Applied Mathematics and Computation manuscript No.
(will be inserted by the editor)

Highlighting numerical insights of an accurate and fast SPH Method

E.Francomano and M.Paliaga

submitted January 10, 2018

Abstract In this paper we focus on two sources of enhancement in accuracy and computational demanding in approximating a function and its derivatives by means of the Smoothed Particle Hydrodynamics method. The approximating power of the standard method is perceived to be poor and improvements can be gained making use of the Taylor expansion of the kernel approximation of the function and its derivatives. The modified formulation is appealing providing more accurate results of the function and its derivatives simultaneously without changing the kernel function adopted in the computation. The scheme received attention from practitioners, but many fundamental issues are still widely open. In this paper we highlight numerical insights of the scheme: studies on the accuracy, the convergence rate and the computational efforts with various data sites are provided. Accuracy of arbitrary order can be reached by employing the derivatives of the kernel with order up to the desired accuracy in approximating the function and with higher order for its derivatives. An infinitely differentiable kernel function, smooth even for high order derivatives, such as the Gaussian, is a suitable choice to successfully provide any order of accuracy for the function approximation or its derivatives. However, the improved formulation requires many summations on the kernel function and its derivatives, strongly limiting the feasibility of the method. Motivated to speed up the computation and to make large scale problems tractable we adopt fast summation approach. Namely, the improved fast Gaussian transform has been opportunely adapted to assembly the corrective linear system for each evaluation point picking up the computational cost at an acceptable level preserving the accuracy. Numerical experiments with various bivariate functions and sets of data are reported to give evidence of the accuracy, convergency and computational demanding.

E.Francomano and M. Paliaga
University of Palermo - Polytechnic School - Department of Industrial and Digital Innovation
Gruppo Nazionale per il Calcolo Scientifico - Istituto Nazionale di Alta Matematica (INDAM)

Keywords Kernel based methods · Smoothed particle hydrodynamics · Accuracy · Convergence · Improved Fast Gaussian Transform

1 Introduction

In recent years, meshless methods have gained growing interest in many different areas of science [2,4,8,9,19,21,40]. The basic idea of these methods is to provide numerical solutions without using any mesh in the problem domain. Methods without a predefinite connections are easily adapted to domains with complex and/or time evolving geometries without the difficulties that would be required to handle those features with topological data structures. They can be useful in non-linear problems involving viscous fluids, heat and mass transfer, linear and non-linear elastic or plastic deformations, etc. In the Lagrangian approach the points, describing the problem domain, move with the medium, and points may be added or deleted in order to maintain a prescribed sampling density. In the Eulerian approach the points are fixed in space, but new points may be added where there is need for increased accuracy. So, in both approaches the nearest neighbors of a point are not set. Numerical simulations usually need the values of a function and its derivatives at certain point and in this paper we focus on their approximation by means of the Smoothed Particle Hydrodynamics (SPH) method. This method was originally developed for solving astrophysical problems [15,16,29,31] and subsequently it has been also used in other areas of science and engineering [1,22–24,26,12,13,34,37,38,42,43]. The method results very attractive but it suffers from several drawbacks due to inaccurate approximation at boundaries and at irregular interior regions. This often confines its utility to limited scenarios. Many techniques have been devised to alleviate these problems and some of these have been documented in [5,25,27,28] and in the references therein. In this paper we discuss on sources of enhancement in accuracy of the standard SPH method via Taylor expansion of the kernel approximation of a function and its derivatives [27]. In this way, accurate estimates of the function and its derivatives are simultaneously provided and no lower order derivatives are inherent in approximating the higher order derivatives. Therefore, the possible numerical errors in a lower order derivative will not be brought to the higher order ones. Moreover, high order of accuracy can be obtained without changes on the kernel function avoiding to lead unphysical results such as negative density or negative energy that can lead to breakdown of the entire computation in simulating some problems [23]. Accuracy of arbitrary order can be reached for a function and its derivatives by employing the derivatives of the kernel function with order up to the accuracy order in approximating the function or higher when accurate results for the derivatives of the function are requested. So an infinitely differentiable and adequately smooth function even for higher order derivatives, such as the Gaussian, is a suitable choice to restore any order of accuracy. In our study we propose numerical investigations on the standard and improved method dealing with the Gaussian kernel function. Many experiments are conducted with the aim to address numerical features of the method accomplished with various data sets locations, gridded and scattered in a unit square domain, referring to bivariate test functions [36,41]. The modified approach is very interesting for the applications, but the computational demanding is a bottleneck as data locations get finer and for an high number of evaluation

points. Increasing the accuracy demanding the computational effort increases and it is essentially due to the summations of kernel and its derivatives in assembling the matrix of the solving system for each evaluation point. Motivated to speed up the computation and to make large scale problems tractable we focus on efficient processings of this fundamental task. Dealing with the infinitely differentiable Gaussian function as kernel one, the derivatives involve sums of products of the polynomials and Gaussian one. This allows us to take advantage in the computation and to make use of fast algorithms [18] for all the summations set out for the proposed strategy. Namely, we consider the improved fast Gaussian transform [35] picking up the computational cost at an acceptable level preserving the accuracy of the computation. Furthermore, the matrix and the known vector assembly is generated by taking into account the Gaussian function as common element in the fundamental tasks. The overall computational work performs to linear for a fixed level of accuracy. We present the computations with the direct and the improved fast summation algorithm showing satisfactory results referring to a bivariate case study. The remainder of the paper is as follows. In section 2 we present a review of the standard formulation. In section 3 we describe the improved method supported by numerical simulations for some test functions in a unit square domain. In this section some discussions on the errors versus the number of data are reported referring to different orders of accuracy and with different data sets. The section 4 is devoted to computational topics presenting the direct and the fast sum computation via Improved Fast Transform Gaussian adapted for our purposes. In section 5 the results presented in the paper are shortly summarized.

2 Ab initio formulation

To make the paper self-contained we briefly review the SPH standard formalism from first principles. The method makes use of a *kernel approximation* using ideas from distribution theory for approximating a function with a delta distribution representation [14].

Definizione 1. Let $f(\mathbf{x}) \in \mathbb{R}$, $\mathbf{x} = (x^{(1)}, ..., x^{(d)})$ and $\boldsymbol{\xi} = (\xi^{(1)}, ..., \xi^{(d)}) \in \mathbb{R}^d$, the *kernel approximation* is defined as

$$f_h(\mathbf{x}) := \int_\Omega f(\boldsymbol{\xi}) \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}; h) d\Omega. \tag{1}$$

The function $\mathsf{K}(\mathbf{x}, \boldsymbol{\xi}; h)$ is named *kernel function* and h is the *smoothing length*. The parameter h localizes the influence of the kernel function which approximates a Dirac δ-function in the limit $h \to 0$ and usually is normalized to unity. It is further required the kernel to be symmetric and sufficiently smooth. Under these assumptions the error of the kernel approximation can be estimated as second order of accuracy, or of first order of consistency [15, 16, 23]. Any function $\mathsf{K}(\mathbf{x}, \xi; h)$ with the above properties can be employed as smoothing kernel function. A common choice is the Gaussian function

$$\mathsf{K}(\mathbf{x}, \boldsymbol{\xi}; h) = \frac{1}{h^d \sqrt{\pi^d}} e^{-\frac{\|\boldsymbol{\xi} - \mathbf{x}\|_2^2}{h^2}}. \tag{2}$$

The kernel clearly decays when $\mathbf{x}$ moves away from $\boldsymbol{\xi}$ and the dimensional constant $\alpha_d = 1/h^d\sqrt{\pi^d}$ is to satisfy the unity requirements [23]. Moreover, it is infinitely differentiable, radial and strictly positive definite function on $\mathbb{R}^d$ for any d [7,10, 11,44] . This function will be taken into consideration as kernel from now on.
When the entire domain is represented by a finite number of data sites we proceed in the approximation as follows

Definizione 2. Given a set of data sites $\Xi = (\boldsymbol{\xi}_j)_{j=1}^N$ and the corresponding measurements $(y_j = f(\boldsymbol{\xi}_j))_{j=1}^N$ the *particle approximation* of the function is defined as

$$f_h(\mathbf{x}) := \sum_{j=1}^{N} f(\boldsymbol{\xi}_j)\mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h)d\Omega_j, \tag{3}$$

where $d\Omega_j$ is the measure of the subdomain Ω_j associated to each data site $\boldsymbol{\xi}_j$.

The triple (K, Ξ, h) essentially characterizes the approximation.

2.1 Some numerical behaviors of the standard SPH

In the following we discuss on the method by proposing various numerical experiments referring to the bivariate functions (4), depicted in Figure 1, originally proposed in the ACM Transaction Software Packages [36,41]

$$f(x^{(1)}, x^{(2)}) = 16x^{(1)}x^{(2)}(1 - x^{(1)})(1 - x^{(2)}), \tag{4a}$$

$$f(x^{(1)}, x^{(2)}) = \frac{1}{9}tanh(9(x^{(2)} - x^{(1)}) + 1), \tag{4b}$$

$$f(x^{(1)}, x^{(2)}) = \frac{1.25 + cos(5.4x^{(2)})}{6 + 6(3x^{(1)} - 1)^2}. \tag{4c}$$

We consider the Gaussian kernel and different data sets as centers of the kernel, taken in number as the progressions $(2^n + 1)^2$. Gridded, Halton, Sobol and random points are taken as data sequences and we will refer to these sets as $\Xi_G, \Xi_H, \Xi_S, \Xi_R$. The second and third ones are available in MATLAB$^{@}$ Statistics and Machine Learning Toolbox as *haltonset* and *sobolset*, respectively. The former was introduced by J.H. Halton [20] and the latter by I.M. Sobol [39]. The random points are generated by means of the random function of MATLAB$^{@}$. The evaluation points $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$ are on a regular mesh layed out over the computational domain and for all the simulations M=1600. In Figure 2 we show N=289 data sites for the four data sequences $\Xi_G, \Xi_H, \Xi_S, \Xi_R$.
The following formulas are adopted to estimate the accuracy of the solution

$$MAE := \max_{1 \le i \le M} |f_h(\mathbf{x}_i) - f(\mathbf{x}_i)|, \tag{5}$$

$$RMSE := \sqrt{\frac{\sum_{i=1}^{M} |f_h(\mathbf{x}_i) - f(\mathbf{x}_i)|^2}{M}}, \tag{6}$$

$$MEAN := \frac{\sum_{i=1}^{M} |f_h(\mathbf{x}_i) - f(\mathbf{x}_i)|}{M}, \tag{7}$$

and the errors are plotted in a loglog scale throughout the paper. In this section the discussion focuses on the function test (4a). First of all we give evidence of the influence of h on the approximation. To this aim we perform a series of experiments with varying smoothing length h_Ξ for the data sets of Figure 2 and in Figure 3 we plot the MAE by fixing the number of data to $N = 1089$. In Table 1 we report the values of h_Ξ and the MAE, RMSE, MEAN error for the function test (4a) by considering $\Xi_G, \Xi_H, \Xi_S, \Xi_R$ with N=1089. The standard method usually does not yield to satisfactory results and by increasing the data density in Ω the accuracy slightly improves for the first few increased values as we observe in Figure 4.

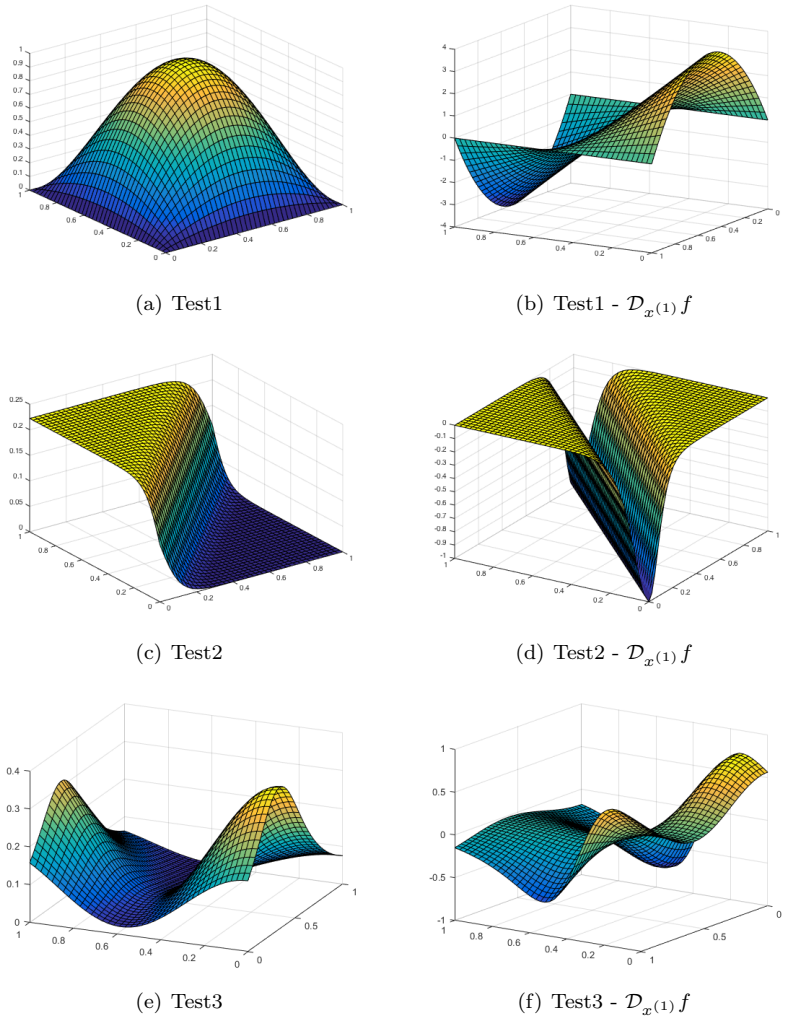
Table 1 MAE, RMSE, MEAN error with h_Ξ - N=1089 - Function test (4a)

	MAE	RMSE	MEAN	h_Ξ
gridded data	0.0302	0.0104	0.0072	0.0214
Halton data	0.0753	0.0214	0.0162	0.0680
Sobol data	0.0624	0.0184	0.0124	0.0560
random data	0.2157	0.0353	0.0226	0.0570

The approximation of the derivatives via standard SPH is also not good enough. We test the approximation on a uniform grid with increasing data density in the unit square domain and the results are displayed in Figure 5. Analogous conclusions are reached with the data sets Ξ_H, Ξ_S, Ξ_R and by approximating the derivatives along the $x^{(2)}$ direction.
The numerical simulations assess that the approximation with the standard formulation is poor and that it is not according with the second order of accuracy claimed in the kernel approximation. This is mainly due to the discrepancy between the kernel and particle approximation especially on the boundary and with non uniform data locations. With the goal to improve the approximation of the function and its derivatives, we consider series expansion of the function, projecting with the kernel function and its derivatives and integrating over the problem domain [27] . This approach improves the accuracy without changes on the kernel function. Moreover, the function and its derivatives are simultaneously computed, for each evaluation point, with the desired order of accuracy. Hence, high order of derivatives are generated independently on the lower one. In the next section we discuss on the basic idea of the improved approach coupled with some validation results.

3 Enhancing particle approximation

To ensure the 1-*st* order of accuracy (or the 0-*th* order of consistency) we consider the Taylor expansion of $f(\boldsymbol{\xi})$, retaining only the first term, multiplying for the



(a) Test1 (b) Test1 - $\mathcal{D}_{x^{(1)}} f$

(c) Test2 (d) Test2 - $\mathcal{D}_{x^{(1)}} f$

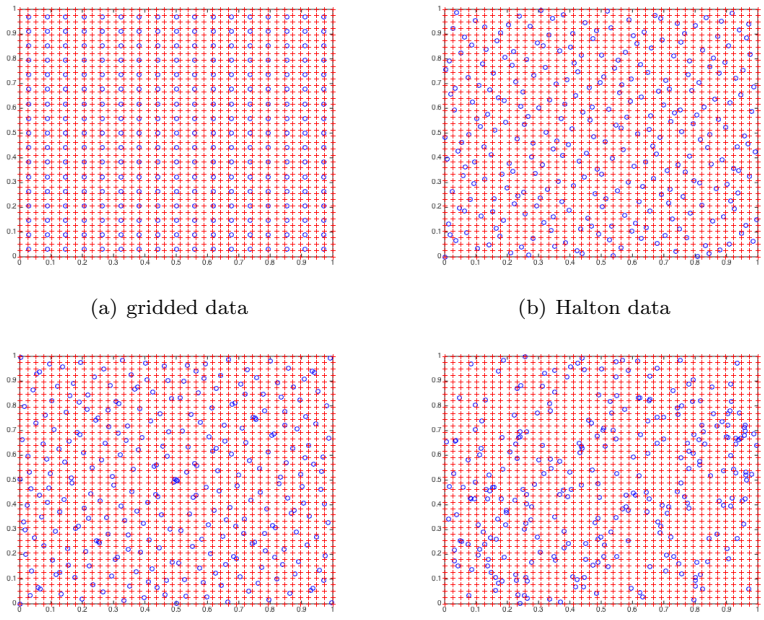
(e) Test3 (f) Test3 - $\mathcal{D}_{x^{(1)}} f$

Fig. 1 Test functions (left) and their first derivatives (right) used in the numerical experiments.

kernel function and integrating over Ω

$$\int_{\Omega} f(\boldsymbol{\xi})\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega = \int_{\Omega} f(\mathbf{x})\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega + \int_{\Omega} O(h)\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega, \tag{8}$$

$$f(\mathbf{x}) = \frac{\int_{\Omega} f(\boldsymbol{\xi})\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega}{\int_{\Omega} \mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega} + O(h). \tag{9}$$



(a) gridded data (b) Halton data

(c) Sobol data (d) random data

Fig. 2 The blue circles as the set of $N = 289$ data sites $\boldsymbol{\xi}_j$; the red crosses as the set of M=1600 evaluation points $\mathbf{x}_i$.

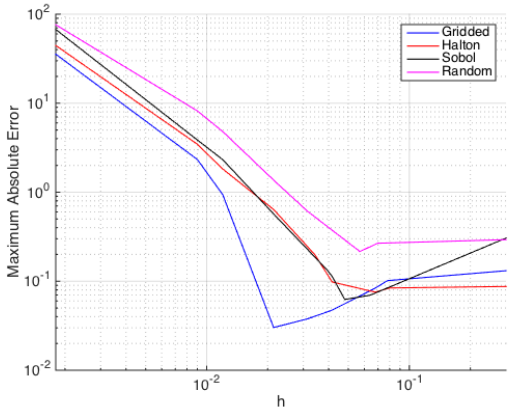
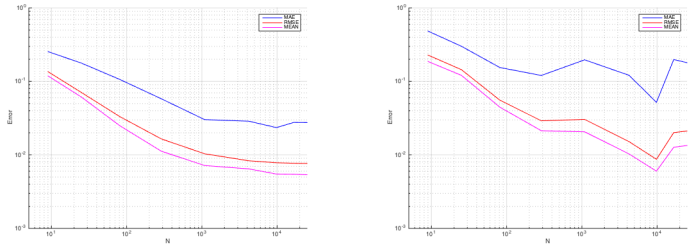


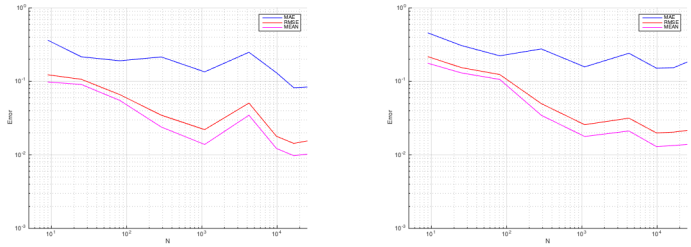
Fig. 3 Maximum absolute error vs. h for N=1089 data sites in $\Xi_G, \Xi_H, \Xi_S, \Xi_R$.

The corresponding discrete formulation is

$$f(\mathbf{x}) = \frac{\sum_{j=1}^{N} f(\boldsymbol{\xi}_j)\mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h)d\Omega_j}{\sum_{j=1}^{N} \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h)d\Omega_j} + O(h). \tag{10}$$



(a) Gridded data (b) Halton data



(c) Sobol data (d) Random data

Fig. 4 Error versus the number of data in $\Xi_G, \Xi_H, \Xi_S, \Xi_R$. Function test (4a).

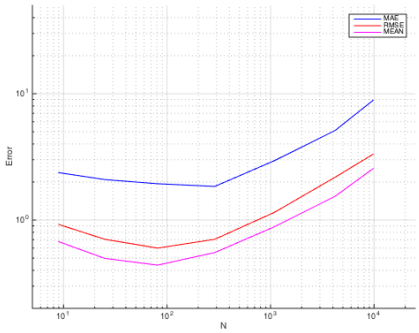
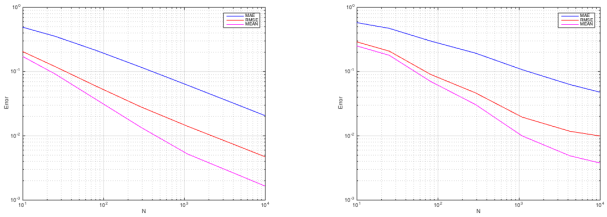


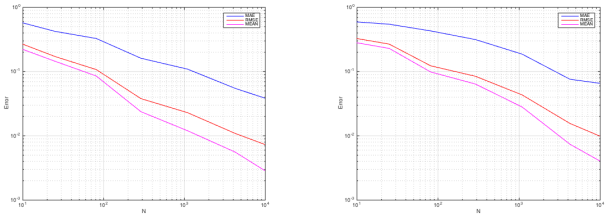
Fig. 5 Error versus the number of data in Ξ_G for $\mathcal{D}_{x^{(1)}} f$. Function test (4a).

We proceed in the approximation of the function test (4a), for the data sets $\Xi_G, \Xi_H, \Xi_S, \Xi_R$ and by increasing the density of the data in the unit square domain. The simulations are summarized in Figure 6.
The comparison between the plots in Figure 4 and in Figure 6 gives evidence of the improvements in the approximation. In Tables 2 and 3, the errors, the rate of convergence and the values of h_Ξ for the first order of accuracy (k=0) with gridded and random data sets are reported. The results confirm the theoretical assumptions making evidence of a less accurate approximation for data in Ξ_R. In

Figure 7 we summarize the maximum absolute error for the test functions (4b) and (4c) for the different data sequences $\Xi_G, \Xi_H, \Xi_S, \Xi_R$.



(a) k=0 Gridded data (b) k=0 Halton data



(c) k=0 Sobol data (d) k=0 Random data

Fig. 6 k=0. Error versus the number of data in $\Xi_G, \Xi_H, \Xi_S, \Xi_R$. Function test (4a).

Table 2 k=0. MAE, RMSE, MEAN error for data in Ξ_G. Function test (4a). Data used in the Figure 6(a).

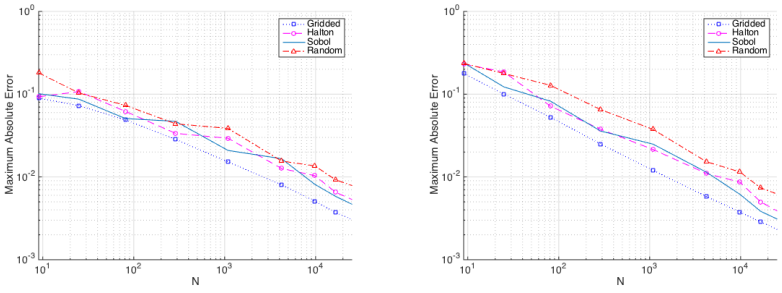
N	f						h_Ξ
	MAE	*rate*	RMSE	*rate*	MEAN	*rate*	
9	0.5097	–	0.2178	–	0.1848	–	0.2357
25	0.3556	0.7046	0.1202	1.1627	0.0922	1.3603	0.1414
81	0.2136	0.8672	0.0592	1.2047	0.0367	1.5642	0.0786
289	0.1175	0.9402	0.0282	1.1629	0.0136	1.5640	0.0415
1089	0.0613	0.9723	0.0141	1.0435	0.0052	1.4407	0.0214
4225	0.0315	0.9865	0.0072	0.9898	0.0025	1.0399	0.0109
9801	0.0207	0.9926	0.0047	0.9946	0.0016	1.0460	0.0071
16641	0.0160	0.9951	0.0037	0.9950	0.0013	0.9931	0.0050
66049	0.0080	0.9968	0.0018	0.9971	6.40e-04	1.0046	0.0028

3.1 Higher order of accuracy

Now we extend the idea presented in the previous section to generate methods with higher approximation order taking into consideration the Taylor expansion

Table 3 k=0. MAE, RMSE, MEAN error for data in Ξ_R. Function test (4a). Data used in the Figure 6(d).

N	f						h_Ξ
	MAE	*rate*	RMSE	*rate*	MEAN	*rate*	
9	0.5480	–	0.2768	–	0.2381	–	0.4679
25	0.5169	0.6975	0.2539	0.2430	0.2213	0.2061	0.3277
81	0.4739	0.3440	0.1284	1.2918	0.1046	1.4182	0.1932
289	0.2500	0.5643	0.0743	0.9011	0.0543	1.0825	0.1055
1089	0.1008	0.7928	0.0208	1.2497	0.0201	1.5628	0.0780
4225	0.0859	1.0513	0.0337	1.1413	0.0084	1.5239	0.0314
9801	0.0616	0.6447	0.0101	0.9963	0.0041	1.2860	0.0182
16641	0.0479	2.3055	0.0085	1.4690	0.0033	1.9937	0.0162
66049	0.0212	0.8653	0.0037	0.8997	0.0015	0.8487	0.0063



(a) (b)

Fig. 7 k=0. MAE versus number of data in $\Xi_G, \Xi_H, \Xi_S, \Xi_R$. (a) Function test (4b) and (b) function test (4c).

of the function $f(\boldsymbol{\xi})$ up to the order k

$$f(\boldsymbol{\xi}) = \sum_{|\alpha|\leq k} \frac{1}{\alpha!}(\boldsymbol{\xi} - \mathbf{x})^{\alpha} \mathcal{D}^{\alpha} f(\mathbf{x}) + O(h^{k+1}), \tag{11}$$

where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_d) \in \mathbb{N}^d$ is a multi-index with $|\alpha| = \sum_{i=1}^{d} \alpha_i$ and the differential operator is defined as

$$\mathcal{D}^{\alpha} := \frac{\delta^{|\alpha|}}{(\delta x^{(1)})^{\alpha_1} ... (\delta x^{(d)})^{\alpha_d}}.$$

Hence, the (11) is multiplied for the kernel function and its derivatives up to the

order k and integrated over Ω

$$\begin{aligned}\int_\Omega f(\boldsymbol{\xi})\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega = \sum_{|\alpha|\le k}\frac{1}{\alpha!}\int_\Omega(\boldsymbol{\xi}-\mathbf{x})^\alpha\mathcal{D}^\alpha f(\mathbf{x})\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega+\\ +\int_\Omega O(h^{k+1})\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega\\ \vdots\\ \int_\Omega f(\boldsymbol{\xi})\mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega = \sum_{|\alpha|\le k}\frac{1}{\alpha!}\int_\Omega(\boldsymbol{\xi}-\mathbf{x})^\alpha\mathcal{D}^\alpha f(\mathbf{x})\mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega+\\ +\int_\Omega O(h^{k+1})\mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega.\end{aligned}$$

Neglecting the error, in linear algebra notation the improved formulation corresponds to find the solution of the pointwise linear systems

$$\mathbf{A}^{(k)}\mathbf{c}^{(k)} = \mathbf{b}^{(k)} \tag{12}$$

where

$$\mathbf{A}^{(k)} = \begin{pmatrix} \int_\Omega \mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega & \dots & \frac{1}{k!}\int_\Omega \mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)(\xi^{(d)}-x^{(d)})^k d\Omega \\ \vdots & \dots & \vdots \\ \int_\Omega \mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega & \dots & \frac{1}{k!}\int_\Omega \mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)(\xi^{(d)}-x^{(d)})^k d\Omega \end{pmatrix}$$

$$\mathbf{c}^{(k)} = \begin{pmatrix} f(\mathbf{x}) \\ \vdots \\ \mathcal{D}^k_{x^{(d)}} f(\mathbf{x}) \end{pmatrix}$$

$$\mathbf{b}^{(k)} = \begin{pmatrix} \int_\Omega f(\boldsymbol{\xi})\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega \\ \vdots \\ \int_\Omega f(\boldsymbol{\xi})\mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x},\boldsymbol{\xi};h)d\Omega \end{pmatrix}.$$

We are now ready to propose the discrete formulation for the function and its derivatives estimate up to order k at the evaluation point $\mathbf{x}$

$$\mathbf{A}^{(k)} = \begin{pmatrix} \sum_{j=1}^N \mathsf{K}(\mathbf{x},\boldsymbol{\xi}_j;h)d\Omega_j & \dots & \frac{1}{k!}\sum_{j=1}^N \mathsf{K}(\mathbf{x},\boldsymbol{\xi}_j;h)(\xi_j^{(d)}-x^{(d)})^k d\Omega_j \\ \vdots & \dots & \vdots \\ \sum_{j=1}^N \mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x},\boldsymbol{\xi}_j;h)d\Omega_j & \dots & \frac{1}{k!}\sum_{j=1}^N \mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x},\boldsymbol{\xi}_j;h)(\xi_j{}^{(d)}-x^{(d)})^k d\Omega_j \end{pmatrix}$$

$$\mathbf{c}^{(k)} = \begin{pmatrix} f(\mathbf{x}) \\ \vdots \\ \mathcal{D}^k_{x^{(d)}} f(\mathbf{x}) \end{pmatrix}$$

$$\mathbf{b}^{(k)} = \begin{pmatrix} \sum_{j=1}^{N} f(\boldsymbol{\xi}_j) \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h) d\Omega_j \\ \vdots \\ \sum_{j=1}^{N} f(\boldsymbol{\xi}_j) \mathcal{D}^k_{\xi^{(d)}} \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h) d\Omega_j \end{pmatrix}.$$

The described process gives more accurate values for the function and its derivatives [27]: it is accurate of order k+1 for the function f and of order (k+1)-p for the derivatives of order p. In the following we report some results for the function test (4a) in approximating the function and its derivatives with k=1 and k=2. In Tables 4 and 5 the results are with k=1 whilst in Tables 6 and 7 we report the results with k =2. These illustrate the approximation does converge well, almost reaching the results predicted by the theory.
In Figure 8, by fixing N=1089, we focus on the MAE comparing SPH and the improved method with k=0, k=1 and k=2. We observe that the error is reduced inside the domain and it is always present on the boundaries but a significant decrease is with k=1,2. We conclude this section presenting the MAE for finer data locations concerning the three functions (4a), (4b), (4c) with the standard SPH formulation and with the modified one by varying k=0,1,2. In Figures 9, 10 and 11 we show the maximum absolute error versus the number of data in Ξ_G, Ξ_H, Ξ_S, Ξ_R.

Table 4 k=1. MAE, RMSE for data in Ξ_G. Function test (4a).

N	f				$\mathcal{D}_{x^{(1)}} f$				h_Ξ
	MAE	$rate$	$RMSE$	$rate$	MAE	$rate$	$RMSE$	$rate$	
9	0.3016	–	0.1424	–	2.7931	–	1.0237	–	0.2357
25	0.1158	1.8745	0.0563	0.1818	1.6832	0.9914	0.5379	1.2598	0.1414
81	0.0366	1.9596	0.0183	1.9120	0.9237	1.0207	0.2512	1.2949	0.0786
289	0.0114	1.8258	0.0052	1.9690	0.4825	1.0211	0.1131	1.2540	0.0415
1089	0.0035	1.8001	0.0014	1.9784	0.2461	1.0145	0.0530	1.1431	0.0214
4225	0.0009	1.9052	0.0003	1.9909	0.1240	1.0111	0.0267	1.0089	0.0109
9801	0.0004	1.9476	0.0001	1.9987	0.0815	0.9965	0.0175	1.0064	0.0071
16641	0.0002	1.9635	9.27e-05	1.9996	0.0629	0.9769	0.0134	1.0002	0.0050
66049	6.39e-05	1.9771	2.33e-05	1.9994	0.0319	0.9855	0.0067	0.9999	0.0028
96721	4.38e-05	1.9857	1.57e-05	2.0526	0.0264	0.9909	0.0056	0.9264	0.0023

Table 5 k=1. MAE, RMSE for data in Ξ_R. Function test (4a).

N	f				$\mathcal{D}_{x^{(1)}} f$				h_Ξ
	MAE	$rate$	$RMSE$	$rate$	MAE	$rate$	$RMSE$	$rate$	
9	0.6282	–	0.3162	–	3.7270	–	1.5004	–	0.4358
25	0.4973	0.4679	0.1643	1.3109	2.7106	0.7505	1.0782	0.7787	0.2666
81	0.1304	1.8281	0.0560	1.4704	1.7442	1.2256	0.6148	1.5616	0.1271
289	0.0678	1.1989	0.0199	1.8844	0.9268	0.9186	0.2709	1.1906	0.0721
1089	0.1086	5.2991	0.0230	1.6080	0.6175	0.8948	0.1496	1.3079	0.0343
4225	0.0821	3.2346	0.0156	1.1541	0.3152	0.6976	0.0476	1.1869	0.0219
9801	0.0065	2.6831	0.0029	1.9517	0.2762	0.6840	0.0377	1.2107	0.0219
16641	0.0082	-0.9494	0.0018	1.9809	0.1579	1.4026	0.0248	1.0440	0.0189
66049	0.0029	1.8465	0.0006	1.9994	0.0676	1.1567	0.0119	1.0013	0.0103
96721	0.0003	2.5147	0.0001	1.9563	0.0794	1.5995	0.0129	0.7976	0.0044

Table 6 k=2. MAE for data in Ξ_G. Function test (4a).

N	f		$\mathcal{D}_{x^{(1)}} f$		$\mathcal{D}^2_{x^{(1)}} f$		h_Ξ
	MAE	$rate$	MAE	$rate$	MAE	$rate$	
9	0.2646	–	1.9476	—	4.5083	–	0.2357
25	0.0741	2.4920	0.0878	1.5591	2.9490	0.8308	0.1414
81	0.0146	2.7587	0.3074	1.7855	1.7287	0.9086	0.0786
289	0.0023	2.8822	0.0921	1.8941	0.9433	0.9523	0.0415
1089	3.32e-04	2.9417	0.0253	1.9473	0.4938	0.9756	0.0214
4225	4.44e-05	2.9710	0.0066	1.9737	0.2528	0.9876	0.0109
9801	1.26e-05	2.9837	0.0028	1.9852	0.1664	0.9930	0.0071
16641	5.73e-06	2.9885	0.0017	1.9895	0.1279	0.9950	0.0055
66049	7.29e-07	2.9954	4.30e-04	1.9934	0.0643	0.9532	0.0028
96721	4.11e-07	2.9954	2.94e-04	1.9959	0.0532	0.9673	0.0023

Table 7 k=2. MAE for data in Ξ_R. Function test (4a).

N	f		$\mathcal{D}_{x^{(1)}} f$		$\mathcal{D}^2_{x^{(1)}} f$		h_Ξ
	MAE	*rate*	MAE	*rate*	MAE	*rate*	
9	53.055	–	2.2001	—	3.1221	—	0.5868
25	32.312	0.7001	0.9378	1.2038	0.9374	1.6073	0.2890
81	3.0333	4.7134	0.0998	4.4629	0.2017	1.4525	0.1749
289	0.4713	3.9067	0.0180	3.5927	0.0733	1.4026	0.1086
1089	0.0158	4.7237	8.06e-04	4.3248	0.0567	1.4021	0.0529
4225	0.2694	14.696	0.0067	-11.009	0.0348	1.4116	0.0436
9801	0.0010	6.9243	3.68e-05	6.5451	0.0041	1.4069	0.0197
16641	9.76e-04	0.2729	2.91e-05	0.5896	5.36e-04	1.4075	0.0132
66049	3.01e-04	2.6961	7.79e-06	3.0239	5.66e-04	1.3876	0.0085
96721	4.28e-05	4.3213	1.24e-06	4.0639	4.76e-05	1.3564	0.0054

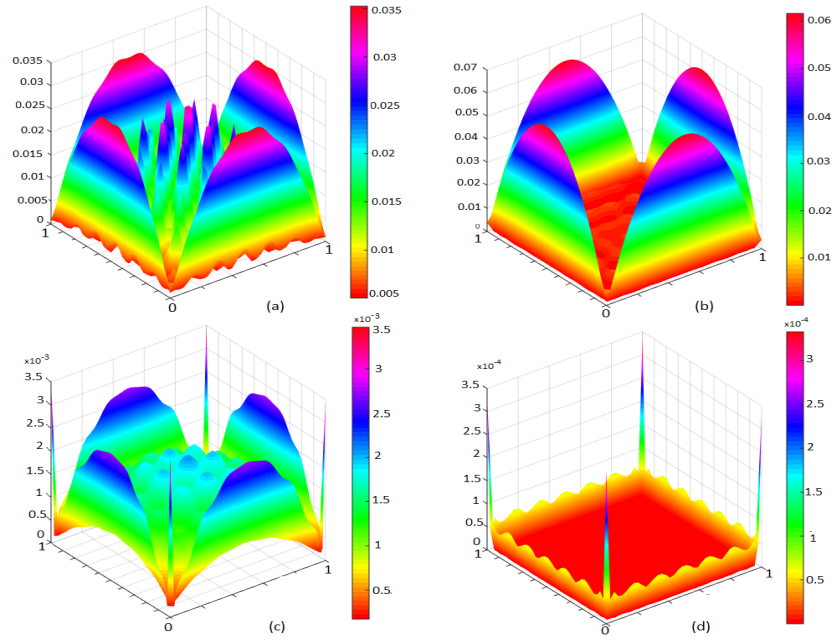
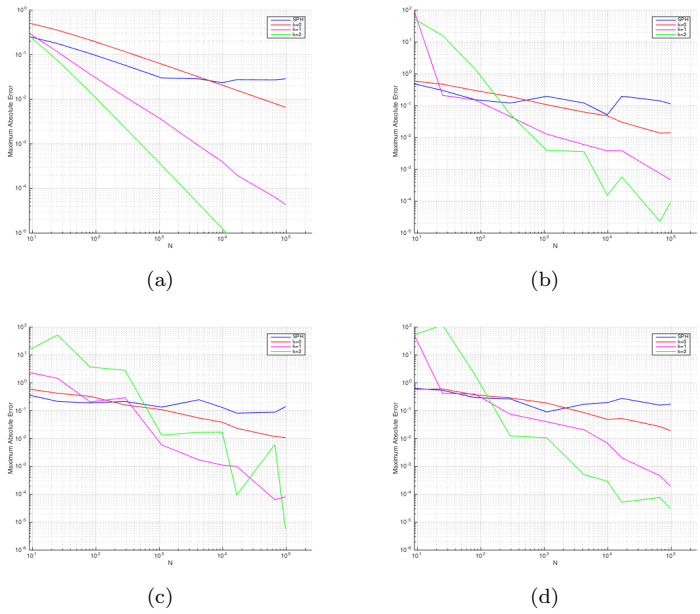


Fig. 8 MAE for (a) SPH standard; (b) Improved method with k=0, (c) k=1, (d) k=2. N=1089 data in Ξ_G. Function test (4a).

4 Computational skills

The computational strategy proposed is more expensive than the standard one. As detailed in the previous section, fixed k, the linear system (12) have to be generated and solved for each evaluation point $\mathbf{x}$. The size of the linear system (12) depends on the domain dimension d and on k, i.e. $s_{k,d}=\frac{(d+k)!}{d!k!}$ and the overall



(a) (b) (c) (d)

Fig. 9 Function test (4a). Comparison of MAE on the function with the standard SPH formulation and the improved ones k=0,1,2. (a) Gridded data; (b) Halton data; (c) Sobol; (d) Random data.

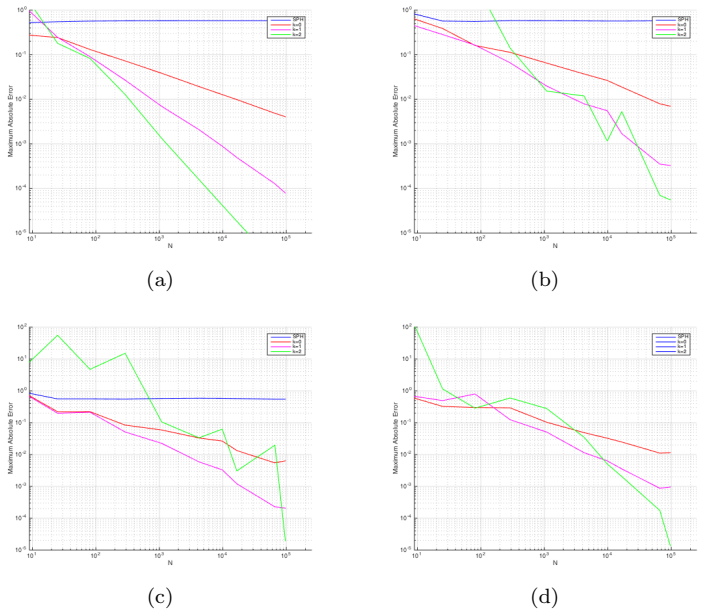
computational effort is

$$C_{k,d} \approx MNs_{k,d}\left(\frac{s_{k,d}+3}{2}\right). \tag{13}$$

We notice that the matrix $\mathbf{A}^{(k)}$ is partially composed by the entries of the matrices $\mathbf{A}^{(j)}$,with $j = 0, ..., k-1$, as shown in Figure 12 and $\frac{d}{2k}s_{k-1,d}[\frac{(d+2k)}{k}s_{k-1,d}+1]$ more elements need to end up the matrix $\mathbf{A}^{(k)}$ starting from the matrix $\mathbf{A}^{(k-1)}$. This computational extra effort is required to generate the system matrix if we desire the $(k+1)$-th order of accuracy only on the boundary or on subdomains where the data are less regularly distributed.
The matrices $\mathbf{K}^{(k)}, \mathbf{V}, \mathbf{P}^{(k)}$

$$\mathbf{K}^{(k)} = \begin{pmatrix} \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_1; h) & \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_2; h) & \dots & \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_N; h) \\ \vdots & \vdots & \dots & \vdots \\ \mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_1; h) & \mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_2; h) & \dots & \mathcal{D}^k_{\xi^{(d)}}\mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_N; h) \end{pmatrix} \tag{14}$$

$$\mathbf{V} = \begin{pmatrix} d\Omega_1 & & & \\ & d\Omega_2 & & \\ & & \ddots & \\ & & & d\Omega_N \end{pmatrix} \tag{15}$$



(a) (b) (c) (d)

Fig. 10 Function test (4b). Comparison of MAE on the function with the standard SPH formulation and the improved ones k=0,1,2. (a) Gridded data; (b) Halton data; (c) Sobol; (d) Random data.

$$\mathbf{P}^{(k)} = \begin{pmatrix} 1 & (\xi_1^{(1)} - x^{(1)}) & \dots & \frac{1}{k!}(\xi_1^{(d)} - x^{(d)})^k \\ 1 & (\xi_2^{(1)} - x^{(1)}) & \dots & \frac{1}{k!}(\xi_2^{(d)} - x^{(d)})^k \\ \vdots & \vdots & \dots & \vdots \\ 1 & (\xi_N^{(1)} - x^{(1)}) & \dots & \frac{1}{k!}(\xi_N^{(d)} - x^{(d)})^k \end{pmatrix} \tag{16}$$

allow us to rewrite the system as

$$(\mathbf{K}^{(k)}\mathbf{V}\mathbf{P}^{(k)})\mathbf{c}^{(k)} = \mathbf{K}^{(k)}\mathbf{V}\mathbf{f} \tag{17}$$

where $\mathbf{f}$ collects the function values.
The result of this formulation is no different than finding each entry individually by means of inner-vector products (BLAS level 1) but, from a performance point of view, computing the system matrix as in (17) should be likely more efficient by taking advantages from matrix-matrix products (BLAS level 3) [17].
Anyway, in order to improve the computation in the next section we introduce a fast summation technique for the system matrix assembly.

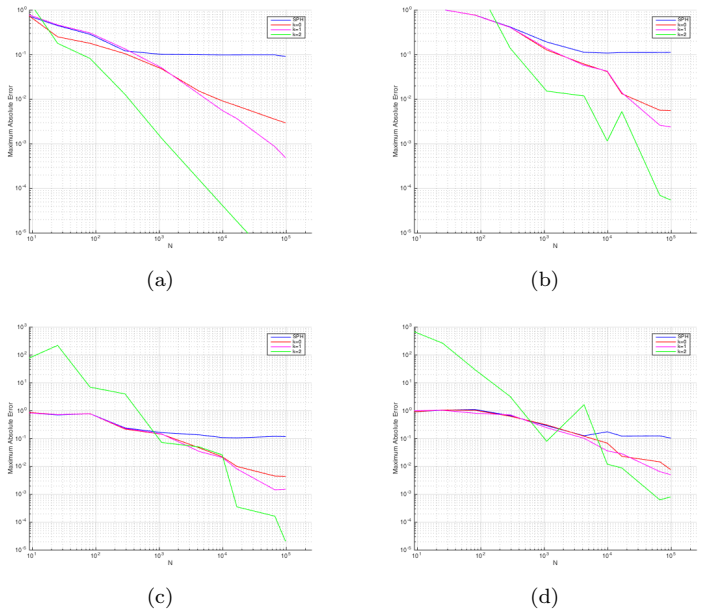


Fig. 11 Function test (4c). Comparison of MAE on the function with the standard SPH formulation and the improved ones k=0,1,2. (a) Gridded data; (b) Halton data; (c) Sobol; (d) Random data.

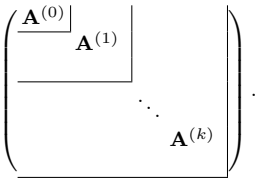


Fig. 12 Skeleton of the matrix associated to the $(k+1)$-th order of accuracy.

4.1 Fast computation via IFGT

The experiments presented in the previous section illustrate the approximation of large data sets with the Gaussian kernel function. This function is infinitely differentiable with high order derivatives sufficiently smooth employing itself as common element of all derivatives. The gained results are perceptible but a controversial aspect concerns the computational effort for the system matrix assembly which makes the procedure rather expensive and not easily approachable in the applications. The computation of summation appears as a bottleneck but a fast technique for M simultaneously evaluation sums of N Gaussian kernel functions allows to reduce the computational complexity from the standard $\mathcal{O}$(MN) to the $\mathcal{O}$(M+N). To this aim the Improved Fast Gaussian Transform (IFGT) offers

advantages, for uniform and non uniform data sites locations, allowing to tune the desidered accuracy. By opportunely combining the proposed approach with the IFGT we considerably speed up the method at a large number of evaluation points. A full description of the IFGT is beyond the scope of this paper and the interested reader is directed to [35] for an expository treatment of this tool. Each entry of the matrix $\mathbf{A}^{(k)}$ and of the known vector $\mathbf{b}^{(k)}$ need a summation and can be computed taking advantages by the IFGT with different weights depending on the derivatives of the kernel interested. The required algebra is somewhat tedious but straightforward and some computations can be re-used due to the Gaussian kernel employed in the formulation of the derivatives.
In the following we provide some computational details referring - for the sake of simplicity - to k=1 and d =2. In this case we need the six fundamental IFGTs

$$\mathbf{1}_{IFGT} := \sum_{j=1}^{N} d\Omega_j \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h)$$

$$\mathbf{2}_{IFGT} := \sum_{j=1}^{N} (d\Omega_j \xi_j^{(1)}) \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h)$$

$$\mathbf{3}_{IFGT} := \sum_{j=1}^{N} (d\Omega_j \xi_j^{(2)}) \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h)$$

$$\mathbf{4}_{IFGT} := \sum_{j=1}^{N} (d\Omega_j \xi_j^{(1)} \xi_j^{(1)}) \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h)$$

$$\mathbf{5}_{IFGT} := \sum_{j=1}^{N} (d\Omega_j \xi_j^{(1)} \xi_j^{(2)}) \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h)$$

$$\mathbf{6}_{IFGT} := \sum_{j=1}^{N} (d\Omega_j \xi_j^{(2)} \xi_j^{(2)}) \mathsf{K}(\mathbf{x}, \boldsymbol{\xi}_j; h)$$

these are coupled with different weights giving rise to the corrective matrix $\mathbf{A}^{(1)}$

$$A_{11}^{(1)} = \mathbf{1}_{IFGT}$$

$$A_{12}^{(1)} = \mathbf{2}_{IFGT} - x_i^{(1)} \mathbf{1}_{IFGT}$$

$$A_{13}^{(1)} = \mathbf{3}_{IFGT} - x_i^{(2)} \mathbf{1}_{IFGT}$$

$$A_{21}^{(1)} = \tfrac{2}{h^2} A_{12}^{(1)}$$

$$A_{22}^{(1)} = \tfrac{2}{h^2} [\mathbf{4}_{IFGT} + (x_i^{(1)})^2 \mathbf{1}_{IFGT} + 2x_i^{(1)} \mathbf{2}_{IFGT}]$$

$$A_{23}^{(1)} = \tfrac{2}{h^2} [\mathbf{5}_{IFGT} - x_i^{(2)} f2_{IFGT} - x_i^{(1)} \mathbf{3}_{IFGT} - x_i^{(1)} x_i^{(2)} \mathbf{1}_{IFGT}]$$

$$A_{31}^{(1)} = \tfrac{2}{h^2} A_{13}^{(1)}$$

$$A_{32}^{(1)} = \frac{2}{h^2} A_{23}^{(1)}$$

$$A_{33}^{(1)} = \frac{2}{h^2} [\mathbf{6}_{IFGT} + (x_i^{(2)})^2 \mathbf{1}_{IFGT} - 2x_i^{(2)} \mathbf{3}_{IFGT}].$$

The Figure 13 compares the cost of direct summation versus IFGT summation and shows the efficiency greatly improved by making use of the IFGT. We compare the CPU times which need to generate and to solve the linear system (12) with or without the IFGT for k=1 with the same approximation error referring to the test function (4a). The simulations are conducted on a computer equipped with a processor Intel(R) Core (TM) i7-3537U CPU 2.00GHz.

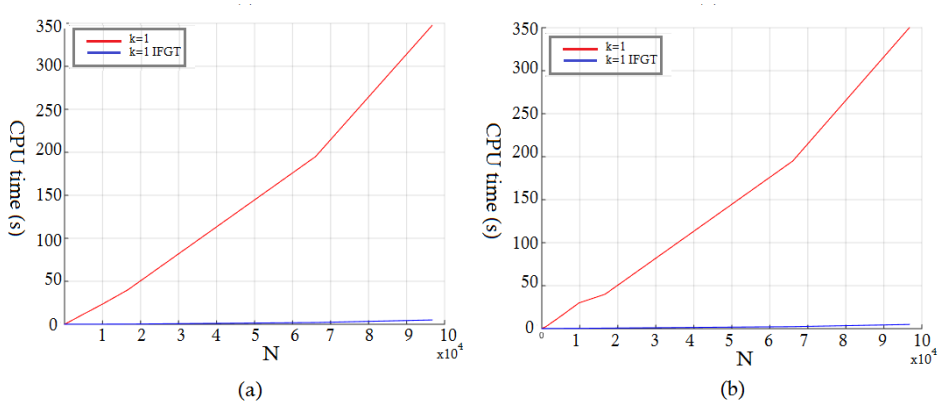


Fig. 13 CPU times (s) versus N. k=1: (a) Gridded data; (b) Random data.

5 Conclusions

In this paper we present numerical investigations of an improved SPH method based on Taylor series expansion. Dealing with the infinitely differentiable and smooth Gaussian kernel function any order of accuracy can be provided in approximating a function and its derivatives. We give evidence of the accuracy, convergency and computational demanding. Many experiments are conducted with the aim to address the basic features of the method accomplished with various data sets referring to bivariate test functions. By combining the proposed approach with the improved fast gaussian transform to assembly the system matrices we considerably speed up the method at a large number of evaluation points. Satisfactory results and computational advantages with uniform and non uniform large data sets encourage to proceed in the approximation of any accuracy order of a function

and its derivatives with the combined methodology.

References

1. G. Ala and E. Francomano, Numerical Investigations of an Implicit Leapfrog Time-Domain Meshless Method, Journal on Scientific Computing, Vol. 62(3), pp. 898–912 (2014).

2. G. Ala, E. Francomano, S. Ganci, G.E. Fasshauer and M. McCourt, The method of Fundamental Solutions in solving coupled boundary value problems for M/EEG, SIAM, Journal on Scientific Computing, Vol. 37(4), pp. B570–B590 (2015).

3. G.Ala, E.Francomano, S. Ganci, Unconditionally stable meshless integration of time-domain Maxwell's curl equations, Applied Mathematics and Computation, Vol. 255 (15), pp. 157-164 (2015).

4. G. Ala, E. Francomano, S. Ganci, G.E. Fasshauer, M. McCourt, An augmented MFS approach for brain activity reconstruction, Mathematics and Computers in Simulation, Vol. 141, pp. 3–15 (2017).

5. T. Belytschko, Y. Krongauz, J. Dolbow and C. Gerlach, On the completeness of meshfree methods, Int J Numer Methods Eng, Vol. 43, pp. 785–819 (1998).

6. J. Bonet and S. Kulasegaram, Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations, Int J Numer Methods Eng, Vol. 47, pp. 1189–1214 (2000).

7. M.D. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge Monogr. Appl. Comput. Math., Vol. 12, Cambridge University Press (2003).

8. X. Chen and J.H. Jung, Matrix stability of multiquadric radial basis function methods for hyperbolic equations with uniform centers, Journal of Scientific Computing, Vol. 51(3), pp. 683–702 (2012).

9. A. Chowdhury, A. Wittek, K. Miller and G.R. Joldes, An element free galerkin method based on the modified moving least squares approximation, Journal of Scientific Computing, pp. 1–15 (2016).

10. G.E. Fasshauer, Meshfree Approximation Methods with MATLAB, Interdiscip. Math. Sci., Vol. 6, World Scientific, Hackensack, NJ (2007).

11. G.E. Fasshauer and M. McCourt, Kernel-based Approximation Methods using MATLAB, Interdiscip. Math. Sci., Vol. 19, World Scientific (2015).

12. E. Francomano, F.M. Hilker, M. Paliaga and E. Venturino, An efficient method to reconstruct invariant manifolds of saddle points, Dolomites Research Notes on Approximation, Vol. 10, pp.25–30 (2017).

13. E. Francomano, F.M. Hilker, M. Paliaga and E. Venturino, Separatrix reconstruction to identify tipping points in an eco-epidemiological model, Applied Mathematics and Computation, Vol. 318, pp. 80-91 (2018).

14. D.A. Fulk, A numerical analysis of Smoothed Particle Hydrodynamics, Dissertation Thesis, Air Force Institute of Technology, School of Engineering of the Air Force, Air University, Montgomery, Alabama (1994)

15. R.A. Gingold and J.J. Monaghan, Smoothed particle hydrodynamics: theory and application on spherical stars, Monthly Notices Roy. Astronom. Soc., Vol. 181, pp. 375–389 (1977).

16. R.A. Gingold and J.J. Monaghan, Kernel estimates as a basis for general particle method in hydrodynamics, J. Comput. Phys., Vol. 46, pp. 429–453 (1982).

17. G.H. Golub and C.F. Van Loan, Matrix Computations, 4th edn. Johns Hopkins, University Press, Baltimore, MD (2012).

18. L. Greengard and J. Strain, The fast Gauss transformation. SIAM J. Sci. Statist. Comput., Vol. 12(1), pp. 79–94 (1991).

19. M. Griebel, M.A. Schweitzer, Meshfree Methods for Partial Differential Equations II, Lectures Notes in Computational Science and Engineering, Vol. 43 (2005).

20. J.H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Num. Math. 2, pp. 84–90 (1960).

21. B. Li, F. Habbal and M. Ortiz, Optimal transportation meshfree approximation schemes for fluid and plastic flows, J. Numer. Meth. Engng, Vol. 83, pp. 1541–1579 (2010).

22. L.D. Libersky and A.G. Petscek, Smooth particle hydrodynamics with strength of materials, Proceedings of the next free-lagrangian method, Vol. 35, Springre pp. 248–257 (1991).

23. G.R. Liu and M.B. Liu, Smoothed Particle Hydrodynamics - A Mesh-Free Particle Method, World Scientific Publishing, Singapore (2003).

24. M.B. Liu, and G.R. Liu, Smoothed particle hydrodynamics (SPH): An overview and recent developments, Archives of Computational Methods in Engineering, Vol. 17(1), pp. 25–76 (2010).

25. M.B. Liu, G.R. Liu and K.Y. Lam, Constructing smoothing functions in smoothed particle hydrodynamics with applications, J. Comput. Appl. Math., Vol. 155, pp. 263–284 (2003).

26. M.B. Liu, W.P. Xie and G.R. Liu, Modeling incompressible flows using a finite particle method, Applied Mathematical Modelling, Vol. 29(12), pp. 1252–1270 (2005).

27. M.B. Liu, W.P. Xie and G.R. Liu, Restoring particle inconsistency in smoothed particle hydrodynamics, Appl. Numer. Math., Vol. 56(1), pp. 19–36 (2006).

28. W.K. Liu, S. Jun and Y.F. Zhang, Reproducing kernel particle methods, Int. Jour. Meth. Fluids, Vol. 20(8–9), pp. 1081–1106 (1995).

29. L.B. Lucy, A numerical approach to the testing of fusion process, Astron J, Vol. 82, pp. 1013–1024 (1977).

30. S. Ma, X. Zhang, Y. Lian and X. Zhou, Simulation of high explosive explosion using adaptive material point method, Computer Modeling in Engineering and Sciences, Vol. 39, pp. 101–123 (2009).

31. J.J. Monaghan and J.C. Lattanzio, A refined particle method for astrophysical problems, Astron Astrophys, Vol. 149, pp. 135–143 (1985).

32. J.J. Monaghan, An introduction to SPH, Comput Phys Commun, Vol. 48, pp. 89–96 (1988).

33. J.J. Monaghan, Smoothed particle hydrodynamics, Ann Rev Astron Astrophys, Vol. 30, pp. 543–574 (1992).

34. J.J. Monaghan, Simulating free surface flows with SPH. Journal of Computational Physics, Vol. 110, pp. 399–406 (1994).

35. V.I. Morariu, B.V. Srinivasan, V.C. Raykar, R. Duraiswami and L.S. Davis, Automatic online tuning for fast Gaussian summation, Advances in Neural Information Processing Systems (NIPS 2008), Vol. 21, pp. 1113–1120 (2009).

36. R.J. Renka and R. Brown, Algorithm 792 : accuracy test of ACM algorithms for interpolation of scattered data in the plane. ACM Trans. Math. Softw., Vol. 25, pp. 78–94 (1999).

37. S. Shao, Incompressible smoothed particle hydrodynamics simulation of multifluid flows, International Journal for Numerical Methods in Fluids, Vol. 69(11), pp. 1715–1735 (2012).

38. H.F. Schwaiger, An implicit corrected SPH formulation for thermal diffusion with linear free surface boundary conditions, International Journal for Numerical Methods in Engineering, Vol. 75(6), pp. 647–671 (2008).

39. I.M. Sobol, Distribution of points in a cube and approximate evaluation of integrals", U.S.S.R Comput. Maths. Math. Phys., Vol. 7, pp. 86–112 (1967).

40. C. Shu, H.Ding and K.S. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. , Vol. 192(7-8), pp. 941–954 (2003).

41. W.I. Thacker, W.I. Zhang, L.T Watson, J.B. Birch, M.A. Iyer and M.W. Berry, Algorithm 905: SHEPPACK-modified Shepard algorithm for interpolation of scattered multivariate data, ACM Trans. Math. Softw., Vol. 37(3), pp. 1–20 (2010).

42. C. Ulrich, M. Leonardi and T. Rung, Multi-physics SPH simulation of complex marine-engineering hydrodynamic problems Ocean Engineering, Vol. 64, pp. 109–121 (2013).

43. D. Violeau, Fluid Mechanics and the SPH Method: Theory and Applications, pp. 1–616 (2012).

44. H. Wendland, Scattered data approximation, Cambridge University Press, Vol. 17 (2005).