GLOBALLY VERSUS COMPACTLY SUPPORTED RBFS

EDWARD J. KANSA
Convergent Solutions, Livermore, USA

ABSTRACT
For many years, a debate has occurred whether radial basis functions having compact support (CS) or global support (GS) is best for engineering and scientific applications. CS RBFs converge as $O(h^{(k+1)})$, h is the fill distance, and its systems of equations have many zeros. In contrast, GS RBFs converge as $O(\lambda(c/h))$, $\lambda < 1$, c is the GS-RBF shape parameter. Previously, the barrier to exploiting the exponential convergence rate of GS-RBFs has been the ill-conditioning problem that is due to computer chip restrictions on the relatively large machine epsilon. Although computer chips with arbitrary precision are very rare presently, extended precision software has allowed the exploitation of the exponential convergence rates of GS-RBFs. When attempting modeling of higher dimension practical problems, previous methods such as domain decomposition, global optimization, pre-conditioning will need to be blended even on massively parallel computers.
Keywords: radial basis functions, compact support, global support, polynomial convergence, exponential convergence, multi-precision arithmetic.

1 INTRODUCTION

Kansa [1], [2] presented a meshless radial basis function (RBF) to solve partial differential equations (PDEs). There are two approaches to solve partial differential and integral equations with meshless RBF methods: 1. "Compactly supported" RBFs (CS-RBFs); 2. "Globally supported" (GS-RBFs). The convergence rate for interpolation is $\lambda^{(c/h)}$, where $\lambda < 1$ [3]–[5]. The ratio, c/h, increases the rate of convergence by decreasing h, the average fill distance between pairs of points (h-refinement) or increasing the average value of the shape parameter, c, (c-refinement). The basic question is for a given application is whether h-refinement or c-refinement is more cost effective to achieve the target accuracy with the least computational effort. The answer to this question is not definitive but depends upon the nature of the application and the available resources.

In general, as system of equations becomes larger, the higher is the probability that rounding-errors will severely contaminate the solution. A measure of this severity is the condition number, $\kappa(\mathbf{A})$, of a system of equations relative to the inverse machine epsilon, $1/\varepsilon_M$. Whenever $\kappa(\mathbf{A}) > 1/\varepsilon_M$ the solution is worthless. While it is impossible to construct an ideal Platonic computer with infinite precision, infinite memory capacity that operates infinitely fast, there are some procedures that can be adopted to control the damage caused by the accumulation of round-off errors on a finite precision computer. Increasing the number of digits of precision requires both sacrificing memory and increasing the processing time per unit operation. This sacrifice is worthwhile if the total processing time to achieve target accuracy is smaller than the total processing time with single or double precision.

2 TREATING ILL-CONDITING

More or less standard methods have been used to treat either CS-RBF or GS-RBF systems such are regularization in which a small positive value is added to the diagonal elements (Tikhonov [6], Tikhonov and Arsenin [7], Tikhonov et al. [8]). Also related, is the truncated singular value decomposition (SVD) m0ethod in which the singular values below a certain threshold are neglected, and a modified set of linear equations is used to find the expansion

WIT Transactions on Engineering Sciences, Vol 120, © 2018 WIT Press
www.witpress.com, ISSN 1743-3533 (on-line)
doi:10.2495/AFM180251

coefficients. As successively more small singular values are deleted. The further does the modified matrix deviate from the original matrix or set of linear or linear set of equations.

Holobodrodko of the Advanpix LLC [9] consider the eigenvalues of a 150x150 symmetric Grcar matrix [10] whose condition number is 3.61. Obviously, a very low condition symmetric matrix ought to have very accurate eigenvalues; and it is obvious that the eigenvalues of both the Grcar matrix and its transpose ought to be identical. However, the Grcar matrix whose condition number is 3.61, on a double precision machine, produces incorrect, unequal eigenvalues. Only when the extended precision software package is used do the eigenvalues in a quadruple precision calculation become identical. As a cautionary note, ideal Platonic arithmetic and mathematics does exist on real computers; rather, fuzzy arithmetic and mathematics is perhaps more appropriate, with the fuzziness dependent upon the available precision on the computer.

The reproducing kernel based Gaussian method expands a Gaussian RBF as an eigenfunction of products of the Gaussian exponentials in $\Re^n$, truncated to a finite number of terms, M [11], [12]. The kernel expansion is still ill-conditioned, so a combination of a truncated SVD and the quotient residual (QR) methods are used to obtain a new set of basis functions. The change of that is much better conditioned for h-refinement.

Domain decomposition has been successfully to both CS-RBF and GS-RBF methods for PDEs. Domain decomposition can use either the overlapping or non-overlapping methods [13]–[19]. Assume the problem is continuous, and the computational domain is arbitrarily decomposed as a union of smaller subdomains containing an equal number of data centers (approximately). Just as with finite element methods, the larger the number of centers within a subdomain, the larger will the condition number be, either with CS-RBFs or GS-RBFs. The IE or PDE in each subdomain has its "boundary" and interior conditions. In each subdomain, the set of equations is solved on separate independent processors in parallel. Then the solutions in each subdomain are iteratively blended to enforce continuity of the function and its normal and tangential derivatives. Typically, only 2–3 iterations are required to obtain satisfactory blending convergence. However, the parallelism is somewhat different if compact or global support is used; this will be discussed later.

If the problem being solved is fairly homogeneous spatially, then increasing the shape parameters or c-refinement while increasing the arithmetic precision is vastly more efficient since the total number of data centers can be reduced by many orders of magnitude [20]. However, some problems may have so much fine scale structure that h-refinement, rather than c-refinement, is required and slower converging compactly supported RBF methods may be required for a given available computer. Common sense is required regarding how to approach a given problem using the available resources.

Very promising is the development of global optimization methods for solving systems of equations [21]–[28]. This is a very broad field of research and development that has the potential to solve large systems of both linear and nonlinear equations without the need for LU methods that exacerbates ill-conditioning. Global optimization can be either search procedures, stochastic procedures, or a hybrid mixture of both.

3 RECOMMENDED STRATEGIES

It is recommended here that a hybrid approach be used such as: matrix scaling, iterative refinement, pre-conditioning, domain decomposition, global optimization, and extended arithmetic precision. Higham [29] discussed the importance of multiple precision in the practical world, and specifically praised the work of Holobodroko for the development of the multi-precision software package [9].

In addition, the curse of dimensionality is a serious impediment to h-refinement because of memory storage issues even on the most recent massively parallel computers. Plasma physics is minimally a six-dimensional problem. Discrediting at 1,000 points/dimension requires a storage capacity of 10^{18} for just the data centers alone.

A hardware computer chip remedy for extended precision beyond 64 bits/word is most preferable. However, the consumer gadget market does not require much precision, unlike scientific computing. For scientific applications, only software methods are the only current method to obtain very high precision. The Advanpix mufti-precision computing toolbox (AMCT) concretely demonstrates that multi-precision arithmetic, if properly constructed, can be executed very efficiently, unlike other software packages for multi-precision calculations. The exponential convergence rate becomes vital in solving multi-dimensional (many independent variables).

It is important to present the case for GS-RBF methods for large scale, multi-bimensal IE and PDE problems that will be executed on parallel computers. Some tools for massively parallel computers are being developed in the languages, C and C++ [30], "The GNU Multiple Precision Arithmetic Library". At first glance, CS-RBFs are very appealing because the matrix associated with calculating the expansion coefficients is sparse.

The performance of direct solvers for sparse matrices using LU, CHOL, and other solvers, depends entirely on the locations of the non-zero elements. Bad locations of the zeros can require a full $O(N^3)$ computational complexity. To alleviate this complexity problem, all algorithms use matrix pre-ordering to minimize the fill-in; the time required for pre-ordering must be considered. The optimal pre-ordering algorithm is known only for symmetric-positive definite matrices. But, in the case of unsymmetrical matrices, optimal pre-ordering is an NP-complete task.

All the existing direct solvers for sparse systems use some kind of heuristic ideas to perform the pre-ordering. The complexity of direct solvers for sparse matrices can vary wildly from $O(N^{3/2})$ to $O(N^3)$. Direct sparse solvers have poor capability for parallel execution, whereas dense solvers are extremely suitable for parallelism. A discussion of the sparse complexity issues can be found at www.mathworks.com/help/matlab/math/sparse-matrix-operations.html.

Theoretically, iterative methods are the only way to beat the complexity of direct solvers since iterative solvers are supposed to converge within $O(N^2)$ operations. However, iterative solvers suffer from slow convergence and good pre-conditioners are required, but finding good pre-conditioners is separate problem that needs to be solved. It is possible that with careful and non-trivial tuning of all parts of sparse solvers (direct or iterative) can be faster than solving small dense matrix in ex studied and the computational environment for comparison.

Both GPUs (graphic processing units) and FPG tended precision on a single CPU. However, everything depends on the particular problem being A (field-programmable gate arrays) possess highly parallel structures that makes them more efficient than general-purpose CPUs (central processing units) for algorithms where the processing of large blocks of data is performed in parallel. However, GPUs and FPGAs have very limited functionality for branching operations that are abundant in sparse solvers making it very difficult for porting and running in parallel.

On the contrary, dense solvers enjoy massive parallelism because there is no branching and all manipulations can be formulated as efficient BLAS3/GEMM operations that are well optimized and are designed for massive parallelism on all imaginable platforms.

If the Advanpix toolbox is executed for full solvers on CPUs with 10 or more cores, or GPUs – the toolbox will beat sparse-based algorithms. The justification for this claim is that

dense solvers in the Advanpix toolbox are well suited for parallel execution whereas sparse solvers are not. Intuitive conclusions about the efficiency of sparse versus dense RBF systems can be misleading. In addition, sparse methods only enjoy polynomial convergence rates whereas global RBF systems enjoy exponential convergence rates.

REFERENCES

[1] Kansa, E.J., Multiquadrics: A scattered data approximation scheme with applications to computational fluid dynamics I: Surface approximations and partial derivative estimates. *Comput. Math. Applic*., **19**(8/9), pp. 127–145, 1990.

[2] Kansa, E.J., Multiquadrics: A scattered data approximation scheme with applications to computational fluid dynamics II: Solutions to parabolic, hyperbolic, and elliptic partial differential equations. *Comput. Math. Applic*., **19**(8/9), pp. 147–161, 1990.

[3] Madych, W.R. & Nelson, S.A., Multivariate interpolation and conditionally positive definite functions ii. *Math. Comp*., **54**, pp. 211–230, 1990.

[4] Madych, W.R., Miscellaneous error bounds for multiquadric and related interpolators. *Com-put. Math. Appl.,* **24**(12), pp. 121–138, 1992.

[5] Madych, W.R. & Nelson, S.A., Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *J. Approx. Theo*., **60**, pp. 94–114, 1992.

[6] Tikhonov, A.N., On the stability of inverse problems. *Dokl.Akad.i Nauk SSSR*, **39**(5), pp. 195–198, 1943.

[7] Tikhonov, A.N. & Arsenin, V.Y., *Solution of Ill-posed Problems*, Winston & Sons: Washington, 1977.

[8] Tikhonov, N., Goncharsky, A.V., Stepanov, V.V. & Yagola, A.G., *Numerical Methods for the Solution of Ill-Posed Problems,* Kluwer Academic Publishers, 1995.

[9] Advanpix LLC., Multiprecision computing toolbox for MATLAB, Yokohama, Japan, 2008–2017. www.advanpix.com.

[10] Grcar, J.F., Operator coefficient methods for linear equations. Technical Report SAND89-8691, Sandia National Labs, 1989.

[11] Fasshauer, G.E., Hickernell, F.J. & Wozniakowski, H., Rate of convergence and tractability of the radial function approximation problem. *SIAM J. Numer. Anal*., **50**(1), pp. 247–271, 2012.

[12] Fasshauer, G.E. & McCourt, M.J., Stable evaluation of Gaussian RBF interpellants. *SIAM J. Scient. Comput*., **34**(2), pp. A737–A762, 2012.

[13] Hon, Y.-C. & Wu, Z., Additive Schwarz domain decomposition with radial basis approximation. *Int. J. Appl. Math. Stat.,* **4**, pp. 81–98, 2002.

[14] Li, J. & Hon, Y.-C., Domain decomposition for radial basis meshless methods. *Num. Meth. PDEs,* **20**, pp. 450–462, 2004.

[15] Ingber, M.A., Chen, C.-S. & Tanski, A., A mesh free approach using radial basis functions and parallel domain decomposition for solving three-dimensional diffusion equations. *Int. J. Num. Meth. Eng*., **60,** pp. 2183–2201, 2004.

[16] Duan, Y., Tang, P.F., Huang, T.Z. & Lai, S.J., Coupling projection domain decomposition method and Kansa's method in electrostatic problems. *Comput. Phys. Commun*., **180**, pp. 200–214, 2009.

[17] Beatson, R.K., Light, W.A. & Billings, S., Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM J. Sci. Comput*., **22**, pp. 1717–1740, 2006.

[18] Hernandez-Rosales, A. & Power, H., Non-overlapping domain decomposition algorithm for the Hermite radial basis function meshless collocation approach:

applications to convection diffusion problems. *J. Algo. & Comput. Tech*., **1**, pp. 127–159, 2007.

[19] Cavoretto, R., De Rossi, A., Donatelli, M. & Serra-Capizzano, S., Spectral analysis and preconditioning techniques for radial basis function collocation matrices. *Numer. Lin. Alg. Appl.*, **19**, pp. 31–52, 2012.

[20] Galperin, E.A. & Kansa, E.J., Application of global optimization and radial basis functions to numerical solutions of weakly singular Volterra integral equations. *Comput. Math. Appl*., **43**(3–5), pp. 491–499, 2002.

[21] Skalna, I., A global optimization method for solving parametric linear systems whose input data are rational functions of interval parameters, *Lecture Notes in Computer Science*, **6068**, Springer: Berlin, Heidelberg. ISBN: 978-3-642-14402-8.

[22] Mafteiu-Scai, L.O. & Mafteiu-Scai, E.J., Solving linear systems of equations using a memetic algorithm. *Int. J. Comput. Applic.,* **58**, pp. 16–22, 2012.

[23] Taheri, S. & Mammadov, M., Solving systems of nonlinear equations using a globally convergent optimization algorithm. *Global J. Tech. & Optim.,* **3**, pp. 132–138, 2012.

[24] Kaeck, J.E., Constrained global optimization with radial basis functions. Research Report MdH-IMa-2004, Department of Mathematics and Physics, Malardalen University, Sweden.

[25] Bjorkman, M. & Holmstroem, K., Global optimization of costly nonconvex functions using radial basis functions. *Optim. & Eng*., **1**, pp. 373–397, 2000.

[26] Gutmann, H.M., A radial basis function method for global optimization. *J. Global Optim.,* **19**, pp. 201–227, 2001.

[27] Le Thi, H.A., Vaz, A.I.F. & Vicente, L.N., Optimizing radial basis functions by DC programming and its use in direct search for global derivative-free optimization. *TOP*, **20**, pp. 190–214, 2012.

[28] Jakobsson, P., Patriksson, M., Rudholm, J. & Wojciechowski, A., A method for simulation based optimization using radial basis functions. *Opt. & Eng.,* **11**, pp. 501–532, 2010.

[29] Higham, N.J., Multi-precision world. *SIAM News*, pp. 2–3, Oct. 2017.

[30] The GNU Multiple Precision Arithmetic Library. https://gmplib.org/.