Iso-geometric Collocation Method for the Fractional Laplacian in the 2D Bounded Domain

Kailai Xu\textsuperscript{a}, Eric Darve\textsuperscript{b}

\textsuperscript{a}Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, 94305
\textsuperscript{b}Mechanical Engineering, Stanford University, Stanford, CA, 94305

Abstract

A numerical method for the fractional Poisson problem in a 2D bounded domain is proposed, based on iso-geometric analysis and the singularity subtraction. The basis functions of iso-geometric analysis have higher order continuous derivatives, and therefore we can construct a fourth order approximation for the numerator of the hyper-singular integrand. The accuracy of the method is numerically shown to be (or better than) $O(N^{-1})$ for smooth solutions. With iso-geometric analysis, we can deal with complex geometries. The singularity subtraction enables us to use efficient and accurate quadrature rules. We also investigate the influence of the solution regularity to the accuracy.

Keywords: Iso-geometric Analysis, Fractional Laplacian, Singularity Subtraction

1. Introduction

Fractional calculus has been a topic of active research due to its applications in anomalous diffusion, soft matter \cite{1}, image denoising \cite{2}, small price changes in the S&P 500 \cite{3}, etc. The fractional Laplacian

$$\left(-\Delta\right)^{\alpha}u(x) := c_{\alpha,2} \text{ p.v.} \int_{\mathbb{R}^2} \frac{u(x) - u(y)}{|x - y|^{2+2\alpha}} \, dy$$

(1)

where

$$c_{\alpha,2} = \frac{2^{2\alpha}\Gamma(\alpha + 1)}{\pi|\Gamma(-\alpha)|}$$

and p.v. denotes that the integral is defined using a principal value, is of particular interest both practically and theoretically.

Fractional Laplacian is well-established on $\mathbb{R}^n$ but remains an open problem in a bounded domain. Among many definitions, two definitions are widely known and intensively studied

\textsuperscript{*}The first author thanks the Stanford Graduate Fellowship in Science & Engineering and the 2018 Schlumberger Innovation Fellowship for the financial support.

Email addresses: kailaix@stanford.edu (Kailai Xu), darve@stanford.edu (Eric Darve)
in the literature [4]: the integral operator eq. (1) and the spectral operator. The latter is defined by considering the eigenvalues \( \{ \lambda_k \}_{k \geq 0} \) and eigenfunctions \( \{ \varphi_k \}_{k \geq 0} \) of the Laplacian operator \(-\Delta\) in \( \Omega \) with zero Dirichlet boundary data on \( \partial \Omega \):

\[
\begin{align*}
-\Delta \varphi_k &= \lambda_k \varphi_k \quad \text{in } \Omega \\
\varphi_k &= 0 \quad \text{on } \partial \Omega
\end{align*}
\]

where \( \varphi_k(x) \) are normalized \( \| \varphi_k(x) \|_{L^2(\Omega)} = 1 \).

For any \( \alpha \in (0, 1), u \in H^2_0(\Omega) \) with

\[
u(x) = \sum_{k=1}^{\infty} a_k \varphi_k(x), \quad x \in \Omega
\]

From this decomposition, we can define a fractional Laplacian by taking a fractional power of the eigenvalues:

\[
(-\Delta)^{\alpha} u = \sum_{k=1}^{\infty} a_k \lambda_k^{\alpha} \varphi_k(x)
\]

This definition has been successful in many applications, for example in image denoising [5].

An important aspect of this approach is that it requires only data on the boundary of the domain. However, we know that many physical applications require the definition of boundary conditions not just on the boundary \( \partial \Omega \), but in \( \Omega^c := \{ y : y \neq x, \forall x \in \Omega \} \) (for example, see [6] for many physical applications). That is needed in the integral definition eq. (1) for example. These two definitions are in fact incompatible, and one operator cannot be mapped to the other through some suitable transformation. Only when \( \Omega = \mathbb{R}^n \) and for periodic functions on cubic domains do these definitions agree.

The choice of definition will depend on the application. In many science and engineering applications, where fractional Laplacians arise from modeling stochastic Lévy processes, the natural definition is the integral definition eq. (1) where boundary values outside of \( \Omega \) need to be specified (because of the presence of stochastic jumps of arbitrary lengths outside of \( \Omega \)). That is the definition we will use in this article.

In this paper, we will focus on the 2D model problem

\[
(-\Delta)^{\alpha} u(x) = f(x) \quad x \in \Omega
\]

\[
u(x) = 0 \quad x \in \Omega^c
\]

where \( \Omega \) is a bounded domain with Lipschitz boundary.

Finding robust and stable numerical methods for eq. (2) has been an active research area. It is not until recently that this problem has been extensively studied. Here we review some of the state-of-the-art methods for this problem.

**Finite element method** [7–10]. It is possible to state eq. (2) in the weak formulation:

Find \( u \in H^{\alpha}(\Omega) \): \( a(u, v) = (f, v), \quad \forall v \in H^{\alpha}(\Omega) \)
where

$$a(u, v) = \frac{c_{\alpha, 2}}{2} \int_\Omega d\mathbf{x} \int_\Omega d\mathbf{y} \frac{(u(x) - u(y))(v(x) - v(y))}{|x - y|^{2+2\alpha}} + \frac{c_{\alpha, 2}}{2} \int_\Omega d\mathbf{x} \int_{\partial\Omega} d\mathbf{y} \frac{u(x)v(x)n_y \cdot (x - y)}{|x - y|^{2+2\alpha}}$$

and where $n_y$ is the inward normal to $\partial\Omega$ at $y$, and the space $H^\alpha(\Omega)$ is defined in [7].

There are several advantages of the finite element method. First, the stiffness matrix entries involve singular integrals, but those can be computed using existing techniques for boundary integral equations. Second, strong assumptions on the shape of the underlying domain or the regularity of the solutions are not required. The numerical experiment and analysis show good convergence for even $C^{0,\alpha}_0$ functions. Computation and storage complexity can also be optimized using low-rank approaches. For example, in [10], the authors proposed a $O(N \log^4 N)$ matrix-vector product complexity method based on the panel clustering. However, it is rather complicated to implement the finite element methods and the convergence can be slow for less regular solutions.

**Walk-on-the-sphere (WOS) method.** From a probabilistic point of view, the fractional Laplacian is the generator of the symmetric $\alpha$-stable process [11–15]. [15] proposed leads to the walk-on-the-sphere method based on the Feynman-Kac formula

$$u(x) = \mathbb{E}_x[g(X_{\sigma_{\Omega}})], \quad x \in \Omega$$

where $X = (X_t, t \geq 0)$ is an isotropic stable Lévy process with index $\alpha$ and $\sigma_{\Omega} = \inf\{t > 0 : X_t \notin \Omega\}$.

The implementation of WOS method is quite simple and direct. It is useful for providing a reference solution pointwise. The main drawback is that such an approach can be inefficient and slow if we need to compute $u(x)$ at many points.

**Meshless method.** The meshless method is very flexible as it does not require any mesh. The basis functions are smooth so that the evaluation of the fractional Laplacian can be very accurate. [16] proposed a radial basis function collocation method for solving eq. (2). The vector Grünwald scheme [17]

$$(-\Delta)^\alpha u(x) = h^{-\alpha} \sum_{m=0}^{\infty} (-1)^m \binom{\alpha}{m} \int_{\|\theta\|=1} f(x - mh\theta) \frac{d\theta}{2\pi} + O(h)$$

is used for evaluating the directional fractional Laplacian.

However, meshless methods typically result in a nearly singular coefficient matrix; the efficient preconditioning and the stabilization can be tricky in some circumstances.
Finite difference. There are many papers on the finite difference method [18, 19] for the fractional Laplacian. [20] proposed an $O(h^2)$ scheme for smooth functions in 1D based on numerical quadrature correction. [21] proposed a $O(h^2)$ scheme for smooth functions in 1D, 2D and 3D based on window function techniques, and the computational cost is $O(N \log N)$ due to the use of FFT.

Compared to the other methods, the finite difference method enjoys higher accuracy for smooth functions, and the resulting linear system is relatively more stable and easier to precondition. However, the methods mentioned require uniform meshes on regular domains. For irregular domains, it may require very fine mesh to attain a certain accuracy, which can be very expensive.

In this paper, we will propose the iso-geometric collocation method. This method enjoys many advantages compared to the methods mentioned above:

- The basis functions for iso-geometric analysis can be as smooth (i.e., the continuity of derivatives) as required, which is needed to evaluate the principal value of the singular integral. It can be compared with the meshless method and the finite difference method.

- The basis functions are locally supported, and the test function space is large enough to incorporate a large class of functions. Therefore the resulting stiffness matrix is usually well-conditioned.

- The iso-geometric collocation method often requires less computational effort for numerical quadratures compared to the Galerkin method [?].

Table 1 shows a comparison of all the methods mentioned.

| Method      | FEA | WOS          | Meshless   | FD         | IgA-C       |
|-------------|-----|--------------|------------|------------|-------------|
| Accuracy    | $O(h)$ | $O(\frac{1}{\sqrt{n}})$ | $O(h)$ | $O(h^2)$ | $O(n^{-1})$ |
| Memory      | $O(n \log^4 n)$ | $O(1)$ | Depends | $O(n \log n)$ | $O(nM)$ |
| Flops       | $O(n \log^4 n)$ | $O(n)$ | Depends | $O(n \log n)$ | $O(nM)$ |
| Uns. Mesh   | ✓   | ✓            | ✓          | ✗          | ✓           |
| Less regular | ✓   | ✓            | ✗          | Less accu. | ✗           |

Table 1: Comparison of different methods. For WOS, the measurement is computed w.r.t. only one point, and $n$ is the number of simulations we carry out. For the others, $n$ is the number of unknowns, $h$ is the average mesh size. FEA: finite element analysis; WOS: walk-on-the-spheres; Meshless: meshless method; FD: finite difference; IgA-C: iso-geometric analysis with collocation; Memory: memory requirement for the method; Flops: floating point operations required; Uns. Mesh: method applies to unstructured meshes; Less regular: allows for solutions that are less differentiable; Less accu.: method has lower accuracy in that case; $M$: number of quadrature points. The convergence rate for IgA-C was observed to be close to $1/n$. Proofs for collocation methods are typically more involved than for variational formulations. The convergence depends on the smoothness of the solution, the order of the NURBS, and the choice of knot vectors.

Our numerical analysis and experiment are carried out in 2D. The method can be extended to higher dimensions without significant changes.

December 21, 2018
For readability, we list the symbols used the most in Table 2. Throughout the paper, unless specified, the notation should be understood as shown in Table 2.

| Notation | Description |
|----------|-------------|
| $\mathbf{x}$ | A point in $\mathbb{R}^2$. |
| $\Omega$ | Domain where the fractional Laplacian is defined. |
| $\sigma(r)$ | Window function eq. (4). |
| $c_{\alpha,2}$ | Coefficient for fractional Laplacian eq. (1). |
| $N_0$ | Number of quadrature points. |
| $N$ | Number of control points/collocation points. |

Table 2: Notations used in the paper.

2. Preliminary: Iso-geometric Collocation Method

Tensor-product B-splines and their rational extension, called NURBS, are commonly used in computer graphics to generate and represent curves and surfaces. Due to their popularity in CAD, they have also been developed for analysis. In iso-geometric analysis (IgA) proposed by Hughes et al. [22], NURBS serves as the fundamental tool to represent the domain but are also used to approximate the solution to PDEs directly. NURBS curves and surfaces have many useful properties such as convex hull property, affine invariance, and variation diminishing property [23]. They can also represent conic sections and quadric surfaces accurately. Compared to the classical finite element analysis which is based on element-wise piecewise polynomials, NURBS basis functions enjoy higher continuity which can be tuned by varying the degree of the functions. As a result, the collocation method can be applied to the strong form of PDEs. For more details on the iso-geometric collocation method, there are many papers on this subject [24, 25].

For completeness, we give a brief introduction to the iso-geometric analysis method here. We first need to introduce the B-splines in one dimension. We define the following notations:

| Notation | Description |
|----------|-------------|
| $n \in \mathbb{N}$ | Number of basis functions |
| $p \in \mathbb{N}_0$ | Degree of polynomial |
| $\Xi$ | $\{0 = \xi_1 \leq \xi_2 \leq \cdots \leq \xi_{n+p+1} = 1\}$ knot vector |
| $r_i$ | Number of repetitions of $\xi_i \in \Xi$ |
| $k$ | $\{k_1, k_2, \ldots, k_m\}$, $k_j = p - r_j + 1$, continuity numbers |
| $B_{i,p}$ | Piecewise polynomial of degree $p$ and continuous derivatives up to order $k_i - 1$ at knot $\xi_j$, with compact support in $[\xi_i, \xi_{i+p+1}]$ |

December 21, 2018
The basis functions $B_{i,p}$ can be defined recursively by

$$B_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,p} = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(\xi)$$

For higher dimensions, the B-splines are defined by the tensor product of B-splines in one dimension

$$B_{i_1,i_2,\ldots,i_d}(x_1, \ldots, x_d) = B_{i_1,p_1}(x_1) \cdots B_{i_d,p_d}(x_d)$$

NURBS in $\mathbb{R}^d$ are conic projections of B-splines in $\mathbb{R}^{d+1}$. We need to distinguish the parametric space $\hat{\Omega} = [0,1]^d$ and physical space $\Omega$. The basis NURBS can be constructed from B-spline basis functions

$$N_{i_1,i_2,\ldots,i_d} = \frac{w_{i_1,i_2,\ldots,i_d}B_{i_1,i_2,\ldots,i_d}}{w}$$

where $w$ is sum of the weighting functions

$$w = \sum_{i_1,\ldots,i_d} w_{i_1,\ldots,i_d} B_{i_1,\ldots,i_d} \quad w_{i_1,\ldots,i_d} > 0$$

The weights $w_{i_1,\ldots,i_d}$ are used to move the mapping closer to/further away from the control points $C_i$ described below. We then define the NURBS surface in $\mathbb{R}^d$ as

$$F(u) = \sum_{i=1}^{n} C_i N_i(u), \quad C_i \in \mathbb{R}^d, \quad u \in [0,1]^d$$

The index $i$ is an integer and corresponds to a “flattened” indexing (with a mapping from the multi-index $i_1, \ldots, i_d$ to $i$). The points $C_i$ are control points. As the corresponding weight $w_i$ increases, $F(u)$ moves closer to $C_i$. In this way, by varying $C_i$ and the weights, one can obtain different mappings.

In this paper, we are concerned with NURBS “surfaces” but in two dimensions; this corresponds to a 2D-parameterization of a domain in $\mathbb{R}^2$. An example of such 2D-parameterization is shown in fig. 1.
Figure 1: An example of NURBS surface: the square $[0, 1]^2$ in the parametric space is mapped to the disk with radius 1.5 in the physical space.

Every function $u(x)$ can be approximated by a linear combination of the NURBS basis with $N$ unknown coefficients $c_i$ ($x = F(u)$)

$$u_h(x) = \sum_{i=1}^{N} c_i N_i(F^{-1}(x))$$

We need to point out that given $u$, it is easy to evaluate $x = F(u)$. However, for a given $x$, finding the corresponding $u$ requires solving a nonlinear equation. This can be quite difficult when the mesh has large distortion. In this case large number of iterations or proper initial guess is required.

The collocation method works by selecting several points in the parametric space $\{x_i\}_{i=1}^{M}$ and impose the equations that $u(x)$ should satisfy at $x_i$ on $u_h(x)$. This will result in a linear system in the form of

$$AX = b$$  \hspace{1cm} (3)

$A$ is not necessarily a square matrix, and if that is the case, the solution may be sought using least square.

One commonly used collocation points are Greville abscissae [22], which are the averages of the knots,

$$\xi_i = \frac{v_{i+1} + v_{i+2} + \ldots + v_{i+p-1}}{p - 1}, \quad \eta_i = \frac{u_{i+1} + u_{i+2} + \ldots + u_{i+p-1}}{p - 1}$$

With this set of collocation points, the matrix in eq. (3) is a square matrix.

December 21, 2018
3. Method

3.1. Discretization of the Fractional Laplacian

One of the challenges in the discretization of the fractional Laplacian is to numerically compute

\[ \frac{u(x) - u(y)}{|x - y|^{2+2\alpha}} \]

which is only defined in the principal value sense. Since \( \alpha < 1 \), we need the numerator to decay at least like \(|x - y|^4\) if we want the integrand to be bounded near \( y = x \). To achieve this, we are going to decompose the numerator into a term that decays as \(|x - y|^4\) and a term that can be integrated analytically.

For this, we will make use of the window function technique introduced in [21]. Let

\[ \sigma(r) = \begin{cases} 1 - 35 \left( \frac{r}{h} \right)^4 + 84 \left( \frac{r}{h} \right)^5 - 70 \left( \frac{r}{h} \right)^6 + 20 \left( \frac{r}{h} \right)^7 & r < h \\ 0 & \text{otherwise} \end{cases} \]

where \( h \) is called window size. This window function satisfies \( 1 - \sigma(r) \sim \mathcal{O}(r^4) \), \( r \to 0 \) (fig. 2).

This vanishing property near 0 is crucial to our analysis, which will be explained below.

![Figure 2: An example of the window function eq. (4) for \( h = 1 \). The function behaves like \( 1 - \mathcal{O}(r^4) \) near the origin.](image)

Let us define \( \mathbf{v} = (v_1, v_2) = y - x \), and

\[
\begin{align*}
    u_i &:= \frac{\partial u}{\partial x_i}, \quad u_{ij} := \frac{\partial^2 u}{\partial x_i \partial x_j}, \quad u_{ijk} := \frac{\partial^3 u}{\partial x_i \partial x_j \partial x_k}, \quad i, j, k \in \{1, 2\}
\end{align*}
\]
The second part of eq. (6) can be computed analytically:

\[
g_k(y) := u_1(x)v_1 + u_2(x)v_2 \\
+ u_{11}(x)\frac{v_1^2}{2} + u_{22}(x)\frac{v_2^2}{2} + u_{12}(x)v_1v_2 \\
+ u_{111}(x)\frac{v_1^3}{6} + u_{122}(x)\frac{v_1^2v_2}{2} + u_{112}(x)\frac{v_1v_2^2}{2} + u_{222}(x)\frac{v_2^3}{6}
\]

(5)

We will use the following decomposition:

\[
u(x) - u(y) = \left[ u(x) - u(y) + \sigma(|x - y|)g_k(y) \right] - \sigma(|x - y|)g_k(y)
\]

Roughly speaking, due to our choice of the window function \(\sigma\) is \(O(|x - y|^{1+2\alpha})\) near \(y \approx x\), whose exponent is larger than that of the hyper-singular kernel \(\frac{1}{|x - y|^{1+2\alpha}}\).

The full integral to calculate is:

\[
\int_{\mathbb{R}^2} \frac{u(x) - u(y)}{|x - y|^{2+2\alpha}} dy = \int_{\mathbb{R}^2} \frac{u(x) - u(y) + \sigma(|x - y|)g_k(y)}{|x - y|^{2+2\alpha}} dy - \int_{\mathbb{R}^2} \frac{\sigma(|x - y|)g_k(y)}{|x - y|^{2+2\alpha}} dy
\]

(6)

The second part of eq. (6) can be computed analytically:

\[
\int_{\mathbb{R}^2} \frac{\sigma(|x - y|)g_k(y)}{|x - y|^{2+2\alpha}} dy = \frac{1}{2} \Delta u(x) \int_{\mathbb{R}^2} \frac{\sigma(|y|)y^2}{|y|^{2+2\alpha}} dy
\]

(7)

We used the fact that \(\frac{\sigma(|y|)}{|y|^{2+2\alpha}}\) is an even function and the terms corresponding to the odd derivatives vanish.

For the first part,

\[
\int_{\mathbb{R}^2} \frac{u(x) - u(y) + \sigma(|x - y|)g_k(y)}{|x - y|^{2+2\alpha}} dy
\]

is a function at least \(C^0\) around \(y = x\) (proved in Theorem 1) and thus can be computed using normal quadratures, e.g., a Gauss quadrature. Figure 3 shows how the integrand becomes continuous over the domain.

**Theorem 1.** Assume \(u(x) \in C^4(\bar{\Omega})\), then \(\frac{u(x) - u(y) + \sigma(|x - y|)g_k(y)}{|x - y|^{2+2\alpha}} \in C^0(\bar{\Omega})\).

**Proof.** According to Taylor’s theorem, there exists \(h_\beta : \mathbb{R}^2 \to \mathbb{R}\), s.t.

\[
u(y) = u(x) + g_k(y) + \sum_{|\beta| = 3} h_\beta(y)(y - x)^\beta
\]

\(h_\beta\) can be written in the following integral form

\[
h_\beta(y) = \frac{4}{\beta!} \int_0^1 (1 - t)^3 D^3 u(x + t(y - x)) dt
\]

December 21, 2018
Figure 3: On the left, the figure shows the plot for $|u(y) - u(x)|/|x - y|^{2+2\alpha}$ (with a logscale), and on the right for $(u(y) - u(x) - \sigma(|x - y|) g_x(y))/(|x - y|^{2+2\alpha})$ (with a linear scale), where $x = (0.2, 0.2)$. The domain is $B_1 = \{ y : |y| < 1 \}$ and $u(x) = (1 - |x|^2)^{1+\alpha}, \alpha = 0.3$. After subtracting the $\sigma(|x - y|) g_x(y)/|x - y|^{2+2\alpha}$ term, the function becomes continuous over the domain $B_1$ and therefore Gauss quadratures can be applied.

Therefore, we have

$$\frac{u(x) - u(y) + \sigma(|x - y|) g_x(y)}{|x - y|^{2+2\alpha}} = \frac{(\sigma(|x - y|) - 1) g_x(y) \sum_{|\beta|=4} h_\beta(y)(y - x)^\beta}{|x - y|^{2+2\alpha}}$$

With our choice for $\sigma$, when $y \to x$, we have $\sigma(|x - y|) - 1 \sim O(|x - y|^4)$. Therefore

$$\frac{(\sigma(|x - y|) - 1) g_x(y)}{|x - y|^{2+2\alpha}} \sim O(|x - y|^{2-2\alpha}) = O(1), \quad y \to x$$

This indicates that the singularity is removed and the function is $C^0(\bar{\Omega})$.

And in addition, $h_\beta(y) \in C^0(\Omega)$ and thus

$$-\sum_{|\beta|=4} h_\beta(y)(y - x)^\beta \sim -\sum_{|\beta|=4} h_\beta(y)(y - x)^{\beta-2-2\alpha} \in C^0(\bar{\Omega})$$

In summary, $\frac{u(x) - u(y) + \sigma(|x - y|) g_x(y)}{|x - y|^{2+2\alpha}} \in C^0(\bar{\Omega})$.

From the proof, if $u(x)$ itself has higher order continuity, so does

$$\frac{u(x) - u(y) + \sigma(|x - y|) g_x(y)}{|x - y|^{2+2\alpha}} \quad (8)$$

One implication is that we will be able to evaluate the integral more accurately.

The integral with eq. (8) is done by numerical integration. This is done by first splitting the integration domain into two parts: $B_x(R), R \geq r$, which is a disk centering at $x$ with radius $R$; and $\mathbb{R}^3 \setminus B_x(R)$.

December 21, 2018
For the first part, we can use a quadrature rule on the disk \( B_0(R) \) and obtain
\[
\int_{B_0(R)} \frac{u(x) - u(y) + \sigma(|x - y|)g_k(y)}{|x - y|^{2+2\alpha}} \, dy \approx \sum_{k=1}^{\hat{N}} w_k \frac{u(x) - u(x + \Delta_k) + \sigma(|\Delta_k|)g_k(x + \Delta_k)}{\Delta_k^{2+2\alpha}}
\]
\[
(9)
\]
Here \( \Delta_k \) are quadrature points and \( w_k \) are the associated quadrature weights on \( B_0(R) \).

For the second part, we have
\[
\int_{\mathbb{R}\setminus B_0(R)} \frac{u(x) - u(y) + \sigma(|x - y|)g_k(y)}{|x - y|^{2+2\alpha}} \, dy = \int_{\mathbb{R}\setminus B_0(R)} \frac{u(x)}{|x - y|^{2+2\alpha}} \, dy
\]
\[
= u(x) \frac{\pi}{\alpha R^{2\alpha}}
\]
thanks to the assumption \( R \geq r \).

In summary, we have rewritten p.v. \( \int_{\mathbb{R}^2} \frac{u(x) - u(y)}{|x - y|^{2+2\alpha}} \, dy \) into
\[
\text{p.v.} \int_{\mathbb{R}^2} \frac{u(x) - u(y)}{|x - y|^{2+2\alpha}} \, dy \approx u(x) \frac{\pi}{\alpha R^{2\alpha}} - \frac{1}{2}\Delta u(x) \int_{\mathbb{R}^2} \frac{\sigma(|y|)y^2}{|y|^{2+2\alpha}} \, dy
\]
\[+ \sum_{k=1}^{\hat{N}} w_k \frac{u(x) - u(x + \Delta_k) + \sigma(|\Delta_k|)g_k(x + \Delta_k)}{\Delta_k^{2+2\alpha}} \]
\[
(10)
\]
If we use NURBS approximation of degree greater or equal to 4, then the NURBS approximation has continuous third derivatives. Therefore eqs. (5) and (7) are evaluable pointwise. In fact, let the NURBS approximation be
\[
u_k(x) = \sum_{i=1}^{N} c_i N_i(F^{-1}(x))
\]
we then have
\[
\frac{\partial^n u_k(x)}{\partial x_{i_1} x_{i_2} \ldots x_{i_n}} = \sum_{i=1}^{N} c_i \frac{\partial^n N_i(F^{-1}(x))}{\partial x_{i_1} x_{i_2} \ldots x_{i_n}}
\]
\[
(11)
\]
However, since NURBS basis functions are parametrized in the parametric space, given \( x \), it is not direct to evaluate \( F^{-1}(x) \) and \( \frac{\partial^n N_i(F^{-1}(x))}{\partial x_{i_1} x_{i_2} \ldots x_{i_n}} \), \( n = 0, 1, 2, \ldots \). Usually a Newton’s method [27] can be applied to find \( F^{-1}(x) \) and chain rule can be applied to compute the latter (see the appendix).

Therefore, we can now write p.v. \( \int_{\mathbb{R}^2} \frac{u(x) - u(y)}{|x - y|^{2+2\alpha}} \, dy \) as a weighted sum of the coefficients \( c_i \). For clarification, we denote the numerical approximation to the fractional Laplacian operator by \( L_{h,x} \), i.e.
\[
(-\Delta)^n u(x) \approx \sum_{j=1}^{N} c_j L_{h,x} N_j(x)
\]

\( \text{December 21, 2018} \)
3.2. Collocation

The collocation points are chosen to be the Greville abscissae for each dimension. Such a choice will result in a square matrix. There are other choices for collocation, and it is possible to derive super convergence collocation points for some particular PDEs. However, in this paper, we focus on Greville abscissae.

For the points inside \( x_i \in \Omega \), we enforce that the numerical Fractional Laplacian be equal to the right-hand side

\[
\sum_{j=1}^{N} c_j L_{h,x} N_j(x_i) = f(x_i)
\]

For the points on the boundary of \( \Omega \), we enforce that the approximation value be equal to boundary conditions

\[
\sum_{i=1}^{N} c_i N_i(x_i) = u(x_i) = 0
\] (12)

This results in a linear system

\[
\begin{bmatrix}
A_I \\
A_B
\end{bmatrix} c = \begin{bmatrix}
0 \\
f
\end{bmatrix}
\] (13)

See algorithm 1 for details.

3.3. Different Types of Error

There are several types of error that might influence the accuracy of the solution.

- The quadrature error in eq. (9). It is not an easy task to derive the optimal cubature with minimal function evaluations, and they are known only for some \( N = 3, 4, 6, 7, 10, 12, 18 \) \cite{28, 29}. An alternative approach is to use the product Gauss rules \cite{30–33} for 2D disks. The idea is to consider the quadrature rules for

\[
\int_{x^2+y^2 \leq 1} x^i y^j dxdy = \int_{-1}^{1} |t| \rho^{i+j} d\rho \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^i \varphi \sin^j \varphi d\varphi
\]

We can use Gauss-Chebyshev formula of the first kind for approximation of the inner integration and Gauss quadrature for the outer integration.

For numerical implementations, we have used the open source software provided by \cite{34}. Figure 4 shows an example of the quadrature nodes in the unit disk \( B_0(1) \) of order 20. There are 400 nodes in total.

- The NURBS approximation error. The approximation ability depends on the number of knots, the degrees as well as the mesh structure of the NURBS surface. It is shown that under appropriate assumptions, the NURBS space on the physical domain delivers the optimal rate of the convergence, similar to the finite element space of degree \( p \) \cite{7}. The following theorem gives the global error estimate

December 21, 2018
Theorem 2. Let \( k \) and \( l \) be integer indices with \( 0 \leq k \leq l \leq p + 1 \), we have

\[
\sum_{K \in \mathcal{K}_h} |v - \Pi_{\mathcal{V}_h} v|_{\mathcal{H}^k_h(K)}^2 \leq C_{\text{shape}} \sum_{K \in \mathcal{K}_h} h_K^{2l-k} \sum_{i=0}^l \| \nabla F \|_{L^\infty(F^{-1}(K))}^2 |v|_{\mathcal{H}^i_h(K)}^2 \quad \forall v \in \mathcal{H}^l(\Omega)
\]

where \( \mathcal{K}_h \) are the patches in the parametric space, \( \mathcal{V}_h \) is the NURBS space, \( C_{\text{shape}} \) is a constant depending on the structure of the NURBS surface, \( \mathcal{H}^i \) is the standard Sobolev space, \( h_K \) is the diameter of \( K \), and \( \mathcal{H}^k_h(K) \) are patches in the physical space endowed with its own norms.

The theorem indicates that if \( v \) is sufficiently smooth, increasing the degree of the NURBS suffice (\( p \)-refinement) or the number of control points (\( h \)-refinement) will yield better approximations. However, it may not be the case for less smooth functions.

- The numerical error of solving the linear system eq. (13). The fractional Poisson equation will give rise to the linear system eq. (13), where \( \begin{bmatrix} A_I \\ A_B \end{bmatrix} \) is a dense matrix. Its condition number grows as we increase the number of control points. The convergence of the iterative solver such as GMRES may be slow if no proper preconditioner is used.

In fact, it is shown that for finite element methods, if a family of shape regular and globally quasi-uniform triangulations with maximal element size \( h \) is used, the stiffness matrix satisfies \[10\]

\[
\kappa(A) = Ch^{-2\alpha}
\]

In the paper we have used Greville abscissae as the collocation points due to its simplicity. However, when the meshes are non-uniform, the method derived from these collocation points might be unstable \[35\]
Algorithm 1 iso-geometric Analysis Collocation Method for the Fractional Laplacian

1: Construct the NURBS surfaces, basis functions and connectivity data for $\Omega$.
2: Compute the Greville points for the NURBS surface.
3: Precompute $\pi_{\alpha R} \sigma(\|y\|_1) \|y\|_2^2 \cdot \int_{\mathbb{R}^2} \sigma(\|y\|_1) \|y\|_2^2 \, dy$.
4: for $i = 1, 2, \ldots, N$ do
5:   if $x_i \in \partial \Omega$ then
6:     Add the contribution to the coefficient matrix and the right hand side according to eq. (12).
7:   else
8:     Find the corresponding coordinates in the parametric space $F^{-1}(x_i)$ and $F^{-1}(x_i + \Delta_j)$.
9:     Compute the associated weights to each coefficients at those coordinates according to eq. (11).
10:    Add the contribution to the linear system eq. (13).
11:  end if
12: end for
13: Solve the linear system eq. (13).
14: Postprocess the result.

4. Numerical Experiments

In this section, we present several numerical examples to show the effectiveness of the iso-geometric analysis. The algorithm is implemented using julia-1.0.

4.1. Smooth Solution

Firstly, we consider the problem with smooth solutions. Consider the fractional Poisson problem on the square $\Omega = [-1, 1]^2$

$$\begin{cases}
(\Delta)^\alpha u = f(x) & x \in \Omega \\
u(x) = 0 & x \in \Omega^c
\end{cases} \quad (14)$$

We generate a square NURBS surface for this problem. However, we need to point out that even if the knots vector has uniform intervals, the corresponding Greville abscissae are not uniform distributed. It can be seen from fig. 5 that the density is larger near the boundary. For the numerical quadrature, we use order 64, which contains $64^2 = 4096$ quadrature points. For higher accuracy, more quadrature points are required.
Since we do not have analytical solution for this case, we run the program for different number of control points and use the solution from the finest mesh as the reference solution. The error is computed from the mean squared error of $50^2 = 2500$ uniform distributed points over $\Omega$, excluding the points near the boundary (with a margin about 0.05). The convergence plots are shown in Fig. 6. We see that the error decreases almost monotonously as $N$ increases. In addition, larger degrees of the NURBS surface give rise to smaller mean squared error as $N$ becomes larger. We also observed that for $\alpha = 0.8$, the error ceases to decrease after round $N = 1000$. This is due to the fact that the error is now dominated by the numerical quadrature error instead of the NURBS approximation error. We can get more accurate results by increasing the order of the numerical quadrature, although it will increase the computational cost.

Figure 6: Convergence plot for eq. (14). The error decreases almost monotonously as $N$ increases. In addition, larger degrees of the NURBS surface give rise to smaller mean squared error as $N$ becomes larger. We also observed that for $\alpha = 0.8$, the error ceases to decrease after round $N = 1000$. This is due to the fact that the error is now dominated by the numerical quadrature error instead of the NURBS approximation error. We can get more accurate results by increasing the order of the numerical quadrature, although it will increase the computational cost.
4.2. Less Smooth Solutions

In this section, we consider the application of the iso-geometric analysis method to problems with less smooth solutions. If we consider a unit disk domain \( \Omega = \mathbb{R}^2 \), it is shown in how to construct explicit eigenfunctions for the fractional Laplacian. Particularly, we have the following result in Table 3 for eq. (14).

| \( u(x) \) in the ball | \((-\Delta)\alpha u \) in the ball |
|-------------------------|-----------------------------|
| \( u_1(x) = (1 - |x|^2)^\alpha \) | \( 2^{2\alpha}\Gamma(\alpha + 1)^2 \) |
| \( u_2(x) = (1 - |x|^2)^{\alpha+1} \) | \( 2^{\alpha+2}\Gamma(\alpha + 2)\Gamma(1+\alpha)\Gamma\left(\frac{1}{2}\right)^{-1}\left[1 - (1 + \alpha)|x|^2\right] \)

Table 3: Several functions whose Laplacian can be computed analytically. The third and fourth equations are not radial.

In the numerical experiments, the quadrature order is 64 and \( h = 0.1 \) for the window function. We show the exact and numerical solutions to eq. (14) for \( u_1 \) in Table 3 in fig. 7. On the one hand, we see that larger degrees for the NURBS basis functions do not necessarily make the numerical solution more accurate. Instead, higher order approximation to the non-smooth functions might lead to instability (we can see in the plots that the numerical solution becomes more oscillatory). On the other hand, the smaller \( \alpha \), the more difficult the approximation is by the NURBS basis functions. This is expected since the exact solution to \( \alpha = 0.2 \) has the least regularity on the boundary.

Figure 8 presents the numerical solutions to eq. (14) for \( u_2 \) in Table 3. Compared to fig. 7, we see that the numerical solution has less oscillation near the boundary using the same setting except the first plot (in both cases, the numerical solutions are unstable). Since the solution \( u_2 \) has continuous derivatives near the boundary while the first order derivatives of \( u_1 \) explode near the boundary, we expect better accuracy of the corresponding numerical solution, which is demonstrated in fig. 9.

We also show the error distribution in the disk in fig. 9. We can see that the error for the less smooth solution \( u_1 \) is much larger than that of the smoother solution \( u_2 \) using the same mesh structure and numerical quadrature rules.

4.3. Application: Fractional Diffusion on An Irregular Domain

Finally, we consider the fractional diffusion equation on an irregular domain

\[
\begin{aligned}
&\frac{du}{dt}(t,x) = -(-\Delta)^{\alpha}u(t,x) & x \in \Omega \\
&u(t,x) = 0 & x \in \Omega_c, t > 0 \\
&u(0,x) = f(x) & x \in \mathbb{R}^2
\end{aligned}
\]  

(15)

where \( f(x) = \exp(-|x-x_0|^2) \) and \( x_0 = [5,5] \).

Here \( \Omega \) is deformed from a square. We created the domain as well as its NURBS representation using the open source software \texttt{gnurbs} \[36\]. Figure 10 shows the irregular domain used in eq. (15). The red dots represent the control points of the NURBS surface. Note

December 21, 2018
Figure 7: Solution to eq. (14) with exact solution $u_1$ in Table 3 on the disk domain. On the one hand, we see that larger degrees for the NURBS basis functions do not necessarily make the numerical solution more accurate. Instead, higher order approximation to the non-smooth functions might lead to instability (we can see in the plots that the numerical solution becomes more oscillatory). On the other hand, the smaller $\alpha$, the more difficult the approximation is by the NURBS basis functions. This is expected since the exact solution to $\alpha = 0.2$ has the least regularity on the boundary.
Figure 8: Solution to eq. (14) with exact solution $u_2$ in Table 3 on the disk domain. Compared to fig. 7 we see that the numerical solution has less oscillation near the boundary using the same setting except the first plot (in both cases, the numerical solutions are unstable). Since the solution $u_2$ has continuous derivatives near the boundary while the first order derivatives of $u_1$ explode near the boundary, we expect better accuracy of the corresponding numerical solution, which is demonstrated in fig. 9.
we have placed more nodes in the center of the domain since the initial condition $f(x)$ is concentrated around $x = x_0$.

Figure 9: Error of the numerical solutions for eq. (14) with exact solutions $u_1$ (left) and $u_2$ (right) in table 3.

Figure 10: The irregular domain used in eq. (15). The red dots represent the control points of the NURBS surface. There are 2209 control points in total.

We consider an implicit scheme for eq. (14). Let $A^\alpha$ be the coefficient matrix in eq. (13) (up to a constant $c_{2,\alpha}$), which maps the coefficients $c_i$ to the point-wise value of the numerical fractional Laplacian operator. In addition, we denote the interpolation matrix of the NURBS basis function by $R$, which maps the coefficients $c_i$ to the point-wise value $u(F(u))$ at the Greville abscissae in the physical domain. Therefore we have

$$\frac{Rc^{n+1} - Rc^n}{\Delta t} = -A^\alpha c^{n+1}$$

which leads to

$$c^{n+1} = (R + \Delta t A^\alpha)^{-1} Rc^n$$

December 21, 2018
The $R + \Delta t A^\alpha$ is factorized at the beginning to save time for later steps. The initial condition is obtained by

$$c^0 = R^{-1}f, f = f(x_i)$$

where $x_i = F(u_i)$ is the $i$-th collocation point. We choose $\Delta = 0.01$ and propagate 100 time steps. We show the solutions in fig. 11. The plots are generated by sampling the NURBS surface uniform on a $200 \times 200$ grid. We see for larger $\alpha$, the diffusion process is much faster than small $\alpha$. In the case $\alpha = 1.0$, we recover the standard diffusion. We note that an extra advantage of the iso-geometric analysis is that the numerical solution is automatically smooth.
Figure 11: Solution to eq. (15) at $T = 1$. $N$ is the number of control points in total. The plots are generated by sampling the NURBS surface uniform on a $200 \times 200$ grid. We see for larger $\alpha$, the diffusion process is much faster than small $\alpha$. In the case $\alpha = 1.0$, we recover the standard diffusion.
5. Conclusion

Thanks to the high order continuity of NURBS basis and iso-geometric analysis techniques, we can deal with the strong form of the integral fractional Laplacian Poisson problem using the collocation method. The window function idea borrowed from [21] reduces the principle value integration to the integral of a smooth function. That enables us to use efficient quadrature rules. The iso-geometric analysis works well with solutions with various continuity behaviors. We observed a $O(N^{-1})$ convergence rate for smooth solutions. Compared to the finite element method, which has convergence rate approximately $O(\sqrt{h})$ or $O(h)$ for graded mesh, our method enjoys higher convergence rate. Compared to uniform grid finite difference methods, our method has comparable convergence rates, and besides, the iso-geometric collocation method can deal with complex geometries.

However, we have also observed the current method cannot deal with $C^{0,\alpha}$ solutions, which has the minimal regularity that the fractional Poisson equation can have. This limitation is attributed to the high continuity of NURBS basis functions: it tries hard to fit the less continuous data on the boundary with highly continuous basis functions and thus creates some spurious oscillations. Another limitation is that the current method requires solving a less structured linear system than that of the uniform grid finite difference method, due to which fast algorithms such as FFT cannot be directly applied.

We also want to comment on the iso-geometric analysis. This method is powerful in that we can implement basis functions with arbitrary order continuity for their derivatives. Besides, we can deal with complex geometries. However, we do not make use of the ‘iso-geometry’ property of the method. In fact, it is fine we use NURBS space for the underlying geometry and other basis functions, e.g., B-splines, for the solution space.

The issues mentioned above, as well as the theoretical investigation, will be left to the future.
Appendices

The codes for the paper are available from the authors upon request.

Here, we briefly mention the detail of implementing high order derivative mappings in eq. (10).

Consider the basis functions

$$R_{ij}(x, y) = \frac{w_{ij}N_i(x)N_j(y)}{\sum_{ij} w_{ij}N_i(x)N_j(y)}$$

Since we need up to third-order derivatives, we need to compute

$$\partial_x R_{ij}, \partial_y R_{ij}, \partial_{xx} R_{ij}, \partial_{xy} R_{ij}, \partial_{yy} R_{ij}, \partial_{xxx} R_{ij}, \partial_{xxy} R_{ij}, \partial_{xyy} R_{ij}, \partial_{yyy} R_{ij}$$

We have the following formulas

In our implementation, let the geometric mapping be

$$x = F(u,v), y = G(u,v)$$

i.e., for any point \((u,v)\) in the parametric space, it is mapped to \((x,y)\) in the physical space. Any function defined on the physical space is related to the parametric space through

$$f(x,y) = f(F(u,v), G(u,v))$$

In eq. (10) we need to find formulas for

$$f_x, f_y, f_{xx}, f_{xy}, f_{yy}, f_{xxx}, f_{xxy}, f_{xyy}, f_{yyy}$$

in the physical space, where we replaced \(u\) by \(f\) to avoid confusion. However, we can only evaluate

$$f_u, f_v, f_{uu}, f_{uv}, f_{vv}, f_{uuu}, f_{uvu}, f_{vvv}$$

in the parametric space. The conversion can be done using chain rules, which are, for convenience, listed here.

$$\begin{bmatrix} f_u \\ f_v \end{bmatrix} = \begin{bmatrix} F_u & G_u \\ F_v & G_v \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

$$\begin{bmatrix} f_{uu} \\ f_{uv} \\ f_{vv} \end{bmatrix} = \begin{bmatrix} F_{uu} & G_{uu} \\ F_{uv} & G_{uv} \\ F_{vv} & G_{vv} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} + \begin{bmatrix} F_u^2 & 2F_uG_u & G_u^2 \\ F_uF_v & F_uG_v + F_vG_u & G_uG_v \\ F_v^2 & 2F_vG_v & G_v^2 \end{bmatrix} \begin{bmatrix} f_{xx} \\ f_{xy} \\ f_{yy} \end{bmatrix}$$
By solving these linear systems, we can obtain

\[
\begin{bmatrix}
    f_{xx} \\
    f_{xy} \\
    f_{yy}
\end{bmatrix}
= \begin{bmatrix}
    F_{uuu} & G_{uuu} \\
    F_{uuu} & G_{uuu} \\
    F_{vvv} & G_{vvv}
\end{bmatrix}
\begin{bmatrix}
    f_x \\
    f_y
\end{bmatrix}
\]

from

\[
\begin{bmatrix}
    f_u \\
    f_v \\
    f_{uu} \\
    f_{uv} \\
    f_{vv} \\
    f_{uuu} \\
    f_{uuv} \\
    f_{uvv} \\
    f_{vvv}
\end{bmatrix}
\]
References

[1] W. Chen. Soft Matter and Fractional Mathematics: Insights into Mesoscopic Quantum and Time-Space Structures. arXiv preprint cond-mat/0405345, 2004.
[2] P. Gatto and J. S. Hesthaven. Numerical Approximation of the Fractional Laplacian via hp-finite Elements, with an Application to Image Denoising. Journal of Scientific Computing, 65(1):249–270, 2015.
[3] W. A. Woyczyński. Lévy Processes in the Physical Sciences. Lévy Processes in the Physical Sciences, pages 241–266. Springer, 2001.
[4] G. M. Bisci, V. D. Radulescu, and R. Servadei. Variational methods for nonlocal fractional problems, volume 162. Cambridge University Press, 2016.
[5] H. Antil and C. N. Rautenberg. Sobolev Spaces with Non-Muckenhoupt Weights, Fractional Elliptic Operators, and Applications. arXiv preprint arXiv:1803.10350, 2018.
[6] C. Bucur and E. Valdinoci. Nonlocal diffusion and applications, volume 20. Springer, 2016.
[7] M. Ainsworth and C. Glusa. Hybrid Finite Element — Spectral Method for the Fractional Laplacian: Approximation Theory and Efficient Solver. ArXiv e-prints, September 2017.
[8] M. D’Elia and M. Gunzburger. The Fractional Laplacian Operator on Bounded Domains As A Special Case of the Nonlocal Diffusion Operator. Computers & Mathematics with Applications, 66(7):1245–1260, 2013.
[9] G. Acosta and J. P. Borthagaray. A Fractional Laplace Equation: Regularity of Solutions and Finite Element Approximations. SIAM Journal on Numerical Analysis, 55(2):472–495, 2017.
[10] M. Ainsworth and C. Glusa. Towards an Efficient Finite Element Method for the Integral Fractional Laplacian on Polygonal Domains. arXiv preprint arXiv:1708.01923, 2017.
[11] W. Chen and S. Holm. Fractional Laplacian, Lévy Stable Distribution, and Time-Space Models for Linear and Nonlinear Frequency-Dependent Lossy Media. arXiv preprint math-ph/0212075, 2002.
[12] P. Tankov. Financial modelling with jump processes, volume 2. CRC press, 2003.
[13] T. Grywny, M. Ryznar, et al. Estimates of Green Functions for Some Perturbations of Fractional Laplacian. Illinois Journal of Mathematics, 51(4):1409–1438, 2007.
[14] P. Graczyk, T. Jakubowski, and T. Luks. Martin Representation and Relative Fatou Theorem for Fractional Laplacian with A Gradient Perturbation. Positivity, 17(4):1043–1070, 2013.
[15] A. E. Kyprianou, A. Osojnik, and T. Shardlow. Unbiased “Walk-On-Spheres” Monte Carlo Methods for the Fractional Laplacian. IMA Journal of Numerical Analysis, 51(4):1409–1438, 2007.
[16] A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M. M. Meerschaert, M. Ainsworth, et al. What Is the Fractional Laplacian? arXiv preprint arXiv:1801.04767, 2018.
[17] M. M. Meerschaert, J. Mortensen, and H.-P. Scheffler. Vector Grünwald Formula for Fractional Derivatives. Fractional calculus and applied analysis, 7(1):61–82, 2004.
[18] Y. Huang and A. Oberman. Numerical Methods for the Fractional Laplacian: A Finite Difference-Quadature Approach. SIAM Journal on Numerical Analysis, 52(6):3056–3084, 2014.
[19] F. del Teso. Finite Difference Method for A Fractional Porous Medium Equation. Calcolo, 51(4):615–638, 2014.
[20] S. Duo, H. W. van Wyk, and Y. Zhang. A Novel and Accurate Finite Difference Method for the Fractional Laplacian and the Fractional Poisson Problem. Journal of Computational Physics, 355:233–252, 2018.
[21] V. Minden and L. Ying. AsimplyolverforthefractionalLaplacininmultipledimensions. ArXiv e-prints, February 2018.
[22] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement. Computer methods in applied mechanics and engineering, 194(39-41):4135–4195, 2005.
[23] M. Samuelčík. Bézier and B-Spline Volumes, 2006.
[24] F. Auricchio, L. B. Da Veiga, T. Hughes, A. Reali, and G. Sangalli. Isogeometric Collocation Methods. Mathematical Models and Methods in Applied Sciences, 20(11):2075–2107, 2010.
[25] L. B. da Veiga, C. Lovadina, and A. Reali. Avoiding Shear Locking for the Timoshenko Beam Problem

December 21, 2018
via Isogeometric Collocation Methods. *Computer Methods in Applied Mechanics and Engineering*, 241:38–51, 2012.

[26] B. Bojanov and G. Petrova. Numerical Integration Over A Disc. A New Gaussian Quadrature Formula. *Numerische Mathematik*, 80(1):39–59, 1998.

[27] A. Karjalainen et al. NURBS Toolbox for Matlab. 2008.

[28] A. H. Stroud. Approximate Calculation of Multiple Integrals. 1971.

[29] R. Cools and K. J. Kim. A Survey of Known and New Cubature Formulas for the Unit Disk. *Korean Journal of Computational & Applied Mathematics*, 7(3):477–485, 2000.

[30] A. H. Stroud and D. Secrest. Gaussian Quadrature Formulas. 1966.

[31] G. H. Golub and J. H. Welsch. Calculation of Gauss Quadrature Rules. *Mathematics of computation*, 23(106):221–230, 1969.

[32] W. Gautschi. *Orthogonal polynomials: computation and approximation*. Oxford University Press on Demand, 2004.

[33] I. Mysovskikh. Interpolatory Cubature Formulas, 1981.

[34] Cubature Formulas for the Unit Disk. [http://www.holoborodko.com/pavel/numerical-methods/numerical-integration/cubature-formulas-for-the-unit-disk/] (Accessed on 12/02/2018).

[35] O. CHANON et al. A Numerical Study of the Isogeometric Collocation Method. 2017.

[36] GNURBS - File Exchange - MATLAB Central. [https://www.mathworks.com/matlabcentral/fileexchange/14248-gnurbs](https://www.mathworks.com/matlabcentral/fileexchange/14248-gnurbs) (Accessed on 12/02/2018).

December 21, 2018