

STABLE AND CONVERGENT UNSYMMETRIC MESHLESS COLLOCATION METHODS

LEE VAN LING* AND ROBERT SCHABACK†

Abstract. In the theoretical part of this paper, we introduce a simplified proof technique for error bounds and convergence of a variation of E. Kansa’s well-known unsymmetric meshless collocation method. For a numerical implementation of the convergent variation, a previously proposed greedy technique is coupled with linear optimization. This algorithm allows a fully adaptive on-the-fly data-dependent meshless selection of test and trial spaces. The new method satisfies the assumptions of the background theory, and numerical experiments demonstrate its stability.

Key words. Kansa’s method, convergence, error bounds, linear optimization.

AMS subject classifications. 65N12, 65N15, 65N70.

1. Introduction. A general framework for solving PDE problems in strong or weak form by kernel-based meshless methods was outlined in [13]. It writes the PDE problem as uncountably many simultaneous scalar equations

$$\lambda(u) = f_\lambda \in \mathbb{R}, \quad \text{for all } \lambda \in \Lambda. \quad (1.1)$$

We call Λ the *test set*, because applying all functionals λ from Λ to a function u tests whether the function u solves the given problem. If several differential or boundary operators are involved, we simply put everything into a single set Λ of functionals of various types.

Testing can be done in *strong* and *weak* form. For testing in *strong form*, the set Λ consists of infinitely many linear real-valued functionals λ that usually take the form of point evaluations of functions or their derivatives at points inside a domain or on some boundary or interface layer.

Testing in *weak* form uses functionals

$$\lambda_{v,a}(u) := a(u, v)$$

which evaluate bilinear forms $a(.,.)$ after inserting a *test function* v . These bilinear forms may contain derivatives, and they always require an integration.

No matter how testing is done, we call (1.1) a *generalized interpolation problem*, and we assume it to be solvable by a function u^* that generates the data via $f_\lambda := \lambda(u^*)$ for all $\lambda \in \Lambda$.

To make generalized interpolation problems numerically accessible, we have to *discretize* them. Discretization on the *trial* side is done by choosing a finite-dimensional *trial space* \mathcal{U} of functions which can approximate the exact solution u^* well. It will be the space of functions from which the numerical solution u will be constructed, and we assume it to be spanned by a basis $\{u_1, \dots, u_m\}$ as

$$u := \sum_{j=1}^m \alpha_j u_j \in \mathcal{U} := \text{span}\{u_1, \dots, u_m\}. \quad (1.2)$$

* 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.

Discretization on the *test* side just consists in replacing the infinite set Λ by some finite unstructured subset $\Lambda_n := \{\lambda_1, \dots, \lambda_n\}$. The space spanned by these functionals can be called the *test space*.

Then, the discretized problem reads as

$$\lambda_i(u) = \sum_{j=1}^m \alpha_j \lambda_i(u_j) = f_{\lambda_i} = \lambda_i(u^*), \quad 1 \leq i \leq n, \quad (1.3)$$

when written as an unsymmetric system of linear equations for a function u of the trial space \mathcal{U} . In most cases, the system will have $n \geq m$, but it is always a hard problem to study the matrix with entries $\lambda_i(u_j)$, e.g. to assert that it has rank m .

Weak formulations use functionals of the form $\lambda_i(u) := a_i(u, v_i)$ with certain bilinear forms a_i and *test functions* v_1, \dots, v_n such that the discretized problem takes the familiar form

$$\lambda_i(u) = a_i(u, v_i) = \sum_{j=1}^m \alpha_j a_i(u_j, v_i) = f_{\lambda_i} = \lambda_i(u^*), \quad 1 \leq i \leq n,$$

of meshless Petrov–Galerkin schemes [1]. If trial functions u_j and test functions v_i coincide, and if there is only a single bilinear form involved, this reduces to the well-known Galerkin method. It has a strong built-in connection of test functionals to trial functions and yields a linear system with a positive semidefinite symmetric Gramian matrix. This makes it easy to handle, and it forms the core of the Finite Element Method and some of its generalizations. However, for unsymmetric weak Petrov–Galerkin schemes, the theoretical and numerical analysis is rather hard [15].

To make problems in strong formulation symmetric, the connection between test functionals and trial functions must be established differently. To get a truly meshless technique and to allow very general problems, the connection is made via a symmetric positive definite *kernel* $\Phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ if the underlying problem is posed for functions of d variables on a domain $\Omega \subset \mathbb{R}^d$. It takes the discretized set Λ_m of test functionals and defines the trial functions in (1.2) as $u_j := \lambda_j^y \Phi(\cdot, y)$ for $1 \leq j \leq m$ where the superscript of λ indicates the variable of Φ on which the functional operates. Then the collocation matrix (1.3) is symmetric with entries $\lambda_i(u_j) = \lambda_i^x \lambda_j^y \Phi(x, y)$ for $1 \leq i, j \leq n$. This *symmetric collocation* technique dates back to [17] and has a solid mathematical basis ([3, 4]). Like in the standard (non–Petrov) Galerkin scheme, the trial and test functions or functionals are closely related in order to maintain symmetry.

But this paper will deal with *unsymmetric meshless collocation*. Following an early idea of Kansa [7, 8] used by many authors afterwards (see an overview in [5]), one takes a set $X_m := \{x_1, \dots, x_m\} \subset \mathbb{R}^d$ of m scattered points and uses a kernel Φ to define the trial space \mathcal{U} spanned by the trial functions

$$u_j = \Phi(\cdot, x_j) = \lambda_{\delta_{x_j}}^y \Phi(\cdot, y), \quad 1 \leq j \leq m \quad (1.4)$$

associated to simple point evaluation functionals $\lambda_{\delta_{x_j}}$. Often, but not necessarily, the points x_j are irregularly placed within Ω . Since they determine the trial functions in (1.4), we can call them *trial centers*. The resulting unsymmetric collocation matrix has the entries

$$\lambda_i(u_j) = \lambda_i^y \Phi(y, x_j), \quad 1 \leq i \leq n, \quad 1 \leq j \leq m$$

and can be singular in exceptional cases [6]. Consequently, there are very limited mathematical results on this technique, though it gives very good results in plenty of applications in science and engineering.

To overcome these solvability problems, one has to modify the setting. To get solvability and error bounds, there should at least be a unique solution to the modified discretized system that converges to the true solution if the discretization is refined. The first question requires that if $n \gg m$ test functionals are fixed and are linearly independent, the system has rank m , if the trial functions are chosen properly. This was proven in [11], while the earlier paper [13] contained an asymptotic analysis for sufficiently many test functionals and trial functions. By a rather complicated and abstract machinery, a fairly general theory leading to error bounds and convergence results for the unsymmetric meshless collocation method was finally provided in [14].

This paper is independent of the theory in [14] and will prove a much more elementary convergence result for a variation of meshless unsymmetric collocation. Furthermore, since [14] does not focus on numerical techniques, we show how certain algorithms allow to satisfy the requirements needed for convergence, both in the context of this paper and [14]. In particular, we combine an adaptive greedy method with linear programming, and some numerical examples show that our new adaptive on-the-fly algorithm works stably in spite of large condition numbers.

2. Well-Posed Problems. We now go back to the infinite problem (1.1) and proceed to define a variation of the unsymmetric collocation technique that will have a convergence proof. To this end, we have to fix the mathematical background.

The problem (1.1) is posed in a Hilbert space \mathcal{U} via a test set $\Lambda \subset \mathcal{U}^*$ of continuous linear functionals on \mathcal{U} . The functionals can clearly be normalized to satisfy $\|\lambda\|_{\mathcal{U}^*} = 1$ for all $\lambda \in \Lambda$. The specific solution to our problem is called u^* from now on. This way, we assume solvability within the Hilbert space \mathcal{U} , even if the space \mathcal{U} is smaller than the “existence space”, i.e. the space in which minimal assumptions guarantee existence of a solution by mathematical analysis. For instance, a Poisson problem on a domain $\Omega \subset \mathbb{R}^d$ can be stated in Sobolev “existence” space $W_2^1(\Omega)$ in weak formulation, but we can also take a Hilbert space \mathcal{U} of at least twice differentiable functions and pose the Poisson problem there in strong form, using the data from a smooth function $u^* \in \mathcal{U}$. Note that standard regularity theory [10] allows this, if the data and the domain are smooth. The above general framework allows both settings, if applied in weak form, since the given data always are of the form $\lambda(u^*)$, $\lambda \in \Lambda$ for a specific function $u^* \in \mathcal{U}$. But here we want to stick to problems posed in strong form, and we assume sufficient additional regularity.

Furthermore, we need that the problem is *well-posed*, and this is expressed by the requirement that

$$\|u\|_{\Lambda} := \sup_{\lambda \in \Lambda} |\lambda(u)| \quad (2.1)$$

is a norm on \mathcal{U} . This definition will always provide a seminorm without further assumptions, but we get a norm if we can prove unique solvability by additional analytic arguments like a maximum principle. Note that this norm will be weaker than the norm in \mathcal{U} because of

$$\|u\|_{\Lambda} \leq \|u\|_{\mathcal{U}} \quad \text{for all } u \in \mathcal{U} \quad (2.2)$$

following from

$$|\lambda(u)| \leq \|\lambda\|_{\mathcal{U}^*} \|u\|_{\mathcal{U}} = \|u\|_{\mathcal{U}} \quad \text{for all } u \in \mathcal{U}, \lambda \in \Lambda$$

and normalization of the test functionals. The well-posedness of the problem is often alternatively expressed in the form

$$\|u\|_A \leq C_A \|u\|_\Lambda \leq C_A \|u\|_{\mathcal{U}} \quad \text{for all } u \in \mathcal{U}, \quad (2.3)$$

where \mathcal{U}_A with norm $\|\cdot\|_A$ is a data-independent function space like \mathcal{U} , but with a weaker norm. This also means that \mathcal{U} is continuously imbedded in \mathcal{U}_A . The upshot here is that the space \mathcal{U}_A and its norm should not be dependent on the test set Λ anymore. We call (2.3) the basic *analytic a-priori inequality* for well-posedness of the problem in the space \mathcal{U}_A . The condition (2.3) looks rather abstract, but it can be explained by

EXAMPLE 2.1. *Let a Poisson problem in 2D be given in strong form as*

$$\begin{aligned} \Delta u &= f && \text{in } \Omega \\ u &= \varphi && \text{on } \partial\Omega \end{aligned}$$

with f continuous in $\overline{\Omega}$ and g continuous on $\partial\Omega$. Furthermore, let Ω be a bounded domain in \mathbb{R}^2 . The domain should allow the standard maximum principle and admit the standard a-priori inequality

$$\|u\|_{W^2(\Omega)} \leq C_A \|\Delta u\|_{L_2(\Omega)}$$

for all u in $W_0^2(\Omega)$ (see [10, p. 66, (6.8)]). Since we are in 2D, there is a Sobolev embedding inequality

$$\|u\|_{\infty, \Omega} \leq C_S \|u\|_{W^2(\Omega)}$$

for all $u \in W^2(\Omega)$. We pose a strong problem by taking the test set Λ to consist of all functionals

$$\begin{aligned} u &\mapsto (\Delta u)(x), && x \in \overline{\Omega} \\ u &\mapsto u(y), && y \in \partial\Omega. \end{aligned}$$

We consider the space $\mathcal{U} = C^2(\overline{\Omega})$ of functions on $\overline{\Omega}$. For all functions $u \in \mathcal{U}$ we define

$$\begin{aligned} \|u\|_\Lambda &:= \max(\|\Delta u\|_{\infty, \Omega}, \|u\|_{\infty, \partial\Omega}) \\ \|u\|_{\mathcal{U}} &:= 2 \cdot \max_{|\alpha| \leq 2} \|D^\alpha u\|_{\infty, \overline{\Omega}} \end{aligned}$$

to get $\|u\|_\Lambda \leq \|u\|_{\mathcal{U}}$. Let there be a $W^2(\Omega)$ -regular true solution u^* , and assume that we have an approximate solution $v \in \mathcal{U} = C^2(\overline{\Omega})$ with

$$\begin{aligned} \Delta v &\approx f && \text{in } \Omega \\ v &\approx \varphi && \text{on } \partial\Omega \end{aligned}$$

such that Δv is still continuous in $\overline{\Omega}$ and v is continuous on $\partial\Omega$. Let w be the solution of

$$\begin{aligned} \Delta w &= 0 && \text{in } \Omega \\ w &= v - \varphi && \text{on } \partial\Omega. \end{aligned}$$

Then, by the maximum principle, we have

$$\|w\|_{\infty, \Omega} \leq \|w\|_{\infty, \partial\Omega} = \|v - \varphi\|_{\infty, \partial\Omega},$$

and we get the error bound

$$\|u^* - v\|_{\infty, \Omega} \leq \|u^* - v + w\|_{\infty, \Omega} + \|w\|_{\infty, \Omega} \leq \|u^* - v + w\|_{\infty, \Omega} + \|v - \varphi\|_{\infty, \partial\Omega}.$$

The function $u^* - v + w$ has vanishing boundary values and thus lies in $W_0^2(\Omega)$. With the Sobolev embedding inequality we then get

$$\begin{aligned} \|u^* - v + w\|_{\infty, \Omega} &\leq C_S \|u^* - v + w\|_{W^2(\Omega)} \\ &\leq C_S C_A \|\Delta(u^* - v + w)\|_{L_2(\Omega)} \\ &= C_S C_A \|\Delta(u^* - v)\|_{L_2(\Omega)} \\ &= C_S C_A \|f - \Delta v\|_{L_2(\Omega)} \\ &\leq C_S C_A \sqrt{\text{vol}(\Omega)} \|f - \Delta v\|_{\infty, \Omega} \end{aligned}$$

and finally

$$\begin{aligned} \|u^* - v\|_{\infty, \Omega} &\leq \|v - \varphi\|_{\infty, \partial\Omega} + C_S C_A \sqrt{\text{vol}(\Omega)} \|f - \Delta v\|_{\infty, \Omega} \\ &\leq (1 + C_S C_A \sqrt{\text{vol}(\Omega)}) \max(\|v - \varphi\|_{\infty, \partial\Omega}, \|f - \Delta v\|_{\infty, \Omega}) \\ &= (1 + C_S C_A \sqrt{\text{vol}(\Omega)}) \|v - u^*\|_{\Lambda}. \end{aligned}$$

Under the above assumptions, there is a constant C , independent of $u^* \in W^2(\Omega)$ and $v \in C^2(\Omega)$, such that

$$\|v - u^*\|_{\infty, \Omega} \leq C \|v - u^*\|_{\Lambda}$$

holds. This altogether proves (2.3) for $\mathcal{U}_A = C(\bar{\Omega})$ with norm $\|\cdot\|_A = \|\cdot\|_{\infty}$. \square
Note that our assumption of well-posedness replaces standard assumptions like uniform ellipticity of the underlying differential operator. Thus we can handle much more general situations.

3. Approximation in Trial Spaces. Now let \mathcal{U}_{ϵ} 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

$$\|u - v_{u, \epsilon}\|_{\Lambda} \leq \epsilon \|u\|_{\mathcal{U}} \quad \text{for all } u \in \mathcal{U} \quad (3.1)$$

with a small number $\epsilon > 0$. This is satisfied for subspaces that come from a sequence of subspaces getting dense in \mathcal{U} , e.g. for Kansa-type trial spaces with sufficiently dense trial centers. One will, however, need a little more regularity in \mathcal{U} than the regularity to define $\|u\|_{\Lambda}$ to get such a bound. Let us explain how to derive (3.1) in special cases. Details and proof techniques are in the recent book [16].

EXAMPLE 3.1. Take a smooth positive definite radial basis function ϕ and define a translation-invariant kernel Φ on \mathbb{R}^d via

$$\Phi(x - y) := \phi(\|x - y\|_2), \quad x, y \in \mathbb{R}^d.$$

Its smoothness is measured via Fourier transforms and a positive real number β satisfying

$$\hat{\Phi}(\omega) \leq C(1 + \|\omega\|_2)^{-d-\beta} \quad \text{for all } \omega \in \mathbb{R}^d.$$

Then there is a unique “native” Hilbert space \mathcal{U} isomorphic to $W^{(d+\beta)/2}(\mathbb{R}^d)$ in which Φ is a reproducing kernel. Now take a large set X of scattered centers in a bounded domain $\Omega \subset \mathbb{R}^d$ such that the fill distance or mesh norm

$$h := \sup_{y \in \Omega} \min_{x \in X} \|x - y\|_2$$

is small. For each $u \in \mathcal{U}$ consider the standard interpolant $s_{u,X}$ based on translates $\Phi(\cdot - x_j)$ with respect to the data in X . Then there is a pointwise error bound

$$|(D^\alpha u)(x) - (D^\alpha s_{u,X})(x)| \leq C_\alpha h^{\beta/2 - |\alpha|} \|u\|_{\mathcal{U}} \text{ for all } u \in \mathcal{U}, x \in \Omega, 0 \leq |\alpha| < \beta/2.$$

where C is a constant depending only on \mathcal{U} and the domain, but not on X, x, h, α , and u . We apply this for all functionals $\lambda \in \Lambda$ if they are in strong form, i.e. evaluations of operators acting on u and evaluated at a point, like $\mu(u) := D_x^\alpha u := (D^\alpha u)(x)$. Then it is clear that (3.1) holds, provided that regularity of the solution and smoothness of the kernel are good enough to guarantee $\beta/2 - |\alpha| > 0$ for all occurring derivative orders $|\alpha|$ in the strong functionals of the test set Λ . Again, our assumptions are much more general than in standard theories, because we just require a reasonably good approximation of the true solution by the trial space, no matter how it is defined.

4. Convergence. Now we describe the basic convergence argument, but we start in a rather abstract setting and do not even specify whether we apply weak or strong testing or how we do the numerical calculations. Using the subspace $\mathcal{U}_\epsilon \subset \mathcal{U}$ provided by the previous section, we would like to construct a function $v_\epsilon \in \mathcal{U}_\epsilon$ that solves

$$v_\epsilon = \arg \min_{v \in \mathcal{U}_\epsilon} \|v - u^*\|_\Lambda = \arg \min_{v \in \mathcal{U}_\epsilon} \sup_{\lambda \in \Lambda} |\lambda(v) - \lambda(u^*)| \quad (4.1)$$

over all $v \in \mathcal{U}_\epsilon$. We do not know whether the minimum is attained and how to calculate it, but we know that there is a function $v_\epsilon^* \in \mathcal{U}_\epsilon$ with

$$\|v_\epsilon^* - u^*\|_\Lambda \leq 2\|v_{u^*,\epsilon} - u^*\|_\Lambda \leq 2\epsilon\|u^*\|_{\mathcal{U}} \quad (4.2)$$

which is sufficient for our purpose. Though almost trivial, we state

THEOREM 4.1. *Let \mathcal{U} be a normed linear space with norm $\|\cdot\|_{\mathcal{U}}$, 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 $\|\cdot\|_\Lambda$ is defined on \mathcal{U} with (2.1) and (2.2). Assume further that the general interpolation problem (1.1) is well-posed, i.e. that $\|\cdot\|_\Lambda$ is a norm. Let $\{\mathcal{U}_\epsilon\}_\epsilon$ be a scale of subspaces of \mathcal{U} for $\epsilon \rightarrow 0$ such that for all $u \in \mathcal{U}$ there is a $v_{u,\epsilon} \in \mathcal{U}_\epsilon$ with (3.1). For all $\epsilon \rightarrow 0$, take a function v_ϵ^* satisfying (4.2). Then the error bound (4.2) holds and there is convergence $\|v_\epsilon^* - u^*\|_\Lambda \rightarrow 0$. If well-posedness holds via (2.3) for a space \mathcal{U}_A , then convergence and error bound can both be rewritten in terms of $\|\cdot\|_A$. The same results hold for solutions v_ϵ of (4.1). \square*

Note that convergence rates of the approximation in \mathcal{U}_ϵ spaces carry over to convergence rates of the approximate solutions v_ϵ^* , precisely as in the FEM case. In other words, the rate of convergence of the approximate solutions is exactly the one of the best approximation in \mathcal{U}_ϵ with respect to the norm $\|\cdot\|_\Lambda$ to functions in \mathcal{U} . Details can be worked out along the lines of the above examples.

5. Abstract Greedy Adaptive Algorithm. Now we want to set Theorem 4.1 to work. For ϵ fixed, we want to show how (4.2) can be solved. The space \mathcal{U}_ϵ will be a standard finite-dimensional Kansa trial space along the lines of Section 3. Following the previous section, we have to solve the semi-infinite linear optimization problem P_Λ defined by

$$\begin{aligned} \text{Minimize } & \eta \quad \text{s.t.} \\ & \lambda(v) - \lambda(u^*) \leq \eta \quad \text{for all } \lambda \in \Lambda \\ & -\lambda(v) + \lambda(u^*) \leq \eta \quad \text{for all } \lambda \in \Lambda \\ & v \in \mathcal{U}_\epsilon, \\ & \eta \in \mathbb{R}. \end{aligned} \quad (5.1)$$

to obtain the solution v_ϵ in (4.1) or at least a suboptimal function v_ϵ^* satisfying (4.2). But we want to *discretize* the test side of this optimization problem, replacing Λ above by a finite subset $\Lambda_n := \{\lambda_1, \dots, \lambda_n\}$ to get a standard finite-dimensional linear optimization problem P_{Λ_n} .

To this end, we define the following abstract *greedy adaptive* algorithm:

- For $n := 0$ define $v_0 := 0$ and $\Lambda_0 := \emptyset$.
- 1. If $\|v_n - u^*\|_\Lambda = 0$ holds, stop.
- 2. Else take $\lambda_{n+1} \in \Lambda$ with

$$\frac{2}{3}\|v_n - u^*\|_\Lambda \leq |\lambda_{n+1}(v_n - u^*)| \leq \|v_n - u^*\|_\Lambda \quad (5.2)$$

and set $\Lambda_{n+1} = \Lambda_n \cup \{\lambda_{n+1}\}$.

- 3. Solve the finite-dimensional linear optimization problem $P_{\Lambda_{n+1}}$ by a function $v_{n+1} \in \mathcal{U}_\epsilon$.
- 4. Set $n \leftarrow n + 1$ and repeat.

We now show how close this algorithm comes to solving (4.1) or (5.1) with the optimal value

$$\eta^* = \min_{v \in \mathcal{U}_\epsilon} \|v - u^*\|_\Lambda = \min_{v \in \mathcal{U}_\epsilon} \sup_{\lambda \in \Lambda} |\lambda(v) - \lambda(u^*)|.$$

THEOREM 5.1. *If the algorithm is finite, then $u^* \in \mathcal{U}_\epsilon$ holds and the true solution of the problem is reached in a finite number of steps. Otherwise*

$$\eta^* \leq \|v_n - u^*\|_\Lambda \leq 2\|v_n - u^*\|_{\Lambda_n} =: 2\eta_n \leq 2\eta^*$$

holds for infinitely many n . After a finite number of steps, the sufficient criterion

$$\|v_n - u^*\|_\Lambda \leq 2\eta_n \quad (5.3)$$

for v_n being a solution of (4.2) is satisfied.

Proof. The first statement is trivial. If the algorithm generates an infinite sequence with $\|v_n - u^*\|_\Lambda > 0$, we renormalize and get convergence of a subsequence to

$$\frac{v_n - u^*}{\|v_n - u^*\|_\Lambda} \rightarrow w \in \mathcal{U}_\epsilon$$

with $\|w\|_\Lambda = 1$, because \mathcal{U}_ϵ and u^* span a finite-dimensional space. By our selection rule,

$$\|v_n - u^*\|_\Lambda \geq \|v_n - u^*\|_{\Lambda_{n+1}} \geq |\lambda_{n+1}(v_n - u^*)| \geq \frac{2}{3}\|v_n - u^*\|_\Lambda$$

and

$$\begin{aligned} \|w\|_{\Lambda_{n+1}} &\geq \left\| \frac{v_n - u^*}{\|v_n - u^*\|_\Lambda} \right\|_{\Lambda_{n+1}} - \left\| w - \frac{v_n - u^*}{\|v_n - u^*\|_\Lambda} \right\|_{\Lambda_{n+1}} \\ &\geq \frac{2}{3} - \left\| w - \frac{v_n - u^*}{\|v_n - u^*\|_\Lambda} \right\|_\Lambda \\ &\geq \frac{1}{2} \end{aligned}$$

for sufficiently large n from our subsequence. Similarly,

$$\begin{aligned} \frac{\eta_n}{\|v_n - u^*\|_\Lambda} &= \frac{\|v_n - u^*\|_{\Lambda_n}}{\|v_n - u^*\|_\Lambda} \\ &\geq \|w\|_{\Lambda_n} - \left\| w - \frac{v_n - u^*}{\|v_n - u^*\|_\Lambda} \right\|_{\Lambda_n} \\ &\geq \|w\|_{\Lambda_n} - \left\| w - \frac{v_n - u^*}{\|v_n - u^*\|_\Lambda} \right\|_\Lambda \end{aligned}$$

implies that we have

$$\frac{\eta_n}{\|v_n - u^*\|_\Lambda} \geq \frac{1}{2}$$

and

$$\frac{1}{2} \|v_n - u^*\|_\Lambda \leq \eta_n \leq \|v_n - u^*\|_\Lambda$$

for infinitely many sufficiently large n from our subsequence. Since

$$0 \leq \eta_1 \leq \dots \leq \eta_n \leq \lim_{j \rightarrow \infty} \eta_j =: \tilde{\eta} \leq \eta^* := \inf_{v \in \mathcal{U}_\epsilon} \|v - u^*\|_\Lambda < \infty$$

we can conclude that

$$\eta^* \leq \|v_n - u^*\|_\Lambda \leq 2\eta_n \leq 2\tilde{\eta} \leq 2\eta^*$$

holds for sufficiently large n of a subsequence. If (5.3) holds, the above inequality is satisfied. Then we have (4.2) due to

$$\|v_n - u^*\|_\Lambda \leq 2\eta^* \leq 2\|v_{u^*, \epsilon} - u^*\|_\Lambda \leq 2\epsilon \|u^*\|_{\mathcal{U}}.$$

□

Note that the criterion (5.3) is rather close to being numerically available.

Theorem 5.1 shows that the selection of useful finite subsets Λ_n can be done by an adaptive greedy method, taking sequences of finite linear programs and checking their solutions on the remaining test functionals, taking in those functionals where large errors occur. The next sections will address the implementation of the above abstract algorithm.

6. Linear Optimization. With the theoretical background of the previous section, we now proceed towards the implementation of a greedy adaptive on-the-fly variant of a linear optimization algorithm, but we have to postpone adaptivity to the next section. Assume that we have a selection $\Lambda_n := \{\lambda_1, \dots, \lambda_n\}$ of test functionals and a selection $\mathcal{U}_m := \text{span}\{u_1, \dots, u_m\}$ of a space of trial functions, such that the linear optimization problem, i.e., the discretized version of (5.1), reads as

$$\begin{aligned} &\text{Minimize } \eta \quad \text{s.t.} \\ &\sum_{k=1}^m \alpha_k \lambda_j(u_k) - \lambda_j(u^*) \leq \eta, \quad 1 \leq j \leq n, \\ &-\sum_{k=1}^m \alpha_k \lambda_j(u_k) + \lambda_j(u^*) \leq \eta, \quad 1 \leq j \leq n, \\ &\alpha_k \in \mathbb{R}, \quad 1 \leq k \leq m, \\ &\eta \in \mathbb{R}. \end{aligned}$$

This is a linear discrete Chebyshev approximation problem of the form

$$\begin{aligned} \text{Minimize } & \eta \quad \text{s.t.} \\ & Ax - b \leq \eta \cdot \mathbf{1} \\ & -Ax + b \leq \eta \cdot \mathbf{1} \end{aligned}$$

written out componentwise with a matrix $A \in \mathbb{R}^{n \times m}$, $m \leq n$ and vectors $b \in \mathbb{R}^n$, $\mathbf{1} \in \mathbb{R}^n$. We explicitly reformulate the standard reduction to a dual problem in canonical form [2] because we want to add adaptivity later. First, we rewrite the constraints as

$$\begin{pmatrix} A & -\mathbf{1} \\ -A & -\mathbf{1} \end{pmatrix} \cdot \begin{pmatrix} x \\ \eta \end{pmatrix} \leq \begin{pmatrix} b \\ -b \end{pmatrix}$$

and the objective function as

$$-\eta = (\mathbf{0}, -1) \cdot \begin{pmatrix} x \\ \eta \end{pmatrix}$$

to be *maximized*. The standard dual of such a problem is

$$\begin{aligned} \text{Minimize } & (u^T, v^T) \cdot \begin{pmatrix} b \\ -b \end{pmatrix} \quad \text{s.t.} \\ & \begin{pmatrix} A^T & -A^T \\ -\mathbf{1}^T & -\mathbf{1}^T \end{pmatrix} \cdot \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -1 \end{pmatrix} \\ & u, v \geq 0 \end{aligned}$$

or, equivalently

$$\begin{aligned} \text{Minimize } & b^T(u - v) \quad \text{s.t.} \\ & A^T(u - v) = 0 \\ & \mathbf{1}^T(u + v) = 1 \\ & u, v \geq 0. \end{aligned} \tag{6.1}$$

Since the original problem is solvable, the weak and strong duality theorems hold. In addition, the complementary slackness conditions at the optimal primal and dual solutions, denoted by x^* , η^* , u^* , v^* will be satisfied. They are

$$\begin{aligned} u_j^* \cdot (Ax^* - b - \eta^* \cdot \mathbf{1})_j &= 0 \\ v_j^* \cdot (-Ax^* + b - \eta^* \cdot \mathbf{1})_j &= 0 \end{aligned}$$

while the factors are nonnegative. The upshot is that there are *active* indices j which can be grouped into two sets

$$\begin{aligned} j \in J^+ & \quad \text{if } u_j^* > 0 \text{ and } (Ax^* - b)_j = \eta \\ j \in J^- & \quad \text{if } v_j^* > 0 \text{ and } (Ax^* - b)_j = -\eta. \end{aligned}$$

If $\eta > 0$, these index sets are disjoint. This implies that the index sets are related to the signs of the nonzero components of w^* , if there are no degenerations.

We work with the revised simplex method based on (6.1). This means that (except for the startup phase, which is postponed here) we always keep the inverse C of a $(m+2) \times (m+2)$ matrix

$$C^{-1} = B := \begin{pmatrix} A_s^T & 0 \\ \mathbf{1}^T & 0 \\ b_s^T & -1 \end{pmatrix}$$

with a sign vector $s \in \{-1, +1\}^{m+1}$ such that B contains a nonsingular selection of $m + 1$ columns of

$$\mathcal{B} := \begin{pmatrix} A^T & -A^T & 0 \\ \mathbf{1}^T & \mathbf{1}^T & 1 \\ b^T & -b^T & -1 \end{pmatrix}$$

with the appropriate signs. In addition, one must store the numbers of the selected columns of \mathcal{B} , or, equivalently, the numbers of the selected test functionals. We call the selection of $m + 1$ signs and functionals occurring in B the **dual basis**. The current dual vertex is a vector $w_s \in \mathbb{R}^{m+1}$ such that

$$\begin{aligned} A_s^T w_s &= 0 \\ \mathbf{1}^T w_s &= 1. \end{aligned} \tag{6.2}$$

This means that

$$\begin{pmatrix} A_s^T & 0 \\ \mathbf{1}^T & 0 \\ b_s^T & -1 \end{pmatrix} \cdot \begin{pmatrix} w_s \\ b_s^T w_s \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} w_s \\ b_s^T w_s \end{pmatrix} = C \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

hold. Thus the dual vertex and the value of the dual objective function can be read off the penultimate column of the matrix C . The corresponding primal vertex is described by

$$\begin{pmatrix} A_s & \mathbf{1} & b_s \\ 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} x_s \\ -\eta \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ or } \begin{pmatrix} x_s \\ -\eta \\ -1 \end{pmatrix} = C^T \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6.3}$$

and can be read off the last column of C^T which is the last row of C .

The revised simplex method based on (6.1) already is an on-the-fly technique, provided that new columns of the constraint matrix are successively added, while the rows are fixed [2]. For the original problem, this means that new test functionals are introduced, while the trial space is fixed. So there are no problems when adding new test functionals to a solver running as a revised simplex method for the dual problem (6.1). We shall explain this in detail in the following section. The actual calculation matrix C (i.e. the inverse of a square part B of the full data matrix \mathcal{B}) will be of constant size for a fixed trial space.

For completeness and later use, let us describe the on-the-fly recipe. In theory, the standard full simplex tableau is the matrix $C \cdot \mathcal{B}$ which is of the form

$$\begin{pmatrix} S_+ & S_- & w_s \\ z_+^T & z_-^T & b_s^T w_s \end{pmatrix}$$

due to the factorization

$$\begin{pmatrix} A_s^T & 0 \\ \mathbf{1}^T & 0 \\ b_s^T & -1 \end{pmatrix} \cdot \begin{pmatrix} S_+ & S_- & w_s \\ z_+^T & z_-^T & b_s^T w_s \end{pmatrix} = \begin{pmatrix} A^T & -A^T & 0 \\ \mathbf{1}^T & \mathbf{1}^T & 1 \\ b^T & -b^T & 0 \end{pmatrix}. \tag{6.4}$$

Note that this tableau is never stored. The crucial row for the simplex method is the final one, containing the vectors

$$z_+^T = b_s S_+ - b^T, \quad z_-^T = b_s S_- + b^T.$$

It is representable as $e_{n+2}^T C \mathcal{B}$, but we know that $e_{n+2}^T C$ contains the primal data $(x_s^T, -\eta, -1)$. This means that

$$(z_+^T, z_-^T, b_s^T w_s) = (x_s^T, -\eta, -1) \cdot \begin{pmatrix} A^T & -A^T & 0 \\ \mathbf{1}^T & \mathbf{1}^T & 1 \\ b^T & -b^T & 0 \end{pmatrix}$$

holds, i.e.

$$\begin{aligned} Ax_s - b - \eta \mathbf{1} &= z_+ \\ -Ax_s + b - \eta \mathbf{1} &= z_- \\ -\eta &= b_s^T w_s. \end{aligned} \tag{6.5}$$

The tableau is optimal iff all components of the z vectors are nonpositive. This means $\|Ax_s - b\|_\infty \leq \eta$, as expected.

7. Adaptive Linear Optimization. We now know that we have to look for a test functional $\lambda \in \Lambda$ with a large positive value of

$$|\lambda(U_m)^T \cdot x_s - f_\lambda| - \eta$$

with $(\lambda(u_1), \dots, \lambda(u_m)) =: \lambda(U_m)^T$. This can be done without storing any matrix data, and the search for a good test functional can be made on an extremely large set of test functionals. Going for the largest possible value can be called a “greedy” strategy. This is the common choice in linear optimization, and it has been used favourably [11] also in meshless adaptive PDE solvers based on unsymmetric collocation. But note that (5.2) is satisfied if we only make sure that our new functional λ has the property

$$|\lambda(U_m)^T \cdot x_s - f_\lambda| \geq \frac{2}{3} \sup_{\mu \in \Lambda} |\mu(U_m)^T \cdot x_s - f_\mu|,$$

i.e. it suffices to find a reasonably good suboptimal functional to make our error bounds and convergence results valid.

Once we have such a functional, we generate the corresponding column of \mathcal{B} , multiply it by C and get the corresponding column of the full tableau. We have to make sure that the corresponding entry of z_- or z_+ is positive. In view of (6.5) we go for the left-hand part of \mathcal{B} , if

$$\lambda(U_m)^T \cdot x_s - f_\lambda - \eta$$

is positive. This sign information must be stored together with the identification or index for λ . Up to here, we have found the column of the pivot for the next step of the simplex algorithm. The pivot row is now generated in the standard way, comparing entries of the pivot column to the final column of the tableau, which is the penultimate column of C . Such a pivot row must exist, because otherwise our discrete linear optimization would be unsolvable. This determines a Gauss–Jordan transformation, which is carried out on C and *not* on the tableau. In fact, since the standard Gauss–Jordan transformation can be written as multiplication of the tableau from the left by an $(m+2) \times (m+2)$ matrix J , we can use (6.4) to see that we only need to multiply C^{-1} with J^{-1} from the right. This means that we can update C by forming $J \cdot C$. This finishes the step of the revised simplex method.

The above algorithm runs on a fixed trial space and can handle extremely large amounts of test functionals. It does not need to arrive at a full minimum, in particular

if the current value of $\eta = -b_s^T w_s$ is too large to be tolerated for the given problem. Note that for any intermediate situation we have the inequality

$$\eta \leq \inf_{u \in U_m} \sup_{\lambda \in \Lambda} |\lambda(u) - f_\lambda|. \quad (7.1)$$

This follows from the weak duality theorem in linear optimization. In fact, if u is any function from the trial space with coefficient vector $x \in \mathbb{R}^m$ in the basis u_1, \dots, u_m , we have $(A_s x - b_s)^T w_s = -b_s^T w_s = \eta$ and find

$$\eta \leq \|w_s\|_1 \|A_s x - b_s\|_\infty = \|A_s x - b_s\|_\infty \leq \sup_{\lambda \in \Lambda} |\lambda(u) - f_\lambda|$$

because $A_s x - b_s$ is a signed collection of values $\lambda(u) - f_\lambda$ with those functionals λ which constitute the actual rows of A_s .

In (7.1) one would finally get equality if the revised simplex algorithm is carried out indefinitely with U_m fixed. But if η is at some stage already rather large to be tolerated, it does not make sense to work all the way towards a full optimization with respect to all test functionals, because η will increase all the time. It is more advisable to enlarge the trial space to make smaller values of η possible at all. This is to be investigated in the next section.

8. Adaptivity of Trial Spaces. We focus now on a situation where there are no new test functionals to be introduced while the error level η is still too large. This calls for an enlargement of the trial space. Adding a new trial function means adding a column to A , and at this point we do not assume anything about that choice. One can use radial basis functions of any shape or scale, finite elements, or singular functions to cope with boundary singularities. However, adding a column to A means adding a row to A^T , and consequently the actual vertex solution of the dual problem (6.1) cannot be used any more. A complete restart should be avoided. We thus look for an on-the-fly technique that makes use of the current optimal vertex of (6.1) while adding a new row to A^T , and which continues effectively.

We first comment on the selection of a new row a^T of A^T . The current dual vector w_s has the property $A_s w_s = 0$, and a^T would be useless if $a^T w_s = 0$. This quantity has the form

$$a^T w_s = \sum_{j=1}^n w_j s_j \lambda_j(u)$$

if a comes out of a new trial function u , and where λ_j and s_j are the functionals and signs of the current dual basis. In this way, one can ask for an “optimal” new trial function u that maximizes this quantity under some normalization of u , e.g. $\|u\|_\Lambda := \sup_{\lambda \in \Lambda} |\lambda(u)| = 1$. In the following, we only assume $a^T w_s \neq 0$ and denote our new trial function by u_{m+1} .

At this point, one could restart the whole process, using the above test functionals first before introducing others. This would mean a new startup of the whole algorithm.

To avoid a restart, we have to produce a new enlarged version of the square matrices B and C . But then we also need an additional column. This means that we have to add another test functional, too. But since we already have a good mechanism for finding good test functionals, we add the zero functional as a dummy that will be

eliminated in the next simplex step. The enlarged B matrix will be

$$B' := \begin{pmatrix} A_s^T & 0 & 0 \\ a^T & 0 & 0 \\ \mathbf{1}^T & 1 & 0 \\ b_s^T & 0 & -1 \end{pmatrix}$$

which is nonsingular because $a^T w_s \neq 0$ and $A_s^T w_s = 0$ ensures that a^T is not a linear combination of the rows of A_s^T . The inverse C of B has the form

$$\begin{pmatrix} C_s & w_s & 0 \\ b_s^T C_s & b_s^T w_s & -1 \end{pmatrix}$$

with

$$\begin{aligned} A_s C_s &= I \\ \mathbf{1}^T C_s &= 0^T. \end{aligned}$$

Then the inverse of B' is

$$\begin{pmatrix} C'_s & \alpha \cdot w_s & 0 & 0 \\ z^T & -\alpha & 1 & 0 \\ b_s^T C'_s & \alpha \cdot b_s^T w_s & 0 & -1 \end{pmatrix}$$

with

$$\begin{aligned} \alpha &= \frac{1}{a^T w_s} \\ z^T &= -\alpha a^T C_s \\ C'_s &= C_s + w_s z^T. \end{aligned}$$

This is an easy $\mathcal{O}(m^2)$ recipe to upgrade C .

Inspection of the new matrix C' via (6.3) shows that the new value of η will be zero and that the old primal coefficients are the same, the new coefficient being $\alpha b_s^T w_s$. If u and u' are the old and new primal solutions written as trial functions, then

$$\lambda(u') = \lambda(u) + \alpha b_s^T w_s \lambda(u_{m+1})$$

can be used as a simplified technique for evaluating new test functionals on the new solution, if the $\lambda(u)$ were stored.

Looking at the system (6.2) and using Cramer's rule, we see that

$$|\det B'| = \left| \det \begin{pmatrix} A_s^T \\ a^T \end{pmatrix} \right| = \left| \det \begin{pmatrix} A_s^T \\ \mathbf{1}^T \end{pmatrix} \right| \cdot |a^T w_s| = |\det B| \cdot |a^T w_s|.$$

This gives a hint to stabilize the full process. One can keep track of the determinant of B and select a (or, equivalently, the new trial function) such that $|\det B'|$ is as close to 1 as possible. This strategy was quite successful in the adaptive greedy method of [11].

We also suggest the method of [11] for startup. It will make a selection of n test functionals $\lambda_1, \dots, \lambda_n$ and n trial functions u_1, \dots, u_n to provide exact generalized interpolation for a function u from the trial space. The inverse \tilde{C} of the $n \times n$ matrix $\tilde{A} = (\lambda_j(u_k))$ will be available when the algorithm of [11] stops.

We then find another functional λ_{n+1} such that $\lambda_{n+1}(u) \neq f_{\lambda_{n+1}}$ and define the $(n+1) \times n$ matrix $A = (\lambda_j(u_k))$ of the standard form of this paper. Then $A^T = (\tilde{A}^T, z)$ is known, where z is the vector with components $\lambda_{n+1}(u_j)$. With $v := \tilde{C}^T z$ we have $\tilde{A}^T v - z = 0$ and $w := (v^T, -1)^T$ is in the nullspace of A^T . From this vector, we derive a sign vector s by taking the signs of its components, and we renormalize the vector to generate a vector w_s of nonnegative components such that $\|w_s\|_1 = 1$ and $A_s^T w_s = 0$. This means that the signs of the components of w now occur as signs associated to columns of A_s^T and that we can write $A_s^T = (\tilde{A}^T, z) \cdot D_s$ with a diagonal matrix D_s carrying the signs in its diagonal. This way we get the system (6.2). Using the sign vector, we can form b_s and have the matrix B that we need.

To get $C = B^{-1}$ cheaply, it suffices to get the inverse of

$$\begin{pmatrix} A_s^T \\ \mathbf{1}^T \end{pmatrix} = \begin{pmatrix} (\tilde{A}^T, z) \\ s^T \end{pmatrix} \cdot D_s$$

cheaply. This can be done at $\mathcal{O}(n^2)$ cost from the inverse \tilde{C} of the $n \times n$ matrix \tilde{A} in the standard rank-one update way.

9. Numerical Demonstrations. In this section, we show some numerical examples that demonstrate the efficiency of our proposed algorithm. In all presented 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 $c > 0$ is the *scaling parameter*. As a test equation, we solve the Poisson problem with Dirichlet boundary conditions on $\Omega = [-1, 1]^2$, i.e.

$$\begin{aligned} \Delta u(x) &= f(x) & \text{for } x \in \Omega \subset \mathbb{R}^d, \\ u(x) &= g(x) & \text{for } x \in \partial\Omega. \end{aligned} \tag{9.1}$$

The functions f and g in (9.1) are generated by the exact solution

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

First, the test problem is run on $\Omega = [-1, 1]^2$ with different shape parameters c in order to demonstrate the performance of the adaptive greedy scheme in [11] and the proposed adaptive greedy linear optimization scheme of this paper. The number of trial functions and test functionals offered to the algorithms for selection are $M = 8822$ and $N = 2129$, respectively, but note that the algorithms select much smaller subsets adaptively. In Figure 10.1(a), the RMS errors of both schemes are given. Most of the tested c values are unusual for the original RBF collocation method, since in the standard set-up the shape parameters are usually chosen to have roughly the same order as the mesh-norm of the test functionals in order to circumvent the problem of ill-conditioning [9].

It turns out that the new linear optimization scheme is more stable in comparison to the adaptive greedy scheme. In particular, when $c = 4.75$, the adaptive greedy scheme fails to provide an acceptable solution, whereas the new linear optimization scheme performs reasonably well. We observe, from Figure 10.1(a), that there is an optimal shape parameter value of $c = 0.75$ for this particular example,

Furthermore, we show the number m of selected trial centers in Figure 10.1(b). Recall that the new adaptive linear optimization scheme uses the selected m trial centers and finally performs an ℓ^∞ fit to *all* $N \gg m$ available test functionals. In contrast, the original adaptive greedy scheme of [11] calculates an exact interpolation using the selected m trial functions and $m = n \ll N$ selected test functionals.

When the shape parameter c increases, the number of trial centers decreases. Moreover, the distributions of the selected trial centers change. As shown in Figure 10.2, in which we show the trial center distributions of two cases when $c = 1$ and $c = 5$, the adaptive algorithm tends to select trial centers away from the domain when the shape parameter is large.

Our next test demonstrates the effect of increasing the number of offered test functionals. This test is run on an L -shaped domain

$$\Omega = \{(x, y) : -1 \leq x, y \leq 1, \text{sign}(x) + \text{sign}(y) \leq 0\}.$$

For all runs, the trial space is adaptively selected from a fixed set of $M = 34692$ trial functions. Four shape parameters $c = 0.5$, $c = 1$, $c = 5$, and $c = 10$ are tested.

Figure 10.3 shows the corresponding RMS errors versus different numbers of test functionals. In all cases, large numbers of test functionals have no significant influence on the error. This is in line with our theory, since the error is dominated by the approximation properties of the trial space alone. Enlarging the set of test functionals will stabilize the algorithm. In fact, the error profiles for all tested c values remain stable even though the non-square background matrix increases in size, which usually causes ill-conditioning in the original formulation.

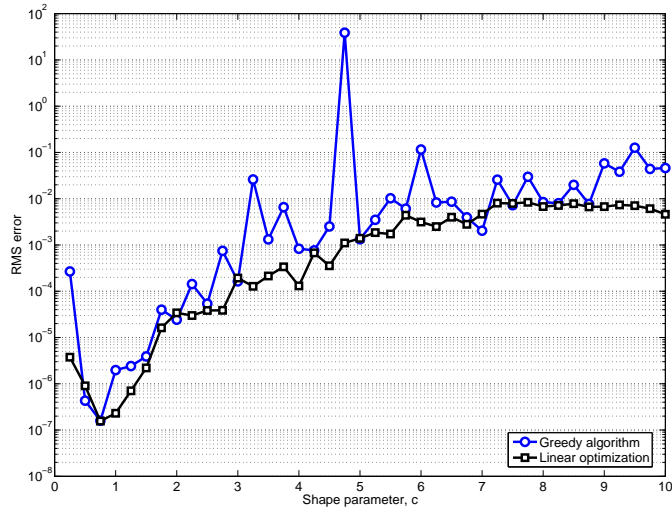
Lastly, Figure 10.4(a) and Figure 10.4(b) show typical error functions of the adaptive greedy method and the proposed linear optimization method, respectively. Since the adaptive greedy method solves the selected small square submatrix system in the sense of exact interpolation, its error function shows clear peaks. In contrast, the linear optimization method handles the overdetermined system with the selected trial functions and *all* test functionals in such a way that the maximal error in the differential equation and the boundary values is minimized on very many points. Thus the error profile usually is smaller in magnitude and more oscillatory with less obvious peaks.

10. Conclusion. On the theoretical side, we prove that the unsymmetric meshless collocation method, if carried out with smooth kernel-based trial functions and sufficiently many test functionals, converges at the same rate as interpolation of the solution. In particular, trial spaces formed by multiquadrics and Gaussian basis would result in exponential convergence [12] if the solution is analytic. However, from the computational point of view, exponential convergence is difficult to be observed under double precision computation. Our proof techniques are much simpler than those of [14] because we focus on minimizing the maximum of the residuals via semi-infinite linear optimization.

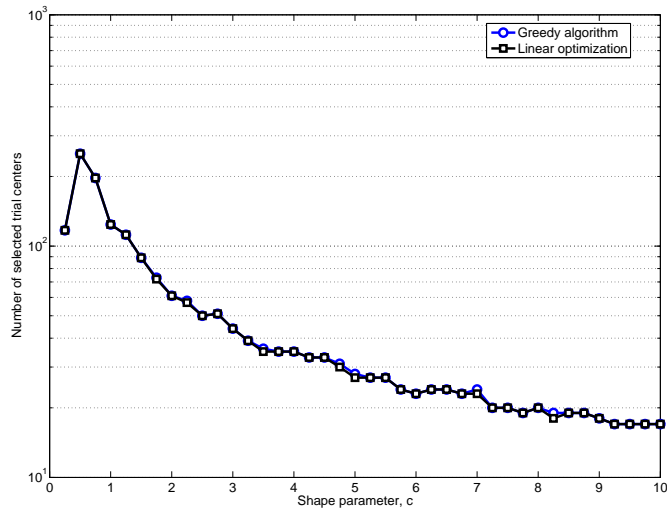
On the practical side, we couple the previously proposed greedy technique of [11] with linear optimization to provide a fully adaptive on-the-fly data-dependent method guaranteeing solvability and fitting seamlessly into our theory. Numerical results show that the proposed coupled method is very stable, even though serious ill-conditioning of the resultant systems must be expected.

REFERENCES

- [1] S. N. ATLURI AND S. P. SHEN, *The Meshless Local Petrov-Galerkin (MLPG) Method*, Tech Science Press, Encino, CA, 2002.
- [2] I. BARRODALE AND A. YOUNG, *Algorithms for best L_1 and L_∞ linear approximations on a discrete set*, Numer. Math., 8 (1966), pp. 295–306.
- [3] C. FRANKE AND R. SCHABACK, *Convergence order estimates of meshless collocation methods using radial basis functions*, Adv. Comput. Math., 8 (1998), pp. 381–399.
- [4] ———, *Solving partial differential equations by collocation using radial basis functions*, Appl. Math. Comput., 93 (1998), pp. 73–82.
- [5] M. GOLBERG AND C. CHEN, eds., *The dual reciprocity method and radial basis functions*, Elsevier, Amsterdam, 2000. Eng. Anal. Boundary Elements **24** (2000), no. 7-8.
- [6] Y. C. HON AND R. SCHABACK, *On unsymmetric collocation by radial basis functions*, Appl. Math. Comput., 119 (2001), pp. 177–186.
- [7] 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 (1990), pp. 127–145.
- [8] ———, *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 (1990), pp. 147–161.
- [9] E. J. KANSA AND Y. C. HON, *Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations*, Comput. Math. Appl., 39 (2000), pp. 123–137.
- [10] O. A. LADYZHENSKAYA, *The boundary value problems of mathematical physics*, vol. 49 of Applied Mathematical Sciences, Springer-Verlag, New York, 1985. Translated from the Russian by Jack Lohwater [Arthur J. Lohwater].
- [11] L. LING, R. OPFER, AND R. SCHABACK, *Results on meshless collocation techniques*, Eng. Anal. Boundary Elements, 30 (2006), pp. 247–253.
- [12] W. R. MADYCH, *Miscellaneous error bounds for multiquadric and related interpolators*, Comput. Math. Appl., 24 (1992), pp. 121–138.
- [13] R. SCHABACK, *On the versatility of meshless kernel methods*, in Advances in Computational & Experimental Engineering & Sciences, S. Atluri, D. Beskos, and D. Polyzos, eds., no. 428, 2003.
- [14] ———, *Convergence of unsymmetric kernel-based meshless collocation methods*, SIAM J. Numer. Anal., (2005). to appear.
- [15] ———, *Unsymmetric meshless methods for operator equations*. Preprint, 2006.
- [16] H. WENDLAND, *Scattered Data Approximation*, Cambridge University Press, 2005.
- [17] Z. M. WU, *Hermite-Birkhoff interpolation of scattered data by radial basis functions*, Approx. Theory Appl., 8 (1992), pp. 1–10.



(a) Error profile.



(b) Degrees of freedom.

FIG. 10.1. Solving a Poisson equation with MQ kernel on a square domain: Error profile and degrees of freedom as a function of shape parameter c .

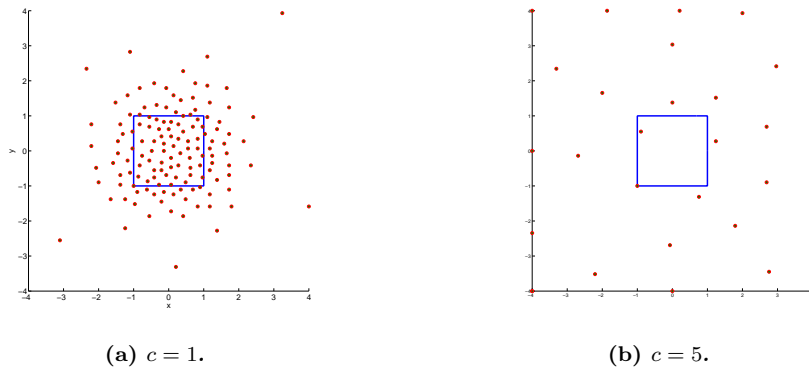


FIG. 10.2. Solving a Poisson equation with MQ kernel on a square domain: Selected trial centers for different shape parameters c .

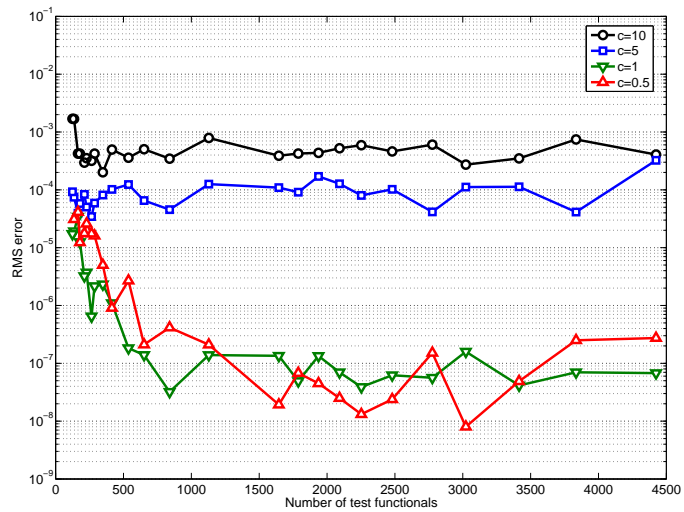
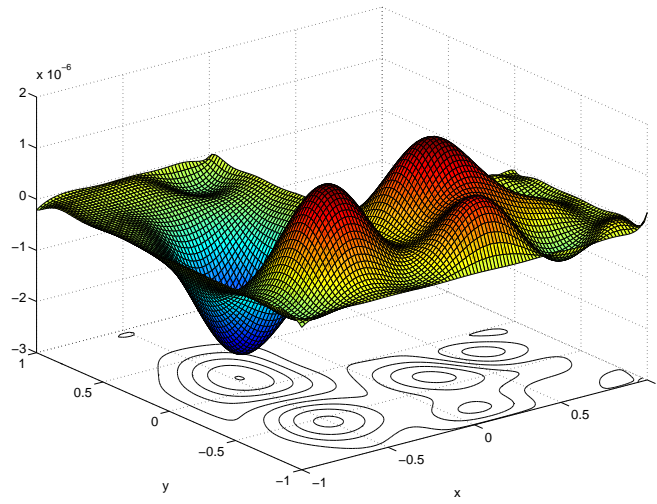
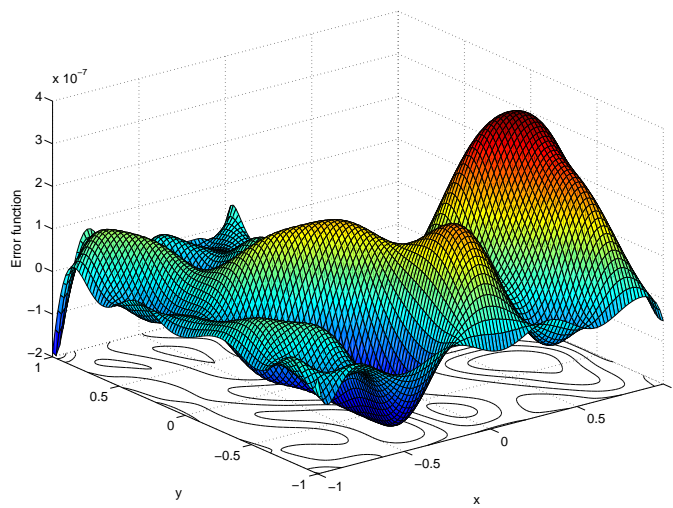


FIG. 10.3. Solving a Poisson equation with MQ kernel on an L-shaped domain: Error profile as a function of numbers of test functionals.



(a) *Adaptive greedy.*



(b) *Linear optimization.*

FIG. 10.4. Solving a Poisson equation with MQ kernel on an L-shaped domain: Error functions.