+1} \right) \quad (66)$$

Evaluating the general term, however, is not as straightforward as these special cases due to involving the ${}_2F_1$, which in general is itself defined as the solution of an integral, i.e., the term is not algebraic for all values. However, for the specific subset of terms we require (which is one of the primary limitations to integer power polynomials), this term can be simplified to a series of algebraic terms that can be evaluated. At the core of these simplifications are a few special cases given as

$${}_2F_1(0, b, c, z) = 1, \quad {}_2F_1(-1, b, c, z) = \frac{c - bz}{c}, \quad {}_2F_1(a, b, b, z) = (1 - z)^{-a}, \quad F(a, b, c, 0) = 1, \quad (67)$$

which are straightforward to implement. Furthermore, using Gauss' contiguous relations allows one to relate any set of three ${}_2F_1$ terms through simple relations if they differ only by integer multiples. Specifically, we utilize

$$(c-1)({}_2F_1(a, b, c-1, z) - {}_2F_1(a, b, c, z)) = b({}_2F_1(a, b+1, c, z) - {}_2F_1(a, b, c, z)) = a({}_2F_1(a+1, b, c, z) - {}_2F_1(a, b, c, z)), \quad (68)$$

which can be used to evaluate the series for arbitrary integer offsets of a from 1 and $1/2$. Using the identities, the only remaining term that we will need is ${}_2F_1(-1/2 + \frac{m}{2}, 1 + b/2, b + 3/2, z)$ as all other required terms can be derived from this term through Gauss' relations. We can then find a closed-form expression for this term for even and odd m , i.e., terms for which the second and third arguments are either integers or half steps away from integers. To define these series expressions, we first define a set of polynomial factors resulting from the relation of the ${}_2F_1$ term to Chebyshev polynomials as

$$S(n) = \binom{2n}{n} \frac{1}{2 \cdot 4^n}, \quad T(n) = \frac{1}{(2n+1)S(n)}, \quad (69)$$

which can then be utilized to define a central series term of the function as

$$F(n, c) = 2 \frac{\sin^{-1} \sqrt{c}}{\sqrt{1-c}} - \sum_{i=0}^{n/2-1} T(i) c^{i+1}, \quad (70)$$

where an important consideration is the definition of the complex inverse sin function. To find consistent results, we utilize the definition used in Mathematica where $\sin^{-1}(c) = -\sin^{-1}(-c)$, even for complex-valued arguments, which stems from the definition using the integral extension, instead of using the logarithm-based extension used in Numpy. Another important note is that this term is very sensitive to numerical errors, as the terms for larger polynomial degrees can become very large, leading to issues of elimination. These, for the cases we consider, can be resolved using Horner's scheme to evaluate the polynomial series on the right side, but in general would need to be handled explicitly, e.g., by using an explicit series representation of the left-hand term. Using this term, the odd Hypergeometric term then becomes

$$B_{\text{odd}}(n, c) = \sqrt{1-c} \left(\frac{n}{n+1} + S\left(\frac{n+1}{2}\right) \frac{1}{c^{n/2}} F(n, c) \right). \quad (71)$$

For even order terms, we first find an equivalent series term

$$G(n, c) = \sum_{j=0}^{n/2+1} S(j) \sqrt{1-c} - \frac{1}{2}. \quad (72)$$

This in turn leads to the even Hypergeometric terms becoming

$$B_{\text{even}}(n, c) = \sqrt{1-c} - \beta\left(\frac{n}{2} + 1, \frac{n}{2}\right) \left(G(n, c) c^{-n/2}\right), \quad (73)$$

with β being the complete β function defined using the complete Γ function as $\beta(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$. Another integral that we require in several places is an integral of the inverse secant function, which can be found in standard integral lists and is given as

$$\begin{aligned} \int_a^b \sec^{-1}(ax) x^m &= -\frac{b^{m+1}}{(m+1)^2} \left((1+m) \sec^{-1}(ab) + a \sqrt{1 - \frac{1}{a^2 b^2}} b {}_2F_1\left(1, \frac{2+m}{2}, \frac{3+m}{2}, a^2 b^2\right) \right) \\ &\quad + \frac{c^{m+1}}{(m+1)^2} \left((1+m) \sec^{-1}(ac) + a \sqrt{1 - \frac{1}{a^2 c^2}} c {}_2F_1\left(1, \frac{2+m}{2}, \frac{3+m}{2}, a^2 c^2\right) \right), \end{aligned} \quad (74)$$

which can be evaluated as above. The final set of integrals we require is based on the Chebyshev Polynomials of the first and second kind, which canonically are given as

$$U_n(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \binom{n-k}{k} 2^{n-2k} x^{n-2k}, \quad (75)$$

$$T_n(x) = \sum_{m=0}^{\lfloor n/2 \rfloor} (-1)^m \left(\binom{n-m}{m} + \binom{n-m-1}{n-2m} \right) \cdot 2^{n-2m-1} \cdot x^{n-2m}. \quad (76)$$

When integrating these terms with respect to x , we can directly utilize the series expansion. While this does

not generally lend itself to finding compact expressions, it does lead to terms that are straightforward to integrate by changing the order of summation and integration. This does potentially lead to a large number of evaluations of integrals, e.g., the square root integral described before, but still yields accurate results.

5.2 Integrals over Cones

Integrating over conical segments is the most fundamental case we need to consider and consists of solving the integral with bounds

$$\int_0^l \int_\alpha^\beta f(x, \theta) d\theta dx, \quad (77)$$

Which for our three terms of concern yields

$$\int_0^l \int_\alpha^\beta f_p(x) d\theta dx = \frac{(\beta - \alpha) l^{n+1}}{n+1}, \quad (78)$$

$$\int_0^l \int_\alpha^\beta f_c(x, \theta) d\theta dx = \left(\frac{\sin(a\beta)}{a} - \frac{\sin(a\alpha)}{a} \right) \frac{l^{n+1}}{n+1}, \quad (79)$$

$$\int_0^l \int_\alpha^\beta f_e(x, \theta) d\theta dx = \left(\frac{\cos(b\beta)}{b} - \frac{\cos(b\alpha)}{b} \right) \frac{l^{n+1}}{n+1}. \quad (80)$$

Note that it can be useful for efficiency to precompute certain special cases, e.g., integrals over the entire support domain, or half the support domain, in advance as these terms occur very regularly. Otherwise, this is all that is necessary for these terms.

5.3 Integrals over Segments

For circular segments we require an integral from $-\cos^{-1}\left(\frac{d}{x}\right)$ to $\cos^{-1}\left(\frac{d}{x}\right)$, where contrary to the previous section, we will first consider f_s as this integral evaluates to 0. Next, we consider the integral for the cosine terms, i.e., f_c , which, after integrating the angular component, yields an integral of the form

$$\int_0^1 \frac{2}{a} \sin\left(a \cos^{-1}\left(\frac{d}{x}\right)\right) x^n dx. \quad (81)$$

To simplify this term, we utilize a somewhat lesser-known identity of the Chebyshev polynomials that is given as:

$$\sin(n \cos^{-1}(x)) = \sqrt{1 - x^2} U_{n-1}(x), \quad (82)$$

which means that we can insert this term into the integral first

$$\frac{2}{a} \int_0^1 \sqrt{1 - \frac{d^2}{x^2}} U_{n-1}\left(\frac{d}{x}\right) x^n dx, \quad (83)$$

which is straightforward to integrate by inserting the series definition of the Chebyshev polynomials of the second kind and flipping the order of integration and summation, i.e., we can expand the term to yield

$$\frac{2}{a} \sum_{k=0}^{\lfloor \frac{a-1}{2} \rfloor} (-1)^k \binom{a-1-k}{k} 2^{a-1-2k} d^{a-1-2k} \int_0^1 \sqrt{1 - \frac{d^2}{x^2}} x^{2k+n+1-a} dx, \quad (84)$$

for which we can use the previously described solution of the square root integral terms. This leaves the integral over the polynomial terms f_p , which gives an integral of

$$2x^n \cos^{-1} \left(\frac{d}{x} \right). \quad (85)$$

Integrating this term is not straight forward directly, however, we can utilize a trigonometric identity as $\cos^{-1}(x) = \sec^{-1}\left(\frac{1}{x}\right)$, to replace the cosine term with a secant term. For this secant term, there exists a well-known closed-form integral, which we described in Sec. 5.1, which can be directly utilized to evaluate this integral. Note that for all of the cases described here, we still utilize the cone segment solutions for $d = -1$, $d = 0$, and $d = 1$, as these terms tend to require special treatment.

5.4 Integrals over Stubs

Moving to the stub triangles we need an integral of the form

$$\int_l^u \int_0^{-\beta + \sin^{-1} \frac{c \sin \beta}{x}} f_n^{a,b}(x, \theta) d\theta dx$$

For readability we can drop the $\sin \beta$ and assume the bounds to be $\sin^{-1}\left(\frac{d}{x}\right) - \beta$. Inserting f_p yields the following term:

$$\left(\sin \left(\frac{\sin d}{x} \right) - \beta \right) x^n = x^n \sin \left(\frac{\sin d}{x} \right) - \beta x^n, \quad (86)$$

where the first term can be integrated as before, i.e., using the arc secant integral term, and the second term is trivial to integrate. Integrating $x^n \cos a\theta$ gives

$$\frac{\sin \left(a \sin \left(\frac{\sin(\beta)l}{x} \right) - a\beta \right) x^n}{a}, \quad (87)$$

which can be simplified using the trigonometric identity $\sin(a + b) = \cos(a)\sin(b) + \sin(a)\cos(b)$ to find

$$\frac{\cos(-a\beta)x^n \sin \left(a \sin^{-1} \left[\frac{d}{x} \right] \right)}{a} + \frac{\sin(-a\beta)x^n \cos \left(a \sin^{-1} \left[\frac{d}{x} \right] \right)}{a}$$

Integrating the $x^n \sin b\theta$ term analogously yields

$$\frac{\left(\frac{1}{b+a} - \frac{\cos \left(b \sin \left(\frac{\sin \beta l}{x} \right) + -b\beta \right)}{b+a} \right) x^n}{2} \quad (88)$$

Simplifying and expanding the trigonometric terms gives

$$\frac{\left(\sin(-b\beta) \sin \left(b \sin \left(\frac{\sin(beta)l}{x} \right) \right) - \cos(-b\beta) \cos \left(b \sin \left(\frac{\sin(\beta)l}{x} \right) \right) + 1 \right) x^n}{2b}$$

For both terms, the central terms are variants of

$$\cos \left(n \sin \left(\frac{dl}{x} \right) \right), \sin \left(n \sin \left(\frac{dl}{x} \right) \right), \quad (89)$$

which can be utilized using the definition of the Chebyshev series, similar to before, as

$$\sin a \left(\sin^{-1} \left(\frac{d}{x} \right) \right) = x U_{a-1} \left(\sqrt{1 - \frac{d^2}{x^2}} \right) \quad (90)$$

$$\cos \left(a \sin^{-1} \left(\frac{d}{x} \right) \right) = T_a \left(\sqrt{1 - \frac{d^2}{x^2}} \right) \quad (91)$$

Both of which are straightforward to solve using the square root integral term derived before, and by inserting the definition of the Chebyshev series as before, and then relying on our solution for the square root term, as all other terms are directly derived from this one.

6 Evaluation

Our proposed method offers an exact analytical and algebraic solution for evaluating boundary integrals over arbitrary 2D triangles, accommodating arbitrary-order quantities and their derivatives. Consequently, evaluating the accuracy of our solution is difficult outside of limited cases with known solutions. Instead, we evaluate its accuracy and efficacy by demonstrating the convergence of established numerical quadrature schemes toward our analytical solution in four distinct evaluation setups: First, we qualitatively compare our solution against conventional numerical quadrature rules on a single triangle. This highlights scenarios where existing methods typically introduce errors, underscoring the precision of our analytical approach. Second, we quantitatively assess convergence using a well-defined test case: random triangulations of $[-1, 1]^2$ with a known global integral and gradient for a piecewise linear quantity. This allows us to observe numerical scheme convergence towards the known solution and allows us to validate our method against a globally exact result, offering insights into its implementation and correctness. Finally, we showcase the practical integration of our boundary integral method within a simple incompressible SPH scheme. While a comprehensive application study is beyond this paper's scope, this demonstration illustrates the potential for enhancing CFD simulations by incorporating our analytical boundary integral solutions.

6.1 Qualitative Evaluation of Quadrature Schemes

To qualitatively evaluate our proposed solutions against numerical schemes, we first compute the boundary integral values for the quantity itself and its gradient for a piecewise-constant field $f(x, y) = 1$ for a small triangle, i.e., a triangle where the longest edge is shorter than the support radius. For such triangles, all integral domains will only partially cover the triangle, and accordingly, no trivial cases are present. We highlight this case in Fig. 1, which showcases good and smooth behavior of the overall solutions with values as expected from this case. To actually evaluate the cases where numerical quadrature schemes have limits, we move to a slightly different test case.

In the second test case here, we evaluate the boundary integral solutions for a large triangle, i.e., one where the support radius is much smaller than the triangle, see Fig. 2. For these geometries, all possible cases discussed in Sec. 4 occur throughout the evaluation domain. Numerical quadratures for triangles cover the entire area of the triangle and utilize a polynomial approximation of the underlying field. While the underlying field we consider in this case is linear, within the triangle, and the kernel function is a polynomial,

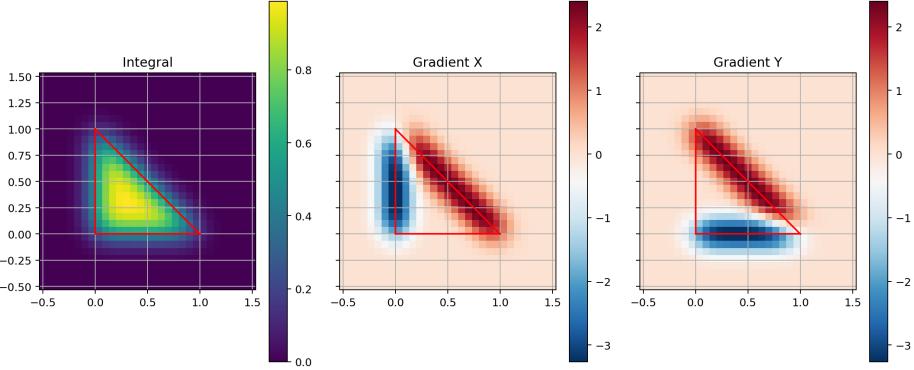


Figure 1: Exemplary evaluation of the boundary integral results for a Cartesian grid of coordinates surrounding a triangle with support $h = 1$, showing the result for the integral and the gradient for a piecewise-constant boundary quantity of 1.

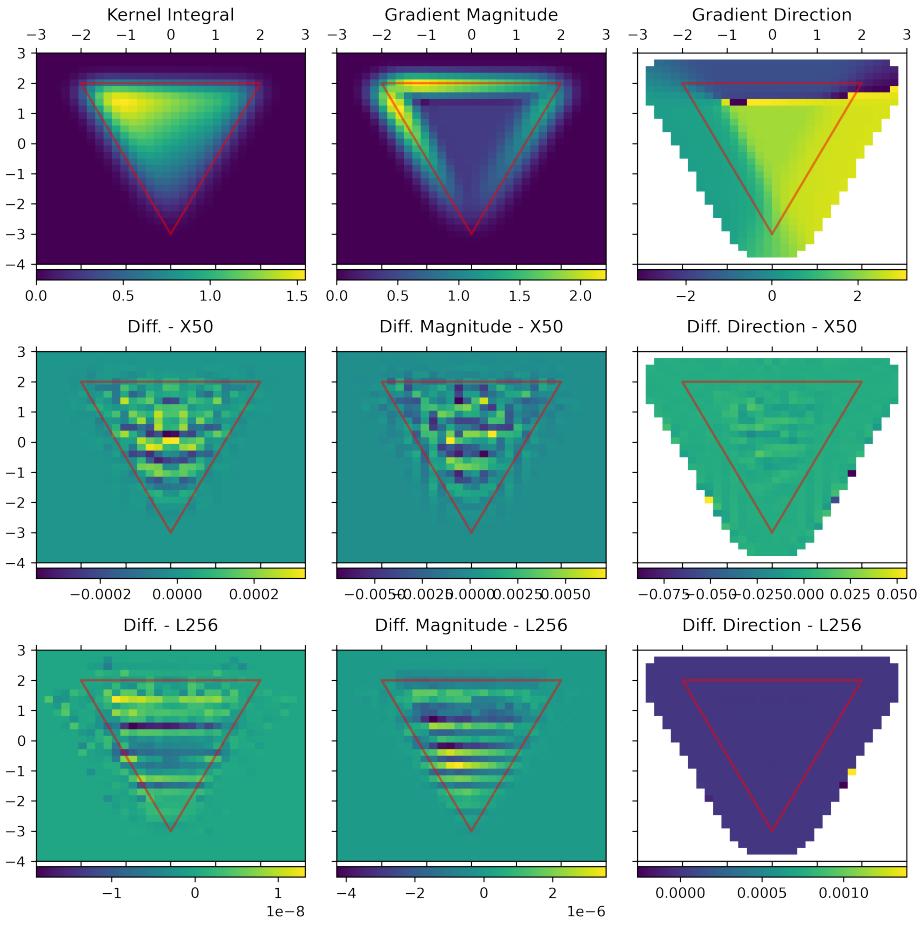


Figure 2: This figure shows the evaluation of the boundary (from left to right) integral as well as the magnitude and direction of the gradient for a triangle using piecewise-linear boundary quantities and a support $h = 1$ for the integral. The top row shows the results evaluated using our analytic solution, the middle row shows the difference between our solution and using the 50th order quadrature scheme of Xiao and Gimbutas [30] and the 256th order scheme of Lether [31] using 65536 quadrature points.

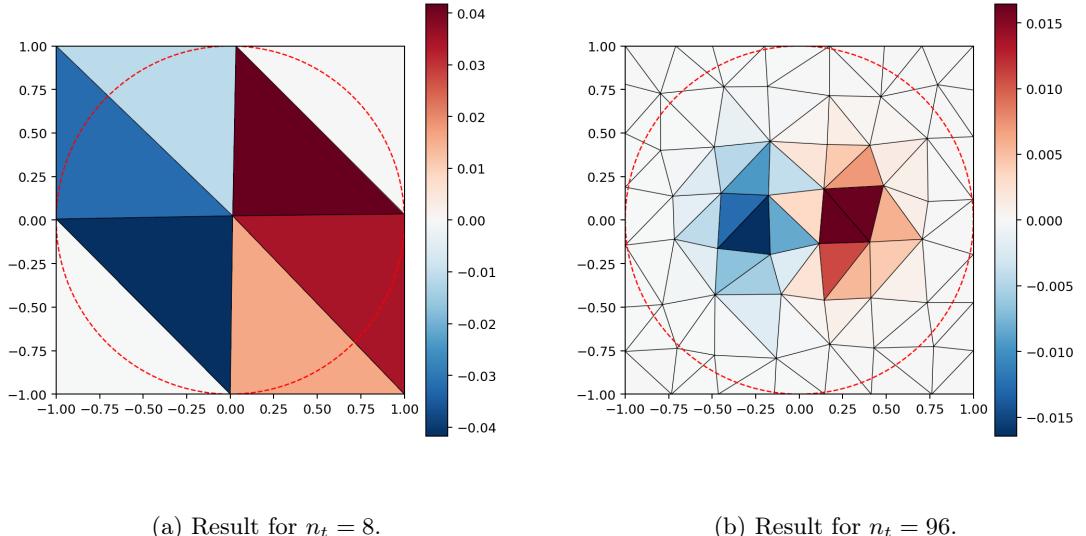


Figure 3: Visualization of the result of evaluating the boundary integral over a set of triangles covering the $[-1, 1]^2$ domain using a test function of $f(x, y) = x$ for two different numbers of triangles. Note that the visualized quantity is after convolving $f(x, y)$ with the kernel function.

the compact nature of the support domain of an SPH evaluation means that the function over the triangle, in this case, is piecewise-polynomial, and approximating this quantity is significantly harder using quadratures. Furthermore, for cases where there is only partial overlap between the support domain and the triangle there might be few, or even no, quadrature points within the support domain.

Consequently, we expect the errors to focus on regions with only marginal overlap with the triangle due to the sparse nature of quadratures and to show the geometric distribution of the quadrature points when the full support domain is within the triangle. Considering the results in Fig. 2, we see both of these effects clearly as the areas of biggest error, with respect to the gradient direction, are for points that are only slightly closer than h to the triangle, with a clear geometric pattern to the integral within the triangle. Note that for small triangles that are fully within the support domain of an evaluation point, numerical quadrature is optimal, assuming the degree of the quadrature rule is appropriate, whereas for partial overlap, the compact nature of the SPH interpolant limits the achievable accuracy. As a result, reducing the size of the triangles used to mesh a boundary geometry should lead to a convergence to the analytic solution as with smaller triangles a larger, relative, part of the domain is evaluated appropriately using quadrature with a smaller, relative, region being subject to the compact cutoff nature of SPH. Consequently, a natural extension of our approach would be to utilize numerical quadrature if the triangle is fully within the support domain and use our analytic solution for all other cases; however, we will focus only on *pure* solutions here.

6.2 Quantitative Evaluation

As discussed before, with numerical quadratures we expect the error of the integral to converge based on the size of the triangles, where we chose two test cases to evaluate this convergence and as test cases to evaluate our analytic solution as these cases have known solutions. The first test case involves coupling a

Numerical Convergence Results

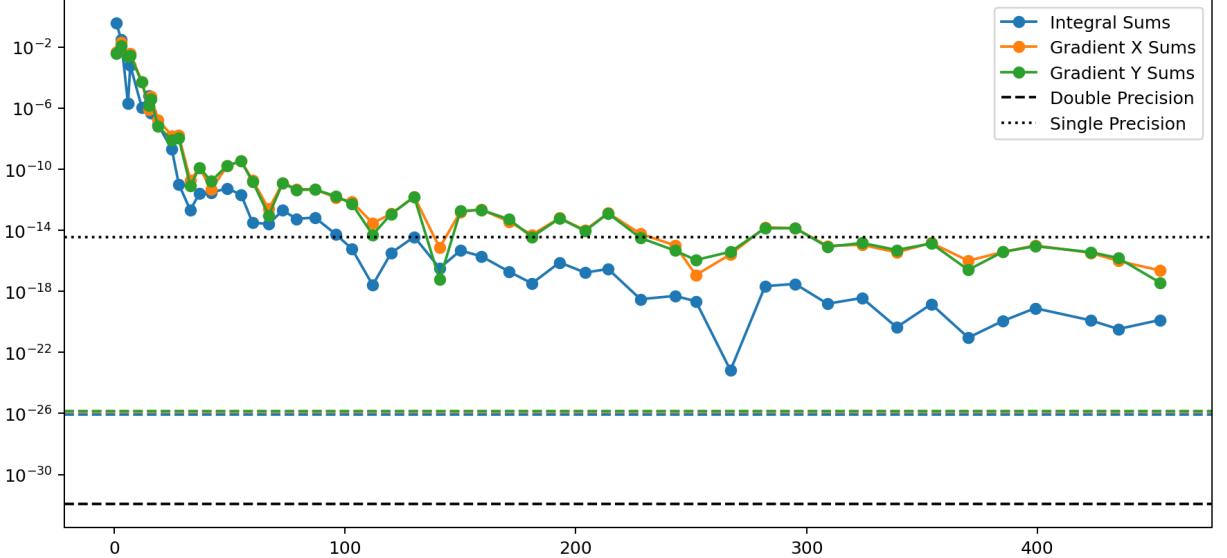
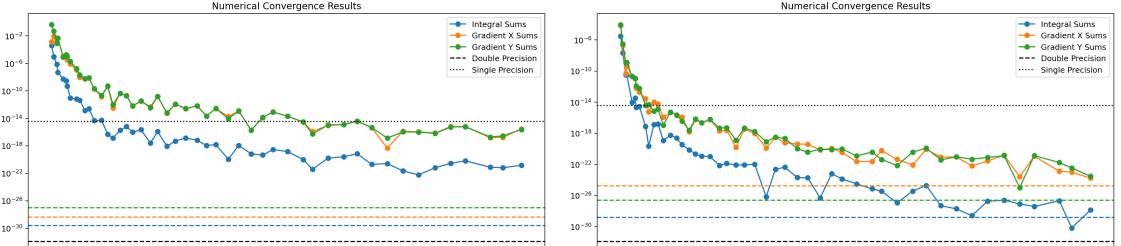


Figure 4: Convergence of the L_2^2 error of the numerical quadrature rule of Xiao and Gimbutas with increasing number of quadrature points per triangle compared to our analytic solution for the integral and gradients for the large triangle problem (see Fig. 3b) with piecewise-constant boundary elements. The black lines indicate machine precision epsilon values.

piecewise constant field with $f(x, y) = 1$ over a coarse triangulation of the $[-1, 1]^2$ domain with 8 mesh elements in total and evaluating the boundary integral for $\mathbf{x} = \mathbf{0}$ with $h = 1$, see Fig. 4, which should yield a boundary integral value of 1 and a gradient of $\mathbf{0}$. As the numerical quadrature scheme here we chose the quadrature rules of Xiao and Gimbutas [30], as they allow us to generate quadrature rules of degree 2 to 50 with increasing quadrature points showing a convergence with respect to the order of quadrature. In our evaluations, we observe a convergence of the numerical scheme towards the known solution at a slow convergence, yielding an L_2 error of 10^{-10} for the integral and 10^{-8} , whereas our analytic solution yields L_2 errors of 10^{-13} for both. While our analytic solution does not yield a solution that is exact within a machine precision epsilon, it is significantly closer to machine precision than any numerical quadrature and the absolute numerical performance could be further improved by changes to the implementation.

As the second test, we then consider $f(x, y) = x$ with a boundary integral of 0 and a gradient of $[1, 0]^T$, for which we evaluate the errors both for a set of 8 large triangles, see Fig. 5a, and a set of 96 smaller triangles, see Fig. 5b. For the large triangles we observe a very similar behavior for the numerical quadratures, as the underlying boundary element is only increased by 1 order and the main difficulty being the compact nature of the SPH kernel, and our analytic solution remains largely unaffected. For the smaller triangles we can see a significant improvement of the L_2 error for the numerical quadratures by 4 orders of magnitude as the central region of the SPH kernel, with the largest values, is now represented by triangles that are fully within the support domain, whereas only a small fraction of triangles with small contributions is affected by the compact cutoff, see Fig. 3. For our analytic solution we observe a slight decrease in the numerical accuracy, especially for the gradient terms, as significantly more numerical operations need to be performed here in total. Overall our analytic solution still remains reliably more accurate than a numerical quadrature rule that required the evaluation of up to $3 \times 96 \times 500 = 144000$ kernel functions to evaluate the quadrature



(a) For large triangle elements, see Fig. 3a

(b) For small triangle elements, see Fig. 3b

Figure 5: Convergence of the L_2^2 error of the numerical quadrature rule of Xiao and Gimbutas with increasing number of quadrature points per triangle compared to our analytic solution for the integral and gradients for the small triangle problem (see Fig. 3b) with piecewise-linear boundary elements. The black lines indicate machine precision epsilon values.

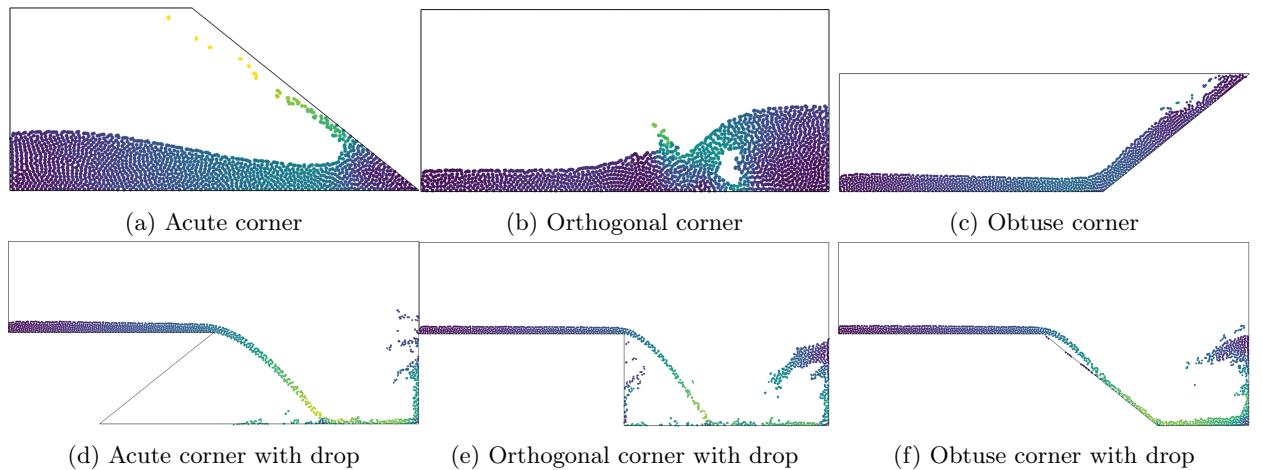


Figure 6: Example simulations of using our boundary integral solution for an incompressible SPH simulation showing smooth flow behavior without artifacts in corners. Velocity color-coded.

points, whereas our analytic solution provided higher accuracy with only 8 mesh elements.

6.3 Incompressible SPH

As our final test problem, we implement our boundary integral scheme in an incompressible SPH simulation, see Appendix 2 for more details on the integration. While developing fluid structure schemes that yield exact and robust results, especially in challenging conditions, is an important aspect of SPH, and more general CFD, research, our boundary integral solution is a general framework that can be used to develop these schemes. Implementing our framework into such a broad range of schemes, including FEM coupling and beyond, is an important area of future research but beyond our scope here, as we focus on the solution of the integrals and how they can be directly integrated into an SPH scheme.¹ This being said, for our test setup we utilize a set of six different boundary geometries that represent a dambreak interacting with a corner that either has an acute, orthogonal or obtuse angle, with half of the test cases including a fluid impact due to the step in the geometry.

¹Note that an implementation of our SPH solver will be available publicly.

In all of our test cases, see Fig. 6, we find a stable and robust simulation that does not yield any boundary penetration or artifacting. These results are in close agreement with those of [7], which utilize a numerical quadrature of 50-th order to compare their semi-analytic solutions against. When evaluating the pressure for the boundary elements we utilized two approaches as (a) computing the pressure only for nodes of the triangle mesh, and using piecewise-linear interpolation and (b) using a local contact point and evaluating the pressure for each contact point (note that this results in a different pressure for the same boundary element for each particle).

For the first approach we found that we needed to utilize very small boundary elements, on the order of $\frac{1}{4}h$, as the pressure field, especially for single particle impacts, is highly non-linear and requires either a small boundary element size, to accurately represent the non-linear field using piecewise-linear functions, or a high order boundary element. This highlights the importance of choosing a correct discretization of the boundary element, to represent the required fluid mechanics, even if the evaluating of the coupling is exact. For the second approach we found no dependence on the boundary element size and were able to utilize very few triangles for the entire domain, e.g., the acute corner impact case in Fig. 6a, consisted of a total of 8 triangles. Our analytic boundary integral in this case yielded exact gradient and kernel interpolation magnitudes, while the local contact point formulation yielded an appropriate pressure value for each particle. Overall this clearly highlights the potential of our analytic boundary integrals and highlights several key directions for future research.

7 Conclusion

In this paper, we presented an analytic and algebraic solution to couple arbitrary (piecewise) order boundary quantities defined on a triangular mesh exactly to an SPH simulation for arbitrary polynomial kernels. Using our analytic solutions, we achieve consistent results across varying resolutions and regardless of triangle geometry, which outperform numerical quadrature-based solutions by up to 5 orders of magnitude, especially for larger triangle elements. As our solution directly models the volume contribution of the boundary, instead of relying on surface integrals, the solution can be applied not solely to gradient quantities, but also directly for SPH interpolants, enabling a seamless and direct integration of SPH with FEM. While using SPH interpolants made it possible to exactly couple from SPH to FEM, our solution closes the gap and enables exact and analytic two-way coupling between FEM and SPH, enabling a promising future avenue of research and coupling.

While our method is analytic and exact, it is not without drawbacks. The evaluation of some of the integral terms can be computationally challenging, especially for GPUs, due to the large number of branching operations. Furthermore, our solution is only applicable to two-dimensional problems, and directly applying our solution to tetrahedra in three dimensions is a significant but promising challenge. Finally, studying the relationship between order of coupling, size of triangle elements, and the resulting fluid behavior is beyond the scope of our work here, but our framework builds the necessary foundations to perform these studies in the future. Note that an implementation of our framework, both in Python and C++, will be made available publicly.

A Integration into Pressure Solvers

To enforce incompressibility, we follow the approach of Bender and Koschier [32], based on Ihmsen et al. [33], where a pressure projection method is used to first ensure that the velocity field is divergence free and then to ensure that the velocity field is incompressible. In this approach the continuity equation $\frac{D\rho}{Dt} = -\rho \nabla \cdot v$ is discretized using a forward difference for the material derivative of density, i.e., $\frac{D\rho}{Dt} = \frac{\rho^{t+\Delta t} - \rho^t}{\Delta t}$ and a difference formulation for the divergence term $\nabla \cdot v$, yielding

$$\begin{aligned} \frac{\rho_i(t + \Delta t) - \rho}{\Delta t} &= \sum_j m_j \mathbf{v}_{ij}^{\text{adv}} \cdot \nabla W_{ij} \\ &\quad + \Delta t \sum_j m_j \left[\frac{\mathbf{F}_i^p}{m_i} - \frac{\mathbf{F}_j^p}{m_j} \right] \cdot \nabla W_{ij}, \end{aligned} \quad (92)$$

with $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, $\nabla W_{ij} = \nabla_{\mathbf{x}_i} W(\mathbf{x}_i - \mathbf{x}_j, h)$ and $\mathbf{F}_i^p = -\sum_j m_j \left[\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right] \nabla W_{ij}$. This yields a system of equations with unknown pressure values p for all particles and an unknown left hand side of the equation system. To solve this system for incompressibility the density in the new timestep is assumed to be equal to the rest density [33, 16], i.e., $\rho_i(t + \Delta t) = \rho_0$, and to solve for divergence-freedom the left hand side is assumed to be equal to 0 [34, 32].

To evaluate the acceleration of a particle i due to pressure forces from other particles, \mathbf{a}_i^f , it is necessary to evaluate the pressure gradient as [15]

$$\mathbf{a}_i^f = -\frac{\nabla p_i}{\rho_i} \approx -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij}, \quad (93)$$

where a symmetric gradient formulation was used to ensure symmetric pressure forces. Evaluating the acceleration of a particle due to pressure forces from a boundary region Ω , \mathbf{a}_i^b , involves evaluating the integral eq. (8) by inserting the pressure field $p(x)$ for A , which yields

$$\mathbf{a}_i^b = -\frac{\nabla p}{\rho} \approx -\int_{\Omega} \rho(\mathbf{x}') \left[\frac{p(\mathbf{x}')}{\rho(\mathbf{x}')^2} + \frac{p_i}{\rho_i^2} \right] \nabla_i W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (94)$$

Where, based on the prior derivation, see Sec. 4, we can evaluate this term for a scalar field $f(\mathbf{x}') = \rho(\mathbf{x}') \left[\frac{p(\mathbf{x}')}{\rho(\mathbf{x}')^2} + \frac{p_i}{\rho_i^2} \right]$ as

$$\mathbf{a}_i^b \approx -\int_{\Omega} f(\mathbf{x}') \nabla_i W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \quad (95)$$

which yields a total acceleration due to pressure forces of a particle i due to neighboring particles j and neighboring triangles \mathcal{T}_j as

$$\begin{aligned} \mathbf{a}_i &= -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij} \\ &\quad - \sum_{\mathcal{T}_j} \int_{\mathcal{T}_j} f(\mathbf{x}') \nabla_i W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \end{aligned} \quad (96)$$

In our implementation we assume that the density of the boundary domain is equal to the rest density

of the boundary object, i.e., $\rho(\mathbf{x}') = \rho_{0,b}$, as we assume boundary objects to be non-deformable and, consequently, their density cannot deviate from their rest density.

Regarding the pressure on the boundary object, $p(\mathbf{x}')$, we propose two distinct choices to model this term. First, a pressure value at each vertex of a triangle could be determined and then barycentrally interpolated across the triangle region and second, a pressure value for the entire triangle could be determined with a distinct value per fluid particle. The first choice involves evaluating a pressure value for each vertex, i.e., $p_1 = p(\nu_1), p_2 = p(\nu_2)$ and $p_3 = p(\nu_3)$ for each triangle, whereas the second choice involves evaluating a pressure value per particle for each triangle $p(\mathbf{x}_{i,\mathcal{T}})$, where $\mathbf{x}_{i,\mathcal{T}}$ is the closest point on the boundary surface $\delta\mathcal{T}$ relative to the position of the particle \mathbf{x}_i .

To evaluate a pressure value at an arbitrary position \mathbf{x} we utilize the approach of Band et. al [34], where the pressure value is evaluated as the result of a Moving Least Squares interpolation. Consequently, for the first choice we need to evaluate a barycentrically interpolated term using the three pressure values p_1, p_2 and p_3 as

$$\mathbf{a}_i^b \approx \iint_{\mathcal{T}} \begin{bmatrix} \tau_1^f g_x^1(r, \theta) + \tau_2^f g_x^2(r, \theta) + \tau_3^f g_x^3(r, \theta) \\ \tau_1^f g_y^1(r, \theta) + \tau_2^f g_y^2(r, \theta) + \tau_3^f g_y^3(r, \theta) \end{bmatrix} d\theta dr, \quad (97)$$

with τ^f defined as before, see Sec. 5, whereas for the second choice the field is constant across the triangle and, consequently, the term can be simplified to yield

$$\mathbf{a}_i^b \approx \rho_{0,b} \left[\frac{p(\mathbf{x}_{i,\mathcal{T}})}{\rho_{0,b}^2} + \frac{p_i}{\rho_i^2} \right] \iint_{\mathcal{T}} \begin{bmatrix} g_x^3(r, \theta) \\ g_y^3(r, \theta) \end{bmatrix} d\theta dr. \quad (98)$$

Accordingly, the first choice requires an evaluation of three pressure values, using MLS, per triangle and a barycentric approach per particle, whereas the second choice requires an evaluation of one pressure value, using MLS, per neighboring triangle per particle and a non barycentric approach per particle. These terms can then be directly integrated into DFSPH [7].

To model boundary terms we follow an approach similar to that of Winchenbach et al. [7], where boundary conditions are modelled by assuming a local planar boundary region. Evaluating boundary condition terms on a per triangle mesh element basis would not yield desirable results as the boundary normals of individual mesh elements does not reflect the overall boundary normal resulting in spurious behavior. To resolve this issue we first evaluate a boundary normal, i.e., we evaluate

$$\mathbf{n}_i = \int_{\mathcal{T}} \nabla W(\mathbf{x}_i - \mathbf{x}', h) d\mathbf{x}', \quad (99)$$

using our proposed analytic boundary integration scheme. It is important to note that this has to be evaluated for any boundary object region, e.g., a particle flowing through a narrow gap inside of a boundary object would otherwise yield an incorrect boundary normal of $\mathbf{0}$. We then evaluate the magnitude of the viscosity term by integrating over all triangle mesh elements and apply only the tangential part, relative to the boundary normal evaluate above, to the particle velocity. It is important to note that most boundary conditions, e.g., a standard Navier-Stokes viscosity term, given by

$$\frac{d\mathbf{v}_i}{dt} = m_i \nu \nabla^2 \mathbf{v}_i, \quad (100)$$

cannot be evaluated in a barycentric boundary formulation. In general, evaluating a direct second order

derivative term leads to a force that is not momentum conserving [27] and instead a finite difference scheme is applied to a first order derivative [35, 27, 6], which yields a discretized laplacian as

$$\nabla^2 \mathbf{A}_i \approx 2(d+2) \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} \frac{\mathbf{A}_{ij} \cdot \mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|^2} \nabla_i W_{ij}. \quad (101)$$

Applying this formulation to the velocity field then yields a viscosity term [28]

$$\frac{d\mathbf{v}_i}{dt} = m_i \nu 2(d+2) \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} \frac{\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|^2 + 0.01h^2} \nabla_i W_{ij}, \quad (102)$$

where the term $0.01h^2$ was added to the denominator to avoid singularities. It is possible to treat this quantity as being linearly interpolated across a boundary element, however, this term is not linear in nature and would yield nonphysical boundary interactions. Accordingly, we opt to only interact with a single planar representation of the boundary, for velocity boundary conditions, with a single contact point [7].

Acknowledgements

Funding was provided by the German Research Foundation (DFG) TH 2034/1-2.

References

- [1] R. A. Gingold and Joseph J. Monaghan. Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly Notices of the Roy. Astronomical Soc.*, 181:375–389, 1977.
- [2] Renato Vacondio, Corrado Altomare, Matthieu De Leffe, Xiangyu Hu, David Le Touzé, Steven Lind, Jean-Christophe Marongiu, Salvatore Marrone, Benedict D. Rogers, and Antonio Souto-Iglesias. Grand challenges for Smoothed Particle Hydrodynamics numerical schemes. *Computational Particle Mechanics*, 8(3):575–588, May 2021.
- [3] M Antuono, PN Sun, S Marrone, and A Colagrossi. The δ -ale-sph model: An arbitrary lagrangian-eulerian framework for the δ -sph model with particle shifting technique. *Computers & Fluids*, 216:104806, 2021.
- [4] Philip F Hopkins. A general class of lagrangian smoothed particle hydrodynamics methods and implications for fluid mixing problems. *Monthly Notices of the Royal Astronomical Society*, 428(4):2840–2856, 2013.
- [5] Nicholas Frontiere, Cody D Raskin, and J Michael Owen. Crksph—a conservative reproducing kernel smoothed particle hydrodynamics scheme. *Journal of Computational Physics*, 332:160–209, 2017.
- [6] Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. In Wenzel Jakob and Enrico Puppo, editors, *Eurographics 2019 - Tutorials*, pages 1–41. The Eurographics Association, 2019.
- [7] Rene Winchenbach, Rustam Akhunov, and Andreas Kolb. Semi-analytic boundary handling below particle resolution for smoothed particle hydrodynamics. *ACM Trans. Graph.*, 39(6), 2020.

- [8] Andrea Colagrossi and Maurizio Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of computational physics*, 191(2):448–475, 2003.
- [9] Larry D Libersky, Albert G Petschek, Theodore C Carney, Jim R Hipp, and Firooz A Allahdadi. High strain lagrangian hydrodynamics: a three-dimensional sph code for dynamic material response. *Journal of computational physics*, 109(1):67–75, 1993.
- [10] Salvatore Marrone, MAGD Antuono, A Colagrossi, G Colicchio, D Le Touzé, and G Graziani. δ -sph model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16), 2011.
- [11] Mehmet Yildiz, RA Rook, and Afzal Suleman. Sph with the multiple boundary tangent method. *International journal for numerical methods in engineering*, 77(10):1416–1438, 2009.
- [12] Aaron English, JM Domínguez, Renato Vacondio, AJC Crespo, PK Stansby, SJ Lind, Luca Chiapponi, and M Gómez-Gesteira. Modified dynamic boundary conditions (mdbc) for general-purpose smoothed particle hydrodynamics (sph): Application to tank sloshing, dam break and fish pass problems. *Computational Particle Mechanics*, 9(5):1–15, 2022.
- [13] Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. Versatile rigid-fluid coupling for incompressible sph. *ACM Transactions on Graphics (TOG)*, 31(4):62, 2012.
- [14] Stefan Band, Christoph Gissler, and Matthias Teschner. Moving least squares boundaries for sph fluids. In *Proceedings of the 13th Workshop on Virtual Reality Interactions and Physical Simulations*, pages 21–28. Eurographics Association, 2017.
- [15] Joe J Monaghan. Simulating free surface flows with sph. *Journal of computational physics*, 110(2), 1994.
- [16] Stefan Band, Christoph Gissler, Markus Ihmsen, Jens Cornelis, Andreas Peer, and Matthias Teschner. Pressure boundaries for implicit incompressible sph. *ACM Transactions on Graphics (TOG)*, 37(2):14, 2018.
- [17] Sivakumar Kulasegaram, Javier Bonet, RW Lewis, and M Profit. A variational formulation based contact algorithm for rigid boundaries in two-dimensional sph applications. *Computational Mechanics*, 33(4):316–325, 2004.
- [18] Martin Ferrand, DR Laurence, Benedict D Rogers, Damien Violeau, and Christophe Kassiotis. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless sph method. *International Journal for Numerical Methods in Fluids*, 71(4):446–472, 2013.
- [19] Jean-Christophe Marongiu, Francis Leboeuf, JoËlle Caro, and Etienne Parkinson. Free surface flows simulations in pelton turbines using an hybrid sph-ale method. *Journal of Hydraulic Research*, 48(sup1):40–49, 2010.
- [20] B. Alessandrini M.D. Leffe, D.L. Touzé. Normal flux method at the boundary for sph. *4th international SPHERIC workshop*, 2009.

- [21] Laurent Chiron, Matthieu De Leffe, Guillaume Oger, and David Le Touzé. Fast and accurate sph modelling of 3d complex wall boundaries in viscous and non viscous flows. *Computer Physics Communications*, 234:93–111, 2019.
- [22] A. Mayrhofer D. Violeau. Exact computation of sph wall renormalising. *9th international SPHERIC workshop*, 2014.
- [23] Agnès Leroy, Damien Violeau, Martin Ferrand, and Christophe Kassiotis. Unified semi-analytical wall boundary conditions applied to 2-d incompressible sph. *Journal of Computational Physics*, 261:106–129, 2014.
- [24] Arno Mayrhofer, Martin Ferrand, Christophe Kassiotis, Damien Violeau, and François-Xavier Morel. Unified semi-analytical wall boundary conditions in sph: analytical extension to 3-d. *Numerical Algorithms*, 68(1):15–34, 2015.
- [25] Dan Koschier and Jan Bender. Density maps for improved sph boundary handling. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, page 1. ACM, 2017.
- [26] Jan Bender, Tassilo Kugelstadt, Marcel Weiler, and Dan Koschier. Volume maps: An implicit boundary representation for sph. In *Motion, Interaction and Games*, page 26. ACM, 2019.
- [27] Daniel J Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):759–794, 2012.
- [28] Joe J Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.
- [29] Walter Dehnen and Hossam Aly. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Monthly Notices of the Royal Astronomical Society*, 425(2):1068–1082, 2012.
- [30] Hong Xiao and Zydrunas Gimbutas. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications*, 59(2):663–676, 2010.
- [31] Frank G. Lether. Computation of double integrals over a triangle. *Journal of Computational and Applied Mathematics*, 2(3):219–224, 1976.
- [32] Jan Bender and Dan Koschier. Divergence-free smoothed particle hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 147–155, 2015.
- [33] Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. Implicit incompressible sph. *IEEE transactions on visualization and computer graphics*, 20(3):426–435, 2013.
- [34] Stefan Band, Christoph Gissler, Andreas Peer, and Matthias Teschner. Mls pressure boundaries for divergence-free and viscous sph fluids. *Computers & Graphics*, 76:37–46, 2018.
- [35] Leigh Brookshaw. A method of calculating radiative heat diffusion in particle simulations. *Publications of the Astronomical Society of Australia*, 6(2):207–210, 1985.