

A Parameter-Free Adaptive Method for Radial Basis Function Interpolation of Bump Functions

Terhemen Aboiyar and Samuel T. Swem
Department of Mathematics/Statistics/Computer Science,
University of Agriculture, Makurdi, Nigeria

Abstract: In this study, researchers describe and implement a parameter free adaptive algorithm for radial basis function interpolation of real valued bump functions. The method is based on thin plate splines which are known for their good approximation properties. Interpolation points are added and coarsened based on residuals computed on a finer discretization. Numerical examples in one dimension are presented to show the effectiveness of the method.

Key words: Radial basis functions, thin plate splines, parameter free, adaptivity, interpolation, algorithm

INTRODUCTION

If X is a finite set of scattered data points in \mathbb{R}^d and if $f = \mathbb{R}^d \rightarrow \mathbb{R}$ is a function known only on X , then an important practical problem is that of finding a smooth function that interpolates f and provides to good approximation to f near X . To this end, interpolation by Radial Basis Functions (RBFs) has become a very powerful tool in multivariate approximation theory. In the past two decades, radial basis functions have been used extensively in fields like the numerical solution of partial differential equations like in Franke (1998) and Lorentz (2003).

They have also been applied in non-uniform sampling, mathematical finance, computer graphics and in optimization. The efficiency of the RBF interpolation on scattered and uniform data has been illustrated by several numerical examples and theoretical results by Wendland (2005) and Fasshauer (2007).

For problems with steep gradients, singularities and other topological changes, it is natural to use adaptive methods to reduce the error of the interpolation and approximation. Adaptive methods have been implemented for RBF interpolation by Behrens and Iske (2002), Sarra (2005) and Driscoll and Heryudono (2007) amongst others.

Although, several radial basis function like Gaussians and multiquadratics have very good approximation properties, they have a shape parameter and numerical experiments in Fasshauer (2007) and elsewhere show how the choice of the shape parameter has profound influence on both the accuracy and

numerical stability of the solution of an interpolation problem. On the other hand, certain RBFs like radial powers and surface splines have the advantage of being shape parameter free. This means that one does not need to worry about the choice of a good shape parameter.

In Driscoll and Heryudono (2007), an adaptive residual subsampling method based on radial basis functions was proposed and implemented. In their research, they used multiquadratics which contain a shape parameter and they show how the value of the shape parameter affects the accuracy of their method. To this end, they had to adaptively modify the shape parameter during simulation.

In this study, the researchers will use a parameter free radial basis function, the thin plate splines (a type of surface splines), to implement an adaptive radial basis function interpolation scheme for bump functions and show how good results can be obtained without the need of adapting the shape parameter. A bump function is a function that is non-zero only on a compact subset of the interior of the domain of definition. The utility of thin plate splines interpolation in this context was highlighted by Gutzmer and Iske (1997) and Kaser (2003).

Radial basis function interpolation: Consider a data vector $f = (f(x_1), \dots, f(x_n))^T \in \mathbb{R}^n$ function values, obtained from a function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ at a scattered finite point set $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, $d \geq 1$. Scattered data interpolation requires computing an appropriate interpolant $s: \mathbb{R}^d \rightarrow \mathbb{R}$ satisfying;

$$s(x_j) = f(x_j) \text{ for all } 1 \leq j \leq n \quad (1)$$

The radial basis function interpolation method utilizes a fixed radial function $\phi: [0, \infty] \rightarrow \mathbb{R}$ so that the interpolant s in Eq. 1 has the form:

$$s(x) = \sum_{j=1}^n c_j \phi(\|x - x_j\|) + p(x), \quad p \in P_{m-1}^d \quad (2)$$

Where $\|\cdot\|$ is the Euclidean norm on \mathbb{R}^d and P_{m-1}^d is the space of all polynomials in d variables of degree at most $m-1$ (order m), where $m = m(\phi)$ is known as the order of the basis function ϕ . Possible choices for ϕ along with their orders are shown in Table 1. When $m = 0$, the interpolant s in Eq. 2 has the form:

$$s(x) = \sum_{j=1}^n c_j \phi(\|x - x_j\|) \quad (3)$$

Using the interpolation conditions Eq. 1, the coefficients $c = (c_1, \dots, c_n)^T$ of s in Eq. 3 can be obtained by solving the linear system:

$$Ac = f$$

$$A = \left(\phi(\|x_i - x_j\|) \right)_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}$$

Where for $m > 0$, the interpolant s in Eq. 2 contains a nonzero polynomial part, yielding q additional degree of freedom where q is the dimension of the polynomial space P_{m-1}^d . These additional degrees of freedom are usually eliminated using q vanishing moment conditions:

$$\sum_{j=1}^n c_j p(x_j) = 0, \quad \text{for all } p \in P_{m-1}^d \quad (4)$$

In total, this amounts to solving the $(n+q) \times (n+q)$ linear system:

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \quad (5)$$

Where:

$$A = \left(\phi(\|x_i - x_j\|) \right)_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}$$

$$P = \left((x_j)^\alpha \right)_{1 \leq i, j \leq n, |\alpha| < m} \in \mathbb{R}^{n \times q} \text{ and } d = (d_\alpha)_{|\alpha| < m}$$

Table 1: Radial Basis Functions (RBFs), their parameters and order

RBF	$\phi(r)$	Parameters	Shape parameter	Order
Surface splines	r^k for $k \in 2\mathbb{N}$	$k \in \mathbb{N}$	-	k
	$r^k \log(r)$ for $k \in 2\mathbb{N}$	$k \in \mathbb{N}$	-	k
Gaussians	$\exp(-\varepsilon r^2)$	-	ε	0
Multiquadrics	$(1 + \varepsilon r^2)^\nu$	$\nu > 0, \nu \in \mathbb{N}$	ε	$\lceil \nu \rceil$
Inverse multiquadrics	$(1 + \varepsilon r^2)^\nu$	$\nu < 0$	ε	0

for the coefficients of the polynomial part in Eq. 2. The important case is the thin plate spline (surface splines with $k = 2$). In this case, for one space dimension, we have $\phi(r) = r^2 \log r$ with:

$$m = 1 \text{ and } q = \dim P_1^1 = 2$$

so that;

$$s(x) = \sum_{j=1}^n c_j |x - x_j|^2 \log |x - x_j| + d_0 + d_1 x \quad (6)$$

This requires us solving an $(n+2) \times (n+2)$ linear system to obtain the coefficients $c_1, \dots, c_n, d_0, d_1$. The effectiveness of this interpolant for bump functions was shown by Bejancu (2001). We note, however that parameter free radial basis functions will not be able to achieve the spectral convergence rates that are possible with other basis functions such as Gaussians and multiquadrics. Moreover, the interpolation matrices are larger in this case because of the polynomial part (Fasshauer, 2007).

The adaptive interpolation method: We will use the residual subsampling of Driscoll and Heryudono (2007) coupled with the strategy of Kaser (2003) to provide a parameter free adaptive interpolation method.

A function $f(x)$ defined on an interval (a, b) is picked and an initial discretization giving a set X_0 of n_0 equally spaced points is obtained. A thin plate spline interpolant $s(x)$ is then obtained for the function $f(x)$. The interpolation error is then computed at the set M_0 of n_0^{-1} midpoints of the subintervals. A midpoint $m_i \in M_0$ becomes an interpolation point if the error at m_i exceeds a threshold value ϑ_T and an interpolation $x \notin X$ is coarsened or removed if it lies between two points in M_0 whose error is below a smaller threshold ϑ_c .

This gives a new discretization X_1 of the interval (a, b) . This algorithm is repeated and further discretizations X_2, X_3, X_4, \dots are generated until no further points are added or coarsened. For the case of thin plate spline interpolation, the local error estimate at $x \in X$ is according to Wu and Schaback (1993) of the form:

$$|f(x) - s(x)| \leq C \cdot h_{x,p}^{\frac{k}{2}} \quad (7)$$

Where $C > 0$ is a constant depending on f and (for some radius $p > 0$):

$$h_{x,p} = \sup_{\|y - x\| < p} d_X(y)$$

is the fill distance of X around x , with:

$$d_x(y) = \min_{v \in X} \|y - v\|$$

being the Euclidean distance between the point y and the set X . As shown by Wu and Schaback (1993), the reduction in the error (Eq. 7) around any interpolation point is accomplished by reducing the distance function;

$$d_x = \min_{v \in X} \|y - v\|$$

in a neighbourhood of x_i . This shows that adaptivity can indeed reduce the interpolation error.

Numerical examples and discussions

Example 1: We will use a simple function $f(x) = \exp(-40x^2)$ on the interval $[-1, 1]$ with $\vartheta_r = 1.5 \times 10^{-5}$ and $\vartheta\alpha = 1 \times 10^{-6}$. The results are shown in the Table 2 where $\text{cond}(A)$ denotes the condition number of the interpolation matrix, E_∞ denotes the error in the infinity norm, na the number of points added, nc the number of

points coarsened and n denotes the number of interpolation points. The method converges in 8 iterations and the errors are quite small. Moreover, the method is stable looking at the condition numbers of the interpolation matrices.

These results are similar to the multiquadric method while the conditioning of the interpolation matrix is even better. The plots for $f(x)$, the node distribution at the final iteration and the errors are shown in Fig. 1a-c, respectively. We observe that the method adaptively places more interpolation points close to where the function has sharp features.

We also note that when interpolating on 111 equally spaced points the error will be 1.1656×10^{-4} with the condition number 2.6166×10^6 which shows that the adaptive method gives superior results for the same number of interpolation points.

Example 2: We will consider the bump function given by the product (Bejancu, 2001):

Table 2: Node distribution, error and condition number for Example 1

Iterate	n	E_∞	Cond(A)	na	nc
1	013	3.4475×10^{-2}	5.0692×10^2	12	0
2	025	1.0480×10^{-2}	3.3436×10^3	22	0
3	047	1.4049×10^{-3}	2.1967×10^4	30	2
4	075	2.5401×10^{-4}	1.0873×10^5	26	1
5	100	7.7772×10^{-5}	4.9485×10^5	12	2
6	110	5.0375×10^{-5}	1.8484×10^6	03	1
7	112	2.6684×10^{-5}	1.8302×10^6	01	0
8	111	1.4497×10^{-5}	1.8670×10^6	00	1

Table 3: Node distribution, error and condition number for Example 2

Iterate	n	E_∞	Cond(A)	na	nc
1	013	5.2124×10^{-2}	1.5458×10^2	12	0
2	025	5.3190×10^{-3}	9.3359×10^3	22	0
3	047	6.3139×10^{-4}	5.5422×10^4	38	0
4	085	8.7013×10^{-4}	3.2702×10^5	54	0
5	139	3.1247×10^{-5}	1.8015×10^5	04	2
6	141	7.3743×10^{-5}	3.8989×10^6	02	1
7	142	3.9584×10^{-5}	3.9236×10^6	02	1
8	141	1.4958×10^{-5}	3.9564×10^6	00	1

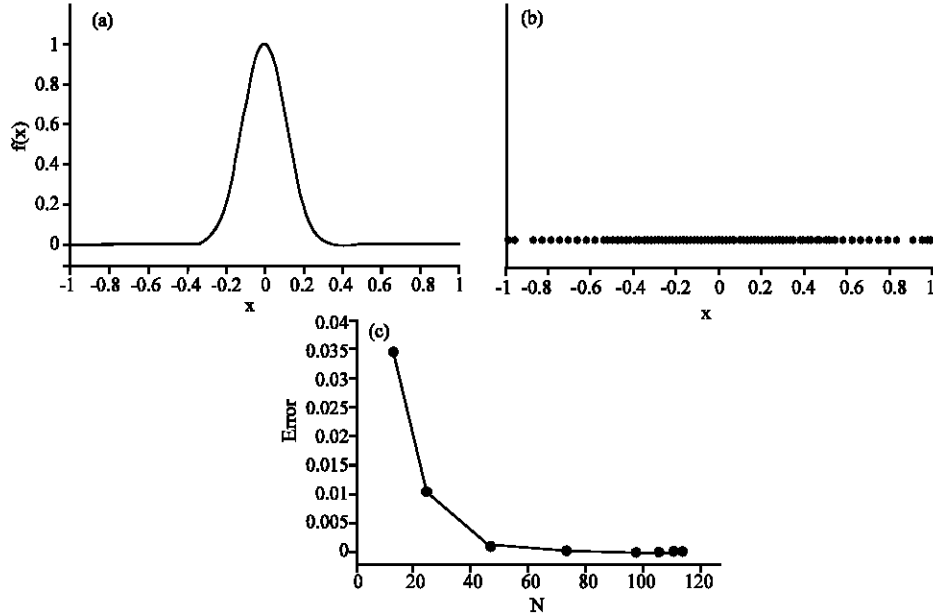


Fig. 1: a) Plot of $f(x) = \exp(-40x^2)$, b) node distribution at final iteration, c) number of interpolation points vs. error

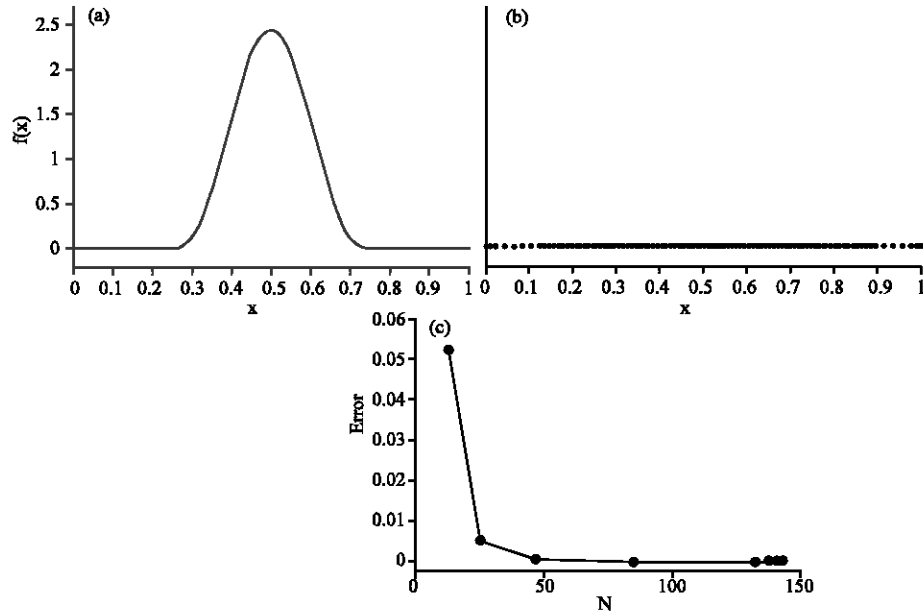


Fig. 2: a) Plot of $g_p^{(6)}$, b) node distribution at final iteration, c) number of interpolation points vs. error

$$g_p(x) = 10^{p+1} \left[\max \left\{ 0, x - \frac{1}{4} \right\} \right]^p \left[\max \left\{ 0, \frac{3}{4} - x \right\} \right]^p \quad (8)$$

$$x \in [0, 1]$$

Therefore, $g_p \in C^{p-1}$ and the support of g_p is the interval:

$$\left[\frac{1}{4}, \frac{3}{4} \right]$$

We will still use $\vartheta_r = 1.5 \times 10^{-6}$ and $\vartheta_d = 1 \times 10^{-6}$ and use $p = 3$. The results are shown in Table 3. The method converges in 8 iterations and the errors and condition numbers of the interpolation matrix are reasonably small. The plots for $g_p^{(6)}$, the node distribution at the final iteration and the errors are shown Fig. 2a-c, respectively. We observe that the method adaptively places more interpolation points close to where the function has sharp features. Finally, when interpolating on 141 equally spaced points the error will be 4.9640×10^{-5} with the condition number 1.0056×10^6 . Therefore, although the condition number of the adaptive method is a bit larger and the error is smaller. This once again proves the utility of the adaptive method.

CONCLUSION

We have shown that the adaptive radial basis function method provides good results for interpolation

of bump functions and can be used for other function with sharp gradients. We also see that the absence of the parameter does not significantly affect our results.

REFERENCES

- Behrens, J. and A. Iske, 2002. Grid-free adaptive semi-Lagrangian advection using radial basis function. *Comp. Math. Appl.*, 43: 319-327.
- Bejancu, A., 2001. On the accuracy of surface spline approximation and interpolation to bump functions. *Proc. Edinburgh Math. Soc.*, 44: 225-239.
- Driscoll, T. and A. Heryudono, 2007. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. *Comp. Math. Appl.*, 53: 927-939.
- Fasshauer, G.E., 2007. *Meshfree Approximation Methods in MATLAB*. World Scientific Press, Singapore.
- Franke, C., 1998. Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comput.*, 93: 73-82.
- Gutzmer, T. and A. Iske, 1997. Detection of discontinuities in scattered data approximation. *Numer. Algor.*, 16: 155-170.
- Kaser, M., 2003. Adaptive methods for the numerical simulation of transport processes. Ph.D. Thesis, Technical University Munich.

- Lorentz, R.A., 2003. Collocation discretizations of the transport equation with radial basis functions. *Appl. Math. Comput.*, 145: 97-116.
- Sarra, S.A., 2005. Adaptive radial basis function methods for time dependent partial differential equations. *Appl. Numer. Math.*, 54: 79-94.
- Wendland, H., 2005. *Scattered Data Approximation*. Cambridge University Press, Cambridge, ISBN: 9780521843355.
- Wu, Z. and R. Schaback, 1993. Local error estimates for radial basis function interpolation of scattered data. *IMA J. Numer. Anal.*, 13: 13-27.