Chapter 9

Applications

9.1 Solving Partial Differential Equations via Collocation

In this section we discuss the numerical solution of elliptic partial differential equations using a collocation approach based on radial basis functions. To make the discussion transparent we will focus on the case of a time independent linear elliptic partial differential equation in $\mathbb{R}^2$.

9.1.1 Kansa's Approach

In [340] Kansa suggested a now very popular non-symmetric method for the solution of elliptic PDEs with radial basis functions. In order to be able to clearly point out the differences between Kansa's method and a symmetric approach proposed in [194] we recall some of the basics of scattered data interpolation with radial basis functions in $\mathbb{R}^s$.

In this context we are given data $\{\boldsymbol{x}_i, f_i\}$, $i = 1, \ldots, N$, $\boldsymbol{x}_i \in \mathbb{R}^s$, where we can think of the values f_i being sampled from a function $f : \mathbb{R}^s \to \mathbb{R}$. The goal is to find an interpolant of the form

$$\mathcal{P}f(\boldsymbol{x}) = \sum_{j=1}^{N} c_j \varphi(\|\boldsymbol{x} - \boldsymbol{x}_j\|), \qquad \boldsymbol{x} \in \mathbb{R}^s, \tag{9.1}$$

such that

$$\mathcal{P}f(\boldsymbol{x}_i) = f_i, \qquad i = 1, \ldots, N.$$

The solution of this problem leads to a linear system $A\boldsymbol{c} = \boldsymbol{f}$ with the entries of A given by

$$A_{ij} = \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \qquad i, j = 1, \ldots, N. \tag{9.2}$$

As discussed earlier, the matrix A is non-singular for a large class of radial functions including (inverse) multiquadrics, Gaussians, and the strictly positive definite compactly supported functions of Wendland, Wu, or Buhmann. In the case of strictly conditionally positive definite functions such as thin plate splines the problem needs to be augmented by polynomials.

We now switch to the collocation solution of partial differential equations. Assume we are given a domain $\Omega \subset \mathbb{R}^s$, and a linear elliptic partial differential equation of the form

$$L[u](\boldsymbol{x}) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \text{ in } \Omega, \tag{9.3}$$

with (for simplicity of description) Dirichlet boundary conditions

$$u(\boldsymbol{x}) = g(\boldsymbol{x}), \qquad \boldsymbol{x} \text{ on } \partial\Omega. \tag{9.4}$$

For Kansa's collocation method we then choose to represent u by a radial basis function expansion analogous to that used for scattered data interpolation, i.e.,

$$u(\boldsymbol{x}) = \sum_{j=1}^{N} c_j \varphi(\|\boldsymbol{x} - \boldsymbol{\xi}_j\|), \tag{9.5}$$

where we now introduce the points $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N$ as centers for the radial basis functions. They will usually be selected to coincide with the collocation points $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \Omega$. However, the discussion below is clearer if we formally distinguish between centers $\boldsymbol{\xi}_j$ and collocation points $\boldsymbol{x}_i$. We assume the simplest possible setting here, i.e., no polynomial terms are added to the expansion (9.5). The collocation matrix which arises when matching the differential equation (9.3) and the boundary conditions (9.4) at the collocation points $\mathcal{X}$ will be of the form

$$A = \begin{bmatrix} \Phi \\ L[\Phi] \end{bmatrix}, \tag{9.6}$$

where the two blocks are generated as follows:

$$\begin{aligned} \Phi_{ij} &= \varphi(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|), \qquad \boldsymbol{x}_i \in B,\ \boldsymbol{\xi}_j \in \mathcal{X}, \\ L[\Phi]_{ij} &= L[\varphi](\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|), \qquad \boldsymbol{x}_i \in I,\ \boldsymbol{\xi}_j \in \mathcal{X}. \end{aligned}$$

Here we have identified (as we will do throughout this section) the set of centers with the set of collocation points. The set $\mathcal{X}$ is split into a set I of interior points, and B of boundary points. The problem is well-posed if the linear system $A\boldsymbol{c} = \boldsymbol{y}$, with $\boldsymbol{y}$ a vector consisting of entries $g(\boldsymbol{x}_i)$, $\boldsymbol{x}_i \in B$, followed by $f(\boldsymbol{x}_i)$, $\boldsymbol{x}_i \in I$, has a unique solution.

We note that a change in the boundary conditions (9.4) is as simple as changing a few rows in the matrix A in (9.6) as well as on the right-hand side $\boldsymbol{y}$. We also point out that Kansa only proposed to use multiquadrics in (9.5), and for that method suggested the use of varying parameters α_j, $j = 1, \ldots, N$, which improves the accuracy of the method when compared to using only one constant value of α (see [340]).

A problem with Kansa's method is that – for a constant multiquadric shape parameter α – the matrix A may for certain configurations of the centers $\boldsymbol{\xi}_j$ be singular. Originally, Kansa assumed that the non-singularity results for interpolation matrices would carry over to the PDE case. However, as the numerical experiments of Hon and Schaback [304] show, this is not so. This is to be expected since the matrix for the collocation problem is composed of rows which are built from *different* functions (which – depending on the differential operator L – might not even be radial). The results for

the non-singularity of interpolation matrices, however, are based on the fact that A is generated by a *single* function φ.

An indication of the success of Kansa's method (which has not yet been shown to be well-posed) are the early papers [165, 166, 262, 341, 467] and many more since. In his paper [340] Kansa describes three sets of experiments using his method and comments on the superior performance of multiquadrics in terms of computational complexity and accuracy when compared to finite difference methods. Therefore, it remains an interesting open question whether the well-posedness of Kansa's method can be established at least for certain configurations of centers. Moreover, Kansa's suggestion to use variable shape parameters α_j in order to improve accuracy and stability of the problem has very little theoretical support. Except for one paper by Bozzini, Lenarduzzi and Schaback [68] (which addresses only the interpolation setting) this problem has not been addressed in the literature.

Before we describe an alternate approach which does ensure well-posedness of the resulting collocation matrix and which is based on basis functions suitable for scattered Hermite interpolation we would like to point out that in [467] the authors suggest how Kansa's method can be applied to other types of partial differential equation problems such as non-linear elliptic PDEs, systems of elliptic PDEs, and time-dependent parabolic or hyperbolic PDEs.

9.1.2 An Hermite-based Approach

The following symmetric approach is based on scattered Hermite interpolation (see, e.g., [315, 484, 598, 651]), which we now also quickly review. In this context we are given data $\{\boldsymbol{x}_i, L_i f\}$, $i = 1, \ldots, N$, $\boldsymbol{x}_i \in \mathbb{R}^s$ where $\mathcal{L} = \{L_1, \ldots, L_N\}$ is a linearly independent set of continuous linear functionals. We try to find an interpolant of the form

$$\mathcal{P}f(\boldsymbol{x}) = \sum_{j=1}^{N} c_j L_j^{\boldsymbol{\xi}} \varphi(\|\boldsymbol{x} - \boldsymbol{\xi}\|), \qquad \boldsymbol{x} \in \mathbb{R}^s, \tag{9.7}$$

satisfying

$$L_i \mathcal{P}f = L_i f, \qquad i = 1, \ldots, N.$$

We have used $L^{\boldsymbol{\xi}}$ to indicate that the functional L acts on φ viewed as a function of the second argument $\boldsymbol{\xi}$. The linear system $A\boldsymbol{c} = Lf$ which arises in this case has matrix entries

$$A_{ij} = L_i L_j^{\boldsymbol{\xi}} \varphi, \qquad i, j = 1, \ldots, N. \tag{9.8}$$

In the references mentioned at the beginning of this subsection it is shown that A is non-singular for the same classes of φ as given for scattered data interpolation in our earlier chapters.

Remark: It should be pointed out that this formulation of Hermite interpolation is very general and goes considerably beyond the standard notion of Hermite interpolation (which refers to interpolation of successive derivative values). Here any kind of linear functional are allowed as long as the set $\mathcal{L}$ is linearly independent.

We illustrate this approach with a simple example using derivative functionals.

Example: Let data $\{\boldsymbol{x}_i, f(\boldsymbol{x}_i)\}_{i=1}^n$ and $\{\boldsymbol{x}_i, \frac{\partial f}{\partial x}(\boldsymbol{x}_i)\}_{i=n+1}^N$ with $\boldsymbol{x} = (x, y) \in \mathbb{R}^2$ be given. Then

$$\mathcal{P}f(\boldsymbol{x}) = \sum_{j=1}^{n} c_j \varphi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) - \sum_{j=n+1}^{N} c_j \frac{\partial \varphi}{\partial x}(\|\boldsymbol{x} - \boldsymbol{x}_j\|),$$

and

$$A = \begin{bmatrix} \Phi & -\Phi_x \\ \Phi_x & -\Phi_{xx} \end{bmatrix},$$

with

$$\begin{aligned}
\Phi_{ij} &= \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \qquad i, j = 1, \ldots, n, \\
-\Phi_{x,ij} &= -\frac{\partial \varphi}{\partial x}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \qquad i = 1, \ldots, n,\ j = n+1, \ldots, N, \\
\Phi_{x,ij} &= \frac{\partial \varphi}{\partial x}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \qquad i = n+1, \ldots, N,\ j = 1, \ldots, n, \\
\Phi_{xx,ij} &= \frac{\partial^2 \varphi}{\partial x^2}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \qquad i, j = n+1, \ldots, N.
\end{aligned}$$

Now we describe an alternative collocation method based on the generalized interpolation theory just reviewed. Assume we are given the same PDE (9.3) with boundary conditions (9.4) as in the section on Kansa's method. In order to be able to apply the results from scattered Hermite interpolation to ensure the non-singularity of the collocation matrix we propose the following expansion for the unknown function u:

$$u(\boldsymbol{x}) = \sum_{j=1}^{\#B} c_j \varphi(\|\boldsymbol{x} - \boldsymbol{\xi}_j\|) + \sum_{j=\#B+1}^{N} c_j L^{\boldsymbol{\xi}}[\varphi](\|\boldsymbol{x} - \boldsymbol{\xi}_j\|), \tag{9.9}$$

where $\#B$ denotes the number of nodes on the boundary of Ω, and $L^{\boldsymbol{\xi}}$ is the differential operator used in (9.3), but acting on φ viewed as a function of the second argument, i.e., $L[\varphi]$ is equal to $L^{\boldsymbol{\xi}}[\varphi]$ up to a possible difference in sign. Note the difference in notation. In (9.7) L is a linear functional, and in (9.9) a differential operator.

This expansion for u leads to a collocation matrix A which is of the form

$$A = \begin{bmatrix} \Phi & L^{\boldsymbol{\xi}}[\Phi] \\ L[\Phi] & L[L^{\boldsymbol{\xi}}[\Phi]] \end{bmatrix}, \tag{9.10}$$

where the four blocks are generated as follows:

$$\begin{aligned}
\Phi_{ij} &= \varphi(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|), \qquad \boldsymbol{x}_i, \boldsymbol{\xi}_j \in B, \\
L^{\boldsymbol{\xi}}[\Phi]_{ij} &= L^{\boldsymbol{\xi}}[\varphi](\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|), \qquad \boldsymbol{x}_i, \in B,\ \boldsymbol{\xi}_j \in I, \\
L[\Phi]_{ij} &= L[\varphi](\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|), \qquad \boldsymbol{x}_i \in I,\ \boldsymbol{\xi}_j \in B, \\
L[L^{\boldsymbol{\xi}}[\Phi]]_{ij} &= L[L^{\boldsymbol{\xi}}[\varphi]](\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|), \qquad \boldsymbol{x}_i, \boldsymbol{\xi}_j \in I.
\end{aligned}$$

The matrix (9.10) is of the same type as the scattered Hermite interpolation matrices (9.8), and therefore non-singular as long as φ is chosen appropriately. Thus, viewed using the new expansion (9.9) for u, the collocation approach is certainly well-posed.

	α	ρ_K	ρ_H	$\text{cond}_K(A)$	$\text{cond}_H(A)$
5×3	1.0	5.248447e-02	2.004420e-01	2.599606e+03	1.627432e+03
8×4	1.0	1.126843e-02	1.124710e-02	2.325758e+05	8.167527e+04
10×6	1.0	5.809472e-03	6.481697e-03	4.321740e+07	1.808001e+07
16×8	1.0	1.347863e-03	1.720007e-03	8.685785e+10	1.496772e+10
20×12	1.0	5.053090e-04	5.973294e-04	5.161540e+15	1.234633e+15

Table 9.1: Error progression for increasingly denser data sets (Ex.1, fixed α).

Another point in favor of the Hermite based approach is that the matrix (9.10) is (anti)-symmetric as opposed to the completely unstructured matrix (9.6) of the same size. This property should be of value when trying to devise an efficient implementation of the collocation method. Also note that although A consists of four blocks now, it still is of the same size, namely $N \times N$, as the collocation matrix (9.6) obtained for Kansa's approach.

Remark: One attempt to obtain an efficient implementation of the Hermite based collocation method is a version of the greedy algorithm described in Section 8.5.1 by Hon, Schaback and Zhou [305].

9.1.3 Numerical Examples

The following test examples are taken from [194]. We restrict ourselves to two-dimensional Poisson problems whose analytic solution is readily available and therefore can easily be verified. We will refer to a point in $\mathbb{R}^2$ as (x, y). In all of the following tests we used multiquadrics in the expansions (9.5) and (9.9) of the unknown function u.

Example 1: Consider the Poisson equation

$$\Delta u(x, y) = y(1 - y) \sin^3 x, \qquad x \in (0, \pi),\ y \in (0, 1),$$

with Dirichlet boundary conditions

$$u(x, 0) = u(x, 1) = u(0, y) = u(\pi, y) = 0.$$

For this test problem we selected various uniform grids as listed in Tables 9.1 and 9.2 on $[0, \pi] \times [0, 1]$. Tables 9.1 and 9.2 show the values of the multiquadric parameter α, the relative maximum errors ρ computed on a fine grid of 60×60 points, and the approximate condition numbers of A. The range of u on the evaluation grid is approximately $[-0.021023, 0.0]$. The "optimal" value for α was determined by trial and error. The subscripts K and H refer to Kansa's and the Hermite based method, respectively.

Figure 9.1 shows the distribution of the errors $|u(x) - s(x)|$ on the evaluation grid for the two methods on the 8×4 grid used in Table 9.2. The scale used for the shading is displayed on the right.

Example 2: Consider the Poisson equation

$$\Delta u(x, y) = \sin x - \sin^3 x, \qquad x \in (0, \frac{\pi}{2}),\ y \in (0, 2),$$

	α_K	α_H	ρ_K	ρ_H	$\text{cond}_K(A)$	$\text{cond}_H(A)$
5×3	1.18	1.39	1.627193e-02	4.180428e-02	5.592238e+03	5.231279e+03
8×4	1.04	1.11	1.103747e-02	1.062891e-02	3.175078e+05	1.735482e+05
10×6	4.80	3.84	2.739293e-03	3.451799e-03	1.193586e+18	1.414927e+15
16×8	3.12	3.12	2.707006e-04	2.082886e-04	1.209487e+19	6.609375e+18
20×12	2.00	2.30	3.894511e-05	1.273363e-05	3.739554e+19	6.750955e+18

Table 9.2: Error progression for increasingly denser data sets (Ex.1, "optimal" α).

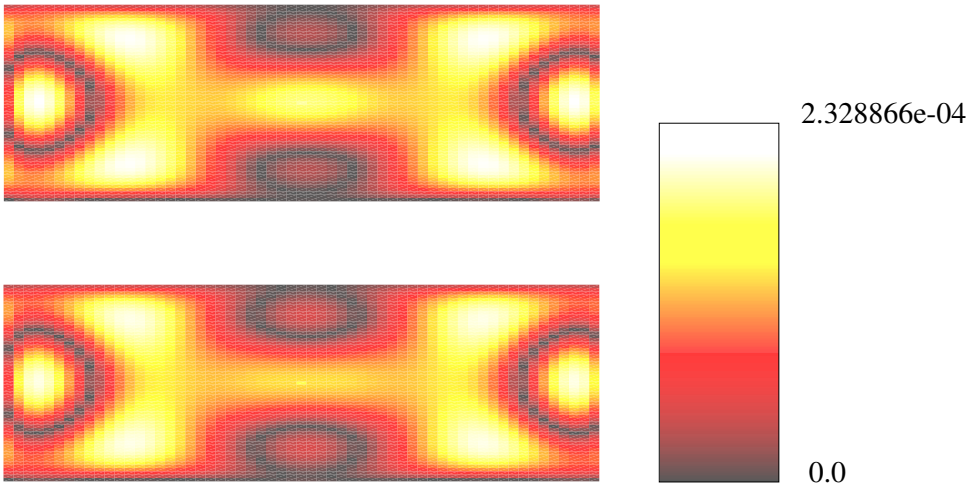

Figure 9.1: Error for Kansa's (top), Hermite (bottom) solution for Ex. 1 on 8×4 grid.

with mixed Dirichlet and Neumann boundary conditions

$$u(0, y) = u_x(\frac{\pi}{2}, y) = u_y(x, 0) = u_y(x, 2) = 0.$$

For this example we selected uniform grids on $[0, \pi/2] \times [0, 2]$ as listed in Table 9.3. This time we only list the results for the "optimal" choice of α. The values listed are analogous to those in Ex. 1.

All in all the Hermite method seems to perform slightly better than Kansa's method. Especially for the cases in which we used relatively many interior points (which is where the methods differ). Also, the matrices for the Hermite method generally have smaller condition numbers. An advantage of the Hermite approach over Kansa's method is

	α_K	α_H	ρ_K	ρ_H	$\text{cond}_K(A)$	$\text{cond}_H(A)$
3×3	109.0	2.19	9.628085e-01	1.141043e-01	1.592286e+16	5.560886e+02
5×5	1.80	1.73	2.181029e-02	4.327029e-02	2.395293e+06	1.271196e+05
7×7	1.58	3.56	6.910084e-03	1.871798e-04	5.762316e+08	1.854850e+12
10×10	2.80	3.29	9.265197e-05	5.126676e-05	2.842111e+18	7.070804e+17
14×14	2.28	2.62	1.138751e-05	1.725526e-06	6.573143e+19	5.891454e+18
20×20	1.53	1.91	5.501057e-06	6.217559e-07	5.889491e+19	7.576112e+19

Table 9.3: Error progression for increasingly denser data sets (Ex.2, "optimal" α).

that for the differential operator L used here, the collocation matrices resulting from the Hermite approach are symmetric. Therefore the amount of computation can be reduced considerably, which is important for larger problems. Kansa's method has the advantage of being simpler to implement (since less derivatives of the basis functions are required).

Remarks:

1. Both of the methods described in this section have been implemented for many different applications. A thorough comparison of the two methods was reported in [520].

2. Since the methods described above were both originally used with globally supported basis functions, the same concerns as for interpolation problems about stability and numerical efficiency apply. Two recent papers by Ling and Kansa [395, 396] address these issues. In particular, they develop a preconditioner in the spirit of the one described in Section 8.3.3, and describe their experience with a domain decomposition algorithm.

3. A convergence analysis for the symmetric method was established by Franke and Schaback [229, 230]. The error estimates established in [229, 230] require the solution of the PDE to be very smooth. Therefore, one should be able to use meshfree radial basis function collocation techniques especially well for (high-dimensional) PDE problems with smooth solutions on possibly irregular domains. Due to the known counterexamples [304] for the non-symmetric method, a convergence analysis is still lacking for that method.

4. Recently, Miranda [462] has shown that Kansa's method will be well-posed if it is combined with so-called *R-functions*. This idea was also used by Höllig and his co-workers in their development of WEB-splines (see, e.g., [299]).

5. Kansa's method has the advantage of being easily adapted for nonlinear elliptic PDEs (see, e.g., [201, 467]).

Some numerical evidence for convergence rates of the symmetric collocation method is given by the examples above, and in the papers [336, 520]. The example above shows very high convergence rates (as predicted by the estimate in [230]) when using multiquadrics on a problem which has a smooth solution. In [336] thin plate splines as well as Wendland's C^4 compactly supported RBF $\varphi_{3,2}$ were tested. The results for thin plate splines are in good agreement with the theory. However, the numerical experiments using the Wendland function show $\mathcal{O}(h^3)$ convergence instead of $\mathcal{O}(h)$ as predicted by the lower bounds of [230] combined with the error bound for Wendland functions. This could suggest that a sharper error estimate may be possible when using compactly supported RBFs.

Other recent papers investigating various aspects of radial basis function collocation are, e.g., [135] by Cheng, Golberg, Kansa and Zammito, [215] by Fedoseyev, Friedman and Kansa, [345] by Kansa and Hon, [360] by Larsson and Fornberg, [365] by Leitão, and [424] by Mai-Duy and Tran-Cong.

For example, in the paper [215] it is suggested that the collocation points on the boundary are also used to satisfy the PDE. However, this adds a set of extra equations

to the problem, and therefore one should also use some additional basis functions in the expansion (9.5). It is suggested in [215] that these centers lie outside the domain Ω. The motivation for this modification is the well-known fact that both for interpolation and collocation with radial basis functions the error is largest near the boundary. In various numerical experiments this strategy is shown to improve the accuracy of Kansa's basic non-symmetric method. It should be noted that there is once more no theoretical foundation for this method.

Larsson and Fornberg [360] compare Kansa's basic collocation method, the modification just described, and the Hermite-based symmetric approach mentioned earlier. Using multiquadric basis functions in a standard implementation they conclude that the symmetric method is the most accurate, followed by the non-symmetric method with boundary collocation. The reason for this is the better conditioning of the system for the symmetric method. Larsson and Fornberg also discuss an implementation of the three methods using the complex Contour-Padé integration method mentioned in Section 8.1. With this technique stability problems are overcome, and it turns out that both the symmetric and the non-symmetric method perform with comparable accuracy. Boundary collocation of the PDE yields an improvement only if these conditions are used as additional equations, i.e., by increasing the problem size. It should also be noted that often the most accurate results were achieved with values of the multiquadric shape parameter α which would lead to severe ill-conditioning using a standard implementation, and therefore these results could be achieved only using the complex integration method. Moreover, in [360] radial basis function collocation is deemed to be far superior in accuracy than standard second-order finite differences or a standard Fourier-Chebyshev pseudospectral method.

Leitão [365] applies the symmetric collocation method to a fourth-order Kirchhoff plate bending problem, and emphasizes the simplicity of the implementation of the radial basis function collocation method. And, finally, Mai-Duy and Tran-Cong [424] suggest a collocation method for which the basis functions are taken to be anti-derivatives of the usual radial basis functions.

All of the experiments just mentioned were conducted without using a multilevel approach. In particular, in order to achieve convergence with the Wendland functions the support had to be chosen so large that only problems with a very modest number of centers could be handled (see [336]). So, as for scattered data interpolation, a multilevel approach is needed to obtain computational efficiency.

We would like to end the discussion of the collocation approach by looking at a multilevel implementation with compactly supported functions.

The most significant difference between the use of compactly supported RBFs for scattered data interpolation and for the numerical solution of PDEs by collocation appears when we turn to the multilevel approach. Recall that the use of the multilevel method is motivated by our desire to obtain a convergent scheme while at the same time keeping the bandwidth fixed, and thus the computational complexity at $\mathcal{O}(N)$.

Here is an adaptation of the basic multilevel algorithm of Section 8.2 to the case of a collocation solution of the problem $Lu = f$:

mesh	ℓ_2-error	rate
5	3.637579e-04	
9	1.892007e-05	4.26
17	3.055339e-06	2.63
33	2.111403e-06	0.53
65	2.062621e-06	0.03
129	2.066411e-06	0.00
257	2.070168e-06	0.00
513	2.072171e-06	0.00
1025	2.073182e-06	0.00
2049	2.073688e-06	0.00

Table 9.4: Multilevel collocation algorithm for symmetric collocation with constant bandwidth.

Algorithm (Multilevel Collocation)

$u_0 = 0$.

For k from 1 to K do

Find $u_k \in \mathcal{S}_{\mathcal{X}_k}$ such that $Lu_k = (f - Lu_{k-1})$ on grid $\mathcal{X}_k$.

Update $u_k \leftarrow u_{k-1} + u_k$.

end

Here $\mathcal{S}_{\mathcal{X}_k}$ is the space of functions used for expansion (9.5) or (9.9) on grid $\mathcal{X}_k$. Whereas we noted above that there is strong numerical (and limited theoretical) evidence that the basic multilevel interpolation algorithm converges (at least linearly), the following example shows that *we cannot in general expect the multilevel collocation algorithm to converge at all.*

Example: Consider the boundary-value problem

$$\begin{aligned} -u''(x) + \pi^2 u(x) &= 2\pi^2 \sin \pi x, \qquad x \in (0,1), \\ u(0) &= u(1) = 0, \end{aligned}$$

with solution $u(x) = \sin \pi x$. As computational grids $\mathcal{X}_k$ we take $2^{k+1} + 1$ uniformly spaced points on $[0, 1]$ as indicated in Table 9.4. We use the C^6 compactly supported Wendland function $\varphi_{3,3}$ and the conjugate gradient method with Jacobi preconditioning is used to solve the resulting linear systems. We take the support size on the first grid to be so large that the resulting matrix is a dense matrix. During subsequent iterations the support size is halved (as is the meshsize) in order to maintain a constant bandwidth of 17 (i.e., work in the stationary setting). Even though the first three iterations seem to indicate significant rates of convergence, the convergence behavior quickly changes, and by the fifth iteration there is virtually no improvement of the error (the fact that the errors actually increase is due to the fact that they are computed on increasingly finer grids).

We note that the same behavior can be observed if the non-symmetric approach is used instead. However, then the convergence ceases at a slightly later stage. We also note that the same phenomenon was observed by Wendland in the context of a multilevel Galerkin algorithm for compactly supported RBFs (see [631] as well as our discussion in the next section).

Remarks:

1. It has been suggested that the convergence behavior of the multilevel collocation algorithm may be linked to the phenomenon of approximate approximation. However, so far no connection has been established.

2. As was shown in [198] a possible remedy for the non-convergence problem is *smoothing*. One might also expect that a slightly different scaling of the support sizes of the basis functions (such that the bandwidth of the matrix is allowed to increase slowly from one iteration to the next, i.e., moving to the non-stationary setting) will lead to better results. In [198] it was shown that this is in fact true. However, smoothing further improved the convergence. A discussion of the idea of post-conditioning via smoothing is beyond the scope of this text. We refer the reader to the paper [209].

9.2 Galerkin Methods

A variational approach to the solution of PDEs with RBFs has so far only been considered by Wendland [630, 631]. In [631] he studies the Helmholtz equation with natural boundary conditions, i.e.,

$$\begin{aligned} -\Delta u + u &= f \qquad \text{in } \Omega, \\ \frac{\partial}{\partial \nu} u &= 0 \qquad \text{on } \partial\Omega, \end{aligned}$$

where ν denotes the outer unit normal vector. The classical Galerkin formulation then leads to the problem of finding a function $u \in H^1(\Omega)$ such that

$$a(u, v) = (f, v)_{L_2(\Omega)} \qquad \text{for all } v \in H^1(\Omega),$$

where $(f, v)_{L_2(\Omega)}$ is the usual L_2 inner product, and for the Helmholtz equation the bilinear form a is given by

$$a(u, v) = \int_\Omega (\nabla u \cdot \nabla v + uv) dx.$$

In order to obtain a numerical scheme the infinite-dimensional space $H^1(\Omega)$ is replaced by some finite-dimensional subspace $\mathcal{S}_\mathcal{X} \subseteq H^1(\Omega)$, where $\mathcal{X}$ is some computational grid to be used for the solution. In the context of RBFs $\mathcal{S}_\mathcal{X}$ is taken as

$$\mathcal{S}_\mathcal{X} = \text{span}\{\phi(\| \cdot - x_j \|_2),\ x_j \in \mathcal{X}\}.$$

This results in a square system of linear equations for the coefficients of $u_{\mathcal{X}} \in \mathcal{S}_{\mathcal{X}}$ determined by

$$a(u_{\mathcal{X}}, v) = (f, v)_{L_2(\Omega)} \qquad \text{for all } v \in \mathcal{S}_{\mathcal{X}}.$$

For more on the Galerkin method (in the context of finite elements) see, e.g., [69, 70]. It was shown in [630] that for those RBFs (globally as well as locally supported) whose Fourier transform decays like $(1 + \| \cdot \|_2)^{-2\beta}$ the following convergence estimate holds:

$$\|u - u_{\mathcal{X}}\|_{H^1(\Omega)} \leq C h^{\sigma-1} \|u\|_{H^\sigma(\Omega)}, \tag{9.11}$$

where h is the meshsize of $\mathcal{X}$, the solution satisfies the regularity requirements $u \in H^\sigma(\Omega)$, and where the convergence rate is determined by $\beta \geq \sigma > s/2 + 1$. For Wendland's compactly supported RBFs this implies that functions which are in $C^{2\kappa}$ and strictly positive definite on $\mathbb{R}^s$ satisfying $\kappa \geq \sigma - \frac{s+1}{2}$ will have $\mathcal{O}(h^{\kappa+(s-1)/2})$ convergence order, i.e., the C^0 function $\varphi_{3,0} = (1-r)_+^2$ yields $\mathcal{O}(h)$ and the C^2 function $\varphi_{3,1} = (1-r)_+^4(4r+1)$ delivers $\mathcal{O}(h^2)$ convergence in $\mathbb{R}^3$. As with the convergence estimate for symmetric collocation there is a link between the regularity requirements on the solution and the space dimension s. Also, so far, the theory is only established for PDEs with natural boundary conditions.

The convergence estimate (9.11) holds for the non-stationary setting, i.e., if we are using compactly supported basis functions, for fixed support radii. By the same argumentation as used in Section 8, one will want to switch to the stationary setting and employ a multilevel algorithm in which the solution at each step is updated by a fit to the most recent residual. This should ensure both convergence and numerical efficiency.

Here is the variant of the stationary multilevel collocation algorithm listed above for the weak formulation (see [631]):

Algorithm (Multilevel Galerkin)

$u_0 = 0$.

For k from 1 to K do

Find $u_k \in \mathcal{S}_{\mathcal{X}_k}$ such that $a(u_k, v) = (f, v) - a(u_{k-1}, v)$ for all $v \in \mathcal{S}_{\mathcal{X}_k}$.

Update $u_k \leftarrow u_{k-1} + u_k$.

end

This algorithm *does not converge in general* (see Tab. 1 in [631]).

Since the weak formulation can be interpreted as a Hilbert space projection method, Wendland was able to show that a modified version of the multilevel Galerkin algorithm, namely

Algorithm (Nested Multilevel Galerkin)

Fix K and $M \in \mathbb{N}$, and set $v_0 = 0$.

For j from 0 while *resiudal* > *tolerance* to M do

Set $u_0 = v_j$.

Apply the k-loop of the previous algorithm and denote the result with $\hat{u}(v_j)$.

Set $v_{j+1} = \hat{u}(v_j)$.

end

does converge. In fact, using this algorithm Wendland proves, and also observes numerically, convergence which is at least linear (see Theorem 3 and Tab. 2 in [631]). The important difference between the two multilevel Galerkin algorithms is the added outer iteration in the nested version which is a well-known idea from linear algebra introduced in 1937 by Kaczmarz [337]. A proof of the linear convergence for general Hilbert space projection methods coupled with Kaczmarz iteration can be found in [585]. This alternate projection idea is also the fundamental ingredient in the convergence proof of the domain decomposition method of Beatson, Light and Billings [42] described in the previous chapter. We mention here that in the multigrid literature Kaczmarz' method is frequently used as a smoother (see e.g. [435]).

Remarks:

1. Aside from difficulties with Dirichlet (or sometimes called *essential*) boundary conditions, Wendland reports that the numerical evaluation of the weak-form integrals presents a major problem for the radial basis function Galerkin approach. Both of these difficulties are also well-known in many other flavors of meshfree weak-form methods. An especially promising solution to the issue of Dirichlet boundary conditions seems to be the use of R-functions as proposed by Höllig and Reif in the context of WEB-splines (see, e.g., [299] or our earlier discussion in the context of collocation methods).

2. In a recent paper by Schaback [559] the author presents a framework for the radial basis function solution of problems both in the strong (collocation) and weak (Galerkin) form.

Many other meshfree methods for the solution of partial differential equations in the weak form appear in the (mostly engineering) literature. These methods come under such names as smoothed particle hydrodynamics (SPH) (e.g., [463]), reproducing kernel particle method (RKPM) (see, e.g., [380, 399]), point interpolation method (PIM) (see, [397]), element free Galerkin method (EFG) (see, e.g., [49]), meshless local Petrov-Galerkin method (MLPG) [14], h-p-cloud method [164], partition of unity finite element method (PUFEM) [16, 443], or generalized finite element method (GFEM) [15]. Most of these methods are based on the moving least squares approximation method discussed in Chapter 7.

There are two recent books by Atluri [12] and Liu [397] summarizing many of these methods. However, these books focus mostly on a survey of the various methods and related computational and implementation issues with little emphasis on the mathematical foundation of these methods. The recent survey paper [15] by Babuška, Banerjee and Osborn, fills a large part of this void.