Fully Analytic Higher-Order Boundary Integrals for Two-Dimensional SPH

Rene Winchenbach
Technical University Munich
Physics-based Simulations
Munich, Germany
rene.winchenbach@tum.de

Andreas Kolb
University of Siegen
Computer Graphics and Multimedia Systems Group
Siegen, Germany

Nils Thuerey
Technical University Munich
Physics-based Simulations
Munich, Germany

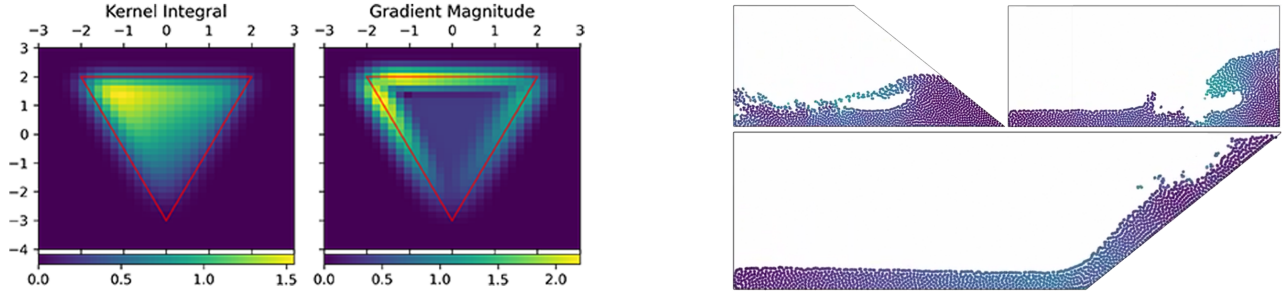

Fig. 1: We derive exact solutions for integrals of kernel functions and kernel gradients for arbitrary triangular elements with barycentric boundary quantities (left, images visualize the result of solving the integral at varying particle positions with $H = 1$). Our solution can also be combined with various boundary approaches, e.g., local MLS-based pressure. We utilize this combination for dam-break simulations with varying corner angles and achieve stable simulations without boundary penetrations (right, velocity color-coded)

***Abstract*— Abstract: In this paper we present a fully analytic boundary handling approach for Smoothed Particle Hydrodynamics in 2D, which works by directly evaluating boundary contributions over triangle meshes using a novel integral factorization for both scalar and gradient terms. In contrast to prior methods that rely upon boundary surface particles or wall-renormalization approaches, our approach can be directly integrated into an SPH formulation without introducing additional terms, e.g., artificial volumes for boundary particles. Furthermore, our method enables assigning quantities, e.g., for pressure values, as constant or linearly varying per triangles using barycentric interpolations which enable more complex boundary interactions, whilst retaining a fully analytic formulation. Moreover, our proposed integral solution works for any triangle geometry, e.g., even degenerate triangle shapes, for polynomial compact kernel functions, e.g., B-spline or Wendland kernels, and regardless of particle and boundary element sizes. To validate our method, we compare the achieved results with a numerical boundary integral.**

Within Smoothed Particle Hydrodynamics, boundary handling has been a long-established issue and serves as one of the Grand Challenges of SPHERIC. The core issue of boundary handling is the discretization of the SPH integral, i.e., $A(\mathbf{x}) \approx \int_\Omega A(\mathbf{x}^\star) W(\mathbf{x}^\star - \mathbf{x})\, dV$, over arbitrary boundary domains Ω_b. In literature, there are two fundamental approaches to solving this integral: (a) The boundary itself is discretized with Lagrangian boundary particles [1], which can then directly interact with fluid particles. However, sampling boundaries with particles leads to several discretization errors, e.g., at boundary corners. (b) Boundary quantities are assumed to be constant for each boundary element Ω_b, i.e., the continuous problem becomes $A(\mathbf{x}) \approx A_{\Omega_b} \int_{\Omega_b} W(\mathbf{x}^\star - \mathbf{x})\, dV$ [2], which is notably easier to evaluate. This is most commonly achieved by evaluating the integral using the divergence theorem via the boundary surface; however, this necessitates that quantities A are constant over the boundary and, at best, results in a piecewise constant boundary approximation if solved per boundary element. Either method introduces significant limitations, e.g., due to low-order approximations or discretization errors, which limit the overall accuracy in boundary handling. A reference implementation for compact kernels is available at https://github.com/wi-re/sphIntegrals.

I. Introduction

Smoothed Particle Hydrodynamics (SPH) is a Lagrangian simulation method originally proposed for astrophysical simulations [3], where rigid bodies only play a minor role. SPH has since found a wide range of applications, including in computational fluid dynamics [4], and computer animation [5], where boundary conditions and rigid bodies are of vital importance to ensure an accurate and realistic simulation. The main issue for incorporating boundary conditions stems from the Lagrangian nature of SPH as rigid object geometries are usually 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. Particle-based rigid boundary handling methods can be distinguished into ghost particle methods [6]–[9], which either place virtual particles inside the boundary that cover the entire support domain of a particle, or on the surface

of the boundary [10]–[12], where either approach may place 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 [11]. Consequently, corrective terms are required to counteract these problems [10], [13], e.g., pseudo-volumes accounting for deficiencies in the particle sampling or to represent the rigid body solely using particles sampled on the boundary surface.

Instead of finding a Lagrangian representation of the boundary, integral formulations directly incorporate a boundary representation, as it is, into the SPH formulation. The most common approach for boundary integrals is determining a wall renormalization factor γ, originally introduced by [14], where the integrals themselves are commonly evaluated using the divergence theorem. There has been significant research in this direction by incorporating gradient operators [15], using semi-analytic [16], [17], numerical [2], [17] or fully analytic solutions to the integrals [18]–[20], for both non-viscous and viscous flows in both 2D and 3D.

Recently in Computer Animation, an alternative to wall renormalization approaches, i.e., *direct evaluation of* the contribution of a boundary object to a fluid particle has been proposed, enabling a straight-forward integration into fluid simulation models. These methods work by either numerically evaluating the contributed density [21] or a representative volume term [22] or by semi-analytically evaluating the kernel function over the boundary domain [23].

In this paper, we propose an extended 2D SPH model that directly incorporates boundary integrals without wall renormalization factors based on the approaches of [21] and [23]. To this extent, our method requires the evaluation of the kernel function and its derivative over arbitrary triangle-based boundary geometries. Furthermore, our method can handle linearly interpolated quantities across boundary triangles, i.e., using barycentric interpolation, enabling more complex interactions compared to prior work [21], [23]. We integrate our model into an existing SPH simulation for incompressible and divergence-free flows where pressure values for the boundary geometry are evaluated either using a local contact-point based pressure estimates or barycentrically interpolated pressure estimates.

II. SPH Background

SPH 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 [24]:

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

where δ is the Dirac delta function.The Dirac delta function is then replaced by a compact kernel function W with a support radius h [24] yielding an approximation for a field quantity A at any point $\mathbf{x}$ based on a compact spherical domain Ω, with radius h and centered at $\mathbf{x}$, as

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

which can also be applied to gradients of generic field quantities to yield

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

SPH then uses particles to discretize the fluid domain, which carry discrete values of the field quantities, where the apparent volume of a particle is used as weights [25], yielding

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

where j refers to the neighboring particles around $\mathbf{x}$, i.e., all particles with $|\mathbf{x}-\mathbf{x}_j| \leq h$, m and ρ 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 [24]. A gradient formulation that is exact for constant functions, but not symmetric, can be derived as

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

also called difference formulation. A gradient formulation that is symmetric, but not exact for constant functions, can be derived as

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

also called symmetric formulation [24]. These formulations discretized, respectively, yield:

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

Note that analogous formulations exist for both divergence $\nabla\cdot A(\mathbf{x})$ and curl $\nabla\times A(\mathbf{x})$ [26]. In their non-discretized forms the difference and symmetric formulations are defined as:

$$\begin{aligned}\nabla A \approx& \frac{1}{\rho(\mathbf{x})}\left[\langle\nabla(\rho A)\rangle - A\langle\nabla\rho\rangle\right]\\ =& \frac{1}{\rho(\mathbf{x})}\int_{\Omega}\rho(\mathbf{x}')\left[A(\mathbf{x}')-A(\mathbf{x})\right]\nabla_{\mathbf{x}} W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}'\\ \nabla A \approx& \rho(\mathbf{x})\left[\left\langle\nabla\left(\frac{A}{\rho}\right)\right\rangle + \frac{A}{\rho^2}\langle\nabla\rho\rangle\right],\\ =& \rho(\mathbf{x})\int_{\Omega}\rho(\mathbf{x}')\left[\frac{A(\mathbf{x}')}{\rho^2(\mathbf{x}')} - \frac{A(\mathbf{x})}{\rho^2(\mathbf{x})}\right]\nabla_{\mathbf{x}} W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}'.\end{aligned} \tag{8}$$

A kernel function W is now defined [27] as:

$$W(\mathbf{x},h) = \frac{1}{h^d}C_d k\left(\frac{|x|}{h}\right), \tag{9}$$

with $k(q)$ being the actual kernel, d the dimension and C_d a normalization factor. Note that, without loss of generality and primarily for legibility, we assume that the support radius and smoothing scale of a kernel are identical. Consequently, the derivative of the kernel function is then defined as

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

where we use the Wendland 4 kernel with $C_2 = \frac{27}{16\pi}$ and $k(q) = [1-q]_+^6\left(1+6q+\frac{35}{3}q^2\right)$, with $[\cdot]_+ = \max\{\cdot, 0\}$.

To enforce incompressibility, we follow the approach of [28], based on [29], 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

$$\begin{aligned}\frac{\rho_i(t+\Delta t)-\rho}{\Delta t} =& \sum_j m_j \mathbf{v}_{ij}^{\text{adv}}\cdot\nabla W_{ij} \\ &+ \Delta t \sum_j m_j \left[\frac{\mathbf{F}_i^p}{m_i} - \frac{\mathbf{F}_j^p}{m_j}\right]\cdot\nabla W_{ij},\end{aligned} \quad (11)$$

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 [13], [29], 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 [28], [30].

III. Boundary Integral Formulation

In our approach individual triangles $\mathcal{T}$ represent a boundary domain Ω, where no requirements are placed on the individual triangles shapes and connectivity, as long as they do not overlap each other. Note that this limitation is not due an inability to handle such cases accurately but rather due to such cases not resulting in physically plausible result, i.e., two objects cannot occupy the same space in reality. 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

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

To 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 support radius is equal to 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 $\mathcal{T}^\star$, and vertices $\nu_i^\star = C\nu_i$, yielding

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

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. For readability, we denote the refactored kernel function $w(r) = C_d k(r)$ with $w(|\mathbf{x}|) = W(\mathbf{x}, 1)$, which yields an integral for the boundary contribution as

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

To simplify the integration process later on it is often necessary to 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 onto the local coordinate system, after scaling and shifting, to yield a new triangle $\mathcal{T}^R$, and vertices $\nu_i^r = R\nu_i^\star$. This, after dropping the subscript of ∇ for readability, yields a term that need to be transformed with the inverse of R, i.e.,

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

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 with 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 gets

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

IV. Piecewise Linear Field Quantities

As arbitrary field quantities $A(x)$ are difficult to handle and hardly appear in SPH, 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. Alterantively, using barycentric interpolation to evaluate A

across a triangle $\mathcal{T}$, the fields defined over boundaries can be piecewise linear. Replacing the generic field $A(\mathbf{x})$ with a barycentric evaluation $A_{\mathcal{T}}(\mathbf{x})$, over the triangle $\mathcal{T}$, and by transforming the integral from cartesian to polar coordinates yields

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

$A_{\mathcal{T}}$ is then 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 in general is defined using the interpolation weights

$$\begin{aligned}
\lambda_0(x,y) &= \frac{(y_1-y_2)(x-x_2)+(x_2-x_1)(y-y_2)}{(y_1-y_2)(x_0-x_2)+(x_2-x_1)(y_0-y_2)} \\
\lambda_1(x,y) &= \frac{(y_2-y_0)(x-x_2)+(x_0-x_2)(y-y_2)}{(y_1-y_2)(x_0-x_2)+(x_2-x_1)(y_0-y_2)} \\
\lambda_2(x,y) &= 1-\lambda_0-\lambda_1,
\end{aligned} \tag{18}$$

where

$$\begin{aligned}
A_{\mathcal{T}}(x,y) =& \lambda_0(x,y)A_0 + \lambda_1(x,y)A_1 + \lambda_2(x,y)A_2 \\
=& A_2 + \lambda_0(x,y)\left[A_0 - A_2\right] + \lambda_1(x,y)\left[A_1 - A_2\right].
\end{aligned} \tag{19}$$

Inserting this back into eq. (17) yields

$$\begin{aligned}
\int_{\mathcal{T}} A(r\cos\theta, r\sin\theta) w(r) r d\theta dr =& A_2 \int_{\mathcal{T}} w(r) r d\theta dr \\
&+ \left[A_0 - A_2\right] \int_{\mathcal{T}} \lambda_0(r\cos\theta, r\sin\theta) w(r) r d\theta dr \\
&+ \left[A_1 - A_2\right] \int_{\mathcal{T}} \lambda_1(r\cos\theta, r\sin\theta) w(r) r d\theta dr,
\end{aligned} \tag{20}$$

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

$$\begin{aligned}
\int_{\mathcal{T}} A_{\mathcal{T}}(r\cos\theta, r\sin\theta) w(r) r d\theta dr =& A_2 \int_{\mathcal{T}} w(r) r d\theta dr \\
&+ \left[\Lambda_0^A + \Lambda_2^A\right] \int_{\mathcal{T}} \left[r\cos\theta - x_2\right] w(r) r d\theta dr \\
&+ \left[\Lambda_1^A + \Lambda_3^A\right] \int_{\mathcal{T}} \left[r\sin\theta - y_2\right] w(r) r d\theta dr,
\end{aligned} \tag{21}$$

with

$$\begin{aligned}
\Lambda_0^A &= \frac{(y_1-y_2)\left(A_0-A_2\right)}{D}, \Lambda_1^A = \frac{(x_2-x_1)\left(A_0-A_2\right)}{D} \\
\Lambda_2^A &= \frac{(y_2-y_0)\left(A_1-A_2\right)}{D}, \Lambda_3^A = \frac{(x_0-x_2)\left(A_1-A_2\right)}{D},
\end{aligned} \tag{22}$$

which can be further refactored to yield

$$\begin{aligned}
\int_{\mathcal{T}} A_{\mathcal{T}}(r\cos\theta, r\sin\theta) r W(r) d\theta dr =& \\
\underbrace{\left[\Lambda_0^A + \Lambda_2^A\right]}_{\tau_1^A} \int_{\mathcal{T}} \underbrace{w(r) r^2 \cos\theta}_{g^I(r,\theta)} d\theta dr& \\
+ \underbrace{\left[\Lambda_1^A + \Lambda_3^A\right]}_{\tau_2^A} \int_{\mathcal{T}} \underbrace{w(r) r^2 \sin\theta}_{g^{II}(r,\theta)} d\theta dr& \\
+ \underbrace{\left[A_2 - \left(\Lambda_0^A x_2 + \Lambda_1^A y_2 + \Lambda_2^A x_2 + \Lambda_3^A y_2\right)\right]}_{\tau_3^A} \int_{\mathcal{T}} \underbrace{w(r) r}_{g^{III}(r,\theta)} d\theta dr =& \\
\int_{\mathcal{T}} \left[\tau_1^A g^I(r,\theta) + \tau_2^A g^{II}(r,\theta) + \tau_3^A g^{III}(r,\theta)\right] d\theta dr.&
\end{aligned} \tag{23}$$

In summary, integrating the SPH interpolant eq. (2) over a single triangle involves finding solutions for the integrals of g^I, g^{II} and g^{III}, where g^I and g^{II} are only required if barycentric interpolation is used, and the third term is required for piecewise linear and constant quantities. To determine gradient terms we first consider a basic gradient formulation, where after inserting the kernel derivative we get for a single triangle

$$\langle \nabla A\rangle \approx -\int_{\mathcal{T}} A_{\mathcal{T}}(r\cos\theta, r\sin\theta) \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \frac{\partial w(r)}{\partial r} r d\theta dr. \tag{24}$$

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. (24) as

$$\langle \nabla A\rangle_x \approx \int_{\mathcal{T}} \left[\tau_1^A g_x^I(r,\theta) + \tau_2^A g_x^{II}(r,\theta) + \tau_3^A g_x^{III}(r,\theta)\right] d\theta dr, \tag{25}$$

and

$$\langle \nabla A\rangle_y \approx \int_{\mathcal{T}} \left[\tau_1^A g_y^I(r,\theta) + \tau_2^A g_y^{II}(r,\theta) + \tau_3^A g_y^{III}(r,\theta)\right] d\theta dr, \tag{26}$$

which results in 6 new integral terms resulting in a total of 9 terms that need to be integrated; see Tab. I.

To perform the actual integration we split the integrals into sub-integrals that can be solved via commonly available mathematical software. These sub-integrals are then combined via constructive solid geometry like approaches to build up the actual entire integral result. As sub-integrals we evaluate integrals over the entire support domain, planar sub-regions, circular wedges and arbitrary wedges, which can then be combined for all cases with zero, one, two or three vertices within the support domain.

	I	II	III
$\cdot$	$g^I(r,\theta) = w(r)r^2\cos\theta$	$g^{II}(r,\theta) = w(r)r^2\sin\theta$	$g^{III}(r,\theta) = w(r)r$
$\frac{\partial}{\partial x}$	$g^I_x(r,\theta) = \frac{\partial w(r)}{\partial r}r^2\cos^2\theta$	$g^{II}_x(r,\theta) = \frac{\partial w(r)}{\partial r}r^2\cos\theta\sin\theta$	$g^{III}_x(r,\theta) = \frac{\partial w(r)}{\partial r}r\cos\theta$
$\frac{\partial}{\partial y}$	$g^I_y(r,\theta) = \frac{\partial w(r)}{\partial r}r^2\cos\theta\sin\theta$	$g^{II}_y(r,\theta) = \frac{\partial w(r)}{\partial r}r^2\sin^2\theta$	$g^{III}_y(r,\theta) = \frac{\partial w(r)}{\partial r}r\sin\theta$

TABLE I: Integral terms for normal and gradient terms, componentwise, for non barycentric (III) and barycentric (I-III) interpolations

V. Solution for non-compact Kernels

While solving the integrals for compact kernels can be done fairly straight forwardly using mathematical software, finding integrals for non-compact kernels is significantly more challenging. Accordingly, extending our approach to Radial Basis Functions directly is not possible. For example the integral for the planar case would look like this:

$$\int_d^\infty \frac{2c^2}{\pi}\operatorname{acos}\left[\frac{d}{x}\right]e^{-c^2r^2}rdr, \tag{27}$$

which cannot be integrated in elementary terms, moreover, common mathematical software cannot transform this integral into more common non-elementary terms. However, changing the coordinate system from polar to cartesian, i.e.

$$\int_d^\infty\int_0^{\operatorname{acos}\left[\frac{d}{x}\right]}\frac{c^2}{\pi}e^{-c^2r^2}rd\theta dr = \int_{-\infty}^\infty\int_d^\infty\frac{c^2}{\pi}e^{-c^2\left(x^2+y^2\right)}dxdy, \tag{28}$$

results in an integral that can at least be simplified to yield

$$\int_{-\infty}^\infty\int_d^\infty\frac{c^2}{\pi}e^{-c^2\left(x^2+y^2\right)}dxdy = \frac{\pi\operatorname{erf}(cd)+\pi}{2\pi}, \tag{29}$$

with erf being the error function defined as

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt, \tag{30}$$

i.e. a non-elementary function. While this does not yield an algebraic integral, it still provides an analytic result and, furthermore, evaluating the erf function is relatively well understood and multiple accurate ways exist to evaluate it, i.e. using a factorial series

$$\operatorname{erf}(z) = \frac{2}{\sqrt{pi}}\sum_{n=0}^\infty\frac{(-2)^n(2n-1)!!}{(2n+1)!}z^{2n+1}. \tag{31}$$

Integrating over triangle stubs is, at least directly, not possible. Instead we first evaluate the integral over a right cone, i.e. a triangle with one vertex at the origin and the angle of the vertex d away from the origin, and on the x-axis, to be $\pi/2$ and the remaining vertex having a positive y coordinate. For these cases we can write the integral as

$$\int_0^d\int_0^{x\tan\beta}\frac{c^2}{\pi}e^{-c^2\left(x^2+y^2\right)}dydx \tag{32}$$

$$= \frac{c}{2\sqrt{\pi}}\int_0^d e^{-c^2x^2}\operatorname{erf}\left(xc\tan\beta\right)dx. \tag{33}$$

For this integral we can then apply the rule from [31] that states

$$\int e^{-(bx)^2}\operatorname{erf}(ax)dx = \frac{2\sqrt{\pi}}{b}\operatorname{OwenT}\left[\sqrt{2}bx, \frac{a}{b}\right], \tag{34}$$

with OwenT being Owen's T function defined as

$$\operatorname{OwenT}[h,a] = \frac{1}{2\pi}\int_0^a\frac{e^{-\frac{1}{2}h^2\left(1+x^2\right)}}{1+x^2}dx; (-\infty < h, a < +\infty), \tag{35}$$

which can be evaluated efficiently and accurately, see [32].

Using this we can then determine the integral over any triangle stub by first evaluating the closest point on the line connecting the furthest point and the point above the y axis. If this point is within the actual triangle we split the triangle into two right cones at this point, where each side will have one right angle at this closest shared point. If the point is not within the actual triangle we still evaluate two triangles but in this case subtract the integral of the one that is entirely outside, see drawing VIII. Note that this split into right triangles, or cones, is also possible for the standard kernel functions but in these cases it becomes less useful as the remaining triangles still need to be evaluated in two parts, see the corresponding sections, instead of two for the entire triangle, i.e. twice the amount of evaluations. In summary we find an integral for the entire domain

$$\iint_{\mathbb{R}^2}\frac{c^2}{\pi}e^{-c^2\left(x^2+y^2\right)}dydx = 1, \tag{36}$$

an integral for a circle of radius d

$$\int_0^r\int_0^{2\pi}\frac{c^2}{\pi}e^{-c^2r^2}rd\theta dr = 1-e^{-(cd)^2}, \tag{37}$$

an integral for a cone of angle γ and infinite radius

$$\int_0^\infty\int_0^\gamma\frac{c^2}{\pi}e^{-c^2r^2}rd\theta dr = \frac{\gamma}{2\pi}, \tag{38}$$

an integral for a cone of angle γ and radius d

$$\int_0^d\int_0^\gamma\frac{c^2}{\pi}e^{-c^2r^2}rd\theta dr = \frac{\left(1-e^{-(cd)^2}\right)\gamma}{2\pi}, \tag{39}$$

an integral for a planar region d away from the origin

$$\int_{-\infty}^\infty\int_d^\infty\frac{c^2}{\pi}e^{-c^2\left(x^2+y^2\right)}dxdy = \frac{\pi\operatorname{erf}(cd)+\pi}{2\pi}, \tag{40}$$

with $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$ and finally for right cones as

$$\int_0^d\int_0^{x\tan\beta}\frac{c^2}{\pi}e^{-c^2x^2}dydx = \operatorname{OwenT}\left[\sqrt{2}cd, \tan\beta\right] - \frac{\beta}{2\pi}, \tag{41}$$

with $\mathrm{OwenT}\left[h,a\right] = \frac{1}{2\pi}\int_0^a \frac{e^{-\frac{1}{2}h^2\left(1+x^2\right)}}{1+x^2}dx$. These terms can then be combined, as for the cubic spline kernel, to yield an overall integral. The gradient of these terms can also be calculated. Barycentric variants can also be derived from these terms by including x and y to the corresponding integrals.

As the support radius of this RBF, however, is infinite, the corresponding cases can be significantly simplified. In all cases all three vertices of the triangle are visible, and no conical sections are required. Accordingly we can (assuming the integral is evaluated at the origin), given a triangle $\mathcal{T} = [\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]$ with $|\mathbf{v}_0| \leq |\mathbf{v}_1|$ and $|\mathbf{v}_0| \leq |\mathbf{v}_2|$ we can create three sub triangles:

$$\mathcal{T}_0 = [\mathbf{0}, \mathbf{v}_0, \mathbf{v}_1], \mathcal{T}_1 = [\mathbf{0}, \mathbf{v}_2, \mathbf{v}_0], \mathcal{T}_2 = [\mathbf{0}, \mathbf{v}_1, \mathbf{v}_2], \quad (42)$$

with corresponding areas a_1, a_2 and a_3. For each of these sub triangles we then first ensure that the closest vertex, besides the origin, is the second vertex, i.e. $|\mathcal{T}[1]| \leq |\mathcal{T}[2]|$, by swapping the second and third vertex if necessary and determining the area of this triangle as a_i^f. We then determine the closest point $\mathbf{c}$ on the line connecting $\mathcal{T}[1]$ and $\mathcal{T}[2]$, relative to the origin, and check if this point lies on the line or outside of it, i.e. Given this point we further create two sub-triangles

$$\mathcal{T}_i^0 = \left[[0,0]^T, \mathbf{c}, \mathcal{T}[1]\right]; \mathcal{T}_i^1 = \left[[0,0]^T, c, \mathcal{T}[2]\right], \quad (43)$$

with corresponding areas a_i^0 and a_i^1. Next we determine both the distances to the closest point $|c$ and the relevant angles in the sub triangles:

$$\beta_i^0 = \mathrm{acos}\left(\frac{\mathcal{T}_i^0[1]\cdot\mathcal{T}_i^0[2]}{|\mathcal{T}_i^0[1]|\cdot|\mathcal{T}_i^0[2]|}\right); \beta_i^1 = \mathrm{acos}\left(\frac{\mathcal{T}_i^1[1]\cdot\mathcal{T}_i^1[2]}{|\mathcal{T}_i^1[1]|\cdot|\mathcal{T}_i^1[2]|}\right),$$

and evaluate the integral term for a right cone accordingly, i.e.

$$F_i^0 = \mathrm{OwenT}\left[\sqrt{2}c|\mathbf{c}|, \tan\beta_i^0\right] - \frac{\beta_i^0}{2\pi}; \quad (44)$$

$$F_i^1 = \mathrm{OwenT}\left[\sqrt{2}c|\mathbf{c}|, \tan\beta_i^1\right] - \frac{\beta_i^1}{2\pi}. \quad (45)$$

Accordingly we find the integral for the overall sub-triangle $\mathcal{T}_i$ as

$$F_i = \mathrm{sgn}\left(a_i^f\right)\left(\mathrm{sgn}\left(a_i^0\right)F_i^0 + \mathrm{sgn}\left(a_i^1\right)F_i^1\right). \quad (46)$$

The overall integral over the triangle is then simply

$$\iint_{\mathcal{T}} \frac{c^2}{\pi}e^{-c^2x^2}dydx = \mathrm{sgn}\left(a_0\right)F_0 + \mathrm{sgn}\left(a_1\right)F_1 + \mathrm{sgn}\left(a_2\right)F_2. \quad (47)$$

Note that this result is not an analytic integral, i.e. Owen's T function is non elementary. However, as efficient evaluation schemes exist for this term we still consider this result at least somewhat interesting, especially regarding how much simpler the overall construction becomes.

VI. Integration into Pressure Solvers

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 [26]

$$\mathbf{a}_i^f = -\frac{\nabla p_i}{\rho_i} \approx -\sum_j m_j\left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right)\nabla_i W_{ij}, \quad (48)$$

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

$$\mathbf{a}_i^b = -\frac{\nabla p}{\rho} \approx -\int_\Omega \rho(\mathbf{x}')\left[\frac{p(\mathbf{x}')}{\rho(\mathbf{x}')^2} + \frac{p_i}{\rho_i^2}\right]\nabla_i W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}'. \quad (49)$$

Where, based on the prior derivation 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

$$\mathbf{a}_i^b \approx -\int_\Omega f(\mathbf{x}')\nabla_i W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}', \quad (50)$$

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

$$\begin{aligned}\mathbf{a}_i = &-\sum_j m_j\left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right)\nabla_i W_{ij} \\ &-\sum_{\mathcal{T}_j}\int_{\mathcal{T}_j} f(\mathbf{x}')\nabla_i W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}'.\end{aligned} \quad (51)$$

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 barycentrically 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 [30], 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

$$\mathbf{a}_i^b \approx \iint_{\mathcal{T}} \begin{bmatrix} \tau_1^f g_x^1(r,\theta) + \tau_2^f g_x^2(r,\theta) + \tau_3^f g_x^3(r,\theta) \\ \tau_1^f g_y^1(r,\theta) + \tau_2^f g_y^2(r,\theta) + \tau_3^f g_y^3(r,\theta) \end{bmatrix} d\theta dr, \tag{52}$$

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

$$\mathbf{a}_i^b \approx \rho_{0,b} \left[\frac{p(\mathbf{x}_{i,\mathcal{T}})}{\rho_{0,b}^2} + \frac{p_i}{\rho_i^2} \right] \iint_{\mathcal{T}} \begin{bmatrix} g_x^3(r,\theta) \\ g_y^3(r,\theta) \end{bmatrix} d\theta dr. \tag{53}$$

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 [23].

To model boundary terms we follow an approach similar to that of Winchenbach et al. [23], 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

$$\mathbf{n}_i = \int_{\mathcal{T}} \nabla W(\mathbf{x}_i - \mathbf{x}', h) d\mathbf{x}', \tag{54}$$

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

$$\frac{d\mathbf{v_i}}{dt} = m_i \nu \nabla^2 \mathbf{v}_i, \tag{55}$$

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 [24] and instead a finite difference scheme is applied to a first order derivative [5], [24], [33], which yields a discretized laplacian as

$$\nabla^2 \mathbf{A}_i \approx 2(d+2) \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} \frac{\mathbf{A}_{ij} \cdot \mathbf{x}_{ij}}{||\mathbf{x}_{ij}||^2} \nabla_i W_{ij}. \tag{56}$$

Applying this formulation to the velocity field then yields a viscosity term [26]

$$\frac{d\mathbf{v_i}}{dt} = m_i \nu 2(d+2) \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} \frac{\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{||\mathbf{x}_{ij}||^2 + 0.01h^2} \nabla_i W_{ij}, \tag{57}$$

where the term $0.01h^2$ was added to the denominator to avoid singularities. It is possible to treat this quantity as being linearily 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 [23].

VII. Discussion

To evaluate our method we evaluated the numerical accuracy of our solutions relative to a numerical boundary integral of high accuracy, e.g., using $\gg 200$ quadrature points, to confirm that our solutions are as close to numerical precision as possible, and found good agreement in all cases to within double precision accuracy. We also evaluated cases with known solutions, e.g., integrals over the entire support domain, by subdividing the domain with triangles and found equally good agreement. This case is also implemented in our implementation available at https://github.com/wi-re/sphIntegrals. Finally, we implemented our boundary integral formulation into an SPH simulation and found stable results without penetration for corner impacts when using piecewise linear pressure, i.e., using MLS extrapolation as described in [23]. However, when using barycentric interpolation we found a strong dependence of stability on boundary element sizes, i.e., the performance was only good when the size of boundary elements was smaller than the inter particle spacing. In the future we would like to investigate prescribed pressures, e.g., for hydrostatic pressure scenarios and more accurate boundary friction formulations with piecewise-linear velocity field extensions into the boundary.

References

[1] J. Domínguez, A. Crespo, P. Stansby, S. Lind, 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," *Comput. Particle Mech.*, vol. 9, no. 5, pp. 1–15, 2022.

[2] L. Chiron, M. De Leffe, G. Oger, and D. Le Touzé, "Fast and accurate sph modelling of 3d complex wall boundaries in viscous and non viscous flows," *Comput. Phys. Commun.*, vol. 234, pp. 93–111, 2019.

[3] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics-theory and application to non-spherical stars," *Mon. Notices Roy. Astron. Soc.*, vol. 181, pp. 375–389, 1977, ISSN: 0035-8711.

[4] R. Vacondio, C. Altomare, M. De Leffe, *et al.*, "Grand challenges for smoothed particle hydrodynamics numerical schemes," *Comput. Particle Mech.*, vol. 8, no. 3, pp. 575–588, 2021.

[5] D. Koschier, J. Bender, B. Solenthaler, and M. Teschner, "Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids," in *Eurographics 2019 - Tutorials*, W. Jakob and E. Puppo, Eds., The Eurographics Association, 2019.

[6] A. Colagrossi and M. Landrini, "Numerical simulation of interfacial flows by smoothed particle hydrodynamics," *J. computational physics*, vol. 191, no. 2, pp. 448–475, 2003.

[7] L. D. Libersky, A. G. Petschek, T. C. Carney, J. R. Hipp, and F. A. Allahdadi, "High strain lagrangian hydrodynamics: A three-dimensional sph code for dynamic material response," *J. computational physics*, vol. 109, no. 1, pp. 67–75, 1993.

[8] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, and G. Graziani, "δ-sph model for simulating violent impact flows," *Comput. Methods Appl. Mech. Eng.*, vol. 200, no. 13-16, pp. 1526–1542, 2011.

[9] M. Yildiz, R. Rook, and A. Suleman, "Sph with the multiple boundary tangent method," *Int. journal for numerical methods engineering*, vol. 77, no. 10, pp. 1416–1438, 2009.

[10] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner, "Versatile rigid-fluid coupling for incompressible sph," *ACM Trans. on Graph. (TOG)*, vol. 31, no. 4, p. 62, 2012.

[11] S. Band, C. Gissler, and M. Teschner, "Moving least squares boundaries for sph fluids," in *Proceedings of the 13th Workshop on Virtual Reality Interactions and Physical Simulations*, Eurographics Association, 2017, pp. 21–28.

[12] J. J. Monaghan, "Simulating free surface flows with sph," *J. computational physics*, vol. 110, no. 2, pp. 399–406, 1994.

[13] S. Band, C. Gissler, M. Ihmsen, J. Cornelis, A. Peer, and M. Teschner, "Pressure boundaries for implicit incompressible sph," *ACM Trans. on Graph. (TOG)*, vol. 37, no. 2, p. 14, 2018.

[14] S. Kulasegaram, J. Bonet, R. Lewis, and M. Profit, "A variational formulation based contact algorithm for rigid boundaries in two-dimensional sph applications," *Comput. Mech.*, vol. 33, no. 4, pp. 316–325, 2004.

[15] M. Ferrand, D. Laurence, B. D. Rogers, D. Violeau, and C. Kassiotis, "Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless sph method," *Int. J. for Numer. Methods Fluids*, vol. 71, no. 4, pp. 446–472, 2013.

[16] J.-C. Marongiu, F. Leboeuf, J. Caro, and E. Parkinson, "Free surface flows simulations in pelton turbines using an hybrid sph-ale method," *J. Hydraul. Research*, vol. 48, no. sup1, pp. 40–49, 2010.

[17] B. A. M.D. Leffe D.L. Touzé, "Normal flux method at the boundary for sph," *4th international SPHERIC workshop*, 2009.

[18] A. M. D. Violeau, "Exact computation of sph wall renormalising," *9th international SPHERIC workshop*, 2014.

[19] A. Leroy, D. Violeau, M. Ferrand, and C. Kassiotis, "Unified semi-analytical wall boundary conditions applied to 2-d incompressible sph," *J. Comput. Phys.*, vol. 261, pp. 106–129, 2014.

[20] A. Mayrhofer, M. Ferrand, C. Kassiotis, D. Violeau, and F.-X. Morel, "Unified semi-analytical wall boundary conditions in sph: Analytical extension to 3-d," *Numer. Algorithms*, vol. 68, no. 1, pp. 15–34, 2015.

[21] D. Koschier and J. Bender, "Density maps for improved sph boundary handling," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, 2017, p. 1.

[22] J. Bender, T. Kugelstadt, M. Weiler, and D. Koschier, "Volume maps: An implicit boundary representation for sph," in *Motion, Interaction and Games*, ACM, 2019, p. 26.

[23] R. Winchenbach, R. Akhunov, and A. Kolb, "Semi-analytic boundary handling below particle resolution for smoothed particle hydrodynamics," in *Proceedings of ACM SIGGRAPH Asia 2020*, 2020.

[24] D. J. Price, "Smoothed particle hydrodynamics and magnetohydrodynamics," *J. Comput. Phys.*, vol. 231, no. 3, pp. 759–794, 2012.

[25] J. J. Monaghan, "Smoothed particle hydrodynamics," *Reports on progress physics*, vol. 68, no. 8, p. 1703, 2005.

[26] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annu. review astronomy astrophysics*, vol. 30, no. 1, pp. 543–574, 1992.

[27] W. Dehnen and H. Aly, "Improving convergence in smoothed particle hydrodynamics simulations without pairing instability," *Mon. Notices Royal Astron. Soc.*, vol. 425, no. 2, pp. 1068–1082, 2012.

[28] J. Bender and D. Koschier, "Divergence-free smoothed particle hydrodynamics," in *Proceedings of the 14th ACM SIGGRAPH/Eurographics symposium on computer animation*, ACM, 2015, pp. 147–155.

[29] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner, "Implicit incompressible sph," *IEEE transactions on visualization computer graphics*, vol. 20, no. 3, pp. 426–435, 2013.

[30] S. Band, C. Gissler, A. Peer, and M. Teschner, "Mls pressure boundaries for divergence-free and viscous sph fluids," *Comput. & Graph.*, vol. 76, pp. 37–46, 2018.

[31] A. Dieckmann, *Table of integrals*, Accessed: 2024-04-17, 2024. [Online]. Available: http://www-elsa.physik.uni-bonn.de/~dieckman/IntegralsIndefinite/IndefInt.html.

[32] M. Patefield, "Fast and accurate calculation of owen's t function," *J. Stat. Softw.*, vol. 5, pp. 1–25, 2000.

[33] L. Brookshaw, "A method of calculating radiative heat diffusion in particle simulations," *Publ. Astron. Soc. Aust.*, vol. 6, no. 2, pp. 207–210, 1985.