ON GENERALIZED MOVING LEAST SQUARES AND DIFFUSE DERIVATIVES

DAVOUD MIRZAEI[†], ROBERT SCHABACK[‡,*], MEHDI DEHGHAN[§]

Abstract. The Moving Least Squares (MLS) method provides an approximation $\hat{u}$ of a function u based solely on values $u(x_j)$ of u on scattered "meshless" nodes x_j. Derivatives of u are usually approximated by derivatives of $\hat{u}$. In contrast to this, we directly estimate derivatives of u from the data, without any detour via derivatives of $\hat{u}$. This is a *generalized* Moving Least Squares technique, and we prove that it produces *diffuse derivatives* as introduced by Nyroles et. al. in 1992. Consequently, these turn out to be efficient direct estimates of the true derivatives, without anything "diffuse" about them, and we prove optimal rates of convergence towards the true derivatives. Numerical examples confirm this, and we finally show how the use of shifted and scaled polynomials as basis functions in the generalized and standard MLS approximation stabilizes the algorithm.

Keywords: Moving least squares (MLS) approximation; Local polynomial reproduction; Full derivative; Diffuse derivative; Shifted scaled polynomials basis; Meshless methods.

1. Introduction

The Moving Least Squares (MLS) approximation has been introduced by [11] inspired by the pioneering work of [19] to approximate surfaces in multidimensional spaces. The MLS approximates the value $u(x)$ of an unknown function u from given data $u(x_1), \ldots, u(x_N)$ at nodes $x_1, \ldots, x_N$ near x by a value

$$\hat{u}(x) = \sum_{j=1}^{N} a_j(x) u(x_j) \approx u(x),$$

where the functions $a_j(x)$ are called *shape functions*. In the sense of [6], this is a meshless method, because it writes an approximate solution entirely in terms

Date: *August 12, 2011.*

*Corresponding author.

[†]Department of Mathematics, University of Isfahan, 81746-73441, Isfahan, Iran. d_mirzaei@aut.ac.ir.

[‡]Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Lotzestraße 16-18, D–37073 Göttingen, Germany. schaback@math.uni-goettingen.de.

[§]Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, No. 424, Hafez Ave.,15914, Tehran, Iran. mdehghan@aut.ac.ir.

of nodal values. The error analysis of the method was presented in [12], [1] and [20, 21] in different ways. In particular, we refer the reader to [12] for an account of the background connection to Backus–Gilbert optimality and related papers, and to [13] for the application to numerical integration.

There have been many meshless techniques based on the MLS approximation for the numerical solution of differential equations in recent years. When setting up large linear system for solving PDEs, MLS approximations are used to provide approximations to derivatives $D^\alpha u(x)$. This can be done via $D^\alpha \hat{u}(x) \approx D^\alpha u(x)$, i.e. taking exact derivatives of the MLS solution, or via a direct estimation of $D^\alpha u(x)$ from the data $u(x_1), \dots, u(x_N)$ near x.

This paper describes the second approach and links it to the concept of *diffuse derivatives* introduced by [18]. It turns out that the second approach calculates diffuse derivatives, and therefore these are a *direct optimal estimation* from the data. We prove optimal convergence rates for the diffuse derivatives and give numerical examples. A forthcoming paper by [16] will apply our results to make the Meshless Local Petrov–Galerkin method (MLPG) by [5, 3, 4] considerably more effective. This is the main motivation behind our approach.

This paper is organized as follows. Section 2 contains a review of the generalized moving least squares (GMLS) approximation in a form very similar to [12]. In section 3, classical and diffuse derivatives in the sense of [18] and their connections to the GMLS are described. It is proven that diffuse derivatives are GMLS approximations of true derivatives. The main contribution of the paper is in section 4 concerning error bounds for GMLS approximations of derivatives. Here, we follow the analysis path introduced by [20, 21] and the concept of norming sets introduced by [9], and adapt it to the approximation of derivatives. Finally, Section 5 provides numerical examples, while Section 6 illustrates the implementation.

2. The Generalized Moving Least Squares (GMLS) Approximation

In the classical MLS, given a set $\{u(x_j)\}$ of values of an unknown function u in a domain $\Omega \subseteq \mathbb{R}^d$ at nodes $x_j \in \Omega \subseteq \mathbb{R}^d$ for $1 \leq j \leq N$, the value $u(x)$ at a fixed point $x \in \mathbb{R}^d$ is approximately recovered by minimizing a certain weighted discrete l_2 norm. But here we start with a generalized version of Moving Least Squares.

Let $u \in C^m(\Omega)$ for some $m \geq 0$, and let $\{\lambda_j(u)\}_{j=1}^N$ be a set of continuous linear functionals λ_j from the dual $C^m(\Omega)^*$ of $C^m(\Omega)$. For a fixed given functional $\lambda \in C^m(\Omega)^*$, our problem is the approximate recovery of the value $\lambda(u)$ from the values $\{\lambda_j(u)\}_{j=1}^N$. The functionals λ and λ_j, $1 \leq j \leq N$, can, for instance, describe point evaluations of u and its derivatives up to order m. The approximation $\widehat{\lambda(u)}$ of $\lambda(u)$ should be a linear function of the data $\lambda_j(u)$, i.e. it should have the form

$$\widehat{\lambda(u)} = \sum_{j=1}^{N} a_j(\lambda)\lambda_j(u), \tag{2.1}$$

and the coefficients a_j should be linear in λ. As in the classical MLS, we assume the approximation equation (2.1) to be exact for a finite dimensional subspace $\mathcal{P} = \text{span}\{p_1, p_2, \ldots, p_Q\} \subset C^m(\Omega)$, i.e.

$$\sum_{j=1}^{N} a_j(\lambda)\lambda_j(p) = \lambda(p) \text{ for all } p \in \mathcal{P}. \tag{2.2}$$

The GMLS approximation $\widehat{\lambda(u)}$ to $\lambda(u)$ is numerically obtained as $\widehat{\lambda(u)} = \lambda(p^*)$, where $p^* \in \mathcal{P}$ is minimizing the weighted least-squares error functional

$$\sum_{j=1}^{N} \big(\lambda_j(u) - \lambda_j(p)\big)^2 w_j, \tag{2.3}$$

among all $p \in \mathcal{P}$, where we use positive weights $w_1, \ldots, w_N$ which later will be chosen in a specific way to localize the approximation. Of course, we then have to prove that (2.1) holds, but we shall get it only for the optimal solution.

Suppose the set point $X = \{x_1, x_2, \ldots, x_N\} \subset \Omega$ and $x \in \Omega$. The classical MLS is a special case of GMLS when λ and λ_j, $1 \le j \le N$ are point evaluation functionals at x and x_j, $1 \le j \le N$ and $\mathcal{P}$ is a finite–dimensional space of polynomials, while the weights are of the form

$$w_j = w(x, x_j),\ 1 \le j \le N \tag{2.4}$$

with a nonnegative weight function w that vanishes when the arguments are at a certain distance.

Furthermore, the classical MLS has an equivalent formulation, which in our generalization amounts to minimizing the quadratic form

$$\frac{1}{2}\sum_{j=1}^{N} a_j^2(\lambda)/w_j \tag{2.5}$$

as a function of the coefficients $a_j(\lambda)$ subject to the linear constraints (2.2). By some linear algebra arguments which arise already for the standard MLS and which we repeat in the next section in order to care for the dependence on the weights, the solutions p^* and $a^*(\lambda) = (a_1^*(\lambda), \ldots, a_N^*(\lambda))^T$ of the minimization problems (2.3) and (2.5), respectively, are connected by the relation

$$\widehat{\lambda(u)} = \lambda(p^*) = \sum_{j=1}^{N} a_j^*(\lambda)\lambda_j(u), \tag{2.6}$$

which also proves (2.1).

Formally, the solution p^* of the minimization problem (2.3) does not depend on λ. By calculating p^* from the data $\lambda_j(u)$ first, one can obtain estimates of $\lambda(u)$ for all λ by just evaluating $\lambda(p^*)$. This is very useful for approximating derivatives as long as the weights are independent of λ, but needs some care because both problems depend on the weights, and the weights will be connected to the functionals in most

cases. This is the main implementation recipe in the general situation. We shall be more precise in the next section.

3. Classical and Diffuse Derivatives

We take a closer look now at estimating derivative values

$$\lambda_{\alpha,x}(u) := u^\alpha(x) = \delta_x D^\alpha u \tag{3.1}$$

for fixed $x \in \Omega$ in standard multi-index notation with $|\alpha| \le m$, and where δ_x denotes the Dirac point–evaluation functional

$$\delta_x \ : \ f \mapsto f(x).$$

This situation was already mentioned as a special case in [12].

We now have to be more careful and take account of the weights. We use the weights (2.4) like in the standard MLS, even when we take more general functionals as in (3.1), but with the same x. By localization at a fixed point x, the indices $j \in \{1,\dots,N\}$ are restricted to

$$J(x) := \{j \ : \ 1 \le j \le N, \ w(x,x_j) > 0\}$$

and we introduce a basis $p_1,\dots,p_Q$ of $\mathcal{P}$ and the notation

$$\begin{aligned}
\boldsymbol{u} :=& \big(u(x_j),\ j \in J(x)\big)^T \in \mathbb{R}^{|J(x)|}\\
P :=& \big(p_\ell(x_j)\big)_{j\in J(x),\ 1\le\ell\le Q}\\
\boldsymbol{b} :=& (b_1,\dots,b_Q)^T \in \mathbb{R}^Q\\
W :=& \big(\delta_{jk} w(x,x_j)\big)_{j,k\in J(x)}\\
p :=& \sum_{k=1}^Q b_k p_k \in \mathcal{P}
\end{aligned}$$

where almost everything depends on x. Then the problem (2.3) is

$$\text{Minimize } \|\sqrt{W}(\boldsymbol{u} - P\boldsymbol{b})\|_2^2 \tag{3.2}$$

over all $\boldsymbol{b} \in \mathbb{R}^Q$, and by classical least–squares argumentation, the solution $\boldsymbol{b}^*$ satisfies the normal equations

$$A\boldsymbol{b}^* = B\boldsymbol{u}, \tag{3.3}$$

where $A = P^T W P$ and $B = P^T W$. The matrix A is of order $Q \times Q$ and plays an important role in the MLS approximation. The solution is unique if the rank of A is Q. We assume this in what follows, i.e. we assume the data point set $X = \{x_1,\dots,x_N\}$ is $\mathcal{P}$-unisolvent.

The minimization of (2.5) can be rewritten as

$$\begin{aligned}
&\text{Minimize } \frac{1}{2} a^T W^{-1} a\\
&\text{subject to } P^T a = p_\lambda
\end{aligned}$$

where $p_\lambda := (\lambda(p_1), \ldots, \lambda(p_Q))^T \in \mathbb{R}^Q$. Introducing a Lagrange multiplier $z^*(\lambda)$, we have to construct the global minimizer $a^*(\lambda)$ of

$$\frac{1}{2} a^T W^{-1} a + (z^*(\lambda))^T (P^T a - p_\lambda)$$

with respect to a. Then the solution $a^*(\lambda)$ is given by the two systems

$$\begin{aligned} a^*(\lambda) =& WPz^*(\lambda) \\ P^T a^*(\lambda) =& p_\lambda \end{aligned}$$

which implies

$$P^T W P z^*(\lambda) = p_\lambda, \; a^*(\lambda) = WP(P^T WP)^{-1} p_\lambda.$$

In some more detail,

$$a_j^*(\lambda) = w_j \sum_{k=1}^{Q} z_k^*(\lambda) \lambda_j(p_k), \tag{3.4}$$

where, due to our assumption of unisolvency, the $\boldsymbol{z}_k^*(\lambda)$ are the unique solution of

$$\sum_{k=1}^{Q} \boldsymbol{z}_k^*(\lambda) \sum_{j=1}^{N} w_j \lambda_j(p_k) \lambda_j(p_\ell) = \lambda(p_\ell), \quad 1 \le \ell \le Q. \tag{3.5}$$

If $\lambda_j = \delta_{x_j}$, $1 \le j \le N$, the two solutions are connected by

$$p_\lambda^T \boldsymbol{b}^* = u^T a^*(\lambda) = \sum_{j \in J} a_j^*(\lambda) u(x_j)$$

which is (2.1).

The solution $\boldsymbol{b}^*$ of the first problem is dependent on x via the weights and the index set, but, except that, **not on** λ. If λ is independent of x, we can get an approximation to $\lambda(u)$ by

$$\lambda(p^*) = p_\lambda^T \boldsymbol{b}^* = \sum_{k=1}^{Q} b_k^* \lambda(p_k). \tag{3.6}$$

If we keep x fixed and let the multi-index α for $\lambda_{\alpha,x} = \delta_x D^\alpha$ vary, we have no problems and can use

$$\widehat{\lambda_{\alpha,x}(u)} = p_{\lambda_{\alpha,x}}^T \boldsymbol{b}^* = \sum_{k=1}^{Q} b_k^* p_k^{(\alpha)}(x) \tag{3.7}$$

to get estimates of all derivatives of u at x after the calculation of $\boldsymbol{b}^*$, yielding (2.6). With (3.1), we have

$$\widehat{\lambda_{\alpha,x}(u)} = \widehat{D^\alpha u(x)} = \sum_{k=1}^{Q} b_k^* p_k^{(\alpha)}(x) = \sum_{j \in J(x)} a_{j,\alpha}^*(x) u(x_j), \tag{3.8}$$

where $a_{j,\alpha}^*$ are generalized MLS shape functions that correspond to the above functionals. Note that by (3.8) we have a direct estimation of $D^\alpha u$ from the data.

The implementation of the method solves the weighted least–squares problem (3.2) first, usually by a QR decomposition of $\sqrt{W}P$, avoiding the stability problems induced by solving the normal equations (3.3). Once the solution vector b^* is known, all target functionals λ that use the same input data and weights can be estimated via (3.6). Note that this requires evaluation of λ on polynomials only, not on any shape functions. This can be used to accelerate certain meshless methods for solving PDEs, as will be demonstrated in a follow–up paper ([16]) focusing on applications.

If $a^*(\lambda) = WP(P^TWP)^{-1}p_\lambda$ is requested, we decompose $\sqrt{W}P = QR$, where Q is unitary and R is upper triangular to get $P^TWP = R^TR$. By some simple calculations, $(WP)(P^TWP)^{-1}R^T = \sqrt{W}Q$. Using backward substitution, $(WP)(P^TWP)^{-1}$ is derived from this, and a^*_λ can be calculated directly. This is what we need in GMLS derivatives when $\lambda = \delta_x D^\alpha$. For standard derivatives of MLS sshape functions, some more but still straightforward calculations are needed. For instance, first derivatives of shape functions are

$$D^{e_i}a^* = \left[\sqrt{W}Q(R^T)^{-1}\right] D^{e_i}(p) + \left[\left(\widehat{W}Q - \sqrt{W^{-1}}QQ^T\widetilde{W}Q\right)(R^T)^{-1}\right] p,$$

where $\widehat{W} = D^{e_i}(W)\sqrt{W^{-1}}$ and $\widetilde{W} = \sqrt{W^{-1}}\widehat{W}$. Both brackets are calculated using backward substitution without taking inverses. Higher order derivatives can be computed similarly, but in a more complicated way. Clearly, the direct estimation of derivatives is computationally much more efficient than calculating the derivatives of the MLS shape functions.

We now connect this to the notion of *diffuse derivatives* (see [7, 10, 18, 22]) that we explain now. If we use the standard MLS with $\lambda = \delta_x$, the vector $\boldsymbol{b}^*$ comes out the same as above, and the resulting approximation is

$$\hat{u}(x) := \sum_{k=1}^{Q} b^*_k p_k(x),$$

but it should be kept in mind that $\boldsymbol{b}^*$ depends subtly on x via the weights and the index set $J(x)$. The derivatives of $\hat{u}$ at x, if calculations are done for varying x, will thus not be what we did above, since the dependence of $\boldsymbol{b}^*$ on x cannot be ignored. If it is ignored, the value

$$D^\alpha_{dif}(\hat{u})(x) := \sum_{k=1}^{Q} b^*_k p^{(\alpha)}_k(x)$$

is called the *diffuse derivative* of $\hat{u}$ at x.

Theorem 3.1. *The GMLS approximation when applied for $\lambda = \lambda_{\alpha,x} := \delta_x D^\alpha$ and $\lambda_j = \delta_{x_j}$, $1 \le j \le N$, calculates the diffuse derivative of the standard MLS. The latter is a good approximation to $D^\alpha u(x)$, but is not the same as the standard derivative $D^\alpha \hat{u}(x)$.* □

For applications in meshless methods, the estimation of $D^\alpha u(x)$ via (3.1) is all that is needed when setting up linear system of equations, since it is the best weighted moving least squares estimate based on the data $u(x_1,), \ldots, u(x_N)$. It is a completely unnecessary detour to go via $\hat{u}(x)$ and take derivatives thereof. We shall support this theoretically and practically in what follows. As we shall see, the accuracies of both schemes are nearly the same. Comparing the diffuse and full (standard) derivatives of $\hat{u}$, the computational cost of the diffuse derivatives is considerably less. For the GMLS, we just have to calculate $\boldsymbol{b}^*$, which takes the same amount of work like in the standard MLS, and then we just need the derivatives of the polynomial basis to get (3.7). Since polynomials of degree up to m are exactly reproduced for all choices of weights, the full and diffuse derivatives of these polynomials coincide ([10]).

The use of $\hat{u}$ and its derivatives is not necessary when setting up the linear system. After solving, we will have approximations for the values $u(x_j)$ at the nodes. Then, for *postprocessing*, it may be necessary to calculate exact derivatives of the approximate solution $\hat{u}$, e.g. for calculation of stress in elasticity problems. At this time, it is up to the user whether exact or diffuse derivatives are calculated. If users want to have a single solution function $\hat{u}$ with exact derivatives, they will have to pay a price. If they can admit small errors, they can get away with diffuse derivatives. For postprocessing, the use of diffuse derivatives makes a lot of sense in certain situations, but not for setting up linear systems.

Since the word "diffuse" may mislead readers to assume that these derivatives are not first–choice, we ignore this term from now on and use $\widehat{D^\alpha u(x)}$ or $\widehat{\lambda(u)}$ instead, to let the notation indicate that we have a direct and usually very good numerical approximation to $D^\alpha u(x)$ or $\lambda(u)$, respectively. For future work, we suggest to drop the term *diffuse derivative* in favor of *GMLS derivative approximation*. There is nothing *diffuse* or *uncertain* about it.

4. Error Bounds

In the MLS, and for other scattered data approximation methods, the quantities *fill distance* and *separation distance* are important to measure the quality of data points and to derive rates of convergence. For a set of points $X = \{x_1, x_2, \ldots, x_N\}$ in a bounded domain $\Omega \subseteq \mathbb{R}^d$, the *fill distance* is defined to be

$$h_{X,\Omega} = \sup_{x\in\Omega} \min_{1\le j\le N} \|x - x_j\|_2,$$

and the *separation distance* is defined by

$$q_X = \frac{1}{2} \min_{i\neq j} \|x_i - x_j\|_2.$$

A set X of data sites is said to be *quasi-uniform* with respect to a constant $c_{\rm qu} > 0$ if

$$q_X \leq h_{X,\Omega} \leq c_{\rm qu} q_X. \tag{4.1}$$

Also,

$$B(x,r) := \{y \in \mathbb{R}^d \ : \ \|x-y\|_2 \leq r\}$$

stands for the ball of radius r around x.

To be more precise with the generation of weights, we choose a continuous function $\phi : [0,\infty) \to \mathbb{R}$ that is positive on $[0,1/2]$ and supported in $[0,1]$, and define

$$w_\delta(x,y) = \phi\left(\frac{\|x-y\|_2}{\delta}\right),$$

for $\delta > 0$ as a weight function. Then we define $J(x) = \{j \in \{1,2,...,N\} : \|x - x_j\|_2 \leq \delta\}$.

Henceforth, we use $\mathcal{P} = \mathbb{P}^d_m$ as a space of d-variate polynomials of degree at most m and dimension $Q = \binom{m+d}{d}$.

At first, the convergence rate of a *local polynomial reproduction* system will be presented and then we will show that the generalized MLS of the first section is a local polynomial reproduction in the following sense.

Definition 4.1. Consider a process that defines for every $\mathbb{P}^d_m$–unisolvent set $X = \{x_1, x_2, \ldots, x_N\} \subset \Omega$ and each multi-index α with $|\alpha| \leq m$ a family of functions $s_{j,\alpha} \ : \ \Omega \to \mathbb{R}$, $1 \leq j \leq N$ to approximate

$$D^\alpha u(x) \approx \sum_{j=1}^N s_{j,\alpha}(x) u(x_j)$$

for functions $u \in C^m(\Omega)$. Then we say that the process *provides a local polynomial reproduction of degree m on Ω* if there exist constants $h_0, C_{1,\alpha}, C_{2,\alpha} > 0$ such that

(1) $\sum_{j=1}^N s_{j,\alpha}(x) p(x_j) = D^\alpha p(x)$, for all $p \in \mathbb{P}^d_m$, $x \in \Omega$,
(2) $\sum_{j=1}^N |s_{j,\alpha}(x)| \leq C_{1,\alpha} h_{X,\Omega}^{-|\alpha|}, \quad \forall x \in \Omega$,
(3) $s_{j,\alpha}(x) = 0 \ if \ \|x - x_j\|_2 > C_{2,\alpha} h_{X,\Omega}$,

is satisfied for all $|\alpha| \leq m$ and all X with $h_{X,\Omega} \leq h_0$.

This definition is a generalized form of Definition 3.1 presented in [21]. We avoided the notation $s_j^{(\alpha)}(x)$ since it is not true that $s_{j,\alpha} = D^\alpha s_{j,0}$, as suggested by item 1 above.

For every sample point $x \in \Omega$, and for the classical MLS with $\lambda = \delta_x$, [21] has proved

Theorem 4.2. *If $\{s_{j,0}(x)\}$ is a local polynomial reproduction of order m, there exists a constant C such that for all $u(x) \in C^{m+1}(\Omega^*)$ and all X with $h_{X,\Omega} \leq h_0$ the classical MLS solution $\hat{u}$ satisfies the error bound*

$$|u(x) - \widehat{u}(x)| \leq C h_{X,\Omega}^{m+1} |u|_{C^{m+1}(\Omega^*)} \tag{4.2}$$

where Ω^* is the closure of $\bigcup_{x\in\Omega} B(x, C_2 h_0)$. The semi-norm in the right hand side is defined by

$$|u|_{C^{m+1}(\Omega^*)} := \max_{|\beta|=m+1} \|D^\beta u\|_{L_\infty(\Omega^*)}.$$

Using the same techniques, we can prove

Theorem 4.3. *Suppose that $\Omega \subset \mathbb{R}^d$ is bounded. Define Ω^* to be the closure of $\bigcup_{x\in\Omega} B(x, C_2 h_0)$. Define*

$$\widehat{D^\alpha u(x)} := \sum_{j=1}^N s_{j,\alpha}(x) u(x_j),$$

where $\{s_{j,\alpha}\}$ is a local polynomial reproduction of order m on Ω for $|\alpha| \le m$. Then there exists a constant $C > 0$ such that for all $u \in C^{m+1}(\Omega^*)$ and all X with $h_{X,\Omega} \le h_0$ there is an error bound

$$|D^\alpha u(x) - \widehat{D^\alpha u(x)}| \le C h_{X,\Omega}^{m+1-|\alpha|} |u|_{C^{m+1}(\Omega^*)}. \tag{4.3}$$

Proof. Let $p \in \mathbb{P}_m^d$ be an arbitrary polynomial. Using the properties of local polynomial reproduction in Definition 4.1 yields

$$\begin{aligned}
\left|D^\alpha u(x) - \widehat{D^\alpha u(x)}\right| &\le |D^\alpha u(x) - D^\alpha p(x)| + \left|D^\alpha p(x) - \sum_{j=1}^N s_{j,\alpha}(x) u(x_j)\right| \\
&\le |D^\alpha u(x) - D^\alpha p(x)| + \sum_{j=1}^N |s_{j,\alpha}(x)|\, |p(x_j) - u(x_j)| \\
&\le \|D^\alpha u - D^\alpha p\|_{L_\infty(\mathcal{D})} + \|u - p\|_{L_\infty(\mathcal{D})} \sum_{j=1}^N |s_{j,\alpha}(x)| \\
&\le \|D^\alpha u - D^\alpha p\|_{L_\infty(\mathcal{D})} + C_{1,\alpha} h_{X,\Omega}^{-|\alpha|} \|u - p\|_{L_\infty(\mathcal{D})}
\end{aligned} \tag{4.4}$$

where $\mathcal{D} = B(x, C_{2,\alpha} h_{X,\Omega})$. Now choose p to be the Taylor polynomial of u around x. This gives for each $|\beta| = m+1$ and $y \in \Omega$ a $\xi(y,\beta) \in \Omega^*$ such that

$$\begin{aligned}
u(y) &= \sum_{|\beta|\le m} \frac{D^\beta u(x)}{\beta!}(y-x)^\beta + \sum_{|\beta|=m+1} \frac{D^\beta u(\xi(y,\beta))}{\beta!}(y-x)^\beta \\
&= p(y) + \sum_{|\beta|=m+1} \frac{D^\beta u(\xi(y,\beta))}{\beta!}(y-x)^\beta.
\end{aligned} \tag{4.5}$$

where

$$p(y) = \sum_{|\beta|\le m} \frac{D^\beta u(x)}{\beta!}(y-x)^\beta. \tag{4.6}$$

Hence

$$\begin{aligned}
\|u - p\|_{L_\infty(\mathcal{D})} &\le (C_{2,\alpha} h_{X,\Omega})^{m+1} \sum_{|\beta|=m+1} \frac{1}{\beta!} \|D^\beta u\|_{L_\infty(\Omega^*)} \\
&\le C h_{X,\Omega}^{m+1} |u|_{C^{m+1}(\Omega^*)}.
\end{aligned} \tag{4.7}$$

Moreover, since $D^\alpha u \in C^{m+1-|\alpha|}(\Omega^*)$ the Taylor expansion of order $m-|\alpha|$ for $D^\alpha u$ around $x \in \Omega$ exists. This gives for each $|\beta| = m+1-|\alpha|$ and every $y \in \Omega$ a $\zeta(y,\beta) \in \Omega^*$ such that

$$D^\alpha u(y) = \sum_{|\beta|\le m-|\alpha|} \frac{D^{\beta+\alpha}u(x)}{\beta!}(y-x)^\beta + \sum_{|\beta|=m+1-|\alpha|} \frac{D^{\beta+\alpha}u(\zeta(y,\beta))}{\beta!}(y-x)^\beta. \quad (4.8)$$

The first part of the right hand side of equation (4.8) is clearly $D^\alpha p(y)$ with $p(y)$ defined in equation (4.6). Therefore

$$\left\|D^\alpha u - D^\alpha p\right\|_{L_\infty(\mathcal{D})} \le C h_{X,\Omega}^{m+1-|\alpha|} |u|_{C^{m+1}(\Omega^*)}. \quad (4.9)$$

Combining (4.7) and (4.9) with (4.4) leads to (4.3). □

Now it suffices to show that the family of functions $\{a^*_{j,\alpha}\}$ in (3.8) forms a local polynomial reproduction in sense of Definition 4.1. It can be done by the concept of *norming sets*, introduced by *et.al* [9]. Before that we need some definitions.

Definition 4.4. A set $\Omega \subset \mathbb{R}^d$ is said to satisfy an *interior cone condition* if there exists an angle $\theta \in (0, \pi/2)$ and a radius $r > 0$ such that for every $x \in \Omega$ a unit vector $\xi(x)$ exists such that the cone

$$C(x,\xi,\theta,r) := \left\{x + ty : y \in \mathbb{R}^d, \|y\|_2 = 1, y^T\xi \ge \cos\theta, t \in [0,r]\right\}$$

is contained in Ω.

Let V be a finite-dimensional vector space with norm $\|.\|_V$ and let $\Lambda \subseteq V^*$ be a finite set consisting of N functionals. Here, V^* denotes the dual space of V consisting of all linear and continuous functionals defined on V.

Definition 4.5. Λ is a *norming set* for V if the mapping $T : V \to T(V) \subseteq \mathbb{R}^N$ defined by $T(v) = (\lambda(v))_{\lambda\in\Lambda}$ is injective. The operator T is called the *sampling operator*.

Theorem 4.6. *([9, [15] and [21]) Suppose V is a finite-dimensional normed linear space and $\Lambda = \{\lambda_1, ..., \lambda_N\}$ is a norming set for V, T being the corresponding sampling operator. For every $\lambda \in V^*$ there exists a vector $s \in \mathbb{R}^N$ depending only on λ such that, for every $v \in V$,*

$$\lambda(v) = \sum_{j=1}^N s_j \lambda_j(v),$$

and

$$\|s\|_{\mathbb{R}^{N*}} \le \|\lambda\|_{V^*} \|T^{-1}\|.$$

Theorem 3.8 of [21] proves:

Theorem 4.7. *If $\Omega \subset \mathbb{R}^d$ is compact and satisfies an interior cone condition with radius r and angle $\theta \in (0, \pi/2)$, for fixed number m if the set X satisfies*

$$h_{X,\Omega} \leq \frac{r \sin\theta}{4(1+\sin\theta)m^2}, \tag{4.10}$$

then $\Lambda = \{\delta_{x_1}, \ldots, \delta_{x_N}\}$ is a norming set for $\mathbb{P}^d_m(\Omega)$ and $\|T^{-1}\| \leq 2$.

Also it is easy to show that

Proposition 4.8. *$\Lambda = \{\delta_{x_1}, \ldots, \delta_{x_N}\}$ forms a norming set for $\mathbb{P}^d_m(\Omega)$ if and only if $X \subset \Omega$ is $\mathbb{P}^d_m$-unisolvent.*

One the other side, from Proposition 2.2 of [17] and 11.6 of [21], we have:

Proposition 4.9. *Suppose that $\Omega \subset \mathbb{R}^d$ is bounded and satisfies an interior cone condition with radius r and angle θ. If $p \in \mathbb{P}^d_m$ and $|\alpha| \leq m$ then*

$$\|D^\alpha p\|_{L_\infty(\Omega)} \leq \left(\frac{2m^2}{r\sin\theta}\right)^{|\alpha|} \|p\|_{L_\infty(\Omega)}. \tag{4.11}$$

If $V = \mathbb{P}^d_m(\Omega)$ and $\lambda = \delta_x D^\alpha$, in the situation of Theorem 4.7, using (4.11) and (4.10) it is easy to show

$$\|\lambda\|_{V^*} \leq \left(\frac{2m^2}{r\sin\theta}\right)^{|\alpha|} \leq \big(2(1+\sin\theta)\big)^{-|\alpha|} h_{X,\Omega}^{-|\alpha|}.$$

Consequently we can state:

Corollary 4.10. *Let $\Omega \subset \mathbb{R}^d$ be bounded and satisfy an interior cone condition with radius r and angle θ. If $X = \{x_1, \ldots, x_N\} \subset \Omega$ and (4.10) is satisfied, then there exist for every $x \in \Omega$ real numbers $s_{j,\alpha}(x)$ such that*

$$\sum_{j=1}^N |s_{j,\alpha}(x)| \leq 2\big(2(1+\sin\theta)\big)^{-|\alpha|} h_{X,\Omega}^{-|\alpha|},$$

and

$$\sum_{j=1}^N s_{j,\alpha}(x) p(x_j) = D^\alpha p(x)$$

for all $p \in \mathbb{P}^d_m$.

Now we should convert this global existence result to the local situation. It can be done using the fact that for every point $x \in \Omega$ we can find a cone $C(x)$ that is completely contained in Ω and since every cone itself satisfies a cone condition, we can apply Corollary 4.10 to the cone $C(x)$ and $Y = X \cap C(x)$. Therefore, as in Theorem 3.14 of [21], we can prove:

Theorem 4.11. *If $\Omega \subset \mathbb{R}^d$ is compact and satisfies an interior cone condition with radius r and angle $\theta \in (0, \pi/2)$, for fixed $m \in \mathbb{N}$ there exist constants*

$$C_{1,\alpha} = 2\big(2(1+\sin\theta)\big)^{-|\alpha|}, \quad C_{2,\alpha} = \frac{16(1+\sin\theta)^2 m^2}{3\sin^2\theta}, \quad h_0 = \frac{1}{C_2}$$

such that for every $X \subset \Omega$ with $h_{X,\Omega} \leq h_0$ and every $x \in \Omega$ we can find real numbers $s_{j,\alpha}(x)$, $1 \leq j \leq N$ such that they form a local polynomial reproduction as in Definition 4.1.

Now using the minimal property of $a^*_{j,\alpha}$ in (3.8), we can show these functions form a local polynomial reproduction. This comes in the following theorem. The proof is same as the proof of Theorem 4.7 of [21].

Theorem 4.12. *Suppose that $\Omega \subset \mathbb{R}^d$ is compact and satisfies an interior cone condition with radius r and angle $\theta \in (0, \pi/2)$. Fix $m \in \mathbb{N}$. Let h_0, $C_{1,\alpha}$, and $C_{2,\alpha}$ denote the above-mentioned constants. Suppose that X satisfies (4.1) and $h_{X,\Omega} \leq h_0$. Let $\delta = 2C_{2,\alpha}h_{X,\Omega}$. Then the basis functions $a^*_{j,\alpha}$ from (3.8) provide a local polynomial reproduction as in Definition 4.1 with constant $\widetilde{C}_{1,\alpha}$ and $\widetilde{C}_{2,\alpha}$ that can be derived explicitly.*

Finally using Theorems 4.3 and 4.12 we conclude the following corollary that includes the order of convergence of the MLS approximation and its diffuse derivatives.

Corollary 4.13. *In the situation of Theorem 4.12, define Ω^* to be the closure of $\bigcup_{x\in\Omega} B(x, C_{2,\alpha}h_0)$. Define*

$$\widehat{D^\alpha u(x)} := \sum_{j=1}^{N} a^*_{j,\alpha}(x) u(x_j)$$

where $a^*_{j,\alpha}(x)$ are functions derived from the MLS approximation in (3.8). Then there exists a constant $c > 0$ such that for all $u \in C^{m+1}(\Omega^*)$ and all $X \subset \Omega$ with $h_{X,\Omega} \leq h_0$ which are quasi–uniform in the sense of (4.1) with the same constant c_{qu} we have

$$\|D^\alpha u(x) - \widehat{D^\alpha u(x)}\|_{L_\infty(\Omega)} \leq c h^{m+1-|\alpha|}_{X,\Omega} |u|_{C^{m+1}(\Omega^*)}. \tag{4.12}$$

The error estimates of MLS approximation and its full derivatives are given in [2] and [23] using different strategies. They have proved that the error of full derivatives of order $|\alpha|$ is of order $O(h^{m+1-|\alpha|})$ where h plays the same role as $h_{X,\Omega}$. Thus, direct estimation of derivatives from function values is recommendable instead of taking full or diffuse derivatives of the classical MLS solution $\hat{u}$.

Following a suggestion of a referee, we finally point out that [14] uses triangulation–based weights to eliminate the requirement of quasi–uniform node placement. It seems to be an open problem to extend this to truly meshless methods avoiding triangulations.

5. Numerical Examples

To confirm the above theoretical bounds, we look at MLS approximation for *Franke's function*

$$\begin{aligned} u(\bar{x},\bar{y}) =& \frac{3}{4}e^{-1/4((9\bar{x}-2)^2+(9\bar{y}-2)^2)} + \frac{3}{4}e^{-(1/49)(9\bar{x}+1)^2-(1/10)(9\bar{y}+1)^2)} \\ &+ \frac{1}{2}e^{-1/4((9\bar{x}-7)^2+(9\bar{y}-3)^2)} - \frac{1}{5}e^{-(9\bar{x}-4)^2-(9\bar{y}-7)^2} \end{aligned}$$

on $[0,1]^2$ which is a standard test function for 2D scattered data fitting since the seminal survey of [8]. Note that $(\bar{x},\bar{y})$ denotes the two components of $x \in \mathbb{R}^2$. First, regular node distributions with distance h along each axis are used. A compactly supported RBF which possesses continuous derivatives up to order 4 is used as weight function, and the shifted scaled polynomials (see Section 6) are employed as basis functions.

INSERT TABLE 1

Table 1 presents the ratios of errors for the function and its first and second standard and GMLS derivatives in a fixed and sufficiently fine test point mesh of size 31×31 on $[0,1]^2$. "Ratio0", "Ratio1" and "Ratio2" refer to the ratios of error of the function, its first derivative with respect to $\bar{x}$ and its second derivative with respect to $\bar{x}$, respectively. The distance h is divided by 2 row by row, so the ratio is computed by

$$\log_2\left(\frac{\|e(h)\|_\infty}{\|e(h/2)\|_\infty}\right). \tag{5.1}$$

We consider both standard and GMLS derivatives. The results are presented for $m = 1,2,3$ and $\delta = 1.5mh$. According to theoretical bounds, the ratios should be approximately $m + 1 - |\alpha|$. As we can see, the numerical results confirm the analytical bounds. Also it is evident that there is no significant difference between the rates of convergence of standard and GMLS derivatives. Note that with $m = 1$ we can not recover the second derivatives. In this example, for instance, the CPU time needed to compute the second GMLS derivative with $h = 0.1/16$ ($N = 25921$) and $m = 2$ in the above test point mesh is 2.60 sec, while it is 3.35 sec. for the standard derivative.

Now, we choose the set of centers to be *Halton points* in $[0,1]^2$. We use the following commands in Matlab to generate such sets:

```
p = haltonset(2,'Skip',1e3,'Leap',1e2);
N = 1000;          % number of selected centers
X = net(p,N);
```

The first 1000 Halton points are depicted in Fig. 1.

INSERT FIGURE 1

Following [20], it is in general too expensive to compute $h_{X,\Omega}$ exactly. Therefore we used the approximation $h_{X,\Omega} \approx h = 1/\sqrt{N}$ together with $\delta = 1.5mh$. The maximum errors and ratios, which are provided in a regular mesh 31×31 in $[0,1]^2$, are presented in Tables 2 and 3 for the first and the second derivatives with respect to first variable, respectively, for $m = 3$. The approximate fill distance h is divided by 2 consecutively and the ratios are computed by (5.1). One can see that the theoretical bounds are obtained and the results are nearly the same for both standard and GMLS derivatives. The CPU time required to execute the GMLS subroutine for computing the second derivative with $N = 16000$ in the above-mentioned test point mesh is 2.03 sec, while it is 3.40 sec for the standard MLS derivative subroutine.

INSERT TABLE 2

6. Some Notes on Numerical Implementation

Sometimes, the set

$$\mathcal{B} = \{x^\alpha\}_{0 \le |\alpha| \le m} \tag{6.1}$$

is used as a basis for $\mathbb{P}_m^d$ in the MLS approximation. The choice of this basis is important and has quite some influence on the matrix $A = P^T W P$ and thus on the matrix R obtained from the QR decomposition of $\sqrt{W}P$. This has major effects on stability, especially when the normal equations (3.3) are used directly. As an example, consider the unit square $[0,1]^2$ in $\mathbb{R}^2$ with regular node distribution of distance h and fix $m = 2$. In Fig. 2, the determinants and condition numbers of A are depicted in terms of decreasing h at a sample point $(\pi/4, \pi/4) \in [0,1]^2$ on the left and right side, respectively. As we see, the results get worse as h decreases.

INSERT FIGURE 2

To overcome this drawback it is better to use the shifted scaled basis polynomials. The shifted basis, which for example was used by [12] and [20], can be defined for fixed z by

$$\mathcal{B}^z = \{(x-z)^\alpha\}_{0 \le |\alpha| \le m} \tag{6.2}$$

and the shifted scaled basis by

$$\mathcal{B}_h^z = \left\{ \frac{(x-z)^\alpha}{h^{|\alpha|}} \right\}_{0 \le |\alpha| \le m}, \tag{6.3}$$

where h can be q_X, $h_{X,\Omega}$ or an average of them for a quasi–uniform set X. In all cases, z is an evaluation point such as a test point or a Gaussian point for integration

in weak-form based techniques. FIG. 3 shows the same results as before for the shifted scaled basis functions.

INSERT FIGURE 3

The effect of this variation is shown in FIG. 4, where we have illustrated the maximum error of reconstruction of Franke's function and its first and second derivatives with respect to $\bar{x}$ on $[0,1]^2$ with and without shifted scaled basis functions. Numerical instabilities are evident on the left side, where the basis (6.1) is applied.

INSERT FIGURE 4

To analyze this phenomenon, we use the notations $A = P^T W P$, where the basis $\mathcal{B}$ is employed and $A^z = P(\cdot - z)^T W P(\cdot - z)$ and $A_h^z = P(\frac{\cdot - z}{h})^T W P(\frac{\cdot - z}{h})$ where $\mathcal{B}^z$ and $\mathcal{B}_h^z$ are used, respectively. By using

$$(x-z)^\alpha = \sum_{\beta \le \alpha} \underbrace{\binom{\alpha}{\beta}(-1)^{|\alpha-\beta|} z^{\alpha-\beta}}_{C_z(\alpha,\beta)} x^\beta = \sum_{\beta \le \alpha} C_z(\alpha,\beta) x^\beta$$

we have $P(\cdot - z) = PC_z$, where C_z is a Q by Q triangular matrix with diagonal elements 1. It is clear that $A^z = C_z^T A C_z$ and $\det(A) = \det(A^z)$. On the other side, we set

$$H_h = \operatorname{diag}\Big\{1, \underbrace{\frac{1}{h}, \cdots, \frac{1}{h}}_{\binom{d}{d-1} \text{ times}}, \underbrace{\frac{1}{h^2}, \cdots, \frac{1}{h^2}}_{\binom{d+1}{d-1} \text{ times}}, \cdots, \underbrace{\frac{1}{h^m}, \cdots, \frac{1}{h^m}}_{\binom{d-1+m}{d-1} \text{ times}}\Big\}_{Q \times Q}.$$

It is obvious that $P(\frac{\cdot - z}{h}) = PC_z H_h$ and $A_h^z = H_h A^z H_h$, hence $\det(A_h^z) = \det(A^z)\big[\det\big(H_h\big)\big]^2 = \det(A)\big[\det\big(H_h\big)\big]^2$. Using the combinatorial formula

$$\sum_{j=0}^{m} j C_{d-1}^{d-1+j} = d C_{d+1}^{m+d} =: \rho,$$

we have $\det\big(H_h\big) = h^{-\rho}$, therefore

$$\det(A_h^z) = h^{-2\rho} \det(A).$$

This is the reason why the determinant of A_h^z remains constant as h decreases. Consequently, we have

$$\det(R_h^z) = h^{-\rho} \det(R),$$

where R_h^z is upper triangular matrix obtained by QR decomposition of $\sqrt{W} P(\frac{\cdot - z}{h})$.

We can also estimate the condition numbers of both matrices A_h^z and R_h^z. Since $P(\frac{\cdot - z}{h}) = PC_z H_h$ and due to the uniqueness property of QR decomposition of full-rank matrices, we have

$$R_h^z = RC_z H_h.$$

Finally, $\mathrm{cond}(C_z) = 1$ and $\mathrm{cond}(H_h) = h^m$ yield

$$\mathrm{cond}(R) \approx \sqrt{\mathrm{cond}(A)}, \quad \mathrm{cond}(R_h^z) \approx \mathrm{cond}(R)h^m, \quad \mathrm{cond}(A_h^z) \approx \mathrm{cond}(A)h^{2m}. \tag{6.4}$$

Although the QR decomposition gives stable results in many cases, (6.4) implies that the shifted scaled basis is recommendable even when the QR method is applied. In numerical results presented in the previous section, we followed this strategy.

The quantity h can be replaced by a function which varies in accordance with the node density in Ω, see [**?**].

7. Conclusion and Outlook

This paper implies that "diffuse" derivatives used within certain applications of the moving–least–squares method (MLS) can be stably implemented and induce no loss in accuracy, because they are identical to direct optimal estimates of derivatives provided by the generalized moving least squares (GMLS) of this paper. In particular, the orders of convergence of both derivatives to the exact values turn out to be the same, and the computational efficiency of GMLS derivatives is better. On the side, we investigated the stabilization effect of shifted scaled polynomial bases. In a forthcoming paper ([16]), the GMLS will be applied to enhance the computational efficiency of the meshless local Petrov-Galerkin (MLPG) method of Atluri and his colleagues ([5, 4, 3]) significantly.

Acknowledgements

The authors deeply thank the anonymous referees for their valuable remarks and suggestions which improved the quality of paper.

References

[1] M.G. Armentano. Error estimates in sobolev spaces for moving least square approximations. *SIAM J. Numer. Anal.*, 39(1):38–51, 2001.

[2] M.G. Armentano and R.G. Durán. Error estimates for moving least square approximations. *Appl. Numer. Math.*, 37:397–416, 2001.

[3] S. N. Atluri. *The meshless method (MLPG) for domain and BIE discretizations*. Tech Science Press, Encino, CA, 2005.

[4] S. N. Atluri and S. Shen. *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press, Encino, CA, 2002.

[5] S.N. Atluri and T.-L. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in Computational mechanics. *Comput. Mech.*, 22:117–127, 1998.

[6] T. Belytschko, Y. Krongauz, D.J. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Comput. Method Appl. Mech. Eng., special issue*, 139:3–47, 1996.

[7] P. Breitkopf, A. Rassineux, G. Touzot, and P. Villon. Explicit form and efficient computation of MLS shape functions and their derivatives. *Int. J. Numer. Methods Eng.*, 48:451–466, 2000.

[8] R. Franke. Scattered data interpolation: tests of some methods. *Mathematics of Computation*, 48:181–200, 1982.

[9] K. Jetter, J. Stöckler, and J.D. Ward. Error estimates for scattered data interpolation on spheres. *Math. Comput.*, 68:733–747, 1999.

[10] D. W. Kim and Y. Kim. Point collocation methods using the fast moving least-square reproducing kernel approximation. *Int. J. Numer. Methods Eng.*, 56:1445–1464, 2003.

[11] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Math. Comput.*, 37:141–158, 1981.

[12] D. Levin. The approximation power of moving least-squares. *Math. Comput.*, 67:1517–1531, 1998.

[13] D. Levin. Stable integration rules with scattered integration points. *Journal of Computational and Applied Mathematics*, 112:181–187, 1999.

[14] Y. Lipman. Stable moving least-squares. *Journal of Approximation Theory*, 161:371–384, 2009.

[15] H.N. Mhaskar, F.G. Narcowich, and J.D. Ward. Spherical Marcinkiewicz-Zygmund inequalities and positive quadrature. *Math. Comput.*, 70:1113–1130, 2001.

[16] D. Mirzaei and R. Schaback. Direct meshless local Petrov-Galerkin (DMLPG) method: A generalized MLS approximation. Preprint, 2011.

[17] F.J. Narcowich, J.D. Ward, and H. Wendland. Sobolev bounds on functions with scattered zeros, with application to radial basis function surface fitting. *Math. Comput.*, 47(250):743–763, 2004.

[18] B. Nyroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Comput. Mech.*, 10:307–318, 1992.

[19] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 23th National Conference ACM*, pages 517–523, 1968.

[20] H. Wendland. Local polynomial reproduction and moving least squares approximation. *IMA J. Numer. Anal.*, 21:285–300, 2001.

[21] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005.

[22] Y.C. Yoon, S.H. Lee, and T. Belytschko. Enriched meshfree collocation method with diffuse derivatives for elastic fracture. *Comput. Math. Appl.*, 51:1349–1366, 2006.

[23] C. Zuppa. Error estimates for moving least square approximations. *Bull. Braz. Math. Soc.*, 34(2):231–249, 2003.

TABLE 1
The ratios of errors of Franke's function and its first and second standard and GMLS derivatives

			Ratio1		Ratio2	
	h	Ratio0	standard	GMLS	standard	GMLS
$m = 1$						
	0.1	—	—	—	—	—
	$\frac{0.1}{2}$	1.82	1.39	1.38	—	—
	$\frac{0.1}{4}$	1.94	0.99	0.98	—	—
	$\frac{0.1}{8}$	1.99	0.98	0.98	—	—
	$\frac{0.1}{16}$	2.00	0.95	0.95	—	—
$m = 2$						
	0.1	—	—	—	—	—
	$\frac{0.1}{2}$	3.06	1.37	1.37	1.33	1.34
	$\frac{0.1}{4}$	3.72	1.80	1.79	1.50	1.48
	$\frac{0.1}{8}$	3.93	1.95	1.94	1.22	1.21
	$\frac{0.1}{16}$	3.86	1.98	1.98	1.13	1.12
$m = 3$						
	0.1	—	—	—	—	—
	$\frac{0.1}{2}$	2.23	1.99	2.05	0.88	0.84
	$\frac{0.1}{4}$	3.36	3.30	3.30	1.59	1.55
	$\frac{0.1}{8}$	3.82	3.80	3.81	1.88	1.87
	$\frac{0.1}{16}$	3.95	3.95	3.95	1.97	1.97

TABLE 2
Maximum and ratios of errors of the first standard and GMLS derivatives of Franke's function at Halton points with $m = 3$

		Standard		GMLS	
N	h	$\|e\|_\infty$	ratio	$\|e\|_\infty$	ratio
1000	0.03162	7.89×10^{-2}	—	7.76×10^{-2}	—
4000	0.01581	1.08×10^{-2}	2.87	1.08×10^{-2}	2.84
16000	0.00791	2.00×10^{-3}	2.43	1.99×10^{-3}	2.45
64000	0.00395	1.08×10^{-4}	4.22	1.08×10^{-4}	4.20

TABLE 3

Maximum and ratios of errors of the second standard and GMLS derivatives of Franke's function at Halton points with $m = 3$

		Standard		GMLS	
N	h	$\|e\|_\infty$	ratio	$\|e\|_\infty$	ratio
1000	0.03162	8.00×10^{0}	–	7.77×10^{0}	–
4000	0.01581	2.16×10^{0}	1.87	2.17×10^{0}	1.84
16000	0.00791	5.76×10^{-1}	1.93	5.74×10^{-1}	1.92
64000	0.00395	1.40×10^{-1}	2.04	1.41×10^{-1}	2.03

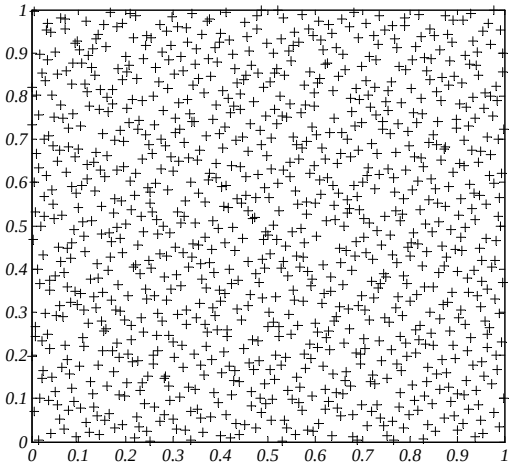

FIGURE 1. First 1000 Halton points

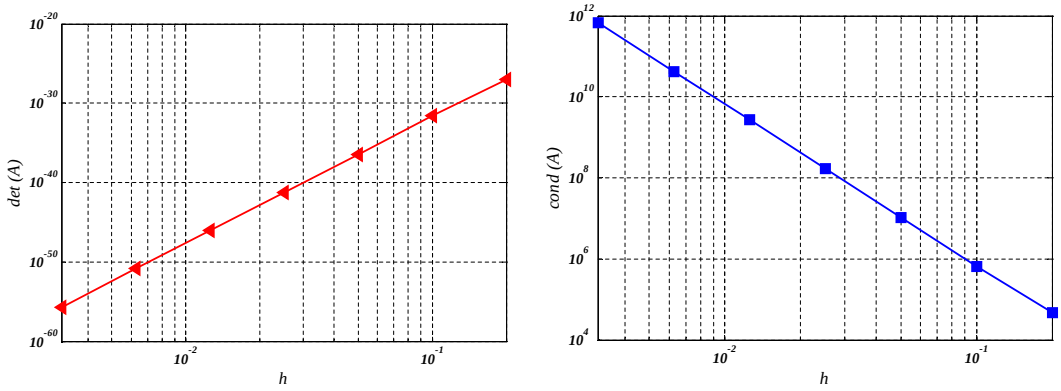

FIGURE 2. Determinants (left) and condition numbers (right) of A at sample point $(\pi/4, \pi/4)$ using basis (6.1)

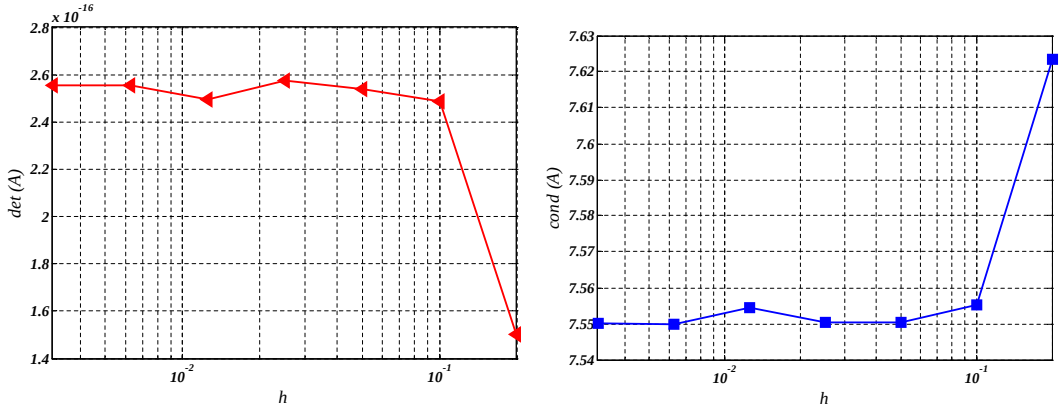

FIGURE 3. Determinants (left) and condition numbers (right) of A at sample point $(\pi/4, \pi/4)$ using basis (6.3)

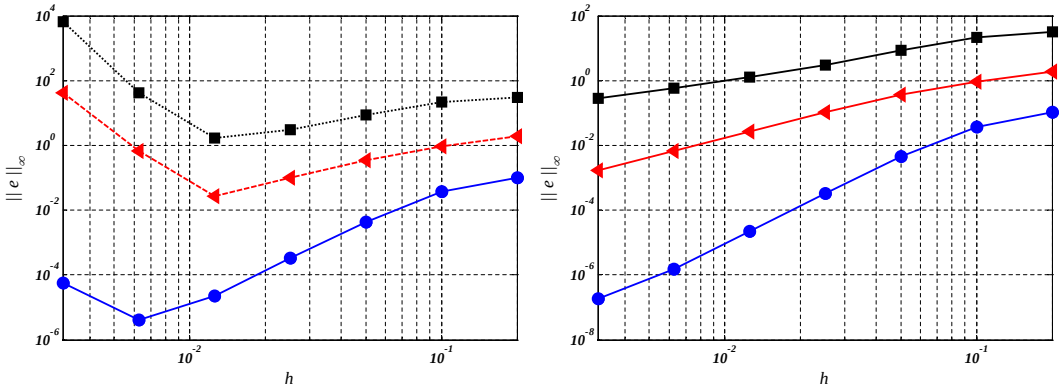

FIGURE 4. Approximation errors of Franke's function (solid lines and circles), its first derivative (dash lines and triangles) and its second derivative (dot lines and squares) using basis (6.1) (left) and basis (6.3) (right).