Approximate Moving Least-Squares Approximation with Compactly Supported Radial Weights

Gregory E. Fasshauer

Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL 60616, USA *

Abstract. We use Maz'ya and Schmidt's theory of approximate approximation to devise a fast and accurate approximate moving least-squares approximation method which does not require the solution of any linear systems. Since we use compactly supported weight functions, the remaining summation is also efficient. We compare our new algorithm with three other approximation methods based on compactly supported radial functions: multilevel interpolation, the standard moving least-squares approximation method, and a multilevel moving least-squares algorithm. A multilevel approximate moving least-squares approximation algorithm is also included.

1 Introduction

A popular approach to scattered data interpolation and approximation in $\mathbb{R}^s$ is the use of radial basis functions (RBFs) (see, e.g., [12] and references therein). The standard RBF interpolation method can yield very high (in some cases even spectral) approximation orders. A disadvantage is the fact that large (possibly dense) linear systems need to be solved. One of the major advantages of RBFs is the fact that they are a meshfree method which is easily adapted to arbitrary space dimensions s. An alternative meshfree method is the so-called moving least-squares (MLS) method. This method, in general, does not interpolate but instead yields only an approximation of the scattered data. In its traditional form, the MLS method also requires the solution of a small linear system (which changes with the evaluation point, and therefore must be resolved for every evaluation). The size of these linear systems is governed by the degree of polynomials which are reproduced by the method, which in turn determines the approximation order of the method. A connection between the RBF, MLS and quasi-interpolation methods is the fact that they can all be based on the same (radial) functions. The major advantage of quasi-interpolants is that no linear systems need to be solved. The only work required for the computation of the approximation at a certain point is the evaluation of a sum (over all data points in the support of the weight function). We will show how the MLS method can be interpreted as a

* Supported by the National Science Foundation under grant DMS-0073636.

quasi-interpolant, and use the idea of approximate approximation (see, e.g., [9], [10]) to compute the generating functions. By doing this we obtain a fast explicit approximate MLS method with high approximation order.

Throughout the paper we will be using multi-index notation. We call $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_s) \in \mathbb{N}^s$ a *multi-index* with *length* $|\boldsymbol{\alpha}| = \sum_{i=1}^{s} \alpha_i$. If $\boldsymbol{x} \in \mathbb{R}^s$, then the monomials are $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \cdots x_s^{\alpha_s}$.

The layout of the paper is as follows. In Sect. 2 we show how MLS approximation can be viewed as quasi-interpolation via the Backus-Gilbert approach. In Sect. 3 we present a heuristic approach to higher-order Shepard methods based on certain discrete moment conditions for the MLS weight functions. Since these discrete moment conditions are difficult to satisfy, we turn to the concept of approximate approximations in Sect. 4. There we approximate the discrete moment conditions by a continuous version with a saturation error that can be made arbitrarily small by choosing a certain scaling parameter appropriately. The paper is concluded with numerical experiments and closing remarks.

2 Moving Least-Squares Approximation

Moving least-squares methods have emerged as the basis of numerous meshfree approximation methods that are being suggested by practitioners as an alternative to the traditional finite element method (see, e.g., [8] and references therein). The MLS method seems to be attributable to Lancaster and Šalkauskas [6] with special cases going back to McLain [11] and Shepard [15].

Assume we are given data $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^N \subset \mathbb{R}^s \times \mathbb{R}$ with distinct data sites $\boldsymbol{x}_i$ and f some (smooth) function. We then want to approximate the data with an expansion of the form

$$\mathcal{Q}f(\boldsymbol{x}) = \sum_{i=1}^{N} f(\boldsymbol{x}_i)\psi_i(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^s \ . \tag{1}$$

In the approximation theory literature this kind of expansion is referred to as a *quasi-interpolant* since the *generating functions* ψ_i in general do not satisfy the cardinality property $\psi_i(\boldsymbol{x}_j) = \delta_{ij}$, where δ in the usual Kronecker-delta. Our goal is to determine the functions ψ_i in such a way that (1) can be computed efficiently and has high approximation order. To ensure the latter we enforce polynomial reproduction, i.e.,

$$\sum_{i=1}^{N} p(\boldsymbol{x}_i)\psi_i(\boldsymbol{x}) = p(\boldsymbol{x}), \qquad \text{for all } p \in \Pi_d^s \ . \tag{2}$$

Here Π_d^s is the space of s-variate polynomials of total degree at most d with $\dim \Pi_d^s = \binom{s+d}{d} =: m$. In addition, a weighted norm of the generating functions is minimized according to

$$\frac{1}{2}\sum_{i=1}^{N}\psi_i^2(\boldsymbol{x})w_i(\boldsymbol{x}) \to \min \ . \tag{3}$$

Here the w_i are positive (radial) weight functions whose value increases with the distance from the center. This interpretation of MLS approximation was referred to as *Backus-Gilbert* approach by Bos and Šalkauskas [1].

Using Lagrange multipliers $\lambda_j(\boldsymbol{x})$ (which will depend on the evaluation point $\boldsymbol{x}$) we can compute the generating functions

$$\psi_i(\boldsymbol{x}) = \frac{1}{w_i(\boldsymbol{x})}\sum_{j=1}^{m}\lambda_j(\boldsymbol{x})p_j(\boldsymbol{x}_i), \qquad i = 1,\dots,N, \tag{4}$$

where the p_j form a basis of Π_d^s and the Lagrange multipliers $\lambda_j(\boldsymbol{x})$ are obtained as the solution of

$$G\boldsymbol{\lambda} = \boldsymbol{p} \ . \tag{5}$$

Here $\boldsymbol{p} = [p_1(\boldsymbol{x}),\dots,p_m(\boldsymbol{x})]^T$, $\boldsymbol{\lambda} = [\lambda_1(\boldsymbol{x}),\dots,\lambda_m(\boldsymbol{x})]^T$, and the matrix G is a Gram matrix with entries $\langle p_j, p_k\rangle_{\phi(\boldsymbol{x})}$, $j,k = 1,\dots,m$. The subscript on the inner product indicates that it is weighted by the (moving) weight functions $\phi_i(\boldsymbol{x}) = \dfrac{1}{w_i(\boldsymbol{x})}$, i.e.,

$$\langle u, v\rangle_{\phi(\boldsymbol{x})} = \sum_{i=1}^{N} u(\boldsymbol{x}_i)v(\boldsymbol{x}_i)\phi_i(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^d \text{ fixed} \ . \tag{6}$$

From now on we will refer to the weight function in terms of ϕ.

The connection between the Backus-Gilbert approach and the standard MLS formulation (as originally described by Lancaster and Šalkauskas [6]) is discussed in detail in [1] as well as in the recent papers [7] and [16]. For our purposes all we need to know is that in the standard approach one starts with

$$\mathcal{P}f(\boldsymbol{x}) = \sum_{j=1}^{m} c_j(\boldsymbol{x})p_j(\boldsymbol{x}) \ , \tag{7}$$

where the coefficients $c_j(\boldsymbol{x})$ (again depending on $\boldsymbol{x}$) are determined by solving the normal equations

$$\sum_{j=1}^{m} c_j\langle p_j, p_k\rangle_{\phi(\boldsymbol{x})} = \langle f, p_k\rangle_{\phi(\boldsymbol{x})}, \qquad k = 1,\dots,m, \tag{8}$$

resulting from a best least squares approximation based on the inner product given in (6). Note that the normal equations have the same system matrix G as above. Since the two approaches are equivalent, we have $\mathcal{P}f = \mathcal{Q}f$, and we will use the quasi-interpolation notation $\mathcal{Q}f$ from now on.

If we choose shifted monomials as basis of the polynomial reproduction space, i.e., $\Pi_d^s = \text{span}\{(\cdot - \boldsymbol{x})^{\boldsymbol{\alpha}} : \boldsymbol{x} \in \mathbb{R}^s \text{ fixed}, |\boldsymbol{\alpha}| \leq d\}$, then much of the following simplifies, and numerical work becomes more stable, since the polynomials are expanded about the evaluation point $\boldsymbol{x}$. For this choice of local basis it is immediately apparent from (7) that – for any degree d –

$$\mathcal{Q}f(\boldsymbol{x}) = c_1(\boldsymbol{x}) \quad . \tag{9}$$

In order to discuss approximation orders we now introduce the following scale of quasi-interpolants

$$\mathcal{Q}_h f(\boldsymbol{x}) = \sum_{i=1}^{N} f(\boldsymbol{x}_i)\psi\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{\rho_i}\right), \tag{10}$$

where $\rho_i = ch$, for some constant c and "meshsize" h of the point set $\{\boldsymbol{x}_i : i = 1, \ldots, N\}$. Note that we have now identified $\psi_i(\boldsymbol{x}) = \psi(\boldsymbol{x} - \boldsymbol{x}_i)$.

In [7] and [16] the following theorem was established:

Theorem 1. *Let $\Omega \subset \mathbb{R}^s$. If $f \in C^{d+1}(\Omega)$, $\{\boldsymbol{x}_i : i = 1, \ldots, N\} \subset \Omega$ are quasi-uniformly distributed with "meshsize" h, the weight functions $\phi_i(\boldsymbol{x}) = \phi(\boldsymbol{x} - \boldsymbol{x}_i)$ are compactly supported with support size $\rho_i = ch$ ($c = \text{const.}$), and if polynomials in Π_d^s are reproduced according to (2), then the scaled MLS approximation (10) with ψ determined via (4) and (5) satisfies*

$$\|f - \mathcal{Q}_h f\|_\infty \leq Ch^{d+1} \quad .$$

Example 2 (Shepard's method). If we take $d = 0$ in (2), i.e., $\Pi_0^s = \text{span}\{1\}$, then the system (5) consists only of a single equation, and it is easy to see that

$$\psi_i(\boldsymbol{x}) = \frac{\phi_i(\boldsymbol{x})}{\sum_{\ell=1}^{n} \phi_\ell(\boldsymbol{x})} \quad .$$

This results in the quasi-interpolant

$$\mathcal{S}f(\boldsymbol{x}) = \sum_{i=1}^{N} f(\boldsymbol{x}_i)\frac{\phi_i(\boldsymbol{x})}{\sum_{\ell=1}^{N} \phi_\ell(\boldsymbol{x})} \tag{11}$$

which is known as *Shepard's method* [15] and has approximation order $\mathcal{O}(h)$.

Note that the Backus-Gilbert approach leads to an explicit formula for the quasi-interpolant (MLS approximation) (1) valid for any $\boldsymbol{x} \in \mathbb{R}^s$ only in the case $d = 0$. If the degree d of polynomial reproduction in (2) is higher, then the Lagrange multipliers have to be determined through the solution of an $m \times m$ linear system. In [2] we presented a simple method based on analytical solution of these (small) linear systems leading to explicit formulas for the λ_j and thus also $\mathcal{Q}f$. We use that method to represent the standard MLS approach in the numerical experiments in Sect. 5.

In this paper we will introduce another approach based on the approximate satisfaction of discrete moment conditions which has a close connection to approximate approximation (see Sect. 4). This results in explicit formulas similar to (11), but based on higher-degree polynomial spaces.

3 High-order Shepard Methods via Moment Conditions

We now present some heuristic considerations for obtaining high-order Shepard methods. This will result in certain conditions on the moments of the weight function ϕ.

Recall that for the basis of shifted monomials $\mathcal{Q}f(x) = c_1(x)$ (see (9)) for any d. Therefore, if we can find weight functions ϕ_i such that the first row (and first column) of G becomes $[\sum \phi_i(\boldsymbol{x}), 0, \ldots, 0]^T$, then $c_1(x) = \langle f, 1\rangle_{\phi(x)} / \langle 1, 1\rangle_{\phi(x)}$ via (8). Thus,

$$\mathcal{Q}f(x) = \mathcal{S}f(x) = \sum_{i=1}^{N} f(x_i) \frac{\phi_i(x)}{\sum_{\ell=1}^{N} \phi_\ell(x)} \quad ,$$

but now – by construction – the method has approximation order $\mathcal{O}(h^{d+1})$ (instead of the mere $\mathcal{O}(h)$ of Shepard's method which permits the use of *arbitrary* weights).

We therefore define the *discrete moments*

$$\mu_{\boldsymbol{\alpha}} = \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x})^{\boldsymbol{\alpha}} \phi_i(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^s, \qquad 0 \le |\boldsymbol{\alpha}| \le d, \tag{12}$$

where $\boldsymbol{\alpha}$ is a multi-index, and then demand

$$\mu_{\boldsymbol{\alpha}} = \delta_{\boldsymbol{\alpha}\mathbf{0}}, \qquad 0 \le |\boldsymbol{\alpha}| \le d. \tag{13}$$

The condition $\mu_{\mathbf{0}} = 1$ ensures that the ϕ_i also form a *partition of unity*, and we end up with a quasi-interpolant of the form

$$\mathcal{Q}f(\boldsymbol{x}) = \sum_{i=1}^{N} f(\boldsymbol{x}_i)\phi_i(\boldsymbol{x}) \quad .$$

We can summarize our heuristics in

Proposition 3. *Let $\Omega \subset \mathbb{R}^s$. If $f \in C^{d+1}(\Omega)$, $\{\boldsymbol{x}_i : i = 1, \ldots, N\} \subset \Omega$ are quasi-uniformly distributed with "meshsize" h, the weight functions $\phi_i(\boldsymbol{x}) = \phi(\boldsymbol{x} - \boldsymbol{x}_i)$ are compactly supported with support size $\rho_i = ch$ ($c =$ const.), and the discrete moment conditions*

$$\mu_{\boldsymbol{\alpha}} = \delta_{\boldsymbol{\alpha}\mathbf{0}}, \qquad 0 \le |\boldsymbol{\alpha}| \le d,$$

are satisfied, then

$$\mathcal{Q}_h f(\boldsymbol{x}) = \sum_{i=1}^{N} f(\boldsymbol{x}_i) \phi\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{\rho_i}\right)$$

has approximation order $\mathcal{O}(h^{d+1})$.

The discrete moment conditions in Proposition 3 lead us to the following interpretation of the weight function ϕ:

$$\phi(\boldsymbol{x}) = \psi_0(\boldsymbol{x})q(\boldsymbol{x}), \qquad q \in \Pi_d^s, \tag{14}$$

where ψ_0 is a new (arbitrary) weight function, and q is a polynomial of degree d orthonormal with respect to ψ_0 in the sense of (6). This view is also taken by Liu and co-workers (see, e.g., [8]) where q is called the *correction function.*

The discrete moment conditions in Proposition 3 are difficult to satisfy, as is the construction of discrete multivariate orthogonal polynomials as needed for (14). In order to obtain quasi-interpolants for arbitrary space dimensions and scattered data sites we consider the concept of approximate approximation in the next section. There one satisfies continuous moment conditions while ensuring that the discrete moment conditions are almost satisfied.

4 Approximate Approximation

We now give the main results on approximate approximation required for our work. In this paper we restrict ourselves to results for regular center distributions. However, irregularly spaced data are also allowed by the theory (see, e.g., [10]). In [10] Maz'ya and Schmidt present a quasi-interpolation scheme of the form

$$\mathcal{M}_h f(\boldsymbol{x}) = \mathcal{D}^{-s/2} \sum_{\boldsymbol{\nu}\in\mathbb{Z}^s} f(\boldsymbol{x}_{\boldsymbol{\nu}})\psi\left(\frac{\boldsymbol{x}-\boldsymbol{x}_{\boldsymbol{\nu}}}{\sqrt{\mathcal{D}}h}\right), \tag{15}$$

where the data sites $\boldsymbol{x}_{\boldsymbol{\nu}} = h\boldsymbol{\nu}$ are required to lie on a regular lattice with meshsize h. The parameter $\mathcal{D}$ scales the support of the generating function ψ. As we will see below, it can be chosen to make a so-called *saturation error* so small that it does not affect numerical computations. The generating function is required to satisfy the continuous moment conditions

$$\int_{\mathbb{R}^s} \boldsymbol{y}^{\boldsymbol{\alpha}}\psi(\boldsymbol{y})d\boldsymbol{y} = \delta_{\boldsymbol{\alpha}\boldsymbol{0}}, \qquad 0 \le |\boldsymbol{\alpha}| \le d \ . \tag{16}$$

This is the continuous analog of (13).

The following *approximate approximation* result is due to Maz'ya and Schmidt (see, e.g., [10]).

Theorem 4. *Let $f \in C^{d+1}(\mathbb{R}^s)$, $\{\boldsymbol{x}_{\boldsymbol{\nu}} : \boldsymbol{\nu} \in \mathbb{Z}^s\} \subset \mathbb{R}^s$ and ψ a continuous generating function which satisfies the moment condition (16) along with the decay requirement*

$$|\psi(\boldsymbol{x})| \le A_K(1+\|\boldsymbol{x}\|^2)^{-K/2}, \qquad \boldsymbol{x} \in \mathbb{R}^s,$$

where A_K is some constant, $K > d+s+1$ and d is the desired degree of polynomial reproduction. Then

$$\|f - \mathcal{M}_h f\|_\infty = \mathcal{O}(h^{d+1} + \varepsilon_0(\psi, \mathcal{D})) \ . \tag{17}$$

Using Poisson's summation formula one can bound the *saturation error* ε_0 by (see Lemma 2.1 in [9])

$$\varepsilon_0(\psi, \mathcal{D}) \leq \sum_{\boldsymbol{\nu} \in \mathbb{Z}^s \setminus \{\mathbf{0}\}} \mathcal{F}\psi(\sqrt{\mathcal{D}}\boldsymbol{\nu}) \ , \tag{18}$$

where $\mathcal{F}\psi$ is the Fourier transform of ψ defined via

$$\mathcal{F}\psi(\boldsymbol{\omega}) = \int_{\mathbb{R}^s} \psi(\boldsymbol{x}) e^{-2\pi i \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle} d\boldsymbol{x} \ .$$

Here $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product in $\mathbb{R}^s$. The saturation error can be interpreted as the discrepancy between the continuous and discrete moment conditions, and its influence can be controlled by the choice of the parameter $\mathcal{D}$ in (15). Theorem 6.1 of [14] gives the following formula for the Fourier transform of a radial function (after adjusting for the different definition of the Fourier transform in that paper):

$$\mathcal{F}\phi(r) = 2\pi r^{-\frac{s-2}{2}} \int_0^\infty \phi(t) t^{\frac{s}{2}} J_{\frac{s-2}{2}}(2\pi r t) dt \ . \tag{19}$$

With the help of (19), the leading term of (18) gives us an estimate for $\mathcal{D}$ for any desired saturation error. If $\mathcal{D}$ is chosen large enough, then the saturation error will be smaller than the machine accuracy for any given computer, and therefore not noticeable in numerical computations. This means, that even though – theoretically – the quasi-interpolation scheme (15) fails to converge, it does converge for numerical purposes. Moreover, the rate of convergence, $d+1$, can be arbitrarily high.

We note that the error bound (17) is for infinite lattices, but an analogous result also holds for finite lattices (see [5]).

4.1 Construction of Generating Functions

In order to construct high-order Shepard functions we assume the generating function $\psi(\boldsymbol{x}) = \psi_0(\|\boldsymbol{x}\|^2) q(\|\boldsymbol{x}\|^2)$ is radial. Therefore, we want (cf. (16))

$$\int_{\mathbb{R}^s} \|\boldsymbol{x}\|^{2k} q(\|\boldsymbol{x}\|^2) \psi_0(\|\boldsymbol{x}\|^2) dx = \delta_{k0}, \qquad 0 \leq k \leq d \ ,$$

with $q \in \Pi_d^1$ to get approximation order $\mathcal{O}(h^{2d+2})$. By using s-dimensional spherical coordinates and the substitution $y = r^2$ we now obtain a set of one-dimensional orthogonality conditions

$$\frac{\pi^{s/2}}{\Gamma(s/2)} \int_0^\infty y^{k-1} q(y) \psi_0(y) y^{s/2} dy = \delta_{k0}, \qquad 0 \leq k \leq d \ . \tag{20}$$

Thus, the strategy for constructing high-order Shepard functions is to pick an arbitrary (univariate) weight function ψ_0 and then compute the coefficients of $q \in \Pi_d^1$ via the (univariate) moment conditions (20). This leads to the (multivariate) radial generating function $\psi(\boldsymbol{x}) = \psi_0(\|\boldsymbol{x}\|^2)q(\|\boldsymbol{x}\|^2)$.

In the following example we do this for the generating functions used in our numerical experiments in $\mathbb{R}^2$.

Example 5. We start with the weight $\psi_0(r) = (1-\sqrt{r})_+^4\,(4\sqrt{r}+1)$. Then

$$\psi(\boldsymbol{x}) = \frac{7}{\pi}\psi_0(\|\boldsymbol{x}\|^2) = \frac{7}{\pi}\,(1-\|\boldsymbol{x}\|)_+^4\,(4\|\boldsymbol{x}\|+1), \qquad \boldsymbol{x} \in \mathbb{R}^2 \ , \tag{21}$$

has approximation order $\mathcal{O}(h^2)$, and

$$\psi(\boldsymbol{x}) = \psi_0(\|\boldsymbol{x}\|^2)\frac{7}{\pi}\left(\frac{504}{229} - \frac{1980}{229}\|\boldsymbol{x}\|^2\right) \tag{22}$$

has one vanishing (continuous) moment and approximation order $\mathcal{O}(h^4)$.

The basic generating function used in Example 5 corresponds to a much-used compactly supported RBF (see, e.g., [14], Theorem 6.3).

Another interesting example is given by $\psi_0(r) = e^{-r}$ which results in Gaussians as basic generating functions. The associated polynomials q turn out to be Laguerre polynomials (see [9]). We study this case in combination with fast summation techniques in [3].

5 Numerical Experiments

We use the following test functions in our examples

$$\begin{aligned} g(x,y) &= \frac{3}{4}\left[\exp\left(-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right) + \exp\left(-\frac{(9x+1)^2}{49} - \frac{(9y+1)^2}{10}\right)\right] \\ &\quad + \frac{1}{2}\exp\left(-\frac{(9x-7)^2}{4} - (9y-3)^2\right) - \frac{1}{5}\exp\left(-(9x-4)^2 - (9y-7)^2\right), \\ f(x,y) &= 15\exp\left(\frac{-1}{1-4(x-1/2)^2}\right)\exp\left(\frac{-1}{1-4(y-1/2)^2}\right) g(x,y) \ . \end{aligned}$$

Due to space limitations all examples given below use the mollified test function f. This is needed for the approximate MLS algorithm to perform well. All other methods yield comparable results for data given by g or by f.

The ℓ_2 errors reported below are always computed on one more refinement of the finest mesh employed in the example via

$$\ell_2\text{–error} = \sqrt{\frac{1}{M}\sum_{j=1}^{M} |f(\boldsymbol{x}_j) - s(\boldsymbol{x}_j)|^2} \ ,$$

where M is the number of points in the evaluation mesh, and s is the approximation to f obtained by the various algorithms.

It was shown in [13] that in order to have a computationally efficient and accurate algorithm based on interpolation with compactly supported RBFs one should use a multilevel algorithm as given by

Algorithm 6. *Create nested point sets* $\mathcal{X}_1 \subset \cdots \subset \mathcal{X}_K \subset \mathbb{R}^s$, *let* $s(\boldsymbol{x}) = 0$. *For* $k = 1, 2, \ldots, K$ *do*

1. *Solve* $u(\boldsymbol{x}) = f(\boldsymbol{x}) - s(\boldsymbol{x})$ *on* $\mathcal{X}_k$.
2. *Update* $s(\boldsymbol{x}) = s(\boldsymbol{x}) + u(\boldsymbol{x})$.

The representation of u at step k depends on the method we use. For RBF interpolation

$$u(\boldsymbol{x}) = \sum_{\boldsymbol{x}_j \in \mathcal{X}_k} c_j^{(k)} \phi\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_j\|}{\rho_k}\right)$$

with $\rho_k \simeq h_k$. This requires the solution of a (sparse) linear system whose size is equal to the number of points in $\mathcal{X}_k$. For the MLS methods

$$u(\boldsymbol{x}) = \sum_{\boldsymbol{x}_j \in \operatorname{supp}\phi_j} (f - s)(\boldsymbol{x}_j)\phi\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_j\|}{\rho_k}\right) ,$$

where ϕ is either determined by the classical approach or as a high-order Shepard function (see Example 5).

In the first example (see Tab. 1) we use the multilevel algorithm for RBF interpolation with the compactly supported function $\phi(r) = (1-r)_+^4(4r+1)$ as a benchmark. The initial scale ρ_1 is chosen so that the function is supported on $[-2, 2]$. Subsequent scales are successively divided by 2 – just as the meshsize of the computational grids $\mathcal{X}_k$. The rate listed in the tables is the exponent $d+1$ in $\mathcal{O}(h^{d+1})$. The %nonzero column indicates the sparsity of the system matrix, and the time is measured in seconds. The multilevel RBF interpolation yields the most accurate fit – at a comparatively high cost.

Table 1. Multilevel interpolation with CS-RBF $\phi(r) = (1-r)_+^4(4r+1)$.

Mesh	ℓ_2-error	rate	% nonzero	time
3×3	2.367490e-01		100	0
5×5	6.665899e-02	1.828	57.8	0
9×9	2.087575e-02	1.675	23.2	0
17×17	1.090837e-04	4.258	7.47	0
33×33	1.497227e-04	2.865	2.13	5
65×65	5.313053e-05	1.495	0.06	30
129×129	1.112638e-05	2.256	0.01	235

For comparison purposes we also list the results for standard MLS approximation with constant (Shepard's method) and linear precision (see Tab. 2).

The program is written so that no linear systems need to be solved, i.e., the local 3×3 systems for linear precision are solved once analytically, and then incorporated into the code (see [2]). One can observe that Shepard's method actually performs much better than the predicted $\mathcal{O}(h)$.

Table 2. MLS approximation with weight $\phi(r) = (1-r)_+^4(4r+1)$.

Mesh	Shepard			linear precision		
	ℓ_2-error	rate	time	ℓ_2-error	rate	time
3×3	2.737339e-01		7	2.749670e-01		14
5×5	1.100713e-01	1.314	7	1.033060e-01	1.412	13
9×9	5.393041e-02	1.029	5	5.242492e-02	0.979	9
17×17	1.507797e-02	1.839	3	1.502361e-02	1.803	5
33×33	4.124059e-03	1.870	3	4.111092e-03	1.870	4
65×65	1.061904e-03	1.957	2	1.047348e-03	1.973	3
129×129	2.628645e-04	2.014	2	2.628645e-04	1.994	3

We also try an MLS analog of the multilevel algorithm (Tab. 3). Notice that this algorithm improves the accuracy considerably at very little extra cost. In $\mathbb{R}^2$ this effect is much more pronounced than in $\mathbb{R}$ (cf. [2]).

Table 3. Multilevel MLS approximation with weight $\phi(r) = (1-r)_+^4(4r+1)$.

Mesh	Shepard			linear precision		
	ℓ_2-error	rate	time	ℓ_2-error	rate	time
3×3	2.737339e-01		7	2.749670e-01		14
5×5	1.076424e-01	1.347	7	1.013114e-01	1.440	12
9×9	3.909725e-02	1.461	5	4.308322e-02	1.234	9
17×17	7.327282e-03	2.416	3	8.549613e-03	2.333	6
33×33	9.545860e-04	2.940	2	8.937409e-04	3.258	4
65×65	1.424136e-04	2.745	2	9.896052e-05	3.175	3
129×129	3.946680e-05	1.851	2	1.361339e-05	2.872	2

For the approximate MLS approximation (see Tab. 4) we use the generating functions of Example 5 with $\mathcal{D} = 18$ (resulting in $\mathcal{O}(10^{-5})$ saturation error) for the function (21) and $\mathcal{D} = 50$ ($\mathcal{O}(10^{-4})$ saturation) for (22). These values are chosen so that the maximum possible convergence rate is reached. A larger value of $\mathcal{D}$ will result in a more pronounced smoothing effect causing the first few iterates to converge very slowly. A small value of $\mathcal{D}$ will result in better initial convergence, but with a larger saturation error, and therefore early termination of convergence. The basic method does reach the predicted rate of $\mathcal{O}(h^2)$, the "linear" method does not quite reach $\mathcal{O}(h^4)$.

Table 4. Approximate MLS approximation with $\psi_0(r) = (1-\sqrt{r})_+^4(4\sqrt{r}+1)$.

Mesh	basic method			1 vanishing moment		
	ℓ_2-error	rate	time	ℓ_2-error	rate	time
3×3	2.968617e-01		8	2.987532e-01		8
5×5	2.133361e-01	0.477	8	2.235494e-01	0.418	8
9×9	1.324904e-01	0.687	10	1.371051e-01	0.705	13
17×17	5.513583e-02	1.265	8	4.629264e-02	1.566	14
33×33	1.703217e-02	1.695	7	7.805027e-03	2.568	15
65×65	4.589252e-03	1.892	6	1.034781e-03	2.915	15
129×129	1.184532e-03	1.954	6	1.173894e-04	3.140	15

Finally, a multilevel approximate MLS algorithm (Tab. 5) yields improved errors. We use slightly lower values of $\mathcal{D}$ (10 and 20, respectively), since the values used in Tab. 4 lead to too much smoothing, and therefore larger errors.

Table 5. Multilevel approximate MLS method with $\psi_0(r) = (1-\sqrt{r})_+^4(4\sqrt{r}+1)$.

Mesh	basic method			1 vanishing moment		
	ℓ_2-error	rate	time	ℓ_2-error	rate	time
3×3	2.894757e-01		7	2.866462e-01		8
5×5	1.746920e-01	0.729	8	1.630121e-01	0.814	8
9×9	8.872801e-02	0.977	7	7.464709e-02	1.127	8
17×17	2.367211e-02	1.906	6	1.233242e-02	2.598	8
33×33	3.263629e-03	2.859	5	1.575959e-03	2.968	7
65×65	4.392617e-04	2.893	4	1.789843e-04	3.138	7
129×129	5.777104e-05	2.927	4	2.187214e-05	3.033	6

6 Closing Remarks

We have presented a computationally efficient and accurate meshfree method based on MLS and approximate approximation. Our work suggests that one can change the interpretation of Shepard's method: if the weight function has sufficiently many vanishing moments, arbitrarily high orders of convergence may be achieved. This explains why many experiments with Shepard's method perform much better than the (traditionally) expected $\mathcal{O}(h)$ convergence rate (cf. Tab. 2).

Computation with local quasi-interpolants seems to be very sensitive to the choice of $\mathcal{D}$, and often requires very large values of $\mathcal{D}$ to achieve the desired rate of convergence (at the cost of much smoothing at initial stages). This phenomenon was also observed in [4], and we plan to explore the analogy of approximate MLS approximation with smoothers for the multilevel solution of differential equations in the future. Global weights (e.g., Gaussians) seem much more stable, but require fast summation techniques. They are studied in [3].

References

1. Bos, L. P., Šalkauskas, K.: Moving least-squares are Backus-Gilbert optimal. J. Approx. Theory **59** (1989) 267–275
2. Fasshauer, G. E.: Matrix-free multilevel moving least-squares methods. Illinois Institute of Technology (preprint 2001)
3. Fasshauer, G. E.: High-order moving least-squares approximation via fast radial Laguerre transforms. Illinois Institute of Technology (in preparation)
4. Fasshauer, G. E., Jerome, J. W.: Multistep approximation algorithms: Improved convergence rates through postconditioning with smoothing kernels: Adv. in Comput. Math. **10** (1999) 1–27
5. Ivanov, T., Maz'ya, V., Schmidt, G.: Boundary layer approximate approximations and cubature of potentials in domains. Adv. in Comput. Math. **10** (1999) 311–342
6. Lancaster, P., Šalkauskas, K.: Surfaces generated by moving least squares methods. Math. Comp. **37** (1981) 141–158
7. Levin, D.: The approximation power of moving least-squares. Math. Comp. **67** (1998) 1517–1531
8. Li, S., Liu, W. K.: Meshfree and particle methods and their applications. Applied Mechanics Review (to appear)
9. Maz'ya, V., Schmidt, G.: On approximate approximations using Gaussian kernels. IMA J. Numer. Anal. **16** (1996) 13–29
10. Maz'ya, V., Schmidt, G.: On quasi-interpolation with non-uniformly distributed centers on domains and manifolds. J. Approx. Theory **110** (2001) 125–145
11. McLain, D. H.: Drawing contours from arbitrary data points. Comput. J. **17** (1974) 318–324
12. Schaback, R.: Creating surfaces from scattered data using radial basis functions. Mathematical Methods for Curves and Surfaces, Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds.), Vanderbilt University Press (1995) 477–496
13. Schaback, R.: On the efficiency of interpolation by radial basis functions. Surface Fitting and Multiresolution Methods, A. LeMéhauté, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press (1997), 309–318
14. Schaback, R., Wendland, H.: Characterization and construction of radial basis functions. Multivariate approximation and applications, N. Dyn, D. Leviatan, D. Levin, and A. Pinkus (eds.), Cambridge Univ. Press, Cambridge (2001), 1–24
15. Shepard, D.: A two dimensional interpolation function for irregularly spaced data. Proc. 23rd Nat. Conf. ACM (1968) 517–523
16. Wendland, H.: Local polynomial reproduction and moving least squares approximation. IMA J. Numer. Anal. **21** (2001) 285–300