# On Convergent Numerical Algorithms for Unsymmetric Collocation

Cheng-Feng Lee[*], Leevan Ling[†] and Robert Schaback[‡]

January 31, 2008

## Abstract

In this paper, we are interested in some convergent formulations for the unsymmetric collocation method or the so-called Kansa's method. We review some newly developed theories on solvability and convergence. The rates of convergence of these variations of Kansa's method are examined and verified in arbitrary–precision computations. Numerical examples confirm with the theories that the modified Kansa's method converges faster than the interpolant to the solution; that is, exponential convergence for the multiquadric and Gaussian RBFs. Some numerical algorithms are proposed for efficiency and accuracy in practical applications of Kansa's method. In double–precision, even for very large RBF shape parameters, we show that the modified Kansa's method, through a subspace selection using a greedy algorithm, can produce acceptable approximate solutions. A benchmark algorithm is used to verify the optimality of the selection process.

[*]Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung, Taiwan.

[†]Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. (`lling@hkbu.edu.hk`)

[‡]Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen, Lotzestraße 16-18, D-37083 Göttingen, Germany.

# 1 Introduction

Many successful applications of recently developed mesh-free methods can be found in different Mathematics, Physics and Engineering journals; for example, see [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. In this paper, we are interested in the radial basis function (RBF) method for solving partial differential equations (PDE) in strong form.

We consider PDE in the general form of

$$Lu = f, \quad L : \mathcal{U} \to \mathcal{F}, \tag{1}$$

where $L$ is a linear operator between two normed linear spaces $\mathcal{U}$ and $\mathcal{F}$. The PDE (1) can be solved by a sufficiently smooth solution $u^* \in \mathcal{U}$ that generates the data $f := Lu^* \in \mathcal{F}$. Equivalently, we write (1) as an uncountably infinite set of simultaneous scalar equations

$$\lambda[u] = f_\lambda \in \mathbb{R}, \quad \text{for all} \ \ \lambda \in \Lambda. \tag{2}$$

In general, a single set $\Lambda$ of functionals contains several types of differential or boundary operators. Discretization consists of replacing the infinite set $\Lambda$ by some finite unstructured subsets $\Lambda_m := \{\lambda_1, \ldots, \lambda_m\}$. The space spanned by these functionals can be called the *test space*, and $\Lambda$ is the infinite *test set*.

In order to obtain a numerical approximation $u$, we assume that there exists an approximation to $u^*$ in the *trial space* given by the span of $\{u_1, \ldots, u_n\}$ such that

$$u := \sum_{j=1}^{n} \alpha_j u_j \ \in \mathcal{U}_n := \text{span}\{u_1, \ldots, u_n\}. \tag{3}$$

Again, we assume sufficient smoothness in $u^*$. In case of nonsmooth or singular solutions, suitable singular functions should be added to the trial space [11, 12]. The discretized problem then reads as

$$\lambda_i[u] = \sum_{j=1}^{n} \alpha_j \lambda_i[u_j] = f_{\lambda_i} =: \lambda_i(u^*), \ \ 1 \le i \le m, \tag{4}$$

when written as linear equations for a function $u$ of the trial space $\mathcal{U}$. Note that

2

(2)–(4) are general enough for different classical methods and for both strong and weak formulations.

The unsymmetric RBF collocation method was first proposed by Kansa [13, 14]. In the original formulation, the trial and test spaces were closely related. Let $\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a symmetric positive definite kernel and $X_n = (x_1, \ldots, x_n) \subset \mathbb{R}^d$ be a set of *centers*. Usually, these centers are irregularly placed in the domain $\overline{\Omega}$ in which the PDE is defined. The trial function in (3) is given by $u_j := \Phi(\cdot, x_j)$ for $1 \le j \le n$ and the test functional $\lambda_i$ $(1 \le i \le n)$ in (4) is defined by applying the differential (or boundary) operator followed by a point evaluation at $x_i \in X_n$. The resulting unsymmetric collocation matrix has the $ij$-entries

$$\lambda_i[u_j] = \lambda_i^y \Phi(y, x_j), \ 1 \le i, j \le n,$$

where the superscript $y$ of $\lambda_i^y$ indicates the variable of $\Phi$ on which the functional operates. Although the method is relatively easy for implementation, there are some open questions that need to be answered.

From the theoretical point of view, the original formulation has neither error bounds nor convergence proofs because the method may result in singular systems in some specially constructed situations [15]. In order to carry out some mathematical analysis, it is necessary to make further assumptions and modify the formulation. In some of our earlier works [16, 17, 18], Kansa's method was modified in such a way that solvability is guaranteed. Later, we proposed in [19, 20] another variant of the method so that error bounds become possible. The solvability and convergence theorems of the modified formulations are reviewed in Section 2. The first goal of this paper is to numerically compare among a few formulations of the unsymmetric meshless collocation methods. We use the arbitrary–precision computations capability of $Mathematica^{©}$ to numerically verify the proven theory: the numerical solution of our modified unsymmetric collocation method converges faster than the interpolant with respect to the residual norm.

It is well-known that the standard use of shifted radial basis functions forms a terrible basis of a space which is extremely useful for approximation [21, 22, 23]. When built upon the standard RBF basis, ill-conditioning will thus naturally occur for any RBF technique, including Kansa's method [24, 25]. For practical purposes,

appropriate implementations of Kansa's method must therefore be able to circumvent this and provide reasonably stable approximations of the PDE solutions. Using a *greedy algorithm* [17], our algorithms begin with a subspace selection that adaptively selects a trial space that keeps the condition number moderate and prevents numerical breakdown. The greedy algorithm is highly efficient because its column-selection process is carried out by a fast formula. In order to study the optimality of this selection process, we couple the subspace selection technique with the idea of *effective condition number*. Despite of its high complexity, this coupling is used as benchmark for the sake of comparison. Details are given in Section 3.

After identifying a suitable trial space (or RBF centers), the next phase is to solve the resultant system based on the modified Kansa's formulations. In Section 4, two approaches are considered:

- *Linear optimization* has a simple and solid mathematical foundation in cases where the maximum principle holds, but is more complicated to implement.

- *Least squares optimization* is much easier to implement, but it has a much more complicated mathematical background [26, 27].

Lastly, some numerical examples solved by *Matlab*$^{©}$ are given to conclude the paper.

## 2  Solvability and convergence

We overview the solvability and convergence results of the modified Kansa's formulation. Readers are referred to the original articles [17, 18, 19, 20] and an extension [27] to weak problems for details. As mentioned before, Kansa's original formulation may fail in certain cases even though it is unlikely to happen. The method is widely used by many researchers. The following theorem addresses that solvability is guaranteed if the trial functions or equivalently the RBF centers are correctly chosen.

**Theorem 1** (see [17])**.** *Assume the kernel $\Phi$ to be smooth enough to guarantee that the functions $u_\lambda := \lambda^y \Phi(y, \cdot)$ for $\lambda \in \Lambda$ are continuous. Furthermore, let the*

*m functionals $\lambda_1, \ldots, \lambda_m$ of $\Lambda_m$ be linearly independent over $\mathcal{U}$. Then the set of functions $\{u_\lambda\}$ for $\lambda \in \Lambda_m$ constructed above is linearly independent, and hence the unsymmetric collocation matrix is nonsingular for properly chosen trial centers.*

Suppose (2) is well-posed in a Hilbert space $\mathcal{H}$. Assume that

$$\|u\|_\Lambda := \sup_{\lambda \in \Lambda} |\lambda(u)| \leq \|u\|_\mathcal{U} \quad \text{for all} \quad u \in \mathcal{U} \tag{5}$$

is a norm on $\mathcal{U}$ and is weaker than the norm in $\mathcal{U}$. Moreover, let $\mathcal{U}_\epsilon$ be a subspace of $\mathcal{U}$ such that for all $u \in \mathcal{U}$ there is some approximation $v_{u,\epsilon} \in \mathcal{U}_\epsilon$ with

$$\|u - v_{u,\epsilon}\|_\Lambda \leq \epsilon \|u\|_\mathcal{U} \quad \text{for all} \quad u \in \mathcal{U} \tag{6}$$

for any small number $\epsilon > 0$. We would like to construct a function $v_\epsilon \in \mathcal{U}_\epsilon$ that solves

$$v_\epsilon = \arg \min_{v \in \mathcal{U}_\epsilon} \|v - u^*\|_\Lambda.$$

We do not know whether the minimum is attained and how it can be obtained, but due to (6) we can assume that there is a function $v_\epsilon^* \in \mathcal{U}_\epsilon$ with

$$\|v_\epsilon^* - u^*\|_\Lambda \leq 2\|v_{u^*,\epsilon} - u^*\|_\Lambda \leq 2\epsilon \|u^*\|_\mathcal{U} \tag{7}$$

which is sufficient for our purpose.

**Theorem 2.** *[18, 19, 20] Let $\mathcal{U}$ be a normed linear space with norm $\|.\|_\mathcal{U}$, dual space $\mathcal{U}^*$ and dual unit sphere $\mathcal{U}_1^* := \{\lambda \in \mathcal{U}^* \;:\; \|\lambda\|_{\mathcal{U}^*} = 1\}$. Let a test set $\Lambda \subset \mathcal{U}_1^*$ be given such that $\|.\|_\Lambda$ is defined on $\mathcal{U}$ with (5). Assume further that the general interpolation problem (2) is well-posed. Let $\{\mathcal{U}_\epsilon\}_\epsilon$ be a scale of subspaces of $\mathcal{U}$ for $\epsilon \to 0$ such that for all $u \in \mathcal{U}$ there is a $v_{u,\epsilon} \in \mathcal{U}_\epsilon$ with (6). For all $\epsilon \to 0$, take a function $v_\epsilon^*$ satisfying (7). Then the error bound (7) holds and there is convergence $\|v_\epsilon^* - u^*\|_\Lambda \to 0$.* $\qquad\square$

Trial spaces generated by spans of RBFs with sufficiently dense centers form a sequence of subspaces getting dense in $\mathcal{U}$. We know the RBF interpolant satisfies (6) for all $\epsilon > 0$. Hence, the convergence rate of the unsymmetric collocation

method is faster than that of the interpolant in the trial space $\mathcal{U}_\epsilon$ to the exact solution in $\mathcal{U}$ with respect to the norm $\|.\|_\Lambda$.

In the next section, we will put all these abstract ideas into numerical algorithms for double–precision computations.

# 3   Selecting Trial Functions Adaptively

This section is devoted to numerical algorithms for applying the modified Kansa's formulation under double–precision. We gave answers to the questions of solvability and convergence in Theorem 1 and Theorem 2, respectively. According to these theorems, the selections of trial and test spaces for the unsymmetric collocation method need special attention from both theoretical and numerical points of view. First, the trial and test spaces are no longer generated by a single set of centers. Consider some discretizations $\Lambda_m \in \Lambda$ and $\mathcal{U}_n \in \mathcal{U}$ with $m < n$. Theorem 1 guarantees that the $m \times n$ full matrix is of full rank for sufficiently large $m$. Hence, there must exist a nonsingular $m \times m$ submatrix within the full matrix that allows us to perform the unsymmetric collocation method.

As mentioned in the introduction, the unknown solution is expanded in terms of an unstable basis in Kansa's method. To achieve some stabilization, we rely on a subspace selection process to choose a set of suitable basis functions for well-conditioned computations in the sense of the condition number of the resultant Kansa's matrix. In [17], an adaptive algorithm is proposed to *properly* select a trial space. The algorithm is called *matrix–free* since the a–priori evaluation of the full $m \times n$ matrix is unnecessary. Instead, the algorithm builds small $k \times k$ matrices starting with $k = 1, 2, 3, \ldots$ iteratively which are updated by calculations of complexity $\mathcal{O}(k^2)$ and it usually stops at rather small values of $k$. If the algorithm runs up to a maximal $\hat{n}$, the total storage requirement is $\mathcal{O}(\hat{n}(n + m + \hat{n}))$ instead of $m \cdot n$ for a full-size matrix algorithm. Its computational complexity is $\mathcal{O}(\hat{n}^2(n + m + \hat{n}))$. Note that if trial functions and test functionals are not chosen adaptively and if $m$ and $n$ are moderate in size, a QR-factorization can be employed with some trade-off of efficiency.

6

## 3.1 Adaptive Greedy algorithm

Here is the numerical procedure of the adaptive greedy algorithm. After the $k$-th iteration, $k$ test functionals $\lambda_1, \ldots, \lambda_k$ (i.e. collocation points) and $k$ trial centers $x_1, \ldots, x_k$ (i.e. RBF centers in case of a Kansa's method) are chosen, leading to trial functions $u_i := \Phi(\cdot, x_i)$, $1 \leq i \leq k$. Instead of a huge unsymmetric $m \times n$ system of unknown rank, we have a small nonsingular $k \times k$ subsystem

$$A_{\Lambda_k, \mathcal{U}_k} \, \alpha^k = f_{\Lambda_k},$$

corresponding to an approximate solution

$$s_k := \sum_{i=1}^{k} \alpha_i^k \, u_i.$$

The adaptive algorithm automatically searches for a suitable new test/trial pair $(\lambda_{k+1}, x_{k+1})$ at each iteration, consisting of a test functional $\lambda_{k+1}$ and a trial center $x_{k+1}$.

First, we pick from $\Lambda_m$ a functional $\lambda_{k+1}$ such that the residual $|\lambda_{k+1}(s_k - u)|$ is the largest. At the same time, this process selects a new row and we define

$$\Lambda_{k+1} := \Lambda_k \cup \{\lambda_{k+1}\}.$$

Next, we select a column in order to complete the $(k+1)$-st selection process. If we consider the new column as a free and variable function $w \in \mathcal{U}_n$, then the determinant of $A_{\Lambda_{k+1}, \mathcal{U}_{k+1}}$ is a function $v(w)$. A fast computation formula for computing such determinants can be found in [17], namely

$$v(w) := |\det(A_{\Lambda_{k+1}, \mathcal{U}_k \cup \{w\}})| = |\det(A_{\Lambda_k, \mathcal{U}_k})| \cdot \left| \sum_{i=1}^{k} \gamma_i \lambda_i w + (-1)^{k+1} \lambda_{k+1} w \right|, \quad (8)$$

where $\gamma_1, \ldots, \gamma_k$ are independent of $w$ and are given by the solution of

$$A_{\Lambda_k, \mathcal{U}_k}^T \gamma = (\lambda_{k+1} u_1, \ldots, \lambda_{k+1} u_k)^T.$$

This is easy to calculate since the algorithm keeps $A^{-1}_{\Lambda_k, \mathcal{U}_k}$ in storage and updates it for increasing $k$. Then, the $(k+1)$-st column is picked such that the new determinant is closest to 1. We iterate until one of the stopping criteria is met.

Figure 1 and Figure 2 show the selections of the greedy algorithm for a Laplace equation in [17]. All settings are identical except the values of $c$ are different: $c = 1$ and $c = 5$, respectively, in these figures. Trial centers placed in $[-6, 6]^2$ are input to the greedy algorithm for the subspace selection process. The underneath coefficient matrix for $c = 1$ is relatively well-conditioned compared to that for $c = 5$. Hence, we observe that there are more trial centers in Figure 1 than in Figure 2. This reflects the fact that double precision computations cannot handle as many trial centers when $c$ increases. Since the far-field of the multiquadric RBF is behaving like a linear function, it is not preferred by the greedy algorithm. Note that trial centers closer to the domain are more likely to be selected for smaller values of $c$.

## 3.2 Benchmark algorithm

The greedy algorithm relies on the determinant function to locate the "best" RBF center in each iteration. However, the determinant function can be misleading. Consider

$$A = \begin{pmatrix} \delta & 0 \\ \delta & \delta^{-1} \end{pmatrix}.$$

The determinant of $A$ is 1 for all $\delta$ while the condition number is

$$\kappa(A) = \frac{\sigma_1}{\sigma_n} = \frac{1}{2\delta^2}\left(1 + 2\delta^4 + \sqrt{4\delta^8 + 1}\right),$$

where $\sigma_1$ and $\sigma_n$ are the maximal and the minimal singular values of $A$, respectively. Note that $\kappa(A) \to \infty$ as $\delta \to \infty$. Moreover, if we solve a system $A\alpha = b$ with a fixed $b$, the condition number may not be a good estimation either. The condition number is used to provide a bound of relative errors from the perturbation of the matrix $A$ and all vectors $b$. The bound is tight only for the worst situation.

The *effective condition number* was introduced and studied in Chan and Foulser [28]. Subsequently, it was applied to boundary value problems in Christiansen and Hansen [29], and to the boundary integral equation in Christiansen and Saranen
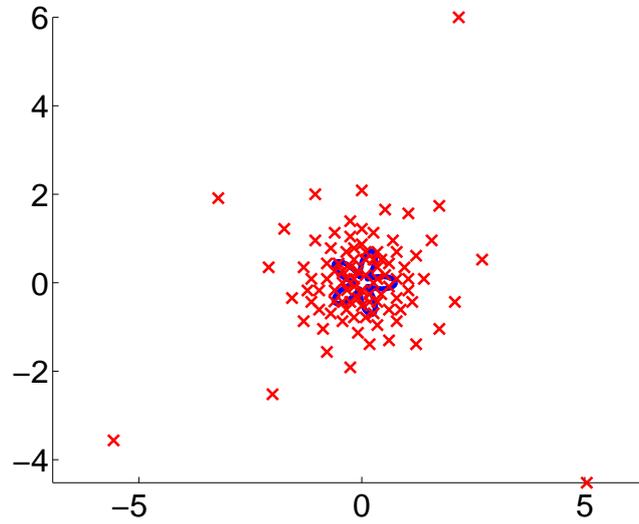
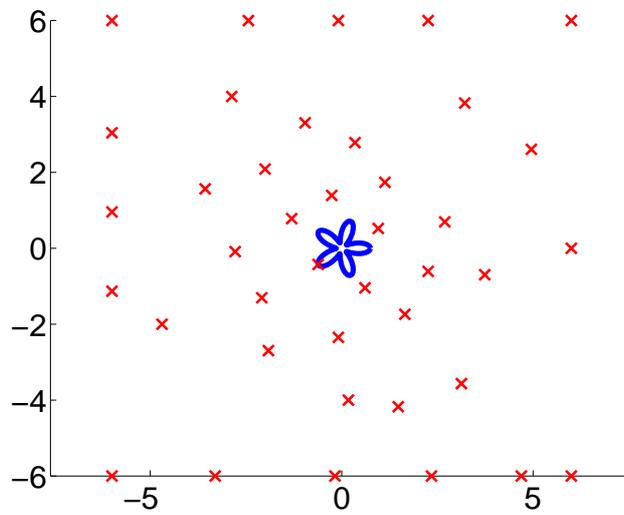**Figure 1:** *Loci of trial centers chosen by the greedy algorithm for $c = 1$.*



**Figure 2:** *Loci of trial centers chosen by the greedy algorithm for $c = 5$.*

[30]. Other kinds of effective condition numbers are given in Banoczi et al. [31], where they are applied to the Gaussian elimination and the QR factorization. More recent references for the effective condition number can be found in [32, 33]. Consider a full-rank matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ with SVD decomposition $A = U\Sigma V^*$. With $\beta = U^*b$ the effective condition number for $A\alpha = b$ is given by

$$\kappa_{\text{eff}}(A, b) := \|b\| \left/ \sigma_n \sqrt{\left(\frac{\beta_1}{\sigma_1}\right)^2 + \ldots + \left(\frac{\beta_n}{\sigma_n}\right)^2} \right. \leq \kappa(A) \text{ for all } b \in \mathbb{R}^n.$$

It is possible to make the column selections dependent on the right-hand vectors by replacing the determinant function (8) with the effective condition number (as a function of $w \in \mathcal{U}_n$)

$$\widetilde{v}(w) := \left| \kappa_{\text{eff}}(A_{\Lambda_{k+1}, \mathcal{U}_k \cup \{w\}}, b_{k+1}) \right|.$$

Within each iteration of the adaptive algorithm, the vector $b_{k+1} := \{\lambda_i(s_k - u)\}_{i=1}^{k+1}$ remains fixed for all $w \in \mathcal{U}_n$. Hence, $\widetilde{v}(w)$ only takes one input argument.

Such modification is persuasive, but the drawback is the lack of a fast formula. At the $k$-th iteration, we need $n$ computations of SVDs with matrices of size $(k + 1)$ by $(k + 1)$. For our numerical examples in Section 5, we treat this $\kappa_{\text{eff}}$-based algorithm as a *benchmark* and we keep using the determinant function as our column-selection criteria which is both efficient and close-to-benchmark.

## 4 Global Testing

Besides of solvability in theory, we showed how the adaptive greedy algorithm selects proper subspaces of the full trial space to guarantee well-condition in double–precision subsystems for practical computations. In this section, we take care of the global error and the ways of obtaining the unknown RBF coefficients $\alpha$. Suppose $m$ test functionals and $n$ trial centers are input to the greedy algorithm. We assume that the greedy algorithm terminates after $\hat{n}$ steps with $\hat{n} \leq \min(m, n)$. This results in $\hat{n}$ test functionals and $\hat{n}$ trial centers being selected.

The next step is to obtain a numerical solution based on the selected $\hat{n}$–

dimensional trial space $\mathcal{U}_{\hat{n}} = \mathrm{span}\{u_1, \ldots, u_{\hat{n}}\}$ or RBF centers $X_{\hat{n}}$. The original algorithm in [17] performs an exact interpolation by solving the $\hat{n} \times \hat{n}$ subsystem. However, this simple computational technique does not have a theoretical foundation or a convergence analysis. To apply the convergent modified Kansa's method, we are only interested in the $\hat{n}$ trial centers.

Theorem 2 gives an error estimation based on the infinite dimensional test space $\Lambda$. Following the same idea as in [19], we get a similar estimation on the discretized finite dimensional test space $\Lambda_{\hat{m}}$ ($\hat{m} \geq m \geq \hat{n}$). Consider $\hat{m} = \hat{m}(X_{\hat{n}})$ collocation conditions, depending also on the selected RBF centers $X_{\hat{n}}$, whose density is dense enough for the convergence theory. In particular, we required that a bound on the PDE residual to hold,

$$\|v_{u^*, \epsilon} - u^*\|_\Lambda \leq 2\|v_{u^*, \epsilon} - u^*\|_{\Lambda_{\hat{m}}}.$$

After discretizing $\Lambda$ by $\Lambda_{\hat{m}}$, we rewrite the problem on $\Lambda_{\hat{m}}$ and this has become a standard finite–dimensional linear optimization problem. Its convergence rate is only off by a constant factor from the solution of the infinite problem on $\Lambda$. Such discretization does not affect the convergence analysis. An adaptive algorithm for the linear optimization process can be found in [19].

We are required to solve an $\hat{m} \times \hat{n}$ overdetermined system by using linear optimization. The convergence rates of the interpolant in the $\mathcal{U}_{\hat{n}}$ spaces carry over to convergence rates of the PDE solutions, precisely as in the FEM case. However, linear optimization is computationally much more difficult than linear system solving.

It is well-known that the least squares optimization is linear and numerically efficient. We prefer to use a least–squares solver instead of linear optimization. Furthermore, we can obtain an iterative scheme by iterating the greedy algorithm on the residual function to select a larger set of trial functions. Lastly, we solve the overdetermined system with all selected trial functions by a full least squares optimization. Our proposed iterative least squares scheme, in fact, has very good efficiency and accuracy. In the next section, we demonstrate some promising close-to-benchmark results in double–precision computations.

# 5 Numerical examples

## 5.1 Examples with arbitrary-precision

First of all, we demonstrate some convergence results of the modified Kansa's formulations in arbitrary–precision computation. Here, RBF interpolations are tested against RBF-PDE with the original Kansa's method. The test problem is a two-dimensional Poisson equation found in [34, 35]. The exact solution is given as

$$u^*(x, y) = \sin\frac{\pi x}{6} \sin\frac{7\pi x}{4} \cos\frac{3\pi y}{4} \cos\frac{5\pi y}{4}.$$

Data points are uniformly spaced with mesh norm $h$ between $\frac{1}{5}$ to $\frac{1}{25}$. For each tested $h$, trial spaces are identical for all methods.

In Figure 3, all multiquadric shape parameters are fixed at $c = 1$. All tested formulations demonstrate exponential rate of convergence. As $h$ decreases, the accuracies among all methods become similar. Although the original Kansa's method may appear to be superior over our algorithm for large $h$, its convergence rate seems to be slower than the others as $h \to 0$.

To see a clear demonstration of the superior convergence rates of the modified Kansa's methods, the shape parameters are chosen to be the PDE–optimal ones $c = c^*(h)$ reported in [35] in the second example. We again observe that the modified Kansa's formulations PDE-LS and PDE-LO show faster convergence rates as $h \to 0$ in Figure 4. As $h$ decreases, $c^*(h)$ increases from 2.074 to 6.637. This is a situation to which our theories cannot be currently applied, because we fix the scale of our radial basis function $\Phi$ throughout. Furthermore, it is very unusual to increase the scaling $c$ with decreasing $h$, because as the number of centers increases, one should reduce the value of $c$ in order to circumvent the problem of ill-conditioning in double–precision computations. However, in [35], the authors argue that by using high-precision computation with less centers, computation time can be saved enormously. This is perfectly in line with the findings of several authors [36, 37, 38] concerning the "flat limit" $c \to \infty$: interpolants using analytic radial basis functions will often converge towards multivariate polynomial interpolants with good approximation properties. However, a thorough error analysis of the "flat limit" is missing. We leave this as an open question for future study.
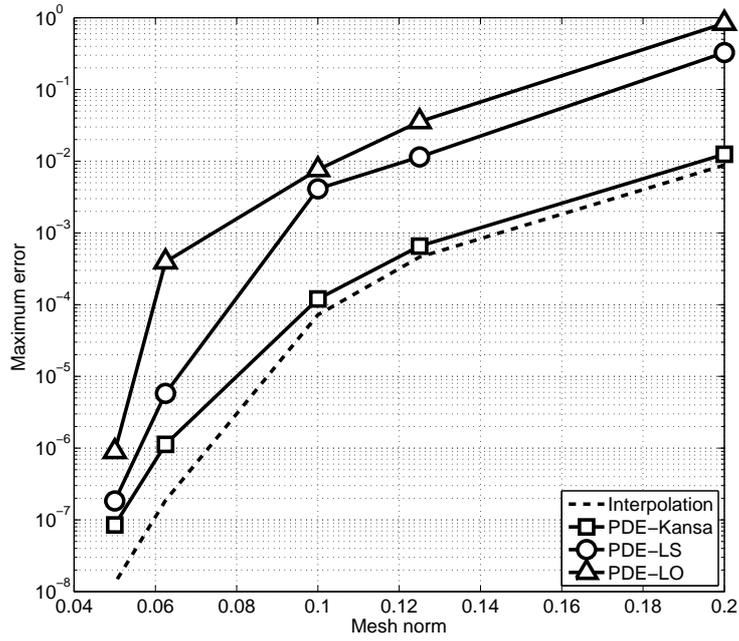
**Figure 3:** *Arbitrary–precision computations: Error profiles for various un-symmetric meshless formulations with a fixed shape parameter $c = 1$.*
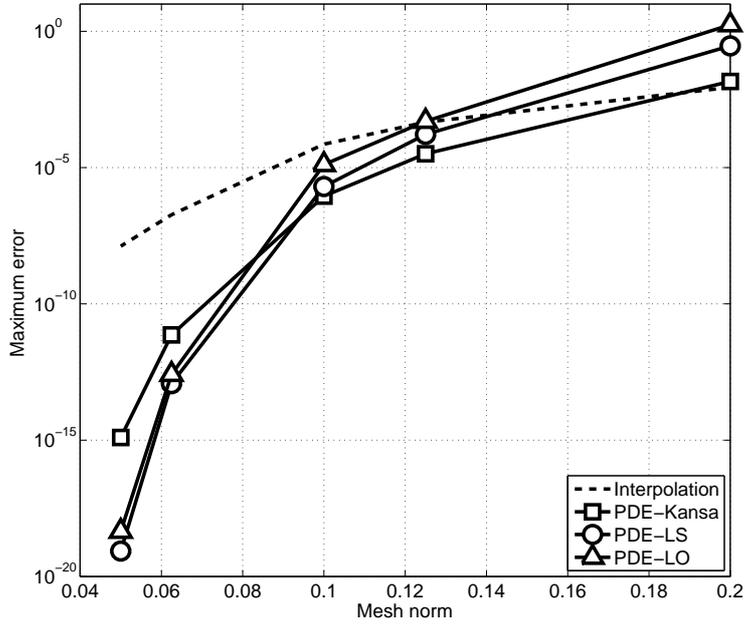


**Figure 4:** *Arbitrary–precision computations: Error profiles for various un-symmetric meshless formulations with the optimal shape parameter $c^*$.*

## 5.2   Examples with double-precision

In this section, we show some numerical examples that demonstrate the efficiency and stability of our proposed algorithms in double–precision. In all examples, we have used the multiquadric kernel

$$\Phi_c(x,y) = \sqrt{1 + \frac{||x-y||^2}{c^2}},$$

where $x, y \in \mathbb{R}^2$ and $c > 0$. As a test equation, we solve the Poisson problem with Dirichlet boundary conditions on $\Omega = [-1,1]^2$, i.e.

$$\begin{aligned}
\triangle u(x) = f(x) & \quad \text{for } x \in \Omega \subset \mathbb{R}^d, \\
u(x) = g(x) & \quad \text{for } x \in \partial\Omega.
\end{aligned} \tag{9}$$

The functions $f$ and $g$ in (9) are generated by the exact solution

$$u^*(x,y) = \frac{1}{2}\log\left((x-2)^2 + (y-2)^2\right).$$

All computations are carried out in *Matlab*$^©$ with double–precision. Under the assumption that the solution is sufficiently smooth, we allow RBF centers to located outside of $\Omega$ by placing $n$ trial centers in $[-4,4]^2$. The numbers of input test functionals are $m \approx \frac{1}{5}n$. The tested $n$ ranges from 643 to 53351. Reported errors are measured by the standard root-mean-square (RMS). Five algorithms are tested:

**Original greedy (OG)** is the algorithm proposed in [17]. Once the greedy algorithm terminates after $\hat{n}$ steps, an $\hat{n} \times \hat{n}$ matrix system is solved in order to obtain the desired numerical solution.

**Linear optimization (LO)** is the algorithm proposed in [19]. Instead of solving a square subsystem, this algorithm will take all test functionals (i.e. $\hat{m} = m$) into account and solve an overdetermined system for the desired numerical solution. By the RBF interpolation theory [39], this algorithm enjoys exponential convergence for certain class of RBF.

**LS–1 loop (LS1)** is a simplified version of LO. We simply replace the LO solver

by the least squares solver for the overdetermined system using $Matlab^{©}$ built-in *backslash* function (e.g. QR-decomposition). Despite the lack of a convergence theory for strong problems, this algorithm is completely linear and hence efficient. Note that [27] allows discrete least–squares solvers for weak problems, and chances are good that the results carry over to strong cases.

**LS–2 loops (LS2)** applies the greedy algorithm to the residual function of LS1 in order to collect a larger set trial centers. This often results in a small increment of the number of trial centers. However, the extra trial centers could greatly improve the accuracy upon LS1.

**LS–3 loops (LS3)** follows the idea of LS2 and applies one more iteration based on the residual function of LS2.

**Effective (Eff)** is based on the greedy algorithm, but replaces the determinant function by the effective condition number function in order to locate the best trial center for each iteration. We have a strong motivation for making such a modification, but we have not developed a fast formulation to make it practical yet. We treat this as the benchmark algorithm for accuracy and stability.

In Figure 5, the error profiles of different algorithms against various shape parameters $c$ are shown. We tested up to $c = 10$ that is a considerably large value in double-precision computations. The numbers of selected trial centers $\hat{n}$ are shown in Figure 6 for each method. Since OG, LO, and LS1 are built upon the output of the greedy algorithm, their corresponding $\hat{n}$ are identical.

Note, from Figure 5, that the OG algorithm proposed in [17] is somehow unstable. For certain values of $c$, the RMS error of the numerical solution could be much larger comparing to the nearby values. Next, the exactly determined system in OG is replaced by some overdetermined approximations. We observe that the LO has slightly better accuracy and gives a more stable error profile. After we replace the linear optimization by the least square minimization, LS1 shows a significant improvement in accuracy comparing to the two previously proposed methods. However, if we compare the LS1 with the benchmark curve Eff, the Eff
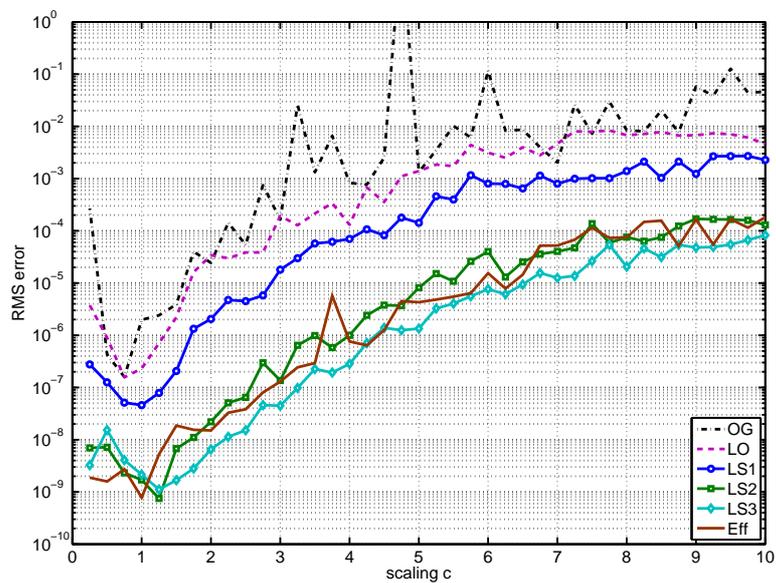
**Figure 5:** *Double–precision computations: Error profiles of different greedy algorithms against different shape parameter c.*
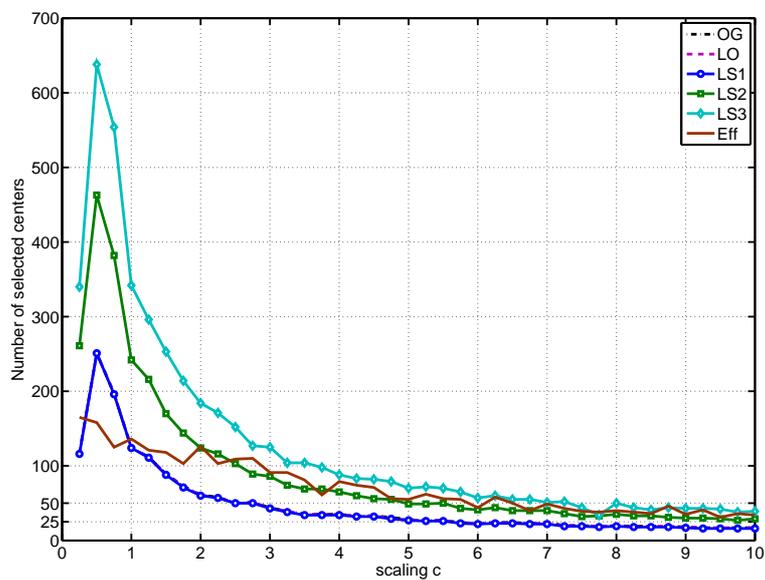


**Figure 6:** *Number of selected trial centers corresponding to Figure 5.*
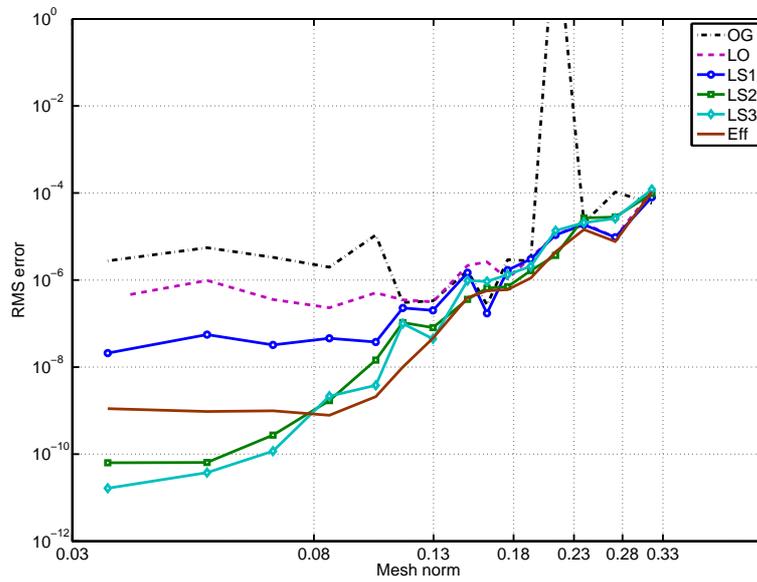
**Figure 7:** *Double–precision computations: Error profiles of different greedy algorithms against the mesh norms of the input trial centers.*
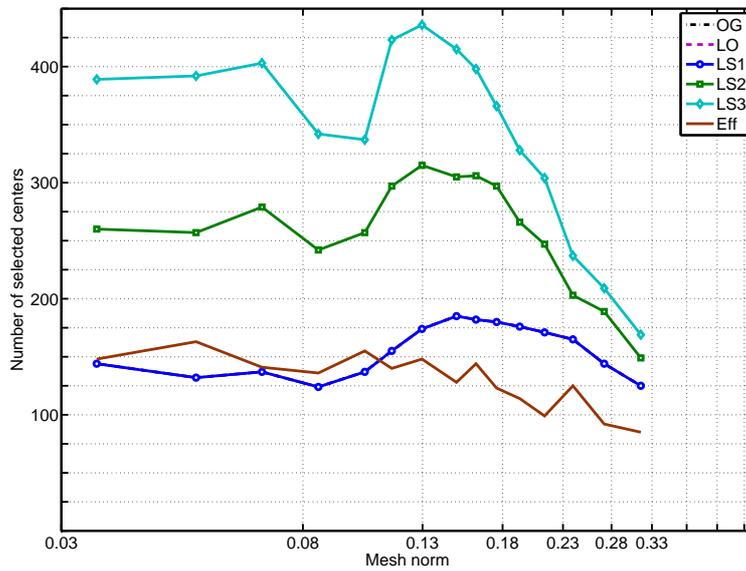


**Figure 8:** *Number of selected trial centers corresponding to Figure 7.*

17

curve shows a rather dramatic improvement in accuracy. In Figure 6, we observe that Eff with $c \approx 1$ are running on a smaller subspace, but has better accuracy. This suggests that the determinant based greedy algorithm is not making the best possible decision in picking trial centers. We overcome this by the iterative approach.

As mentioned in the previous sections, the effective condition number based strategy of trial centers selection is extremely slow. To overcome this problem, we iterate on the residual of LS1 and collect a larger set of trial functions. This gives the curve LS2. In the same way, we have LS3 if we iterate one more time based on the residual of LS2. The three error curves of Eff, LS2 and LS3 are very close to each other. Moreover, the number of selected trial centers are similar in size when $c$ is large. We believe that the iterative least squares approach provides a fast and efficient way to obtain a close-to-benchmark unsymmetric collocation solution.

In Figure 7, the shape parameter is fixed at $c = 1$ and we want to observe the behaviour as the number of *input* trial centers and test functionals increase. The dimensions of the selected subspaces are shown in Figure 8; again, the curves of OG, LO and LS1 are overlapped. In Figure 7, the mesh norm denotes that of the trial centers input to the greedy algorithm. Since the input trial centers are placed in $[-4, 4]^2$, the underneath matrix is of very large-scale. Due to the *ill-conditioning* problem of the underneath matrix system, decreasing the mesh norm of the input trial centers does not imply that of the selected ones. Instead of convergence, Figure 7 should be viewed as a demonstration of numerical stability of our algorithms.

From Figure 7, we observe instability in the curve OG once again. For (input) mesh norms down to around 0.13, all curves show the similar tendencies of convergence. As the density of input trial centers decreases, the algorithms begin to show different performances. Among all single loop algorithms, Eff shows the best accuracy. However, its RMS error appears to stay flat and convergence ceases for mesh norm smaller than 0.08 (with an error around $10^{-9}$). On the other hand, we can clearly observe that LS1 is better than OG and LO in accuracy. Note that the curve LO terminates earlier than the other curves due to the "out-of-memory" problem. In Figure 9, we show the computation times[1] of OG, LS1 and LS2 for

---

[1] *Matlab*© Version R2007a running on a Dell Precision WorkStation 670–Intel(R) Xeon(TM)
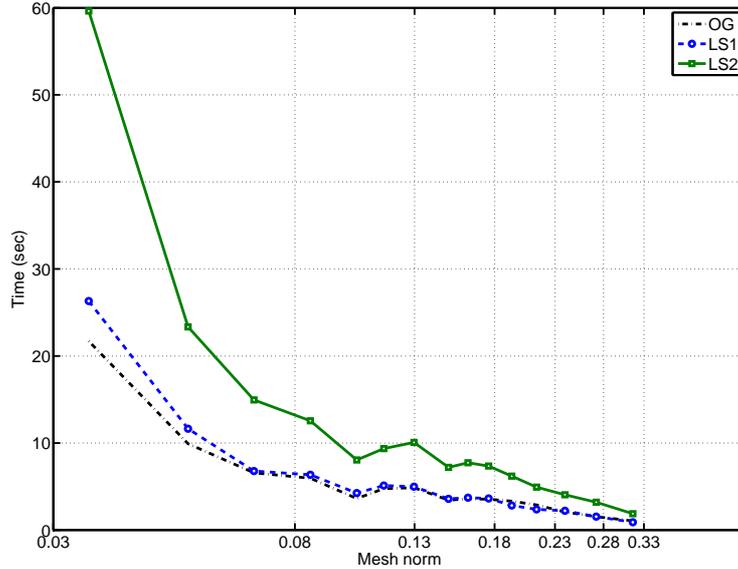
**Figure 9:** *Computation times corresponding to Figure 7.*

completeness.

As we shift our focus to LS2 and LS3, they both show better accuracy than Eff. Moreover, LS2 and LS3 are running on large subspaces in all tested cases in comparison to all single-loop algorithms. Overall, LS3 shows slightly better accuracy than LS2. While LS2 flats out at the last two data points, LS3 remains converging. One reason for the success of LS2 and LS3 is their large number of "good" trial functions. Suppose a fast formulation is available for Eff, it is of no doubt that iterating the effective condition number based greedy algorithm will give us an even better algorithm. We leave this to our future study.

# 6 Conclusions

In this paper, a new class of convergent formulations is proposed for solving PDEs via an unsymmetric radial basis function collocation method. Some results obtained by arbitrary–precision computations are given as a numerical verification

CPU 3.00GHz (2CPUs) with 3.00GB of RAM.

of our theories and new formulations of Kansa's method. For practical purposes, we compare the same formulations by applying some numerical algorithms with subspace selection in double–precision. Our proposed algorithms show significant improvement in different aspects. The trial function selection criteria based on effective condition numbers demonstrates the approximation power of a good greedy algorithm. Numerical efficiency is greatly improved by replacing the linear optimization by least–squares. Despite the lack of a full convergence theory, the least–squares approach demonstrates better accuracy. Furthermore, its corresponding iterative schemes are fast and show accuracies as good as that of the benchmark algorithm based on effective condition numbers.

# References

[1] T. Cecil, J. Qian, S. Osher, Numerical methods for high dimensional hamiltonvjacobi equations using radial basis functions, J. Compt. Phys. 196 (1) (2004) 327–347.

[2] G. E. Fasshauer, Solving differential equations with radial basis functions: Multilevel methods and smoothing., Adv. Comput. Math. 11 (2-3) (1999) 139–159.

[3] R. Keck, D. Hietel, A projection technique for incompressible flow in the meshless finite volume particle method., Adv. Comput. Math. 23 (1-2) (2005) 143–169.

[4] J. Li, Mixed methods for fourth-order elliptic and parabolic problems using radial basis functions., Adv. Comput. Math. 23 (1-2) (2005) 21–30.

[5] G. J. Moridis, E. J. Kansa, The Laplace transform multiquadrics method: a highly accurate scheme for the numerical solution of linear partial differential equations, J. Appl. Sci. Comput. 1 (2) (1994) 375–407.

[6] A. S. M. Wong, Y. C. Hon, T. S. Li, S. L. Chung, E. J. Kansa, Multizone decomposition for simulation of time-dependent problems using the multiquadric scheme, Comput. Math. Appl. 37 (8) (1999) 23–43.

[7] S. Wong, Y. Hon, M. Golberg, Compactly supported radial basis functions for the shallow water equations, Internat. J. Appl. Sci. Comput. 127 (2002) 79–101.

[8] G. B. Wright, B. Fornberg, Scattered node compact finite difference-type formulas generated from radial basis functions, J. Compt. Phys. 212 (1) (2006) 99–123.

[9] D. L. Young, S. C. Jane, C. Y. Lin, C. L. Chiu, K. C. Chen, Solutions of 2D and 3D Stokes laws using multiquadrics method, Eng. Anal. Bound. Elem. 28 (10) (2004) 1233–1243.

[10] X. Zhou, Y. Hon, K. Cheung, A grid-free, nonlinear shallow-water model with moving boundary, Eng. Anal. Bound. Elem. 28 (9) (2004) 1135–1147.

[11] H.-Y. Hu, Z.-C. Li, A. H.-D. Cheng, Radial basis collocation methods for elliptic boundary value problems, Comput. Math. Appl. 50 (1-2) (2005) 289–320.

[12] Z. C. Li, T. T. Lu, Singularities and treatments of elliptic boundary value problems, Math. Comput. Modelling 31 (8-9) (2000) 97–145.

[13] E. J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. I. Surface approximations and partial derivative estimates, Comput. Math. Appl. 19 (8-9) (1990) 127–145.

[14] E. J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations, Comput. Math. Appl. 19 (8-9) (1990) 147–161.

[15] Y. C. Hon, R. Schaback, On unsymmetric collocation by radial basis functions, Appl. Math. Comput. 119 (2-3) (2001) 177–186.

[16] Y. C. Hon, R. Schaback, Solvability of partial differential equations by meshless kernel methods, to appear in Adv. Comput. Math. (2006).

[17] L. Ling, R. Opfer, R. Schaback, Results on meshless collocation techniques, Eng. Anal. Bound. Elem. 30 (4) (2006) 247–253.

[18] L. Ling, R. Schaback, On adaptive unsymmetric meshless collocation, in: S. Atluri, A. Tadeu (Eds.), Proceedings of the 2004 International Conference on Computational & Experimental Engineering and Sciences, Vol. CD-ROM, Advances in Computational & Experimental Engineering & Sciences, Tech Science Press, Forsyth, USA, 2004, p. paper # 270.

[19] L. Ling, R. Schaback, Stable and convergent unsymmetric meshless collocation methods, to appear in *SIAM J. Numer. Anal.* (2007).

[20] R. Schaback, Convergence of unsymmetric kernel-based meshless collocation methods,, SIAM J. Numer. Anal. 45 (1) (2007) 333–351.

[21] T. A. Driscoll, B. Fornberg, Interpolation in the limit of increasingly flat radial basis functions, Comput. Math. Appl. 43 (3-5) (2002) 413–422.

[22] B. Fornberg, T. A. Driscoll, G. Wright, R. Charles, Observations on the behavior of radial basis function approximations near boundaries, Comput. Math. Appl. 43 (3-5) (2002) 473–490.

[23] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, Comput. Math. Appl. 48 (5-6) (2004) 853–867.

[24] L. Ling, E. J. Kansa, A least-squares preconditioner for radial basis functions collocation methods, Adv. Comput. Math. 23 (1-2) (2005) 31–54.

[25] S. Rippa, An algorithm for selecting a good value for the parameter $c$ in radial basis function interpolation., Adv. Comput. Math. 11 (2-3) (1999) 193–210.

[26] R. Schaback, Recovery of functions from weak data using unsymmetric meshless kernel-based methods, to appear in APNUM (2006).

[27] R. Schaback, Unsymmetric meshless methods for operator equations, preprint (2006).

[28] T. F. Chan, D. E. Foulser, Effectively well-conditioned linear systems, SIAM J. Sci. Statist. Comput. 9 (6) (1988) 963–969.

[29] S. Christiansen, P. C. Hansen, The effective condition number applied to error analysis of certain boundary collocation methods, J. Comput. Appl. Math. 54 (1) (1994) 15–36.

[30] S. Christiansen, J. Saranen, The conditioning of some numerical methods for first kind boundary integral equations, J. Comput. Appl. Math. 67 (1) (1996) 43–58.

[31] J. M. Banoczi, N.-C. Chiu, G. E. Cho, I. C. F. Ipsen, The lack of influence of the right-hand side on the accuracy of linear system solution, SIAM J. Sci. Comput. 20 (1) (1998) 203–227.

[32] H. T. Huang, Z. C. Li, Effective condition number and superconvergence of the Trefftz method coupled with high order FEM for singularity problems, Eng. Anal. Bound. Elem. 30 (4) (2006) 270–283.

[33] Z.-C. Li, C.-S. Chien, H.-T. Huang, Effective condition number for finite difference method, J. Comput. Appl. Math. 198 (1) (2007) 208–235.

[34] A. H.-D. Cheng, M. A. Golberg, E. J. Kansa, G. Zammito, Exponential convergence and $h$-$c$ multiquadric collocation method for partial differential equations, Numer. Methods Partial Differential Equations 19 (5) (2003) 571–594.

[35] C.-S. Huang, C.-F. Lee, A.-D. Cheng, Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method, Eng. Anal. Bound. Elem. 31 (7) (2007) 614–623.

[36] T. Driscoll, B. Fornberg, Interpolation in the limit of increasingly flat radial basis functions, Comput. Math. Appl. 43 (2002) 413–422.

[37] E. Larsson, B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, Comput. Math. Appl. 49 (1) (2005) 103–130.

[38] R. Schaback, Multivariate interpolation by polynomials and radial basis functions, Constructive Approximation 21 (2005) 293–317.

[39] W. R. Madych, Miscellaneous error bounds for multiquadric and related interpolators, Comput. Math. Appl. 24 (12) (1992) 121–138.