



Article An *h*-Adaptive Poly-Sinc-Based Local Discontinuous Galerkin Method for Elliptic Partial Differential Equations

Omar A. Khalil¹ and Gerd Baumann^{1,2,*}

- ¹ Mathematics Department, German University in Cairo, New Cairo City 11835, Cairo, Egypt
- ² Faculty of Natural Sciences, University of Ulm, D-89069 Ulm, Germany

* Correspondence: gerd.baumann@uni-ulm.de

Abstract: For the purpose of solving elliptic partial differential equations, we suggest a new approach using an *h*-adaptive local discontinuous Galerkin approximation based on Sinc points. The adaptive approach, which uses Poly-Sinc interpolation to achieve a predetermined level of approximation accuracy, is a local discontinuous Galerkin method. We developed an *a priori* error estimate and demonstrated the exponential convergence of the Poly-Sinc-based discontinuous Galerkin technique, as well as the adaptive piecewise Poly-Sinc method, for function approximation and ordinary differential equations. In this paper, we demonstrate the exponential convergence in the number of iterations of the *a priori* error estimate derived for the local discontinuous Galerkin technique under the condition that a reliable estimate of the precise solution of the partial differential equation at the Sinc points exists. For the purpose of refining the computational domain, we employ a statistical strategy. The numerical results for elliptic PDEs with Dirichlet and mixed Neumann-Dirichlet boundary conditions are demonstrated to validate the adaptive greedy Poly-Sinc approach.

Keywords: Poly-Sinc methods; Lagrange interpolation; local discontinuous Galerkin methods; adaptive methods; exponential convergence; elliptic partial differential equations; mean absolute deviation; sample standard deviation

MSC: 41A05; 41A10; 65D05; 65N50; 65N12

1. Introduction

Discontinuous Galerkin (DG) methods [1–5] are a class of numerical methods for finding accurate approximate solutions to differential equations. In a DG method, the domain of interest is partitioned into cells, which allows faster computation of the numerical solution of the partial differential equation (PDE) on the domain of interest. The partitioning process introduces local boundaries, leading to jumps at the interior edges of the cells. It was shown in [4] that the accuracy of the DG approximation is affected by these jumps. We use Poly-Sinc approximation [6-9], in which Lagrange interpolation with non-equidistant points generated by conformal mappings, such as Sinc points, are used to minimize the jumps at the interior edges of the cells. Another type of discontinuity is singularities. Common singularities appear in the form of corner-like or L-shaped domains [10–15]. Sinc methods are characterized by their fast convergence rate of the exponential order and effective handling of singularities [16]. The local DG (LDG) method [3,17,18] has been rigorously studied and applied to a wide variety of problems due to their local conservativity as well as ease of handling of hanging nodes, elements of various types and shapes, and local spaces of different orders [19]. For a review of the LDG method as well as the different forms of the DG method, the reader is referred to [18–22]. There has been an increased interest in developing adaptive DG methods for elliptic problems [22,23], compressible and incompressible flow problems [24–26], unsteady flow problems [27,28], and shallow water equations [29-31].



Citation: Khalil, O.A.; Baumann, G. An *h*-Adaptive Poly-Sinc-Based Local Discontinuous Galerkin Method for Elliptic Partial Differential Equations. *Axioms* **2023**, *12*, 227. https:// doi.org/10.3390/axioms12030227

Academic Editor: Clemente Cesarano

Received: 29 January 2023 Revised: 10 February 2023 Accepted: 17 February 2023 Published: 21 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). We extend our work on the DG method [32,33] and the adaptive piecewise collocation method [34,35] using the Poly-Sinc approximation to elliptic partial differential equations. It was shown in [6,7] that the *a priori* error estimate of the Poly-Sinc approximation is exponentially convergent in the number of Sinc points, provided that the exact solution belongs to the set of analytic functions.

We develop an adaptive LDG method based on Poly-Sinc approximation to solve elliptic problems. In [34,35], the fraction of a standard deviation was computed as the ratio of the mean absolute deviation to the sample standard deviation. It was shown in [36] that the ratio approaches $\sqrt{\frac{2}{\pi}} \approx 0.798$ for an infinite number of normal samples.

This paper is organized as follows. Section 2 discusses the approximation spaces. Section 3 provides a brief overview of the LDG method. We present the adaptive Poly-Sinc-based LDG algorithm, provide an *a priori* error estimate for the approximation error and show its convergence, and demonstrate the convergence of the adaptive algorithm in Section 4. We provide numerical examples to validate the adaptive algorithm in Section 5. Finally, our concluding remarks are given in Section 6.

2. Approximation Spaces

2.1. Univariate Approximation Spaces

We will introduce some notations related to Sinc methods [6,7,37,38]. Let φ : $D \rightarrow D_d$ be a conformal map that maps a simply connected region $D \subset \mathbb{C}$, where

$$D = \{z \in \mathbb{C} : |\arg\left(\frac{z-a}{b-z}\right)| < d\}, \ -\infty < a < b < \infty,$$

onto the region [37,39]

$$D_d = \{z \in \mathbb{C} : |\operatorname{Im}\{z\}| < d\},\$$

where *d* is a given positive number. The region *D* has a boundary ∂D , and let *a* and *b* be two distinct points on ∂D . Let $\psi = \varphi^{-1}, \psi : D_d \to D$ be the inverse conformal map. Let Γ be an arc defined by

$$\Gamma = \{ z \in [a,b] : z = \psi(x), x \in \mathbb{R} \},\$$

where $a = \psi(-\infty)$ and $b = \psi(\infty)$. For finite real numbers a, b, and $\Gamma \subseteq \mathbb{R}$, $\varphi(x) = \log((x-a)/(b-x))$ and $x_k = \psi(kh) = (a+be^{kh})/(1+e^{kh})$ are the Sinc points with spacing $h(d,\beta_s) = \left(\frac{\pi d}{\beta_s N}\right)^{1/2}$, $\beta_s > 0$ [38,40]. Sinc points can be also generated for semi-infinite or infinite intervals. For a comprehensive list of conformal maps, see [7,38].

We now briefly discuss the function space for y. Let $\rho = e^{\varphi}$, α_s be an arbitrary positive

integer number and $\mathbb{L}_{\alpha_s,\beta_s}(D)$ be the family of all functions that are analytic in $D = \varphi^{-1}(D_d)$ such that for all $z \in D$, we have

$$u(z)| \le C \frac{|\rho(z)|^{\alpha_{\mathsf{s}}}}{[1+|\rho(z)|]^{\alpha_{\mathsf{s}}+\beta_{\mathsf{s}}}}.$$

We next set restrictions on α_s , β_s , and d such that $0 < \alpha_s \le 1$, $0 < \beta_s \le 1$, and $0 < d < \pi$. Let $\mathbb{M}_{\alpha_s,\beta_s}(D)$ be the set of all functions g defined on D that have finite limits $g(a) = \lim_{z \to a} g(z)$ and $g(b) = \lim_{z \to b} g(z)$, where the limits are taken from within D such that $f \in \mathbb{L}_{\alpha_s,\beta_s}(D)$, where

$$f = g - \frac{g(a) + \rho g(b)}{1 + \rho}$$

The transformation guarantees that *f* vanishes at the endpoints of (a, b). We assume that *u* is analytic and uniformly bounded by $\mathcal{B}(u)$ (i.e., $|u(x)| \leq \mathcal{B}(u)$) in the larger region

$$D_2 = D \cup_{t \in (a,b)} B(t,r),$$

where r > 0 and $B(t, r) = \{z \in \mathbb{C} : |z - t| < r\}.$

Lagrange interpolation is a polynomial interpolation scheme [41] which is constructed by Lagrange basis polynomials such that

$$\ell_k(x) = \frac{g(x)}{(x - x_k)g'(x_k)}, \quad k = 1, 2, \dots, m,$$
(1)

where $\{x_k\}_{k=1}^m$ represents the interpolation points and $g(x) = \prod_{l=1}^m (x - x_l)$. The Lagrange basis polynomials satisfy the property

$$\ell_k(x_j) = \delta_{kj} = \begin{cases} 1, & \text{if } k = j \\ 0, & \text{if } k \neq j. \end{cases}$$
(2)

Hence, the polynomial approximation in the Lagrange form can be written as

$$u_h(x) = \sum_{k=1}^m u_k \ell_k(x),$$
 (3)

where $u_h(x)$ is a polynomial of the degree m - 1 and $u_k = u(x_k)$. The polynomial $u_h(x)$ interpolates the function u(x) at the interpolation points (i.e., $u_h(x_k) = u_k$). For the Sinc points, the polynomial approximation $u_h(x)$ becomes

$$u_{h}(x) = \sum_{k=-N}^{N} u_{k} \ell_{k}(x),$$
(4)

where m = 2N + 1 is the number of Sinc points.

2.2. Bivariate Approximation Spaces

We now define some notations for the function space $S_{\alpha,d}$ of the function u(x,y) [37,39,42,43]. Let X = (x,y) be a point in $Q = [a,b] \times [c,d]$, and let D be defined as in Section 2.1 with $d = \pi/2$. Let \overline{D} be the closure of D. Define Ω as ([37], §6.5.2):

$$\Omega = \{(a,b) \times \overline{D}\} \cup \{\overline{D} \times (c,d)\}.$$

Let $u : \Omega \to \mathbb{C}$, $u \in \mathbf{C}(\Omega)$. Define the following two functions:

$$u^x$$
 : $D \to \mathbb{C}$ such that $u^x(y) = u(x, y)$ for fixed $x \in [a, b]$,

 u^y : $D \to \mathbb{C}$ such that $u^y(x) = u(x, y)$ for fixed $y \in [c, d]$.

We note that $u^{x}(y) \in \mathbb{M}_{\alpha_{s,y},\beta_{s,y}}(D_{2,y})$ and $u^{y}(x) \in \mathbb{M}_{\alpha_{s,x},\beta_{s,x}}(D_{2,x})$ ([37], §4.6.4). Define the following space of functions $S_{\alpha,d}$ to be the space of all functions u(x, y) such that the following apply:

- For all $x \in [a, b]$, $u^x \in Hol(D)$;
- For all $y \in [c, d]$, $u^y \in Hol(D)$;
- There exists $\alpha \in (0, 1)$ such that the following apply:
 - For all $x \in [a, b]$, $u^x \in \operatorname{Lip}_{\alpha}(\overline{D})$;
 - for all $y \in [c,d]$, $u^y \in \operatorname{Lip}_{\alpha}(\overline{D})$.

Here, Hol(D) is the family of all functions *u* that are analytic in a domain *D*. A function *v* is said to be in a class Lip_{α} on a closed interval [a, b] if there exists a positive constant *C* such that [37]

$$|v(x_1) - v(x_2)| \le C |x_1 - x_2|^{\alpha},$$

for all points $x_1, x_2 \in [a, b]$. Define the following two Lagrange approximations:

$$(L_1 u)(x) = \sum_{j=-M_x}^{N_x} u^y(x_j) \ell_j(x)$$

and

$$(L_2 u)(y) = \sum_{k=-M_y}^{N_y} u^x(y_k)\ell_k(y),$$

where L_i , i = 1, 2, is the one-dimensional interpolation operator [37,39] and $m_x = M_x + N_x + 1$ and $m_y = M_y + N_y + 1$ are the number of interpolation points in the *x* axis and the *y* axis, respectively. Then, the bivariate Lagrange approximation for a function $u(x, y) \in S_{\alpha, d}$ can be written as

$$u_h(x,y) = L_1(L_2u)(x,y) = \sum_{j=-M_x}^{N_x} \sum_{k=-M_y}^{N_y} u(x_j, y_k) \ell_j(x) \ell_k(y),$$
(5)

where $\ell_j(x)$ and $\ell_k(y)$ are the Lagrange basis functions defined in Section 2.1 for the variables *x* and *y*, respectively. For simplicity, we assume that $M_x = N_x$ and $M_y = N_y$ throughout the remaining discussion.

The LDG approximation using the Poly-Sinc interpolation becomes [44]

$$u_{\rm c}(x,y) = \sum_{l=1}^{K} \mathbb{1}_{\Omega_l} \sum_{j=1}^{m_{\rm x}} \sum_{k=1}^{m_{\rm y}} c_{j,k,l} \ell_{j,l}(x) \ell_{k,l}(y), \tag{6}$$

where $\{c_{j,k,l}\}_{j,k,l=1}^{m_x,m_y,K}$ are the unknown coefficients to be determined by the LDG method. The function $\mathbb{1}_C$ is an indicator function which outputs one if the condition C is satisfied and zero otherwise.

3. A Brief Overview of the Local Discontinuous Galerkin Method

Consider the following elliptic equation [3,18]:

$$-\Delta u = f \qquad \text{in } \Omega,$$

$$u = g_{\mathcal{D}} \qquad \text{on } \Gamma_{\mathcal{D}},$$

$$\nabla u \cdot \boldsymbol{n} = \boldsymbol{g}_{\mathcal{N}} \cdot \boldsymbol{n} \qquad \text{on } \Gamma_{\mathcal{N}},$$
(7)

where f, g_D , and g_N are given functions. The computational domain Ω is a bounded domain in \mathbb{R}^2 , as discussed in Section 2.2, and n is the outward normal to its boundary $\Gamma = \Gamma_D \cup \Gamma_N$ [22]. The computational domain Ω is divided into a Cartesian grid \mathcal{T}_h which consists of K rectangular elements $\Omega_l = \Omega_{l,x} \times \Omega_{l,y}$, $l = 1, \ldots, K$ [22]. For ease of exposition, we will use the symbol Δ to denote the element Ω_l in this section. It is assumed that the one-dimensional measure of Γ_D is nonzero [3,18,22]. The model in Equation (7) can be written as a system of first-order equations:

$$q = \nabla u \qquad \text{in } \Omega,$$

$$-\nabla \cdot q = f \qquad \text{in } \Omega,$$

$$u = g_{\mathcal{D}} \qquad \text{on } \Gamma_{\mathcal{D}},$$

$$q \cdot n = g_{\mathcal{N}} \cdot n \qquad \text{on } \Gamma_{\mathcal{N}}.$$
(8)

The weak formulation of Equation (8) can be obtained in what follows. By multiplying the first and second equations in Equation (8) by the test functions v and w, respectively, integrating over an element $\Delta \in T_h$, and applying the divergence theorem [22], we obtain

$$\iint_{\Delta} \boldsymbol{q} \cdot \nabla v \, \mathrm{d}x \, \mathrm{d}y - \int_{\Gamma} \boldsymbol{q} \cdot \boldsymbol{n} \, v \, \mathrm{d}s = \iint_{\Delta} f \, v \, \mathrm{d}x \, \mathrm{d}y,$$

-
$$\iint_{\Delta} u \, \nabla \cdot \boldsymbol{w} \, \mathrm{d}x \, \mathrm{d}y + \int_{\Gamma} u \, \boldsymbol{w} \cdot \boldsymbol{n} \, \mathrm{d}s = \iint_{\Delta} \boldsymbol{q} \cdot \boldsymbol{w} \, \mathrm{d}x \, \mathrm{d}y.$$
 (9)

We replace the exact solution (q, u) with its LDG approximation (q_c , u_c) in the finite element space $V_c^p \times V_c^p$, where [3,22]

$$\mathbf{V}_{c}^{p} = \{ \boldsymbol{q} \in \left(L^{2}(\Omega) \right)^{2} : \boldsymbol{q} |_{\Delta} \in \mathcal{Q}^{p}(\Delta), \, \forall \Delta \in \mathcal{T}_{h} \}, \\
\mathbf{V}_{c}^{p} = \{ u \in L^{2}(\Omega) : u |_{\Delta} \in \left(\mathcal{Q}^{p}(\Delta) \right)^{2}, \, \forall \Delta \in \mathcal{T}_{h} \},$$
(10)

and where $Q^p(\Delta)$ is a tensor product space consisting of polynomials of a degree of at most p in each variable for the element Δ [22]. We look for approximate solutions $u_c \in V_c^p$ and $q_c = [q_1, q_2]^\top \in V_c^p$. For any test functions $v \in V_c^p$ and $w \in V_c^p$, the LDG method consists of finding $(q_c, u_c) \in V_c^p \times V_c^p$ [3,22] such that

$$\iint_{\Delta} q_{c} \cdot \nabla v \, dx \, dy - \int_{\Gamma} \hat{q}_{c} \cdot n \, v \, ds = \iint_{\Delta} f \, v \, dx \, dy \,, \quad \forall v \in V_{c}^{p} \,,$$

$$-\iint_{\Delta} u_{c} \, \nabla \cdot w \, dx \, dy + \int_{\Gamma} \hat{u}_{c} \, w \cdot n \, ds = \iint_{\Delta} q_{c} \cdot w \, dx \, dy \,, \quad \forall w \in V_{c}^{p} \,,$$
(11)

where \hat{u}_c and \hat{q}_c are numerical fluxes. The fluxes are nothing but discrete approximations to the traces of u and q on the boundary of the element Δ [3,18,22]. Throughout the remaining discussion, we will use the notations \hat{u} and \hat{q} . Let Δ^+ and Δ^- be two adjacent elements of the Cartesian grid \mathcal{T}_h [3,22]. Let x be an arbitrary point of the edge $\Gamma = \partial \Delta^+ \cap \partial \Delta^-$ and n^+ and n^- be the corresponding outward normal vectors at x. The mean values $\{\!\{\cdot\}\!\}$ and jumps $[\![\cdot]\!]$ at $x \in \Gamma$ are defined as [3,18]

$$\{\!\!\{u\}\!\!\} = (u^+ + u^-)/2, \qquad \{\!\!\{q\}\!\!\} = (q^+ + q^-)/2, \\ [\![u]\!] = u^+ n^+ + u^- n^-, \qquad [\![q]\!] = q^+ \cdot n^+ + q^- \cdot n^-,$$

where the superscript + denotes the cell Δ , the superscript – denotes an adjacent cell sharing the edge Γ , and $n^+ = -n^-$ [18,45]. If the edge Γ is inside the domain Ω , then the fluxes are [3]

$$\hat{q} = \{\!\!\{q\}\!\!\} - C_{11}[\![u]\!] - C_{12}[\![q]\!], \\ \hat{u} = \{\!\!\{u\}\!\} + C_{12} \cdot [\![u]\!],$$

where C_{11} and C_{12} are parameters that depend on $x \in \Gamma$. If the edge Γ lies on the boundary $\partial \Omega$, then the fluxes are [3]

$$\hat{q} = \begin{cases} q^+ - C_{11}(u^+ - g_{\mathcal{D}})n & \text{on } \Gamma_{\mathcal{D}}, \\ g_{\mathcal{N}} & \text{on } \Gamma_{\mathcal{N}} \end{cases} \text{ and } \hat{u} = \begin{cases} g_{\mathcal{D}} & \text{on } \Gamma_{\mathcal{D}}, \\ u^+ & \text{on } \Gamma_{\mathcal{N}}. \end{cases}$$

The stabilization parameter C_{11} and the auxiliary parameter C_{12} are defined on the edge Γ as

$$C_{11}(\Gamma) = \zeta > 0,$$
 $C_{12}(\Gamma) \cdot \boldsymbol{n} = \operatorname{sign}(\boldsymbol{v} \cdot \boldsymbol{n})/2,$

where v is an arbitrary vector with strictly positive components [45]. The vector v is used to select the fluxes for all elements [3]. For simplicity, we set $C_{11}(\Gamma) = 1$ and v = (1, 1). A simple rule for \hat{u} can be written as [3]

$$\hat{u} = \frac{1}{2}(1 + \operatorname{sign}(\boldsymbol{v} \cdot \boldsymbol{n}))u^+ + \frac{1}{2}(1 - \operatorname{sign}(\boldsymbol{v} \cdot \boldsymbol{n}))u^-$$

As mentioned in [3], \hat{u} is always equal to the left trace of u on the vertical edges, and \hat{u} is always equal to the trace of u from below on the horizontal edges. Similarly, $\{\!\{q\}\!\} - C_{12}[\![q]\!]$ is always equal to the right trace of q on the vertical edges, and $\{\!\{q\}\!\} - C_{12}[\![q]\!]$ is always equal to the trace of q from above on the horizontal edges. The quantities \hat{u} and $\{\!\{q\}\!\} - C_{12}[\![q]\!]$ are demonstrated in Figure 1.



Figure 1. Illustration of \hat{u} is shown in red, and illustration of $\{\!\{q\}\!\} - C_{12}[\!\{q\}\!]$ is shown in blue.

4. Adaptive Poly-Sinc-Based LDG Algorithm

The greedy algorithmic strategy utilized in adaptive Poly-Sinc techniques is introduced in this section. The non-overlapping characteristics and the uniform exponential convergence of the Sinc points on each cell of the computational domain serve as the major features of the adaptive algorithm. Greedy algorithms look for the "best" option among the possible solutions at the current iterate [46]. Model order reduction for the parametrized partial differential equations was performed using greedy methods [47–49]. The adaptive Poly-Sinc algorithm is greedy in that it makes a decision with the goal of locating the "best" approximation for the current step of the PDE's solution [46]. Iteratively, the technique computes the L^2 norm values of the local error for each cell of the computational domain. The algorithm refines the cells whose L^2 norm values of the local error will decrease with each step of the cell refinement described above. The approach anticipates locating the "best" polynomial approximation for the PDE's solution as the algorithm iterates.

4.1. Local Error

The local error is used as a measure of the accuracy of the adaptive Poly-Sinc-based LDG method. The local error can be expressed as

$$e(x,y) = u(x,y) - u_{c}(x,y).$$
(12)

The local error in the *l*th cell and *i*th iteration can be written as

$$v_{I}^{(i)}(x,y) = u(x,y) - u_{c}^{(i)}(x,y),$$
(13)

where $u_{c}(x, y)$, defined in Equation (6), for the *i*th iteration becomes

$$u_{\mathbf{c}}^{(i)}(x,y) = \sum_{l=1}^{K_i} \mathbb{1}_{\Omega_l^{(i)}} \sum_{j=1}^{m_{\mathbf{x}}} \sum_{k=1}^{m_{\mathbf{y}}} c_{j,k,l}^{(i)} \ell_{j,l}^{(i)}(x) \ell_{k,l}^{(i)}(y),$$
(14)

where K_i is the number of cells in the *i*th iteration.

4.2. Algorithm Description

We now discuss the adaptive algorithm for the Poly-Sinc-based LDG approximation. The following steps of the adaptive algorithm are performed in an iterative loop [50]:

$$SOLVE \rightarrow ESTIMATE \rightarrow MARK \rightarrow REFINE$$

The adaptive Poly-Sinc-based LDG algorithm is outlined in Algorithm 1. The refinement strategy is performed as follows. For the *i*th iteration, we compute the set of L^2 norm values $\left\{ \left\| e_l^{(i)}(x,y) \right\|_{L^2(\Omega_l^{(i)})} \right\}_{l=1}^{K_i}$, where $e_l^{(i)}(x,y)$ is defined in Section 4.1, over the K_i cells, from which the sample mean $\xi_i = \frac{1}{K_i} \sum_{l=1}^{K_i} \left\| e_l^{(i)}(x,y) \right\|_{L^2(\Omega_l^{(i)})}$ and the sample standard deviation [51]

$$s_{i} = \frac{1}{\sqrt{K_{i}-1}} \sqrt{\sum_{l=1}^{K_{i}}} \left(\left\| e_{l}^{(i)}(x,y) \right\|_{L^{2}(\Omega_{l}^{(i)})} - \xi_{i} \right)^{2}$$

are computed [52–54]. The symbol $\Omega_l^{(i)}$ denotes the *l*th cell in the *i*th iteration. The cells with the indices $\Im_i = \left\{ j : \left\| e_j^{(i)}(x, y) \right\|_{L^2(\Omega_j^{(i)})} - \xi_i \ge \omega_i s_i \right\}$ are marked for refinement, where the test of normality ω_i is defined as [36]

$$\omega_{i} = \frac{\frac{1}{K_{i}} \sum_{l=1}^{K_{i}} \left| \left\| e_{l}^{(i)}(x, y) \right\|_{L^{2}(\Omega_{l}^{(i)})} - \xi_{i} \right|}{s_{i}}.$$
(15)

Using Hölder's inequality for sums with p = q = 2 ([55], §3.2.8), one can show that $\omega_i \leq \sqrt{\frac{K_i-1}{K_i}} < 1$. We restrict this to second-order moments only. The points in the cells with the indices \Im_i are used as partitioning points, and $(2N + 1)^2$ Sinc points are inserted in the newly created cells, where 2N + 1 is the number of Sinc points in the *x* axis or the *y* axis. The algorithm terminates when the stopping criterion is satisfied. The approximate solution $u_c^{(i)}(x, y), i = 1, ..., \kappa$, for the *i*th iteration is computed using Equation (14).

The L^2 norm of a two-dimensional function f(x, y) is computed as follows:

$$\|f(x,y)\|_{L^{2}(\Omega)}^{2} = \left(\iint_{\Omega} f^{2}(x,y) \,\mathrm{d}x \,\mathrm{d}y\right)^{1/2},\tag{16}$$

where the definite integral is computed numerically using a Sinc quadrature [38]; in other words, we have

$$\int_{a}^{b} f(x) \,\mathrm{d}x \approx h_{\mathrm{Q}} \sum_{k=-N}^{N} \frac{1}{\varphi'(x_{k})} f(x_{k}),\tag{17}$$

where h_Q is the Sinc spacing used for the Sinc quadrature, $\{x_k\}_{k=-N}^N \in [a, b]$ represents the quadrature points, which are also Sinc points, and $\varphi(x)$ is the conformal mapping in Section 2.1.

Algorithm 1: Adaptive Poly-Sinc-based LDG algorithm. **input** :C, threshold $\varepsilon_{\text{stop}}$, N output:S: set of points κ : number of iterations $e = {\xi_i}_{i=1}^{\kappa}$: set of mean values of $\|e_l^{(i)}(x,y)\|_{L^2(\Omega^{(i)})}, l = 1, \dots, K_i, i = 1, \dots, \kappa$ $u_{c}^{(\kappa)}(x,y)$: approximate solution 1 init: $S = \{\}, C = \{\Omega_l^{(1)}\}_{l=1}^{K_1=1}, e = \{\}$ Start with $(2N + 1)^2$ Sinc points in each of the cells in C. Append the points to 2 S. (Solve). Compute $u_c^{(1)}(x, y)$ (Section 4.1). (Estimate). Compute $\{\|e_l^{(1)}(x,y)\|\}_{l=1}^{K_1=1}$ over the domain Ω . Append the mean 3 value ξ_1 in e. (Mark). Set $\mathcal{I}_1 = \{1\}$. 4 (**Refine**). Use the points in the cells with index set \mathcal{I}_1 as partitioning points. 5 Insert $(2N + 1)^2$ Sinc points in each of the newly created cells. Update S and C. Set i = 2. 6 while $\xi_{i-1} > \varepsilon_{\text{stop}}$ do 7 (Solve). Compute $u_c^{(i)}(x, y)$ (Section 4.1). 8 (Estimate). Compute $\left\{ \left\| e_l^{(i)}(x,y) \right\|_{L^2(\Omega_l^{(i)})} \right\}_{l=1}^{K_i}$ over K_i cells. Compute the 9 sample mean value ξ_i and the sample standard deviation s_i . Append ξ_i in e. (Mark). Identify the cells with indices 10 $\mathfrak{I}_{i} = \left\{ j : \left\| e_{j}^{(i)}(x,y) \right\|_{L^{2}(\Omega_{i}^{(i)})} - \xi_{i} \ge \omega_{i} s_{i} \right\}, \ l = 1, \dots, K_{i}.$ (**Refine**). Use the $(2N+1)^2$ points in the cells with indices \mathfrak{J}_i as partitioning 11 points. Insert $(2N + 1)^2$ Sinc points in each of the newly created cells. Update S and C. $i \leftarrow i + 1$. 12 13 end 14 $\kappa \leftarrow i - 1$.

4.3. Error Analysis

We state below the main theorem:

Theorem 1 (Estimate of the upper bound). Let u be analytic and bounded in Ω , and let $u_c^{(i)}(x, y)$ be the Poly-Sinc-based local discontinuous Galerkin approximation in the ith iteration. Let m_x and m_y be the number of Sinc points in the x axis and y axis, respectively. Let Λ be the Lebesgue constant. Let $c_{j,k,l}^{(i)}$ be the estimated coefficients and $u_{j,k,l}^{(i)}$ be the exact values of u(x, y) at the Sinc points $\{(x_{i,l}^{(i)}, y_{k,l}^{(i)})\}$ with

$$\sum_{j,k,l} |c_{j,k,l}^{(i)} - u_{j,k,l}^{(i)}| < \delta, \qquad i = 1, 2, \dots, \kappa$$

Then, there exists a constant A, independent of the ith iteration, such that

$$\max_{(x,y)\in\Omega} \left| u(x,y) - u_{\rm c}^{(i)}(x,y) \right| \le K_i E_i + \delta \Lambda_{m_{\rm x}} \Lambda_{m_{\rm y}},\tag{18}$$

where
$$E_i = \frac{A}{(2r)^m} \lambda^{m(i-1)} [(b-a)^{m_x} + \Lambda_{m_x} (d-c)^{m_y}], r > 1, and 0 < \lambda < 1.$$

Proof. The derivation follows that of [34,35]. The upper bound on the error estimate $u(x,y) - u_c^{(i)}(x,y)$ is

$$\begin{split} \max_{(x,y) \in \Omega} \left| u(x,y) - u_{c}^{(i)}(x,y) \right| &= \left\| u(x,y) - u_{c}^{(i)}(x,y) \right\|_{\Omega} \\ &= \left\| u(x,y) - u_{c}^{(i)}(x,y) + u_{h}^{(i)}(x,y) - u_{h}^{(i)}(x,y) \right\|_{\Omega} \\ &\leq \left\| u(x,y) - u_{h}^{(i)}(x,y) \right\|_{\Omega} + \left\| u_{h}^{(i)}(x,y) - u_{c}^{(i)}(x,y) \right\|_{\Omega} \\ &\leq K_{i} E_{i} + \sum_{l=1}^{K_{i}} \left\| u_{h}^{(i)}(x,y) - u_{c}^{(i)}(x,y) \right\|_{\Omega_{l}^{(i)}} \\ &= K_{i} E_{i} + \sum_{l=1}^{K_{i}} \left\| u_{h}^{m_{x}} \sum_{j=1}^{m_{y}} (c_{j,k,l}^{(i)} - u_{j,k,l}^{(i)}) e_{j,l}^{(i)}(x) e_{k,l}^{(i)}(y) \right\|_{\Omega_{l}^{(i)}} \\ &\leq K_{i} E_{i} + \sum_{l=1}^{K_{i}} \left\| \sum_{j=1}^{m_{x}} \sum_{k=1}^{m_{y}} |c_{j,l}^{(i)} - u_{j,k,l}^{(i)}| |e_{k,l}^{(i)}(y)| \|_{\Omega_{l}^{(i)}} \\ &\leq K_{i} E_{i} + \frac{\delta}{K_{i}} \sum_{l=1}^{K_{i}} \left\| \sum_{j=1}^{m_{x}} \sum_{k=1}^{m_{y}} |e_{j,l}^{(i)}(x)| e_{k,l}^{(i)}(y)| \|_{\Omega_{l}^{(i)}} \\ &\leq K_{i} E_{i} + \frac{\delta}{K_{i}} \sum_{l=1}^{K_{i}} \left\| \sum_{j=1}^{m_{x}} |e_{j,l}^{(i)}(x)| |e_{k,l}^{(i)}(y)| \|_{\Omega_{l}^{(i)}} \\ &\leq K_{i} E_{i} + \frac{\delta}{K_{i}} \sum_{l=1}^{K_{i}} \left\| \sum_{j=1}^{m_{x}} |e_{j,l}^{(i)}(x)| |e_{k,l}^{(i)}(y)| \|_{\Omega_{l}^{(i)}} \\ &\leq K_{i} E_{i} + \frac{\delta}{K_{i}} \sum_{l=1}^{K_{i}} \left\| \sum_{j=1}^{m_{x}} |e_{j,l}^{(i)}(x)| |e_{k,l}^{(i)}(y)| \|_{\Omega_{l}^{(i)}} \\ &\leq K_{i} E_{i} + \frac{\delta}{K_{i}} \sum_{l=1}^{K_{i}} \left\| \sum_{j=1}^{M_{x}} |e_{j,l}^{(i)}(x)| |e_{k,l}^{(i)}(y)| \|_{\Omega_{l}^{(i)}} \\ &\leq K_{i} E_{i} + \frac{\delta}{K_{i}} \sum_{l=1}^{K_{i}} \left\| \sum_{j=1}^{M_{x}} |e_{j,l}^{(i)}(x)| |e_{k,l}^{(i)}(y)| \|_{\Omega_{l}^{(i)}} \\ &\leq K_{i} E_{i} + \frac{\delta}{K_{i}} \sum_{l=1}^{K_{i}} \left\| \sum_{j=1}^{M_{x}} |e_{j,l}^{(i)}(x)| |e_{k,l}^{(i)}(y)| \|_{\Omega_{l}^{(i)}} \\ &\leq K_{i} E_{i} + \frac{\delta}{K_{i}} K_{i} \Lambda_{m_{x}} \Lambda_{m_{y}} \\ &\leq K_{i} E_{i} + \delta_{n_{x}} \Lambda_{m_{y}} \\ &\leq K_{i} E_{i} \\ &\leq K_{i} E_{i} \\ &\leq K_{i} E_{i} \\ &\leq K_{i} E$$

where $\Omega_l^{(i)} = \Omega_{x,l}^{(i)} \times \Omega_{y,l}^{(i)}$, $\Lambda_m = \frac{1}{\pi} \log(m) + 1.07618$ is the Lebesgue constant for the Poly-Sinc approximation [9,39,56], $m_x = \max_l m_{x,l}$, $m_y = \max_l m_{y,l}$, $\Lambda_{m_x} = \max_l \Lambda_{m_{x,l}}$, $\Lambda_{m_y} = \max_l \Lambda_{m_{y,l}}$, and the multiplicative property [57] of the supremum norm was used (i.e., $||xy|| \le ||x|| ||y||$). The iteration index *i* was dropped in $\Lambda_{m_{x,l}}^{(i)}$ and $\Lambda_{m_{y,l}}^{(i)}$ since the number of interpolation points m_x or m_y is independent of the *i*th iteration. On average, the term $|c_{j,k,l}^{(i)} - u_{j,k,l}^{(i)}| < \frac{\delta}{n_i} = \frac{\delta}{K_i m_x m_y} < \frac{\delta}{K_i}$, where $n_i = K_i m_x m_y$ is the total number of interpolation points in the *i*th iteration. Next, we derive the upper bound E_i , which follows the derivation of [37,39]. We start with the computational domain Ω . An upper bound on the error estimate $||u - u_h||$ becomes

$$\begin{aligned} \|u(x,y) - u_h^{(i)}(x,y)\|_{\Omega} &= \|u(x,y) - L_1(L_2(u^{(i)}))\|_{\Omega} \\ &= \|u(x,y) - L_1(L_2(u^{(i)})) + L_1(u^{(i)}) - L_1(u^{(i)})\|_{\Omega} \\ &\leq \|u(x,y) - L_1(u^{(i)})\|_{\Omega} + \|L_1(u^{(i)}) - L_1(L_2(u^{(i)}))\|_{\Omega}. \end{aligned}$$

(19)

The upper bound of the error $u(x, y) - L_1(u^{(i)})$ is

$$\begin{aligned} \|u(x,y) - L_{1}(u^{(i)})\|_{\Omega} &= \max_{(x,y) \in \Omega} \left| u(x,y) - L_{1}(u^{(i)}) \right| \\ &= \max_{(x,y) \in \Omega} \left| u(x,y) - \sum_{j=1}^{m_{x}} u(x_{j}^{(i)},y)\ell_{j}^{(i)}(x) \right| \\ &\leq \max_{x \in [a,b]} \left| u(x,y^{*}) - \sum_{j=1}^{m_{x}} u(x_{j}^{(i)},y^{*})\ell_{j}^{(i)}(x) \right| \\ &\leq \sum_{l=1}^{K_{i}} \max_{x \in \Omega_{x,l}} \left| u(x,y^{*}) - \sum_{j=1}^{m_{x}} u(x_{j,l}^{(i)},y^{*})\ell_{j,l}^{(i)}(x) \right| \\ &\leq \sum_{l=1}^{K_{i}} \frac{A_{x,l}^{(i)}}{(2r_{x,l}^{(i)})^{m_{x}}} (\mathsf{L}(\Omega_{x,l}^{(i)}))^{m_{x}} N_{x,l}^{(i)} \exp\left(-\gamma_{l,x}^{(i)}(N_{x,l}^{(i)})\beta_{l,x}^{(i)}\right) \\ &\leq \frac{A_{x}}{(2r_{x})^{m_{x}}} (\mathsf{L}(\Omega_{x}^{(i)}))^{m_{x}} \sum_{l=1}^{K_{i}} 1 \\ &= K_{i} \frac{A_{x}}{(2r_{x})^{m_{x}}} (\lambda_{x})^{m_{x}(i-1)} (b-a)^{m_{x}} \end{aligned}$$

where $\Omega_{l} = \Omega_{x,l} \times \Omega_{y,l}, A_{x} = \max_{l,i} A_{x,l}^{(i)} N_{x,l}^{(i)} \exp\left(-\gamma_{l,x}^{(i)}(N_{x,l}^{(i)})^{\beta_{l,x}^{(i)}}\right), r_{x} = \min_{l,i} r_{x,l}^{(i)},$ $L(\Omega_{x}^{(i)}) = \max_{l} L(\Omega_{x,l}^{(i)}), L(\Omega_{x}^{(i)}) = \lambda_{x}^{i-1}(b-a)$ [35], where $L(\cdot)$ denotes the length, and

$$y^* = \left\{ y : \left| u(x,y) - \sum_{j=1}^{m_x} u(x_j^{(i)},y) \ell_j^{(i)}(x) \right| \text{ is maximum } \forall x \in [a,b] \text{ and } \forall i \right\}.$$

The upper bound of the error $L_1(u^{(i)}) - L_1(L_2(u^{(i)}))$ is

$$\begin{split} \|L_{1}(u^{(i)}) - L_{1}(L_{2}(u^{(i)}))\|_{\Omega} &= \|L_{1}(u^{(i)} - L_{2}(u^{(i)}))\|_{\Omega} \\ &= \max_{(x,y) \in \Omega} \left| \sum_{j=1}^{m_{x}} \left[u(x_{j}^{(i)}, y) - \sum_{k=1}^{m_{y}} u(x_{j}^{(i)}, y_{k}^{(i)}) \ell_{k}^{(i)}(y) \right] \ell_{j}^{(i)}(x) \right| \\ &\leq \max_{y \in [c,d]} \left| u(x^{*}, y) - \sum_{k=1}^{m_{y}} u(x^{*}, y_{k}^{(i)}) \ell_{k}^{(i)}(y) \right| \underbrace{\max_{x \in [a,b]} \sum_{j=1}^{m_{x}} \left| \ell_{j}^{(i)}(x) \right|}_{\Delta_{m_{x}}} \\ &\leq \Lambda_{m_{x}} \sum_{l=1}^{K_{i}} \max_{y \in \Omega_{y,l}} \left| u(x^{*}, y) - \sum_{k=1}^{m_{y}} u(x^{*}, y_{k,l}^{(i)}) \ell_{k,l}^{(i)}(y) \right| \\ &\leq \Lambda_{m_{x}} \sum_{l=1}^{K_{i}} \frac{A_{y,l}^{(i)}}{(2r_{y,l}^{(i)})^{m_{y}}} (\mathsf{L}(\Omega_{y,l}^{(i)}))^{m_{y}} N_{y,l}^{(i)} \exp\left(-\gamma_{l,y}^{(i)}(N_{y,l}^{(i)}) \beta_{l,y}^{(i)}\right) \\ &\leq \Lambda_{m_{x}} \frac{A_{y}}{(2r_{y})^{m_{y}}} (\mathsf{L}(\Omega_{y}^{(i)}))^{m_{y}} \sum_{l=1}^{K_{i}} 1 \\ &= K_{i} \Lambda_{m_{x}} \frac{A_{y}}{(2r_{y})^{m_{y}}} (\lambda_{y})^{m_{y}(i-1)} (d-c)^{m_{y}}, \end{split}$$

$$x^* = \left\{ x : \left| u(x,y) - \sum_{k=1}^{m_y} u(x,y_k^{(i)}) \ell_k^{(i)}(y) \right| \text{ is maximum } \forall y \in [c,d] \text{ and } \forall i \right\}.$$

From the inequalities in Equations (20) and (21), the upper bound on the error estimate $u - u_h$ becomes

$$\begin{split} \|u(x,y) - u_{h}^{(i)}(x,y)\|_{\Omega} &\leq \|u(x,y) - u_{h}^{(i)}(x,y)\|_{\Omega} \\ &= \|u(x,y) - L_{1}(L_{2}(u^{(i)})) + L_{1}(u^{(i)}) - L_{1}(u^{(i)})\|_{\Omega} \\ &\leq \|u(x,y) - L_{1}(u^{(i)})\|_{\Omega} + \|L_{1}(u^{(i)}) - L_{1}(L_{2}(u^{(i)}))\|_{\Omega} \\ &\leq K_{i} \frac{A_{x}}{(2r_{x})^{m_{x}}} \lambda_{x}^{m_{x}(i-1)}(b-a)^{m_{x}} + K_{i} \Lambda_{m_{x}} \frac{A_{y}}{(2r_{y})^{m_{y}}} \lambda_{y}^{m_{y}(i-1)}(d-c)^{m_{y}} \\ &\leq K_{i} \frac{A}{(2r)^{m}} \lambda^{m(i-1)}[(b-a)^{m_{x}} + \Lambda_{m_{x}}(d-c)^{m_{y}}] \\ &= K_{i} E_{i}, \end{split}$$

where $m = \min\{m_x, m_y\}$, $r = \min\{r_x, r_y\}$, $\lambda = \max\{\lambda_x, \lambda_y\}$, $A = \max\{A_x, A_y\}$, and

$$E_i = \frac{A}{(2r)^m} \lambda^{m(i-1)} [(b-a)^{m_x} + \Lambda_{m_x} (d-c)^{m_x}].$$
(22)

The inequality in Equation (19) becomes

 $L(\Omega_{y}^{(i)}) = \lambda_{y}^{i-1}(d-c)$ [35], and

$$\max_{(x,y)\in\Omega} \left| u(x,y) - u_{\rm c}^{(i)}(x,y) \right| \leq K_i E_i + \delta \Lambda_{m_{\rm x}} \Lambda_{m_{\rm y}}.$$

We compute the mean value of the error estimate; in other words, we calculate

$$\max_{(x,y)\in\Omega}\frac{1}{K_i}\left|u(x,y)-u_{\mathsf{c}}^{(i)}(x,y)\right|\leq E_i+\frac{\delta}{K_i}\Lambda_{m_x}\Lambda_{m_y}.$$
(23)

For fitting purposes, we can rewrite the inequality in Equation (23) as

$$\max_{(x,y)\in\Omega} \frac{1}{K_i} \left| u(x,y) - u_{\rm c}^{(i)}(x,y) \right| \le E_i + \frac{\delta}{K_i} \Lambda_{m_{\rm x}} \Lambda_{m_{\rm y}}$$

$$= E_i + \tilde{\delta} \Lambda_{m_{\rm x}} \Lambda_{m_{\rm y}},$$
(24)

where $\tilde{\delta} = \frac{\delta}{K_i}$. We note that K_i is increasing and K_i is finite for a finite value of the iteration index *i*. The factor $\tilde{\delta}$ becomes small as K_i , the number of cells in the *i*th iteration, becomes large.

4.4. Convergence Analysis of the Adaptive Algorithm

This section discusses the convergence of the adaptive algorithm. The convergence analysis for the one-dimensional case was shown in [35]. It suffices to show that the sequence $\{u_c^{(i)}(x, y)\}$ is contractive and that it converges uniformly on Ω to u(x, y).

Lemma 1 (Contraction mapping [58]). *The sequence* $\{u_c^{(i)}(x, y)\}$ *is contractive; in other words, there exists some constant* 0 < C < 1 *such that*

$$\|u_{c}^{(i+1)}(x,y) - u_{c}^{(i)}(x,y)\|_{\Omega} \le C \|u_{c}^{(i)}(x,y) - u_{c}^{(i-1)}(x,y)\|_{\Omega}.$$

Proof. Let $K_{i+1,i,i-1} = K_{i+1}K_iK_{i-1}$. Recalling the inequality in Equation (23), we have

$$\frac{1}{K_{i+1,i,i-1}} \| u_{c}^{(i+1)}(x,y) - u_{c}^{(i)}(x,y) \|_{\Omega} = \frac{1}{K_{i+1,i,i-1}} \| u_{c}^{(i+1)}(x,y) - u_{c}^{(i)}(x,y) + u(x,y) - u(x,y) \|_{\Omega}
\leq \frac{1}{K_{i+1,i,i-1}} \| u(x,y) - u_{c}^{(i+1)}(x,y) \|_{\Omega} + \frac{1}{K_{i+1,i,i-1}} \| u(x,y) - u_{c}^{(i)}(x,y) \|_{\Omega}
< \frac{1}{K_{i+1}} \| u(x,y) - u_{c}^{(i+1)}(x,y) \|_{\Omega} + \frac{1}{K_{i}} \| u(x,y) - u_{c}^{(i)}(x,y) \|_{\Omega}
\leq E_{i+1} + \frac{\delta}{K_{i+1}} \Lambda_{m_{x}} \Lambda_{m_{y}} + E_{i} + \frac{\delta}{K_{i}} \Lambda_{m_{x}} \Lambda_{m_{y}}
= E_{i+1} + E_{i} + \delta \Lambda_{m_{x}} \Lambda_{m_{y}} \left(\frac{1}{K_{i+1}} + \frac{1}{K_{i}}\right)$$
(25)

Similarly, we have

$$\frac{1}{K_{i+1,i,i-1}} \left\| u_{c}^{(i)}(x,y) - u_{c}^{(i-1)}(x,y) \right\|_{\Omega} \le E_{i} + E_{i-1} + \delta \Lambda_{m_{x}} \Lambda_{m_{y}} \left(\frac{1}{K_{i}} + \frac{1}{K_{i-1}} \right)$$
(26)

We compare the ratio of the upper bounds in Equations (25) and (26); in other words, we have $E_{i} = E_{i} + E_{i} +$

$$\theta = \frac{E_{i+1} + E_i + \delta \Lambda_{m_x} \Lambda_{m_y} \left(\frac{1}{K_{i+1}} + \frac{1}{K_i}\right)}{E_i + E_{i-1} + \delta \Lambda_{m_x} \Lambda_{m_y} \left(\frac{1}{K_i} + \frac{1}{K_{i-1}}\right)}.$$

Since $E_i < E_{i-1}$, $i = 2, ..., \kappa$ from Equation (22) and $K_i > K_{i-1}$, $i = 2, ..., \kappa$, one can directly verify that

$$E_{i+1} + E_i + \delta \Lambda_{m_x} \Lambda_{m_y} \left(\frac{1}{K_{i+1}} + \frac{1}{K_i} \right) < E_i + E_{i-1} + \delta \Lambda_{m_x} \Lambda_{m_y} \left(\frac{1}{K_i} + \frac{1}{K_{i-1}} \right),$$

In other words, $\theta < 1$. This implies that there is a constant C > 0 such that

$$\frac{1}{K_{i+1,i,i-1}} \left\| u_{c}^{(i+1)}(x,y) - u_{c}^{(i)}(x,y) \right\|_{\Omega} \le \frac{C}{K_{i+1,i,i-1}} \left\| u_{c}^{(i)}(x,y) - u_{c}^{(i-1)}(x,y) \right\|_{\Omega}.$$
 (27)

Multiplying the inequality in Equation (26) by the constant *C* yields

$$\frac{C}{K_{i+1,i,i-1}} \left\| u_{c}^{(i)}(x,y) - u_{c}^{(i-1)}(x,y) \right\|_{\Omega} \le C \left(E_{i} + E_{i-1} + \delta \Lambda_{m_{x}} \Lambda_{m_{y}} \left(\frac{1}{K_{i}} + \frac{1}{K_{i-1}} \right) \right).$$
(28)

From the inequalities in Equations (27) and (28), we have

$$\frac{1}{K_{i+1,i,i-1}} \left\| u_{c}^{(i+1)}(x,y) - u_{c}^{(i)}(x,y) \right\|_{\Omega} \le C \left(E_{i} + E_{i-1} + \delta \Lambda_{m_{x}} \Lambda_{m_{y}} \left(\frac{1}{K_{i}} + \frac{1}{K_{i-1}} \right) \right).$$
(29)

The upper bound of the inequality in Equation (29) cannot be larger than that of Equation (25); in other words, we have

$$C\left(E_i + E_{i-1} + \delta \Lambda_{m_x} \Lambda_{m_y} \left(\frac{1}{K_i} + \frac{1}{K_{i-1}}\right)\right) \le E_{i+1} + E_i + \delta \Lambda_{m_x} \Lambda_{m_y} \left(\frac{1}{K_{i+1}} + \frac{1}{K_i}\right)$$

This implies that $C \le \theta < 1$. Hence, there exists a constant $0 < C \le \theta < 1$ such that

$$\frac{1}{K_{i+1,i,i-1}} \left\| u_{c}^{(i+1)}(x,y) - u_{c}^{(i)}(x,y) \right\|_{\Omega} \le \frac{C}{K_{i+1,i,i-1}} \left\| u_{c}^{(i)}(x,y) - u_{c}^{(i-1)}(x,y) \right\|_{\Omega}.$$
 (30)

Multiplying the inequality in Equation (30) by $K_{i+1,i,i-1}$, we obtain

$$\|u_{c}^{(i+1)}(x,y)-u_{c}^{(i)}(x,y)\|_{\Omega} \leq C \|u_{c}^{(i)}(x,y)-u_{c}^{(i-1)}(x,y)\|_{\Omega}.$$

Lemma 2 (Uniform convergence [37,58]). Let $\{u_c^{(i)}(x,y)\}, i = 1, 2..., be the sequence of bounded functions on <math>\Omega \subseteq \mathbb{R}^2$. If $\{u_c^{(i)}(x,y)\}$ forms a Cauchy sequence on Ω , then

$$\lim_{M\to\infty}\left\{\sup_{j>M,k>M}\left\|u_{c}^{(j)}(x,y)-u_{c}^{(k)}(x,y)\right\|_{\Omega}\right\}=0,$$

and the sequence $\{u_{c}^{(i)}(x, y)\}$ converges uniformly on Ω to u(x, y).

Proof. The derivation follows that of ([58], §8.1.10). Since the sequence $\{u_c^{(i)}(x,y)\}$ is contractive by Lemma 1, then the sequence $\{u_c^{(i)}(x,y)\}$ is a Cauchy sequence ([58], Theorem 3.5.8). Hence, there exists a natural number M such that if j, k > M, then $\sup_{j>M,k>M} \|u_c^{(j)}(x,y) - u_c^{(k)}(x,y)\|_{\Omega} \le \epsilon$. Since ϵ is arbitrary, $\lim_{M\to\infty} \{\sup_{j>M,k>M} \|u_c^{(j)}(x,y) - u_c^{(k)}(x,y)\|_{\Omega}\} = 0$. Hence, the sequence $\{u_c^{(i)}(x,y)\}$ converges to a limit $u(x,y), (x,y) \in \Omega$. Let $u : \Omega \to \mathbb{R}$, where

$$u(x,y) \coloneqq \lim_{i \to \infty} u_{c}^{(i)}(x,y), \quad (x,y) \in \Omega.$$

It was shown in Theorem 1 that $\max_{(x,y) \in \Omega} \frac{1}{K_i} \left| u(x,y) - u_c^{(i)}(x,y) \right| < E_i + \frac{\delta}{K_i} \Lambda_{m_x} \Lambda_{m_y}$ for $(x,y) \in \Omega$. Since $0 < \lambda < 1$, then $E_i \to 0$ as $i \to \infty$. The term $\frac{\delta}{K_i} \Lambda_{m_x} \Lambda_{m_y} \to 0$ as $i \to \infty$ since $K_i > 1$. We conclude that $\lim_{i\to\infty} u_c^{(i)}(x,y) = u(x,y)$, $(x,y) \in \Omega$, and the sequence $\{u_c^{(i)}(x,y)\}$ converges uniformly on Ω to u(x,y), which completes the proof. \Box

Lemmas (1) and (2) are used to prove the contraction mapping principle [37]. We consider the Banach space $\mathbf{X} = \Omega$ [37]. While the supremum norm has a theoretical advantage, the L^2 norm will be used as discussed in [34]. Next, we will present some results for PDEs with Dirichlet boundary conditions and mixed Dirichlet-Neumann boundary conditions.

5. Numerical Examples

The results in this section were computed using *Mathematica* [59]. The Sinc spacing used for generating the Sinc points was $h = h(\frac{\pi}{2}, \frac{1}{4}) = \frac{\pi}{\sqrt{\frac{1}{2}N}}$, and the Sinc spacing used for generating the quadrature points was $h_Q = \frac{\pi}{\sqrt{N}}$. The stopping criterion $\varepsilon_{\text{stop}} = 10^{-4}$. A precision of 200 digits was used. For all examples, we set the number of points per cell to be constant (i.e., $m_X m_y = m^2 = (2N + 1)^2 = 9$). The traces were implemented as discussed in [60]. The number of quadrature points used in the Sinc quadrature (17) for a cell was

Example 1. Solution of two-dimensional Poisson equation [61] with Dirichlet boundary conditions:

$$\begin{cases} -\Delta u = 2\pi^2 \sin(\pi x) \sin(\pi y) & (x, y) \in [0, 1]^2, \\ u(x, 0) = u(x, 1) = 0 & x \in [0, 1], \\ u(0, y) = u(1, y) = 0 & y \in [0, 1], \end{cases}$$
(31)

whose exact solution is

$$u(x,y) = \sin(\pi x)\sin(\pi y).$$

The algorithm terminates after $\kappa = 3$ *iterations and the number of points* |S| = 684. *Figure 2a shows the 3D plot of* $u_c^{(3)}(x, y)$ *, and Figure 2b shows the contour plot of* $u_c^{(3)}(x, y)$ *.*

We perform least squares fitting of the logarithm of the set e to the logarithm of the upper bound of the inequality in Equation (24) as shown in Figure 3a. The dots represent the set e, and the solid line represents the least squares fitted model in Equation (24). The fitted model demonstrates an exponential decay in the mean value of $||e_l^{(i)}(x,y)||, l = 1, ..., K_i$, over the K_i cells. The parameter δ , which results from the estimation of the coefficients of $u_c^{(3)}(x, y)$ under the Poly-Sinc-based LDG method, is quite small. This indicates that our adaptive method can estimate the exact values of u(x, y) at the Sinc points up to a negligible error. Figure 3b shows the logarithmic plot of the absolute value of the local error $e^{(3)}(x, y)$. The jumps in the local error plot result from the partitioning process of the Poly-Sinc-based LDG method. The jumps are almost negligible, as seen in Figure 2. The L² norm of the approximation error $||u(x, y) - u_c^{(3)}(x, y)||_{L^2} \approx 1.2 \times 10^{-3}$.

We tabulate the test of normality ω_i as a function of the iteration index i, i = 2, 3, in Table 1. The test of normality ω_i is less than one, as discussed in [34,35].



Figure 2. A 3D plot and contour plot of $u_c^{(3)}(x, y)$ for Example 1. (a) 3D plot of $u_c^{(3)}(x, y)$; (b) Contour plot of $u_c^{(3)}(x, y)$.

Table 1. Test of normality ω_i , i = 2, 3, for Example 1.

i	ω_i
2	0.8385
3	0.7718



(a) $A = 1.34 \times 10^5$, r = 62.8, $\lambda = 0.197$, $\tilde{\delta} = 3.56 \times 10^{-5}$. (b) Logarithmic plot of $|u(x, y) - u_c^{(3)}(x, y)|$.

Figure 3. (a) Fitting the model in Equation (24) to the set e of Example 1. (b) Local error plot.

$$\begin{cases} \Delta u = 0 & (x, y) \in [-1, 1]^2, \\ u(x, -1) = 0 & x \in [-1, 1], \\ u(-1, y) = u(1, y) = 0 & y \in [-1, 1], \\ u(x, 1) = \sin\left(\frac{\pi}{2}(x+1)\right) & x \in [-1, 1], \end{cases}$$
(32)

whose exact solution is

$$u(x,y) = \frac{\sinh\left(-\frac{\pi}{2}(y+1)\right)\sin\left(\frac{\pi}{2}(x+1)\right)}{\sinh(-\pi)}$$

The algorithm terminates after $\kappa = 4$ iterations and the number of points |S| = 1494. Figure 4a shows the 3D plot of $u_c^{(4)}(x, y)$, and Figure 4b shows the contour plot of $u_c^{(3)}(x, y)$.

We perform least squares fitting of the logarithm of the set \in to the logarithm of the upper bound of the inequality in Equation (24) as shown in Figure 5a. The dots represent the set \in , and the solid line represents the least squares fitted model in Equation (24). The fitted model demonstrates an exponential decay in the mean value of $||e_l^{(i)}(x, y)||, l = 1, ..., K_i$ over the K_i cells. The parameter $\tilde{\delta}$, which results from the estimation of the coefficients of $u_c^{(4)}(x, y)$ under the Poly-Sinc-based LDG method, is quite small. This indicates that our adaptive method can estimate the exact values of u(x, y) at the Sinc points up to a negligible error. Figure 5b shows the logarithmic plot of the absolute value of the local error $e^{(4)}(x, y)$. The jumps in the local error plot result from the partitioning process of the Poly-Sinc-based LDG method. The jumps are almost negligible, as seen in Figure 4. The L^2 norm of the approximation error $||u(x, y) - u_c^{(4)}(x, y)||_{x_2} \approx 7.84 \times 10^{-4}$

The L² norm of the approximation error $||u(x,y) - u_c^{(4)}(x,y)||_{L^2} \approx 7.84 \times 10^{-4}$. We tabulate the test of normality ω_i as a function of the iteration index i, i = 2, ..., 4, in Table 2. The test of normality oscillates around the value $\overline{\omega_i} \approx 0.67$.

Table 2. Test of normality ω_i , i = 2, ..., 4, for Example 2.

i	ω_i
2	0.665
3	0.7206
4	0.6191



Figure 4. A 3D plot and contour plot of $u_c^{(4)}(x, y)$ for Example 2. (a) A 3D plot of $u_c^{(4)}(x, y)$; (b) Contour plot of $u_c^{(4)}(x, y)$.



(a) $A = 1.33 \times 10^5$, $r = 1.9 \times 10^2$, $\lambda = 0.38$, $\tilde{\delta} = 1.12 \times 10^{-5}$.

Figure 5. (a) Fitting the model in Equation (24) to the set e of Example 2. (b) Local error plot.

Example 3. Consider the following elliptic problem [43,62] with Dirichlet boundary conditions:

$$\begin{cases} -\Delta u = f(x, y) & \text{ in } \Omega, \\ u(x, y) = 0 & \text{ on } \partial\Omega, \end{cases}$$
(33)

where $\Omega = [-1, 4] \times [0, 1]$ and the exact solution is

$$u(x,y) = \frac{(x+1)(x-4)(1-y^2)y^2}{3.1596}.$$

The algorithm terminates after $\kappa = 4$ iterations and the number of points |S| = 954. Figure 6a shows the 3D plot of $u_c^{(4)}(x, y)$, and Figure 6b shows the contour plot of $u_c^{(4)}(x, y)$.

We perform least squares fitting of the logarithm of the set e to the logarithm of the upper bound of the inequality in Equation (24) as shown in Figure 7a. The dots represent the set e, and the solid line represents the least squares fitted model in Equation (24). The fitted model demonstrates an exponential decay in the mean value of $||e_l^{(i)}(x,y)||$, $l = 1, ..., K_i$, over the K_i cells. The parameter δ , which results from the estimation of the coefficients of $u_c^{(4)}(x, y)$ under the Poly-Sinc-based LDG method, is quite small. This indicates that our adaptive method can estimate the exact values of u(x, y) at the Sinc points up to a negligible error. Figure 7b shows the logarithmic plot of the absolute value of the local error $e^{(4)}(x, y)$. The jumps in the local error plot result from the partitioning process of the Poly-Sinc-based LDG method. The jumps are almost negligible, as seen in Figure 6. The L² norm of the approximation error $||u(x, y) - u_c^{(4)}(x, y)||_{L^2} \approx 1.7 \times 10^{-3}$.





0.0

We tabulate the test of normality ω_i , as a function of the iteration index i, i = 2, ..., 4, in Table 3. The test of normality oscillates around the value $\overline{\omega_i} \approx 0.62$.

Figure 6. A 3D plot and contour plot of $u_c^{(4)}(x, y)$ for Example 3. (a) 3D plot of $u_c^{(4)}(x, y)$; (b) Contour plot of $u_c^{(4)}(x, y)$.



n

2

4

(a)



3

x (b)

(a) $A = 1.33 \times 10^5$, $r = 2.1 \times 10^2$, $\lambda = 0.26$, $\tilde{\delta} = 6.37 \times 10^{-5}$.

Figure 7. (a) Fitting the model in Equation (24) to the set e of Example 3. (b) Local error plot.

Table 3. Test of normality ω_i , i = 2, ..., 4, for Example 3.

i	ω_i
2	0.7329
3	0.5189
4	0.5945

Example 4. Consider the following Poisson problem [22] with mixed Dirichlet-Neumann boundary conditions:

$$\begin{cases}
-\Delta u = 8\pi^2 \sin(2\pi(x+y)), & (x,y) \in [0,1]^2, \\
u(0,y) = \sin(2\pi y), & u(x,0) = \sin(2\pi y), & x \in [0,1], & y \in [0,1], \\
u_x(1,y) = 2\pi \cos(2\pi y), & u_y(x,1) = 2\pi \cos(2\pi x), & x \in [0,1], & y \in [0,1],
\end{cases}$$
(34)

whose exact solution is

$$u(x,y) = \sin(2\pi(x+y)).$$

The algorithm terminates after $\kappa = 5$ iterations and the number of points |S| = 7839. Figure 8a shows the 3D plot of $u_c^{(5)}(x, y)$, and Figure 8b shows the contour plot of $u_c^{(5)}(x, y)$. We perform least squares fitting of the logarithm of the set e to the logarithm of the upper bound

We perform least squares fitting of the logarithm of the set e to the logarithm of the upper bound of the inequality in Equation (24) as shown in Figure 9a. The dots represent the set e, and the solid line represents the least squares fitted model in Equation (24). The fitted model demonstrates an exponential decay in the mean value of $||e_l^{(i)}(x, y)||$, $l = 1, ..., K_i$, over the K_i cells. The parameter δ , which results from the estimation of the coefficients of $u_c^{(5)}(x, y)$ under the Poly-Sinc-based LDG method, is quite small. This indicates that our adaptive method can estimate the exact values of u(x, y) at the Sinc points up to a negligible error. Figure 9b shows the logarithmic plot of the absolute value of the local error $e^{(5)}(x, y)$. The jumps in the local error plot result from the partitioning process of the Poly-Sinc-based LDG method. The jumps are almost negligible, as seen in Figure 8. The L^2 norm of the approximation error $||u(x, y) - u^{(5)}(x, y)||_{r^2} \approx 5.6 \times 10^{-3}$

The L² norm of the approximation error $||u(x,y) - u_c^{(5)}(x,y)||_{L^2} \approx 5.6 \times 10^{-3}$. Figure 10 shows the distribution of the Sinc points for i = 5. The plot shows a dense distribution of Sinc points near the boundaries as well as the center of the domain Ω .

We tabulate the test of normality ω_i as a function of the iteration index i, i = 2, ..., 5, in Table 4. It can be observed that the test of normality ω_i is decaying.



Figure 8. A 3D plot and contour plot of $u_c^{(5)}(x, y)$ for Example 4. (a) 3D plot of $u_c^{(5)}(x, y)$; (b) Contour plot of $u_c^{(5)}(x, y)$.



Figure 9. (a) Fitting the model in Equation (24) to the set e of Example 4. (b) Plot of $|u(x,y) - u_c^{(5)}(x,y)|$.

i	ω_i
2	0.8372
3	0.7744
4	0.7222
5	0.7002

Table 4. Test of normality ω_i , i = 2, ..., 5, for Example 4.



Figure 10. Distribution of Sinc points for Example 4.

6. Conclusions

In this paper, we derived an *a priori* error estimate for the *h*-adaptive Poly-Sinc-based LDG algorithm and showed its exponential convergence in the number of iterations, provided that a good estimate of the exact solution u(x, y) at the Sinc points existed. The convergence of the adaptive algorithm was also shown. We used a statistical approach for refinement of the computational domain. We validated our adaptive method on elliptic PDEs with Dirichlet and mixed Neumann-Dirichlet boundary conditions. Our study demonstrates that better approximation can be obtained if the stopping criterion is less than 10^{-4} at the expense of more iterations.

Author Contributions: Conceptualization, G.B.; methodology, O.A.K.; software, O.A.K. and G.B.; validation, O.A.K. and G.B.; formal analysis, O.A.K.; investigation, O.A.K. and G.B.; resources, not applicable; data curation, not applicable; writing—original draft preparation, O.A.K.; writing—review and editing, G.B. and O.A.K.; visualization, O.A.K.; supervision, G.B.; project administration, G.B.; funding acquisition, not applicable. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the anonymous reviewers for their comments on an earlier draft of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Reed, W.H.; Hill, T. *Triangular Mesh Methods for the Neutron Transport Equation;* Technical Report; Los Alamos Scientific Lab.: Los Alamos, NM, USA, 1973.
- Cockburn, B.; Karniadakis, G.E.; Shu, C.W. The Development of Discontinuous Galerkin Methods. In Discontinuous Galerkin Methods; Springer: Berlin/Heidelberg, Germany, 2000; pp. 3–50. [CrossRef]
- Cockburn, B.; Kanschat, G.; Perugia, I.; Schötzau, D. Superconvergence of the Local Discontinuous Galerkin Method for Elliptic Problems on Cartesian Grids. *SIAM J. Numer. Anal.* 2001, 39, 264–285. [CrossRef]
- 4. Cockburn, B. Discontinuous Galerkin methods. ZAMM-J. Appl. Math. Mech. Angew. Math. Mech. Appl. Math. Mech. 2003, 83, 731–754. [CrossRef]
- 5. Dolejší, V.; Feistauer, M. Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow; Springer, Cham, Switzerland, 2015.
- 6. Stenger, F. Polynomial function and derivative approximation of Sinc data. J. Complex 2009, 25, 292–302. [CrossRef]

- Stenger, F.; Youssef, M.; Niebsch, J. Improved approximation via use of transformations. In *Multiscale Signal Analysis and Modeling*; Springer: New York, NY, USA, 2013; pp. 25–49. [CrossRef]
- Stenger, F.; El-Sharkawy, H.A.M.; Baumann, G., The Lebesgue constant for Sinc approximations. In New Perspectives on Approximation and Sampling Theory: Festschrift in Honor of Paul Butzer's 85th Birthday; Springer International Publishing: Cham, Switzerland, 2014; pp. 319–335. [CrossRef]
- 9. Youssef, M.; El-Sharkawy, H.A.; Baumann, G. Lebesgue constant using Sinc points. Adv. Numer. Anal. 2016, 2016. [CrossRef]
- Lybeck, N.J.; Bowers, K.L. Domain decomposition via the Sinc-Galerkin method for second order differential equations. In Domain Decomposition Methods in Scientific and Engineering Computing, Proceedings of the 7th International Conference on Domain Decomposition, Pennsylvania, PA, USA, 27–30 October 1993; American Mathematical Society: Providence, RI, USA, 1994; pp. 271–276.
- 11. Lybeck, N.J.; Bowers, K.L. Domain decomposition in conjunction with sinc methods for Poisson's equation. *Numer. Methods Partial. Differ. Equ.* **1996**, *12*, 461–487. [CrossRef]
- 12. Burda, P.; Novotný, J.; Šístek, J. Precise FEM solution of a corner singularity using an adjusted mesh. *Int. J. Numer. Methods Fluids* 2005, 47, 1285–1292. [CrossRef]
- 13. Youssef, M.; Baumann, G. On bivariate Poly-Sinc collocation applied to patching domain decomposition. *Appl. Math. Sci.* 2017, 11, 209–226. [CrossRef]
- 14. Frei, W. How to Identify and Resolve Singularities in the Model When Meshing. 2013. Available online: https://www.comsol. com/blogs/how-identify-resolve-singularities-model-meshing/ (accessed on 28 January 2023).
- Sönnerlind, H. Singularities in Finite Element Models: Dealing with Red Spots. 2015. Available online: https://www.comsol. com/blogs/singularities-in-finite-element-models-dealing-with-red-spots/ (accessed on 28 January 2023).
- 16. Baumann, G.; Stenger, F. Fractional calculus and Sinc methods. Fract. Calc. Appl. Anal. 2011, 14, 568–622. [CrossRef]
- 17. Cockburn, B.; Shu, C.W. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.* **1998**, *35*, 2440–2463. [CrossRef]
- Castillo, P.; Cockburn, B.; Perugia, I.; Schötzau, D. An A Priori Error Analysis of the Local Discontinuous Galerkin Method for Elliptic Problems. SIAM J. Numer. Anal. 2000, 38, 1676–1706. [CrossRef]
- Cockburn, B.; Kanschat, G.; Schötzau, D. The local discontinuous Galerkin method for linearized incompressible fluid flow: A review. *Comput. Fluids* 2005, 34, 491–506. [CrossRef]
- 20. Yan, J.; Shu, C.W. Local discontinuous Galerkin methods for partial differential equations with higher order derivatives. J. Sci. Comput. 2002, 17, 27–47. .:1015132126817. [CrossRef]
- Castillo, P. A review of the Local Discontinuous Galerkin (LDG) method applied to elliptic problems. *Appl. Numer. Math.* 2006, 56, 1307–1313. [CrossRef]
- 22. Baccouch, M. Optimal superconvergence and asymptotically exact a posteriori error estimator for the local discontinuous Galerkin method for linear elliptic problems on Cartesian grids. *Appl. Numer. Math.* **2021**, *162*, 201–224. [CrossRef]
- Cangiani, A.; Georgoulis, E.H.; Sabawi, Y.A. Adaptive discontinuous Galerkin methods for elliptic interface problems. *Math. Comp.* 2018, 87, 2675–2707. [CrossRef]
- Kompenhans, M.; Rubio, G.; Ferrer, E.; Valero, E. Adaptation strategies for high order discontinuous Galerkin methods based on Tau-estimation. J. Comput. Phys. 2016, 306, 216–236. [CrossRef]
- Kane, B.; Klöfkorn, R.; Dedner, A. Adaptive Discontinuous Galerkin Methods for Flow in Porous Media. In Proceedings of the Numerical Mathematics and Advanced Applications ENUMATH 2017, Voss, Norway, 25–29 September 2017; Radu, F.A., Kumar, K., Berre, I., Nordbotten, J.M., Pop, I.S., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 367–378. [CrossRef]
- 26. Wang, F.; Ling, M.; Han, W.; Jing, F. Adaptive discontinuous Galerkin methods for solving an incompressible Stokes flow problem with slip boundary condition of frictional type. *J. Comput. Appl. Math.* **2020**, *371*, 112700. [CrossRef]
- 27. Noventa, G.; Massa, F.; Rebay, S.; Bassi, F.; Ghidoni, A. Robustness and efficiency of an implicit time-adaptive discontinuous Galerkin solver for unsteady flows. *Comput. Fluids* **2020**, *204*, 104529. [CrossRef]
- Bassi, F.; Colombo, A.; Crivellini, A.; Fidkowski, K.J.; Franciolini, M.; Ghidoni, A.; Noventa, G. Entropy-Adjoint *p*-Adaptive Discontinuous Galerkin Method for the Under-Resolved Simulation of Turbulent Flows. *AIAA J.* 2020, *58*, 3963–3977. [CrossRef]
- Bonev, B.; Hesthaven, J.S.; Giraldo, F.X.; Kopera, M.A. Discontinuous Galerkin scheme for the spherical shallow water equations with applications to tsunami modeling and prediction. *J. Comput. Phys.* 2018, 362, 425–448. [CrossRef]
- 30. Beisiegel, N.; Vater, S.; Behrens, J.; Dias, F. An adaptive discontinuous Galerkin method for the simulation of hurricane storm surge. *Ocean. Dyn.* **2020**, *70*, 641–666. [CrossRef]
- 31. Faghih-Naini, S.; Aizinger, V. *p*-adaptive discontinuous Galerkin method for the shallow water equations with a parameter-free error indicator. *GEM-Int. J. Geomath.* **2022**, *13*, 18. [CrossRef]
- Khalil, O.A.; Baumann, G. Discontinuous Galerkin methods using poly-sinc approximation. *Math. Comput. Simul.* 2021, 179, 96 110. [CrossRef]
- Khalil, O.A.; Baumann, G. Convergence rate estimation of poly-Sinc-based discontinuous Galerkin methods. *Appl. Numer. Math.* 2021, 165, 527–552. [CrossRef]
- 34. Khalil, O.; El-Sharkawy, H.; Youssef, M.; Baumann, G. Adaptive piecewise Poly-Sinc methods for ordinary differential equations. *Algorithms* **2022**, *15*, 320. [CrossRef]
- 35. Khalil, O.A.; El-Sharkawy, H.A.; Youssef, M.; Baumann, G. Adaptive piecewise Poly-Sinc methods for function approximation. *Appl. Numer. Math.* **2023**, *186*, 1–18. [CrossRef]

- 36. Geary, R.C. The ratio of the mean deviation to the standard deviation as a test of normality. *Biometrika* **1935**, 27, 310–332. [CrossRef]
- Stenger, F. Numerical Methods Based on Sinc and Analytic Functions; Springer Series in Computational Mathematics; Springer Science & Business Media: New York, NY, USA, 1993; Volume 20. [CrossRef]
- 38. Stenger, F. Handbook of Sinc Numerical Methods; CRC Press: Boca Raton, FL, USA, 2011. [CrossRef]
- 39. Youssef, M.; El-Sharkawy, H.A.; Baumann, G. Multivariate Lagrange interpolation at Sinc points: Error estimation and Lebesgue constant. *J. Math. Res.* 2016, *8*, 4. [CrossRef]
- 40. Baumann, G.; Stenger, F. Sinc-approximations of fractional operators: A computing approach. *Mathematics* **2015**, *3*, 444–480. [CrossRef]
- 41. Stoer, J.; Bulirsch, R. Introduction to Numerical Analysis; Springer: New York, NY, USA, 2002. [CrossRef]
- Youssef, M.; Baumann, G. Collocation method to solve elliptic equations, Bivariate Poly-Sinc approximation. J. Progress. Res. Math. 2016, 7, 1079–1091.
- 43. Youssef, M. Poly-Sinc Approximation Methods. Ph.D. Thesis, German University in Cairo, Cairo, Egypt, 2017.
- 44. Rivière, B. Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation; SIAM: Philadelphia, PA, USA, 2008.
- 45. Adjerid, S.; Chaabane, N. An improved superconvergence error estimate for the LDG method. *Appl. Numer. Math.* 2015, 98, 122–136. [CrossRef]
- 46. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. Introduction to Algorithms; MIT Press: Cambridge, MA, USA, 2009.
- Haasdonk, B.; Ohlberger, M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. M2AN Math. Model. Numer. Anal. 2008, 42, 277–302. [CrossRef]
- 48. Grepl, M.A. Model order reduction of parametrized nonlinear reaction–diffusion systems. *Comput. Chem. Eng.* **2012**, *43*, 33–44. [CrossRef]
- 49. Sleeman, M.K.; Yano, M. Goal-oriented model reduction for parametrized time-dependent nonlinear partial differential equations. *Comput. Methods Appl. Mech. Eng.* 2022, 388, 114206. [CrossRef]
- Nochetto, R.H.; Siebert, K.G.; Veeser, A. Theory of adaptive finite element methods: An introduction. In *Proceedings of the Multiscale, Nonlinear and Adaptive Approximation*; DeVore, R., Kunoth, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 409–542. [CrossRef]
- 51. Walpole, R.E.; Myers, R.H.; Myers, S.L.; Ye, K. *Probability and Statistics for Engineers and Scientists*, 9th ed.; Pearson Education, Inc.: Boston, MA, USA, 2012.
- 52. Carey, G.F.; Humphrey, D.L. Finite element mesh refinement algorithm using element residuals. In *Proceedings of the Codes for Boundary-Value Problems in Ordinary Differential Equations*; Childs, B., Scott, M., Daniel, J.W., Denman, E., Nelson, P., Eds.; Springer: Berlin/Heidelberg, Germany, 1979; pp. 243–249. [CrossRef]
- 53. Carey, G.F. Adaptive refinement and nonlinear fluid problems. *Comput. Methods Appl. Mech. Eng.* **1979**, 17–18, 541–560. [CrossRef]
- 54. Carey, G.F.; Humphrey, D.L. Mesh refinement and iterative solution methods for finite element computations. *Int. J. Numer. Methods Eng.* **1981**, *17*, 1717–1734. [CrossRef]
- 55. Abramowitz, M.; Stegun, I.A. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables,* 10th printing ed.; Dover Publications, Inc.: New York, NY, USA, 1972.
- 56. Youssef, M.; Pulch, R. Poly-Sinc solution of stochastic elliptic differential equations. J. Sci. Comput. 2021, 87, 82. [CrossRef]
- 57. Rudin, W. Functional Analysis; McGraw-Hill, Inc.: New York, NY, USA, 1973.
- 58. Bartle, R.G.; Sherbert, D.R. Introduction to Real Analysis, 3rd ed.; John Wiley & Sons, Inc.: New York, NY, USA, 2000.
- 59. Wolfram Research, Inc. *Mathematica*; Version 13.0; Wolfram Research, Inc.: Champaign, IL, USA, 2021.
- Arndt, D.; Bangerth, W.; Feder, M.; Fehling, M.; Gassmöller, R.; Heister, T.; Heltai, L.; Kronbichler, M.; Maier, M.; Munch, P.; et al. Reference Documentation for deal.II Version 9.4.1: The Step-12 Tutorial Program. 2023. Available online: https: //www.dealii.org/current/doxygen/deal.II/step_12.html (accessed on 19 January 2023).
- 61. Dell'Accio, F.; Di Tommaso, F.; Nouisser, O.; Siar, N. Solving Poisson equation with Dirichlet conditions through multinode Shepard operators. *Comput. Math. Appl.* 2021, *98*, 254–260. [CrossRef]
- 62. Lybeck, N.J. Sinc Domain Decomposition Methods for Elliptic Problems. Ph.D. Thesis, Montana State University, Bozeman, Montana, 1994.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.