Results in Applied Mathematics 12 (2021) 100200

Contents lists available at ScienceDirect

Results in Applied Mathematics

journal homepage: www.elsevier.com/locate/results-in-applied-mathematics

Remarks on the numerical approximation of Dirac delta functions

Vincenzo Schiano Di Cola [a,*], Salvatore Cuomo [b], Gerardo Severino [c]

[a] Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80125 Napoli, Italy
[b] Department of Mathematics and Applications “Renato Caccioppoli”, University of Naples Federico II, 80126 Napoli, Italy
[c] Department of Agricultural Sciences, University of Naples Federico II, 80055 Portici, Italy

ARTICLE INFO

Article history:
Received 11 August 2021
Accepted 28 September 2021
Available online 12 November 2021

MSC:
65-02
65-04
47-08
35-01
65M30

Keywords:
Approximate Dirac delta function
Immersed boundary (IB) method
Level set methods

ABSTRACT

We investigate the convergence rate of the solutions of one and two-dimensional Poisson-type PDEs where the Dirac delta function, representing the forcing term, is approximated by several expressions.

The goal is to see if the solution to a Poisson's equation converges when solved by a numerical method, with a source or sink approximated using a delta proposed in the literature. We will look at two parameters, how fast it converges, by estimating the order of convergence, and how well, by calculating the error between the analytical form and the numerical result. We investigate smoothed discrete delta functions based on the Immersed boundary (IB) approach, and we revisit their definitions, as in level set methods, by expanding their support for assessing higher-order of convergence in PDE solutions. We developed a Python package utilizing FiPy, a PDE solver based on the finite volume (FV) technique, and accelerated the solver with the AmgX package, a GPU solution. We have observed that when the support is wider, better results may be achieved. Moreover, the overall trends of error and the convergence rate in the 2D configuration differ from the 1D problems.

1. Introduction

Flows generated by single/distributed sources, such as those determined by well(s), are of central interest in reservoir-engineering and hydrology [1,2]. The equivalent problem is encountered in many other branches of physics, and therefore it represents a topic of wide interest for the applications. Many analytical (closed form) solutions for such flows have been derived in the past. These solutions have served as a basis for solving many practical problems. However, there are numerous applications where the complexity of the boundary conditions as well as the spatial variability of the input parameters [3], *de facto* defies any attempt to get analytical solutions unless one develops a completely different procedure which in any case is of limited applicability [4,5]. In these cases, one must resort to numerical methods.

In the present paper, we deal with a steady flow driven by a Dirac distribution. As it is well known, such a configuration gives rise to an elliptic PDE with a singular forcing term at the right-hand side. We aim at setting up a procedure to gain an accurate numerical approximation of the δ-function.

Common approximations methods for Dirac delta distributions are immersed boundary (IB) methods [6], or level set methods [7] and vortex (VOF) methods [8]. Lee et al. [8] present a brief review on the numerical methods for regularized Dirac delta functions. The level set method formulation utilizes Dirac delta which is supported on curves or surfaces. In a standard approach of the level set, the surfaces or curves are implicitly described as the zero level set of a continuous

* Corresponding author.
E-mail addresses: vincenzo.schianodicola@unina.it (V. Schiano Di Cola), salvatore.cuomo@unina.it (S. Cuomo), severino@unina.it (G. Severino).

https://doi.org/10.1016/j.rinam.2021.100200

function, regularized before being shown on the computational grid. An elastic boundary is represented by a set of Lagrangian points in the immersed boundary technique, and the singular force at the Lagrangian points is given by the generalized Hooke's law. Using a delta function, this force is distributed to the surrounding Eulerian points.

Paper's structure and contributions. This paper presents a brief background on Dirac delta functions approximations, including preliminary definitions, as well as the most relevant literature and examples. Then the paper presents the main result, the convergence test conducted on two Poisson's PDE, a one-dimensional and a two-dimensional, and the Python code that generated all the results. We argue that, while the Dirac delta approximations proposed in the literature over the last 20 years are capable of producing good results in a 1D context, there is still potential for improvement in 2D scenarios. Final remarks on the resulting experimental setup are given in the conclusion.

2. Mathematical background

In this section we discuss some theoretical results about the Dirac delta approximations. The goal of numerically solving a PDE with a sink or source is to approximate the delta distribution δ using alternative distributions δ_ϵ for $\epsilon > 0$, with the property that $\delta_\epsilon \to \delta$ in some reasonable sense and that the solution of such PDE $\mathcal{L}$ converges $\mathcal{L}(\delta_\epsilon) \to \mathcal{L}(\delta)$. These approximations are known as *regularizations* [9]. The function $\delta_\epsilon(r)$ is known as discrete delta function regularization of the Dirac delta function. The discrete delta function is assumed to be represented by a tensor product of a single-variable kernel $\phi(r)$ defined on the real line,

$$\delta_h(\boldsymbol{x}) = \frac{1}{h^n}\prod_{i=1}^{n}\phi\left(\frac{x_i}{h}\right), \boldsymbol{x} = (x_1, \ldots, x_n)^T \tag{1}$$

where h is the mesh size of a uniform Cartesian grid. This representation is not required, but it substantially simplifies the study, by focusing on the single-variable kernel $\phi(r)$.

Functions $\phi(r)$ is frequently found in literature, e.g. [10], to satisfy a subset of the following properties:

1. $\phi(r)$ is continuous for all real r;
2. ϕ has compact support, i.e.,

   $$\phi(r) = 0 \text{ for } |r| \geq r_s,$$

   where r_s is the radius of the support;
3. ϕ satisfies the sum of squares condition:

   $$\sum_{l\in\mathbb{Z}}\phi(l-r)^2 = K, \qquad \forall r \in \mathbb{R} \text{ , for some constant } K;$$

4. ϕ satisfies the jth order moment conditions, when ϕ satisfies

   $$\sum_{l\in\mathbb{Z}}\phi(l-r) = 1 \qquad \text{if } j = 0$$

   $$\sum_{l\in\mathbb{Z}}(l-r)^j\phi(l-r) = K \qquad \text{if } j > 0$$

   as $K = 0$ up to j. If ϕ verifies the moment conditions up to order $m-1$, we can say that ϕ is of *moment order m*;
5. ϕ satisfies the even–odd condition: $\sum_{i\ even}\phi(r-i) = \sum_{i\ odd}\phi(r-i) = \frac{1}{2}$ for all r;

We can observe that:

1. To avoid abrupt jumps in interpolation, ϕ continuity is assumed.
2. For computational efficiency a ϕ with compact support is required. This also ensures that each point influences a finite number of grid points, regardless of mesh spacing h. The compact support criterion also means that the sums in the definitions of moment order have a finite number of terms.
3. The sum of square indicates that the coupling of $\phi(r)$ between any two locations r_1, r_2 is just a function of r_1-r_2 [11].
4. The moment condition assures that interpolation operations conducted using discrete delta functions are accurate [12].
5. The even–odd condition implies the 0-th order moment. In [10] the authors extended this condition to what they called the *smoothing order* condition and showed that it suppresses high-frequency errors and prevents Gibbs-type phenomena. Note that ϕ is of *smoothing order* $s \geq 1$ if there is a function $\psi(r)$ of compact support such that

   $$\phi(r) = \frac{1}{2^s}\sum_{l=0}^{s}\binom{s}{l}\psi(r-l)$$

   where $\binom{s}{l}$ is the binomial coefficient.

In the following paragraph, we present an alternate formulation of delta approximations, i.e. the level set technique, which is denoted in the literature as φ. Before we observe that the delta approximations ϕ can be dilated and scaled by a positive integer λ to create a new function ϕ' as $\lambda\phi'(\lambda x) = \phi(x)$. The new ϕ' meets the same requirements as ϕ.

Typically, the discrete approximants δ_ϵ are piecewise smooth functions selected to ensure that these criteria are satisfied. It is most practical to deal with continuous δ_ϵ functions in computations, and we can define an approximation delta functions δ_ϵ in a different form:

$$\delta_\epsilon(x) = \begin{cases} \frac{1}{\epsilon}\varphi\left(\frac{x}{\epsilon}\right), & \text{if } |\frac{x}{\epsilon}| \le r_\varphi \\ 0, & \text{if } |\frac{x}{\epsilon}| > r_\varphi \end{cases} \tag{2}$$

where φ is a re-scaled version of ϕ. This formulation can be used to define the immersed boundary (IB) method or the level set method (for a review on the numerical methods for regularized Dirac delta functions see [8]). In the IB method, the elastic boundary is represented by a collection of Lagrangian points, and the singular force at these points is distributed to the surrounding Eulerian points using a delta function. Whereas, in the level set technique, delta functions are frequently employed to spread a singular force.

- In the level set method, the support is r_φ, and ϵ is proportional to the grid size, that is, $\epsilon = kh$ for a positive number k, possibly indicating the moment order.
- In the immersed boundary method, $\epsilon = h$ and for all $\frac{x}{h} = r$, where r is the parameter representing the position of the submerged boundary point and is scaled with respect to the grid size h, the approximation can be written as $\phi(r)$, and the support r_ϕ is usually equal to $1, 1.5, 2, 2.5$.

To be consistent with formulation (1), we can actually define, for (2), $\frac{1}{k}\varphi(\frac{x}{kh}) = \phi(\frac{x}{kh})$. In Section 3.3 we will re-scale the various delta approximations using k by extending IB based approximations with a level set approach.

In the following examples, we see that $\varphi(x)$ is a scaling of $\phi(x)$ in the variable x and not of the function coefficient. IB and level set approximations result as being scaled versions of the same formalism but, to the best of the authors' knowledge, there is no systematic study of how IB behave when scaled. One of the goals of this research is to show what happens to IB approximation when they are scaled. When available in the literature, we provide the link between IB and level set formulations, following that, we present solely IB methods.

With notation (2), the narrow linear 2-point hat function δ_h^L, with $\epsilon = h$, is defined as

$$\phi_1^L(r) = 1 - |r| \qquad = \qquad \varphi_1^L(\xi) = 1 - |\xi|$$

where $r = x/h = \xi$ and $r_{\phi_1^L} = 1$.

The wider 4-point hat function δ_{2h}^L, with $\epsilon = 2h$, is defined as

$$\phi_2^L(r) = \frac{1}{2} - \frac{1}{4}|r|; \qquad \varphi_2^L(\xi_2) = \frac{1}{2}\cdot(1 - |\xi_2|) = \frac{1}{2}\phi_1^L(r/2)$$

where $r = x/h$, $\xi_2 = x/2h$, and $r_{\phi_2^L} = 2$, which is equivalent to the 2-point as being shorted and wider, by having $\delta_{2h}^L = 1/h\phi_2^L(r)$ for $|r| \le 2$ on support 2 or, equivalently, $\delta_{2h}^L = 1/(2h)\varphi_2^L(\xi_2)$ for $|\xi_2| \le 1$.

Most commonly used delta functions are the 4-point cosine function $\phi_2^{\cos}(r) = \frac{1}{4}(1-\cos(\frac{\pi r}{2}))$, with $r_{\phi_2^{\cos}} = 2$ equivalent to $\varphi^{\cos}(\xi) = \frac{1}{2}(1 - \cos(\pi\xi))$. In either case,

$$\delta^{\cos}(x) = \begin{cases} \frac{1}{4}(1 - \cos(\frac{\pi x}{2})), & \text{if } |\frac{x}{h}| \le 2 \\ 0, & \text{otherwise.} \end{cases} \qquad \delta^{\cos}(x) = \begin{cases} \frac{1}{\epsilon}\frac{1}{2}(1 + \cos(\frac{\pi x}{\epsilon})), & \text{if } |\frac{x}{\epsilon}| \le 1 \\ 0, & \text{otherwise.} \end{cases}$$

Another kind of delta approximations are the smoothed versions, introduced in [6] as an IB method. An example is the smoothed 2-point function $\phi_1^*(r)$:

$$\phi_1^*(r) = \begin{cases} \frac{3}{4} - r^2, & \text{if } |r| \le 0.5 \\ \frac{9}{8} - \frac{3}{2}|r| + \frac{r^2}{2}, & \text{if } 0.5 < |r| \le 1.5 \end{cases}$$

with support $r_{\phi_1^*} = 1.5$. More regular functions can be obtained by changing the support size or adding/deleting conditions.

Another approach discussed in [9], provides a solution based on *shifted Legendre polynomials*. The authors observed that, δ_h is restricted to be a sufficiently smooth and integrable function and is compactly supported inside a set of diameter less than h, it will satisfy the moment conditions; however, the number m of moment conditions satisfied depends on the choice of smoothness parameter. As long as the approximations we want are also polynomials, the shifted Legendre polynomials give a strong foundation for addressing the moment problem. Hosseni et al. [9] obtains $\delta_h(x)$ as a scaled, even extension of $\phi(r) = \eta_{m,p}(r)$ where $\eta_{m,p}$ is a polynomial of degree p and moment m, e.g. $\eta_{2,2}(r)$ obtained for a 1D space is the polynomial $9/2 - 18r + 15r^2$.

A Legendre polynomial expansion may be used to express the approximate delta function (ADF) of order N explicitly. In [13] is described a finite-order polynomial that has the same integral features as the Dirac delta function when combined with a finite-order polynomial integrand over a finite domain, so

$$\tilde{\delta}_N(x, z) = \frac{1}{h}\sum_{i=0}^{N}(2i+1)L_i(\xi)L_i(\eta), \qquad \xi = \frac{x}{h/2}, \quad \eta = \frac{z}{h/2},$$

where $L_i(\xi)$ represents for the Legendre polynomial of order i.

Final remark can be given on the assessment of convergence of such Dirac delta approximations. According to [7], for a very narrow support, the δ_ϵ function is not sufficiently resolved to examine the error by breaking it into an analytical and a numerical part. For a very narrow support, the error must be studied explicitly, taking into consideration the computational grid's discrete impacts. Such approximations must be studied with narrow ϵ proportional to h, usually $\epsilon = mh$, $m = 1$, 2 or 3. This strategy works well in one dimension, when the one-dimensional delta approach complies with certain moment criteria. However, the extension to a multi-dimensional system can cause $O(1)$ errors, as proven for the Γ curve of $/R^2$ [12]. To achieve convergence, δ_ϵ must be increased and be selected to scale with h algebraically. Moreover, even if the error is theoretically $O(h^m)$ for certain m, it is not so constantly and this may influence in numerical tests the comparing of errors for convergence on finer grids. The formal order can be estimated by averaging over numerous shifts in the grid [7].

3. Main results

In our research, we estimate the order of convergence of two boundary value problems with a delta function as the right-hand side, one in one dimension and the other in two dimensions.

The one-dimensional problem is a simple boundary value problem presented in [12]:

$$-u_{xx} = \delta(x - 0.5); \qquad u(0) = u(1) = 0 \tag{3}$$

where this equation has the solution

$$u(x) = G(x, 0.5) = \begin{cases} x(1 - 0.5), & 0 \le x \le 0.5 \\ 0.5(1 - x), & 0.5 < x \le 1 \end{cases}$$

where G is the Green's function.

The 2D problem, will be the same as in [14],

$$-\nabla^2 u(x, y) = \delta(x, y); \qquad u(x, y) = 0 \text{ on } \partial\Omega \tag{4}$$

where $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$; whose fundamental solution (Green function) is

$$u(x, y) = \frac{\ln(2|(x, y)|)}{2\pi},$$

where $|\cdot|$ is the order 2 norm, i.e. $|(x, y)| = \sqrt{x^2 + y^2}$.

In this section, we introduce the Dirac delta regularization's that we used, and the numerical setups, for solving the Poisson problem and estimate the rate of convergence of the solution. The objective is to see if a Poisson's solution converges with an approximated source or sink by means of a delta as suggested in the literature when solved by a numerical method.

3.1. Dirac delta approximations

We tested the most commonly used delta functions,[8]. The 2-point hat function with semi-support 1 is compactly indicated as `p2h-s1`, the 3-point function with semi-support 1.5 is written as `p3-s15`, the 4-point function with semi-support 2 (`p4-s2`), and the 4-point cosine with semi-support length 2 is denoted as `p-cos-s2`. We also included the smoothed version introduced by Yang et al. [6] in the test set, which includes the smoothed 2-point function (`p*2-s15`), the smoothed 3-point function (`p*3-s2`), and the smoothed 4-point cosine function (`p*-cos-s25`) that has a support of length 2.5. In a recent development in [11], we took the implementation of the *new 5-point kernel* (`pg5-s25`) based on Gaussian-Like Immersed Boundary Kernels. We also tested the delta distribution regularizations proposed in [9], which are based on Legendre and Trigonometric polynomials and have normalized support to 1. The approximations were designed specifically for 1D and 2D spaces, and we tested them for both spaces. The results presented here encompass a cosine approximations $\eta_{1,\cos}$ (`cos-1-1d`) and the two Legendre approximations, $\eta_{1,1}$ (`l-1-1-1d`) and $\eta_{2,3}$ (`l-2-5-1d`); where `l-1-1-1d` is actually the 2-point hat function `p2h-s1`. Another Legendre polynomial expansions were from [13]: the one with $N = 0$ and $z = 0$ (`adf-0-z0`) and the one with $N = 4$ and $z = 1$ (`adf-4-z1`). Fig. 1 shows the profile of the various regular functions, ϕ_i, $i \in \{$l-1-1-1d, ..., p-cos-s2$\}$ that form the delta functions, δ_i.

We also investigated the extra smoothing impact of a discrete delta function with greater support for all of these deltas by multiplying h (and so the support) by multiple k between 1 and 4, spacing by $\Delta k = 0.1$ and 0.25. Finally, as to deal with the extension from the 1D to the 2D version of the delta approximation, we used the usual tensor product, but we also used the observations in [9] and investigated whether there were significant differences for a scaled tensor product or the radial form. In terms of the scaled tensor product, we defined a smaller support for the 2D case, using $\tilde{h} = h/\sqrt{2}$ so to fit the square $[-\tilde{h}, \tilde{h}] \times [-\tilde{h}, \tilde{h}]$ inside the ball of radius h.

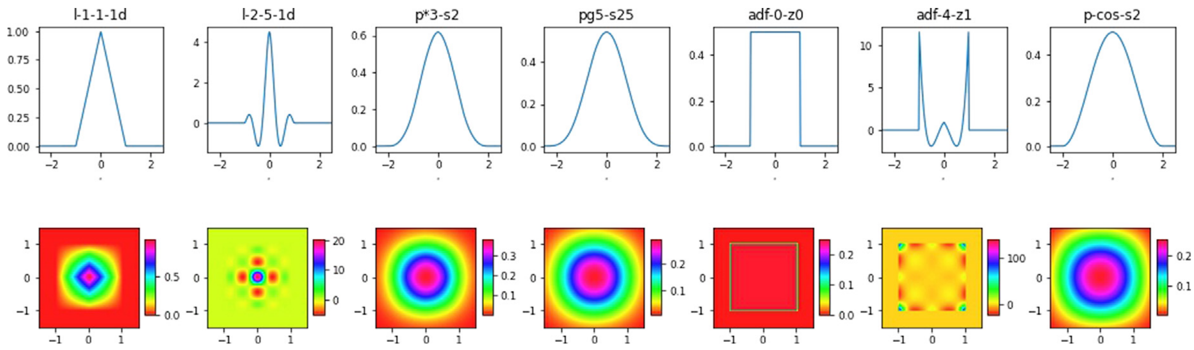

Fig. 1. Example of a delta regularization, first row are in 1D, second row are drown in 2D using the tensor product. All have $k = 1$.

3.2. Numerical issues

We examine the convergence rate of various delta approximations within a 1D or 2D known sample test in this brief work. We will look at two parameters: the difference between the analytical form and the numerical result, and how fast the numerical solution converges to the analytical one.

The convergence rate is determined by adjusting the mesh width $h_n = 1/(2^9 + 2^n)$ for $n = \{0, 1, \dots, 7\}$. The error between the numerical solution s and the analytical solution a is determined for each mesh with the mean absolute error (MAE) $e = E[|s - a|]$. Then, between two realizations e_{n+1} and e_n, an *estimate* of order of convergence is calculated:

$$q_n = \frac{\log \frac{e_n}{e_{n-1}}}{\log \frac{h_n}{h_{n-1}}} \tag{5}$$

Finally, for $n = \{2, \dots, 7\}$, we provide the average of all q_n and the convergence estimate among the extremes $q_{7,0}$. These two values were used to assess the convergence of the setup; in Table 1 we compactly report their average as an estimate for the order of convergence.

The table and figure provided in this paper were obtained in Python, by creating a framework based on FiPy and AmgX. FiPy [15] is an object-oriented, partial differential equation (PDE) solver written in Python that is based on a conventional finite volume (FV) technique. The AmgX package (pyamgx.readthedocs.io/) is a solver library designed to accelerate intensive linear solver simulations with NVIDIA GPUs; we configured it to be used as a solver within the FiPy package, by employing an Algebraic Multigrid (AMG) technique with a l^1-Jacobi smoother, a Jacobi relaxation used also in [16]. All the tests were run by setting the desired residual at $2 \cdot 10^{-5}$ for FiPy's sweeps and the tolerance as $2 \cdot 10^{-7}$ for the AmgX solver.

Other tests that are not shown here were performed by using PETSc [17] for some 1D tests, but a GPU accelerated solver library was required for all 2D testing. The code, as well as all of the full-length tests and graphs, can be found at the GitHub link: https://github.com/MthBr/DEDICATE.

3.3. Numerical experiments

We study the order of convergence with the application of Eq. (5) by adopting varied mesh with width $h_n = 1/(2^9+2^n)$ for $n = \{0, 1, \dots, 7\}$. In our simulation results, we investigate what happens when $h \to 0$ as the mesh becomes finer, as well as by altering the kind of delta and the size for the support of these delta approximations. We present the results for five different values of $k = \{1, 1.5, 2, 2.5, 3\}$, which result in five different scaled versions of the IB approximations.

In Table 1, the numerical solution converges to the analytical solution very well in the one-dimensional case, especially for non-Legendre approximations. In the 2D instance, the cosine delta types seem non to converges to the real solution, and the MAE tends to stabilize when varying k. Using standard tensor product, we discovered that the best results are obtained when the support is amplified; also, the overall trends found in the 2D configuration differ from the 1D model. Our results confirm that for most delta approximations there is an order 2 convergence for the 1D, but in the 2D case this is not observed, and there are cases, as in `adf-4-z1` with $k = 2.5$, where there is a high order of convergence but also a high MAE, and the numerical solution does not qualitatively adhere to the analytical solution. Maybe the best solution for the 2D case is by using the new approach by [11], `pg5-s25`, when the support is three times wider (i.e. $r_\phi = 7.5$). MAE and order of convergence estimation were worst when adopting the Radial definition of the delta in 2D or applying the scaled tensor product, as done by [9]. These results take into account that, for finite volume (FV) methods applied to elliptic PDEs, even if the numerical solution is qualitatively similar to the analytical solution, there is an O(1) convergence in the 2D case, a problem that has been known in literature for decades [12], but even if there have been novel Dirac approximations proposals, there is still much to be done.

Table 1
Mean absolute error and convergence rates for the 1D (3) and 2D (4) problems. All delta approximations were studied utilizing the support defined by the authors as well with a larger *k*-scaled version.

delta	k	MAE 1D	**q 1D**	MAE 2D	**q 2D**	delta	k	MAE 1D	**q 1D**	MAE 2D	**q 2D**	delta	k	MAE 1D	**q 1D**	MAE 2D	**q 2D**
	1	**1,000E−9**	−0,082733	0,015098	−0,000012		1	0,598633	−3,954E−10	0,999123	−0,000002		1	**1,000E−9**	−0,082721	0,015098	−0,000012
	1,5	0,020937	−10,801203	0,039177	−3,282841		1,5	1,057842	−2,009271	7,899682	−3,816873		1,5	0,003694	−0,000303	0,019516	−0,000067
adf-0-z0	2	0,000001	**1,997945**	**0,015098**	0,000077	adf-4-z1	2	**0,117983**	0,000030	0,066205	−0,000007	p3-s15	2	0,000001	1,997874	0,015098	0,000045
	2,5	0,018801	−0,490432	0,030287	**1,088726**		2,5	0,547909	**1,765054**	2,242771	**6,487220**		2,5	0,001103	0,003652	**0,013808**	**0,000280**
	3	0,000004	1,999229	0,015100	0,000248		3	0,011263	0,002364	**0,012890**	0,002311		3	0,000003	**1,999150**	0,015099	0,000172
	1	**1,000E−9**	−0,082721	0,015098	−0,000012		1	**1,000E−9**	−0,082721	0,015098	−0,000012		1	**3,940E−7**	1,993013	**0,015098**	0,000008
	1,5	0,013889	0,000000	0,014615	−0,000008		1,5	0,002314	−0,000365	0,017851	−0,000057		1,5	0,000001	1,997858	0,015098	0,000051
l-1-1-1d	2	0,000001	**1,995903**	0,015098	0,000032	p*2-s15	2	0,000001	**1,997266**	0,015098	0,000040	p4-s2	2	0,000002	1,998891	0,015099	0,000130
	2,5	0,005001	0,000324	**0,010975**	**0,000131**		2,5	0,000502	0,006552	**0,014511**	**0,000140**		2,5	0,000004	1,999324	0,015100	0,000226
	3	0,000002	1,998463	0,015099	0,000095		3	0,000003	1,998975	0,015099	0,000139		3	0,000006	**1,999540**	0,015101	**0,000344**
	1	0,382812	−5,782E−10	0,254817	−0,000002		1	**1,780E−7**	1,984541	0,015098	−0,000019		1	0,006856	**18,071287**	0,024952	**2,518579**
	1,5	0,231996	−1,016E−9	0,017682	−0,000003		1,5	0,000852	0,001795	**0,014099**	0,000093		1,5	0,000182	−0,011291	0,015314	0,000032
l-2-5-1d	2	0,023927	0,000073	0,022339	0,000055	p*3-s2	2	0,000002	1,998394	0,015098	0,000083	pg5-s25	2	**0,000002**	1,998812	0,015099	0,000115
	2,5	0,032455	−0,000057	0,029008	−0,000055		2,5	0,000153	0,035280	0,014922	0,000170		2,5	0,000028	0,262111	**0,015071**	0,000201
	3	**0,004726**	**0,000121**	**0,011033**	**0,000090**		3	0,000004	**1,999357**	0,015100	**0,000224**		3	0,002289	12,651476	0,017989	1,279456
	1	**1,000E−9**	−0,082715	0,015098	−0,000012		1	**3,940E−7**	1,993013	**0,015098**	0,000008		1	**4,880E−7**	1,994366	**0,015098**	0,000018
	1,5	1,000E−9	−0,082721	0,015098	−0,000012		1,5	0,000001	1,997805	0,015098	0,000045		1,5	0,000001	1,998152	0,015098	0,000066
cos-1-1d	2	3,940E−7	1,993013	0,015098	0,000008	p-cos-s2	2	0,000002	1,998889	0,015099	0,000131	p*cos-s25	2	0,000003	1,999055	0,015099	0,000152
	2,5	0,000001	1,996293	0,015098	0,000041		2,5	0,000004	1,999322	0,015100	0,000226		2,5	0,000005	1,999419	0,015100	0,000257
	3	0,000001	**1,997805**	0,015098	**0,000045**		3	0,000006	**1,999540**	0,015101	**0,000345**		3	0,000007	**1,999608**	0,015101	**0,000384**

4. Conclusion

The data analysis of the convergence rate of numerous delta approximations found in the literature is presented in this micro-article. We investigated delta regularizations, which were introduced using the immersed boundary technique, and assessed their order of convergence by extending their support with a non-integer k coefficient, like in the level set method, by scaling the Dirac approximations. Instead of utilizing the typical $k = 1, 2$ values used in the literature, we discovered that there can be smaller errors and a higher order of convergence with non-integer scaling values of k. Furthermore, despite the fact that the approximations presented in the literature were designed to have either high moment conditions or to be convergent in some aspects, there are still various constraints when it comes to solving Poisson's equation in two dimensions. We believe that there is still an opportunity for improvement in the quality of delta regularizations, and that future research should investigate numerical simulations of a three-dimensional problem, which will become more viable as graphics processing units improve and become more widely available.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

S.C. thanks for the support the GNCS-INdAM. His research has been done within the Italian Network on Approximation (RITA) and the thematic group on "Approximation Theory and Applications" of the Italian Mathematical Union.

V.S. would like to thank the Campania Region, for his scholarship from *POR Campania FSE*, Italy.

Appendix A. Supplementary material

This work is based on a Python package, available on GitHub: https://github.com/MthBr/DEDICATE

The code was developed in order to generate Table 1, and compare the performances of different types of Dirac Delta regularization. Tests can be extended with more delta approximations and other scaled supports $k \in \mathbb{R}$. The reader can tune the tests with other implemented solvers, or with their own solvers, or evaluate the performance on different PDEs that can also be time-dependent.

References

[1] Severino G. Macrodispersion by point-like source flows in randomly heterogeneous porous media. Transp Porous Media 2011;89(1):121–34. http://dx.doi.org/10.1007/s11242-011-9758-1.
[2] Severino G, Santini A, Sommella A. Macrodispersion by diverging radial flows in randomly heterogeneous porous media. J Contam Hydrol 2011;123(1):40–9. http://dx.doi.org/10.1016/j.jconhyd.2010.12.005.
[3] Fallico C, De Bartolo S, Veltri M, Severino G. On the dependence of the saturated hydraulic conductivity upon the effective porosity through a power law model at different scales. Hydrol Process 2016;30(13):2366–72. http://dx.doi.org/10.1002/hyp.10798.
[4] Severino G, Santini A. On the effective hydraulic conductivity in mean vertical unsaturated steady flows. Adv Water Resour 2005;28:964–74. http://dx.doi.org/10.1016/j.advwatres.2005.03.003.
[5] Severino G, Cuomo S. Uncertainty quantification of unsteady flows generated by line-sources through heterogeneous geological formations. SIAM/ASA J Uncertain Quant 2020;8(2):807–25. http://dx.doi.org/10.1137/19M1288966.
[6] Yang X, Zhang X, Li Z, He G-W. A smoothing technique for Discrete Delta functions with application to immersed boundary method in moving boundary simulations. J Comput Phys 2009;228(20):7821–36. http://dx.doi.org/10.1016/j.jcp.2009.07.023.
[7] Engquist B, Tornberg A-K, Tsai R. Discretization of Dirac delta functions in level set methods. J Comput Phys 2005;207(1):28–51. http://dx.doi.org/10.1016/j.jcp.2004.09.018.
[8] Lee HG, Kim J. Regularized Dirac delta functions for phase field models. Internat J Numer Methods Engrg 2012;91(3):269–88. http://dx.doi.org/10.1002/nme.4262.
[9] Hosseini B, Nigam N, Stockie JM. On regularizations of the Dirac delta distribution. J Comput Phys 2016;305:423–47. http://dx.doi.org/10.1016/j.jcp.2015.10.054.
[10] Liu Y, Mori Y. Properties of Discrete Delta functions and local convergence of the immersed boundary method. SIAM J Numer Anal 2012;50(6):2986–3015. http://dx.doi.org/10.1137/110836699.
[11] Bao Y, Kaiser AD, Kaye J, Peskin CS. Gaussian-like immersed boundary kernels with three continuous derivatives and improved translational invariance. J Comput Phys 2016;316:139–44. http://dx.doi.org/10.1016/j.jcp.2016.04.024.
[12] Tornberg A-k, Engquist B. Numerical approximations of singular source terms in differential equations. J Comput Phys 2003;200:462–88.
[13] Xuan L-J, Majdalani J. A point-value enhanced finite volume method based on Approximate Delta functions. J Comput Phys 2018;355:37–58. http://dx.doi.org/10.1016/j.jcp.2017.10.059.
[14] Indelman P. Steady-state source flow in heterogeneous porous media. Transp Porous Media 2001;45(1):105–27. http://dx.doi.org/10.1023/A:1011814418142.
[15] Guyer JE, Wheeler D, Warren JA. FiPy: partial differential equations with python. Comput Sci Eng 2009;11(3):6–15. http://dx.doi.org/10.1109/MCSE.2009.52.
[16] Brannick J, Chen Y, Hu X, Zikatanov L. Parallel unsmoothed aggregation algebraic multigrid algorithms on GPUs. In: Iliev OP, Margenov SD, Minev PD, Vassilevski PS, Zikatanov LT, editors. Numerical solution of partial differential equations: theory algorithms, and their applications. New York, NY: Springer New York; 2013, p. 81–102. http://dx.doi.org/10.1007/978-1-4614-7172-1_5.
[17] Balay S, Gropp W, McInnes LC, Smith BF. PETSC, the portable, extensible toolkit for scientific computation. Argon Natl Lab 1998;2(17).