WCCM V
Fifth World Congress on
Computational Mechanics
July 7–12, 2002, Vienna, Austria
Eds.: H.A. Mang, F.G. Rammerstorfer,
J. Eberhardsteiner

Approximate Moving Least-Squares Approximation for Time-Dependent PDEs

Gregory E. Fasshauer*
Illinois Institute of Technology
Department of Applied Mathematics
Chicago, IL, 60616, U.S.A.
e-mail: fass@amadeus.math.iit.edu

Key words: Moving Least Squares, Radial Basis Functions, Approximate Approximation, Partial Differential Equations

Abstract

For multivariate problems with many scattered data locations the use of radial functions has proven to be advantageous. However, using the usual radial basis function approach one needs to solve a large (possibly dense) linear system. In the moving least squares (MLS) method one obtains a best approximation of the given data in a (moving) weighted least-squares sense. The computational burden is now shifted, and one needs to solve many small linear systems. Recently we have employed the theory of approximate approximations (see [12]) to develop a completely matrix-free approximate MLS approximation algorithm. So far we have only discussed applications of this method to scattered data approximation problems (see [5], [6]). In this paper we present a comparison of two approaches to the solution of time dependent (parabolic) PDEs of the form

$$\frac{\partial u}{\partial t}(\mathbf{x},t) = Lu(\mathbf{x},t) + F(\mathbf{x},t), \qquad \mathbf{x} \in \Omega \subset \mathbb{R}^d,\ t > 0,$$

based on the use of approximate moving least-squares approximation. In the first approach one assumes the solution to be an approximate MLS approximation of the form

$$u(\mathbf{x},t) = \sum_{j=1}^{N} \alpha_j(t)\psi_j(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^d,$$

where the generating functions $\psi_j(x) = \Psi(\|x - x_j\|)$ satisfy certain moment conditions to ensure a desired approximation order. This leads to a system of ordinary differential equations for the coefficients $\alpha_j(t)$. Many traditional techniques can be applied to solve this ODE or DAE system. For the second approach one first discretizes in time, and then applies approximate MLS collocation to the spatial part. This part of the solution is analogous to scattered Hermite interpolation. Similarities and differences of the two methods as well as numerical experiments are presented.

1 Introduction

Meshless (or meshfree) methods are becoming increasingly popular for dealing with multivariate approximation problems. Once it has been established that these methods possess good approximation properties it is natural to extend ones investigations to their use for the solution of (partial) differential equations. Various approaches have been suggested in recent years, most of them based on some form of Galerkin approach (see, e.g., [9] and references therein). Another commonly used approach is based on radial basis function collocation (see, e.g., [3] and references therein). This approach is closely related to Hermite interpolation, and generally requires the solution of a large system of linear equations. Motivated by our recent work on matrix-free moving least-squares (MLS) approximation methods (see [4], [5], [6]) we are led to investigate the use of a matrix-free collocation-based approach to the solution of partial differential equations.

In this paper we explore two basic approaches, one akin to the classical method of lines, the other related to Hermite interpolation. We are interested in the fundamental differences and similarities of the two approaches, as well as possible problems involved.

We will show below that the method of lines approach leads to a system of differential-algebraic equations for the coefficients of the radial basis functions of the moving least-squares quasi-interpolant. Implementation of boundary conditions requires some care, and we mention a few possibilities for doing this. The Hermite collocation approach arises from first discretizing the time component of the PDE. One then obtains an elliptic equation, which can be interpreted as a generalized Hermite interpolation/approximation problem. Since solving this problem with a matrix-free moving least-squares method turns out to be rather tricky, we concentrate on the Hermite problem, and leave an implementation of the second method for time-dependent PDEs for future work.

The paper is organized as follows. In the next section we briefly review the idea of approximate approximation of given function-value data. We also develop a matrix-free generalized moving least-squares approximation method for derivative data. In Sect. 3 we present a general description of the two methods for solving time-dependent PDEs. This general discussion is illustrated with two model problems (one-dimensional transport, and one-dimensional diffusion) in Sect. 4. Numerical experiments for the generalized MLS approximation method, as well as for the PDE model problems are given in Sect. 5. The paper is concluded with some remarks in Sect. 6.

2 Matrix-Free Moving Least-Squares Approximation

2.1 Approximation of Function Value Data

First we consider data of the form $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^N \subset \mathbb{R}^d \times \mathbb{R}$ with distinct data sites $\mathbf{x}_i$, and f some (smooth) function. Based on the theory of approximate approximations by Maz'ya and Schmidt (see, e.g., [12]) we can approximate the data with an expansion of the form

$$\mathcal{Q}_h f(\mathbf{x}) = \frac{1}{\sqrt{\mathcal{D}^d}} \sum_{j=1}^N f(\mathbf{x}_j) \psi\left(\frac{\mathbf{x}-\mathbf{x}_j}{\sqrt{\mathcal{D}}h}\right), \qquad \mathbf{x} \in \mathbb{R}^d, \tag{1}$$

where h is the "meshsize" of the collection of data sites, and $\mathcal{D}$ is a parameter which ensures that the saturation error involved in the approximation is below any desired tolerance. In the approximation theory literature this kind of expansion is referred to as a *quasi-interpolant* since, in general, the *generating* *functions* $\psi((\cdot - \mathbf{x}_j)/(\sqrt{\mathcal{D}}h))$ do not satisfy the cardinality property $\psi((\mathbf{x}_i - \mathbf{x}_j)/(\sqrt{\mathcal{D}}h)) = \delta_{ij}$, where δ is the usual Kronecker-delta. Following [12], we showed in [5] and [6] that such generating functions of the form $\psi_j(\mathbf{x}) = \Psi(\|\mathbf{x} - \mathbf{x}_j\|)$ can be constructed with arbitrary approximation order (up to the controllable saturation error) using radial basis functions Ψ_0 as a starting point. The construction of multivariate (radial) generating functions ψ_j in (1) is based on certain univariate orthogonality conditions for the function Ψ (cf. Sct.4.2 in [5]). A few of these generating functions (for various space dimensions) are listed in Table 1. We note that all generating functions below are computed based on equally spaced data sites. It is also possible to handle arbitrarily spaced sites, but this is more complicated, so for simplicity we will focus only on the regular case in this paper. Also, compactly supported radial functions can be used instead of globally supported ones (see [5]).

Table 1: Some globally supported generating functions based on $\Psi_0(r) = e^{-r}$ for approximation in $\mathbb{R}^d$ (see [6]).

d	$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$
1	$\frac{1}{\sqrt{\pi}}e^{-\|\mathbf{x}\|^2}$	$\frac{1}{\sqrt{\pi}}\left(\frac{3}{2} - \|\mathbf{x}\|^2\right)e^{-\|\mathbf{x}\|^2}$	$\frac{1}{\sqrt{\pi}}\left(\frac{15}{8} - \frac{5}{2}\|\mathbf{x}\|^2 + \frac{1}{2}\|\mathbf{x}\|^4\right)e^{-\|\mathbf{x}\|^2}$
2	$\frac{1}{\pi}e^{-\|\mathbf{x}\|^2}$	$\frac{1}{\pi}\left(2 - \|\mathbf{x}\|^2\right)e^{-\|\mathbf{x}\|^2}$	$\frac{1}{\pi}\left(3 - 3\|\mathbf{x}\|^2 + \frac{1}{2}\|\mathbf{x}\|^4\right)e^{-\|\mathbf{x}\|^2}$
3	$\frac{1}{\pi^{3/2}}e^{-\|\mathbf{x}\|^2}$	$\frac{1}{\pi^{3/2}}\left(\frac{5}{2} - \|\mathbf{x}\|^2\right)e^{-\|\mathbf{x}\|^2}$	$\frac{1}{\pi^{3/2}}\left(\frac{35}{8} - \frac{7}{2}\|\mathbf{x}\|^2 + \frac{1}{2}\|\mathbf{x}\|^4\right)e^{-\|\mathbf{x}\|^2}$

In order to guide the choice of $\mathcal{D}$ we consider the saturation error ε_0 introduced by the approximation scheme (1). An estimate is given by (see Lemma 2.1 in [11])

$$\varepsilon_0(\psi, \mathcal{D}) \leq \sum_{\nu \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} \mathcal{F}\psi(\sqrt{\mathcal{D}}\nu) \ , \tag{2}$$

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

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

Here $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product in $\mathbb{R}^d$. Theorem 6.1 of [13] 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):

$$\mathcal{F}\Psi(r) = 2\pi r^{-\frac{d-2}{2}} \int_0^\infty \Psi(t) t^{\frac{d}{2}} J_{\frac{d-2}{2}}(2\pi r t) dt \ , \tag{3}$$

where the J_ν are the classical Bessel functions of the first kind. With the help of (3), the leading term of (2) 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. We will list the choices we make for $\mathcal{D}$ in our numerical experiments below.

2.2 Approximation of Derivative Data

Next we consider a generalized moving least-squares problem. The data now are of the form $\{(\mathbf{x}_i, L_i(f))\}_{i=1}^N \subset \mathbb{R}^d \times \mathbb{R}$ with distinct data sites $\mathbf{x}_i$, and L_i continuous linear functionals, e.g., evaluation of (a combination of) derivatives at $\mathbf{x}_i$ (see, e.g., [8] or [14] for a more detailed discussion of this problem).

For this generalized problem an explicit quasi-interpolation scheme such as (1) is not yet available. We must therefore solve the small linear systems arising in the generalized Backus-Gilbert approach associated with the generalized moving least-squares problem in a way analogous to the method described in [4]. We now describe this generalized MLS method. Our goal is to obtain an approximation of the form

$$\mathcal{G}f(\mathbf{x}) = \sum_{j=1}^{N} L_j(f)\psi_j(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^d, \tag{4}$$

where the generating functions ψ_j are determined by enforcing polynomial reproduction in the sense

$$\sum_{j=1}^{N} L_j(p)\psi_j(\mathbf{x}) = L(p), \qquad \text{for all } p \in \mathcal{P}_Q^d.$$

Here $\mathcal{P}_Q^d$ is the space of d-variate polynomials of degree at most Q. In addition, a weighted norm of the generating functions is minimized according to

$$\frac{1}{2}\sum_{j=1}^{N} \psi_j(\mathbf{x}) w(L_j, L) \rightarrow \min \ .$$

Here w is a function correlating the functionals L_j and L. In our case L is usually evaluation at $\mathbf{x}$, since we are interested in constructing the value of the approximation at $\mathbf{x}$. By using Lagrange multipliers, the generating functions ψ_j are given by

$$\psi_j(x) = \frac{1}{w(L_j, L)} \sum_{k=1}^{Q} \lambda_k L_j(p_k)$$

with the λ_k determined as the unique solution of the linear system

$$\sum_{k=1}^{Q} \lambda_k \sum_{j=1}^{N} \frac{1}{w(L_j, L)} L_j(p_k) L_j(p_\ell) = L(p_\ell), \qquad 1 \le \ell \le Q. \tag{5}$$

We now describe how to construct generating functions for two specific sets of functionals L_j.

2.2.1 Full Hermite Problem

In the full Hermite approximation problem we assume we are given the function value as well as the value of the first derivative of some univariate function f at the data sites $\mathbf{x}_j$, $j = 1, \ldots, N$. Thus,

$$L_j(f) = \begin{cases} f(\mathbf{x}_j), & j = 1, \ldots, N, \\ f'(\mathbf{x}_{j-N}), & j = N+1, \ldots, 2N, \end{cases}$$

and the approximation to f will be of the form (cf. (4))

$$\mathcal{G}f(\mathbf{x}) = \sum_{j=1}^{N} f(\mathbf{x}_j)\psi_j(\mathbf{x}) + \sum_{j=N+1}^{2N} f'(\mathbf{x}_{j-N})\psi_j(\mathbf{x}) \ . \tag{6}$$

For the following derivation we choose $d = 1$ and $Q = 1$, and represent the space of univariate linear polynomials $\mathcal{P}_1^1$ with the basis $\{1, (\cdot - x)\}$ of monomials shifted to the evaluation point x. We represent the weight $1/w(L_j, L)$ by the symmetric expression $\phi_j(x) = \Phi(\|x - x_j\|)$. In this case (5) becomes

$$\begin{bmatrix} \sum_{j=1}^{N} \phi_j(x) & \sum_{j=1}^{N} (x_j - x)\phi_j(x) \\ \sum_{j=1}^{N} (x_j - x)\phi_j(x) & \sum_{j=1}^{N} (x_j - x)^2 \phi_j(x) + \sum_{j=N+1}^{2N} \phi_j(x) \end{bmatrix} \begin{bmatrix} \lambda_1(x) \\ \lambda_2(x) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ ,$$

so that the Lagrange multipliers are

$$\begin{aligned} \lambda_1(x) &= \frac{\sum_{j=1}^{N} (x_j - x)^2 \phi_j(x) + \sum_{j=N+1}^{2N} \phi_j(x)}{\sum_{j=1}^{N} \phi_j(x) \left[\sum_{j=1}^{N} (x_j - x)^2 \phi_j(x) + \sum_{j=N+1}^{2N} \phi_j(x) \right] - \left[\sum_{j=1}^{N} (x_j - x)\phi_j(x) \right]^2} \\ \lambda_2(x) &= -\frac{\sum_{j=1}^{N} (x_j - x)\phi_j(x)}{\sum_{j=1}^{N} \phi_j(x) \left[\sum_{j=1}^{N} (x_j - x)^2 \phi_j(x) + \sum_{j=N+1}^{2N} \phi_j(x) \right] - \left[\sum_{j=1}^{N} (x_j - x)\phi_j(x) \right]^2}, \end{aligned} \tag{7}$$

and the generating functions are given by

$$\psi_j(x) = \begin{cases} \phi_j(x) \left[\lambda_1(x) + \lambda_2(x)(x_j - x) \right], & j = 1, \ldots, N, \\ \lambda_2(x)\phi_j(x), & j = N+1, \ldots, 2N. \end{cases} \tag{8}$$

In this case we end up with an approximation of the form (6) or

$$\mathcal{G}f(x) = \sum_{j=1}^{N} f(x_j) \left[\lambda_1(x) + \lambda_2(x)(x_j - x) \right] \phi_j(x) + \sum_{j=N+1}^{2N} f'(x_{j-N}) \lambda_2(x) \phi_j(x) \ .$$

To simplify matters, we let $\phi_{j+N}(x) = \phi_j(x)$, $j = 1, \ldots, N$. This results in a final approximation of the form

$$\mathcal{G}f(x) = \sum_{j=1}^{N} \left[\left(\widetilde{\lambda}_1(x) + \widetilde{\lambda}_2(x)(x_j - x) \right) f(x_j) + \widetilde{\lambda}_2(x) f'(x_j) \right] \phi_j(x) \ ,$$

where the notation $\widetilde{\lambda}$ reflects the fact that we have made the appropriate changes in the formulas for the Lagrange multipliers.

Note, however, that this is not the same approximation one would obtain by applying a partition of unity approach (modified Shepard's method, see, e.g., [1]) to the Taylor data $L_j(f) = f(x_j) + (x - x_j)f'(x_j)$.

Higher-order polynomial reproduction can also be enforced, but the resulting formulas are too complicated to reproduce here. We have implemented a second-order method using Maple, and some results obtained with this method are included below.

2.2.2 Endpoint Hermite Problem

The previous problem does not really reflect the situation we face when trying to solve boundary-value problems. Therefore, we also briefly give the formulas one obtains for the problem where the functionals L_j (again for a one-dimensional problem) are of the form $L_1(f) = f(x_1)$, $L_N(f) = f(x_N)$, and $L_j(f) = f'(x_j)$, $j = 2, \ldots, N-1$. This corresponds to an (overdetermined) first-order two-point boundary-value problem. Following the same procedure as in the previous section we end up with

$$\mathcal{G}f(x) = \gamma_1(x)f(x_1)\phi_1(x) + \gamma_2(x)f(x_N)\phi_N(x) + \gamma_3(x)\sum_{j=2}^{N-1} f'(x_j)\phi_j(x) \ ,$$

where

$$\begin{aligned}
\gamma_1(x) &= \frac{(x_N - x)(x_N - x_1)\phi_N(x) + \sum_{j=2}^{N-1}\phi_j(x)}{D}, \\
\gamma_2(x) &= \frac{(x_1 - x)(x_1 - x_N)\phi_1(x) + \sum_{j=2}^{N-1}\phi_j(x)}{D}, \\
\gamma_3(x) &= \frac{(x - x_1)\phi_1(x) + (x - x_N)\phi_N(x) + \sum_{j=2}^{N-1}\phi_j(x)}{D},
\end{aligned}$$

and

$$D = (x_1 - x_N)^2\phi_1(x)\phi_N(x) + (\phi_1(x) + \phi_N(x))\sum_{j=2}^{N-1}\phi_j(x) \ .$$

As in the previous section, we have chosen $d = 1$ and $Q = 1$ for this derivation. This approximation scheme is included in our numerical experiments below.

3 Two Methods for Solving Time-dependent PDEs

We now present a general description of two methods for solving time-dependent PDEs of the form

$$u_t(\mathbf{x}, t) = Lu(\mathbf{x}, t) + F(\mathbf{x}, t), \qquad \mathbf{x} \in \Omega,\ t > 0, \tag{9}$$

where L is some linear spatial differential operator, and F is an arbitrary function of space and time. The PDE (9) will be supplemented with an initial condition

$$u(\mathbf{x}, 0) = f(\mathbf{x}), \qquad \mathbf{x} \in \overline{\Omega}, \tag{10}$$

and possibly with boundary conditions

$$Du(\mathbf{x}, t) = g(t), \qquad \mathbf{x} \in \partial\Omega, \tag{11}$$

where D is another differential operator defining the boundary conditions. For both methods we will assume that the solution can be approximated by a meshfree quasi-interpolant of the type (1).

3.1 Method of Lines

For the classical (spectral) method of lines approach one usually assumes

$$u(\mathbf{x},t) \approx \sum_{j=1}^{N} \alpha_j(t)\psi_j(\mathbf{x}) \ , \tag{12}$$

with some set of basis functions $\{\psi_1,\ldots,\psi_N\}$ defined on the spatial domain, and the $\alpha_j(t)$ time-dependent coefficients to be determined.

Collocation of the PDE (9) at N collocation points ξ_i, $i = 1,\ldots,N$, leads to a system of ODEs for the coefficients $\alpha_j(t)$ of the form

$$\sum_{j=1}^{N} \alpha_j'(t)\psi_j(\xi_i) = \sum_{j=1}^{N} \alpha_j(t)L\psi_j(\xi_i) + F(\xi_i,t), \qquad i = 1,\ldots,N.$$

Using matrix-vector notation we can write this problem as

$$\mathbf{A}\alpha'(t) = \mathbf{A}^L\alpha(t) + \mathbf{f} \ , \tag{13}$$

where the matrices $\mathbf{A}$ and $\mathbf{A}^L$ have entries $\mathbf{A}_{ij} = \psi_j(\xi_i)$ and $\mathbf{A}^L_{ij} = L\psi_j(\xi_i)$, and $\mathbf{f}$ contains the forcing term. An initial condition for this ODE system is obtained from the initial condition (10) using meshfree quasi-interpolation, i.e.,

$$u(\mathbf{x},0) = f(\mathbf{x}) \approx \sum_{j=1}^{N} \alpha_j(0)\psi_j(\mathbf{x}) \ .$$

According to (1) we get (with appropriate scaling of the ψ_j)

$$\alpha_j(0) = f(\mathbf{x}_j), \qquad j = 1,\ldots,N,$$

where the $\mathbf{x}_j$ are the centers of the basis functions ψ_j. One can use standard software to solve the ODE system (13) and then obtain an approximation to the solution at time t (for any spatial location $\mathbf{x}$) from (12). We have done this for the examples involving the transport equation below.

The problem becomes more complicated when boundary conditions are added (as for the diffusion equation below). If we add boundary conditions of the type (11), then the system (13) now turns into a system of differential-algebraic equations of the form

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{0} \end{bmatrix} \alpha'(t) = \begin{bmatrix} \mathbf{A}^L \\ \mathbf{A}^D \end{bmatrix} \alpha(t) + \begin{bmatrix} \mathbf{f} \\ -\mathbf{g} \end{bmatrix} \ . \tag{14}$$

In order to have a square system, the number of collocation points ξ_i and centers $\mathbf{x}_j$ need to be equal. We will discuss the choice of these points in more detail below. Again, standard software can be used to solve (14), and an approximate solution is obtained via (12).

We note that an ODE-only approach for this type of problem (with an iterative treatment of boundary conditions) was presented in [7].

3.2 Hermite Collocation

For the second approach we first apply a standard discretization to the time derivative. We will use a standard backward Euler method, but other discretizations are of course also possible. Thus, the PDE (9) becomes

$$u(\mathbf{x},t) \approx u(\mathbf{x},t-\Delta t) + \Delta t \left(Lu(\mathbf{x},t) + F(\mathbf{x},t) \right) \ ,$$

or

$$(I - \Delta t L)u(\mathbf{x},t) \approx u(\mathbf{x},t-\Delta t) + \Delta t F(\mathbf{x},t) \ . \tag{15}$$

Therefore, at every fixed time step t the problem is reduced to solving the elliptic PDE

$$\mathcal{L}u(\mathbf{x},t) \approx \mathcal{F}(\mathbf{x},t) \ , \tag{16}$$

where $\mathcal{L} = I - \Delta t L$, and $\mathcal{F}(\mathbf{x},t) = u(\mathbf{x},t-\Delta t) + \Delta t F(\mathbf{x},t)$. Boundary conditions as in (11) can be added as additional constraints. It is clear that the problem (16) is related to the generalized Hermite problem (see Sect.2.2 or [2]), and it is that problem which we will focus our numerical experiments on.

A similar approach using RBF collocation (which involves the solution of large linear systems) was studied in [10].

4 Two Model Problems

We now illustrate the abstract approaches described in the previous section with two simple model problems.

4.1 1D Transport Equation

Our first model problem is

$$\begin{aligned} u_t(x,t) + cu_x(x,t) &= 0, \qquad x > 0,\ t > 0, \\ u(x,0) &= f(x), \qquad x \geq 0, \end{aligned} \tag{17}$$

where c is the constant wave speed, and f is some finitely supported initial profile. The exact solution to this problem is $u(x,t) = f(x-ct)$.

Using the method of lines approach we obtain the following system of ODEs:

$$\mathbf{A}\alpha'(t) = -c\mathbf{A}^x \alpha(t) \ , \tag{18}$$

where the matrices $\mathbf{A}$ and $\mathbf{A}^x$ have entries $\mathbf{A}_{ij} = \psi_j(\xi_i)$ and $\mathbf{A}^x_{ij} = \psi'_j(\xi_i)$, respectively. Below we will use symmetric basis functions $\psi_j(x) = \psi(x - x_j) = \Psi(|x - x_j|^2)$ as listed in Tab.1.

If the set of centers $\{x_j\}$ coincides with the collocation points $\{\xi_i\}$ then, for the symmetric basis functions we are suggesting, it is known that the matrix $\mathbf{A}$ is nonsingular (see, e.g., [13]). In the case of differing sets this fact is no longer known, and we need to monitor the performance of the ODE solver.

If we discretize first in time and then collocate, we get the problem (cf. (15))

$$(I + c\Delta t \frac{\partial}{\partial x}) u(x,t) \approx u(x,t-\Delta t) \ .$$

This corresponds to a generalized Hermite problem where $L_j(f) = f(x_j) + c\Delta t f'(x_j)$. We study similar Hermite problems below (without the time stepping).

4.2 1D Diffusion Equation

Our second model problem is the one-dimensional diffusion equation (heat equation)

$$\begin{aligned} u_t(x,t) + cu_{xx}(x,t) &= 0, \qquad x \in (0,1),\ t > 0, \\ u(x,0) &= f(x), \qquad x \in [0,1], \end{aligned} \tag{19}$$

where c is the diffusivity, and f represents the initial temperature distribution. We will add homogeneous Dirichlet as well as Neumann boundary conditions below. The exact solution can of course be obtained using Fourier series.

We discuss only the method of lines approach for this problem. We now have to solve a differential-algebraic system as in (14).

To be specific, let's consider the heat equation (19) together with homogeneous Dirichlet boundary conditions at each end. In order to obtain acceptable results we use additional basis functions centered outside the domain, and associated collocation points near the boundary (inside the domain). Thus the set of centers is no longer identical to the set of collocation points, and we have no guarantee that the matrix block $\mathbf{A}$ below has full rank. We take the set of centers $x_j,\ j = 1, \ldots, N+4$, as $\{-2h, -h, 0, h, 2h, \ldots, 1, 1+h, 1+2h\}$, and the collocation points $\xi_i,\ i = 1, \ldots, N+4$, as $\{0, h/4, h/2, h, 2h, \ldots, 1-2h, 1-h, 1-h/2, 1-h/4, 1\}$, where the meshsize h is given by $h = 1/(N-1)$. Then (14) becomes

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{A} \\ \mathbf{0} \end{bmatrix} \alpha'(t) = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}^{xx} \\ \mathbf{A}_N \end{bmatrix} \alpha(t) \ . \tag{20}$$

The matrix blocks are given by

$$\begin{aligned} \mathbf{A}_{ij} &= \psi_j(\xi_i), \qquad i = 2, \ldots, N+3,\ j = 1, \ldots, N+4, \\ \mathbf{A}_{1ij} &= \psi_j(\xi_i), \qquad i = 1,\ j = 1, \ldots, N+4, \\ \mathbf{A}^{xx}_{ij} &= \psi''_j(\xi_i), \qquad i = 2, \ldots, N+3,\ j = 1, \ldots, N+4, \\ \mathbf{A}_{Nij} &= \psi_j(\xi_i), \qquad i = N+4,\ j = 1, \ldots, N+4, \end{aligned}$$

and $\mathbf{0}$ are zero (row-)vectors of length $N+4$. Homogeneous Neumann conditions are implemented similarly, with the first and last rows on the right-hand side replaced with the appropriate derivatives of the basis functions.

5 Numerical Experiments

5.1 Method of Lines

We use either Maple's numerical ODE solver `rk45` or Matlab's stiff ODE solver `ode15s` for the ODE and DAE systems encountered by the method of lines approach.

5.1.1 Transport Equation

We present the results of four experiments. They basically differ in the choice of generating functions ψ_j used for the approximate approximation (1). Our three choices are listed in the row corresponding to

Table 2: Method of lines solution for transport equation.

N	Gaussian, $\mathcal{D}=3$		Gauss-Laguerre1, $\mathcal{D}=3$		Gauss-Laguerre2, $\mathcal{D}=3$		Gauss-Laguerre2, $\mathcal{D}=5$	
	ℓ_∞-err	rate	ℓ_∞-err	rate	ℓ_∞-err	rate	ℓ_∞-err	rate
9	$5.7903(-1)$		$4.2012(-1)$		$3.2148(-1)$		$4.2678(-1)$	
17	$3.5533(-1)$	0.70	$1.5886(-1)$	1.40	$7.5236(-2)$	2.10	$1.5264(-1)$	1.48
33	$1.3457(-1)$	1.40	$2.5122(-2)$	2.66	$7.7958(-3)$	3.27	$1.8499(-2)$	3.05
65	$3.7964(-2)$	1.83	$2.9209(-3)$	3.10	$1.1129(-3)$	2.81	$2.2541(-3)$	3.06

$d=1$ of Table 1 and we denote them by Gaussian, Gauss-Laguerre 1, and Gauss-Laguerre 2 in Table 2. For the first two experiments we choose $\mathcal{D}=3$. For the second order Gauss-Laguerre functions we provide two results: one obtained with the scaling constant $\mathcal{D}$ in (1) set to $\mathcal{D}=3$, the other with $\mathcal{D}=5$. We take wave speed $c=1$ in (17), and let the initial profile be given by $f(x)=64x^3(1-x)^3$ for $x\in[0,1]$, and $f(x)=0$ outside the interval. The simulation is allowed to run for t from 0 to 2. The results displayed in Table 2 are the maximum error at the final time evaluated on a fine evaluation mesh, along with an indication of the h-convergence rate of the method.

We see that the use of a more accurate quasi-interpolant pays off and results in higher accuracy. In particular, going from the basic Gaussian generating function to the first-order Gauss-Laguerre method makes a considerable difference. With the second-order Gauss-Laguerre method the saturation error seems to be starting to play a role in the computation for higher values of N. This claim is substantiated by the fact that, with the larger value of $\mathcal{D}=5$, the rate of convergence does not deteriorate. However, this results in an overall smoothing effect, so that the absolute errors are not significantly smaller than for the first-order method. Also, the accuracy of the time solver, `rk45`, now has an impact on the overall performance.

Snapshots (at time $t=0$, $t=1$, and $t=2$) of the approximate solution to the transport equation are presented in Figures 1 and 2. The dash-dotted (green) curves indicate the exact solution, the solid (red) curves indicate the numerical approximation. The first three plots are for the case $N=33$, and show the

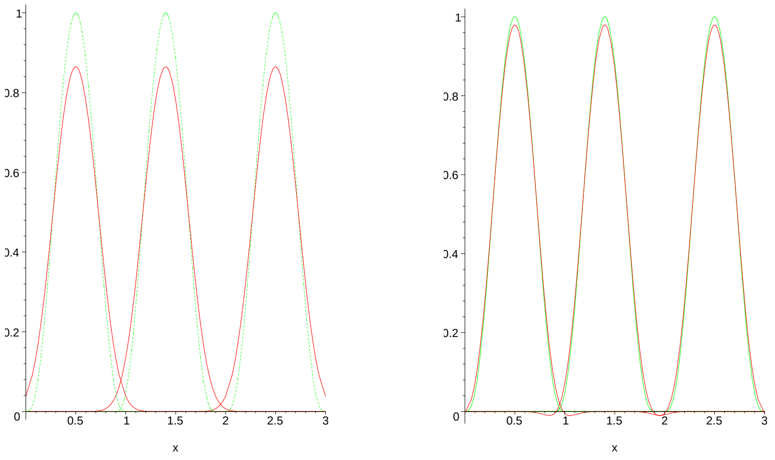

Figure 1: Snapshots (at $t=0$, $t=1$, and $t=2$) of the solution for the transport equation using 33 points. Gaussian approximation (left) and Gauss-Laguerre1 (right).

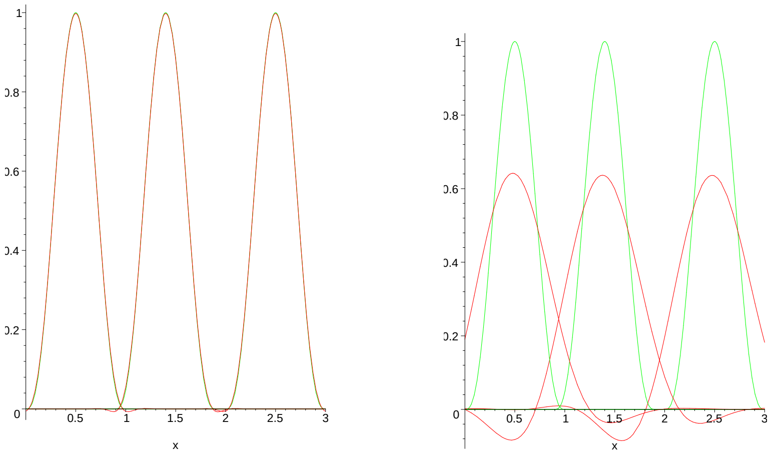

Figure 2: Snapshots (at $t = 0$, $t = 1$, and $t = 2$) of the solution for the transport equation using the Gauss-Laguerre2 method. $N = 33$ points (left) and 9 points (right).

Gauss method, the Gauss-Laguerre1 and Gauss-Laguerre2 method, respectively (all with $\mathcal{D} = 3$). The fourth plot (right part of Fig. 2) shows the Gauss-Laguerre2 method for $N = 9$. We see that the basic Gaussian approximation is not very good. For a low number of points (e.g. $N = 9$ or 17) the higher-order Gauss-Laguerre methods produce solutions that oscillate considerably. This does not happen for the basic Gaussian approximation. It is, however, exactly this oscillatory behavior of the generating functions that ensures the higher rates of approximation.

We would also like to point out that radial basis function theory (essentially Bochner's theorem) guarantees non-singularity (even positive definiteness) of the matrix $\mathbf{A}$ in (18) in case of all three types of generating functions since it is easily verified that their Fourier transforms are positive.

5.1.2 Diffusion Equation

Two sets of experiments were performed. The first for equation (19) with homogeneous Dirichlet boundary conditions. The second for the same partial differential equation, but with homogeneous Neumann boundary conditions. In all examples the diffusivity was set to $c = 1$. For the Dirichlet problem we use the piecewise linear function

$$f(x) = \begin{cases} 2x, & 0 \le x \le 1/2, \\ 2(1-x), & 1/2 \le x \le 1 \end{cases}$$

as initial temperature distribution. For the problem with Neumann boundary conditions we use a piecewise quadratic initial temperature distribution

$$f(x) = \begin{cases} 4x^2, & 0 \le x \le 1/2, \\ 4(1-x)^2, & 1/2 \le x \le 1. \end{cases}$$

We perform two sets of experiments. One without special choice of collocation points, i.e., $x_i = \xi_i$, $i = 1, \ldots, N$, (see Fig. 3 left and Fig. 4 right). The second set is based on the choice of auxiliary collocation points described in Sect. 4.2 (see Fig. 3 right, Fig. 4 left, and Fig. 5). Errors computed at $t = 0$ and $t = 1$ on a fine evaluation mesh are listed in Tables 3 and 4. We see that the error in the approximation of the initial conditions decreases only linearly with the number of collocation points

Table 3: Diffusion equation with Dirichlet boundary conditions.

N	Gaussian		with aux. points	Gauss-Laguerre1 (with aux. points)	
	ℓ_∞-err ($t=0$)	ℓ_∞-err ($t=1$)	ℓ_∞-err ($t=1$)	ℓ_∞-err ($t=0$)	ℓ_∞-err ($t=1$)
17	0.1151	0.0303	0.0318	0.0503	0.0308
33	0.0576	0.0316	0.0323	0.0251	0.0316
65	0.0288	0.0322	0.0325	0.0126	0.0321
129	0.0144	0.0323	0.0326	0.0063	0.0323
257	0.0072	0.0325	0.0325	0.0031	0.0325
513	0.0036	0.0325	0.0326	0.0016	0.0325

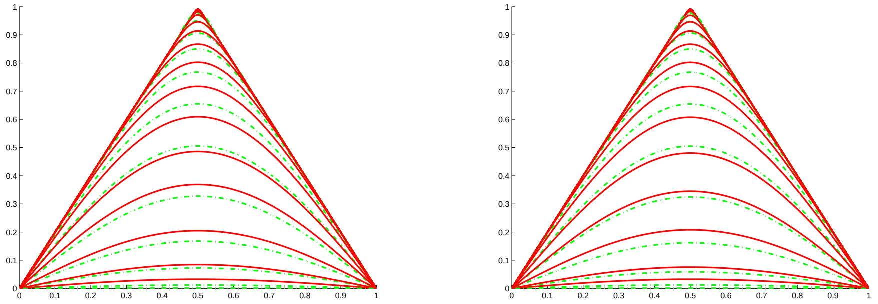

Figure 3: Time profile of the solution of the diffusion equation with Dirichlet boundary conditions using $N = 513$ points. Gaussian approximation without (left) and with (right) auxiliary boundary points.

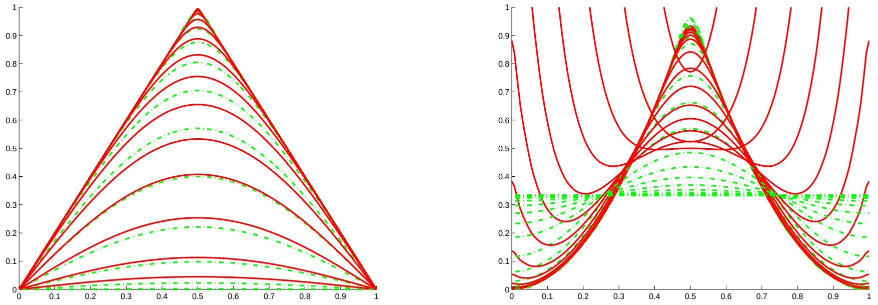

Figure 4: Time profile of the solution of the diffusion equation with Dirichlet boundary conditions using the Gauss-Laguerre1 method with auxiliary boundary points (left). Time profile for the diffusion equation with Neumann boundary conditions based on Gaussian approximation without auxiliary boundary points (right). $N = 513$ for both plots.

Table 4: Diffusion equation with Neumann boundary conditions.

N	Gaussian without aux. points		Gaussian with with aux. points	
	ℓ_∞-err ($t=0$)	ℓ_∞-err ($t=1$)	ℓ_∞-err ($t=0$)	ℓ_∞-err ($t=1$)
17	0.1151	53.9230	0.1151	0.0037
33	0.0576	7.2294(6)	0.0576	0.0045
65	0.0288	7.3030(28)	0.0288	0.0081
129	0.0144	7.5726(119)	0.0144	0.0160
257	0.0072	2.0274(304)	0.0072	0.0327
513	0.0036	5.0685(303)	0.0036	0.0687

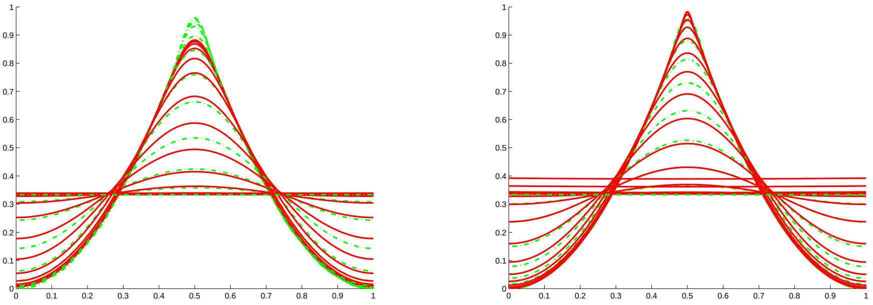

Figure 5: Time profile for the diffusion equation with Neumann boundary conditions based on Gaussian approximation with auxiliary boundary points. $N = 65$ points (left), and $N = 513$ (right).

used. This is due to the fact that the initial temperature profile is not smooth. Therefore, using the higher-order Gauss-Laguerre approximation does improve the error slightly, but not the rate of approximation. In Figures 3 and 4 we present plots of the time profile (for $t = 0$ to $t = 1$ at discrete time steps) of the approximate solution based on approximate approximations (solid curves, red) and the "exact" solution based on 20 terms of the corresponding Fourier series expansion (dash-dotted curves, green). At time $t = 1$ all solutions for the problem with Dirichlet boundary conditions basically look the same. Note that for the Neumann problem the use of auxiliary collocation points is absolutely essential (see Fig.4 right). However, even with auxiliary points, as the number of points increases the approximate solution tends to end up increasingly above the correct solution (see Fig. 5 right).

5.2 Hermite Collocation

We now consider the second solution method described in Sect. 3.2. The numerical experiments below are only for Hermite approximation (without the addition of time stepping, see Sect. 2.2). The test function for all experiments in this section is

$$f(x) = \frac{3}{4}\left[\exp\left(-\frac{(9x-2)^2}{4}\right) + \exp\left(-\frac{(9x+1)^2}{49}\right)\right] + \frac{1}{2}\exp\left(-\frac{(9x-7)^2}{4}\right) - \frac{1}{5}\exp\left(-(9x-4)^2\right).$$

5.2.1 Full Hermite Approximation

We approximate given function and derivative information sampled from the test function as described in Sect.2.2.1. The left part of Figure 6 shows a sequence of increasingly better approximations (solid

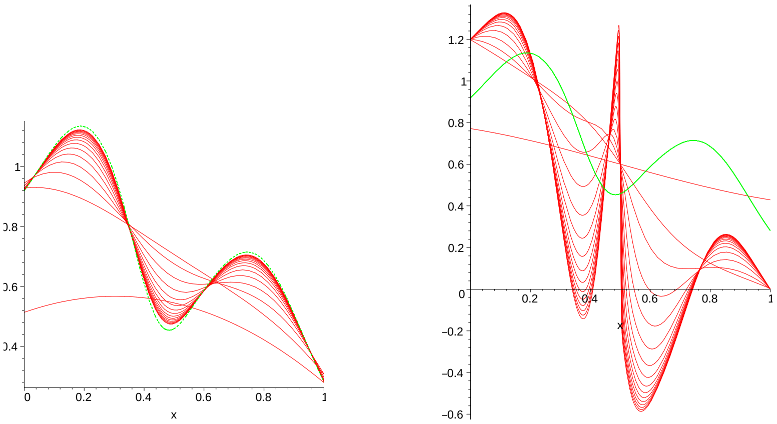

Figure 6: Typical full (left) and endpoint (right) Hermite approximations using linear precision.

curves, red) to the function (green curve, dash-dotted). The first approximation is based on data at 3 points, the last one uses 65 equally spaced points in [0,1]. Even though we show only the plots, it is clear that this method works well, but is irrelevant for the solution of boundary-value problems. Therefore, we next consider the modified problem for which function values are given only at the endpoints, and elsewhere we know only derivative information.

5.2.2 Endpoint Hermite Collocation

The endpoint Hermite problem was described in Sect. 2.2.2. This problem is equivalent to a two-point boundary value problem. As we can see from the plot in the right part of Figure 6 this method does not work. The problem with this method is that the information from the boundary can not be communicated to the interior points. Problems similar to this have also been observed in multigrid methods. As in that application, we propose to solve the endpoint Hermite problem with an iterative algorithm which alternates between a coarse grid and subsequent finer grids.

5.2.3 Nested Multilevel Hermite Collocation

In order to obtain an algorithm which yields a converging solution for the two-point boundary-value problem/endpoint Hermite approximation problem we embed the basic quasi-interpolant for the endpoint Hermite problem of the previous section in a nested multilevel algorithm. The algorithm can be described as follows:

Algorithm

```
for l=1 to outeriterations
  for k=1 to inneriterations
    for j=1 to N
      $\text{res}_j = L_j(f - \mathcal{G}f)$          % use residuals as data to be fitted
    end
    update = mlsapprox(N,supportsize,res)
    $\mathcal{G}f = \mathcal{G}f + \text{update}$
  end
end
```

Table 5: Endpoint Hermite approximation with nested multilevel algorithm. 3 outer and 5 inner iterations.

k	N	support size	$l = 1$		$l = 2$		$l = 3$	
			ℓ_∞-err	rate	ℓ_∞-err	rate	ℓ_∞-err	rate
1	5	1.00	$6.5528(-1)$		$1.0725(-1)$	.897	$8.5521(-2)$	.767
2	9	.667	$3.5408(-1)$	1.85	$1.3463(-1)$	.797	$6.1028(-2)$	1.40
3	17	.444	$2.8035(-1)$	1.26	$1.0290(-1)$	1.31	$4.4788(-2)$	1.36
4	33	.296	$1.9991(-1)$	1.40	$8.1239(-2)$	1.27	$2.7042(-2)$	1.66
5	65	.198	$9.6185(-2)$	2.08	$6.5560(-2)$	1.24	$3.4117(-2)$	.793

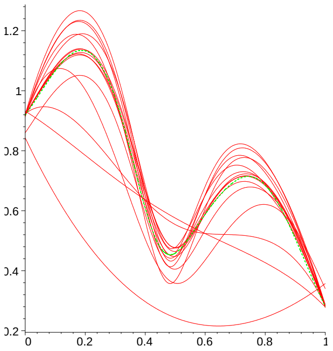

Figure 7: Typical endpoint Hermite approximations using quadratic precision and a nested multilevel algorithm.

Results for an experiment using the same test function as before, and nested grids of 5 to 65 equally spaced points are listed in Table 5 with the corresponding plots shown in Figure 7. We used 3 outer iterations, along with 5 inner iterations for the multilevel algorithm. The results are still far from ideal, but do show a marked improvement over the simple endpoint Hermite approximation results of the previous experiment. By returning to the coarsest grid with $N = 5$ points at the beginning of each outer iteration we are able to transfer some more of the information from the boundary to the interior of the domain. We scaled the support size by a factor of 1.5 at each inner iteration instead of 2 (as the mesh is). Scaling the support size is equivalent to modifying the scale parameter $\mathcal{D}$ in (1). This improves the performance of the algorithm (but contradicts the philosophy of approximate approximations where one value of $\mathcal{D}$ is supposed to cover all scales).

6 Remarks

It was the aim of this paper to compare two alternative approaches to solving time-dependent PDEs with meshfree radial MLS approximation methods. We have shown that the use of a more accurate quasi-interpolant yields better results for the transport equation. In particular, the first-order Gauss-Laguerre method is recommended instead of basic Gaussians. However, there still remains considerable work to be done. This holds especially for the treatment of boundary conditions for the method of lines approach, and for the transfer of an accurate Hermite collocation method to the PDE setting. Once these problems have been overcome an extension to nonlinear problems can be achieved via Newton iteration.

Acknowledgements

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

References

[1] R. Farwig, *Multivariate interpolation of arbitrarily spaced data by moving least squares methods*, J. Comput. Appl. Math., 16, (1986), 79–93.

[2] G. E. Fasshauer, *Solving partial differential equations by collocation with radial basis functions*, in A. Le Méhauté, C. Rabut, L. L. Schumaker, eds., *Surface Fitting and Multiresolution Methods*, Vanderbilt University Press, Nashville (1997), 131–138.

[3] G. E. Fasshauer, *Solving differential equations with radial basis functions: multilevel methods and smoothing*, Advances in Comp. Math., 11, (1999), 139–159.

[4] G. E. Fasshauer, *Matrix-free multilevel moving least-squares methods*, in C. K. Chui, L. L. Schumaker, J. Stöckler, eds., *Approximation Theory X: Wavelets, Splines and Applications*, Vanderbilt Univ. Press, Nashville (2002), 271–281.

[5] G. E. Fasshauer, *Approximate moving least-squares approximation with compactly supported weights*, in M. Griebel, et al., eds., *Lecture Notes in Computer Science and Engineering*, Springer, Berlin, to appear.

[6] G. E. Fasshauer, *High-order moving least-squares approximation via fast radial Laguerre transforms*, manuscript.

[7] Y.-C. Hon, X.-Z. Mao, *A radial basis function method for solving options pricing model*, Financial Engineering, 8, (1999), 31–49.

[8] D. Levin, *The approximation power of moving least-squares*, Math. Comp., 67, (1998), 1517–1531.

[9] S. Li, W. K. Liu, *Meshfree and particle methods and their applications*, Applied Mechanics Review, 55, (2002), 1–34.

[10] R. Lorentz, F. J. Narcowich, J. D. Ward, *Collocation discretizations of the transport equation with radial basis functions*, preprint.

[11] V. Maz'ya, G. Schmidt, *On approximate approximations using Gaussian kernels*, IMA J. Numer. Anal., 16, (1996), 13–29.

[12] V. Maz'ya, G. Schmidt, *On quasi-interpolation with non-uniformly distributed centers on domains and manifolds*, J. Approx. Theory, 110, (2001), 125–145.

[13] R. Schaback, H. Wendland, *Characterization and construction of radial basis functions*, in N. Dyn, D. Leviatan, D. Levin, and A. Pinkus, eds., *Multivariate approximation and applications*, Cambridge Univ. Press, Cambridge (2001), 1–24.

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