J Eng Math (2007) 62:149–160
DOI 10.1007/s10665-007-9197-8

Adaptive cross-approximation for surface reconstruction using radial basis functions

Richards Grzhibovskis · Markus Bambach · Sergej Rjasanow · Gerhard Hirt

Received: 26 October 2006 / Accepted: 10 September 2007 / Published online: 2 October 2007

Abstract The efficient handling of matrices arising in surface interpolation and approximation with radial basis functions (RBF) is considered. To find a data-sparse approximation of the system matrix, the adaptive cross-approximation (ACA) technique is used. The approximation of the matrix requires $O\left(N \log^2 N\right)$ units of storage and arithmetic operations, where N is the number of interpolation points. Because basis functions are not explicitly used, the implementation is applicable to a wide class of interpolation kernels. Numerical examples involving generated data and measurements of formed sheet-metal parts are presented.

Keywords Blockwise low-rank approximation · Digitized surface · Meshless interpolation · Metal-sheet thickness

1 Introduction

Fitting a smooth function through a set of N scattered, pairwise distinct data points arises in a variety of applications. Radial basis functions (RBFs) have become increasingly popular for the construction of such functions. Among the key advantages of RBFs are high accuracy, differentiability of the interpolant, and absence of an underlying mesh. Applications of RBF interpolation include bathymetry (ocean-depth measurement), topography (altitude measurements), hydrology (rainfall interpolation), surveying, mapping, geophysics, and geology (see the survey of applications [1]), image warping [2–4], and medical imaging [5]. Experience in a variety of applications has shown that RBF interpolation performs very well for small problems. However, the straightforward application of the method requires $O\left(N^2\right)$ storage units and at least as many floating-point operations. It is computationally prohibitively expensive to use the RBF interpolation when $N > 10{,}000$, even on modern computers.

Recent approximation methods based on data trees [6,7], various forms of the fast multipole method [8–10] or panel clustering [11] can be applied to relax the requirements on storage and floating-point operations to $O\left(N \log^\beta N\right)$, where β depends on the chosen approach. Fast multipole methods were adopted to the RBF interpolation problems in [12,13]. The convolution structure of the method was exploited in [14], where a fast RBF evaluation procedure based on nonequispaced fast Fourier transform was proposed.

R. Grzhibovskis (✉) · S. Rjasanow
Applied Mathematics, University of Saarland, Saarbrücken, Germany
e-mail: richards@num.uni-sb.de

M. Bambach · G. Hirt
Institut für Bildsame Formgebung, RWTH Aachen, Aachen, Germany

Our approach to reducing the storage and complexity employs the adaptive cross-approximation (ACA) algorithm [15–18]. The resulting procedure requires $O\left(N \log^2 N\right)$ storage units and floating-point operations. The method is applicable to all common basis functions, such as multiquadratics, thin-plate splines, Gaussians etc. The algebraic nature of the ACA makes it possible to use the same implementation for interpolation as well as for evaluation of the interpolant and its derivatives regardless of the type of basis function. This property gives the method an advantage over the above-mentioned approaches: changing to a new basis function requires only coding a one- or two-line function. In contrast, the multipole-like procedures are designed to work with a particular class of basis functions, and a lot of analysis and re-coding is required in order to use a different one.

The RBF interpolation procedure is described in Sect. 2. Section 3 is devoted to the modules necessary to construct a blockwise low-rank approximation to the interpolation matrix, including clustering (reordering) of data points and the ACA algorithm. Results of numerical experiments with synthetic data sets and measurements of sheet-metal forming are reported in Sect. 4. Concluding remarks are found in Sect. 5.

2 RBF interpolation

We consider fitting a smooth function $s\colon \mathbb{R}^n \mapsto \mathbb{R}$ through a set of scattered, pairwise distinct data $(x_i, f_i)_{i=1}^N$, $x_i \in \mathbb{R}^n$, and introduce an interpolation function in the form

$$s(x) = \sum_{i=1}^{N} a_i \phi\left(\|x - x_i\|\right) + P_m(x), \tag{1}$$

where a_i are unknown coefficients, P_m is an mth-degree polynomial, and ϕ is a basis function. In addition to the interpolation condition $s(x_k) = f_k$ at each data point, we require

$$\sum_{k=1}^{N} a_k q_j(x_k) = 0 \tag{2}$$

for each basis function $q_j(x)$, $j = 1, \ldots, l(m)$, in the $l(m)$-dimensional space of mth-degree polynomials in $\mathbb{R}^n$. This yields a system of linear algebraic equations,

$$\begin{pmatrix} \Phi & Q \\ Q^T & 0 \end{pmatrix} \begin{pmatrix} a \\ c \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{3}$$

where $\Phi_{ij} = \phi\left(\|x_j - x_i\|\right)$, f is the vector of data values f_i, a is the vector containing the unknown coefficients a_i, c is the vector of coefficients of P_m with respect to the basis $(q_k)_{k=1}^{l(m)}$, and $Q_{ij} = q_j(x_i)$. The system has a unique solution if ϕ is strictly conditionally positive-definite of order $m+1$ and the set of data points is unisolvent for the space of polynomials of degree m [19]. If we restrict ourselves to data in $\mathbb{R}^2$, i.e., $x_i = \left(x_{i,1}, x_{i,2}\right)$, then adopting the thin-plate splines (TPS) interpolant

$$\phi(r) = r^2 \log(r) \tag{4}$$

and a first-degree polynomial,

$$P_1(x) = c_1 + (c_2, c_3) \cdot x \tag{5}$$

guarantees the invertibility of system (3). Other examples of strictly conditionally positive-definite functions of order two are the multiquadratics (MQS)

$$\phi(r) = \left(r^2 + \alpha^2\right)^{3/2}, \quad \alpha \in \mathbb{R}, \tag{6}$$

and the power function (POW)

$$\phi(r) = r^3. \tag{7}$$

The TPS is a fundamental solution of the biharmonic equation in $\mathbb{R}^2$. An interpolant (1) with TPS as basis function minimizes the so-called "bending energy" $\mathcal{E}(f) = \langle f, f \rangle$ (see [20]), where

$$\langle f, g \rangle = \int_{\mathbb{R}^2} (\partial_{11} g \; \partial_{11} f + 2 \; \partial_{12} g \; \partial_{12} f + \partial_{22} g \; \partial_{22} f) \, \mathrm{d}x \mathrm{d}y. \tag{8}$$

Since the interpolation method discussed in this paper will be applied to sheet-metal forming, we favor the TPS basis function. The properties of the TPS allow us to easily construct an approximating function with an even smaller value of $\mathcal{E}$. We look for coefficients a_i and c_i such that

$$\mathcal{F} = \frac{1}{N} \sum_{i=1}^{N} (s(x_i) - f_i)^2 + \lambda \langle s, s \rangle \tag{9}$$

attains a minimum, where λ is a positive smoothing parameter. Differentiating $\mathcal{F}$ with respect to a_i and c_i, we obtain a system similar to (3),

$$\begin{pmatrix} \Phi + \lambda I & Q \\ Q^T & 0 \end{pmatrix} \begin{pmatrix} a \\ c \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}. \tag{10}$$

System (3) arises as a special case for $\lambda = 0$.

For large N, we encounter two difficulties:

1. Because the matrix Φ is fully populated, $O\left(N^2\right)$ storage units are required. Even if the solution is found through an iterative method, at least $O\left(N^2\right)$ floating-point operations would be required. These order estimates render the procedure prohibitively expensive when N is of order 10^4.
2. The system matrix has a large condition number, and this results in slow convergence and may lead to numerical instability.

To circumvent the first difficulty, we partition the matrix Φ into blocks and approximate each block with a low-rank matrix, as described below. To handle the high condition number, a change of basis is employed (see Sect. 4.1) [21].

3 Construction of a blockwise low-rank approximant

In this section, we discuss the construction of a blockwise low-rank approximant of the matrix Φ defined in (10). To discuss the fast RBF evaluation procedure, we consider two point clouds in $\mathbb{R}^2$, $(x_i)_{i=1}^N$ and $\left(y_j\right)_{j=1}^M$. The elements of the matrix Φ are $\Phi_{ij} = \phi\left(\|x_i - y_j\|\right)$. Suppose that we have found two sets of indices $I \subseteq (i)_{i=1}^N$ and $J \subseteq (i)_{i=1}^M$ such that the sets of points $\tau = \cup_{i \in I} x_i$ and $\nu = \cup_{j \in J} y_j$ are well separated; that is,

$$\max(\operatorname{diam} \tau, \; \operatorname{diam} \nu) \leq \eta \operatorname{dist}(\tau, \nu), \tag{11}$$

for some $\eta \in (0, 1)$. The diameter of a set of points ξ is the maximal distance between any pair of points in ξ,

$$\operatorname{diam} \xi = \max_{p,q \in \xi} \|p - q\|. \tag{12}$$

The distance between two sets of points is defined as

$$\operatorname{dist}(\tau, \nu) = \min_{x \in \tau, \; y \in \nu} \|x - y\|. \tag{13}$$

Consider the sub-block $\left\{\Phi_{ij}\right\}_{i \in I, \; j \in J}$ of the matrix Φ. This block has a low-rank approximant provided that the partial derivatives of $\tilde{\phi}(x, y) = \phi(\|x - y\|)$ decay sufficiently fast [15,16]. More precisely, $\tilde{\phi}$ must satisfy

$$\left|\partial_y^\alpha \tilde{\phi}(x, y)\right| \leq C_1 |\alpha|! \; C_2^{|\alpha|} R^{-|\alpha|} \sup_{\|y - z\| < R} |\tilde{\phi}(x, z)| \quad \text{for } R > 0, \tag{14}$$

where $R = |x - y|$, the constants C_1, C_2 are positive, and α is a multi-index. This low-rank approximant can be found by several methods. Although the truncated singular-value decomposition produces the best approximant, the procedure is computationally too expensive. An inexpensive alternative uses the interpolation of ϕ as in panel clustering [11], or the adaptive cross-approximation (ACA) algorithm [15–18]. Following this strategy, we construct a blockwise low-rank approximant of Φ using ACA.

3.1 Clustering of the data points

We use a hierarchical clustering method similar to that formulated in [15] for a set of boundary elements in $\mathbb{R}^3$. For a given set of points $(x_i)_{i=1}^N \subset \mathbb{R}^2$, we construct a cluster tree T_x as follows:

1. Let the set of all points be the root cluster of the tree.
2. Compute the mass center, and axes of inertia of the cluster.
3. Split the cluster into two parts by a straight line that passes through the center of mass and is orthogonal to the longest axis of inertia.
4. Assign the obtained groups of points as cluster offsprings.
5. Recursively apply the subdivision procedure to the offsprings so long as the cluster consists of more than one point.

A cluster tree T_y is then constructed for the second point set $\left(y_j\right)_{j=1}^M$. Having computed T_x and T_y, we can recursively generate a list of disjoint admissible blocks P that, together with the additional sparse part S, cover the whole matrix Φ. This can be accomplished by the following recursive procedure:

1. Set $\tau = T_x$, $\nu = T_y$, $P = \emptyset$, and $S = \emptyset$.
2. If τ or ν has only one element (no offsprings), add $\tau \times \nu$ to S and end the procedure.
3. If (τ, ν) satisfy (11), add $\tau \times \nu$ to P and end the procedure.
4. Denote by τ_1 and τ_2 the offsprings of τ, and by ν_1 and ν_2 the offsprings of ν. Proceed to Step 2 with (τ, ν) being (τ_1, ν_1), (τ_1, ν_2), (τ_2, ν_1), or (τ_2, ν_2).

On completion, the algorithm produces a list of admissible blocks P, satisfying (11), and a list S of small blocks. We compute all small blocks directly and approximate each admissible block using the ACA procedure.

3.2 Adaptive cross-approximation

Given an admissible block $\tau \times \nu$, the following procedure allows us to construct its low-rank approximation.

Algorithm 1

1. Initialization

$$R_0 = \tau \times \nu, \quad S_0 = 0. \tag{15}$$

2. for $i = 0, 1, 2, \ldots$

2.1. Pivot element

$$k_{i+1}, \ell_{i+1} = \text{ArgMax} \left|(R_i)_{k\ell}\right|. \tag{16}$$

2.2. Normalizing constant

$$\gamma_{i+1} = \left((R_i)_{k_{i+1}\ell_{i+1}}\right)^{-1}. \tag{17}$$

2.3. New vectors

$$u_{i+1} = \gamma_{i+1} R_i e_{\ell_{i+1}}, \quad v_{i+1} = R_i^\top e_{k_{i+1}}. \tag{18}$$

2.4. New residual

$$R_{i+1} = R_i - u_{i+1} v_{i+1}^\top. \tag{19}$$

2.5. New approximation

$$S_{i+1} = S_i + u_{i+1} v_{i+1}^\top. \tag{20}$$

Since the search for a pivot element (Step 2.1) involves scanning the full block, the method is coined the fully pivoted ACA. The numerical cost is $O(|\tau||\nu|)$, which results in a quadratic complexity for the blockwise low-rank approximant. However, the final memory requirement is almost linear.

A minor modification of the above procedure in Step 2.1 leads to so-called partially pivoted version of the ACA. In this procedure, the pivot element is chosen among already computed entries of the residual R. This reduces the complexity of the algorithm to $O(r(|\nu| + |\tau|))$, where r is the rank of the approximation. A closer investigation of r shows that the numerical cost for constructing the blockwise low-rank approximant of the whole matrix is almost linear (see [15,18]). Using a limited amount of block entries during the procedure requires corrections to the stopping criteria to ensure that the approximant covers the whole block even in some pathological cases. An extensive discussion of the partially pivoted ACA algorithm can be found in [16] and [22].

4 RBF interpolation and approximation

A blockwise low-rank approximant yields an efficient matrix-vector multiplication procedure, which is used to solve the system (10) by the generalized minimal residual iterative solution scheme (GMRES) [23]. Before turning to discussing numerical experiments, we describe a preconditioning technique that accelerates the convergence.

4.1 Preconditioning

Let diam X denote the diameter of our point set $X = \cup_i x_i$ (see (12)), and h_X denote the smallest distance between neighboring points in X,

$$h_X = \min_i \min_{j \neq i} \|x_i - x_j\|. \tag{21}$$

Because in the case of the TPS basis function, the upper bound on the condition number of Φ can be large [24]

$$\text{cond } \Phi \leq CN\phi(\text{diam } X)\, h_X^{-2}, \tag{22}$$

we use a preconditioning technique described in [21]. The preconditioning method is based on a special change of basis, which is particularly effective when a GMRES solution scheme is used. Consider a set of cardinal functions $\{\psi_i\}_{i=1}^N$, $\psi_i(x_j) = \delta_{ij}$, where δ_{ij} is the Kronecker delta. The interpolation matrix for such functions is a perfectly conditioned unit matrix I. To improve the conditioning of our interpolation system, we change the basis from ϕ to ψ,

$$\psi_j(x) = P_m(x) + \sum_{i=1}^{N} \nu_{ji}\phi(\|x_i - x\|). \tag{23}$$

However, performing this change of basis requires more computations then obtaining the solution of the original system. We get around this difficulty by using approximate cardinal functions, where the sum in the above formula is taken over $\beta + \gamma \ll N$ terms. First, we assign $\beta - 1$ data points closest to each x_j and call this set S_j. We also specify γ uniformly distributed points in the interpolation domain and denote this set by $\hat{S}$. Then, instead of (23), we define the new basis by

$$\psi_j(x) = P_m(p) + \sum_{x_i \in S_j \cup \hat{S}} \nu_{ji}\phi(\|x_i - x\|), \tag{24}$$

where

$$\psi_j(x_i) = \delta_{ij}. \tag{25}$$

In these settings, all coefficients ν_{ji} can be obtained at a linear cost of $O\left(N(\beta + \gamma + l(m))^3\right)$ flops. These coefficients form a sparse preconditioning matrix for the system (10).

Typical values of parameters used in our computations are $\beta \sim 40$ and $\gamma \sim 16$. With the preconditioning strategy, the system for interpolating $\sim 10^5$ points and $h_X \sim 10^{-4}$ requires about 100 GMRES iterations to converge to a residual of 10^{-10}. Without the preconditioner, the computations do not converge to the residual.

The right-hand side of (22) suggests choosing the ACA approximation accuracy $\epsilon \leq (\text{cond}\,\Phi)^{-1}$ when constructing the approximant to Φ.

4.2 Reconstruction

Having solved the system, we evaluate the RBF on a set of points $X' = \cup_{i=1}^{M} y_j$,

$$s(y_j) = \sum_{i=1}^{N} a_i \phi(\|x_i - y_j\|) + P(y_j), \quad j = 1, \ldots, M. \tag{26}$$

Evaluation with a desired accuracy ϵ' can be accomplished efficiently by constructing the blockwise low-rank approximant to the matrix $\Phi' = \left(\phi\left(\|x_i - y_j\|\right)\right)_{i=1,N,j=1,M}$. Now the value of the ACA approximation accuracy ϵ can be chosen so that $\epsilon \leq \epsilon'$. With the coefficients a_i at hand, we can also find the energy $\mathcal{E} = a^T \Phi a$ and partial derivatives $s_1'(x)$ and $s_2'(x)$ of the interpolant (or approximant). The evaluation of partial derivatives on X' can be accomplished efficiently using the same blockwise low-rank approximation strategy. In the case of s_1', we rewrite the derivative as

$$\begin{aligned} s_1'(y_j) &= \sum_{i=1}^{N} a_i \phi'_{y_{j,1}}\left(\|x_i - y_j\|\right) + c_1 \\ &= \sum_{i=1}^{N} a_i \left(2 \log \|x_i - y_j\| + 1\right)\left(x_{i,1} - y_{j,1}\right) + c_1 \\ &= \tilde{\Phi}^1 a + c_1, \end{aligned} \tag{27}$$

where $x_i = (x_{i,1}, x_{i,2})$ and $\tilde{\Phi}^1_{ij} = \left(2 \log \|x_i - y_j\| + 1\right)\left(x_{i,1} - y_{j,1}\right)$ when $\|x_i - y_j\| > 0$ or $\tilde{\Phi}^1_{ji} = 0$ otherwise. This generating function is asymptotically smooth. Therefore, the matrix $\tilde{\Phi}^1$ can be approximated by a blockwise low-rank using the ACA. The same argument holds for s_2' and the corresponding matrix $\tilde{\Phi}^2$.

4.3 Application to Franke's function

Before turning our attention to measured data, we consider a function introduced by Franke [25] as a standard test,

$$\begin{aligned} f(x) = &\frac{3}{4} e^{-\left((9x_1-2)^2 + (9x_2-2)^2\right)/4} + \frac{3}{4} e^{-(9x_1+1)^2/49 - (9x_2+1)^2/10} \\ &+ \frac{1}{2} e^{-\left((9x_1-7)^2/4 + (9x_2-3)^2\right)} - \frac{1}{5} e^{-(9x_1-4)^2 - (9x_2-7)^2}. \end{aligned} \tag{28}$$

The graph of this function is given in Fig. 1. The tests are based on sets of scattered data sites with increasing point density from $N = 1{,}000$ to $500{,}000$ on the square $[-0.1, 1.1] \times [-0.1, 1.1]$. The coordinates of each point were obtained using a pseudo-random generator with a uniform distribution. After solving system (10) using a preconditioned GMRES scheme, we evaluate the interpolant $s(p)$ and gradient $\nabla s(p) = \left(s_x'(p), s_y'(p)\right)$ on a regular grid of 200×200 points. On this grid, we compute the mean square errors

$$\delta_2 = M^{-1}\left(\sum_{i=1}^{M} \left(s(y_i) - f(y_i)\right)^2\right)^{1/2}, \tag{29}$$

$$\delta_{\nabla 2} = M^{-1}\left(\sum_{i=1}^{M} \|\nabla s(y_i) - \nabla f(y_i)\|^2\right)^{1/2}, \tag{30}$$

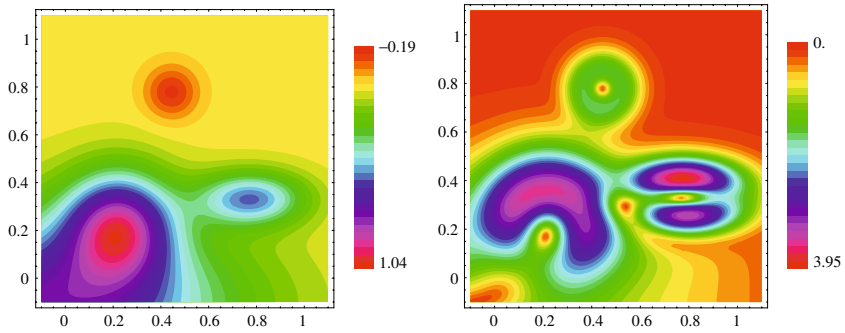

Fig. 1 Contours of Franke's function, $\mathcal{F}$, and its gradient, $\|\nabla\mathcal{F}\|$

Table 1 Accuracy of the interpolation for Franke's function

N	H_X	δ_2	δ_∞	$\delta_{\nabla 2}$
1,000	6.366E−02	3.40E−06	5.32E−03	8.897E−5
5,000	2.962E−02	4.38E−07	1.13E−03	2.576E−5
10,000	2.357E−02	2.55E−07	1.04E−03	1.819E−5
25,000	1.653E−02	9.76E−08	4.26E−04	1.038E−5
50,000	9.895E−03	4.71E−08	1.31E−04	7.518E−6
100,000	7.106E−03	2.51E−08	1.24E−04	5.189E−6
500,000	3.454E−03	4.85E−09	1.93E−05	2.447E−6

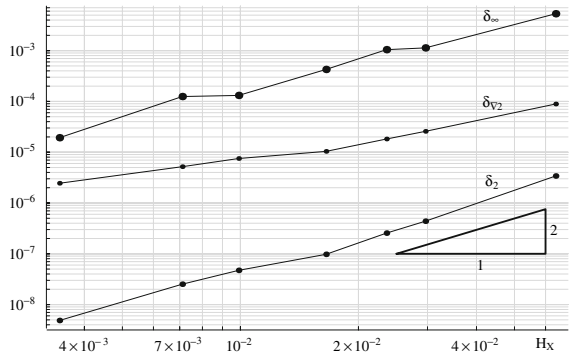

Fig. 2 Dependence of the mean square error δ_2, pointwise error δ_∞, and mean square error of the gradient $\delta_{\nabla 2}$ on the data spacing, H_X

and the maximum difference $\delta_\infty = \max_i |s(y_i) - f(y_i)|$ with respect to the known function values (28). The interpolation accuracy is reported in Table 1 and Fig. 2. As expected, a quadratic dependence of δ_2 on the separation distance $H_X = \max_i \min_{j\neq i} |x_i - x_j|$ is observed (see [26]). We see that δ_∞ behaves as $O\left(H_X^2\right)$ and $\delta_{\nabla 2}$ behaves somewhat better than $O(H_X)$. Table 2 illustrates the performance of the ACA compression technique when solving the interpolation problem. The memory requirement ("Mem") is given along with the compression rate ("rat") for the approximation as a measure of the data reduction achieved through ACA. The number of GMRES iterations (#) and CPU time ("sol. t.") are also given. The approximation precision for ACA was chosen $\epsilon = 10^{-10}$ in all numerical experiments. Such accurate approximation is necessary for large point densities since the expected matrix condition number is large. However, computations for $N < 10^5$ can be performed with a larger ϵ value, thus achieving better timing and compression.

Table 2 Performance of the interpolation procedure for Franke's function

N [10^3]	h_X	Mem [MB]	rat. [%]	compr. t. [s]	GMRES #	sol. t. [s]
1	9.449E−04	2.5	33.80	< 1	25	< 1
5	9.635E−05	28.1	14.73	3	27	1
10	9.635E−05	69.2	9.07	6	29	4
25	2.602E−05	199.5	4.18	22	30	12
50	1.744E−05	466.4	2.44	55	33	33
100	1.379E−05	1122.4	1.47	141	40	110
500	2.193E−06	7540.8	0.39	1028	199	3693

Table 3 Performance of the ACA compression for different basis functions

N [10^3]	BF	Mem [MB]	rat. [%]	Time [s]
25	TPS	199.5	4.18	22
	MQS	189.2	3.96	23
	POW	195.5	4.10	24
	G	55.5	1.16	3
50	TPS	466.4	2.44	55
	MQS	431.0	2.26	65
	POW	471.2	2.47	83
	G	117.6	0.61	8
100	TPS	1122.4	1.47	141
	MQS	968.7	1.26	105
	POW	1067.5	1.39	137
	G	290.5	0.38	20
500	TPS	7540.8	0.39	1028
	MQS	5531.3	0.29	733
	POW	6827.8	0.36	908
	G	1647.3	0.08	115

4.4 Performance for other basis functions

Since the ACA procedure involves only evaluations of the basis function, the same construction of a blockwise low-rank approximant can be applied for any asymptotically smooth ϕ. To test the compression procedure, we compute the blockwise low-rank approximants to matrices Φ for the MQS (6), POW (7), and the Gaussians (G),

$$\phi(r) = \mathrm{e}^{-r^2/2}. \tag{31}$$

The performance of the compression procedure is illustrated in Table 3. Thin plain splines (TPS), MQS, and POW result in similar compression ratio and timing. The compression rates for the Gaussian basis function is about four times higher than those of TPS, MQS or POW. However, the Gaussian leads to a much higher condition numbers for the matrix Φ (see [27]).

Table 4 Performance of the interpolation procedure for data sets A, B, and C

Set	N	h_X	H_X	rat. [%]	compr. t. [s]	GMRES #	sol. t. [s]
A	10,201	6.0E−03	7.2E−03	8.40	10	36	6
B	30,727	4.3E−05	2.7E−02	3.97	25	118	64
C	118,205	2.9E−04	3.9E−03	1.35	220	141	413

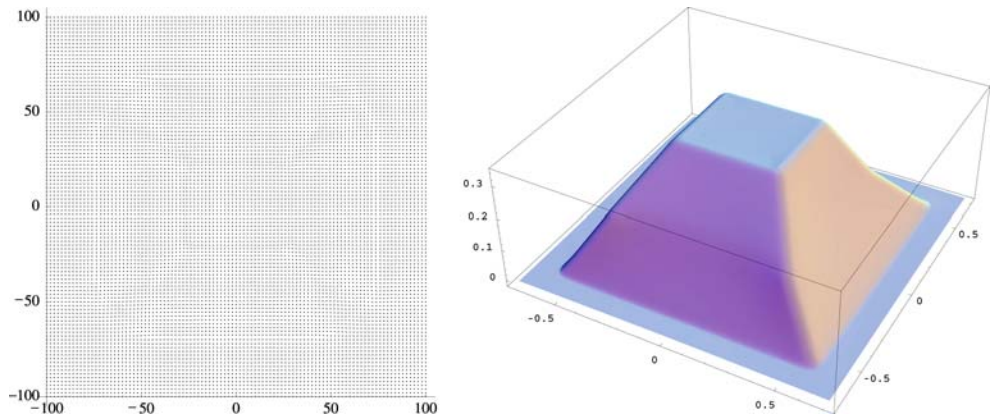

Fig. 3 Data set A and the corresponding interpolation results

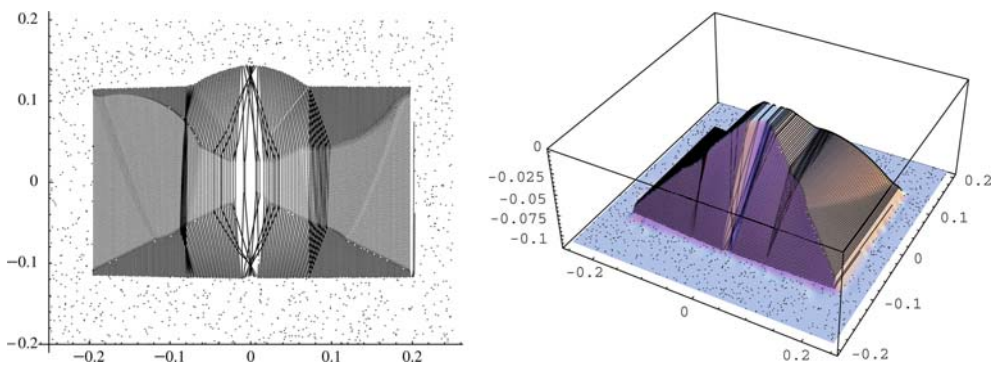

Fig. 4 Data set B and the corresponding interpolation results

4.5 Application to digitized surfaces

We now consider an example from a sheet-metal forming process yielding physical parts whose inclination with respect to the forming axis (z-axis) is less than 90 degrees. A part without undercuts can be represented by an RBF interpolant. After the forming is finished, the coordinates of points on the top and bottom surface of the sheet are obtained by digitizing both surfaces, and the measured surfaces are interpolated using the RBF (1). Table 4 illustrates the performance of the interpolation procedure for the data sets depicted in Figs. 3–5. We see that compression rates for these data sets are comparable to those obtained in Sect. 4.3. We also observe an increase in iteration number for the set B due to small h_X values. The value of H_X for this data set is rather large, and this can result in considerable deviations of the graph of $s\,(p)$ from the true shape of the metal sheet.

If we take a non-zero λ, the graph of $s\,(p)$ will not necessarily pass through data points $(x_i,\, f_i)$, but the bending energy $\mathcal{E}$ will decrease. This decrease is desirable, provided that the values of s at the data points are still close to f_i i.e., $\max |f_i - s\,(x_i)| \leq \mu$, where μ is the accuracy of the measurements. To investigate this balance, in Figs. 6

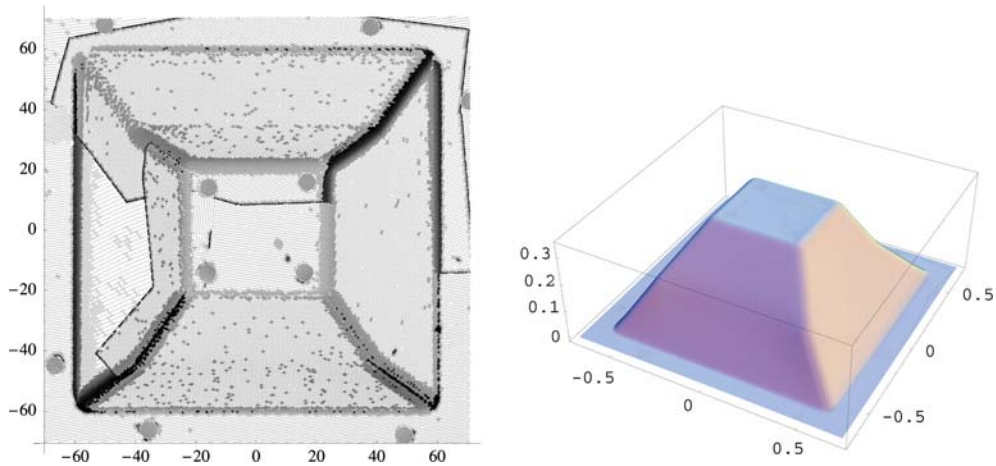

Fig. 5 Data set C and the corresponding interpolation results

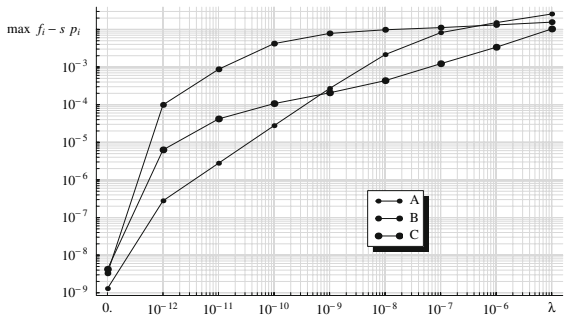

Fig. 6 Influence of λ on max $|f_i - s(p_i)|$. The error of magnitude 10^{-9} for $\lambda = 0$ is due to the GMRES residual (set to 10^{-10}) and to the ACA precision set to 10^{-10}

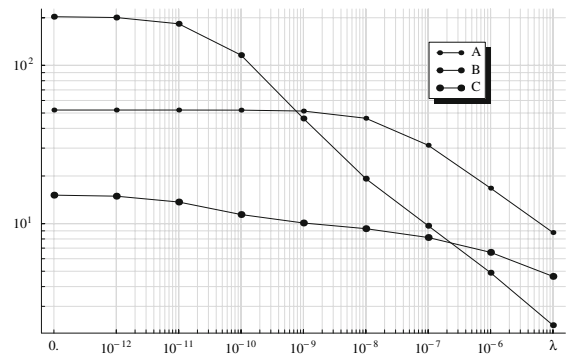

Fig. 7 Influence of λ on the bending energy, $\mathcal{E}$

and 7 we plot max $|f_i - s(x_i)|$ and $\mathcal{E}$ versus λ. Since the error in our measurements is 0.1 mm $= 10^{-4}$, we may take $\lambda \sim 10^{-10}$ when interpolating the data sets A and C.

4.6 Estimate of thickness

A typical objective in sheet-metal forming is the measurement of the thickness of formed sheet-metal products. This is especially important when the parts should meet stiffness specifications. By measuring the shape of both the top and bottom face of the sheet-metal part (in the same coordinate system), we can calculate the sheet-thickness distribution for the whole part as follows. Points from the first set represent the top surface, and points from the second set represent the bottom surface of the sheet. We find RBFs s_t and s_b approximating both surfaces and evaluate their half sum, $\tilde{f}_i = (s_t(y_i) + s_b(y_i))/2$, at regular grid points y_i. Then we construct the RBF interpolants $\tilde{s}$, $\tilde{s}'_1$ and $\tilde{s}'_2$ that approximate this middle surface and its partial derivatives. Having obtained $\tilde{s}'_1$ and $\tilde{s}'_2$ allows us to compute the normal vector $\mathcal{N}$. For each y_i we can now compute an approximate thickness value t using a cosine rule $t = \mathcal{N}_3(s_t - s_b)$. The computed thickness distribution for the data set A is shown in Fig. 8. The initial thickness of the undeformed sheet is 1 mm. Comparing this distribution with the shape of the metal part, we see that the sheet appears thinner in areas where the slope is steep. Some small thickening regions along the sides of the deformed area

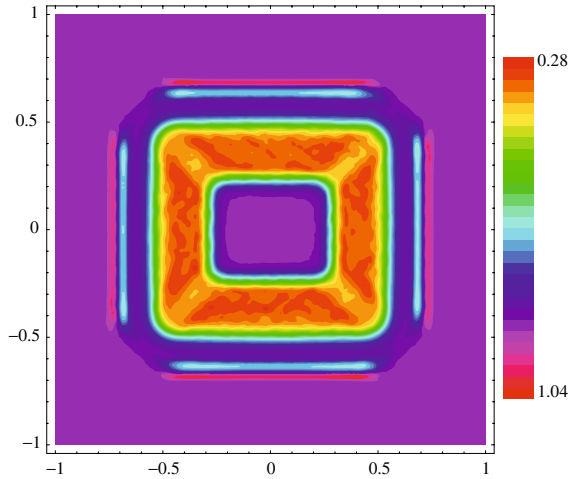

Fig. 8 Estimated distribution of thickness in mm along the deformed metal sheet

are also observed. Overall, the developed interpolation technique can be used to calculate the thickness distribution of a sheet-metal part non-destructively, provided that both sides can be accessed by a digitizer.

5 Conclusions

Our study shows that the matrices arising in the RBF interpolation can be approximated with high accuracy and efficiency by blockwise low-rank matrices. We have used this result to solve scattered data interpolation problems avoiding construction of a mesh. The ACA technique allows to treat problems with up to 500,000 data points. The construction of the RBF interpolant in this case takes about one hour on a single CPU PC. In contrast to multipole-like methods, it is possible to perform the interpolation using different basis functions without much re-coding.

Overall, the numerical experiments confirm the almost linear storage and complexity order estimates. The method performs well on the measurement data for sheet-metal forming, where the use of TPS basis function offers a number of benefits. The differentiability of the RBF interpolant in this case allows the fast reconstruction of the sheet's normal vector, which is used to calculate thickness estimates.

Although our study focuses on two-dimensional considerations, the method of constructing blockwise low-rank approximants to the corresponding matrices also applies to RBF interpolation in three or more dimensions. Future developments of the technique presented here might also involve an efficient construction of preconditioners using the hierarchical matrix structure, as it was done in [28,29] for matrices arising in boundary-element and finite-element methods.

Acknowledgements This study is the result of a collaboration under the DFG project SPP1146 "Modellierung inkrementeller Umformverfahren". The authors would like to thank the German Research Foundation (DFG) for financial support.

References

1. Hardy R (1990) Theory and applications of the multiquadric-biharmonic method. 20 years of discovery 1968–1988. Comput Math Appl 19(8–9):163–208
2. Bookstein FL (1989) Principal warps: thin-plate splines and the decomposition of deformations. IEEE Trans Pattern Anal Mach Intell 11:567–585
3. Barrodale I, Kuwahara R, Poeckert R, Skea D (1993) Side-scan sonar image processing using thin plate splines and control point matching. Numer Algorithms 5(1–4):85–98

4. Flusser J (1992) An adaptive method for image registration. Pattern Recognition 25(1):45–54
5. Dyn N, Levin D, Rippa S (1986) Numerical procedures for surface fitting of scattered data by radial functions. SIAM J Sci Statist Comput 7(2):639–659
6. Appel A (1985) An efficient program for many-body simulation. SIAM J Sci Statist Comput 6(1):85–103
7. Barnes J, Hut P (1986) A hierarchical O(N log N) force-calculation algorithm. Nature 324(6096):446–449
8. Greengard L, Rokhlin V (1987) A fast algorithm for particle simulations. J Comput Phys 73(2):325–348
9. Greengard L (1988) The rapid evaluation of potential fields in particle systems. ACM Distinguished Dissertations, MIT Press, Cambridge, MA
10. Ying L, Biros G, Zorin D (2004) A kernel-independent adaptive fast multipole algorithm in two and three dimensions. J Comput Phys 196(2):591–626
11. Hackbusch W, Nowak ZP (1989) On the fast matrix multiplication in the boundary element method by panel clustering. Numer Math 54(4):463–491
12. Beatson RK, Newsam GN (1992) Fast evaluation of radial basis functions. I. Comput Math Appl 24(12):7–19. Advances in the theory and applications of radial basis functions.
13. Cherrie J, Beatson R, Newsam G (2002) Fast evaluation of radial basis functions: methods for generalized multiquadrics in $\mathbb{R}^n$. SIAM J Sci Comput 23(5):1549–1571 (electronic)
14. Potts D, Steidl G, Nieslony A (2004) Fast convolution with radial kernels at nonequispaced knots. Numer Math 98(2):329–351
15. Bebendorf M (2000) Approximation of boundary element matrices. Numer Math 86(4):565–589
16. Bebendorf M, Grzhibovskis R (2006) Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation. Math Meth Appl Sci 29(14):1721–1747
17. Bebendorf M, Rjasanow S (2000) Matrix compression for the radiation heat transfer in exhaust pipes. In: Wendland W, Sändig A-M, Schiehlen W (eds) Multifield Problems. State of the Art. Springer, Berlin, pp 183–192
18. Bebendorf M, Rjasanow S (2003) Adaptive low-rank approximation of collocation matrices. Computing 70:1–24
19. Micchelli CA (1986) Interpolation of scattered data: distance matrices and conditionally positive definite functions. Constr Approx 2(1):11–22
20. Meinguet J (1979) Multivariate interpolation at arbitrary points made simple. Z Angew Math Phys 30(2):292–304
21. Beatson RK, Cherrie JB, Mouat CT (1999) Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration. Adv Comput Math 11(2–3):253–270. Radial basis functions and their applications
22. Rjasanow S, Steinbach O (2007) The fast solution of boundary integral equations, Mathematical and Analytical Techniques with Applications to Engineering. Springer-Verlag, Berlin (to appear)
23. Saad Y, Schultz MH (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Statist Comput 7(3):856–869
24. Schaback R (2002) Stability of radial basis function interpolants. In: Approximation theory, X (St. Louis, MO, 2001). Vanderbilt University Press, Innov Appl Math Nashville, pp 433–440
25. Franke R (1982) Scattered data interpolation: tests of some methods. Math Comp 38(157):181–200
26. Wendland H (1997) Sobolev-type error estimates for interpolation by radial basis functions. In: Surface fitting and multiresolution methods (Chamonix–Mont-Blanc, 1996). Vanderbilt University Press, Nashville, pp 337–344
27. Wendland H (2002) Scattered data modelling by radial and related functions. Habilitation Thesis, Universität Göttingen, Gottingen
28. Bebendorf M (2005) Hierarchical LU decomposition-based preconditioners for BEM. Computing 74(3):225–247
29. Bebendorf M (2006) Approximate inverse preconditioning of finite element discretizations of elliptic operators with nonsmooth coefficients. SIAM J Matrix Anal Appl 27(4):909–929 (electronic)