

FULL ARTICLE

Efficient quadrature rules for finite element discretizations of nonlocal equations

Eugenio Aulisa¹ | Giacomo Capodaglio^{*2} | Andrea Chierici³ | Marta D'Elia⁴

¹Department of Mathematics and Statistics,
Texas Tech University, TX, USA

²Computational Physics and Methods Group,
Los Alamos National Laboratory, NM, USA

³Department of Industrial Engineering,
University of Bologna, Bologna, Italy

⁴Computational Science and Analysis,
Sandia National Laboratories, CA, USA

Correspondence

*Giacomo Capodaglio, Los Alamos National
Laboratory. Email: gcapodaglio@lanl.gov

Abstract

In this paper we design efficient quadrature rules for finite element discretizations of nonlocal diffusion problems with compactly supported kernel functions. Two of the main challenges in nonlocal modeling and simulations are the prohibitive computational cost and the nontrivial implementation of discretization schemes, especially in three-dimensional settings. In this work we circumvent both challenges by introducing a parametrized mollifying function that improves the regularity of the integrand, utilizing an adaptive integration technique, and exploiting parallelization. We first show that the “mollified” solution converges to the exact one as the mollifying parameter vanishes, then we illustrate the consistency and accuracy of the proposed method on several two- and three-dimensional test cases. Furthermore, we demonstrate the good scaling properties of the parallel implementation of the adaptive algorithm and we compare the proposed method with recently developed techniques for efficient finite element assembly.

KEYWORDS:

Nonlocal models, finite element discretizations, numerical quadrature, asymptotic behavior of solutions

1 | INTRODUCTION

Nonlocal equations have become the model of choice in applications where the global behavior of the system is affected by long-range forces at small scales. In particular, these equations are preferable to partial differential equations (PDEs) in presence of anomalous behavior, such as superdiffusion and subdiffusion, multiscale behavior, and discontinuities or irregularities in the solution that cannot be captured by classical models. For these reasons, nonlocal models are currently employed in several scientific and engineering applications including surface or subsurface transport^[1,2,3,4,5], fracture mechanics^[6,7,8], turbulence^[9,10], image processing^[11,12,13,14] and stochastic processes^[15,16,17,18,19].

The most general form^[20] of a nonlocal operator for a scalar function $u : \mathbb{R}^n \rightarrow \mathbb{R}$ is given by

$$\mathcal{L}u(\mathbf{x}) = 2 \int_{\mathbb{R}^n} (u(\mathbf{y}) - u(\mathbf{x})) \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y},$$

where γ , the kernel, is a compactly supported function over $B_\delta(\mathbf{x})$, the ball of radius δ centered at $\mathbf{x}$. We refer to δ as horizon or interaction radius; this quantity determines the extent of the nonlocal interactions and represents the length scale of the operator. The integral form allows one to catch long-range forces within the length scale and reduces the regularity requirements on the solutions. It also highlights the main difference between nonlocal models and PDEs, i.e., the fact that interactions can occur at distance, without contact. The kernel γ depends on the application and determines the regularity properties of the

solutions; the choice of its parameters or functional form is among the most investigated open questions in the current nonlocal literature^[10,21,22,23,24,25,26,27,28,29]. In this work we limit ourselves to smooth integrable kernels since the treatment of more complex functions is not germane to the issues investigated in this paper, as clarified later on.

The integral nature of the operator poses several modeling and numerical challenges including the treatment of nonlocal interfaces^[30,31], the prescription of nonlocal boundary conditions^[32,33] and the design of efficient discretization schemes and numerical solvers^[34,35,36,37,38,39,40]. In fact, the numerical solution of nonlocal equations becomes prohibitively expensive when the ratio between the interaction radius and the discretization size increases. Even though the nonlocal literature offers several examples of meshfree, particle-type discretizations^[41,42,43,39], in this paper we focus on finite element (FE) methods. This allows us to easily deal with nontrivial domains, achieve high-order accuracy, and use mesh adaptivity. Furthermore, the nonlocal vector calculus^[44] provides a means for a rigorous stability and convergence analysis of variational methods as it allows us to analyze nonlocal diffusion problems in a similar way as elliptic PDEs^[45].

When cast in a variational form, the nonlocal problem associated with the operator $\mathcal{L}$ results in a bilinear form characterized by the following double integral

$$\int_{\Omega \cup \Gamma} \int_{(\Omega \cup \Gamma) \cap B_\delta(\mathbf{x})} (u(\mathbf{y}) - u(\mathbf{x})) (\varphi(\mathbf{y}) - \varphi(\mathbf{x})) \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x}$$

where we explicitly reported the domain of integration in the inner integral. Here, $\Omega \in \mathbb{R}^n$ and Γ are the domain of interest and the corresponding “nonlocal boundary” and φ is an appropriate test function. Thus, the variational setting introduces further computational challenges. Not only do we have to numerically evaluate a double integral, but the integrand function is discontinuous, due to the compact support of γ and to the fact that, in FE settings, the tests functions are also compactly supported.

The paper by D’Elia et al.^[37] thoroughly describes the challenges associated with nonlocal FE discretizations and proposes approximation techniques for efficient and accurate implementations. In particular, the authors introduce “approximate balls” that facilitate the assembly procedure by substituting the Euclidean ball $B_\delta(\mathbf{x})$ with suitable polygonal approximations. They also suggest a set of quadrature rules for the outer and inner integration and analyze the convergence properties of the resulting scheme. With the same spirit, in this work, we propose an alternative way to efficiently evaluate the integral above by circumventing the issue of integrating a truncated function. The key idea of this paper is the introduction of a mollifier^[46] to approximate the discontinuous kernel function; by doing so, the new, approximate, and parameterized kernel is a smooth function for which standard Gaussian quadrature rules can be employed over every element without compromising their accuracy. Additionally, we introduce adaptive quadrature rules for the numerical integration of the outer integral. In fact, contrary to intuition, sophisticated integration techniques for the outer integral are required in order to prevent the quadrature error from exceeding the FE one^[37].

The main contributions of this paper are

- The introduction of a parametrized, smooth, approximate kernel, by means of a mollifier, that yields a smooth integrand over every element. This allows us to avoid the tedious and impractical task of determining the intersections of the ball with the elements, and hence represents a major advantage of our method in three-dimensional simulations.
- The design of adaptive quadrature rules for the outer integral and of a parallel algorithm for efficient simulations.
- The theoretical proof and numerical illustration of the convergence of the approximate, mollified solution to the analytic one as the mollified kernel approaches γ .
- The numerical illustration of the convergence of the mollified solution to the exact one as we refine the mesh and a numerical study of the dependence of the convergence behavior with respect to the parameters.
- The demonstration via two-dimensional and three-dimensional numerical tests of the scalability of our algorithm.

Outline of the paper

In the following section we define the notation that is used throughout the paper and recall important results on nonlocal calculus. In Section 3 we introduce the mollifier function and the associated approximate, parametrized weak form of the nonlocal diffusion problem. We also analyze the convergence of the solution of the latter to the original weak solution. In Section 4 we describe the nonlocal FE discretization, with special focus on the assembly procedure, and briefly recall its challenges. In Section 5 we introduce adaptive quadrature rules for the numerical integration of the outer integral. In Section 6 we illustrate our theoretical results via two- and three-dimensional tests. We also discuss a parallel implementation of the FE assembly procedure and show the corresponding scaling results. Finally, we summarize our contributions in Section 7.

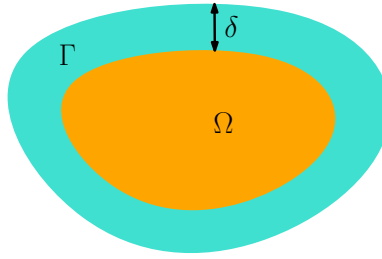


FIGURE 1 Example of domain Ω with associated interaction domain Γ .

2 | PRELIMINARIES

Let $\Omega \in \mathbb{R}^n$ be open and bounded, $n = 1, 2, 3$. Given some $\delta > 0$, we define the interaction domain Γ of Ω as the set of all points not in Ω that are within a δ distance from points in Ω , i.e.

$$\Gamma = \{y \in \mathbb{R}^n \setminus \Omega : |x - y| \leq \delta \text{ for some } x \in \Omega\}, \quad (1)$$

see Figure 1 for a graphical example in $\mathbb{R}^2$. Note that Γ depends on δ even if it is not explicitly indicated. Let $\gamma : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ be an integrable nonnegative symmetric kernel that is also radial^a, namely $\gamma(x, y) = \gamma(y, x)$ and $\gamma(x, y) = \gamma(|x - y|)$. We also assume that γ has bounded support over the ball of radius δ centered at x , i.e. $B_\delta(x)$. For a scalar function $u : \mathbb{R}^n \rightarrow \mathbb{R}$ we define the nonlocal Laplacian as

$$\mathcal{L}u(x) = 2 \int_{\mathbb{R}^n} (u(y) - u(x))\gamma(x, y) dy. \quad (2)$$

The strong form of a nonlocal Poisson problem is then given by: for $f : \Omega \rightarrow \mathbb{R}$, and $g : \mathbb{R}^n \setminus \Omega \rightarrow \mathbb{R}$, find u such that

$$\begin{cases} -\mathcal{L}u(x) = f(x), & x \in \Omega \\ u(x) = g(x), & x \in \Gamma \end{cases} \quad (3)$$

where the second condition in (3) is the nonlocal counterpart of a Dirichlet boundary condition for PDEs and it is referred to as *Dirichlet volume constraint*^b. Such condition is required⁴⁵ to guarantee the well-posedness of (3). The weak form of the Poisson problem is obtained by multiplying the first equation in (3) by a test function $\varphi = 0$ in Γ and by applying the nonlocal first Green's identity⁴⁴. This yields

$$\begin{aligned} 0 &= \int_{\Omega} (-\mathcal{L}u - f) \varphi dx \\ &= \iint_{(\Omega \cup \Gamma)^2} (u(y) - u(x))(\varphi(y) - \varphi(x)) \gamma(x, y) dy dx - \int_{\Omega} f(x) \varphi(x) dx, \end{aligned} \quad (4)$$

Then, the weak form of the nonlocal diffusion problem reads as follows. For $f \in V'$ and $g \in V_\Gamma$, find $u \in V$ such that

$$\mathcal{A}(u, v) = F(v), \quad \forall v \in V_0, \quad \text{subject to } u = g \text{ in } \Gamma, \quad (5)$$

where

$$\begin{aligned} \mathcal{A}(u, \varphi) &= \iint_{(\Omega \cup \Gamma)^2} (u(y) - u(x))(\varphi(y) - \varphi(x)) \gamma(x, y) dy dx, \\ F(\varphi) &= \int_{\Omega} f(x) \varphi(x) dx, \end{aligned} \quad (6)$$

^aFor a discussion on nonpositive kernels and nonsymmetric kernels, see Mengesha and Du⁴⁷ and D'Elia et al.¹⁶, respectively.

^bFor definition and analysis of Neumann volume constraints we refer to Du et al.⁴⁸ and for its numerical treatment we refer to, e.g., D'Elia et al.⁴³.

and where the function spaces are defined as

$$\begin{aligned} V &= \{\varphi \in L^2(\Omega \cup \Gamma) : |||\varphi||| < \infty \text{ and } v|_{\mathbb{R}^n \setminus \Omega} = 0\} \\ V_0 &= \{\varphi \in V : \varphi|_{\Gamma} = 0\}, \\ V_{\Gamma} &= \{p : \Gamma \rightarrow \mathbb{R} : \exists \varphi \in V \text{ such that } \varphi|_{\Gamma} = p\}. \end{aligned} \quad (7)$$

Here, the *energy* semi-norm $||| \cdot |||$ is defined as

$$|||\varphi|||^2 = \mathcal{A}(\varphi, \varphi), \quad (8)$$

the space V' is the dual space of V and V_{Γ} is a nonlocal trace space. Note that since the kernel is integrable and translation invariant, the energy semi-norm is a norm in the constrained space V_0 and satisfies a Poincaré inequality^[45]. Furthermore, by construction, the bilinear form $\mathcal{A}(\cdot, \cdot)$ defines an inner product on V_0 and it is continuous and coercive with respect to the energy norm $||| \cdot |||$. Finally, the latter is equivalent to the L^2 norm; this allows us to establish an equivalence relationship between V and $L^2(\Omega \cup \Gamma)$. Together with the continuity of $\mathcal{F}$, these facts yield the well-posedness of the weak form (5)^[45].

As commonly done in the PDE context, we recast the problem in V_0 by simply rewriting the solution as $u = w + \tilde{g}$, where $w \in V_0$ and $\tilde{g} \in V$ is an extension of g to zero into $\Omega \cup \Gamma$, known in the FE framework as a lifting function. Thus, equation (5) can be rewritten in terms of w as follows

$$\begin{aligned} \iint_{(\Omega \cup \Gamma)^2} (w(\mathbf{x}) - w(\mathbf{y})) (\varphi(\mathbf{x}) - \varphi(\mathbf{y})) \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} &= \int_{\Omega} f(\mathbf{x}) \varphi(\mathbf{x}) d\mathbf{x} \\ &+ \iint_{(\Omega \cup \Gamma)^2} (\tilde{g}(\mathbf{x}) - \tilde{g}(\mathbf{y})) (\varphi(\mathbf{x}) - \varphi(\mathbf{y})) \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x}, \quad \forall \varphi \in V_0, \end{aligned} \quad (9)$$

or, equivalently,

$$\mathcal{A}(w, \varphi) = \tilde{F}(\varphi) \quad \forall \varphi \in V_0. \quad (10)$$

The latter is useful for implementation purposes as it allows us to automatically take into account the presence of a non-homogeneous Dirichlet volume constraint.

3 | WEAK FORM APPROXIMATION

We introduce a parametrized approximation of the bilinear form $\mathcal{A}$ defined in (6) with the purpose of obtaining a weak problem that is computationally less challenging. Specifically, the approximated bilinear form is associated with a parametrized kernel function that is still integrable, radial, and compactly supported, but not discontinuous in $\Omega \cup \Gamma$. This fact makes the numerical integration of the inner integral in, e.g., (5), a much simpler task, compared to the case of discontinuous kernel functions.

For simplicity of exposition, we rewrite the “exact” kernel γ as

$$\gamma(\mathbf{x}, \mathbf{y}) = C_{\delta} \eta(\mathbf{x}, \mathbf{y}) \mathcal{X}(\mathbf{y} \in B_{\delta}(\mathbf{x})) \quad (11)$$

where C_{δ} is a scaling constant that guarantees that the nonlocal operator $\mathcal{L}$ associated with γ is such that $\mathcal{L} \rightarrow \Delta$ as $\delta \rightarrow 0$. Clearly, by definition, $\eta(\mathbf{x}, \mathbf{y}) = \eta(|\mathbf{x} - \mathbf{y}|)$. Given $\varepsilon \in \mathbb{R}^+$, we approximate the bilinear form $\mathcal{A}(\cdot, \cdot)$ defined in (6) with the parametrized bilinear form $\mathcal{A}_{\varepsilon}(\cdot, \cdot)$ obtained by replacing the kernel $\gamma(\mathbf{x}, \mathbf{y})$ with

$$\gamma_{\varepsilon}(\mathbf{x}, \mathbf{y}) = C_{\delta, \varepsilon} \eta(\mathbf{x}, \mathbf{y}) \mu_{\delta, \varepsilon}(\mathbf{x}, \mathbf{y}), \quad (12)$$

where $\mu_{\delta, \varepsilon}$ is an appropriately scaled mollifier function. Inspired by the mollifier function introduced in Mousavi et al.^[46], for $\varepsilon < \delta$ we define $\mu_{\delta, \varepsilon} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ as the following radial function

$$\begin{aligned} \mu_{\delta, \varepsilon}(|\mathbf{x} - \mathbf{y}|) &= \begin{cases} 1 & \text{for } 0 \leq |\mathbf{x} - \mathbf{y}| < \delta - \varepsilon \\ \xi \left(\frac{(\delta - \varepsilon) - |\mathbf{x} - \mathbf{y}|}{\varepsilon} \right) & \text{for } \delta - \varepsilon \leq |\mathbf{x} - \mathbf{y}| \leq \delta + \varepsilon \\ 0 & \text{for } |\mathbf{x} - \mathbf{y}| > \delta + \varepsilon \end{cases} \\ \xi(r) &= \left(\frac{128}{256} + \frac{315}{256}r - \frac{420}{256}r^3 + \frac{378}{256}r^5 - \frac{180}{256}r^7 + \frac{35}{256}r^9 \right), \end{aligned} \quad (13)$$

where, for given $\epsilon > 0$, $C_{\delta,\epsilon}$ is such that the nonlocal operator $\mathcal{L}_\epsilon$, associated with γ_ϵ , converges to Δ as $\delta \rightarrow 0$. The constant $C_{\delta,\epsilon}$ is also such that it converges to C_δ as $\epsilon \rightarrow 0$. Furthermore, it follows from the definition of $\mu_{\delta,\epsilon}$ that

$$\lim_{\epsilon \rightarrow 0} \mu_{\delta,\epsilon}(\mathbf{x}, \mathbf{y}) = \mathcal{X}(\mathbf{y} \in B_\delta(\mathbf{x})),$$

which, together with the property of $C_{\delta,\epsilon}$, implies that γ_ϵ converges pointwise to γ as $\epsilon \rightarrow 0$. Note that the support of the mollifier is bigger than the one of the original kernel function γ as it corresponds to $B_{\delta+\epsilon}(\mathbf{x})$. The parametrized bilinear form $\mathcal{A}_\epsilon(\cdot, \cdot)$ is therefore defined as follows

$$\mathcal{A}_\epsilon(u, \varphi) = \iint_{(\Omega \cup \Gamma)^2} (u(\mathbf{y}) - u(\mathbf{x}))(\varphi(\mathbf{y}) - \varphi(\mathbf{x}))\gamma_\epsilon(\mathbf{x}, \mathbf{y})d\mathbf{y}d\mathbf{x}, \quad (14)$$

and consequently the approximate weak formulation of the nonlocal volume-constrained problem now reads: find $u_\epsilon \in V$ such that

$$\mathcal{A}_\epsilon(u_\epsilon, \varphi) = F(\varphi) \quad \forall \varphi \in V_0 \quad \text{subject to } u_\epsilon = g \text{ on } \Gamma. \quad (15)$$

Note that the weak formulation above is defined over the same function spaces of (5). This is allowed because the parametrized kernel γ_ϵ belongs to the same class of kernels as γ . As a consequence, problem (15) is also well-posed.

3.1 | Convergence of the approximate weak solution

We find a bound for the energy norm of the difference between solutions of the weak form (5) and (15). First, we note that by subtracting (5) from (15) we obtain

$$\mathcal{A}(u, \varphi) = \mathcal{A}_\epsilon(u_\epsilon, \varphi) \quad \forall \varphi \in V_0. \quad (16)$$

Our ultimate goal is to find a bound for $\|u - u_\epsilon\|$, or, equivalently, for $|\mathcal{A}(u - u_\epsilon, u - u_\epsilon)|$, being u and u_ϵ solutions of (5) and (15) respectively. We first consider a generic test function; equality (16) implies

$$\begin{aligned} |\mathcal{A}(u - u_\epsilon, \varphi)| &= |\mathcal{A}(u, \varphi) - \mathcal{A}_\epsilon(u_\epsilon, \varphi)| = |\mathcal{A}_\epsilon(u_\epsilon, \varphi) - \mathcal{A}(u_\epsilon, \varphi)| \\ &= \left| \iint_{(\Omega \cup \Gamma)^2} (u_\epsilon(\mathbf{x}) - u_\epsilon(\mathbf{y}))(\varphi(\mathbf{x}) - \varphi(\mathbf{y}))(\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y}))d\mathbf{y}d\mathbf{x} \right|. \end{aligned}$$

By expanding the product, we have

$$\begin{aligned} &\left| \iint_{(\Omega \cup \Gamma)^2} (u_\epsilon(\mathbf{x}) - u_\epsilon(\mathbf{y}))(\varphi(\mathbf{x}) - \varphi(\mathbf{y}))(\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y}))d\mathbf{y}d\mathbf{x} \right| \\ &\leq \iint_{(\Omega \cup \Gamma)^2} |u_\epsilon(\mathbf{x})\varphi(\mathbf{x})| |\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y})|d\mathbf{y}d\mathbf{x} \\ &\quad + \iint_{(\Omega \cup \Gamma)^2} |u_\epsilon(\mathbf{y})\varphi(\mathbf{y})| |\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y})|d\mathbf{y}d\mathbf{x} \\ &\quad + \iint_{(\Omega \cup \Gamma)^2} |u_\epsilon(\mathbf{x})\varphi(\mathbf{y})| |\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y})|d\mathbf{y}d\mathbf{x} \\ &\quad + \iint_{(\Omega \cup \Gamma)^2} |u_\epsilon(\mathbf{y})\varphi(\mathbf{x})| |\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y})|d\mathbf{y}d\mathbf{x}. \end{aligned}$$

By switching the order of integration and renaming dummy variables in the second and fourth terms above, we obtain

$$\begin{aligned}
& 2 \int_{\Omega \cup \Gamma} |u_\epsilon(\mathbf{x})\varphi(\mathbf{x})| \int_{\Omega \cup \Gamma} |\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y})| d\mathbf{y} d\mathbf{x} \\
& + 2 \int_{\Omega \cup \Gamma} |\varphi(\mathbf{x})| \int_{\Omega \cup \Gamma} |u_\epsilon(\mathbf{y})| |\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y})| d\mathbf{y} d\mathbf{x} \\
& \leq 2c_1(\epsilon) \int_{\Omega \cup \Gamma} |u_\epsilon(\mathbf{x})\varphi(\mathbf{x})| d\mathbf{x} \\
& + 2 \int_{\Omega \cup \Gamma} |\varphi(\mathbf{x})| \|u_\epsilon\|_{L^2(\Omega \cup \Gamma)} \|\gamma_\epsilon - \gamma\|_{L^2(\Omega \cup \Gamma)} d\mathbf{x} \\
& \leq 2c_1(\epsilon) \|u_\epsilon\|_{L^2(\Omega \cup \Gamma)} \|\varphi\|_{L^2(\Omega \cup \Gamma)} + 2c_2(\epsilon) |\Omega \cup \Gamma|^{\frac{1}{2}} \|u_\epsilon\|_{L^2(\Omega \cup \Gamma)} \|\varphi\|_{L^2(\Omega \cup \Gamma)},
\end{aligned} \tag{17}$$

where we used the Cauchy-Schwarz inequality for the outer and inner integral for the first and second term, respectively, and where

$$\begin{aligned}
c_1(\epsilon) &= \max_{\mathbf{x} \in \Omega \cup \Gamma} \int_{B_{\delta+\epsilon}(\mathbf{x}) \cap (\Omega \cup \Gamma)} |\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y})| d\mathbf{y} \leq \int_{B_{\delta+\epsilon}(\mathbf{0})} |\gamma_\epsilon(\mathbf{0}, \mathbf{y}) - \gamma(\mathbf{0}, \mathbf{y})| d\mathbf{y}, \\
c_2^2(\epsilon) &= \max_{\mathbf{x} \in \Omega \cup \Gamma} \int_{B_{\delta+\epsilon}(\mathbf{x}) \cap (\Omega \cup \Gamma)} (\gamma_\epsilon(\mathbf{x}, \mathbf{y}) - \gamma(\mathbf{x}, \mathbf{y}))^2 d\mathbf{y} \leq \int_{B_{\delta+\epsilon}(\mathbf{0})} (\gamma_\epsilon(\mathbf{0}, \mathbf{y}) - \gamma(\mathbf{0}, \mathbf{y}))^2 d\mathbf{y}.
\end{aligned}$$

We recall that, by definition, $\gamma_\epsilon \rightarrow \gamma$ pointwise as $\epsilon \rightarrow 0$; thus, $c_i(\epsilon) \rightarrow 0$ as $\epsilon \rightarrow 0$, for $i = 1, 2$. Furthermore, thanks to the properties of η and $\mu_{\delta, \epsilon}$, the integrals above are well-defined.

To obtain the final estimate, we consider $\varphi = u - u_\epsilon$ and recall that for the kernels considered in this work the energy norm $||| \cdot |||$ is equivalent to the L^2 norm. In particular there exists a positive constant C_{eq} such that $\|\varphi\|_{L^2(\Omega \cup \Gamma)} \leq |||\varphi|||$. Thus, we have the following estimate

$$\begin{aligned}
\|u - u_\epsilon\|_{L^2(\Omega \cup \Gamma)}^2 &\leq C_{eq} |||u - u_\epsilon|||^2 \\
&= C_{eq} \mathcal{A}(u - u_\epsilon, u - u_\epsilon) \\
&\leq C_{eq} k(\epsilon) \|u_\epsilon\|_{L^2(\Omega \cup \Gamma)} \|u - u_\epsilon\|_{L^2(\Omega \cup \Gamma)},
\end{aligned}$$

where $k(\epsilon)$ is obtained from the constants in (17). We finally conclude that

$$\|u - u_\epsilon\|_{L^2(\Omega \cup \Gamma)} \leq C_{eq} k(\epsilon) \|u_\epsilon\|_{L^2(\Omega \cup \Gamma)}, \tag{18}$$

where the constant $k(\epsilon)$ is such that $k(\epsilon) \rightarrow 0$ as $\epsilon \rightarrow 0$.

4 | FINITE ELEMENT FORMULATION

In this section we introduce a FE discretization of problem (10), highlight the associated computational challenges, and describe how the formulation introduced in the previous section helps circumventing them. Let $\mathcal{T}_h$ be a shape-regular triangulation of $\Omega \cup \Gamma$ into N_L finite elements $\{\mathcal{E}_l\}_{l=1}^{N_L}$; the latter $\mathcal{E}_l$ can either be triangles and/or quadrilaterals in two dimensions and tetrahedra and/or hexahedra in three dimensions³⁷. The parameter h represents the size of the triangulation and corresponds to the larger element diameter. Also, let $V_0^{N_h}$ be a finite dimensional subspace of V_0 of dimension N_h , proportional to h^{-1} , and let $\{\varphi_i\}_{i=1}^{N_h}$ be a basis for $V_0^{N_h}$. In this work we consider Lagrange basis functions over the triangulation $\mathcal{T}_h$. Thus, we can write the FE solution w_h of equation (10) as $w_h(\mathbf{x}) = \sum_{i=1}^{N_h} W_i \varphi_i(\mathbf{x})$. By using this expression and $\varphi \in \{\varphi_i\}_{i=1}^{N_h}$, equation (10) reduces to the algebraic system

$$A\mathbf{W} = \mathbf{F}, \tag{19}$$

³⁷See D'Elia et al. [37] for a description of appropriate triangulation techniques for nonlocal problems.

where $\mathbf{W} \in \mathbb{R}^{N_h}$ is the vector whose components are the degrees of freedom of the numerical solution w_h , $\mathbf{F}$ is such that $\mathbf{F}_i = \tilde{F}(\varphi_i)$, and A is the stiffness matrix with entries

$$A_{ij} = \mathcal{A}(\varphi_i, \varphi_j) = \iint_{(\Omega \cup \Gamma)^2} (\varphi_i(\mathbf{x}) - \varphi_i(\mathbf{y})) (\varphi_j(\mathbf{x}) - \varphi_j(\mathbf{y})) \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x}. \quad (20)$$

4.1 | Circumventing computational challenges

The computation of the entries of the stiffness matrix A raises several diverse challenges. In this work we specifically focus on the challenges related to the presence of the indicator function in the definition of the kernel. Other challenges, such as the presence of singularities in fractional-type kernels or peridynamics kernels are not considered here. We point out that our method can be combined with any technique that takes into account the presence of the singularity. In fact, while the singularity is located at the center of the ball, the issues considered in this paper arise at the boundary. To make our description clear, we rewrite (20) by explicitly indicating the domain of integration, i.e.

$$\begin{aligned} A_{ij} &= C_\delta \int_{\Omega \cup \Gamma} \int_{(\Omega \cup \Gamma) \cap B_\delta(\mathbf{x})} (\varphi_i(\mathbf{x}) - \varphi_i(\mathbf{y})) (\varphi_j(\mathbf{x}) - \varphi_j(\mathbf{y})) \eta(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \\ &= C_\delta \sum_{l=1}^{N_L} \sum_{k=1}^{N_L} \int_{\mathcal{E}_l} \int_{\mathcal{E}_k \cap B_\delta(\mathbf{x})} (\varphi_i(\mathbf{x}) - \varphi_i(\mathbf{y})) (\varphi_j(\mathbf{x}) - \varphi_j(\mathbf{y})) \eta(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x}, \end{aligned} \quad (21)$$

where we split the integrals over the elements with the purpose of using composite quadrature rules. In fact, global quadrature rules used, e.g., over the ball $B_\delta(\mathbf{x})$ for the inner integration are not convenient due to the basis functions' bounded support^[37].

It is evident that the first challenge that one has to face is the integration over partial elements: when the element $\mathcal{E}_k$ is not fully contained in the ball, standard quadrature rules such as Gauss quadrature rules defined over $\mathcal{E}_k$ are not suitable due to the presence of the discontinuity induced by the indicator function. Thus, it is necessary to determine the intersection regions $\mathcal{E}_k \cap B_\delta(\mathbf{x})$ and define quadrature rules there. This task, while affordable in two dimensions, becomes extremely complex and impractical in three dimensions. Furthermore, when $B_\delta(\mathbf{x})$ is a Euclidean ball, such regions are curved so that appropriate approximations or quadrature rules for curved domains must be taken into account^[37]. A key observation is that these issues do not arise in case of smooth kernel functions, e.g. functions that do not abruptly jump to zero outside of $B_\delta(\mathbf{x})$, but that approach zero smoothly. This would allow the use of quadrature rules defined over the whole element $\mathcal{E}_k$, circumventing the issue of determining intersections or integrating over curved regions.

The parametrized kernel introduced in Section 3 is such that the transition to zero happens smoothly (as an example, for constant kernel functions η , the kernel function is a piece-wise polynomial in C^4). Thus, the inner integration can be performed over the whole element $\mathcal{E}_k$, using accurate enough quadrature rules, without worrying about the presence of a discontinuity. We then propose to solve the approximate, parametrized problem

$$\mathcal{A}_\epsilon(w_{h,\epsilon}, \varphi_i) = \tilde{F}(\varphi_i), \quad \forall i = 1, \dots, N_h, \quad (22)$$

for which the entries of the stiffness matrix, that, with an abuse of notation, we still denote by A , read

$$A_{ij} = C_{\delta,\epsilon} \sum_{l=1}^{N_L} \sum_{k=1}^{N_L} \int_{\mathcal{E}_l} \int_{\mathcal{E}_k} (\varphi_i(\mathbf{x}) - \varphi_i(\mathbf{y})) (\varphi_j(\mathbf{x}) - \varphi_j(\mathbf{y})) \eta(\mathbf{x}, \mathbf{y}) \mu_{\delta,\epsilon}(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x}. \quad (23)$$

By avoiding the problem of determining intersecting elements, this approach makes three-dimensional implementation a much simpler task.

Remark 1. The convergence of the solution $w_{h,\epsilon}$ to the continuous solution w depends on both the discretization parameter h and the mollifying parameter ϵ . An adaptive quadrature procedure, introduced in the following section, will further contribute to the overall approximation error, as we discuss and illustrate in Section 6.

5 | ADAPTIVE QUADRATURE RULES

As already pointed out, the use of the mollifier, in place of the characteristic function, removes the difficulty of integrating discontinuous functions. The transition region of the mollifier has thickness 2ϵ and it is important to choose quadrature rules that can appropriately capture this region, especially if $\epsilon \ll h$. Note that the presence of the transition region affects the regularity of both the inner and the outer integrands. As we explain below, only one adaptive rule is necessary, applied to either the outer or inner integral. A quadrature rule with few points can be fast but is also inaccurate, one with many points can be accurate but is also expensive, especially in higher dimensions. To this end, adaptive quadrature rules have been proven to be accurate and efficient^[46]. The advantage of using an adaptive scheme is that only the portion of the element overlapping with the transition region needs to be recursively refined, considerably reducing the computational time. Moreover, for fixed ϵ , each partitioning has the effect of halving the ratio h/ϵ . Thus, it is always possible to choose a number of adaptive refinements such that $h \approx \epsilon$ and for which a quadrature rule with few points is accurate enough. Note that the adaptive quadrature rule devised here is not standard, because the refinement criterion is controlled by the distance between points in the outer and inner integrals. The details of the algorithm are given below.

We recall that we denote by A the stiffness matrix corresponding to the parametrized bilinear form $\mathcal{A}_\epsilon$. It is convenient to rewrite its entries as $A_{ij} = A_{ij}^{11} + A_{ij}^{12} + A_{ij}^{21} + A_{ij}^{22}$, where each term is given by

$$A_{ij}^{11} = \int_{\Omega \cup \Gamma} \int_{\Omega \cup \Gamma} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) \varphi_i(\mathbf{x}) \varphi_j(\mathbf{x}) d\mathbf{y} d\mathbf{x}, \quad (24)$$

$$A_{ij}^{12} = - \int_{\Omega \cup \Gamma} \int_{\Omega \cup \Gamma} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) \varphi_i(\mathbf{x}) \varphi_j(\mathbf{y}) d\mathbf{y} d\mathbf{x}, \quad (25)$$

$$A_{ij}^{21} = - \int_{\Omega \cup \Gamma} \int_{\Omega \cup \Gamma} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) \varphi_i(\mathbf{y}) \varphi_j(\mathbf{x}) d\mathbf{y} d\mathbf{x}, \quad (26)$$

$$A_{ij}^{22} = \int_{\Omega \cup \Gamma} \int_{\Omega \cup \Gamma} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) \varphi_i(\mathbf{y}) \varphi_j(\mathbf{y}) d\mathbf{y} d\mathbf{x}. \quad (27)$$

The following proposition allows us to express A_{ij} only as a sum of two of the terms above, as we show in Corollary [1](#)

Proposition 1. Let f_1 and $f_2 \in V$, and let $g(\mathbf{x}, \mathbf{y})$ be a symmetric function, then

$$\int_{\Omega \cup \Gamma} \int_{\Omega \cup \Gamma} g(\mathbf{x}, \mathbf{y}) f_1(\mathbf{x}) f_2(\mathbf{y}) d\mathbf{y} d\mathbf{x} = \int_{\Omega \cup \Gamma} \int_{\Omega \cup \Gamma} g(\mathbf{x}, \mathbf{y}) f_1(\mathbf{y}) f_2(\mathbf{x}) d\mathbf{y} d\mathbf{x}. \quad (28)$$

Proof.

$$\begin{aligned} & \int_{\Omega \cup \Gamma} \int_{\Omega \cup \Gamma} g(\mathbf{x}, \mathbf{y}) f_1(\mathbf{x}) f_2(\mathbf{y}) d\mathbf{y} d\mathbf{x} \\ &= \int_{\Omega \cup \Gamma} \int_{\Omega \cup \Gamma} g(\mathbf{x}, \mathbf{y}) f_1(\mathbf{x}) f_2(\mathbf{y}) d\mathbf{x} d\mathbf{y} && \text{reversing the order} \\ &= \int_{\Omega \cup \Gamma} \int_{\Omega \cup \Gamma} g(\mathbf{x}, \mathbf{y}) f_1(\mathbf{y}) f_2(\mathbf{x}) d\mathbf{y} d\mathbf{x} && \begin{array}{l} \text{renaming variables} \\ \text{and using the symmetry of } g \end{array} \end{aligned}$$

□

Corollary 1. The entries of the stiffness matrix A satisfy the following equality

$$A_{ij} = 2A_{ij}^{11} + 2A_{ij}^{12} = 2A_{ij}^{21} + 2A_{ij}^{22}$$

.

Proof. The proof follows from the definition of A_{ij} and Proposition [1](#). Namely, $A_{ij}^{11} = A_{ij}^{22}$ and $A_{ij}^{12} = A_{ij}^{21}$. □

Let $\mathcal{Q}_1 = \{(\mathbf{x}_{q_1}, w_{q_1})\}_{q_1}$ and $\mathcal{Q}_2 = \{(\mathbf{x}_{q_2}, w_{q_2})\}_{q_2}$ denote sets of quadrature points and associated weights representing two different *composite quadrature* rules for the numerical integration over the region $\Omega \cup \Gamma$. To preserve the equality between A_{ij}^{11} and A_{ij}^{22} , and between A_{ij}^{12} and A_{ij}^{21} we use $\mathcal{Q}_1$ and $\mathcal{Q}_2$ to numerically evaluate the inner and outer integrals as follows:

$$A_{ij}^{11} = \sum_{q_1 \in \mathcal{Q}_1} \sum_{q_2 \in \mathcal{Q}_2} \gamma_\epsilon(\mathbf{x}_{q_1}, \mathbf{x}_{q_2}) \varphi_i(\mathbf{x}_{q_1}) \varphi_j(\mathbf{x}_{q_1}) w_{q_2} w_{q_1}, \quad (29)$$

$$A_{ij}^{12} = \sum_{q_1 \in \mathcal{Q}_1} \sum_{q_2 \in \mathcal{Q}_2} \gamma_\epsilon(\mathbf{x}_{q_1}, \mathbf{x}_{q_2}) \varphi_i(\mathbf{x}_{q_1}) \varphi_j(\mathbf{x}_{q_2}) w_{q_2} w_{q_1}, \quad (30)$$

$$A_{ij}^{21} = \sum_{q_2 \in \mathcal{Q}_2} \sum_{q_1 \in \mathcal{Q}_1} \gamma_\epsilon(\mathbf{x}_{q_2}, \mathbf{x}_{q_1}) \varphi_i(\mathbf{x}_{q_1}) \varphi_j(\mathbf{x}_{q_2}) w_{q_1} w_{q_2}, \quad (31)$$

$$A_{ij}^{22} = \sum_{q_2 \in \mathcal{Q}_2} \sum_{q_1 \in \mathcal{Q}_1} \gamma_\epsilon(\mathbf{x}_{q_2}, \mathbf{x}_{q_1}) \varphi_i(\mathbf{x}_{q_1}) \varphi_j(\mathbf{x}_{q_1}) w_{q_1} w_{q_2}. \quad (32)$$

According to Corollary 1, only two terms among the ones above need to be computed. We choose to compute A_{ij}^{21} and A_{ij}^{22} in (31) and (32). We adopt an adaptive scheme for the outer quadrature and Gaussian composite quadrature rules for the inner integration over the elements that intersect the ball. These choices are empirical, i.e. they have been guided by numerical experiments that showed that for A_{ij}^{21} and A_{ij}^{22} using an adaptive quadrature for the outer integral is more efficient than using it for the inner one.

For every $\mathcal{E}_l \in \mathcal{T}_h$ we define

$$\mathcal{K}_l = \{m \in \{1, \dots, N_L\} : \|\mathbf{x} - \mathbf{y}\|_{\ell_\infty} \geq \delta + \epsilon, \forall \mathbf{x} \in \mathcal{E}_l, \forall \mathbf{y} \in \mathcal{E}_m\}, \quad (33)$$

and $\mathcal{J}_l = \mathcal{K}_l^c$, i.e. the complement of $\mathcal{K}_l$ in $\{1, \dots, N_L\}$. For $k = 1, 2$ and $l = 1, \dots, N_L$ let $\mathcal{Q}_k^l = \{(\mathbf{x}_{q_{k_l}}, w_{q_{k_l}})\}_{q_{k_l}} \subset \mathcal{Q}_k$ denote the subset of $\mathcal{Q}_k$ composed of those quadrature points and weights obtained by only considering the quadrature points (and associated weights) that lie within element $\mathcal{E}_l$. Then, integrals (31) and (32) can be rewritten as

$$A_{ij}^{21} = \sum_{l=1}^{N_L} \sum_{q_2 \in \mathcal{Q}_2^l} \sum_{m \in \mathcal{J}_l} \sum_{q_1 \in \mathcal{Q}_1^m} \gamma_\epsilon(\mathbf{x}_{q_2}, \mathbf{x}_{q_1}) \varphi_i^m(\mathbf{x}_{q_1}) \varphi_j^l(\mathbf{x}_{q_2}) w_{q_1} w_{q_2}, \quad (34)$$

$$A_{ij}^{22} = \sum_{l=1}^{N_L} \sum_{q_2 \in \mathcal{Q}_2^l} \sum_{m \in \mathcal{J}_l} \sum_{q_1 \in \mathcal{Q}_1^m} \gamma_\epsilon(\mathbf{x}_{q_2}, \mathbf{x}_{q_1}) \varphi_i^m(\mathbf{x}_{q_1}) \varphi_j^l(\mathbf{x}_{q_1}) w_{q_1} w_{q_2}. \quad (35)$$

Note that in the equations above the terms in the sum are nonzero only when the support of the basis functions intersects the elements. An adaptive quadrature with midpoint refinement is adopted for $\mathcal{Q}_2^l$. The pseudo-code that describes the adaptivity algorithm is reported in Algorithm 1: we employ recursive calls with input arguments L_{min} , L_{max} , L_{cur} , $\mathcal{E}_l$, and $\mathcal{J}_l$. In each call L_{min} and L_{max} are fixed parameters, and represent the minimum and the maximum level of refinement, with $L_{max} \geq L_{min} \geq 1$. L_{cur} is the current level of refinement. In the initial call $\mathcal{E}_l$ and $\mathcal{J}_l$ are the ones defined above and $L_{cur} = 1$, while in the recursive calls these three arguments are subject to changes as described below.

- If $L_{cur} < L_{min}$, then $\mathcal{E}_l$ is split in 2^N sub-elements $\mathcal{E}_{l_i}$ using the midpoint rule and for each of them the adaptive integration function is called again increasing L_{cur} by one and using the same index set $\mathcal{J}_l$.
- If $L_{cur} = L_{max}$ integration is performed on $\mathcal{E}_l$ for the outer integral and on each element indexed by $\mathcal{J}_l$ for the inner integral, without any further refinement for $\mathcal{E}_l$. The numerical integration is performed using standard Gauss Legendre quadrature rules both for the outer and inner integrals.
- If $L_{min} \leq L_{cur} < L_{max}$, from the index set $\mathcal{J}_l$ two new index sets $\mathcal{J}_l^{int}$ and $\mathcal{J}_l^{ref}$ are extracted, for which $\mathcal{E}_l$ is either integrated or further refined. For any m in $\mathcal{J}_l$, the maximum distance from $\mathcal{E}_m$ to $\mathcal{E}_l$ is computed. If this distance is less than $\delta - \epsilon$, then m is added to $\mathcal{J}_l^{int}$, otherwise the minimum distance from $\mathcal{E}_m$ to $\mathcal{E}_l$ is computed. If this distance is less than $\delta + \epsilon$ then m is added to the index set $\mathcal{J}_l^{ref}$. If $\mathcal{J}_l^{int}$ is non-empty, integration is performed on $\mathcal{E}_l$ for the outer integral and on each element indexed by $\mathcal{J}_l^{int}$ for the inner integral. If $\mathcal{J}_l^{ref}$ is non-empty, then $\mathcal{E}_l$ is split in 2^N sub-elements $\mathcal{E}_{l_i}$ and for each of them the adaptive integration function is called again increasing L_{cur} by one and using $\mathcal{J}_l^{ref}$ as index set.

Algorithm 1 Pseudo-code for the adaptive refinement integration.

```

function ADAPTIVE INTEGRATION( $L_{min}, L_{max}, L_{cur}, \mathcal{E}_l, J_l$ )
  if  $L_{cur} < L_{min}$  then
    split  $\mathcal{E}_l$  into  $2^N$  new sub-elements  $\mathcal{E}_{l_i}$ 
    for  $i = 1, \dots, 2^N$  do
      ADAPTIVE INTEGRATION( $L_{min}, L_{max}, L_{cur} + 1, \mathcal{E}_{l_i}, J_l$ )
    end for
  else if  $L_{cur} = L_{max}$  then
    INTEGRATION( $\mathcal{E}_l, J_l$ )
  else
     $J_l^{int} = \emptyset$ 
     $J_l^{ref} = \emptyset$ 
    for all  $m \in J_l$  do
      if  $\max \text{dist}(\mathcal{E}_l, \mathcal{E}_m) < \delta - \epsilon$  then
        add  $m$  to the index set  $J_l^{int}$ 
      else if  $\min \text{dist}(\mathcal{E}_l, \mathcal{E}_m) < \delta + \epsilon$  then
        add  $m$  to the index set  $J_l^{ref}$ 
      end if
    end for
    if  $J_l^{int} \neq \emptyset$  then
      INTEGRATION( $\mathcal{E}_l, J_l^{int}$ )
    end if
    if  $J_l^{ref} \neq \emptyset$  then
      split  $\mathcal{E}_l$  into  $2^N$  new sub-elements  $\mathcal{E}_{l_i}$ 
      for  $i = 1, \dots, 2^N$  do
        ADAPTIVE INTEGRATION( $L_{min}, L_{max}, L_{cur} + 1, \mathcal{E}_{l_i}, J_l^{ref}$ )
      end for
    end if
  end if
end function

```

5.1 | Approximate maximum and minimum distances between elements

Evaluating the exact distances between two elements can be computationally expensive, especially for unstructured three-dimensional meshes. Hence, in practice, we use conservative distances that are simple to compute in place of the maximum and the minimum. Namely, we first loop over the nodes of each element to find the minimum and maximum coordinates in each dimension, denote them by $\mathbf{x}_{\min}$, $\mathbf{x}_{\max}$ and by $\mathbf{y}_{\min}$, $\mathbf{y}_{\max}$. Here $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ are the vectors containing the minimum and maximum coordinates of the bounding box containing $\mathcal{E}_l$. Similarly, $\mathbf{y}_{\min}$ and $\mathbf{y}_{\max}$ are the vectors containing the minimum and maximum coordinates of the bounding box containing $\mathcal{E}_m$. For each dimension $k = 1, \dots, N$ evaluate the two quantities $d_1^k = x_{\min}^k - y_{\max}^k$ and $d_2^k = y_{\min}^k - x_{\max}^k$. Finally, we approximate the maximum and minimum distances with the two quantities

$$\text{aprx max dist}(\mathcal{E}_l, \mathcal{E}_m) = \sqrt{\sum_{k=1, \dots, N} \max(d_1^{k^2}, d_2^{k^2})}, \quad (36)$$

$$\text{aprx min dist}(\mathcal{E}_l, \mathcal{E}_m) = \max_{k=1, \dots, N} (\max(0, d_1^k, d_2^k)). \quad (37)$$

The approximate maximum distance in (36) is the maximum among the distances between opposite vertices of the two bounding boxes. For example in two dimensions it would be one among the distances $\|SW - NE\|_{\ell_2}$, $\|SE - NW\|_{\ell_2}$, $\|NE - SW\|_{\ell_2}$ and $\|NW - SE\|_{\ell_2}$, with S meaning South, N meaning North and so on. This is true regardless of the reciprocal position of the two boxes. To better understand (37), consider the projections of the bounding boxes in the direction of k . Recall that, for fixed k , at least one between d_1^k and d_2^k is always negative. The other is positive only if the 2 projections do not overlap. In this

case, $\max(0, d_1^k, d_2^k)$ is the minimum (positive) distance between the 2 non-overlapping projections. Finally, we approximate the minimum distance with the largest projected distance.

It is easy to see that

$$\text{aprx max dist}(\mathcal{E}_l, \mathcal{E}_m) \geq \max \text{dist}(\mathcal{E}_l, \mathcal{E}_m), \quad (38)$$

$$\text{aprx min dist}(\mathcal{E}_l, \mathcal{E}_m) \leq \min \text{dist}(\mathcal{E}_l, \mathcal{E}_m), \quad (39)$$

where the sign of the inequalities assures the conservative approach in the adaptive integration algorithm. For a kernel whose support is identified by a ball in the topology defined by the L^2 norm, the minimum distance could be also approximated by

$$\text{aprx min dist}(\mathcal{E}_l, \mathcal{E}_m) = \sqrt{\sum_{k=1, \dots, N} \max(0, d_1^k, d_2^k)^2}.$$

This last would give a sharper inequality in (39), however it would not work for the case where the support of the kernel is an L^∞ ball. In this work, for generality, we have chosen to always use formula (37). Note that (37) is also used in (33) in place of $\|\cdot\|_{l_\infty}$ to identify the elements indexed by $\mathcal{J}_l$.

6 | NUMERICAL RESULTS

In this section, we present the results of numerical tests for FE discretizations of two-dimensional ($n = 2$) and three-dimensional ($n = 3$) problems. These results allow us to illustrate the theoretical results presented in the previous sections and highlight the efficiency of our approach.

We first show the consistency of the proposed method; specifically, fixing the mesh and letting the maximum level of adaptive refinement L_{max} increase, we study the behavior of the discretization error with respect to an analytic solution that belongs to the FE space. Then, we investigate the convergence of discretized solutions to the continuous one as the mesh is refined. To better understand the behavior of the algorithm and the specific sources of error, we devised a specific numerical test to isolate the error induced by the presence of the mollifier and analyze the convergence behavior with respect to ε . Accuracy comparisons with the algorithm proposed in the paper by D'Elia et al. [37] are also provided.

Due to the intrinsically high computational costs of nonlocal simulations, a parallel implementation of the algorithm is proposed and its scalability properties are analyzed both in two and three dimensions.

The two-dimensional tests are carried out on quadrilateral, triangular, and mixed meshes, i.e. meshes consisting of both quadrilateral and triangular elements. The three-dimensional simulations are carried out on a hexahedral mesh.

We consider constant kernels supported on Euclidean balls of radius δ . To guarantee the consistency of the nonlocal diffusion operator with the classical Laplacian for polynomials up to degree three and its convergence to the classical Laplacian as $\delta \rightarrow 0$, we select the constants C_δ and $C_{\delta, \varepsilon}$ in (11) and (12), respectively, as follows

$$\begin{aligned} n = 2 : \quad C_\delta &= \frac{4\kappa}{\pi\delta^4}, \quad C_{\delta, \varepsilon} = \frac{C_\delta}{1 + \frac{6}{11} \left(\frac{\varepsilon}{\delta}\right)^2 + \frac{3}{143} \left(\frac{\varepsilon}{\delta}\right)^4}, \\ n = 3 : \quad C_\delta &= \frac{15\kappa}{4\pi\delta^5}, \quad C_{\delta, \varepsilon} = \frac{C_\delta}{1 + \frac{10}{11} \left(\frac{\varepsilon}{\delta}\right)^2 + \frac{15}{143} \left(\frac{\varepsilon}{\delta}\right)^4}. \end{aligned}$$

For sufficiently smooth u , these choices guarantee quadratic convergence of the nonlocal Laplacian to the local one for $\delta \rightarrow 0$:

$$\mathcal{L}u(\mathbf{x}) = \Delta u(\mathbf{x}) + O(\delta^2) \quad \text{and} \quad \mathcal{L}_\varepsilon u(\mathbf{x}) = \Delta u(\mathbf{x}) + O(\delta^2).$$

Sources of numerical error

Let u_h be the FE solution; the numerical error can heuristically be split into three separate contributions, i.e.

$$\|u - u_h\|_{L^2(\Omega \cup \Gamma)} \leq C_1 h^{p_1} + C_2 \varepsilon^{p_2} + E_i(\varepsilon, h, L_{max}), \quad (40)$$

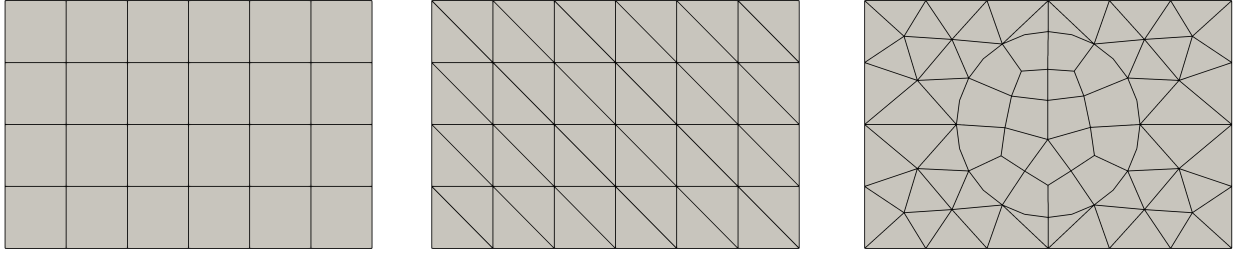


FIGURE 2 Meshes used for the two-dimensional numerical simulations: quadrilateral (left, number of degrees of freedom with linear FE discretization $N_h = 35$, and $N_h = 93$ for quadratic FE discretization), triangular (center, $N_h = 35$ for linear FE, $N_h = 117$ for quadratic FE) and mixed (right, $N_h = 58$ for linear FE, $N_h = 190$ for quadratic FE).

where, h is the characteristic size of the mesh, C_1 and C_2 are positive constants independent of h , ε and L_{max} , and p_1 and p_2 are positive integers that represent the rates of convergence with respect to h and ε , respectively. The first term on the right-hand side is the interpolation error and depends on the FE family used to discretize the problem^[4]. The second term is the error induced by the presence of the mollifier and, in the L^2 metric, it is bounded by the expression in (18). The last term, E_i , is the numerical integration error, that for fixed external and internal Gauss quadrature rules depends on ε , h and L_{max} .

For $h \rightarrow 0$ or $L_{max} \rightarrow \infty$ the integration error E_i vanishes, whereas for $\varepsilon \rightarrow 0$ it increases, since the transition to zero of the mollifier features higher gradients. In all simulations we empirically set the integration parameters so that the integration error E_i could be negligible compared to $C_1 h^{p_1} + C_2 \varepsilon^{p_2}$. In such a context, ε cannot be selected independently of h or L_{max} ; explicit dependence is provided in each simulation.

6.1 | Two-dimensional tests

Two-dimensional numerical simulations of nonlocal operators can be found in several works in literature, see, e.g., the papers by Macek^[49], Wang^[50], and Vollman^[51]. However, such studies are often designed for structured mesh only. On the other hand, our method can be applied to any type of mesh. We consider the domain $\Omega = [-0.6, 0.6] \times [-0.4, 0.4]$ and three different meshes, see Figure 2 for a coarse example. Linear and quadratic Lagrange FE spaces are considered. Unless otherwise stated, we use Gauss-Legendre 3×3 product rule for quadrilateral elements and Dunavant 7-point rule for triangles. In all tests we consider the error with respect to an analytic, manufactured solution, $u(\mathbf{x})$. For all $\mathbf{x} \in \Gamma$, the nonlocal Dirichlet volume constraint is set to $g(\mathbf{x}) = u(\mathbf{x})$ and the forcing term is known analytically as $f(\mathbf{x}) = -\mathcal{L}u(\mathbf{x})$.

Consistency

We choose $u \in V^{N_h}$, so that the interpolation error contribution in (40) is identically zero, regardless of h . We compute the discretization error $\|u - u_h\|_{L^2}$ for a fixed mesh while increasing the values of the maximum level of adaptivity L_{max} .

For the linear manufactured solution $u = 1 + x + y$ the forcing term is $f(\mathbf{x}) = 0$. In this case the numerical errors are always zero for both the linear and the quadratic FE spaces, for any L_{max} and ε . Although welcome, this result is an over-achievement, as it is obtained only because the forcing terms are zero. Thus, it should not be taken as a reference. For the quadratic manufactured solution $u = x^2 + y^2$ the forcing term is $f(\mathbf{x}) = -2$. In this case we consider only the quadratic FE space, because it is the only one that can reproduce exactly the solution. We choose ε as

$$\varepsilon = \varepsilon_0 \left(\frac{3}{4} \right)^{L_{max}-3}$$

so that $\varepsilon \rightarrow 0$ for $L_{max} \rightarrow \infty$, i.e. for $L_{max} \rightarrow \infty$ both the mollifier and the integration errors in (40) vanish.

In Table 1 we report the errors of the numerical tests for the quadrilateral (QUAD) and triangular (TRI) meshes after one refinement, together with all the values of the parameters h , δ , ε_0 and L_{min} . The L^2 -norm of the error decreases down to machine precision, when L_{max} is increased, for both quadrilateral and triangular meshes, illustrating the consistency of the implemented adaptivity method in two dimensions.

^[4]For convergence rates of FE discretizations in presence of integrable kernels, we refer the reader to Du et al.^[45]

TABLE 1 Consistency test: errors $\|u - u_h\|_{L^2}$ as L_{max} increases for $u = x^2 + y^2$, $h = 0.1$, $\delta = 0.2$, $\varepsilon_0 = 0.0125$, $L_{min} = 1$, $\varepsilon = \varepsilon_0(3/4)^{L_{max}-3}$ and quadratic FE.

L_{max}	QUAD	TRI
3	8.123E-05	2.713E-05
4	1.278E-05	2.052E-06
5	3.453E-07	1.193E-07
6	1.793E-09	1.774E-09
7	7.073E-10	5.479E-11

h -Convergence

We consider the convergence with the respect to the grid size h on quadrilateral, triangular and mixed meshes. We use the manufactured solution $u(\mathbf{x}) = x^3 + y^3$ for which the corresponding source term is given by $f(\mathbf{x}) = -\mathcal{L}u(\mathbf{x}) = -\Delta u(\mathbf{x}) = -6(x + y)$ for $\mathbf{x} \in \Omega$. We analyze the convergence of the finite FE adaptive nonlocal solution u_h to the analytic solution u as we progressively halve the mesh size h by operating on a parameter referred to as ml , as shown in (41). We consider fixed L_{max} , L_{min} , and δ , whereas h and ε depend on ml (mesh level) as follows

$$h = h_0 \left(\frac{1}{2}\right)^{ml-2} \quad \text{and} \quad \varepsilon = \varepsilon_0 \left(\frac{2}{3}\right)^{ml-2}. \quad (41)$$

In Table 2 we report the values of $\|u - u_h\|_{L^2}$ and the corresponding rate of convergence p as ml grows, evaluated in full awareness with respect to h only, with the approximate formula

$$p \cong \ln(E(h)/E(h/2))/\ln(2). \quad (42)$$

For linear FE discretization we obtain quadratic convergence. This is optimal since it resembles the optimal convergence rate $p_1 = 2$ of the interpolation error^[45]. Namely, it indicates that in (40) the mollifier and the integration errors are negligible with respect to the interpolation error. Instead, in case of quadratic FE discretization, the observed rate is $p \approx 3$ (similar to the optimal one $p_1 = 3$) only for $ml \leq 4$, but it deteriorates for higher values of ml , i.e. as we refine the meshes (this behavior happens consistently on all the tested meshes, quadrilateral, triangular and mixed). This is due to the combined effect of the mollifier and integration errors that start dominating for $h \rightarrow 0$. Since $\varepsilon \propto (2/3)^{ml-2}$, the mollifying function defined in (13) exhibits a sharper gradient as we increase ml inducing a less accurate numerical integration, and, hence, higher values of E_i . Moreover, since the mesh refinement significantly reduces the interpolation error $C_1 h^{p_1}$, as we increase ml the error contribution $C_2 \varepsilon^{p_2}$ becomes dominant affecting the overall convergence rate p .

We further test the convergence rate with respect to h considering the fourth-degree polynomial

$$u(\mathbf{x}) = x^4 + y^4, \quad (43)$$

for which the corresponding source term is given by $f(\mathbf{x}) = -12(x^2 + y^2) - \delta^2$ for $\mathbf{x} \in \Omega$. Similarly to the previous test, in Table 3 we report the numerical results for linear and quadratic FE discretizations, on quadrilateral, triangular and mixed meshes. Again, $p \approx 2$ for all the linear discretizations, while $p \approx 3$ for quadratic discretizations only for $ml \leq 3$. Same considerations as for the previous test can be inferred.

ε -Convergence

We analyze the contribution of the mollifier to the discretization error, i.e. $C_2 \varepsilon^{p_2}$ in (40). We consider the fourth-order polynomial in (43) on a triangular mesh with a quadratic FE discretization. In Table 4, for fixed δ and L_{min} , we report on the error $\|u - u_h\|_{L^2}$ and the computed convergence rate p_2 , starting from $\varepsilon = 0.1$ and progressively halving it. To minimize the interpolation error $C_i h^{p_i}$ and the integration error E_i , so that the overall error is dominated by the ε -contribution, for each ε , during the tests the number of adaptive refinements L_{max} and of the mesh level ml have been increased with a brute force procedure until the values

TABLE 2 h -Convergence test: errors $\|u - u_h\|_{L^2}$ and computed order p as the mesh level ml increases for $u = x^3 + y^3$, $L_{min} = 1$, $L_{max} = 3$, $\delta = 0.2$, $h_0 = 0.1$, $h = h_0(1/2)^{ml-2}$, $\varepsilon_0 = 0.0125$, $\varepsilon = \varepsilon_0(2/3)^{ml-2}$.

	QUAD		TRI		MIXED	
ml	bilinear	quadratic	linear	quadratic	linear	quadratic
2	4.363E-03 1.996	6.077E-05 3.090	4.373E-03 1.999	6.389E-05 3.015	2.386E-03 2.028	3.352E-05 3.188
3	1.094E-03 1.998	7.135E-06 2.995	1.094E-03 1.999	7.906E-06 2.989	5.848E-04 2.015	3.677E-06 2.865
4	2.738E-04 2.000	8.950E-07 2.690	2.737E-04 1.999	9.956E-07 2.694	1.447E-04 2.006	5.046E-07 1.674
5	6.845E-05 2.000	1.387E-07 1.012	6.845E-05 2.000	1.538E-07 1.315	3.602E-05 2.003	1.581E-07 0.880
6	1.711E-05	6.878E-08	1.711E-05	6.179E-08	8.983E-06	8.589E-08

TABLE 3 h -Convergence test: errors $\|u - u_h\|_{L^2}$ and computed order p as the mesh level ml increases for $u = x^4 + y^4$, $L_{min} = 1$, $L_{max} = 3$, $\delta = 0.2$, $h_0 = 0.1$, $h = h_0(1/2)^{ml-2}$, $\varepsilon_0 = 0.0125$, $\varepsilon = \varepsilon_0(2/3)^{ml-2}$.

	QUAD		TRI		MIXED	
ml	bilinear	quadratic	linear	quadratic	linear	quadratic
2	6.383E-03 1.972	9.544E-05 2.892	6.398E-03 1.976	1.027E-04 2.914	3.267E-03 2.005	5.561E-05 2.769
3	1.627E-03 1.986	1.285E-05 2.161	1.626E-03 1.985	1.362E-05 2.240	8.138E-04 2.000	8.155E-06 1.677
4	4.106E-04 1.990	2.872E-06 1.374	4.105E-04 1.990	2.883E-06 1.373	2.034E-04 1.994	2.550E-06 1.214
5	1.033E-04 1.990	1.108E-06 1.178	1.033E-04 1.990	1.113E-06 1.176	5.106E-05 1.988	1.099E-06 1.159
6	2.599E-05	4.895E-07	2.599E-05	4.923E-07	1.287E-05	4.920E-07

of the overall error did not change significantly anymore. The error values reported in the table are those obtained only after this steady state was reached. The interpolation error (IE) in the table is reported as a reference and is obtained by numerically solving the local counterpart of the nonlocal Poisson problem at the finer mesh level, i.e. the one to which is associated the steady state. Also as a reference, we report the relative error (RE) between the interpolation error and the overall error

$$RE = \frac{IE}{\|u - u_h\|_{L^2(\Omega \cup \Gamma)}}.$$

The smaller the value of RE the more accurate the data, since the impact of the interpolation error on the overall error vanishes. The ε -convergence order is deliberately computed as

$$p_2 \approx \ln(E(\varepsilon)/E(\varepsilon/2))/\ln(2), \quad (44)$$

and is approximately $p_2 \approx 2$ for all the tested mollifier thicknesses.

Comparison with other algorithms

In Section 5.1 we introduced the maximum and minimum distance between elements in order to determine all elements intersecting the ball that identifies the support of the kernel function, for which integration is performed. However, in the literature, there are other techniques to determine whether an element should be considered during the assembly of the FE matrix. As an

TABLE 4 ε -Convergence test: $\|u - u_h\|_{L^2}$ and computed order p_2 as ε decreases for $u = x^4 + y^4$, $\delta = 0.2$, $L_{min} = 1$, and quadratic FE.

ε	IE	TRI	RE
0.1	1.602E-06	7.107E-04 2.09	2.25E-03
0.05	1.602E-06	1.671E-04 1.86	9.58E-03
0.025	2.002E-07	4.603E-05 1.95	4.35E-03
0.0125	2.002E-07	1.189E-05 1.95	1.68E-02
0.00625	2.503E-08	3.086E-06 1.98	8.10E-03
0.003125	2.503E-08	7.835E-07	3.19E-02

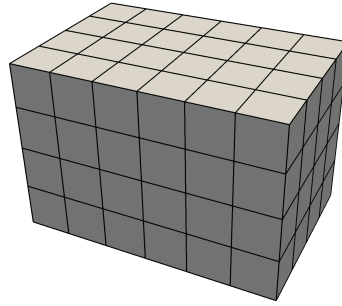


FIGURE 3 Hexahedron mesh used for the three-dimensional numerical simulations, with $N_h = 175$ (linear FE discretization) and $N_h = 605$ (quadratic FE discretization).

example, the paper by D’Elia et al. [\[37\]](#) proposes a technique for which only the elements whose barycenter lies inside the ball are considered for integration. This approach results in an approximation of the ball by a union of whole finite elements, and shows second order h -convergence for all FE spaces (linear, quadratic, etc.), as the rate is determined by the ball approximation. With the purpose of testing the performance of our method against current approaches, we conduct a comparison study, where we consider two different values of ε and compare our results with those obtained with a barycenter approach. Specifically, we use $\varepsilon_0 = 0.0125$, with $\varepsilon = \varepsilon_0(2/3)^{ml-2}$ and $\varepsilon_0 = 0.0250$, with $\varepsilon = \varepsilon_0(1/2)^{ml-2}$ (note that the ε decrease differently). For the the adaptive quadrature rule presented in this paper we use a Legendre quadrature rule both for the internal and external integral, whereas for the barycenter method we use a hybrid Lobatto $\times$ Legendre quadrature rule, Lobatto in the outer integral and Legendre in the inner one. In Table [5](#) we report the discretization errors for linear FE on a quadrilateral mesh as the mesh level ml increases. The h -convergence rate p is computed as in [\(42\)](#). We see that this rate is always optimal ($p \approx 2$) regardless of the value of ε ; in other words, the mollifier and numerical quadrature contributions are negligible and the error is dominated by the FE interpolation contribution. On the other hand, the barycenter method shows bigger errors and less regular convergence order, due to geometric error introduced by the ball approximation. The computational times are all comparable since L_{max} has been taken equal to 1, and all quadrature rules have the same number of points. The interpolation error IE reported in the table has the same meaning as already discussed.

6.2 | Three-dimensional tests

TABLE 5 Comparison test: errors $\|u - u_h\|_{L^2}$ and h -convergence rate p with a quadrilateral mesh and linear FE as the mesh refinement level ml increases for $u = x^4 + y^4$, $\delta = 0.2$, $h_0 = 0.1$, $h = h_0(1/2)^{ml-2}$. For the adaptive algorithm: (Legendre 3×3) $\times$ (Legendre 3×3), $L_{min} = 1$, $L_{max} = 3$. For the barycenter algorithm: (Lobatto 3×3) $\times$ (Legendre 3×3)

	IE	Adaptive		Barycenter
ml	-	$\varepsilon = 0.0125 (2/3)^{ml-2}$	$\varepsilon = 0.025 (1/2)^{ml-2}$	
2	3.280E-03 1.99	3.281E-03 2.02	3.290E-03 2.01	5.428E-03 1.95
3	8.250E-04 2.00	8.092E-04 1.99	8.147E-04 2.00	1.401E-03 1.82
4	2.066E-04 2.00	2.034E-04 1.99	2.038E-04 2.00	3.974E-04 2.00
5	5.168E-05 2.01	5.118E-05 1.99	5.104E-05 2.00	9.909E-05 1.69
6	1.287E-05 1.99	1.287E-05 1.98	1.273E-05 2.00	3.073E-05 1.67
7	3.231E-06	3.270E-06	3.195E-06	9.645E-06

TABLE 6 Consistency in 3D: $h = 0.1$, $\delta = 0.2$, $\varepsilon_0 = 0.0125$, $u = x^2 + y^2 + z^2$, $L_{min} = 1$, $\varepsilon = \varepsilon_0 \left(\frac{3}{4}\right)^{L_{max}-2}$, quadratic FE.

$\ u - u_h\ _{L^2}$	
L_{max}	HEX
2	6.863E-04
3	2.499E-05
4	3.241E-06
5	7.035E-08

Three-dimensional simulations of nonlocal problems are incredibly challenging, especially in a variational setting, due to the prohibitively high computational effort. At the time of this study, the nonlocal literature still does not offer efficient, scalable algorithms for three-dimensional FE implementations of nonlocal solvers for kernels with bounded support. Efficient algorithms for fractional operators are proposed in the work by Glusa^[52], whereas implementations for compactly-supported, integrable kernels on structured grids can be found in the works by Wang^[50] and Vollmann^[51]. Here, not only do we introduce an efficient three-dimensional implementation, but we also propose a scalable, parallel implementation.

We proceed as in the two-dimensional case. In all our tests we consider the numerical domain $\Omega = [-0.6, 0.6] \times [-0.4, 0.4] \times [-0.4, 0.4]$, discretized with the regular hexahedron mesh reported in Figure 3.

We first consider a consistency test; for fixed values of δ , L_{min} and h we increase the refinement level L_{max} . For the same reasons explained in the previous section, we consider the analytic solution $u = x^2 + y^2 + z^2$ for which the corresponding forcing term is given by $f(\mathbf{x}) = -6$ for $\mathbf{x} \in \Omega$. Results for quadratic FE on the structured hexahedral mesh are reported in Table 6. As expected, the error decreases when the adaptive refinement level L_{max} is increased, illustrating the consistency of the adaptive algorithm in three dimensions.

Next, we test the h -convergence on the structured hexahedral mesh for both linear and quadratic FE. For the analytic solution $u(\mathbf{x}) = x^3 + y^3 + z^3$ and fixed L_{min} , L_{max} , and δ we study the behavior of the error $\|u - u_h\|_{L^2}$ as we halve h . Here, the corresponding source term is $f(\mathbf{x}) = -\Delta u(\mathbf{x}) = -\mathcal{L}u = -6(x + y + z)$ for $\mathbf{x} \in \Omega$. In Table 7 (left), we report on the errors and the computed convergence rate p . We also consider the analytic solution $u(\mathbf{x}) = x^4 + y^4 + z^4$ with the corresponding source term $f(\mathbf{x}) = -12(x^2 + y^2 + z^2) - 6/7\delta^2$ for $\mathbf{x} \in \Omega$. For the same parameters as for the cubic polynomial, results of numerical experiments are reported in Table 7 (right). We observe an optimal convergence rate $p \approx 2$ for linear FE discretizations for both the cubic and the quartic cases. Similarly to the two dimensional case, the convergence rate for the quadratic FE discretizations

TABLE 7 h -convergence: errors $\|u - u_h\|_{L^2}$ and convergence rate p as the mesh level ml increases for $u = x^3 + y^3 + z^3$ (left) and $u = x^4 + y^4 + z^4$ (right), $L_{min} = 1$, $L_{max} = 2$, $\delta = 0.2$, $h_0 = 0.2$, $h = h_0(1/2)^{ml-1}$, $\varepsilon_0 = 0.01875$, $\varepsilon = \varepsilon_0(2/3)^{ml-1}$.

	HEX			HEX	
ml	linear	quadratic	ml	linear	quadratic
1	2.220E-02	6.739E-04	1	3.299E-02	9.328E-04
	1.988	2.972		1.964	2.864
2	5.593E-03	8.587E-05	2	8.452E-03	1.281E-04
	1.998	2.797		1.985	2.674
3	1.400E-03	1.235E-05	3	2.134E-03	2.006E-05
	1.999	2.816		1.992	2.303
4	3.501E-04	1.753E-06	4	5.361E-04	4.064E-06

starts from $p \approx 3$ and it deteriorates when ml grows. In general, the three-dimensional numerical results are consistent with the two-dimensional ones and the same considerations can be inferred.

6.3 | Parallel implementation

The high computational effort necessary to perform three-dimensional nonlocal simulations requires a parallel implementation, even for relative small-sized problems. This is mostly due to the fact that, unlike what happens in the local settings, the bandwidth of a nonlocal matrix increases with mesh refinement, since the radius, δ , of the nonlocal neighborhood remains fixed. In this section, we first highlight the need of a parallel implementation, we then propose a parallel algorithm, and finally illustrate its efficiency on two- and three-dimensional problems.

Complexity of nonlocal simulations

For each refinement, i.e. every time the mesh size is halved, the average number of elements contained in the kernel's support increases of a factor 2^N . Consequently, the number of unknowns increases of the same factor and the number of nonzero entries in the matrix increases of 2^{2N} , as opposed to local settings where the average increase at every mesh refinement is 2^N . To better understand the impact of such increase let us consider the quadrilateral mesh used in the previous simulations for the quadratic solution, with a sparse matrix where the allocation for a non-zero entry is 12-bytes: 4 to specify the column location in the row (int) and 8 to store the value (double). At mesh level $ml = 6$, the number of unknowns is $N_h = 148353$ and, for $\delta = 0.2$, the maximum number of entries in one row is 34749. The memory allocation for this mesh level requires approximately 80 GB of memory, while a corresponding local problem would require no more than 50 MB of memory.

Similarly, for each mesh refinement, the computational time to assemble the nonlocal matrix increases in average by the same factor 2^{2N} , since the number of elements increases of 2^N and the average number of elements contained inside the kernel's support increases by 2^N . For a local problem the assembly time increases only of a factor 2^N . These differences in memory allocation and CPU time indicate that a parallel implementation of the nonlocal assembly is vital to make nonlocal models a preferred and viable modeling option.

Details on the implementation

We begin our description by stressing the fact that, even in a local context, a parallel implementation of a FE algorithm is nontrivial. In what follows, we assume that the reader is familiar with the FE method and relatively accustomed to its parallel implementation^[53]; thus, we proceed by highlighting the main challenges that arise in nonlocal implementations.

Our parallel algorithm has been implemented in FEMuS^[54] and is publicly available on GitHub. FEMuS is a in-house FE C++ library that interfaces with PETSC^[55], which provides the linear algebra library for the parallel solver. The parallelization of the nonlocal assembly has been entirely developed within FEMuS.

As it is common in parallel FE settings, the mesh elements are partitioned among the processes N_p ; FEMuS uses the METIS/-PARAMETIS^[56] library for partitioning unstructured meshes. For each process I , we let Ω_I be the domain composed of the

elements owned by I that overlap with Ω and Π_I be the domain composed of the elements owned by I that overlap with Γ . Although highly desirable, each sub-domain $\Omega_I \cup \Pi_I$ does not need to be simply connected. Similar to the definition of Γ , we define the interaction domain Γ_I of Ω_I as

$$\Gamma_I = \{\mathbf{y} \in (\Omega \cup \Gamma) \setminus \Omega_I : |\mathbf{x} - \mathbf{y}| \leq \delta + \varepsilon \text{ for some } \mathbf{x} \in \Omega_I \cup \Pi_I\}. \quad (45)$$

While in the local case two processes I and J have to exchange information only between elements on a shared boundary, i.e. on $\partial\Omega_I \cap \partial\Omega_J$, in the nonlocal case the two processes need to exchange information whenever an element of I intersects the interaction domain Γ_J or vice-versa. We denote by Γ_{JI} the region made up by all the elements owned by process J that intersect with the interaction domain Γ_I , for $I, J = 1, \dots, N_p$. Set $\tilde{\Gamma}_I = \cup_{J \in N_p} \Gamma_{JI}$. The following relations hold

$$\begin{aligned} \Gamma_{JI} \cap \Gamma_{KI} &= \emptyset \text{ for } J \neq K, \\ \Gamma_I &\subseteq \tilde{\Gamma}_I, \\ \Gamma_{II} &= \Pi_I, \\ \Gamma &= \cup_{I \in N_p} \Gamma_{II}, \\ \Gamma_{II} \cap \Gamma_{JJ} &= \emptyset \text{ for } I \neq J. \end{aligned}$$

The following expression clarifies what operations can be performed within one process and which ones require exchange of information among processes. We consider a general integral, whose numerical computation is ubiquitous in Algorithm 1. For any function $w(\mathbf{x}, \mathbf{y})$, any double integral such as those defined in the first part of the paper can be rewritten as

$$\begin{aligned} & \iint_{(\Omega \cup \Gamma)^2} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \\ &= \sum_{I=1}^{N_p} \int_{\Omega \cup \Gamma} \int_{\Omega_I \cup \Gamma_{II}} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \\ &= \sum_{I=1}^{N_p} \int_{\Omega_I \cup \Gamma_I} \int_{\Omega_I \cup \Gamma_{II}} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \\ &= \sum_{I=1}^{N_p} \int_{\Omega_I \cup \tilde{\Gamma}_I} \int_{\Omega_I \cup \Gamma_{II}} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \\ &= \sum_{I=1}^{N_p} \int_{\Omega_I \cup (\cup_{J \in N_p} \Gamma_{JI})} \int_{\Omega_I \cup \Gamma_{II}} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \\ &= \sum_{I=1}^{N_p} \left(\iint_{(\Omega_I \cup \Gamma_{II})^2} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} + \sum_{J \neq I}^{N_p} \int_{\Gamma_{JI}} \int_{\Omega_I \cup \Gamma_{II}} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \right) \\ &= \sum_{I=1}^{N_p} \left(\iint_{(\Omega_I \cup \Gamma_{II})^2} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} + \sum_{J \neq I}^{N_p} \int_{\Gamma_{JI}} \int_{(\Omega_I \cup \Gamma_{II}) \cap \tilde{\Gamma}_J} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \right) \\ &= \sum_{I=1}^{N_p} \left(\iint_{(\Omega_I \cup \Gamma_{II})^2} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} + \sum_{J \neq I}^{N_p} \int_{\Gamma_{JI}} \int_{\Gamma_{IJ}} \gamma_\epsilon(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \right). \quad (46) \end{aligned}$$

A few remarks are in order. In the last equality the first double integral relies on information fully available on process I . In contrast, the outer integrals of the second term are defined on sub-domains Γ_{JI} , which belong to processes different from I . Thus, for process I , this information has to be made available through parallel implementation. Namely, before integration, each process J sends to I the needed information stored in the elements that make up Γ_{JI} through the MPI send/receive protocol^[57]. The inner integral is on the sub-region Γ_{IJ} owned by process I . In the definition of A_{ij}^{21} and A_{ij}^{22} , the inner integral identifies

TABLE 8 Assembly time t_a and total time t_t of simulations with 480641 dofs, linear FEM, and 1440513 dofs, quadratic FEM, $h = 1.5625\text{E-}03$, $\varepsilon = h/2$, $\delta = 0.05$, $L_{max} = 3$, $L_{min} = 1$.

	Linear		Quadratic	
N_p	t_a [s]	t_t [s]	t_a [s]	t_t [s]
36	4268.57 1.89	4470.87 1.90	1795.86 1.77	2168.10 1.68
72	2251.59 1.92	2346.46 1.94	1013.76 1.88	1290.14 1.94
144	1166.77 1.72	1207.84 1.73	538.12 1.64	663.53 1.70
288	675.67 1.85	694.44 1.85	327.75 1.65	388.88 1.72
576	364.65	373.77	197.83	225.32

the i -th row of the matrix through the test function $\varphi_i(\mathbf{y})$. Hence, the degree of freedom of that row is owned by process I . We allocate the memory to store the entries of each row of the sparse parallel matrix on the process that owns the row itself; this results in an optimal parallel assembly since it minimizes (almost removes) the communication time.

Remark 2. As already pointed out in Section 5.1, identifying if a given element intersect a given region can be challenging and computationally expensive. As described before, we rely on a simple and fast conservative test to identify if an element $\mathcal{E}_j$, owned by process J belongs to Γ_{JI} . Namely, we use formula (37) to approximate the minimum distance between the bounding box enveloping $\mathcal{E}_j$ and the bounding box enveloping all the elements owned by processor I . If the approximate distance is less than $\delta + \varepsilon$, then $\mathcal{E}_j$ is included into Γ_{JI} .

Performance tests

We first consider two-dimensional problems with both linear and quadratic FE discretizations with 480641 and 1440513 degrees of freedom, respectively. We consider a quadrilateral mesh with $ml = 7$ (for which $h = 1.5625\text{E-}03$), $\varepsilon = h/2$, $\delta = 0.05$, $L_{max} = 3$, $L_{min} = 1$. Since the high computational costs are mainly due to the assembly procedure, we report separately the CPU times required by the assembly routine (t_a) and the total CPU time (t_t), which also includes the solver time. Computational times, as the number of processors N_p grows, are reported in Table 8. We also report the time ratios $\text{TR}_a(N_p) = t_a(N_p/2)/t_a(N_p)$ and $\text{TR}_t(N_p) = t_t(N_p/2)/t_t(N_p)$. All cases feature large time ratios, illustrating the good scaling properties of the parallel implementation for both linear and quadratic FE discretizations.

We also test the scaling properties of the parallel algorithm with three-dimensional numerical simulations. Similarly to the two-dimensional case, we consider a hexahedral mesh with 615489 degrees of freedom and a quadratic FE discretization. We also set $h = 0.025$, $\varepsilon = h/2$, $\delta = 0.1$, $L_{max} = 2$, $L_{min} = 1$. In Table 9 we report the assembly time t_a , the total time t_t and the time ratios of the numerical simulation for increasing number of processes. As in the two-dimensional case, the parallel simulations show high time ratios, indicating good scaling properties of the algorithm.

In Figure 4 we show the assembly speedup S values referred to $N_p = 36$, defined as $S = t_a(36)/t_a(N_p)$ for two-dimensional linear (left), quadratic (center) and three-dimensional quadratic (right) simulations. We also report the linear speedup (dashed line). On the basis of the presented results, the code shows excellent scalability properties especially in 3D. Thus, it proves to be suitable for large scale nonlocal simulations.

TABLE 9 Assembly time t_a , total time t_t and time ratios of simulations with 615489 dofs, quadratic FEM, $h = 0.025$, $\varepsilon = h/2$, $\delta = 0.1$, $L_{max} = 2$, $L_{min} = 1$.

N_p	t_a [s]	t_t [s]
36	17598.5 1.92	17727.83 1.92
72	9155.14 2.00	9223.08 2.00
144	4556.1 1.99	4591.51 1.97
288	2286.17	2314.36

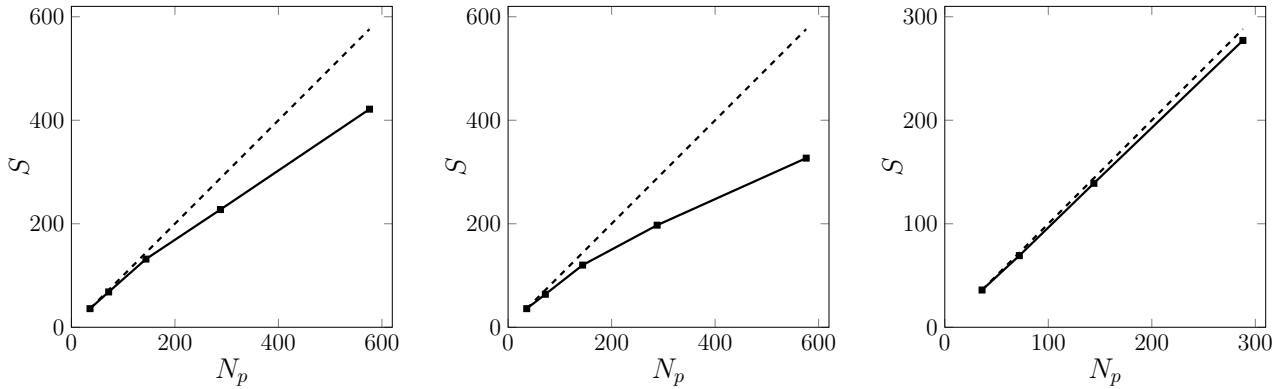


FIGURE 4 Speedup as a function of the number of processors for two-dimensional linear (left) and quadratic FE (center), and three-dimensional quadratic FE (right).

7 | CONCLUSION

We introduced an efficient, flexible, and scalable algorithm for FE discretizations of nonlocal diffusion problems characterized by compactly supported kernels. The novelty of our work is three fold. First, we circumvent the numerical difficulties arising from the integration of discontinuous kernels by multiplying the integrand function by a mollifier. This allows us to avoid the tedious and costly process of identifying ball-element intersections, hence avoiding integration over partial elements and/or curved elements. These tasks, even though manageable in two-dimensional settings, become nontrivial in three dimensions. Second, we introduce an adaptive quadrature rule that allows us accurately resolve the high gradients featured by the integrand function without compromising the computational efficiency. Third, we propose a parallel implementation of the mollified, adaptive algorithm that shows excellent scalability properties especially in three dimensions, where the cost of numerical integration dominates. Our numerical results illustrate the theoretical findings and show that, in two dimensions, our algorithm is competitive with other efficient approximations of FE implementations. In three dimensions, this is the first efficient and scalable parallel implementation that has no constraints on the type of mesh or FE spaces.

As such, this work contributes to making variational discretizations of nonlocal models a viable option, even for large scale problems. It also represents an effort towards increasing the usability of nonlocal equations, for which the high computational costs often hinders their popularity in engineering contexts, despite their undeniable improved accuracy.

ACKNOWLEDGMENTS

Author contributions

This is an author contribution text. This is an author contribution text. This is an author contribution text. This is an author contribution text. This is an author contribution text.

Financial disclosure

This work was supported by the National Science Foundation (NSF) Division of Mathematical Sciences (DMS) program, project 1912902, by Sandia National Laboratories (SNL) Laboratory-directed Research and Development (LDRD) program, project 218318 and by the U.S. Department of Energy, Office of Advanced Scientific Computing Research under the Collaboratory on Mathematics and Physics-Informed Learning Machines for Multiscale and Multiphysics Problems (PhILMs) project. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration contract number DE-NA0003525. This paper, SAND2021-0672, describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Conflict of interest

The authors declare no potential conflict of interests.

SUPPORTING INFORMATION

None reported.

How to cite this article: E. Aulisa, G. Capodaglio, A. Chierici, and M. D'Elia (2021), Efficient quadrature rules for finite element discretizations of nonlocal equations, *NMPDE*, 2021.

APPENDIX

A SECTION TITLE OF FIRST APPENDIX

A.1 Subsection title of first appendix

B SECTION TITLE OF SECOND APPENDIX

References

1. Benson D, Wheatcraft S, Meerschaert M. Application of a fractional advection-dispersion equation. *Water Resources Research* 2000; 36(6): 1403-1412.
2. Benson D, Schumer R, Meerschaert M, Wheatcraft S. Fractional Dispersion, Lévy Motion, and the MADE Tracer Tests. *Transport in Porous Media* 2001; 42: 211-240.
3. Deng ZQ, Singh V, Bengtsson L. Numerical Solution of Fractional Advection-Dispersion Equation. *Journal of Hydraulic Engineering* 2004; 130(5).
4. Schumer R, Benson D, Meerschaert M, Baeumer B. Multiscaling fractional advection-dispersion equations and their solutions. *Water Resources Research* 2003; 39(1): 1022-1032.

5. Schumer R, Benson D, Meerschaert M, Wheatcraft S. Eulerian derivation of the fractional advection-dispersion equation. *Journal of Contaminant Hydrology* 2001; 48: 69-88.
6. Ha YD, Bobaru F. Characteristics of dynamic brittle fracture captured with peridynamics. *Engineering Fracture Mechanics* 2011; 78(6): 1156–1168.
7. Littlewood D. Simulation of Dynamic Fracture using Peridynamics, Finite Element Modeling, and Contact. In: Proceedings of the ASME 2010 International Mechanical Engineering Congress and Exposition, Vancouver, British Columbia, Canada. ; 2010.
8. Silling S. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids* 2000; 48: 175–209. [doi: 10.1016/S0022-5096\(99\)00029-0](https://doi.org/10.1016/S0022-5096(99)00029-0)
9. Leoni PCD, Zaki TA, Karniadakis G, Meneveau C. Two-point stress-strain rate correlation structure and non-local eddy viscosity in turbulent flows. Submitted to *Journal of Fluid Mechanics*; 2020.
10. Pang G, D’Elia M, Parks M, Karniadakis GE. nPINNs: nonlocal Physics-Informed Neural Networks for a parametrized nonlocal universal Laplacian operator. Algorithms and Applications. *Journal of Computational Physics* 2020. To appear.
11. Buades A, Coll B, Morel J. Image denoising methods. A new nonlocal principle. *SIAM Review* 2010; 52: 113-147.
12. D’Elia M, De los Reyes JC, Miniguano-Trujillo A. Bilevel parameter optimization for nonlocal image denoising models. arXiv:1912.02347; 2019.
13. Gilboa G, Osher S. Nonlocal linear image regularization and supervised segmentation. *Multiscale Model. Simul.* 2007; 6: 595-630.
14. Lou Y, Zhang X, Osher S, Bertozzi A. Image recovery via nonlocal operators. *Journal of Scientific Computing* 2010; 42: 185-197.
15. Burch N, D’Elia M, Lehoucq R. The exit-time problem for a Markov jump process. *The European Physical Journal Special Topics* 2014; 223: 3257-3271.
16. D’Elia M, Du Q, Gunzburger M, Lehoucq R. Nonlocal convection-diffusion problems on bounded domains and finite-range jump processes. *Computational Methods in Applied Mathematics* 2017; 29: 71-103.
17. Meerschaert M, Sikorskii A. *Stochastic models for fractional calculus*. Studies in mathematics, Gruyter . 2012.
18. Metzler R, Klafter J. The random walk’s guide to anomalous diffusion: a fractional dynamics approach. *Physics Reports* 2000; 339: 1-77.
19. Metzler R, Klafter J. The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics. *Journal Physics A* 2004; 37: 161-208.
20. D’Elia M, Gulian M, Olson H, Karniadakis GE. A Unified Theory of Fractional, Nonlocal, and Weighted Nonlocal Vector Calculus. arXiv:2005.07686; 2020.
21. Burkovska O, Glusa C, D’Elia M. An optimization-based approach to parameter learning for fractional type nonlocal models. Preprint arXiv:2010.03666; 2020.
22. D’Elia M, Gunzburger M. Optimal distributed control of nonlocal steady diffusion problems. *SIAM Journal on Control and Optimization* 2014; 55: 667-696.
23. D’Elia M, Gunzburger M. Identification of the diffusion parameter in nonlocal steady diffusion problems. *Applied Mathematics and Optimization* 2016; 73: 227-249.
24. Gulian M, Raissi M, Perdikaris P, Karniadakis GE. Machine learning of space-fractional differential equations. *SIAM Journal on Scientific Computing* 2019; 41(4): A2485-A2509.

25. Pang G, Lu L, Karniadakis GE. fPINNs: Fractional Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing* 2019; 41: A2603-A2626.
26. Pang G, Perdikaris P, Cai W, Karniadakis GE. Discovering variable fractional orders of advection–dispersion equations from field data using multi-fidelity Bayesian optimization. *Journal of Computational Physics* 2017; 348: 694 - 714.
27. Xu X, D’Elia M, Foster J. A machine-learning framework for peridynamic material models with physical constraints. Preprint arXiv:2101.01095; 2021.
28. You H, Yu Y, Trask N, Gulian M, D’Elia M. Data-driven learning of robust nonlocal physics from high-fidelity synthetic data. arXiv:2005.10076; 2020.
29. You H, Yu Y, Silling S, D’Elia M. Data-driven learning of nonlocal models: from high-fidelity simulations to constitutive laws. Preprint arXiv:2012.04157; 2020.
30. Alali B, Gunzburger M. Peridynamics and material interfaces. *Journal of Elasticity* 2015; 120(2): 225-248.
31. Capodaglio G, D’Elia M, Bochev P, Gunzburger M. An energy-based coupling approach to nonlocal interface problems. *Computers and Fluids* 2020.
32. Cortazar C, Elgueta M, Rossi J, Wolanski N. How to approximate the heat equation with Neumann boundary conditions by nonlocal diffusion problems. *Archive for Rational Mechanics and Analysis* 2008; 187: 137-156.
33. D’Elia M, Tian X, Yu Y. A physically-consistent, flexible and efficient strategy to convert local boundary conditions into nonlocal volume constraints. *SIAM Journal of Scientific Computing* 2020; 42(4): A1935-A1949.
34. Ainsworth M, Glusa C. Towards an efficient finite element method for the integral fractional Laplacian on polygonal domains. In: Springer. 2018 (pp. 17-57).
35. Capodaglio G, D’Elia M, Gunzburger M, Bochev P, Klar M, Vollmann C. A general framework for substructuring-based domain decomposition methods for models having nonlocal interactions. arXiv:2008.11780; 2020.
36. D’Elia M, Du Q, Glusa C, Tian X, Zhou Z. Numerical methods for nonlocal and fractional models. *ACTA Numerica* 2020; 29.
37. D’Elia M, Gunzburger M, Vollmann C. A cookbook for finite element methods for nonlocal problems, including quadrature rule choices and the use of approximate neighborhoods. arXiv:2005.10775; 2020.
38. Pasetto M. *Enhanced Meshfree Methods for Numerical Solution of Local and Nonlocal Theories of Solid Mechanics*. PhD thesis. University of California, San Diego, CA; 2019.
39. Silling SA, Askari E. A meshfree method based on the peridynamic model of solid mechanics. *Computers & structures* 2005; 83(17-18): 1526–1535.
40. Wang H, Wang K, Sircar T. A direct $\mathcal{O}(N \log^2 N)$ finite difference method for fractional diffusion equations. *Journal of Computational Physics* 2010; 229(21): 8095–8104.
41. Chen Y, Lee J, Eskandarian A. *Meshless methods in solid mechanics*. Springer Science & Business Media . 2006.
42. Parks M, Littlewood D, Mitchell J, Silling S. Peridigm Users Guide. Tech. Rep. SAND2012-7800, Sandia National Laboratories; NM, USA: 2012.
43. Parks M, Seleson P, Plimpton S, Lehoucq R, Silling S. Peridynamics with LAMMPS: A User Guide. Tech. Rep. SAND2010-5549, Sandia National Laboratories; NM, USA: 2010.
44. Du Q, Gunzburger M, Lehoucq R, Zhou K. A NONLOCAL VECTOR CALCULUS, NONLOCAL VOLUME-CONSTRAINED PROBLEMS, AND NONLOCAL BALANCE LAWS. *Mathematical Models and Methods in Applied Sciences* 2013; 23: 493–540.

45. Du Q, Gunzburger M, Lehoucq R, Zhou K. Analysis and approximation of nonlocal diffusion problems with volume constraints. *SIAM review* 2012; 54: 667–696.
46. Mousavi S, Pask J, Sukumar N. Efficient adaptive integration of functions with sharp gradients and cusps in n-dimensional parallelepipeds. *International journal for numerical methods in engineering* 2012; 91(4): 343–357.
47. Mengesha T, Du Q. Analysis of a scalar nonlocal peridynamic model with a sign changing kernel. *Discrete and Continuous Dynamical Systems - B* 2013; 18(5): 1415-1437.
48. Du Q, Gunzburger M, Lehoucq R, Zhou K. Analysis and approximation of nonlocal diffusion problems with volume constraints. *SIAM Review* 2012; 54(4): 667-696.
49. Macek RW, Silling SA. Peridynamics via finite element analysis. *Finite Elements in Analysis and Design* 2007; 43(15): 1169-1178.
50. Wang H, Tian H. A fast and faithful collocation method with efficient matrix assembly for a two-dimensional nonlocal diffusion model. *Computer Methods in Applied Mechanics and Engineering* 2014; 273: 19 - 36.
51. Vollmann C, Schulz V. Exploiting multilevel Toeplitz structures in high dimensional nonlocal diffusion. *Computing and Visualization in Science* 2019; 20: 29-46.
52. Ainsworth M, Glusa C. Aspects of an adaptive finite element method for the fractional Laplacian: A priori and a posteriori error estimates, efficient implementation and multigrid solver. *Computer Methods in Applied Mechanics and Engineering* 2017; 327: 4-35.
53. Babuška I, Elman HC. Some aspects of parallel implementation of the finite-element method on message passing architectures. In: . I. Elsevier. 1990 (pp. 157–187).
54. Aulisa E, Bnà S, Bornia G. FEMuS. <https://github.com/eaaulisa/MyFEMuS.git>, 2014.
55. Balay S, Abhyankar S, Adams M, et al. *PETSc users manual*. Argonne National Laboratory . 2019.
56. Karypis G, Schloegel K, Kumar V. *Parmetis: Parallel graph partitioning and sparse matrix ordering library*. University of Minnesota . 1997.
57. Gropp W, Gropp WD, Lusk E, Skjellum A, Lusk ADFEE. *Using MPI: portable parallel programming with the message-passing interface*. 1. MIT press . 1999.

AUTHOR BIOGRAPHY



Eugenio Aulisa is Professor of Computational Mathematics in the Department of Mathematics and Statistics at Texas Tech University, Lubbock, USA. His primary research interests are in computational fluid mechanics, modeling and simulation of multiphase flows, fluid-structure interaction problems, non-linear analysis of fluid flow filtration in porous media, multigrid solvers with domain decomposition methods and geometric control. He holds a Ph.D in Energetic, Nuclear, and Environmental Control Engineering from the University of Bologna, Italy.



Giacomo Capodaglio is a Postdoctoral Researcher at Los Alamos National Laboratory, currently developing and implementing time-stepping schemes for the efficient simulation of coastal ocean. He obtained his master degree in Energy Engineering from Università di Bologna, in Italy, and his Ph.D in Applied Mathematics from Texas Tech University, where he worked on multigrid algorithms for FE applications with arbitrary-level hanging node configurations. He then joined Florida State University as a Postdoctoral Research Associate, to work with Prof. Gunzburger on probability density estimation applied to random PDEs and on the modeling and simulation of domain decomposition and interface problems in nonlocal diffusion.



Andrea Chierici is a Research fellow at the Department of Industrial Engineering of the University of Bologna, Italy. He obtained his master degree in Energy Engineering from University of Bologna in 2017. From 2017 to 2020, he was a PhD student at DIMSAI (University of Bologna) where he worked on the numerical modeling of fluid-structure interaction problems and optimal control theory with Prof. Manservigi. He also worked on multiphase flows with Prof. Scardovelli. In 2020, he was a short-term scholar at Texas Tech University, working on numerical modeling of fractional operators.



Marta D'Elia obtained her master degree in Mathematical Engineering at Politecnico di Milano under the supervision of Prof. Quarteroni. She obtained her Ph.D in Applied Mathematics at Emory University under the supervision of Prof. Veneziani. Here, she worked on optimal control in CFD for cardiovascular applications. She was a postdoctoral fellow at Florida State University where she worked with Prof. Gunzburger on optimization for nonlocal and fractional models. In February 2014 she joined Sandia National Laboratories, where she works in the Computational Science and Analysis group at the California site. Her interests include nonlocal modeling and simulation, optimization and optimal control, and scientific machine learning.